

# Learning Data-Driven Model of Damped Coupled Oscillators from System Impulse Response

Jacob Fabro<sup>1</sup>, Gregory W. Vogl<sup>2</sup>, Yongzhi Qu<sup>3</sup>, and Reese Eischens<sup>4</sup>

<sup>1,3,4</sup>*University of Minnesota Duluth, Duluth, MN, 55812, USA*  
*fabro001@d.umn.edu, yongzhi@d.umn.edu, eisch112@d.umn.edu*

<sup>2</sup>*National Institute of Standards and Technology, Gaithersburg, MD, 20899, USA*  
*gregory.vogl@nist.gov*

## ABSTRACT

Data-driven modeling of dynamical systems has drawn much research attention recently, with the goal of approximating the underlying governing rules of a dynamical system with data-driven differential equations. Many real-world dynamical systems can be modeled as coupled oscillators, such as chemical reactors, ecological systems, integrated circuits, and mechanical systems. In those systems, the response can be described by coupled differential equations in which multi-dimensional state variables describe the time-evolution. However, in practice, the full set of state variables is often difficult or expensive to measure. This paper shows an attempt to develop a data-driven model for damped coupled oscillators from a mixed-mode response signal using neural differential equations. The univariate time-series data of the impulse response is first resampled into a multi-variate time-delayed embedding. A singular value decomposition is then applied to find the dominant orthogonal basis (oscillator modes). The decoupled modes are then modeled with parameterized neural differential equations. The unknown parameters can be learned from a segment of historical data. The proposed methodology is validated using impact testing data of an end mill in a machine tool spindle. The results demonstrate that the proposed method can effectively model damped coupled oscillators.

## 1. INTRODUCTION

Accurate modeling of dynamical systems is an important task towards health monitoring, prediction, and control. Traditionally, dynamical models are derived from known physical laws and the inherent properties of the systems. Such a dynamical model is typically represented by one or more differential equations, which can be used to model and

simulate the system responses. The term ‘dynamical’ generally refers to the autonomous evolving process over time, governed by inherent system properties. In contrast, a ‘dynamic’ system can refer to any time-evolving system that either evolves autonomously or is driven by external forces. To model a dynamic system with an external force, we often first need to understand and model the dynamical behaviors of the system. Mathematically, many of the prediction problems of dynamical systems can be formulated as initial value problems. Furthermore, the goal of dynamical system modeling includes two parts: solving differential equations and identifying differential equations if unknown. For a simple human-designed system, an accurate physics-based dynamical model via differential equations may be constructed with high confidence. In that case, efficient solving of the differential equations is the main remaining task. However, for complex systems or in-situ systems, where the system parameters cannot be measured accurately, one would first need to identify the underlying differential equations. To achieve that, the missing information on the system properties can be determined with historical observation data. In practice, data-driven models or hybrid models can utilize the observed system data to help construct a dynamic model (Kutz *et al.*, 2016).

Physics-informed machine learning is one of the most popular hybrid methods for the data-driven discovery of dynamic systems. Physics-informed neural networks (PINNs) have been proposed to learn and solve the underlying differential equations (Raissi *et al.*, 2019). When the underlying physics is fully known, a neural network can be used as a numerical solver for either ordinary differential equations (ODEs) or partial differential equations (PDEs), as first reported by Lagaris *et al.* (Lagaris *et al.*, 1998). In the case that the underlying differential equations are not fully known, a PINN can be adopted to learn the unknown coefficients in the differential equations with the help of historical data. The diagram given in Figure 1 summarizes how a PINN works with an ODE. The PINN assumes that the

Fabro 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.

solution to an ODE or a PDE can be approximated by a neural network. The solution is then constrained by the underlying physics-based differential equations. By minimizing the loss of the data and physics constraints, a data-driven solution and the unknown coefficients of the underlying ODE can be learned.

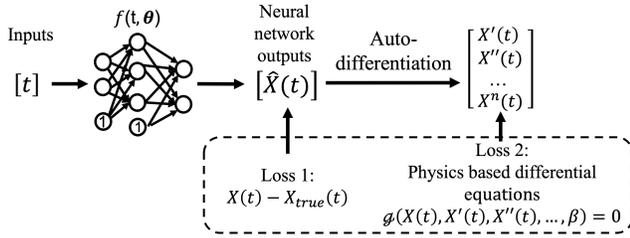


Figure 1. Schematic of a PINN to learn a physics-informed ordinary differential equation.

However, PINNs can only be applied when the forms of the underlying differential equations are fully defined, while the coefficients can be unknown. For complex and in-situ systems, one may not be able to fully determine an accurate form of the equations. In those cases, PINNs fail to learn and model the unknown systems. Intuitively, fully data-driven models can be built without any physics to model in-situ dynamic systems. In previous work, Fabro *et al.* built a regression model with a nonlinear function basis to represent a fully data-driven frequency response function in the frequency domain for machine spindles (Fabro *et al.*, 2022). However, the model did not mimic physics in a *physics-aware* manner.

