Model updating in rotor dynamic system using FEM based neural network

Rishith Ellath Meethal
Technische Universität München

Mohamed Khalil
Siemens

Birgit Obst
Siemens

Kai-Uwe Bletzinger
Technische Universität München

Model updating in rotor dynamic system using FEM based neural network

R. Ellath Meethal [1,2], **M. Khalil** [1], **B. Obst** [1], **K.-U. Bletzinger** [2], **R. Wüchner** [3]
[1] Technology, Siemens AG,
Munich, Germany
e-mail: **rishith.ellath_meethal@siemens.com**

[2] Chair of structural analysis, Technical University of Munich,
Munich, Germany

[3] Institute of Structural Analysis, Technische Universität Braunschweig,
Braunschweig, Germany

Abstract
Executable Digital Twins (xDTs) assist maintenance engineers by calculating current and future health status of physical assets based on operational conditions. In xDTs, the simulation describes the ideal physical behavior of the system, whereas the real data from sensors assist in updating the xDTs following the changes to the physical asset. We introduce a novel algorithm combining neural networks (NN) and finite element method (FEM) for the system updating in rotor dynamic systems xDTs. In the presented algorithm, the known parts of the system are modeled using FEM package, whereas the unknown parts such as bearing parameters is modeled using a neural network. The discretized form of the residual of the governing partial differential equation (PDE) is backpropagated to learn the neural network parameters. The estimated parameters are used to predict the vibration in an unseen scenario. The algorithm facilitates an ecosystem combining numerical methods and data science to result in more accurate and continuously-updating xDTs.

1 Introduction

In the early 1960s, NASA has used physical twins on the ground to match their systems in space. A famous example is the digital twin they used to assess and simulate conditions on board Apollo 13. That idea of twins expanded over the years and gave rise to the imperative Digital Twins (DTs) [1, 2] that we see today. As per Boschert et al. in [2], DT itself refers to a comprehensive physical and functional description of a component, product, or system, which includes more or less all information which could be useful in all—the current and subsequent—lifecycle phases. It is one of the essential tools of Industry 4.0 and enables the seamless integration of digital and physical space. With the support of the Internet of Things (IoT), DTs found themselves valuable in the design, production, health management, maintenance, etc.

The unavoidable nature of DTs introduced the concept of executable DTs (xDTs). xDTs are the self-contained realization of DTs. As per [3], An Executable Digital Twin is a specific encapsulated realization of a Digital Twin with its execution engines. It can be deployed on the cloud, on-premise, or on edge. This makes the engineering design, knowledge, and findings available for the entire lifecycle of the product. The xDTs enables better decision-making and alleviate the potential failure of the product. The main characteristic is that it can be executed on a system or asset as a simulation to support in condition monitoring or failure diagnosis as well as forecasting potential or alternative future states of a product/physical asset. In xDTs, the simulation acts as the core, whereas real data from sensors assist in updating the xDTs following the changes to the physical asset.

Historically simulation started as a tool to replace costly hardware and numerous experiments. Over the years, simulation grew from a simple design tool to much more. For example, simulation is used in systems like xDTs along with operation to assist operation decisions by analyzing what-if scenarios. Simulation is also part of fault diagnosis or predictive maintenance scenarios. Simulating equipment or product accurately requires updated details of the product and well calibrated models. Most of the time, the properties of the equipment or product vary over time and result in non-matching simulation and sensor data. But with the help of xDTs, it is possible to update the model to match the real asset and perform more accurate simulations.

This contribution discusses model updating in an xDT for rotordynamic systems. Rotordynamic systems are an essential part of a wide range of industries in the form of turbines, generators, motors, compressors, blowers, etc. The accurate prediction of its dynamic behavior is vital for uninterrupted operation and safety. The estimation of bearing coefficients has been a primary barrier to predicting or simulating the dynamic behavior of such systems. For example, the dynamic coefficients of fluid bearings can be theoretically estimated using Reynolds equations. But it may result in wrong values as their inputs, like the bearing static load, the viscosity of the fluid, operating temperature, pressure, etc., may not follow the approximations used for theoretical estimation. Therefore, experimental estimate is essential in calculating and updating the exact bearing coefficients to increase the simulation accuracy.

