

# Estimating dynamic cutting forces of machine tools from measured vibrations using sparse regression with nonlinear function basis

Yongzhi Qu<sup>1</sup> and Gregory W. Vogl<sup>2</sup>

<sup>1</sup>*University of Minnesota Duluth, Duluth, MN, 55804, USA*

*yongzhi@umn.edu*

<sup>2</sup>*National Institute of Standards and Technology, Gaithersburg, MD, 20899, USA*

*gvogl@nist.gov*

## ABSTRACT

Estimating relationships between system inputs and outputs can provide insight to system characteristics. Furthermore, with an established input-output relationship and measured output, one can estimate the corresponding input to the system. Traditionally, the relationship between input and output can be represented with transfer functions or frequency response functions. However, those functions need to be built on physical parameters, which are hard to obtain in practical systems. Also, the inverse problem of solving for the input with a known/measured output is often more difficult to solve than the forward problem. This paper aims to explore the data-driven input-output relationship between system inputs and outputs for system diagnostics, prognostics, performance prediction, and control. A data-driven relationship can provide a new way for system input estimation or output prediction. In this paper, a sparse linear regression model with nonlinear function basis is proposed for input estimation with measured outputs. The proposed method explicitly creates a nonlinear function basis for the regression relationship. A threshold-based sparse linear regression is designed to ensure sparsity. The method is tested with experimental data from a spindle testbed that simulates cutting forces within machine tools. The results show that the proposed approach can predict the input force based on the measured vibration response with high accuracy. Force prediction errors are around 5 percent on randomly reserved testing data and about 10 percent to 15 percent for unforeseen testing data from the training dataset. The prediction results are also compared with neural networks, which represents a more complicated nonlinear regression method.

---

Qu et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

## 1. INTRODUCTION

Online system identification is an important task in system prediction and control. In the past, model-based system identification was a major focus. For example, Kalman filter-based techniques can be employed to estimate system parameters and perform prediction. In recent years, data-driven system identification has drawn much research attention due to the expressive power of machine learning. Dynamic mode decomposition (DMD) (Schmid, 2010) and Koopman operator theory (Williams et al., 2015) are two examples of data-driven approaches for dynamical system modeling. A dynamical system refers to a system in which a function describes the time dependence of the evolution process of the state variables. DMD provides a decomposition method to learn a matrix-based representation of a high-dimensional linear system. The Koopman operator further provides a framework that converts the problem of learning a representation of a nonlinear system into learning an infinite-dimensional linear system representation. Physics-informed neural networks (PINN) have been developed as another popular approach that aims to include physics during the machine learning process (Raissi et al., 2019). The major benefit of PINN lies in that the imposed physics constraint can help the machine learning algorithm converge faster with a relatively small dataset while respecting physical rules to a certain extent. The purpose of PINN is to learn an expressive model for the time-evolving function of a dynamical system, which can then be used for response prediction or model parameter identification.

However, most of the existing data-driven modeling of system dynamics mainly focus on dynamical systems in which the evolution of the system is governed solely by the internal properties of the system, without external excitations, and following zero input response. Mathematically, the state evolution obeys differential equations involving time derivatives. In practice, under real-

time operating conditions, a physical system cannot be modeled as an unforced dynamical system due to the forced response from continuous external inputs. In this scenario, a time-based dynamical model is not the best for predicting the response of the system. Instead, an input-output model should be obtained to model the system. Such an input-output model can be considered as a generalized frequency response function (Qu et al., 2021), which defines the input-output relationship for systems undergoing a general forced excitation. Such a model is typically hard to obtain by a physics-based approach, since many physical parameters are difficult to measure or estimate for run-time systems, especially those involving rotating contacts as in a spindle of a machine tool. To help address this problem, this paper outlines how to build a data-driven input-output relationship for run-time systems that provides useful information for estimation and control.

For example, the cutting tool within a machine tool makes dynamic contact with the workpiece and generates vibration, which further affects the cutting performance, such as the surface quality. In general, cutting forces can be considered as the excitation to the spindle system and serves as the input of the system. Therefore, estimating cutting forces is more of an inverse problem rather than a forward problem. Nonetheless, vibration sensors are often installed on the spindle housing for spindle health monitoring. A hypothesis is that if the measured vibration on the spindle housing can be correlated to the real-time cutting forces, then the cutting forces can be estimated in real-time during machining. To achieve this goal, we created a cutting simulation testbed, in which the cutting forces can be simulated and measured. Then, we set out to investigate the possibility of learning a data-driven model that can estimate cutting forces with only vibration data. The learned model may then be applied for real-time force estimation on a machining spindle if some calibration data can be collected on the machine tool.

Manufacturers desire a method to monitor cutting forces during machining processes, in order to transform machine tools into smart machine tools that are self-aware with real-time diagnostics and prognostics for asset management. If cutting forces are monitored, then tool wear, tool breakage, surface quality, and chatter can potentially be monitored in real time. Hence, production could be stopped at the moment a fault in the cutting process is detected, before a failure or violation of a production specification is breached. Currently, one difficulty in monitoring cutting forces is that they cannot be measured without invasive force sensors installed on the machine tool or underneath the workpiece. However, because of their small size, accelerometers could potentially be used to estimate cutting forces non-invasively. Accelerometers were determined to have a great potential to be integrated for use in smart spindles because they can satisfy requirements including (1) no reduction in spindle stiffnesses, (2) no limitations on cutting parameters, (3) reliable signal

transmission, and (4) long operational life and simple maintenance (Cao et al., 2017). In fact, Postel et al. (2019) used five low-cost accelerometers on a stationary spindle housing to estimate the AC-components of cutting forces and vibrations at the tool tip. Despite the lack of application for low-to-moderate spindle speeds, the method verified that spindle-mounted accelerometers can be used to estimate high-speed cutting forces and vibrations at the tool tip (Postel et al., 2019). Similarly, this paper explains a technique that uses vibrations measured by on-machine accelerometers to estimate cutting forces in real time.