Recently, neural ordinary differential equations (NODEs) have been proposed to learn the underlying differential equations with no or little physics (Chen *et al.*, 2018). In contrast to a PINN, a NODE constructs a parameterized differential equation and integrates an ODE solver into the model. Details of the learning process will be introduced in Section 2.1.2. NODEs provide a *physics-aware* approach to learning the underlying differential equations only from data. The term ‘physics-aware’ refers to the preservation of certain physical rules, such as a differential relationship. This research aims to investigate the feasibility of learning coupled differential equations from data.

Coupled oscillators are commonly found in practical engineering systems, such as bridges, buildings, geared systems, and machining spindles. Also, many in-situ systems can be modeled as coupled oscillators since there exists complex coupling among the sub-components. One of the characteristics of coupled oscillators is mixed modes in the measured system response. For example, a simple coupled oscillator with two springs and two masses will have two modes, one for each degree of freedom (DOF). Furthermore, each mode will have its own natural frequency. In practice, if the response for each DOF can be measured, the modes can

be separated as an eigen-decomposition problem, even if the modes are coupled together in each measurement. If the response cannot be measured separately, then modal separation is needed. For coupled oscillators without damping, modal separation can be achieved with Fourier transforms to separate different frequency components. However, when damping is present, modal separation becomes more complicated as both the frequencies and associated time-decay terms must be separated. Therefore, the data-driven learning of damped coupled oscillators requires three steps: 1. Modal separation from data; 2. ODE learning for each mode; 3. Model synthesis. For linearly coupled systems, the model synthesis is just a simple sum of all modes.

Towards this end, this paper summarizes an attempt to investigate the feasibility of learning a data-driven model for damped coupled oscillators and use it to describe and predict the system dynamics. The rest of the paper is organized as follows: Section 2 presents the related background and the proposed learning method; Section 3 briefly introduces the experimental study to collect data and test the proposed method; Section 4 discusses the results and Section 5 concludes the paper.

## 2. METHODOLOGY

### 2.1. Related Background

In this section, we will introduce the existing methods for modal separation and ODE learning.

#### 2.1.1 Mixed-mode decomposition

Modal separation of univariate mixed-mode signals has been extensively studied. Fourier transforms can effectively decompose a mixed-mode signal into various frequency components. In that case, each frequency component is represented with sinusoidal basis functions, so Fourier transforms are best suited for the analysis of steady-state responses. While Fourier transforms are effective and efficient for periodic signals, to approximate a non-periodic signal, an infinite number of Fourier basis functions are required, which is numerically inefficient. Independent component analysis (ICA) is one of the statistics-based methods for mixed-mode separation, which is widely used in the cocktail party problem or similar situations (De Lauro *et al.*, 2005; Hyvärinen, 2013). ICA assumes the underlying processes are independent of each other and can handle non-Gaussian processes. However, ICA does not guarantee that the separated components are orthogonal, as with Fourier basis functions. Empirical mode decomposition (EMD) is another popular method for modal decomposition (Huang *et al.*, 1998). Similar to ICA, EMD does not generate orthogonal modes, either.

Recently, Dylewsky *et al.* proposed a learning method named principal component trajectories to model spectrally continuous dynamics (Dylewsky *et al.*, 2022). It was proved that by constructing a time-delay embedding for the time series and applying singular value decomposition (SVD) to the embedding data matrix, the principal component trajectories, which is a set of orthogonal basis functions, can be separated. While Dylewsky *et al.* modeled the underlying system as linear dynamics with dynamic mode decomposition (DMD), this work proposes to learn a neural ODE model for each mode and then assemble the models.

### 2.1.2 Neural Ordinary Differential Equations

As discussed above, NODEs provide a data-driven approach to learning the underlying differential equations. The learning process is summarized as follows. First, assume we have a differential equation in the form of Eq. 1.

$$\dot{X} = f(X) \quad (1)$$

It was first proposed by Chen *et al.* that the function  $f(X)$  can be defined as a neural network with  $X$  as the inputs and  $\dot{X}$  as the outputs of the neural network (Chen *et al.*, 2018). Then, the solution of this neural differential equation can be solved with an integration-based ODE solver, such as the Runge-Kutta fourth-order method. More advanced ODE solvers, such as Dopri5 (Dormand and Prince, 1980), can also be utilized in this process with varying step sizes as an attempt to obtain a more accurate solution, especially for irregular time intervals. Next, assume that the solution of the underlying differential equation can be written as a function of  $X$ , more precisely as  $f(X(t))$ . With an ODE solver using the neural network as the differential function, and  $X(t_0)$  as the initial condition,  $X(t_0 + \Delta t)$  can be solved as:

$$\begin{aligned} X(t_0 + \Delta t) &= X(t_0) + \int_{t_0}^{t_0 + \Delta t} X'(\tau) d\tau \\ &= X(t_0) + \int_{t_0}^{t_0 + \Delta t} f(X(\tau)) d\tau \end{aligned} \quad (2)$$

Finally, the overall process of the neural ODE method can be written as:

$$\dot{X}(t_0) = nn(X(t_0)) \quad (3)$$

$$X(t_1) = ODESolver(\dot{X}(t_0), X(t_0), t_0, t_1) \quad (4)$$

where  $nn(\cdot)$  represents a neural network.