In this contribution, we demonstrate how rotordynamic system update for bearing coefficients is done with the help of a hybrid algorithm called FEM-NN. This algorithm combines the known parts of the system modeled using the Finite Element Method (FEM) [4] and unknown parts using the neural network (NN) [5]. The discretized form of the governing PDE is backpropagated [6] for the training of the neural network. In the following, Section 2 investigates existing methods for bearing parameter identification in rotordynamic systems. Section 3 explains the introduced algorithm and details the procedure for system updating. In Section 4, numerical results on simulated and real data are presented.

2 State of the art

Developments in rotordynamic evolved according to the interplay between theory and practice. Requirements from industry were the driving force for theoretical developments. In the early days, in the 1860s, the concentration was on basic modeling of the spinning shaft to identify effects like whirling (Ranking 1869). In 1957 the role of fluid film bearing in rotor dynamics was explained in a graphical way by Newkirk [7]. Tiwari et al. provided a detailed review of the conventional methods of bearing parameter estimation in [8]. This include methods based on incremental static load, dynamic load, excited load, unbalance mass [8], impact hammer [9], impulse [9, 10] etc. Most of the methods either used bearings in isolation or a rigid shaft. In 2002, Tiwari et al. [11] introduced a method treating shaft as flexible for the identification of speed-dependent bearing parameters. Normally such methods do not consider the foundation flexibility, and its contribution is not estimated. And the method also requires matrix inversion, which demands the consideration of the condition number of the matrix and methods to improve the condition number. Another problem associated with the method is that the data corresponding to each speed has to be considered independently to estimate the parameter corresponding to the particular speed. A model mapping speed to parameters was not possible with such methods.

A Kalman filter based approach was developed by Yang et al. in [12] for the dynamic bearing coefficient identification in which displacement of the shaft is measured only at one location. Kalmar filter was employed to estimate displacements of the shaft at bearings locations. The method provides a more practical approach for the estimation of bearing parameters but still uses the conventional least-square method for calculating bearing parameters from the calculated displacement at bearing locations. Least-square methods may not guarantee global optimum, especially when noise or other uncertain factors are present. Kriging surrogate model and Differential Evolution (DE) algorithm was employed in parameter identification of rotor-bearing system in [13]. It is found that the Kriging surrogate model is more robust to the noise and costs less time but at the cost of generating a dataset prior to the initial surrogate model creation.

A neural network based approach was introduced in [14] for learning bearing parameters against rotational speed. They used a numerically simulated dataset of possible parameters and the corresponding critical speeds. Then a neural network surrogate was created to predict bearing parameters given the critical speed.

It was observed that the neural network approach resulted in more accurate results compared to conventional regression methods. They attribute this mainly to the universal approximation capacity of neural networks. But this method demands the creation of a dataset prior.

In this contribution, we developed an algorithm based on FEM-NN introduced in [15] which does not require a prior dataset. It uses the force and displacement vectors in the frequency domain and known parts of the stiffness, damping, and mass matrices as input and learns the unknown bearing coefficients. The proposed method results in a surrogate model which maps variables like the speed of rotation, the temperature of oil, the viscosity of oil, etc., to the bearing parameters. We restrict our study to the speed of rotation in the following Sections. The Section 3 details the proposed method.

3 Algorithm

Consider a rotordynamic system as given in Figure 1. The system consists of a flexible shaft driven at a rotating speed of ω and supported on two bearings. The shaft contains disks carrying unbalance mass. The equation of motion of the system in frequency domain is given below

$$\begin{aligned} \left[-\omega^2 M + j\omega(G\omega + C) + K\right] Q &= F \\ DQ &= F \end{aligned} \tag{1}$$

where M, G, C and K are the mass, gyroscopic, damping and stiffness matrices respectively. Q is the displacement vector and F is the force vector in the frequency domain. $D = \left[-\omega^2 M + j\omega(G\omega + C) + K\right]$ is called dynamic stiffness matrix.