Vibrations, as generally high-frequency signals, contain a lot of noise and non-stationary patterns. Moreover, machining involves time-varying inputs, so the relationship between force and vibration cannot be easily correlated in a point-to-point manner in the time domain. However, in the frequency domain, the relationship between force and vibration should be more easily determined. Therefore, in this paper, we formulate the problem of estimating the cutting force in the frequency domain as a vector regression problem from the vibration spectrum to the force spectrum.

Traditionally, the frequency response function (FRF) is defined as the ratio between the output component at a particular frequency ( $\omega_i$ ) and the same frequency component ( $\omega_i$ ) in the input signal (Feucht, 1990). However, this definition is only valid when the system is linear and does not involve frequency modulation due to rotational motion. In machine spindles, the force frequency can be modulated by the rotational transmission path, such as gears and bearings. Structural resonances are another source that create high-frequency components even under low frequency input. Also, as the spindle speed increases, the natural frequency can decrease and the damping can increase due to gyroscopic moments and centrifugal forces that cause the bearing stiffness to decrease (Grossi et al., 2017). Hence, the FRF between vibration and the cutting force depends on spindle speed. Therefore, we consider the relationship between the input spectrum and the output spectrum as a generalized FRF, where each output frequency depends on each of the input frequencies and vice versa (Qu et al., 2021). Thus, the process of the spectrum estimation is defined as a multi-variate regression problem. To this end, we propose a sparse linear regression method with nonlinear function basis to model the input and output relationship in the frequency domain.

The rest of the paper is organized as follows: Section 2 introduces the background and mathematical formulations of the methods. Section 3 presents our experimental setup and the flow chart of the proposed methodology, while Section 4 discusses the results with the proposed methodology. Finally, Section 5 compares the results with those from neural networks, and Section 6 concludes the paper.

## 2. METHODOLOGY

### 2.1. Related Background

In this section, a sparse linear regression model for input-output modeling is proposed. As discussed in Section 1, the input-output relationship learning can be formulated as a vector-to-vector regression problem. Intuitively, we can consider each dimension in the output vector as independent and formulate the problem as a multiple linear regression problem. The original problem can be formulated as a vector function:

$$Y^m = F^{\mathbb{R}^n \rightarrow \mathbb{R}^m}(X^n) \quad (1)$$

If we take each dimension of  $Y$  separately, it becomes:

$$Y_i = f^{\mathbb{R}^n \rightarrow \mathbb{R}}(X^n), i \in (1, 2, \dots, m) \quad (2)$$

where the superscripts  $(n, m)$  denote the dimension of the variable,  $Y^m$  means  $Y \in \mathbb{R}^m$  and  $X^n$  means  $X \in \mathbb{R}^n$ .  $F^{\mathbb{R}^n \rightarrow \mathbb{R}^m}$  means  $F$  is a function that maps values from space  $\mathbb{R}^n$  to  $\mathbb{R}^m$ , similar with  $f^{\mathbb{R}^n \rightarrow \mathbb{R}}$ .

We seek to estimate a functional relationship between  $Y$  and  $X$ . If a linear relationship suffices to model the relationship between  $Y$  and  $X$ , then we can perform a linear regression between each dimension of  $Y$  and  $X$ . In the case that a linear relationship is insufficient, one can build a nonlinear model, such as a neural network, for  $Y_i$  and  $X$  in the form of  $Y_i = f(w * (f(... f(w * X))))$ , which is essentially a kernel-based nonlinear regression model. However, depending on the size of the hidden layer in the neural network, a large number of samples may be required to train such a neural network model. Neural network results will be provided in Section 5 for comparison.

Recently, in physics-informed machine learning, sparse identification of nonlinear dynamics (Sindy) (Brunton et al., 2016) has been proposed to approximate nonlinear differential equations with linear regression and a nonlinear function basis. In Sindy, the nonlinear terms can be constructed with a nonlinear function basis to enrich the ‘observation’ and achieve a linear relationship between states. Inspired by the work of ‘Sindy’, we propose to use linear regression with nonlinear function basis for the general multi-variate regression.

In the original work, Sindy was designed to learn a functional representation of the time-evolving function of a dynamical system. For example, a generic dynamical system can be written as:

$$\frac{d}{dt}x = f(x) \quad (3)$$

Without losing generality,  $x$  is assumed to be in  $\mathbb{R}$ . In the discretized form, Eq. (3) can be written as:

$$x_{k+1} = f(x_k) \quad (4)$$

In the Sindy algorithm, a library of candidate nonlinear functions  $\Theta(x)$  can be constructed from  $x$ ,

$$\Theta(x) = [1, x, x^2, \dots, x^d, \sin(x), \cos(2x), \dots] \quad (5)$$

where  $\Theta(x)$  can be constructed with any number of arbitrary functions. Then, a regular linear regression can be performed as

$$\dot{x} = \Theta(x)\xi \quad (6)$$

where  $\xi$  represents the coefficient for each of the basis functions. While  $\xi$  is solved using the measured data of  $x$ ,  $\xi$  also provides the weights for the basis functions as expressed in  $\Theta(x)$ . Therefore, when  $\xi$  is solved, it is equivalent that the nonlinear function  $f(x)$  can be represented as  $\Theta(x)\xi$  with a nonlinear functional basis.

### 2.2. Proposed Approach- Linear Regression with Nonlinear Function Basis (LR-NFB)

In this paper, we extend the Sindy framework to a general multi-variate linear regression problem. Instead of Eq. (2), we use:

$$Y_i = f^{\mathbb{R}^{n \times K} \rightarrow \mathbb{R}}(g_1(X), g_2(X), \dots, g_K(X)) \quad (7)$$

where  $g_i$  ( $i = 1, 2, \dots, K$ ) is an arbitrary nonlinear function, with  $i \in (1, 2, \dots, K)$ , and  $f^{\mathbb{R}^{n \times K} \rightarrow \mathbb{R}}$  is a linear function from space  $\mathbb{R}^{n \times K}$  to  $\mathbb{R}$ . Equation (7) can be rearranged as:

$$Y_i = G(X)\zeta, \zeta \in \mathbb{R}^{K \times n} \quad (8)$$

where  $G(X)$  represents the augmented data from  $X$ , and  $\zeta$  is the regression coefficients. Compared with a neural network, where the nonlinear activation is taken on the weighted sum of all dimensions, the linear regression model with nonlinear function basis allows nonlinear mappings to be taken on each dimension of the input separately and therefore provides more flexibility to handle nonlinearities.