The flowchart in Figure 2 shows the process of learning a neural differential equation to represent the system dynamics. If an example dataset in the form of a time series for  $X(t)$  can be collected, data pairs of  $X(t)$  and  $X(t + \Delta t)$  can be constructed as training data to train the data-driven model

such that Eq. (2) can be satisfied. As a result, the optimized function  $f(X(t))$  will represent a data-driven form of the underlying differential equation.

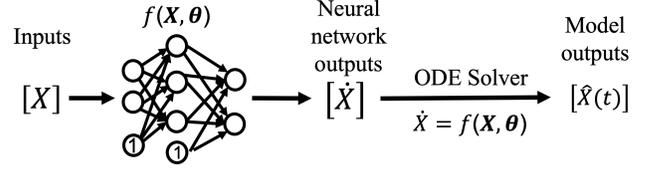


Figure 2. Schematic of NODE.

## 2.2. Proposed Approaches

Based on the information presented in the introduction, a data-driven model for damped coupled oscillators is summarized in Figure 3. First, one can collect the impulse response from an in-situ dynamic system. Typically, a univariate mixed-mode response signal can be measured, denoted as  $X = [x_1, x_2, x_3, \dots, x_n]$ , where  $n$  is the number of samples.

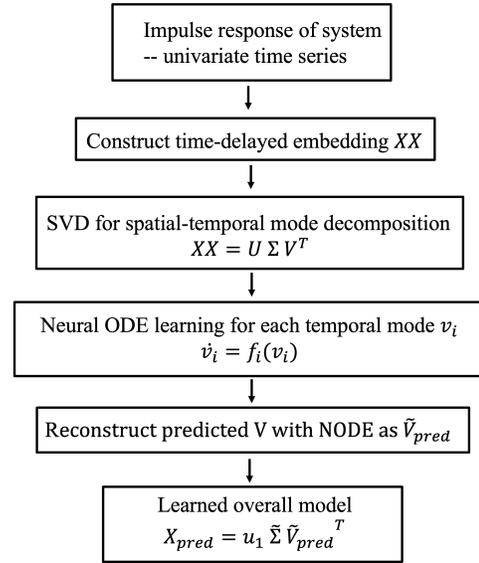


Figure 3. Overall learning method for impulse response.

Next, a time-delayed embedding can be constructed as:

$$XX = \begin{bmatrix} x_1 & x_2 & \dots & x_{n-k+1} \\ x_2 & x_3 & \dots & x_{n-k+2} \\ \vdots & \vdots & \ddots & \vdots \\ x_k & x_{k+1} & \dots & x_n \end{bmatrix} \quad (5)$$

where  $k$  is the number of embeddings. Then, a SVD can be performed on the embedding matrix as  $XX = U\Sigma V^T$ . If we consider the columns as the spatial direction and the rows as the temporal direction, in an analogy to a spatial-temporal data matrix, the time-evolving trends will be captured by the columns of  $V$  after SVD. In the ideal scenario, each column of  $V$  will represent one temporal mode that can be modeled

as an ODE function. For a spectrally-continuous system, we truncate  $V$  to keep only the dominant modes. Each preserved mode is then learned separately as a NODE model. The NODE model can represent the dynamic response at any time. To predict the response, one can reconstruct the dynamic response with the simulated multiple-mode dynamics of  $V_{pred}$  and the corresponding  $U$  and truncated  $\tilde{X}$ . The corresponding  $U$  and  $\tilde{X}$  are constant assuming that the dynamical system is a linear time-invariant (LTI) system. This will allow us to use a small segment of the dynamical response to learn the system and extrapolate the full dynamic response.

### 3. EXPERIMENTAL SETUP AND CASE STUDY

The discussions in Section 3 are organized as follows: Section 3.1 introduces the experimental setup and data collection process, and Section 3.2 presents the learning process with the collected response data from an end mill.

#### 3.1. Experimental Setup and Data Collection

Figure 4 shows the experimental setup used to collect tool impact data on a machine tool.

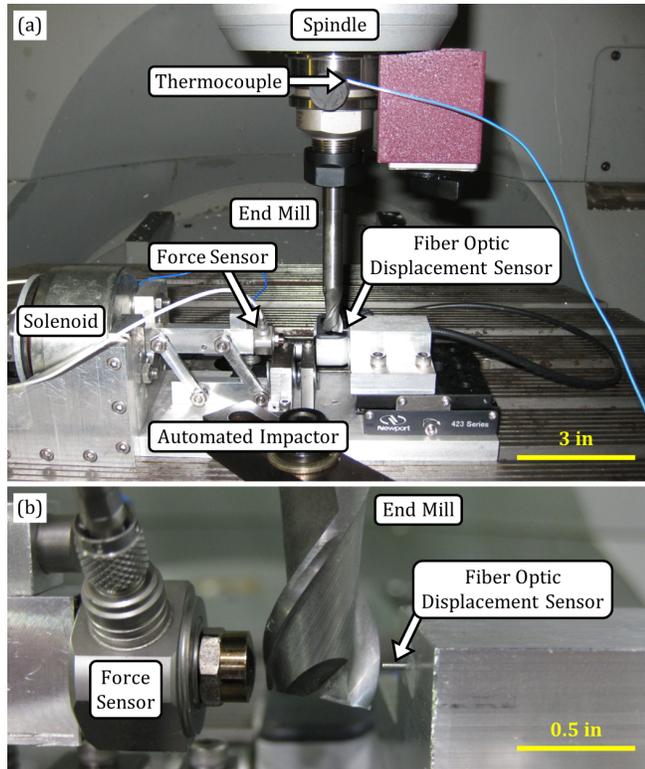


Figure 4: (a) Experimental setup within a machine tool and (b) close-up view of end mill and displacement sensor.