The FEM-NN algorithm introduced in [15] is used for the learning of bearing parameters. The shaft is modeled with two-node Timoschenko beam elements [16] with four degrees of freedom (DOF) at each node. For the inverse problems, the algorithm models unknowns as neural network model. In our case, the bearing parameters are modeled as a neural network output as in Equation 2. Equation 2 models $x-$ and $y-$ direction stiffnesses k_{xx} and k_{yy} and cross-coupling stiffnesses k_{xy} and k_{yx} as the output of a neural network.

$$\begin{pmatrix} k_{xx} \\ k_{xy} \\ k_{yx} \\ k_{yy} \end{pmatrix} = f(\omega, \theta) \tag{2}$$

Similarly, unknown damping coefficients can also be modeled as output of a neural network. The modeled parameters are assembled to the systems stiffness or damping matrices accordingly to results in the final dynamic stiffness matrix. Dynamic stiffness matrix can be written as

$$K = \begin{pmatrix} D_{rr} & D_{rb} \\ D_{br} & D_{bb} \end{pmatrix} \tag{3}$$

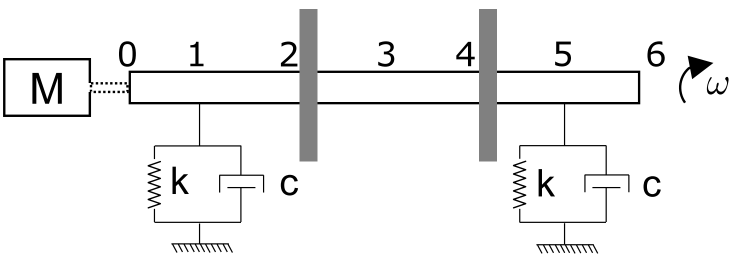


Figure 1: Rotordynamic system

where D_{rr} represents the nodes only part of rotor (0 and 6 in Figure 1), D_{rb} and $D_b r$ represents nodes which are connected to the bearing nodes (nodes 0, 2, 4 and 6 in Figure 1) and D_{bb} represents nodes where bearings are attached (nodes 1 and 5 in Figure 1). We know that the D_{bb} will have contribution D_R from rotor and D_B from bearing. So, one can write

$$D_{bb} = D_R + D_B \tag{4}$$

In this, D_R is known and D_B is unknown and hence modeled using a neural network as in Equation 2. The displacement q of the nodes can be calculated using the measured vibration and known unbalance force. From this, equation of motion and corresponding residual can be written as

$$\begin{pmatrix} D_{rr} & D_{rb} \\ D_{br} & D_{bb} \end{pmatrix} \begin{pmatrix} q_r \\ q_b \end{pmatrix} = \begin{pmatrix} f_r \\ f_b \end{pmatrix} \tag{5}$$

residual r is

$$r = \begin{pmatrix} D_{rr} & D_{rb} \\ D_{br} & D_{bb} \end{pmatrix} \begin{pmatrix} q_r \\ q_b \end{pmatrix} - \begin{pmatrix} f_r \\ f_b \end{pmatrix} \tag{6}$$

The second norm of the calculated residual $||r||_2$ is backpropagated to learn the neural network model. This require calculation of the gradient of residual with respect to the neural network. Since the contribution of neural network lies in D_{bb}, gradient $\frac{\partial r}{\partial D_{bb}}$ is calculated as follows

$$\frac{\partial \delta}{\partial D_{bb}} = \frac{1}{\sqrt{(D_{rr}q_r + D_{rb}q_b - f_r)^2 + (D_{br}U_r r + D_{bb}U_b - f_b)^2}}(D_{br}q_r + D_{bb}q_b - f_b)q_b \tag{7}$$