Because the dimension of the original vector has been augmented  $K$  times in Eq. (8), it may seem that more parameters have been added to the model. However, compared with a neural network, the number of parameters is kept at a moderate level in Eq. (8). For example, for a 200-dimensional input vector, the augmented dimension in the proposed approach will be 1000 when  $K = 5$ , which indicates that 1000 linear parameters will need to be learned. However, with a shallow neural network with one hidden layer, the number of parameters is  $201 * 5 + 6 * 1 = 1011$  when there are only 5 nodes in the hidden layer. The number of parameters will increase dramatically when the number of nodes in the hidden layer is greater than 5. With 100 hidden nodes in the hidden layer, the total number of parameters for a neural network will be  $201 * 100 + 101 * 1 = 20201$ . For a deep neural network, the total number of parameters will further increase. In general, as a rule of thumb, the number of samples should be close to the number of parameters in the model to have the

model well converged and avoid overfitting. To this end, the LR-NFB problem will be overdetermined rather than underdetermined as long as the number of samples is greater than approximately 1000 in the given example.

### 2.3. Sparse Regularization of LR-NFB (SLR-NFB)

It is possible that the artificially created nonlinear function basis may not be useful for the regression task. In practice, many of the coefficients are typically effectively zero and should not be included in the final model. Therefore, a sparse penalty should be added to the model to ensure sparsity. An L1 norm is a common regularization term that can be added to the loss function, such as mean square error of the prediction, as:

$$\zeta = \operatorname{argmin}_{\zeta'} \|Y - G(X)\zeta\|_2 + \lambda\|\zeta\|_1 \quad (9)$$

However, in practice, the optimization of Eq. (9) is computationally expensive when the dimension of input is large. Instead, a sequential threshold least squares (STLS) algorithm, which is proposed in the original Sindy method (Brunton et al., 2016), is adopted for the sparse regularization. STLS can be summarized with the following pseudo code:

---

Input:  $G(X)$ ,  $Y$ , coefficient threshold  $\gamma$ , iteration  $N$

- 1: Compute  $\zeta$  as  $G(X)\backslash Y$  as initial value
  - 2: **for**  $iteration = 1, 2, \dots, N$  **do**
  - 3:   set  $\zeta[i] = 0$ , if  $\zeta[i] < \gamma$
  - 4:   **for**  $dimension = 1, 2, \dots, \text{length}(Y)$  **do**
  - 5:      $G(X)\backslash Y$  on entries of  $G(X)[i]$  if  $\zeta[i] \neq 0$
  - 6:   **end for**
  - 7: **end for**=0
- 

Figure 1. Pseudo code of the sequential threshold least squares (STLS) algorithm.

## 3. EXPERIMENTAL SETUP AND METHODOLOGY

### 3.1. Spindle Testbed

A new method for estimating real-time cutting forces was realized in a spindle testbed, shown in Figure 2. A motor rotates an instrumented tool holder in a custom spindle above a metrology suite. The instrumented tool holder is the “tool” and the metrology suite is the “workpiece”. Accordingly, the metrology suite measures the “tool-to-workpiece” forces, torques, and error motions during rotation of the tool holder. A data acquisition (DAQ) box collects the data from the metrology suite sensors as well as data from two triaxial inertial sensors attached to the custom spindle housing. Six accelerations were sensed from two Dytran 7576A1 tri-axial vibration sensors, and the forces were sensed from two ATI Mini45 force-torque sensors. The sampling rate for both force and vibration was set at 10 kHz. In this paper, only one X-

axis vibration signal and the total X-axis force were used for model development.

For the purposes of explaining the testbed data, Figure 3 shows an illustration of the portable components of the spindle testbed on a machine tool. As the tool holder rotates, the inertial sensors attached to the spindle housing measure the spindle housing vibrations. Simultaneously, a magnet on the tool holder interacts with magnets fixed on two force-torque sensors in the metrology suite; the “tool-to-workpiece” forces and torques are also measured during rotation. Figure 4 shows how the force-torque sensors are attached to linear positioning stages so that the magnetic interactions can be changed via the separation distance,  $d$ .

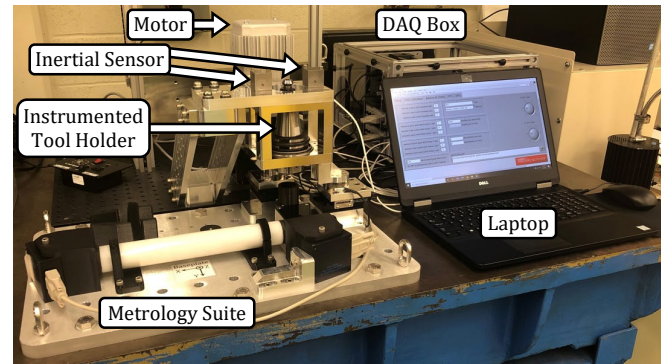


Figure 2. Spindle testbed.

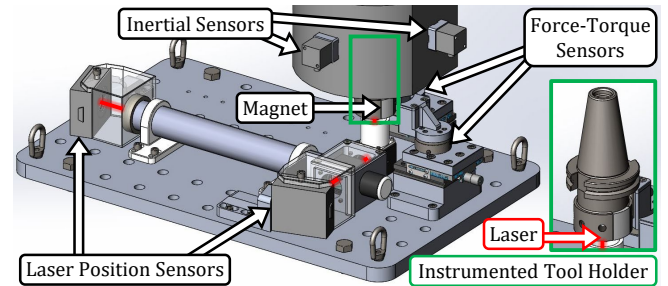


Figure 3. Illustration of application of the new method on a machine tool.