As seen in Figure 4a, an end mill with a diameter of 12.7 mm (0.5 in) was placed within a tool holder in a machine tool spindle, and an automated impactor was attached to the

machine tool worktable. During impacting, a solenoid actuates the hammer of the impactor to cause an impact against the end mill, and a force sensor attached to the hammer measures the impact force while a fiber-optic displacement sensor measures the tool-to-workpiece displacement (Figure 4b). Furthermore, the fiber optic displacement sensor is mounted on a linear positioning stage with motion measured by a digital micrometer for sensor calibration purposes to convert voltage to displacement. For each of 50 impacts, force and displacement data were collected for two seconds at 51.2 kHz, with at least one second occurring after the impact. Figure 5 shows an example of the univariate (displacement) impulse response data. Because the displacement due to impacting is the tool-to-workpiece response, the displacement includes multiple modes that account for the entire structural loop within the machine tool. Hence, the system in Figure 4 represents a coupled mechanical system, and each impulse response, such as the example in Figure 5, is for mixed modes.

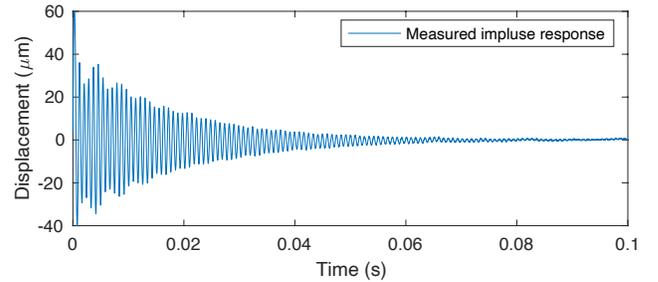


Figure 5: Mixed-mode impulse response of coupled mechanical system.

#### 3.2. Learning Data-Driven Impulse Response Model of an End Mill

The univariate data is first embedded to a multivariate space to allow the extraction of oscillatory modes. After SVD was performed on the time-delayed embedding data, the first  $r$  SVD components, columns of the  $V$  matrix, were extracted. The first four modes are shown in Figure 6. It is worth noting that, in the decomposed singular vectors,  $v_1$  and  $v_2$  represent together the first intrinsic mode, because  $v_1$  and  $v_2$  share the same frequency (1187.84 Hz) but differ with a phase shift of  $\pi/2$  rad. This can be understood since SVD constructs two sinusoidal functions of the same frequency to represent one underlying mode with a phase shift, like  $A \sin(\omega t) + B \cos(\omega t) = \sqrt{A^2 + B^2} \cos(\omega t + \varphi)$ , where  $\varphi$  is a phase angle. Similarly,  $v_3$  and  $v_4$  represent the second intrinsic mode (942.08 Hz).

Figure 7 shows the frequency spectrum of the mixed-mode impulse response. It can be seen that the decomposed oscillation frequencies (1187.84 Hz and 942.08 Hz) from SVD are relatively close to the dominant frequencies (1186.13 and 938.67 Hz) in the impulse response.

The decomposed modes after the first four components do not possess a consistent convergent dynamical process and are not shown. However, to evaluate their effects and determine how many modes are needed for accurate response modeling, we tested the results by using the first four and first six SVD components.

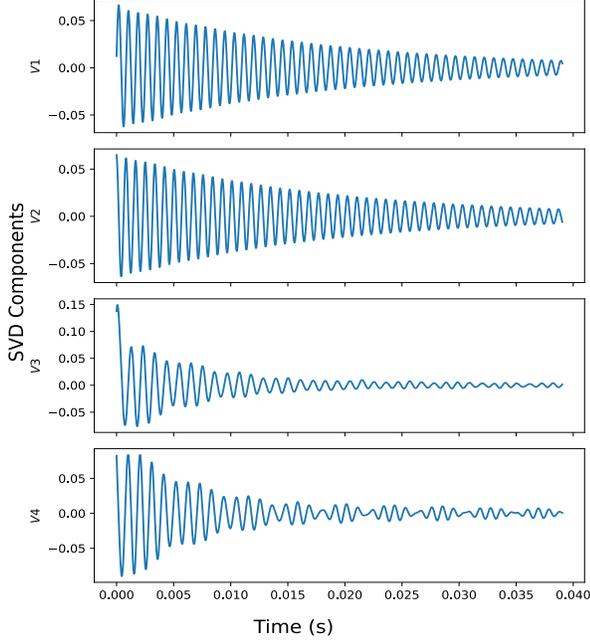


Figure 6: Plots of decomposed SVD components,  $v_1$  to  $v_4$ .

