Run-Time Cutting Force Estimation Based on Learned Nonlinear Frequency Response Function

Jacob Fabro
Department of Mechanical and Industrial Engineering,
University of Minnesota Duluth,
Duluth, MN 55804
e-mail: fabro001@umn.edu

Gregory W. Vogl
Production Systems Group,
National Institute of Standards and Technology,
Gaithersburg, MD 20899
e-mail: gregory.vogl@nist.gov

Yongzhi Qu[1]
Department of Mechanical and Industrial Engineering,
University of Minnesota Duluth,
Duluth, MN 55804
e-mail: yongzhi@umn.edu

The frequency response function (FRF) provides an input–output model that describes the system dynamics. Learning the FRF of a mechanical system can facilitate system identification, adaptive control, and condition-based health monitoring. Traditionally, FRFs can be measured by off-line experimental testing, such as impulse response measurements via impact hammer testing. In this paper, we investigate learning FRFs from operational data with a nonlinear regression approach. A regression model with a learned nonlinear basis is proposed for FRF learning for run-time systems under dynamic steady state. Compared with a classic FRF, the data-driven model accounts for both transient and steady-state responses. With a nonlinear function basis, the FRF model naturally handles nonlinear frequency response analysis. The proposed method is tested and validated for dynamic cutting force estimation of machining spindles under various operating conditions. As shown in the results, instead of being a constant linear ratio, the learned FRF can represent different mapping relationships under different spindle speeds and force levels, which accounts for the nonlinear behavior of the systems. It is shown that the proposed method can predict dynamic cutting forces with high accuracy using measured vibration signals. We also demonstrate that the learned data-driven FRF can be easily applied with the few-shot learning scheme to machine tool spindles with different frequency responses when limited training samples are available. [DOI: 10.1115/1.4054157]

Keywords: smart manufacturing, data-driven dynamics, frequency response function, few-shot regression, Industry 4.0, machine tool dynamics, machining processes, sensing, monitoring, and diagnostics

1 Introduction

1.1 Motivation. Data-driven system modeling and dynamic analysis have gained increasing popularity as more data are available for Industry 4.0. There are several focuses for data-driven dynamic modeling: (a) data-driven dynamical system modeling, (b) data-driven discovery of system dynamics, and (c) data-driven online system identification. The above focuses are closely related and sometimes overlap. However, they differ in the modeling space, i.e., the observation space, state space, or both. First, data-driven dynamical system modeling works in a high-dimensional observation space and aims to model the time-evolving dynamics with a purely data-driven approach. Dynamic mode decomposition [1] and Koopman operator theory [2] are two examples of data-driven approaches for dynamical system modeling. The major benefits of a data-driven Koopman operator-based approach are that (1) it transforms the representation of nonlinear dynamics for state variables, in the form of nonlinear differential equations, into a linear representation in terms of observation variables when lifted to the infinite dimensional observation space and (2) it provides a convenient approach for high-dimensional data modeling, such as a dynamic fluid field [3]. The Koopman operator can be learned as a high-dimensional coefficient matrix that captures the time-evolving relationship in a high-dimensional spatial field. Second, data-driven discovery of system dynamics shares similar goals as data-driven dynamical system modeling. It works in the state space directly and aims to learn a data-driven solution to system equations, such as a set of differential equations. It can also be used to learn system parameters with measured dynamical data in the state space. Data-driven discovery of system dynamics is best known as physics-informed machine learning (ML), which combines data-driven approaches with physics-based models. Physics-informed neural networks (PINNs) are among the most popular approaches in physics-informed machine learning (ML) [4]. One major benefit of PINN lies in that the imposed physics constraint can help the ML algorithm converge faster with a relatively small dataset while respecting physical rules. Third, data-driven online system identification works in both the observation space and the state space. It not only models the time-evolving patterns in the state space but also explicitly correlates state variables and observation variables. Kalman-filter-based identification is one example of system identification that can be used to estimate system parameters and perform prediction in the joint state and observation space.

This work proposes to learn the frequency response function (FRF) of a dynamic system, which can be used to represent the system dynamics and to establish an input–output mapping relationship. The proposed approach is related to focus (c) in building a correlation model, as we learn a FRF that correlates the system response with the system input, whereas focus (c) correlates the observation (or response) to state variables. While state variables are typically hidden and can be hard to measure in practical application, the system response and input are generally measurable. A learned model between the system response and the input can be used for multiple purposes, such as output prediction, system evaluation, and input estimation. Such a model will be preferrable in a real-time situation, where the input can vary with time and the system itself may change due to wear and degradation.

[1]Corresponding author.

Manuscript received December 23, 2021; final manuscript received February 25, 2022; published online April 8, 2022. Assoc. Editor: Andy Henderson.

A physics-based model that captures the relationship between an input and an output is typically hard to obtain because many physical parameters are difficult to measure or estimate. For example, there is no standard way to measure the contact stiffness and other physical parameters of a run-time dynamic system involving rotating contacts like a machine tool spindle. Also, during run-time operation, there are both steady state and transient components in the dynamic response. The traditional FRF is a function of frequency (ω) and is independent of the input amplitude. Thus, the traditional FRF yields a linear ratio of the output to the input at each frequency and represents a static view of the system dynamics. However, for machines undergoing different operating conditions, such as varying rotating speeds and force levels, the dynamic stiffness and boundary conditions can change and introduce nonlinearities [5]. Moreover, force can introduce geometric structural nonlinearities through Hertzian contact stiffness associated with bearings [6]. This leads to an intrinsic dependency of the FRF on the input data. Therefore, a classic input-independent physics-based model is not only hard to build but also insufficient to reflect the complex input–output mapping relationships.

On the contrary, machine learning models, such as a deep neural network, possess the capability to learn a versatile model that can handle a nonlinear mapping model as well as input dependencies [7]. Intuitively, machine learning algorithms are generally considered as local approximations because they always approximate a mapping relationship that is confined by the given training samples. Generalization and extrapolation can be challenging for machine learning (ML) since traditional ML models cannot approximate functional relationships and generalize the learned model from the trained domain to the full domain of a function. To learn a data-driven FRF, it is desired that the learned model can be generalized to the full domain of the FRF, which implicitly mimics the physical relationship. To address those drawbacks of machine learning, a learning approach that features regression with a learned function basis is adopted and developed for frequency response function learning.

1.2 Review. Next, we provide a short discussion on the motivation and literature review of spindle process modeling and FRF learning. Advanced manufacturing depends upon innovative sensor-based monitoring and modeling systems that are intelligent, robust, reliable, and ideally inexpensive [8]. Building models from sensory data that can assist the analysis and prediction of machining performance is one of the core tasks to realize advanced manufacturing. As an example, the quality and repeatability of the part surface produced with a milling machine tool depend on the quality and consistency of the cutting forces. Chatter, which is a particular frequency response phenomenon, can happen at certain combinations of rotational speeds and cutting forces during milling operations and can significantly affect the part quality [9]. As a cutting tool wears from usage, cutting forces can increase or vary due to worn, damaged, or broken teeth on the cutting tool [10]. As a cutting tool wears from usage, cutting forces can increase or vary due to worn, damaged, or broken teeth on the cutting tool. Such changes result in part variations as production proceeds, potentially resulting in the violation of quality specifications or damage to the part [11]. If cutting forces are known, then tool wear, surface quality, and chatter can potentially be monitored in real time.

However, monitoring cutting forces is difficult because the measurements require invasive force sensors installed on the machine tool or underneath the workpiece. The typical use of a plate dynamometer [12] for force measurements is impractical for part manufacturing due to workspace constraints and inertial forces that may corrupt the measured forces [13]. Moreover, direct force measurement approaches on tool tips are generally expensive and can introduce calibration problems for spindles. Alternatively, indirect methods for estimating cutting forces have been investigated, including the use of motor currents with structural dynamics compensation and servo information [14,15]. Accelerometers, especially the miniature type, can be hidden within machine tool spindle housings and used to estimate cutting forces. In addition, accelerometers provide the benefits of long operational life and simple maintenance [16]. Because of these advantages, spindle-mounted accelerometers have been used to estimate cutting forces. Wang et al. proposed a dynamic force identification method in the time domain using conjugate gradient least square optimization to reconstruct the milling forces using acceleration signals [17]. Postel et al. proposed a Kalman-filter-based approach to estimate the forces and displacements at the tool tip with accelerations measured on the spindle housing [18]. Even though considerable efforts have been seen to estimate cutting forces, existing works are exclusively based on the FRF identified off-line with impact hammer testing. However, these off-line FRF models can be significantly different than real-time models that change over time.