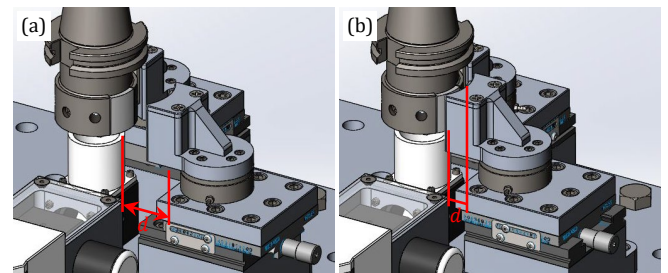


Figure 4. Illustration of a magnetic assembly located to yield a (a) relatively low tool-to-workpiece force or (b) relatively high tool-to-workpiece force.

### 3.2. Method Overview

Whether data is collected on the spindle testbed or on a machine tool, all signals are collected simultaneously and processed to yield the 30 signals shown in Figure 5: the “A” prefix is for an acceleration, the “E” prefix is for an error motion, the “F” prefix is for a force, the “G” prefix is for a rate velocity, and the “T” prefix is for a torque. The metrology suite yields the 3 forces and 3 torques (FX1, FY1, FZ1, TX1, TY1, and TZ1) associated with the X-direction force-torque sensor, the 3 forces and 3 torques (FX2, FY2, FZ2, TX2, TY2, and TZ2) associated with the Y-direction force-torque sensor, the 2 translational error motions (EXC and EYC) of the laser diode in the “tool”, and the 2 rotational error motions (EAC and EBC) of the laser diode in the “tool”. The two inertial sensors yield the 6 inertial signals (AX1, AY1, AZ1, GX1, GY1, and GZ1) associated with the first sensor and the 6 inertial signals (AX2, AY2, AZ2, GX2, GY2, and GZ2) associated with the second sensor.

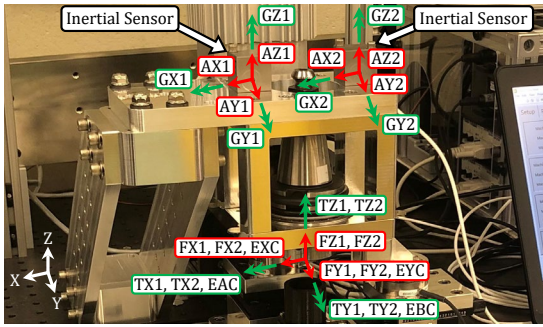


Figure 5. Close-up view of instrumented tool holder and inertial sensors in spindle testbed with associated signals.

Figure 6 shows a summary of a new method for estimating real-time cutting forces. Before machining, on-machine sensors (e.g., accelerometers), an instrumented tool holder, and a metrology suite are placed within the machine tool. The instrumented tool holder in the spindle works in tandem with the metrology suite to induce simulated cutting forces during rotation, per the spindle testbed explained in Section 3.1. The forces are composed of three orthogonal components, to represent a general cutting force, composed of the cutting force (parallel to the velocity), the thrust force (perpendicular to the velocity and spindle axis), and the axial force (aligned with the spindle axis). Any component could be estimated by the new method, but the thrust force component, in the X-axis direction, is estimated by the new method within this manuscript, because it is the largest component created within the spindle testbed.

During rotation, data is collected simultaneously and synchronously from the on-machine sensors and the metrology suite for a variety of force levels and rotational speeds. This dataset is then processed via machine learning to create an estimator of the tool-to-workpiece forces based solely on the on-machine sensor data as inputs. Finally, during machining, the instrumented tool holder and

metrology suite are removed from the machine, and a different tool holder with the same taper mechanism is placed in the spindle. As the tool rotates and removes material from the workpiece, the on-machine sensor data is inputted into the estimator, which outputs a real-time estimation of the forces. In this sense, the new method may transform a conventional spindle into a smart spindle for future smart machine tools.

The new method for simulating and monitoring cutting forces has several advantages over existing methods. Not only does the method not affect the cutting process, but as long as inertial sensors can be mounted on the spindle housing, the method can be applied to all spindles and tools that utilize tool holders with the same taper mechanism as the instrumented tool holder.

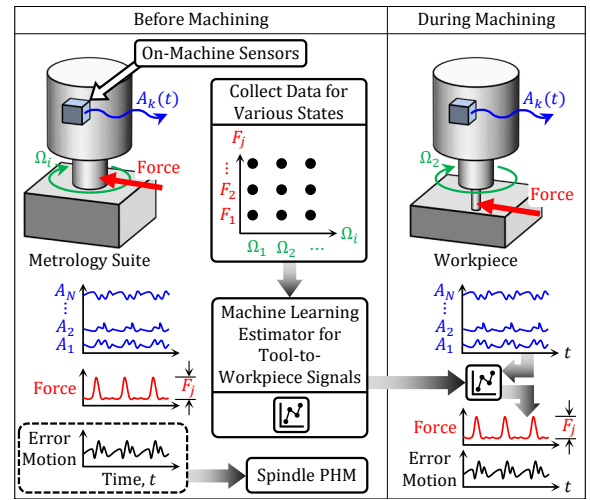


Figure 6. Summary of new method for estimating real-time cutting forces.

### 3.3. Overall Methodology

Again, the purpose of this research is to estimate the cutting force from the measured vibration on the spindle housing. In order to fully evaluate the feasibility of using a data-driven approach to build an inverse FRF model that can be used to estimate the cutting force under various spindle speeds and force levels, spindle data was collected for 100 different spindle speeds and 13 different force levels. Table 1 summarizes the dataset. There are a total of 1300 different operating conditions at which data was collected, with spindle speeds ranging from 400 rpm to 3000 rpm, selected uniformly every 26 rpm, and thrust force impulse magnitudes ranging from 4 N to 40 N, selected uniformly every 3 N. For each operating condition, 1 s of data was collected.