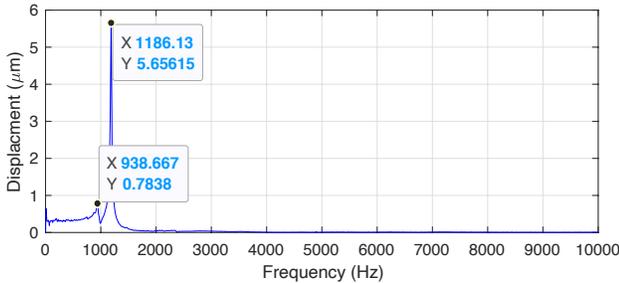


Figure 7. The frequency spectrum of the impulse response.

One difficulty with learning oscillational responses is that to learn an ODE, each point on the trajectories must be unique over time. However, for trajectories that exhibit oscillation, the points along the trajectory are not unique, as shown in Figure 8, where multiple points on the curve at different times can take exactly the same value.

To uniquely define an oscillational trajectory, data with more than one dimension are needed. In order to solve this issue, a NODE model with two-dimensional (2D) inputs is used for the ODE learning. Since each pair of the temporal modes in  $V$  correspond essentially to one intrinsic mode, a pair of temporal components can be learned together by a two-dimensional NODE model. Thus, a two-dimensional ODE

can be learned with the combination of SVD components  $v_1$  and  $v_2$  for the first mode, components  $v_3$  and  $v_4$  for the second mode, etc. This allows, in most scenarios, a unique state along the entire trajectory; the trajectory does not cross itself at separate times. Figure 9 shows the two-dimensional trajectory of SVD components  $v_1$  and  $v_2$ , with the red asterisk showing the initial condition at  $t = 0$ .

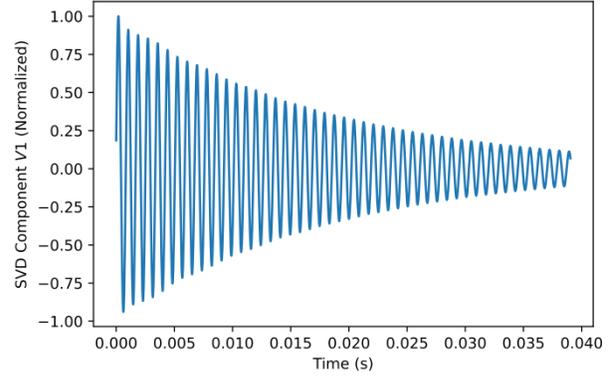


Figure 8: SVD component 1,  $v_1$ .

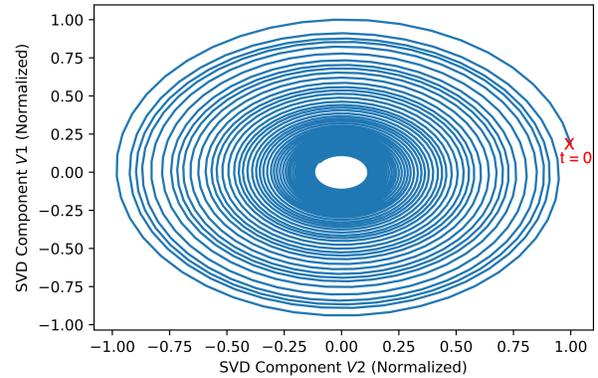


Figure 9: 2D system of SVD components,  $v_1$  and  $v_2$ .

A two-dimensional NODE simply uses two-dimensional vectors as the input and output:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = nn \left( \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \right) \quad (6)$$

$$\begin{bmatrix} x_1^{t_1} \\ x_2^{t_1} \end{bmatrix} = ODESolver \left( nn \left( \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \right), \begin{bmatrix} x_1^{t_0} \\ x_2^{t_0} \end{bmatrix}, t_0, t_1 \right) \quad (7)$$

The two-dimensional NODE obtains a unique solution in the functional space such that the system response is sufficiently learned. The learning process of the model is explained as follows. First, the model iteratively solves an initial value problem, where a random point along the curve is taken as the initial condition and then the model predicts a short segment forward in time. Then, this process is repeated for  $m$  initial conditions in a batch. The loss is then calculated between the predicted response and the true response. This process is repeated until the loss converges. Finally, the

learned modes are reconstructed back to the original mixed-mode impulse response, as illustrated in Figure 3.

#### 4. RESULTS AND DISCUSSIONS

This section presents the results of the proposed method for impulse response modeling. Section 4.1 presents the reconstruction results with NODEs to determine if the multi-modal response can be approximated from learned SVD components using full-length data. Then, in Section 4.2, we show the prediction results from the learned data-driven ODE function when only a small segment of data is used for training.

##### 4.1. Case Study A: Learning of Multi-Modal Dynamics from SVD Components using Full-Length Data

The task of the first case study is to determine whether or not a multi-modal impulse response can be effectively reconstructed from learned SVD components. The measured impulse response of the end mill in the machine tool spindle, as described in Section 3.1, was used for this case study. For the first evaluation, the entire measured impulse response was used to perform the time-delay embedding and thereafter, the SVD. A total of 6000 data points, about 0.12 seconds of data, were used to construct a 1000-step time-delayed embedding of size  $[5000 \times 1000]$ , following Eq (5). The first six columns of the  $V$  matrix, which represent the first six temporal modes, were taken and learned by the two-dimensional NODE model. Those SVD components can also be called ‘principal component trajectories’, as in (Dylewsky *et al.*, 2022), or simply ‘modes’. After each principal component trajectory is learned by a NODE, the impulse response was reconstructed with the original  $U$  and  $\tilde{\Sigma}$  matrices. Figure 10 shows the learned first SVD component,  $v_1$ , versus the decomposed mode. The reconstruction shows high accuracy, with slight accumulated frequency mismatch over a relatively long period.