Research on real-time FRF learning and the application to cutting force estimation have been rare. Kushnir proposed estimating the run-time FRF for a lathe using the measured vibration spectrum at the top plate and spindle headstock. The FRF was represented as a vector specifying the averaged linear ratio between output and input spectrums at each frequency [19]. Deng et al. investigated a transfer learning method to obtain a mapping relationship between the frequency response and the parameters in a finite element model [20]. While their work is in line with frequency response analysis, they did not aim to learn a frequency response function. Thenozhi and Tang proposed to use a radial basis neural network and support vector machine to learn the FRF from simulation data of a single-degree-of-freedom mechanical system [21]. A regression model was built between the inputs and outputs, in the form of $\mathrm{OFR} = f(a, \omega)$, where OFR is the amplitude of the corresponding output frequency response, a is the amplitude, and ω is the frequency. While Thenozhi and Tang's work is closely related to our work, they did not build a full FRF learning algorithm with both amplitude and phase (or real and imaginary terms), which would require a multiple-input, multiple-output model structure.

Nonlinear frequency response analysis (NFRA) has drawn recent attention in battery analysis related to electrochemical impedance spectroscopy [22]. NFRA aims to give a full and detailed representation of the system response that can establish a relationship between model parameters and the experimentally observed phenomena. Nonetheless, NFRA has seen limited research in mechanical system analysis. Lang and Billings proposed the concept of nonlinear output frequency response functions (NOFRFs) [23]. As discussed in the NOFRFs approach, the system nonlinear behavior depends on a combined effect of the input spectrum and the system frequency domain characteristics. Bayma et al. further proposed a polynomial nonlinear auto-regressive with exogenous input model for output response prediction based on NOFRFs, which handled the nonlinearities based on a time-domain auto-regressive model with external input [24]. In contrast, this paper focuses on learning the nonlinear FRF in the frequency domain directly with a nonlinear function basis of the input spectrum and the corresponding frequency. In a previous work [25], we proposed to use a preselected nonlinear function basis for FRF learning. However, the model in Ref. [25] cannot effectively extrapolate to unforeseen machine conditions.

To this end, this paper outlines how to learn a data-driven nonlinear FRF and presents its application in learning a mapping model from the on-machine vibration response to the cutting force for run-time spindle systems, which can provide an effective estimation of cutting force. *To the best knowledge of the authors, this is the first work that learns a full nonlinear FRF from run-time experimental data*. The contributions of the paper are threefold: (1) we formulated and developed a FRF learning method in the frequency domain. The learned data-driven FRF represents a complex model for systems under dynamic steady state and can be used for real-time dynamic input/output prediction; (2) we proposed a multivariate regression model that can model the relationships between multiple input variables and multiple output variables and can be

applied to a general multi-input, multi-output (MIMO) system; and (3) we showed that with a few-shot FRF learning strategy, the model can be easily adopted to a new machine with a different FRF. The performance of the proposed method and the few-shot learning scheme are validated using experimental data.

The rest of the paper is organized as follows: Sec. 2 presents the related background and proposed methodology; Sec. 3 describes the experimental setup and illustrates the datasets; Sec. 4 presents the results and discusses the sensitivity of the model structures and parameters; and Sec. 5 concludes the paper. The data and codes are provided in a GitHub repository.[2]

2 Methodology

In this section, a general regression method for data-driven FRF learning of a multiple-input, multiple-output system is proposed.

2.1 Related Background. This section discusses the background of regression methods with a learned function basis to obtain a compressed representation of a general nonlinear function [26,27].

A regression task aims to learn a function that accurately maps an input to an output for all possible input values in the defined function domain. Mapping the input to the output of a complex system can be done effectively with a function basis, sometimes noted as mode decomposition, where various simple functions contribute on a weighted basis to make up the more complex function output. Function basis mapping of an unknown function $F(x)$ may be represented generally as follows [26]:

$$F(x) = \boldsymbol{P}\boldsymbol{\psi} \tag{1}$$

$$\boldsymbol{P} = [p_1, p_2, \ldots, p_n] \tag{2}$$

$$\boldsymbol{\psi} = [\psi_1, \psi_2, \ldots, \psi_n]^T \tag{3}$$

where $\boldsymbol{P}$ is a set of weights and $\boldsymbol{\psi}$ is a set of basis functions. Equation (1) can be represented as the summation of the products of the individual weights and the respective basis functions, that is,

$$Y = F(x) = \sum_i p_i \psi_i(x) \tag{4}$$

$$\psi_i(x) \in \{1,\ x,\ \sin(x),\ e^x,\ x^2 + 1, \ldots\} \tag{5}$$

Equation (5) shows that $\psi_i(x)$ can be an arbitrary function. In practice, the set of basis functions can be learned to best approximate the unknown function $F(x)$.

Furthermore, the weights, $\boldsymbol{P}$, can be learned as a solution for a given set of basis functions $\boldsymbol{\psi}$ and the target value of $F(x)$ as Y. Therefore, $\boldsymbol{P}$ can be represented as a function of $\boldsymbol{\psi}$ and Y:

$$\boldsymbol{P} = H(\boldsymbol{\psi}, Y) \tag{6}$$

$$p_i = h_i(\boldsymbol{\psi}, Y) \tag{7}$$

Next, the adopted function-basis-based few-shot regression model [26] is briefly summarized with the model architecture and how the model learns to approximate an unknown function. Figure 1 shows a diagram of the model architecture. There are three major modules in the regression model: *basis function generator* (BFG) module, *weights generator* (WG) module, and *dot product* (DP) module, as indicated by the dashed boxes. The data flow is separated into two routes: the training process, represented by the black arrows and boxes, and the testing process, represented by the blue arrows and boxes in Fig. 1. Each module is described briefly below as a step in the model.

[2]https://github.com/yongzhiqu/FRF-Learning-with-Few-Shot-Multivariate-Regression

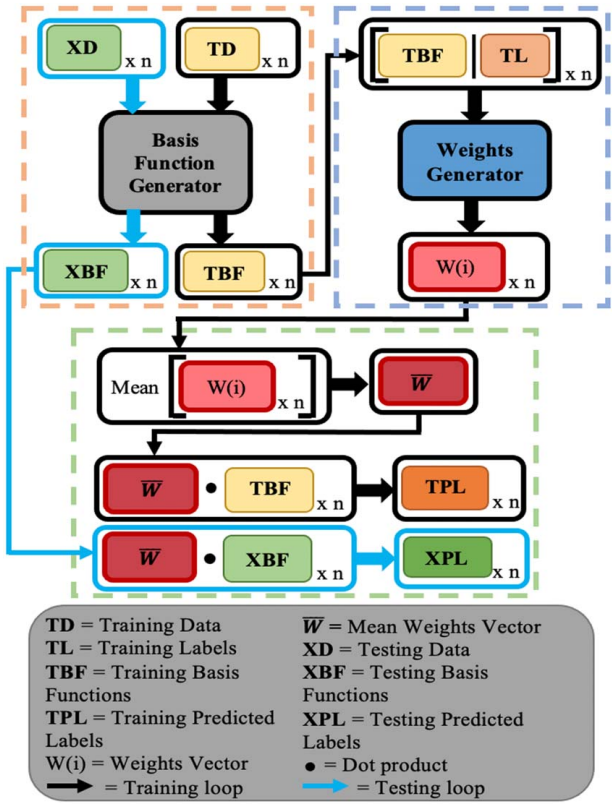

Fig. 1 Model architecture diagram

First, the BFG module serves as a space explorer, which finds the high-dimensional embedding space for the input variables. Therefore, it learns the basis functions that augment the input data. The BFG module can be implemented as a feedforward neural network. The BFG neural network is initialized randomly with a preselected number of outputs, where each output represents the result of one basis function in the high-dimensional embedding space. During the model training stage, the BFG neural network will be trained and optimized to approximate the most suitable set of basis functions.

Then, the WG module learns the weights as a function of the basis functions and the ground truth labels, according to (6). As proposed in Ref. [26], the weights can be best solved by an attention model, which explicitly encourages different levels of emphasis when mapping the inputs to the outputs in a neural network.

Finally, the DP module performs two operations. It first takes the mean of the learned weights across n points over the function domain. Taking the mean approximates a voting mechanism for the final weights that represent the underlying function across the function domain. Second, the DP module performs the dot product of the mean weights and the learned basis functions to yield the predictions. The predictions are then compared with the true labels, and the errors can be backpropagated to optimize the parameters in both the BFG and WG modules.