To create training data and testing data, we divide the data into 0.5-s-long segments, which doubles the number of samples to 2600 samples. Each sample contains the actual force signal and the corresponding vibration signal in the time domain. Then, a Fourier transform is taken to convert the temporal data to the frequency domain. These single-sided



Fourier spectrums have a frequency resolution of 2 Hz, due to the sample length of 0.5 s. The DC components are not utilized, however, to focus on the AC response of the dynamical system.

Table 1. Summary of dataset.

Force Speed	4 N	7 N	...	37 N	40 N
400 rpm	X	X	...	X	X
426 rpm	X	X	...	X	X
...	...	...	...	...	...
2974 rpm	X	X	...	X	X
3000 rpm	X	X	...	X	X

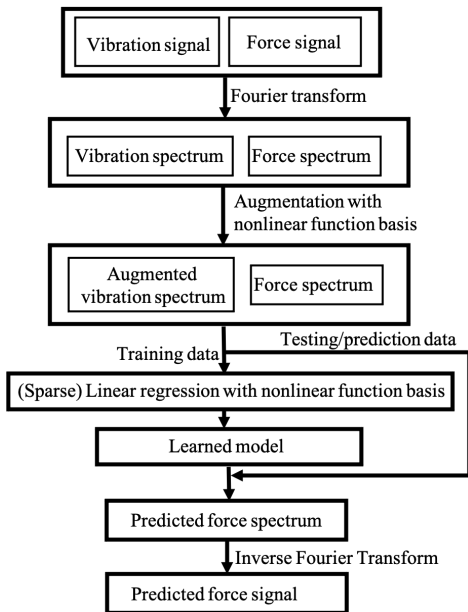


Figure 7. Flow chart of the overall methodology to estimate the cutting force.

It is worth noting that in the implementation process, the complex spectra are separated into their real and imaginary parts, which are concatenated together to form a double-length real vector as an input with the function basis augmentation. The regression method with function basis can be directly applied to complex numbers as well. In our testing, it was found that the complex-number regression result is similar or slightly inferior to the real-number regression. A possible reason might be that some of the basis functions are not well defined for complex numbers. The route of directly working with complex numbers is not presented due to its inferior results. Figure 7 shows the flow chart of the overall methodology.

In the initial testing, four basis functions were chosen, namely, square root, square, natural exponential function, and the original form (first order). We also included a

constant term to accommodate for a bias. The proposed model was run on a MacBook-Air laptop in matlab. The computer is configured with 1.6 GHz core i5 CPU and 8GB of memory. Detailed results will be shown and discussed in Section 4.

## 4. RESULTS AND DISCUSSIONS

This section is organized as the following: In Section 4.1, we first show examples of raw data collected from the spindle testbed, which illustrates the data and gives an overview of the force prediction problems. Section 4.2 presents three test cases that illustrate how the model is learned and predicts the forces. Finally, in Section 4.3, the effect of different basis functions is briefly discussed.

### 4.1. Overview of the Dataset

Figure 8 shows examples of force and vibration signals. Due to the varying operating conditions, the vibration-to-force relationship in the frequency domain under different spindle speeds are very different at the same frequency. Thus, the regression model for the force should take into consideration the operating conditions and amplitude relationships between different frequencies. Our hypothesis is that those relationships are hidden in the vibration spectrum and can be utilized if the regression model has enough expressive power.

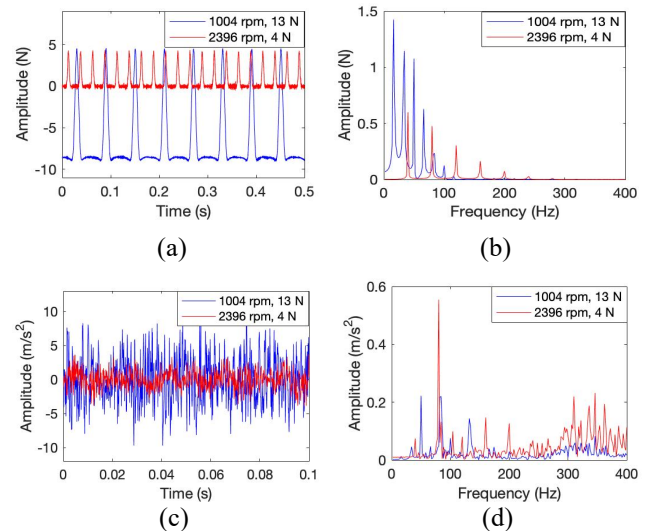


Figure 8. Example force and vibration signals and their spectrums: (a) the force signals, (b) the force spectrums, (c) the vibration signals, and (d) the vibration spectrums for two spindle states with different spindle speeds and force levels.

### 4.2. Three Test Cases and Results

To investigate the possibility of learning a generalized data-driven FRF, testing is performed for three different scenarios:

- Case 1: The full set of data is randomly separated into 70 percent for training and 30 percent for testing and prediction. In this case, it is largely guaranteed that similar operating conditions either under the same spindle speed or under the same force level have been seen in the training dataset. 780 samples out of 2600 samples are reserved for testing.
- Case 2: The training is only performed with some force levels. Then the model is used to predict the forces at untrained force levels. More specifically, 16 N, 19 N, 34 N, and 40 N out of the thirteen (13) different force levels are never trained under any spindle speeds. It should be mentioned that 40 N is the highest force level in this experiment. Omitting all training data for 40 N will make it a fully out-of-range prediction. 800 samples out of 2600 samples are reserved for testing and prediction.
- Case 3: The training data set only includes some spindle speeds. The prediction is then performed on unforeseen spindle speeds. In this test, data under spindle speeds of 505 rpm, 1004 rpm, 1503 rpm, 2002 rpm, 2501 rpm, and 3000 rpm are never trained. This is the most difficult case, since the spindle can have dramatically different FRFs at different spindle speeds. 156 samples out of 2600 samples are used for testing and prediction.