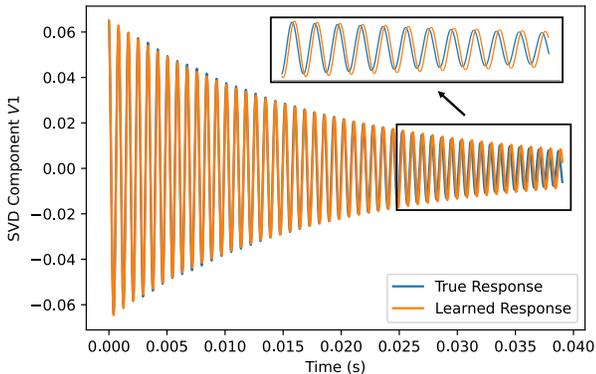


Figure 10: Learned SVD component  $v_1$  with NODE.

The NODE model exhibits some difficulties when learning the SVD components during the training process. The learned ODE model can fail to converge sufficiently: the loss can stop

decreasing and remain at a value that is relatively high, resulting in convergence failures. To remedy this issue, the model is simply reinitialized and trained again. The primary reason is the optimization process can get stuck in a local minimum and fails to converge to a lower loss, due to poor initialization. Similar instability issues have also been reported by other ODE papers (Norcliffe *et al.*, 2020).

Two different reconstructions were made with four or six respective SVD components. The product of the original  $U$ ,  $\tilde{\Sigma}$ , and the new  $V_{pred}$  matrices are used to obtain the reconstructed results. Firstly, the four-component reconstruction was performed. The results are shown in Figure 11 and Figure 12. It can be seen that with four components, the reconstructed impulse response is mostly accurate in both the amplitude and frequency. The learned response has a slightly lower amplitude as can be seen when time trends to 0.04 s (see Figure 11). As the result is evaluated further in time for 0.1 s, the learned response follows an accurate decay trend as the true response (see Figure 12).

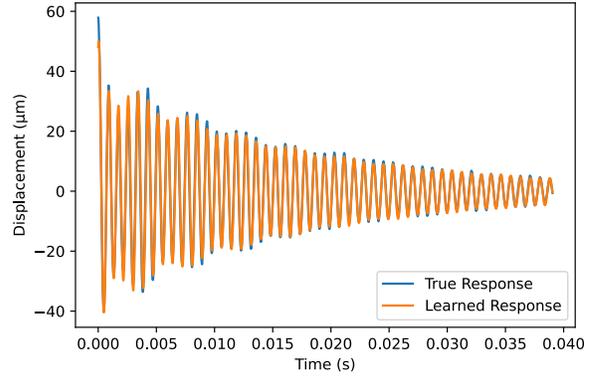


Figure 11: Mixed-mode impulse response reconstruction on four SVD components over 0.04 s.

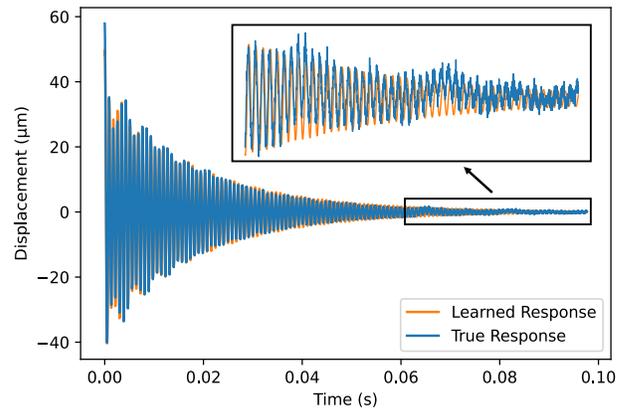


Figure 12: Mixed-mode impulse response reconstruction on four SVD components over 0.1 s.

The reconstructed impulse responses with the first six SVD components are shown in Figure 13 (0.04 s window) and Figure 14 (0.1 s window). It can be seen in Figure 13 that the model again effectively learns the multi-modal response

since the reconstructed result matches the true response quite well. As the time approaches 0.04 s (see Figure 13), it can be seen that there is a slight discrepancy in terms of amplitude. When the plot is extended to 0.1 s (see Figure 14), an overfitting trend develops as time progresses, as seen in the inset of Figure 14. In that case, the true response begins to dampen out while the learned response still exhibits strong multi-modal characteristics.

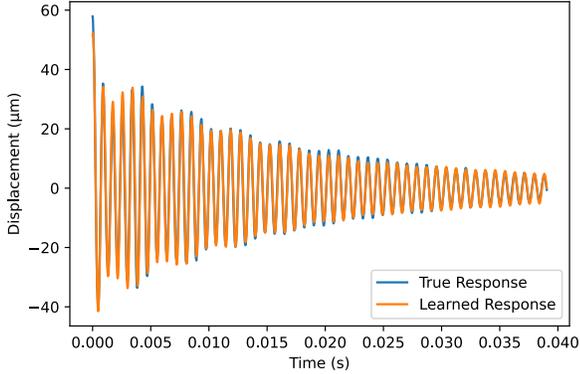


Figure 13: Mixed-mode response reconstruction from 2D NODE model on six SVD components over 0.04 s.