2.2 Problem Formulation. A FRF is defined as the ratio between the output spectrum to the input spectrum in the frequency domain of a system or device. More broadly, FRFs are special cases of transfer functions. A FRF is defined for a linear system under dynamic steady state, in which there are sinusoidal inputs and the transient response has dissipated. Obtaining the FRF of a system allows informed operation and predictive control of that system. It is noteworthy that in experimentally collected real-time operating data, there are numerous transient responses that cannot be modeled with traditional FRFs.

Nonetheless, we consider the nominal operating state of a real-time system as dynamic steady state, and without losing generality, we will use the traditional FRF model to illustrate the proposed method. A general FRF is

$$H(\omega) = \frac{Y(\omega)}{X(\omega)} \tag{8}$$

where $H(\omega)$ is the FRF, $X(\omega)$ is the input spectrum, and $Y(\omega)$ is the output spectrum. Hence, $Y(\omega)$ can be represented as a function of the product of $H(\omega)$ and the input $X(\omega)$ in the frequency domain, that is,

$$Y(\omega) = H(\omega)X(\omega) \tag{9}$$

In practice, one often needs to approximate the FRF, $H(\omega)$, for a given system such that the relationship between the system output $Y(\omega)$ and input $X(\omega)$ can be determined analytically. In a data-driven manner, $H(\omega)$ can be learned as a mapping relationship between $Y(\omega)$ and $X(\omega)$. Since the mapping relationship varies as ω changes, $H(\omega)$ should be learned as a function not only of the input spectrum $X(\omega)$ but also as a function of ω. Towards this end, $Y(\omega)$ is represented as a general function of ω and $X(\omega)$ as

$$Y(\omega) = F(\omega, X(\omega)) \tag{10}$$

Equation (10) can be learned as a data-driven model that approximates the FRF. The FRF learning problem naturally becomes a regression problem between $(\omega, X(\omega))$ and $Y(\omega)$. The task is then to determine the function F, which is equivalent to a FRF and can be used to predict the output $Y(\omega)$.

2.3 Challenges. The major difficulty with learning a FRF is that, for a complex system, FRFs may be highly nonlinear and depend on the inputs [28], such as speeds and force amplitudes. Also, FRFs are typically not smooth or continuously differentiable. Approximating a FRF with polynomial terms can lead to overfitting as well. Additionally, it is difficult to interpolate along a FRF because the response of the FRF at each given frequency is relatively independent.

A practical challenge comes from the nature of operating data. To obtain a full FRF, an ideal impulse input excitation is desired since it includes all frequency components uniformly. During impulse testing, the testing system undergoes a free-vibration response. However, during normal operating conditions, the system is often under forced vibration. The input excitations are typically at discretized harmonics, such as the multiples of the spindle speed frequency. Therefore, collecting frequency response data at each frequency is infeasible during normal operation, which makes learning the FRF from real-time operating data challenging. For example, Fig. 2(*a*) shows a FRF for a machine spindle with sampled points marked as red dots. It is shown that if the only sampled responses are the red dots, then many of the harmonics of the FRF would not be captured, making it difficult to model via a data-driven learning model.

The goal of learning a FRF is to accurately predict all responses across the function domain. However, it is typically not possible to collect operating data at each frequency point. Due to the challenge of limited data, the model must be capable to "extrapolate" to unforeseen frequencies, so a learning scheme with a function basis is preferred to introduce some extrapolating capability.

Other realistic application challenges for learning the FRF include that machine conditions can change over time due to system degradation and FRFs can vary significantly from one machine to another. Once a FRF associated with a given system is learned by a model, it is challenging to apply the model to a new system without enough training data. Figure 2(*b*) shows two different FRFs. A model that learns the first FRF, the blue curve in Fig. 2(*b*), will only be effective in practical application if it can adapt to the changes of a new FRF, the red curve in Fig. 2(*b*), without requiring too much new training data. As discussed above, training data under one operating condition captures the frequency response effectively at only certain discrete frequencies. Thus, to learn a continuous FRF that can predict the force spectrum at a broader frequency range than the frequencies covered in the training dataset, few-shot regression can be employed. Compared to random initialization of the model for the normal training process, the few-shot regression training takes the trained model for the existing dataset and performs further training with a few samples from the new machine until the model converges for the new dataset.

The last challenge in learning the FRF is that the data of $X(\omega)$ and $Y(\omega)$ are complex numbers. To handle complex numbers with a machine learning model, one can either separate the complex numbers into real and imaginary parts or convert them into amplitude and phase representations. This creates a multivariate regression problem like a MIMO system. To handle a multivariate

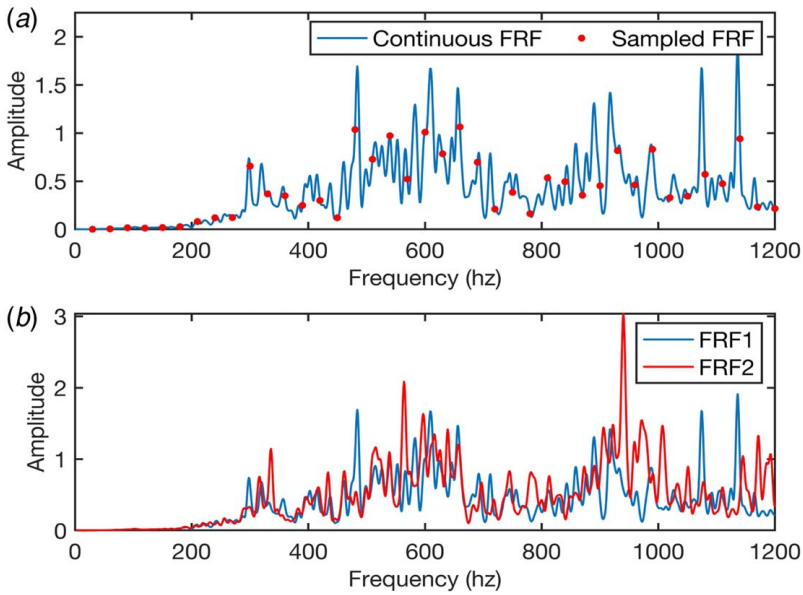

Fig. 2 (*a*) Plot of continuous FRF versus sampled FRF and (*b*) comparison of two different FRFs

regression problem with a learned function basis, we designed a model with a shared function basis and separate weight vectors for each output dimension. Each function basis is a function containing all input dimensions.

2.4 Proposed Frequency Response Function Learning Approach. To tackle the above challenges, the nonlinear function basis-based regression model is adopted and expanded to develop a FRF learning model. In this paper, real and imaginary parts are used separately. Thus, the relationship between the input (X) and output (Y) of the system can be rewritten as shown in (11), where the input is a [3 × 1] vector and the output is a [2 × 1] vector:

$$\begin{bmatrix} Y_R \\ Y_I \end{bmatrix} = F\left(\begin{bmatrix} \omega \\ X_r \\ X_i \end{bmatrix}\right) \tag{11}$$

More generally, (11) can be written as

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = F\left(\begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix}\right) \tag{12}$$

The expansion of $F(\omega, X(\omega))$ reflects the weights and basis functions of the model, as shown here:

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = F\left(\begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix}\right) = \begin{bmatrix} P_1\psi \\ P_2\psi \end{bmatrix} \tag{13}$$

Equation (13) can be further expanded into (14), where all basis functions (f_j) and the respective weights ($p_{i,j}$) are shown:

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,n} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,n} \end{bmatrix} \cdot \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix} \tag{14}$$

As stated in Sec. 2.1, the model learns both the set of basis functions, $\boldsymbol{\psi}$, as well as the set of weights, $\boldsymbol{P}$. Each basis function within the set is a function of ω and the inputs spectrum value at ω. In other words, each output of the basis function generator (see Fig. 1) can be noted as $\varnothing_i = \psi_i(\omega, X(\omega))$. The weights are then a function of the learned set of basis functions and the ground truth output of the system. Therefore, each output from the weights generator (see Fig. 1) can be written as $p_{i,j} = h(\boldsymbol{\psi}, Y(\omega)) = g(\omega, X(\omega), Y(\omega))$. In the formulation of this paper, the inputs to the model are the frequency ω and the real and imaginary parts of a complex number; the outputs of the model are the real part (y_R) and the imaginary part (y_I) of a complex number.