Next, we present the results for the above three cases. We also present results for regular linear regression (LR), linear regression with nonlinear function basis (LR-NFB), and sparse linear regression with nonlinear function basis (SLR-NFB).

Figure 9 shows example predictions for Case 1, with the solid black line as the true force, the blue line as the results of linear regression (LR), the red line represents results for LR-NFB, and the green line shows the SLR-NFB results. It can be seen from the results that the multi-variate regression-based approach can effectively predict the cutting force based on the vibration response. It can also be seen that the proposed regression approaches with nonlinear function basis generate more accurate prediction results than simple linear regression.

Quantitative comparisons among the methods are given in Table 2. In this error table and all subsequent error tables, the values are expressed as mean  $\pm$  standard deviation in newtons (N). Two metrics are evaluated to compare the force prediction accuracy. The first one is Mean Absolute Error (MAE), which is computed as absolute error between the predicted force versus true force, and then the average over all tested cases is reported. MAE not only evaluates the amplitude error between the true and predicted forces, but also includes the phase shift error. The second metric is the mean root mean square error (M-RMSE). To calculate M-RMSE, we first calculate the absolute error between the RMS for each sample and the RMS for each predicted force, and

then take the average across all samples. The RMS value of a force signal represents the effective force in the cutting process, similar to the effective value of a voltage signal in electrical engineering. Both mean error and standard deviation of the error distribution are given in Table 2 and the following tables.

Comparison of the MAE and M-RMSE metrics for the three methods (LR, LR-NFB, SLR-NFB) in Table 2 shows that LR-NFB outperforms simple linear regression. The sparse version of the algorithms further improved the accuracy slightly. It is noted that the prediction accuracy is an average error for all force levels from 4 N to 40 N. For example, the force prediction error for SLR-NFB of 1.15 N is roughly 5 percent for the median force level of 22 N for the dataset. Such small differences are illustrated in Figure 10, which shows the true force and SLR-NFB-based predicted force from Figure 9. Therefore, the average prediction error is potentially acceptable for force monitoring purposes within manufacturing.

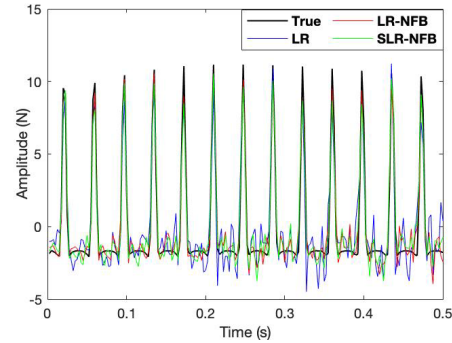


Figure 9. Example predicted force using vibration response with three different regression approaches.

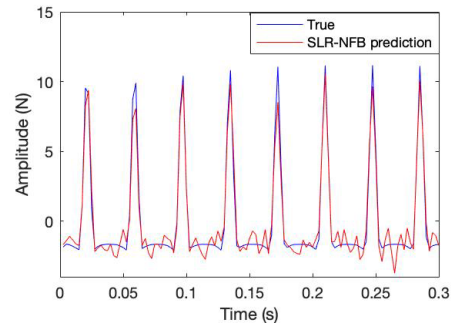


Figure 10. Example predicted force with SLR-NFB.

Table 2. Testing error for reserved testing data.

Metric	LR	LR-NFB	SLR-NFB
MAE	1.43 $\pm$ 1.95	1.18 $\pm$ 1.77	<b>1.15 <math>\pm</math> 1.73</b>
M-RMSE	0.43 $\pm$ 0.53	0.34 $\pm$ 0.53	<b>0.34 <math>\pm</math> 0.53</b>

We continue to show the results for Case 2, in which the prediction is performed on force levels that have never been trained. Table 3 shows the prediction accuracies via the MAE

and M-RMSE metrics for the three methods. First, as expected, the prediction error increased compared with Case 1. However, the prediction accuracy is still very high without significant performance degradation. Second, the prediction errors with the nonlinear function basis are slightly smaller than those without the nonlinear function basis. However, we also notice that the error standard deviation for LR-NFB and SLR-NFB are larger than that of the LR model, which indicate that the predictions of the SLR-NFB model have more fluctuation.

Table 3. Prediction error for untrained force levels.

Metric	LR	LR-NFB	SLR-NFB
MAE	$1.61 \pm 2.21$	$1.60 \pm 2.93$	<b><math>1.56 \pm 2.82</math></b>
M-RMSE	$0.54 \pm 0.57$	$0.51 \pm 1.16$	<b><math>0.50 \pm 1.10</math></b>

Nonetheless, the SLR-NFB method was shown to be accurate for force predictions at 40 N, which is out of the trained force range and represent extrapolating. Figure 11 shows an example of the prediction results for 40 N, which are observed to be highly accurate.

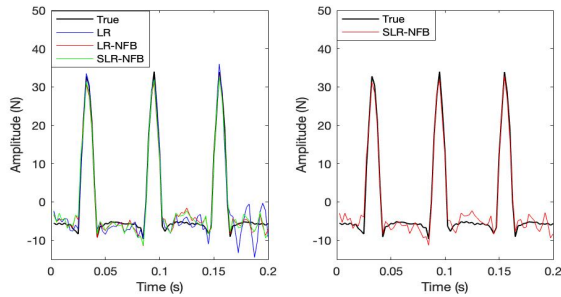


Figure 11. Example predicted force for 40 N, which was untrained and out of the trained force range.