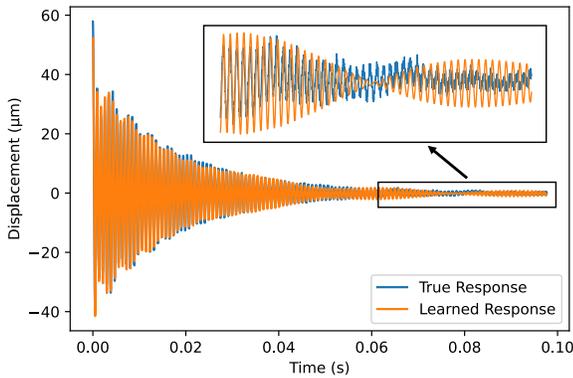


Figure 14: Mixed-mode response reconstruction from 2D NODE model on six SVD components over 0.1 s.

Components for Model	RMSE
4	1.3350 ( $\mu\text{m}$ )
6	1.3853 ( $\mu\text{m}$ )

Table I: RMSE of four and six component reconstructions.

The overfitting that existed in the six-component reconstruction does not exist in the four-component reconstruction, which leads to the conclusion that the mode approximation with SVD components  $v_5$  and  $v_6$  were the sources of the long-term overfitting for the model. Table I shows each model's reconstruction error in terms of the root mean square error (RMSE), which reveals that it is better to approximate the multi-modal response with only the first four

principal component trajectories of the time-delay embedding SVD.

#### 4.2. Case Study B: Prediction of the Multi-Modal Impulse Response using Less Data

After proving that the multi-modal response could be learned by the proposed learning methods, the remaining question is whether this model is capable of predicting the response into the future. In order to evaluate this, a time-delay embedding was performed over only the first 1500 points instead of the full 6000 points. This translated to about 0.03 s of data. In this case, 500 delay steps are used, resulting in the columns of  $V$  with a length of only 1000 points (about 0.02 s of data). As previously discussed, the components beyond the first four principal component trajectories led to model inaccuracies as time progresses. Therefore, only the first four SVD components were used for this case study.

The components were learned in the same manner as before, except that the NODE model was trained over only about 0.02 s of data, which equates to 1000 points along the  $V$  matrix. Next, the model performed predictions up to about 0.1 s, or 5000 points in length. When the reconstruction was performed, the original  $\tilde{\Sigma}$  matrices had to be padded with zeros to predict up to a time of 0.1 s.

The reconstructed prediction from the first four principal component trajectories is shown in Figure 15 and Figure 16. Figure 15 shows that in the near term (0.04 s), the prediction captures fairly well the amplitude and the frequency of the multi-modal response. Figure 16 shows the long-term predicted response versus the true response. It can be seen that the predicted response matches the true response with relatively minor differences in amplitude after 0.04 s. In contrast, it can also be seen that the predicted response has a slightly larger amplitude compared to the true impulse response as time progresses.

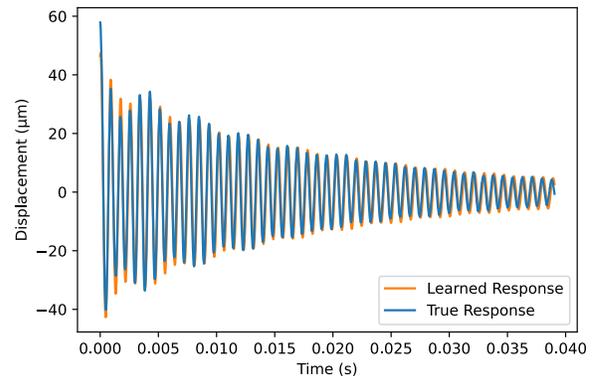


Figure 15: Mixed-mode response prediction on four SVD components over 0.04 s.

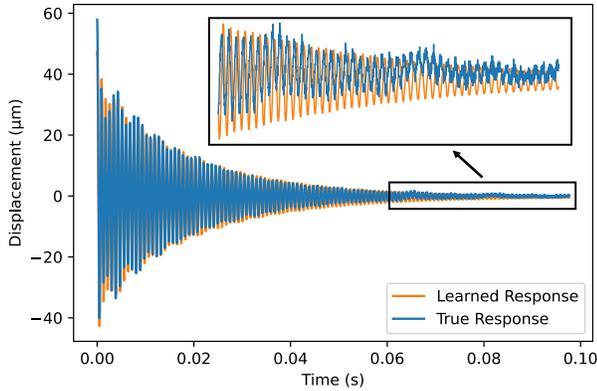


Figure 16: Mixed-mode response prediction on four SVD components over 0.1 s.

Table II summarizes the RMSE of the prediction, which is higher than that for the model learned from the full-length data (see Table I), but is still relatively small compared to the measured displacements. The prediction was also evaluated with only 1000 points with 500 time-delay embeddings, resulting in 500 points length (0.01 s) for each column of  $V$  matrix. However, the training of the NODE model suffers more instability and non-convergence results due to the shorter training window.