Learning the data-driven model (see Fig. 1) involves minimization of the overall loss function, L, defined as

$$L = \frac{1}{n}\sum_{i=1}^{n}[(Y_{i,R} - y_{i,R})^2 + (Y_{i,I} - y_{i,I})^2] + \|Q\|_1 + \|Q\|_2 \tag{15}$$

where $Y_{i,R}$ is the ground truth real part, $y_{i,R}$ is the predicted real part, $Y_{i,I}$ is the ground truth imaginary part, and $y_{i,I}$ is the predicted imaginary part for the ith point in the frequency domain. Also, $\|\cdot\|_1$ represents the L1 norm, $\|\cdot\|_2$ represents the L2 norm, and Q is the full set of model parameters for the BFG and the WG. The summation in (17) is over the n complex data points. The completion of model training provides a fully trained BFG and WG and a final mean weights matrix. Testing consists of evaluating the response of each set of basis functions and computing the prediction with the final mean weights matrix. The L1 and L2 norms are added to the cost function to regularize the solution space. L1 regularization ensures a sparse model and the L2 norm reduces overfitting.

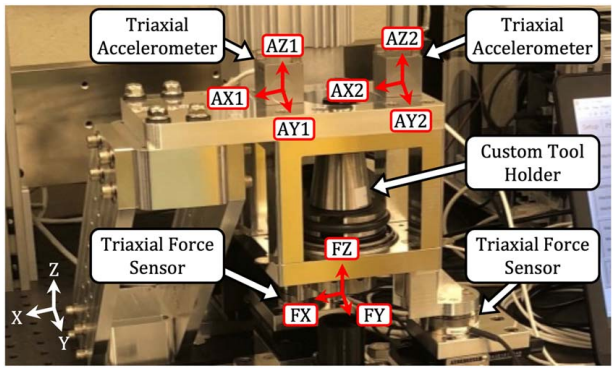

Fig. 3 Spindle testbed comprising a custom tool holder, accelerometers, and force sensors

3 Experimental Study

3.1 Experimental Setup. Figure 3 shows the spindle testbed that was created for the development and testing of new methods for estimating real-time cutting forces from on-machine accelerations. In the testbed, a motor connects to the custom instrumented tool holder with a rubber belt under tension. When the motor operates, it spins the tool holder via the belt. The motor speed can be controlled similarly to a spindle motor in a machining center. The tool holder has a rotational speed up to 3000 rpm. Simultaneously, a magnet on the custom tool holder rotates and interacts with two stationary magnets. Figure 4 shows how the peak-to-peak magnetic forces can be controlled by changing the distance, d, between the rotating and stationary magnets. As the distance is decreased, the magnetic force increases. For any spindle speed, force magnitudes

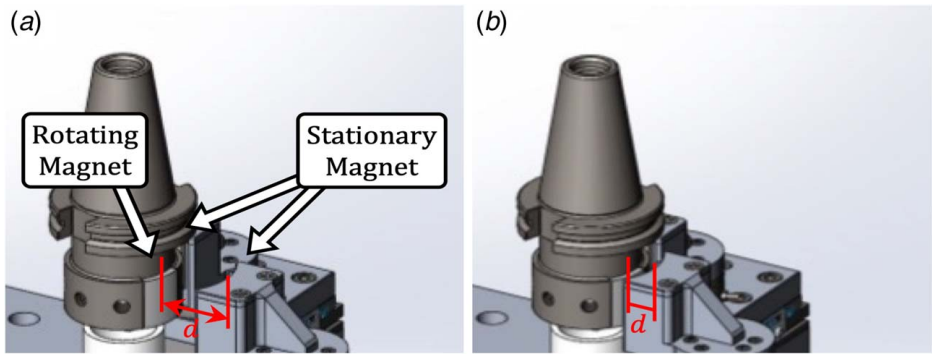

Fig. 4 Illustration of a magnetic assembly located to yield a (*a*) relatively low tool-to-workpiece force or (*b*) relatively high tool-to-workpiece force

range up to 40 N. These forces are measured during rotation along with six accelerations (AX1, AY1, AZ1, AX2, AY2, and AZ2) from two triaxial accelerometers that are mounted near one of the spindle bearings. Because the magnetic forces simulate cutting forces, the spindle testbed mechanically simulates a machine tool spindle during parts production for any combination of spindle speed and force range.

The accelerations represent on-machine sensor readings that will be utilized in the proposed approach to estimate cutting forces, as summarized in Fig. 5. Before machining, accelerations and magnetic forces at various spindle speeds and force ranges are collected with the custom instrumentation. This dataset is then processed via the proposed method to learn a data-driven FRF between the forces and accelerometer data. Finally, during machining, a different tool holder is placed in the spindle, and as the tool rotates and removes material from the workpiece, the accelerometer data are inputted into the learned FRF model, which outputs a real-time estimation of the forces. If implemented on a machine tool during production, the method will yield real-time force estimates that can be used to determine the health of the cutting process for process control purposes and prognostics and health management.

It is worth noting that the force is the actual excitation (input) and the vibration on the spindle housing is the response (output) in traditional system dynamic modeling. *In contrast, the prediction model for force prediction represents the inverse FRF in a strict sense.* However, a machine learning model is flexible with the selection of inputs and outputs. Therefore, the model in this paper is still considered as a data-driven FRF.

3.2 Data Collection. Three datasets were collected to validate the proposed FRF learning method for FRF learning, dynamic cutting force prediction, and adaptation to a new machine. Dataset A is used to demonstrate the performance of FRF learning. Dataset B is used to validate the learned FRF for the application of dynamic force prediction. Dataset C is used to demonstrate that the learned FRF can be adopted to a new machine with limited new data. The details of the datasets are summarized in Table 1.

One second of acceleration and force signals were recorded for each data sample in all datasets. For dataset A, data were collected at 1300 different operating states. Specifically, data were collected for 100 different spindle speeds, from 400 rpm to 3000 rpm at every 26 rpm, and 13 different force levels, from 4 N to 40 N at every 3 N. The data were then divided into 0.5-s-long segments, which double the number of samples to 2600 samples. Then, the temporal data were converted to the frequency domain via Fourier transforms. These single-sided Fourier spectrums have a frequency resolution of 2 Hz, due to the sample length of 0.5 s. The direct current components are not utilized, however, to focus on the alternating current response of the dynamical system. Figure 6 shows two examples of force and vibration signals in both the time and frequency domains.

Table 1 Description of experimental datasets

Dataset	A	B	C
Number of samples	2600	5200	1260
Speed range (rpm)	400–3000	400–3000	150–3000
Speed interval (rpm)	26	13	500
Force range (N)	4–40	4–28	4–40
Force interval (N)	3	2	5
Reduced stiffness?	[1] No	[2] No	[3] Yes

Dataset B was collected with the same simulated machine settings as dataset A, making them suitable for regular model training and testing. The spindle speed range was the same for both datasets, from 400 rpm to 3000 rpm, but the speed interval for dataset B was 13 rpm, which is half the size as 26 rpm for dataset A. Another difference is that the force range was 4–28 N with an interval of 2 N for dataset B. Therefore, dataset B includes more data than dataset A.

Dataset C is different than datasets A and B. These data were collected to simulate a different machining center than the other two datasets with a significantly reduced stiffness, adjusted by clamp springs. Thus, it makes the dataset an effective one for evaluating force predictions on different machines. Dataset C had seven different spindle speeds, from 150 rpm to 3000 rpm with a nominal speed interval of 500 rpm, and nine force levels per spindle speed, from 4 N to 40 N with a nominal force interval of 5 N. Twenty samples were collected for each operating condition. This dataset includes a total of 1260 samples.