Next, Table 4 shows the prediction performances for Case 3, in which the prediction is performed for spindle speeds that have never been trained. In general, the prediction error metrics increase for the unforeseen speed levels compared to those for the untrained force levels (Case 2). The major reason is perhaps the extrapolation limitations of the models without physics. First, from a mechanical point of view, under different spindle speeds, the FRF function can be dramatically different due to different physical properties from different bearing contact stiffnesses and motor influences. Second, the vibration signal during rotation is dominated by harmonics at the spindle speed and its integer multiples, leading to a sub-accurate relationship at untrained speeds away from the trained harmonic frequencies. Nonetheless, because the accuracy of around 3 N represents a force prediction error of roughly 15 percent, the model demonstrates potential for data-driven FRF learning and application for estimating machining forces that vary by relatively large magnitudes.

However, compared with simple linear regression, the prediction performance decreased slightly. One of the

possible reasons for the performance degradation could be associated with overfitting of the model. Although the linear regression problem is overdetermined, with a sparse penalty, the solution space is significantly larger than the number of model parameters. Another possible reason for the performance degradation is overfitting by dominant dependencies in the model, such as a higher-order term. This can be potentially solved by adding an L2 regularizer that limit dominant weights on the function basis.

Table 4. Prediction error for untrained spindle speeds.

Metric	LR	LR-NFB	SLR-NFB
MAE	<b><math>2.90 \pm 4.15</math></b>	$3.60 \pm 5.84$	$3.56 \pm 5.72$
M-RMSE	<b><math>0.76 \pm 0.89</math></b>	$1.16 \pm 2.21$	$1.13 \pm 2.12$

Table 5 shows a summary of the performance improvement of using sparse regularization with linear regression with nonlinear function basis (SLR-NFB). It can be seen that the proposed SLR-NFB model generally outperforms linear regression model except for the test case of untrained spindle speeds.

Table 5. Force prediction accuracy improvement of SLR-NFB over linear regression model.

Metric	Testing data	Untrained force	Untrained speed
MAE	<b>19.6 %</b>	<b>3.11 %</b>	-22.8 %
M-RMSE	<b>20.9 %</b>	<b>7.41 %</b>	-48.7 %

While the sparse version of SLR-NFB only sees about 2.5% of improvement over the non-sparse version of LR-NFB in our tested case, it is expected in future work that the improvement will be more significant with an increased number of basis functions.

### 4.3. Evaluation of Different Function Basis

It is interesting to see how the function basis affects the performance of force prediction. In a simple evaluation, we tested  $\cos(2x)$  as a replacement of the exponential function in the initial function basis. The results are shown in Table 6 to Table 9. Table 9 shows that the results are very similar as those for the initial test (see Table 5). It appears that the regression model is relatively not very sensitive to the highest-order term in the function basis.

Table 6. Testing error for reserved testing data.

Metric	LR	LR-NFB	SLR-NFB
MAE	$1.42 \pm 1.92$	$1.21 \pm 1.80$	<b><math>1.18 \pm 1.75</math></b>
M-RMSE	$0.44 \pm 0.49$	$0.38 \pm 0.52$	<b><math>0.38 \pm 0.52</math></b>



Table 7. Prediction error for untrained force levels.

Metric	LR	LR-NFB	SLR-NFB
MAE	1.61 ± 2.21	1.63 ± 3.25	<b>1.58 ± 3.17</b>
M-RMSE	0.53 ± 0.57	0.49 ± 1.34	<b>0.48 ± 1.29</b>

Table 8. Prediction error for untrained spindle speeds.

Metric	LR	LR-NFB	SLR-NFB
MAE	<b>2.90 ± 4.15</b>	3.30 ± 5.36	3.22 ± 5.26
M-RMSE	<b>0.76 ± 0.88</b>	1.12 ± 2.16	1.14 ± 2.18

Table 9. Force prediction accuracy improvement of SLR-NFB over linear regression model.

Metric	Testing data	Untrained force	Untrained speed
MAE	<b>16.9 %</b>	<b>1.9 %</b>	-11.0 %
M-RMSE	<b>13.6 %</b>	<b>9.4 %</b>	-50.0 %

## 5. COMPARISON WITH DEEP NEURAL NETWORK

Different neural networks were tested for regression and compared against the models discussed in the previous sections. It was found that the deep neural network outperformed the shallow neural network. A fully connected feedforward neural network with four layers was created and tested for the task. The number of nodes in each sequential layer is 200, 100, 100, and 200. It is worth mentioning that tanh activation functions were adopted, while sigmoid and rectified linear unit (ReLU) activation functions do not converge in the training stage.

To compare the computation cost of the proposed model and deep neural networks, we timed the training process for both models on Case 1, where 70% of data out of 2600 samples was used for training. It took 111.2 sec to run the sparse regression model with 10 iterations (similar to epochs). In contrast, it took about 148.5 seconds for the neural networks to converge with 500 epochs.

Table 10 shows the prediction results for the deep neural network. It is seen that the prediction performance of the neural network is not as good as that for linear regression for the given problem. The major reason is that the number of samples is relatively small for the neural network to be fully trained for the large solution space. Another reason is that for the regression model, each output dimension is treated independently, and effectively, a linear model for each dimension is then trained. However, it is unrealistic to train and save a neural network for each output dimension. Therefore, the neural network model represents a large vector-to-vector regression model.

Table 10. Prediction accuracy of deep neural network.

Metric	Case 1	Case 2	Case 3
MAE	1.68 ± 2.28	1.91 ± 2.62	3.24 ± 4.61
M-RMSE	0.52 ± 0.67	0.74 ± 0.75	0.82 ± 0.98

Table 11 compares the results of the SLR-NFB model with those from the deep neural network. In general, the SLR-NFB model performs better than the deep neural network, by as much as about 50 percent.

Based on the above observations, it is inferred that the linear model has the best extrapolation capability, while the SLR-NFB model has a more accurate expressive capability compared with the linear model. It can also be inferred that with a relatively small dataset, neural networks may not be a good candidate for this type of learning task.

Table 11. Force prediction accuracy improvement of SLR-NFB over deep neural network.