Test	RMSE
Prediction	2.8417 ( $\mu\text{m}$ )

Table II: RMSE of four component prediction reconstruction.

## 5. CONCLUSIONS AND FUTURE WORK

Modeling in-situ dynamic system with unknown physics is an unsolved challenge. We present a machine learning method for impulse response function modeling in this paper. The learned data-driven differential equations for in-situ response modeling can be further used for real-time response prediction and in-process performance optimization. In-situ dynamic modeling can help close the loop of real-time feedback control for performance improvement and quality control, especially in cyber-physical manufacturing systems.

The task of learning a multi-modal response was approached by first performing a SVD over a time-delayed embedding to further allow a NODE model to learn the principal component trajectories and then reconstruct the multi-modal impulse response. It was found that, by learning the dominant principal component trajectories with a NODE model, the multi-modal impulse response could be accurately reconstructed. Additionally, it was determined that prediction of the response forward in time could be achieved with a small segment of response data. However, the accuracy of

prediction generally decreases for longer prediction times. Also, the accuracy improves when a longer segment is provided for training and learning.

In the future, more robust mode separation methods, for example, to learn the implicit Laplacian basis, can be investigated besides SVD to get clean separated mode. Furthermore, continuous in-situ response predictions with external excitations can be approached by uniting the proposed methods within a convolutional kernel.

## ACKNOWLEDGEMENT

Jacob Fabro, Yongzhi Qu, and Reese Eischens acknowledge the Department of Commerce of the United States for partial support of this research under contract No. 70NANB20H175.

## 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.

## REFERENCES

- Chen, R. T. Q., Rubanova, Y., Bettencourt, J., Duvenaud, D. (2018). Neural Ordinary Differential Equations, *Advances in Neural Information Processing Systems*, 31, 6571-6583.
- De Lauro, E., De Martino, S., Falanga, M., Ciaramella, A., & Tagliaferri, R. (2005). Complexity of time series associated to dynamical systems inferred from independent component analysis. *Physical Review E*, 72(4), 046712.
- Dormand, J. R., & Prince, P. J. (1980). A family of embedded Runge-Kutta formulae. *Journal of Computational and Applied Mathematics*, 6(1), 19-26.
- Dylewsky, D., Kaiser, E., Brunton, S. L., & Kutz, J. N. (2022). Principal component trajectories for modeling spectrally continuous dynamics as forced linear systems. *Physical Review E*, 105(1), 015312.
- Fabro, J., Vogl, G. W., & Qu, Y. (2022). Run-time cutting force estimation based on learned nonlinear frequency response function. *Journal of Manufacturing Science and Engineering*, 144(9), 091002.
- Huang, N. E., Shen, Z., Long, S. R., Wu, M. C., Shih, H. H., Zheng, Q., Yen, N. C., Tung, C. C., & Liu, H. H. (1998). The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society of London*.

*Series A: Mathematical, Physical and Engineering Sciences*, 454(1971), 903-995.

- Hyvärinen, A. (2013). Independent component analysis: Recent advances. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1984), 20110534.
- Kutz, J. N., Brunton, S. L., Brunton, B. W., and Proctor J. (2016). *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*. SIAM
- Lagaris, I. E., Likas, A., & Fotiadis, D. I. (1998). Artificial neural networks for solving ordinary and partial differential equations. *IEEE Transactions on Neural Networks*, 9(5), 987-1000.
- Norcliffe, A., Bodnar, C., Day, B., Simidjievski, N., & Liò, P. (2020). On second order behaviour in augmented neural ODEs. *Advances in Neural Information Processing Systems*, 33, 5911-5921.
- 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*, 378, 686-707.

## BIOGRAPHIES



**Jacob Q. Fabro** received the Bachelor's of Science in Mechanical Engineering degree and began the Master's of Science in Mechanical Engineering and Master's of Business Administration degrees at the University of Minnesota Duluth, Duluth, Minnesota, USA in 2020. He programmed and oversaw

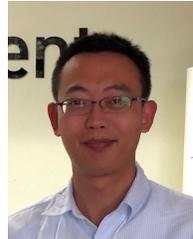
operation of industrial welding robots to ensure both production rate and ASME Pressure Vessel Code compliance were met as a Automation Programming Engineer Intern and Manufacturing Engineering Co-op at Trane Technologies. He currently works as a Research and Teaching Assistant at the University of Minnesota Duluth, Duluth, Minnesota, USA.



**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. Currently, Dr. Vogl develops

sensor-based solutions for real-time health assessment of components and processes within machine tools. For his contributions, Dr. Vogl is the recipient of a NIST Engineering Laboratory Mentoring Award and NIST Colleagues' Choice Award.



**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.



**Reese C. Eischens** received the Bachelor's of Science in Mechanical Engineering degree and will begin the Master's of Science in Mechanical Engineering at the University of Minnesota Duluth, Duluth, Minnesota, USA in 2022. He assisted with the redesign of a hydraulic test bench to

diagnose and differentiate between types of hydraulic cylinder leaks at Atlec Industries. He currently works as a Research Assistant at the University of Minnesota Duluth, Duluth, Minnesota, USA.