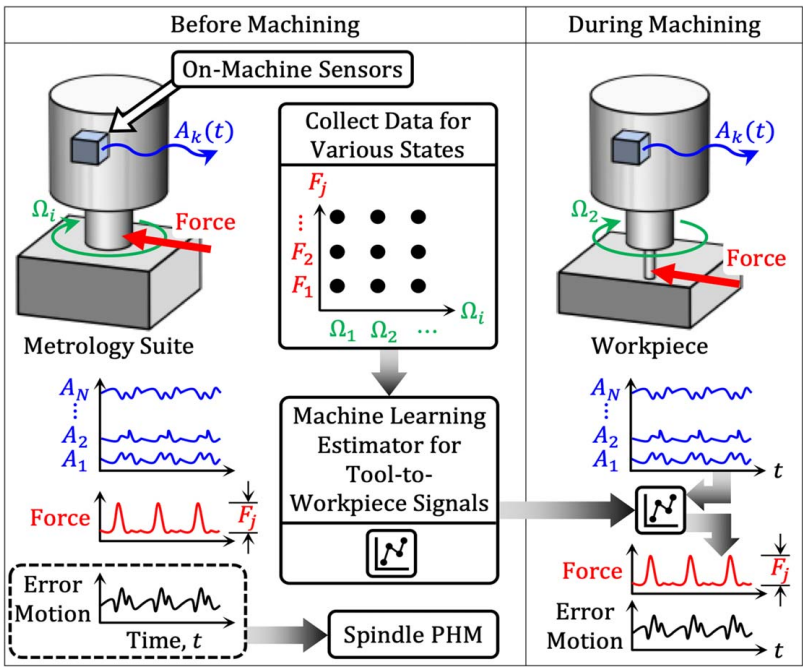

Fig. 5 Summary of the high-level approach for estimating real-time cutting forces

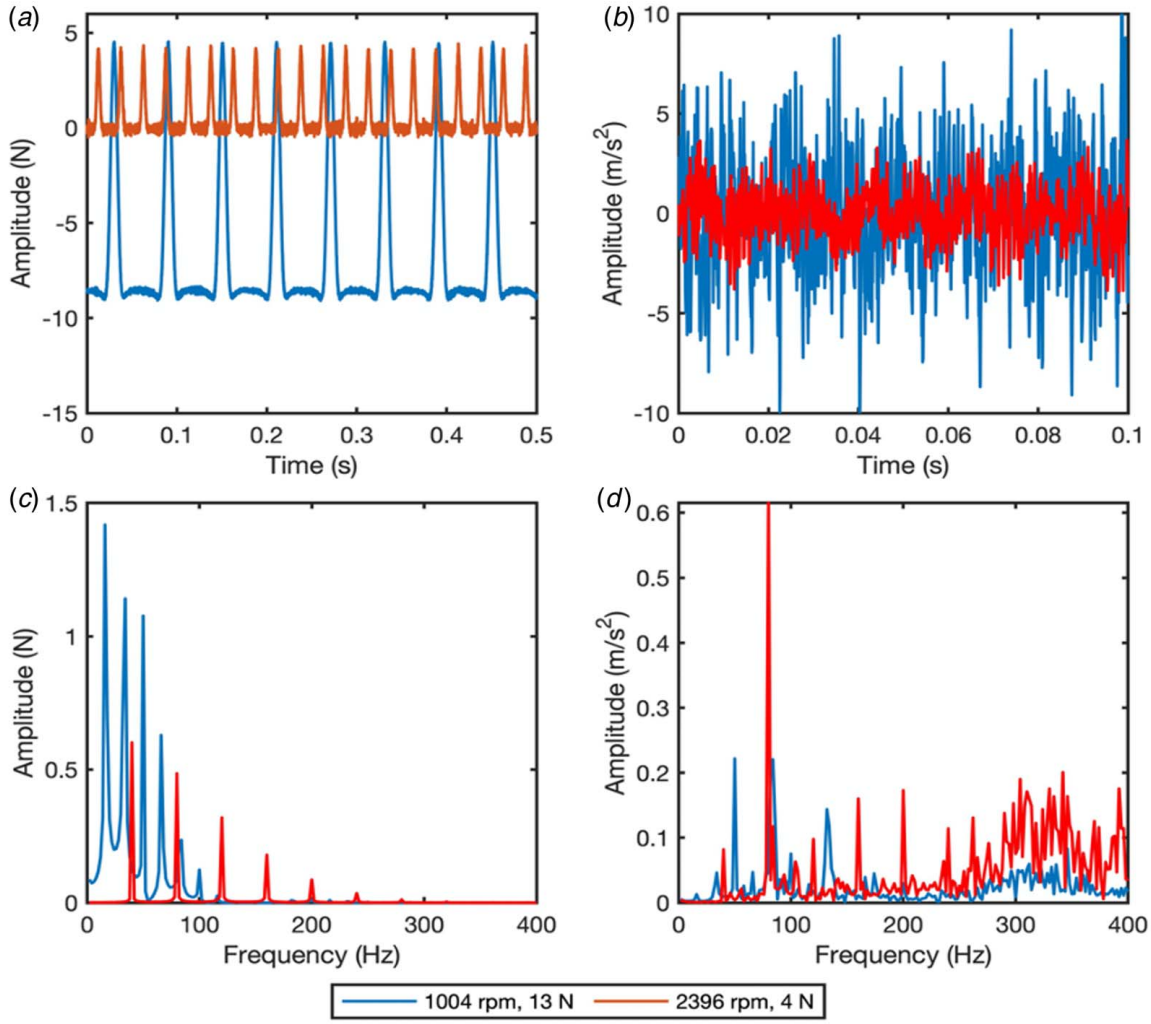

Fig. 6 Example force and vibration signals and their spectrums: (*a*) the force signals, (*b*) the vibration signals, (c) the force spectrum, and (*d*) the vibration spectrums for two spindle states with different spindle speeds and force levels

After the Fourier transform is performed on the collected temporal data, the real and imaginary parts of the complex spectrums are separated. Real and imaginary parts of vibration spectrums and the corresponding frequency create a 3×1 dimensional input data. The ground truth labels are the corresponding real and imaginary parts of the complex force spectrums with a dimension of 2×1.

3.3 Model Architecture. To learn a FRF between the vibration and force signals, the first 200 Hz of the frequency spectrums were used. With a frequency resolution of 2 Hz, each sample contains 100 points. It is worth noting that, even if the training data are provided at every 2 Hz, under a specific spindle speed, the useful training points are mainly at the spindle speed frequency and its harmonics. Sixty-four basis functions are chosen for both case studies reported in Sec. 4. The model structure is as follows: the BFG takes a three-dimensional input and has ten subsequent fully connected layers with 64 nodes per layer. The activation functions are rectified linear unit for all but the last layer, where a sigmoid activation function is chosen. The WG network is an attention model with four attention layers. Before each attention layer, there is an additional embedding layer. The WG has 66 nodes in the input (64 from the BFG concatenated with two ground truth labels), 64 nodes in each embedding layer, 128 nodes in each attention layer, and 128 nodes for the output (64×2 for two-dimensional output). The learning rate was set to 0.0005, the weight decay was set to 0.000001, the L1 regularization term was set to 0.00001, and the L2 regularization term was set to 0.0001. Model structure analysis and model convergence are covered in the results sensitivity analysis in Sec. 4.

For case study 2 reported in Sec. 4, few-shot training is performed for dataset C in addition to the original training of the model based on dataset A. The few-shot learning aims to learn the FRF for a new machine with limited testing samples. The few-shot process takes a similar route as the original training procedure except that the trained model for dataset A is initialized as the model for dataset C. Then, the few-shot training is performed for dataset C with 10, 20, and 50 shots.

The metrics used to evaluate the results are the mean absolute error (MAE), the mean root mean square error (MRMSE), and the mean maximum error (MME), defined as

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{m}\frac{|Y_{ij} - y_{ij}|}{m}\right) \tag{16}$$

$$\mathrm{MRMSE} = \frac{1}{n}\sum_{i=1}^{n}\left(\sqrt{\sum_{j=1}^{m}\frac{(Y_{ij} - y_{ij})^2}{m}}\right) \tag{17}$$

$$\mathrm{MME} = \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{m}\frac{\max(|Y_{ij} - y_{ij}|)}{m}\right) \tag{18}$$

where Y_{ij} is the jth data point in the ith sample of the ground truth, y_{ij} is the jth data point in the ith sample of the prediction, n is the number of samples, and m is the number of data points in each sample.

4 Results and Discussions

To test and validate the proposed methodology, the learned FRF of the machining spindle will be first presented, followed with two case studies on dynamic cutting force prediction using the learned FRF, and then a sensitivity analysis is presented.

4.1 Learned Nonlinear Frequency Response Function. As we emphasized in the introduction, there is no ground truth for the run-time FRF since the FRF will vary depending on speed and force levels. To validate the learned FRF, we averaged FRFs for various repeated measurement samples near a specific operational condition and assumed this computed ratio to be the true FRF. We then compared the learned (inverse) FRFs and the computed FRFs from the experimental measurements. For various levels (low, medium, and high) of speeds and forces, we compare the predicted FRFs with the measured FRFs.