Metric	Testing data	Untrained force	Untrained speed
MAE	<b>46.1 %</b>	<b>23.2 %</b>	-8.9 %
M-RMSE	<b>52.9 %</b>	<b>48.0 %</b>	-27.4 %

## 6. CONCLUSIONS

In this paper, a study of cutting force prediction with measured vibration signals was performed. More specifically, acceleration data collected on the spindle housing of a testbed, which was designed to simulate cutting processes on a machine tool, were used to predict the testbed “cutting” force signals. Regression methods with a nonlinear function basis were proposed for data-driven input-output relationship learning. The proposed methods were tested with experimental data collected from a spindle testbed. The learned model represents a data-driven inverse frequency response function.

The results demonstrate that the proposed methods can use the acceleration signals to effectively predict the cutting force with good accuracy. It was further shown that the proposed regression method (SLR-NFB) outperforms linear regression on the testing dataset with random training/testing. SLR-NFB also outperformed linear regression for force predictions at unforeseen spindle forces. However, the performance of the SLR-NFB model was not better than linear regression for unforeseen spindle speeds, which indicates that there was a relative lack of extrapolation capability for SLR-NFB. Finally, in comparison with deep neural networks, linear regression performed better in all tested cases. SLR-NFB performed better than the deep neural network in all cases, except for force predictions at untrained spindle speeds. The proposed model is not optimized in the sense of overfitting, so for future work, an L2 regularizer will be included to improve the extrapolation capability. Furthermore, hybrid

model like physics informed neural network will be investigated for the FRF learning.

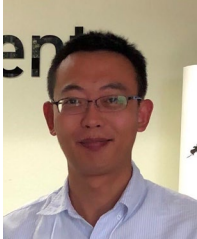
#### ACKNOWLEDGEMENT

The authors thank Brian Pries (NIST, Retired), Jarred Nace, Justin Cullum, Marvin Kingsbury, Mike Beacham, Rick Lake, and Travis Shatzley of the Fabrication Technology Office (NIST) for their outstanding contributions with the experimental setup. Yongzhi Qu acknowledges the Department of Commerce of the United States for partial support of this research under contract No. 70NANB20H175, and partial support from the University of Minnesota in Grant-in-Aid of Research, Artistry and Scholarship (GIA).

#### NIST DISCLAIMER

Certain commercial equipment, instruments, or materials are identified in this paper in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States. Approved for public release; distribution is unlimited.

#### BIOGRAPHIES



**Yongzhi Qu** received the Bachelor's degree in measurements and control and Master's degree in measurement and testing from Wuhan University of Technology in 2008 and 2011, and Ph.D. in Industrial Engineering and Operations Research from the University of Illinois Chicago in 2014.

Dr. Qu is currently an assistant professor in Mechanical and Industrial Engineering at the University of Minnesota Duluth. Dr. Qu's research interests include data-driven discovery of system dynamics, physics-informed machine learning, signal processing, system diagnostics and prognostics.



**Gregory W. Vogl** received the Bachelor's degree in Engineering Science and Mechanics and Master's degree and Ph.D. in Engineering Mechanics from Virginia Polytechnic Institute and State University of Blacksburg, Virginia, USA in 2000, 2003, and 2006, respectively. He

designed, fabricated, and experimented on microelectromechanical systems as a National Research Council Postdoctoral Researcher at the National Institute of Standards and Technology (NIST). He then joined the Production Systems Group at NIST, in which he worked on standards development for machine tools and vibration

metrology for accelerometers. Dr. Vogl is currently a member of the Monitoring, Diagnostics, and Prognostics for Manufacturing Operations (MDP4MO) project, which seeks to develop methodologies, protocols, and reference datasets to enable robust real-time diagnostics and prognostics for smart manufacturing systems. He develops sensor-based solutions for on-machine and real-time health assessment of machine tool linear axes and spindles. For his contributions, Dr. Vogl is the recipient of a NIST Engineering Laboratory Mentoring Award and NIST Colleagues' Choice Award.

#### REFERENCES

- Brunton, S. L., Proctor, J. L., & Kutz, J. N. (2016). Sparse identification of nonlinear dynamics. *Proceedings of the National Academy of Sciences*, vol. 113, issue 15, pp. 3932-3937. doi:10.1073/pnas.1517384113
- Cao, H. R., Zhang, X. W., & Chen, X. F. (2017). The Concept and Progress of Intelligent Spindles: A Review. *International Journal of Machine Tools & Manufacture*, vol. 112, pp. 21-52.
- Feucht, D. L., *Handbook of Analog Circuit Design*. Elsevier Science. p. 192. ISBN 978-1-4832-5938-3. (1990).
- Grossi, N., Sallese, L., Scippa, A., & Campatelli, G. (2017). Improved Experimental-Analytical Approach to Compute Speed-Varying Tool-Tip FRF. *Precision Engineering*, vol. 48, pp. 114-122.
- Postel, M., Aslan, D., Wegener, K., & Altintas, Y. (2019). Monitoring of Vibrations and Cutting Forces with Spindle Mounted Vibration Sensors. *CIRP Annals-Manufacturing Technology*, vol. 68, issue 1, pp. 413-416.
- Qu, Y., Vogl, G. W., & Wang, Z. (2021). A Deep Neural Network Model for Learning Runtime Frequency Response Function Using Sensor Measurements. *Proceedings of the ASME 2021 16th International Manufacturing Science and Engineering Conference*. June 21-25, Cincinnati, OH.
- Raissi, M., Perdikaris, P., Karniadakis, G.E., (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *Journal of Computational Physics*, vol. 378, pp. 686-707, ISSN 0021-9991. <https://doi.org/10.1016/j.jcp.2018.10.045>
- Schmid, P. (2010). Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, vol. 656, pp. 5-28. doi:10.1017/S0022112010001217
- Williams, M.O., Kevrekidis, I.G. & Rowley, C.W. A Data-Driven Approximation of the Koopman Operator: Extending Dynamic Mode Decomposition. *J Nonlinear Science*, 25, 1307-1346 (2015). <https://doi.org/10.1007/s00332-015-9258-5>