For example, Fig. 7 shows the FRFs at three different rotational speeds. It shows that at different rotational speeds, the FRF varied significantly, perhaps due to gyroscopic and centripetal effects as well as nonlinearities, because at different spindle speeds, the dynamic properties and coupling boundary conditions, such as the contact stiffness between moving parts, can change significantly. Figure 7 shows that the predicted FRFs not only match the measured FRFs with high accuracy but also capture the variations at different operational speeds.

The learned FRFs can also be compared with the measured FRFs for different force levels. Because the cutting force modifies the boundary conditions within the spindle, when the cutting force is different, the spindle is subjected to different boundary conditions that can affect the system response in a nonlinear manner. For example, Fig. 8 shows that the measured FRF changes with different force levels and that these variations are captured accurately by the predicted FRFs.

Lastly, Fig. 9 compares the learned average FRF and the measured average FRF for all tested speeds and forces. Even though the predicted FRF matches the measured average FRF fairly accurately, it is worth noting that this FRF only represents the average dynamic behavior of the system; the average FRF never governs the system dynamics at any moment. Instead, the average FRF represents a traditional static view of the FRF for a linear system. It can be understood that if a static FRF were used to represent the system

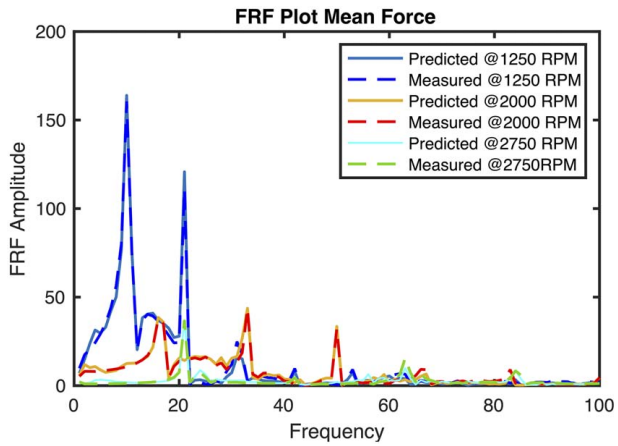

Fig. 7 Learned and measured FRFs under different rotational speeds

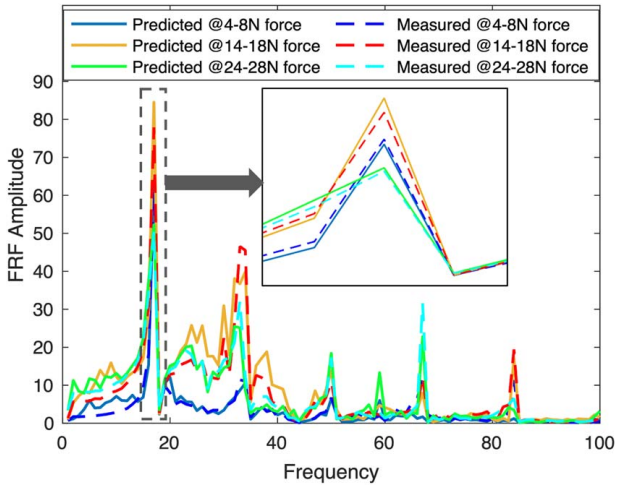

Fig. 8 Learned and measured FRFs under different force levels for a spindle speed of 1000 rpm

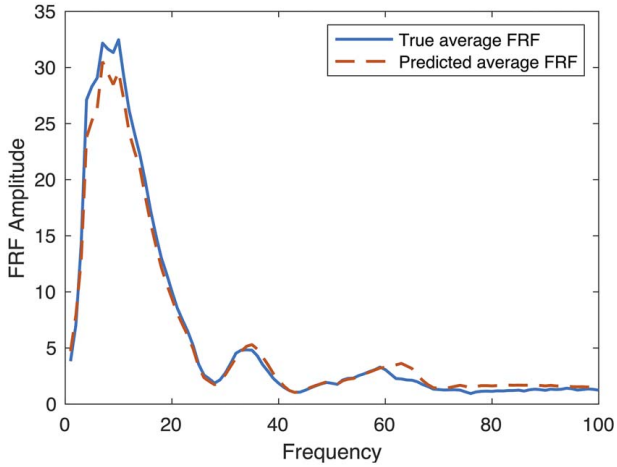

Fig. 9 Comparison of the average measured FRF and the average predicted FRF under all operating conditions

dynamics, it would not be able to reflect the varying relationship between force and vibration under different operating conditions.

Next, we will use two case studies to evaluate the proposed FRF learning method for the application of run-time dynamic cutting force prediction.

4.2 Force Prediction Case Study 1: Model Training and Testing With Datasets From the Same Machine. This case study aims to evaluate the performance of the FRF learning method for run-time force prediction. We purposely trained the model with a small dataset and test the performance on a relatively larger dataset and make predictions over unforeseen operating conditions under the same machine setting. Particularly, a model was trained on dataset A and the learned FRF was then tested with dataset B. Since dataset B is collected at a smaller spindle speed interval compared with dataset A, this process allows for the determination of whether the model can handle unforeseen predictions at untrained frequencies. Hundred and fifty epochs were run for the training stage in this case study. Figure 10 shows example comparisons of force predictions against the ground truth for low, medium, and high spindle speeds (1000 rpm, 2000 rpm, and 3000 rpm, respectively) as well as low and high force levels (6 N and 26 N, respectively). The relatively small differences between the predicted and measured forces reveal that the model reconstructs the

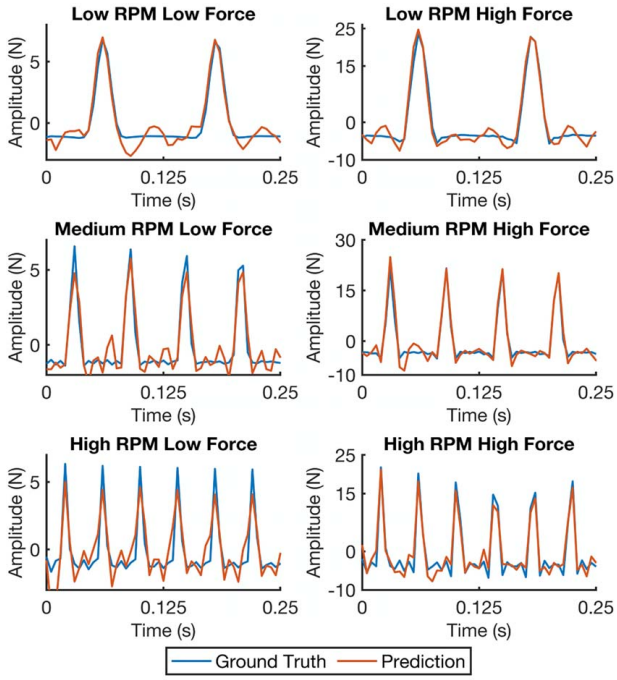

Fig. 10 Reconstructed force signal plots from dataset B

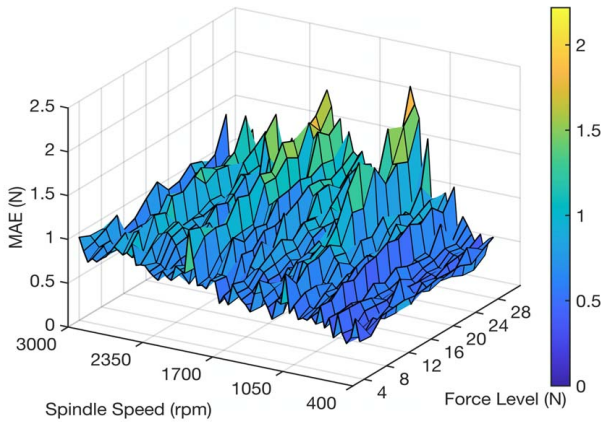

Fig. 11 Surface plot of MAE of testing results

real-time forces with minor errors. Nonetheless, all six plots in the figure show some relative inaccuracy in the valleys between the force peaks. This could possibly be caused by the noise in the vibration spectrums.

Figure 11 shows a surface plot for the MAE metric across various spindle speeds and force levels. The notable trend is that the error metric increases as the force level of the signal increases. However, the increase is considered moderate, because if the metric is evaluated relative to the force level, the relative MAE metric would be nominally insensitive to the force level. Also, with regard to spindle speed, there does not seem to be a significant correlation between spindle speed and the MAE metric.

Table 2 summarizes the overall prediction errors for case study 1 with population values reported in the form of "mean ± standard deviation." The MAE metric shows that the average force prediction error is less than 1 N, which is only about 4% of the median force level of 24 N. Since dataset B has twice the number of spindle speeds as dataset A, half of the spindle speeds in dataset B are not present in dataset A, which means that the results show that

Table 2 Force prediction error of case study 1

Metric	MAE	MRMSE	MME
Value (N)	0.958 ± 0.464	0.258 ± 0.222	3.513 ± 1.745

the model can extrapolate the frequency response to untrained spindle speeds.

4.3 Force Prediction Case Study 2: Model Training and Prediction With Datasets From Different Machines. The second case study was also performed with training on dataset A. However, the goal is to use the model learned with dataset A to predict the force in dataset C with a very small amount of new data. As stated previously, the prediction data, dataset C, were collected with different machine settings, which simulate a different machine. The initial training process is considered as meta-learning and a few-shot learning was further carried out on the prediction dataset. Predictions were first evaluated without few-shot learning. Then, 10-, 20-, and 50-shot training was performed on dataset C and the predictions were evaluated again. Random samples from dataset C were selected for the few-shot training. Compared with the full dataset C, the few-shot learning only requires a minor fraction of training data.

Figure 12 shows the reconstructed force, with few-shot learning ("few-shot prediction") and without few-shot learning ("prediction"), and the measured forces ("ground truth") for dataset C. Similar to the plots in Fig. 10, Fig. 12 shows results for low, medium, and high spindle speeds (1000 rpm, 2000 rpm, and 3000 rpm, respectively) with low and high force levels (4 N and 30 N, respectively).

The predictions without few-shot learning show significant deviations from the ground truth, especially for low and medium spindle speeds. Therefore, the initial model from meta-learning, based on the full dataset A, cannot adequately reconstruct the force signal

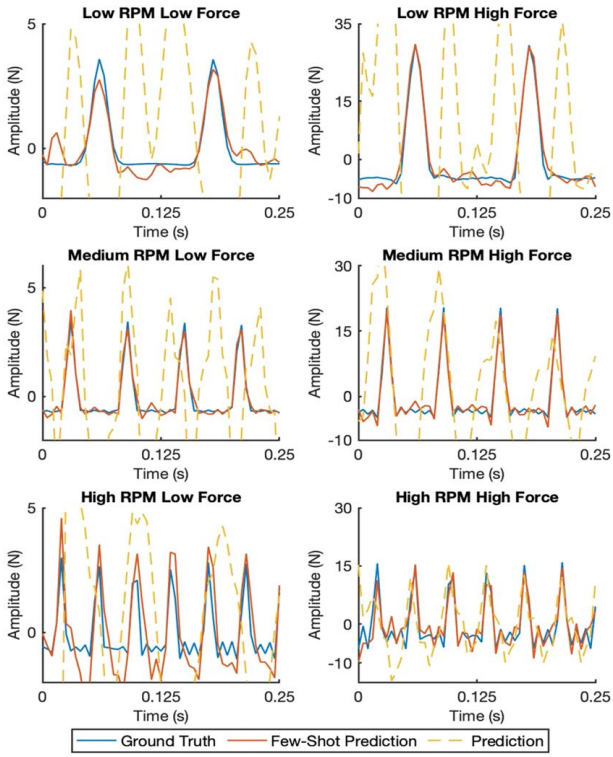

Fig. 12 Model prediction and few-shot prediction plots

Table 3 Model prediction and 50-shot prediction results

Metric	Original model	50-shot model
MAE (N)	9.296 ± 5.829	[4] 1.212 ± 1.147
MRMSE (N)	4.954 ± 4.076	[5] 0.504 ± 0.794
MME (N)	31.196 ± 19.279	[6] 4.592 ± 3.928

from vibration data from dataset C. This is understandable because the machine settings, and hence the FRFs, are different for datasets A and C. In contrast, Fig. 12 shows that the model after few-shot learning updated and learned the FRF for the different machine (for dataset C) to reconstruct the ground truth signals with a much higher accuracy. The results indicate significant improvements to the FRF learning with the few-shot learning scheme.

Quantitative results for predictions from the original model and the 50-shot model are shown in Table 3, with all values reported in the form of "mean ± standard deviation." Table 3 shows the improvement in prediction across all three metrics of 50-shot learning compared to those without few-shot learning. In particular, the MAE error decreased by at least 85%, and the standard deviations improved similarly, due to the few-shot learning. For 10- and 20-shot models, MAE errors of 2.71 N and 2.24 N were obtained, respectively, compared to an error of 1.21 N for the 50-shot model. Again, this demonstrates that few-shot learning can utilize a small amount of data to update the learned FRF to perform accurately in the new function domain.

4.4 Sensitivity Analysis. To determine the optimal structure of the model and evaluate the computation cost, an extended analysis was performed to determine the optimal number of basis functions (the model width), the optimal numbers of layers in the BFG and WG modules (the model depth), and the convergence of the model.

Four different models, various model widths (32, 64, 128, or 256) were tested to determine the optimal number of basis functions. To ensure time efficiency in the comparison process, for the model width and depth comparisons, each model was trained for 100 epochs. Table 4 details the metrics for these models based on case study 1. A model with 64 basis functions performed the best out of the four models evaluated.

For the purpose of simplicity, the reported results for the following comparisons will be limited to the MAE, since the results from the MRMSE and MME metrics generally lead to the same conclusion as that from the MAE. Also, the standard deviation is omitted in the following tables. A comparison of model depth was performed with the number of basis functions fixed at 64. Based on the model structure used in Ref. [27], the model structure was varied extensively to determine the optimal structure. A partial list of results is given in Table 5 to show the general effect of model depth on the MAE metric. The model depth is represented as the numbers of layers in BFG and WG. The optimal depth for the model was determined to be ten layers in the BFG module and four attention layers in the WG module.

Finally, model convergence was evaluated for both the original model training and few-shot training. Both models were trained until the onset of overfitting and the deterioration of the error metrics. For the original model training with dataset A, convergence

Table 4 Model width comparison

Model width	32	64	128	256
MAE (N)	1.341	[7] 0.994	1.306	2.839
MRMSE (N)	0.303	[8] 0.268	0.388	2.193
MME (N)	5.304	[9] 3.681	5.075	10.858

Table 5 Model depth comparison

Model depth	(5, 4)	(6, 4)	(6, 5)	(9, 4)	(10, 4)	(12, 4)
MAE (N)	1.25	1.22	1.28	1.07	[10] 0.994	1.71

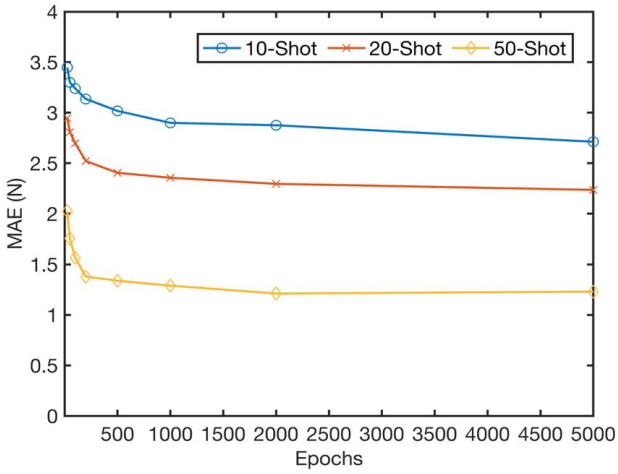

Fig. 13 Few-shot convergence results

occurred between 150 epochs and 200 epochs. In contrast, for the few-shot model in case study 2, convergence occurred around 2000 epochs and 5000 epochs, as observed in Fig. 13. It is shown in the above results that the few-shot scheme is highly effective in learning a new FRF even with a very low number of samples. However, it does require more epochs for the model to converge to the new dataset. Figure 13 shows how the MAE metric converges as the epochs increase from 25 to 5000 for 10, 20, and 50 shots. The MAE converged around 5000 epochs for 10-shot and 20-shot models. In contrast, convergence occurred around 2000 epochs for the 50-shot model. This can be explained as with a small number of samples, the learning rate is slow and the model takes many epochs to converge. However, since the few-shot learning sample size is small, it took less than 10 min for 5000 epochs with training on a Macbook Pro computer with 8 GB of memory and an Intel® Core™ i5 processor.

In summary, even though different model structures lead to different performances, the variance among them is relatively small. Also, the best model for our dataset has similar numbers of layers and basis functions as the model reported in Ref. [27], while our model features multivariate regression and is applied to a completely different scenario than the image recognition problem in Ref. [27]. Admittedly, there is not a logical way to determine the best model structure. The model structure in this paper provides a rule-of-thumb configuration and can be used as a starting model for other learning tasks.

5 Conclusion

This paper presented a data-driven method to learn a FRF under dynamic steady state with a sparse function basis. More specifically, a multivariate few-shot regression method was proposed in this paper to learn an implicit FRF that represents the relationship between inputs and outputs for spindles under operating conditions. The proposed FRF model can effectively learn a general FRF model that includes speed-dependent effects and the nonlinearities of the system at different operating conditions.

The learning methods were applied to a spindle testbed that uses accelerations measured from on-machine sensors to estimate real-time machining forces. It was shown that the proposed

method can predict dynamic cutting forces with high accuracy. In the case of model migration to a new machine, a few-shot learning strategy can be applied. It was shown that with a small number of few-shot learning samples from a new machine, the model can learn a new FRF. It is worth noting that although the model in this paper is developed for FRF learning, the learning framework on regression can be applied for vector regression or to a general multiple-input and multiple-output system.

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.

Funding Data

- Jacob Fabro and Yongzhi Qu acknowledge that this work was supported in part by the Department of Commerce of the United States under contract no. 70NANB20H175 and supported in part by the University of Minnesota in Grant-in-Aid of Research, Artistry and Scholarship.

Conflict of Interest

There are no conflicts of interest.

Data Availability Statement

The datasets generated and supporting the findings of this article are obtainable from the corresponding author upon reasonable request. The data and information that support the findings of this article are freely available.[3] The authors attest that all data for this study are included in the paper. Data under embargo.

References

[1] Schmid, P., 2010, "Dynamic Mode Decomposition of Numerical and Experimental Data," J. Fluid Mech., **656**(8), pp. 5–28.
[2] Williams, M. O., Kevrekidis, I. G., and Rowley, C. W., 2015, "A Data-Driven Approximation of the Koopman Operator: Extending Dynamic Mode Decomposition," J. Nonlinear Sci., **25**(6), pp. 1307–1346.
[3] Mezic, I., 2013, "Analysis of Fluid Flows Via Spectral Properties of the Koopman Operator," Annu. Rev. Fluid Mech., **45**(1), pp. 357–378.
[4] Raissi, M., Perdikaris, P., and Karniadakis, G. E., 2019, "Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations," J. Comput. Phys., **378**(2), pp. 686–707.
[5] Shi, J., Cao, H., Maroju, N. K., and Jin, X., 2019, "Dynamic Modeling of Aerostatic Spindle With Shaft Tilt Deformation," ASME J. Manuf. Sci. Eng., **142**(2), p. 021006.
[6] Tlusky, J., Livingston, R., and Teng, Y. B., 1986, "Nonlinearities in Spindle Bearings and Their Effects," CIRP Ann., **35**(1), pp. 269–273.
[7] Koyuncu, A., Cigeroglu, E., and Özgüven, H. N., 2017, "Localization and Identification of Structural Nonlinearities Using Cascaded Optimization and Neural Networks," Mech. Syst. Signal Process., **95**(10), pp. 219–238.
[8] Teti, R., Jemielniak, K., O'Donnell, G., and Dornfeld, D., 2010, "Advanced Monitoring of Machining Operations," CIRP Ann., **59**(2), pp. 717–739.
[9] Vashisht, R. K., and Peng, Q., 2020, "Online Chatter Detection for Milling Operations Using LSTM Neural Networks Assisted by Motor Current Signals of Ball Screw Drives," ASME J. Manuf. Sci. Eng., **143**(1), p. 011008.
[10] Varghese, A., Kulkarni, V., and Joshi, S. S., 2020, "Tool Life Stage Prediction in Micro-Milling From Force Signal Analysis Using Machine Learning Methods," ASME J. Manuf. Sci. Eng., **143**(5), p. 054501.
[11] Ambhore, N., Kamble, D., Chinchanikar, S., and Wayal, V., 2015, "Tool Condition Monitoring System: A Review," Mater. Today: Proc., **2**(4–5), pp. 3419–3428.
[12] Abdullah, L., Jamaludin, Z., Tsung Heng, C., and Rafan, N., 2012, "Systematic Method for Cutting Forces Characterization for XY Milling Table Ball Screw Drive System," Int. J. Mech. Mechatron. Eng., **12**(6), pp. 28–33.
[13] Brecher, C., Eckel, H., Motschke, T., Fey, M., and Epple, A., 2019, "Estimation of the Virtual Workpiece Quality by the Use of a Spindle-Integrated Process Force Measurement," CIRP Ann., **68**(1), pp. 381–384.
[14] Altintas, Y., and Aslan, D., 2017, "Integration of Virtual and On-Line Machining Process Control and Monitoring," CIRP Ann., **66**(1), pp. 349–352.
[15] Aslan, D., and Altintas, Y., 2018, "Prediction of Cutting Forces in Five-Axis Milling Using Feed Drive Current Measurements," IEEE/ASME Trans. Mechatron., **23**(2), pp. 833–844.
[16] Cao, H., Zhang, X., and Chen, X., 2017, "The Concept and Progress of Intelligent Spindles: A Review," Int. J. Mach. Tools Manuf., **112**(1), pp. 21–52.
[17] Wang, C., Zhang, X., Qiao, B., Cao, H., and Chen, X., 2019, "Dynamic Force Identification in Peripheral Milling Based on CGLS Using Filtered Acceleration Signals and Averaged Transfer Functions," ASME J. Manuf. Sci. Eng., **141**(6), pp. 1–8.
[18] Postel, M., Aslan, D., Wegener, K., and Altintas, Y., 2019, "Monitoring of Vibrations and Cutting Forces With Spindle Mounted Vibration Sensors," CIRP Ann., **68**(1), pp. 413–416.
[19] Kushnir, E., 2004, "Determination of Machine Tool Frequency Response Function During Cutting," Proceedings of the ASME 2004 International Mechanical Engineering Congress and Exposition. Pressure Vessels and Piping, Anaheim, CA, Nov. 13–19, pp. 63–69.
[20] Deng, Z., Zhang, X., and Zhao, Y., 2020, "Transfer Learning Based Method for Frequency Response Model Updating With Insufficient Data," Sensors, **20**(19), p. 5615.
[21] Thenozhi, S., and Tang, Y., 2018, "Learning-Based Frequency Response Function Estimation for Nonlinear Systems," Int. J. Syst. Sci., **49**(11), pp. 2287–2297.
[22] Vidaković-Koch, T., Miličić, L., Živković, L. A., Chan, H. S., Krewer, U., and Petkovska, M., 2021, "Nonlinear Frequency Response Analysis: A Recent Review and Perspectives," Curr. Opin. Electrochem., **30**(12), p. 100851.
[23] Lang, Z. Q., and Billings, S. A., 2005, "Energy Transfer Properties of Non-Linear Systems in the Frequency Domain," Int. J. Control, **78**(5), pp. 345–362.
[24] Bayma, R. S., Zhu, Y., and Lang, Z. Q., 2018, "The Analysis of Nonlinear Systems in the Frequency Domain Using Nonlinear Output Frequency Response Functions," Automatica, **94**(3), pp. 452–457.
[25] Qu, Y., and Vogl, G. W., 2021, "Estimating Dynamic Cutting Forces of Machine Tools From Measured Vibrations Using Sparse Regression With Nonlinear Function Basis," Annu. Conf. PHM Soc., **13**(1), pp. 1–10.
[26] Loo, Y., Lim, S. K., Roig, G., and Cheung, N. M., 2019, "Few-Shot Regression Via Learned Basis Functions," International Conference on Learning Representations, New Orleans, LA, May 6–9.
[27] Loo, Y., Guo, Y., and Cheung, N., 2019, "Few-Shot Regression Via Learning Sparsifying Basis Functions," Open Review Manuscript for International Conference on Learning Representations, Online.
[28] Irino, N., Imabeppua, Y., Higuchia, Y., Shinbaa, Y., Kawaia, K., Suzukib, N., Kanekoc, J., Kakinuma, Y., and Mori, M., 2021, "Vibration Analysis and Cutting Simulation of Structural Nonlinearity for Machine Tool," CIRP Ann. Manuf. Technol., **70**(1), pp. 317–320.

[3]See Note 2.