Algorithm 1 FEM-NN training for rotordynamic bearing parameter identification problems

procedure TRAIN
  Read simulation parameters
  Initialize neural network
  Initialize weights and biases
  Initialize FEM Package
  Let L be the number of layers in the neural network
  while not Stop Criterion **do**
    Compute the system matrices $D_{rr}, D_{rb}, D_{br}, f_r, f_b, q_r$ and q_b
    $D_B \leftarrow$ Neural network prediction
    Compute $D_{bb} = D_B + D_R$ where D_R is the known contribution from rotor
    Compute the residual $r = DQ - F$
    Compute the loss as the Euclidean norm of the residual vector $\delta = ||r||_2$
    Compute the derivative $\frac{\partial \delta}{\partial D_{bb}}$
    Compute the derivative $\frac{\partial \delta}{\partial D_B}$ from $\frac{\partial \delta}{\partial D_{bb}}$ and $D_{bb} = D_B + D_R$
    for all $l \in \{1, \dots, L\}$ **do**
      Compute the derivative using chain rule $\frac{\partial \delta}{\partial w_l} = \frac{\partial \delta}{\partial D_B}\frac{\partial D_B}{\partial w_l}$
      Update trainable parameters (weights and biases) $w_l = w_l - \eta\frac{\partial \delta}{\partial w_l}$
    end for
  end while
end procedure

The gradient of loss δ with respect to neural network output can be easily calculated from Equation 7 as D_{bb} only involves summation of neural network output to known contribution from rotor. Detailed algorithm on calculating gradients can be seen in [15]. The proposed procedure for finding bearing parameters in given is Algorithm 1. Once a rained model is obtained, the model can be used to predict the bearing parameters for the given speed. The predicted parameters can be used for the forward simulation to calculate expected vibration for different scenarios.

4 Results

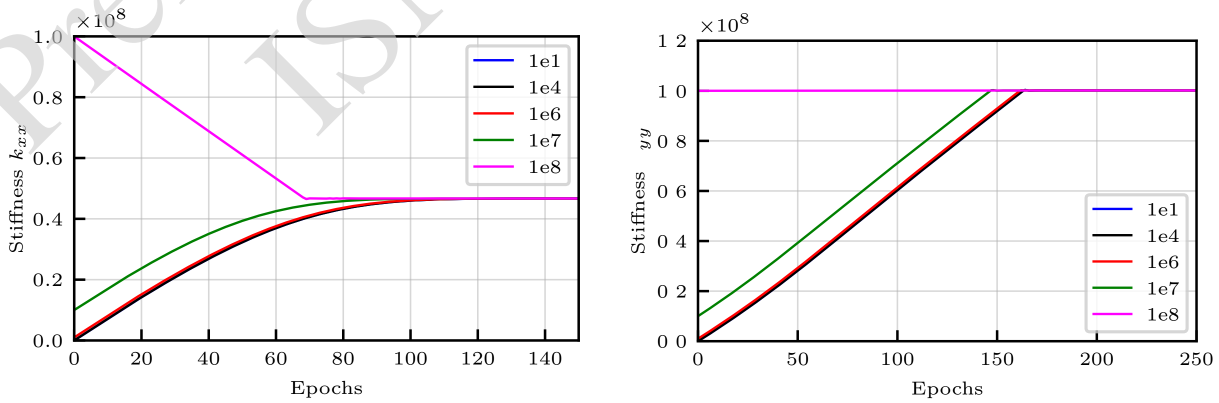
In this section proposed algorithm is applied to two examples. In both examples, we consider a rotordynamic system with a flexible shaft supported on two bearings. ROSS, an open-source software created explicitly for rotordynamic systems [17], is used for modeling the system. Different elements of a rotorynamic system like shaft, disc, bearings can be modeled using software. Vibration data and system matrices are created using ROSS. Optuna tuner [18] is used for the hyperparameter tuning of all the neural network models in the examples. A Tree-structured Parzen Estimator (TPE) sampler was used in the tuning process. The training was performed using an Adam optimizer [19].

4.1 Flexible shaft supported on ball bearing

In example 1, we consider a flexible shaft with a disc supported on two ball bearings. The shaft is divided into two elements and modelled as explained in Section 3. The ball bearing consisting of 8 balls each having 0.03m diameter is modeled following [20]. The cross-coupling stiffness and damping are not modeled in this case. Damping is low in rolling-element bearings, typically in the range of $1e^{-5} \times k_{xx}$, and hence is not learned in this case. So, only x-direction stiffness k_{xx} and y-direction stiffness k_{yy} are learned. The modeled rotordynamic system is used to generate data and then test the algorithm explained in Section 3. Since the bearing parameters are constant for a ball bearing, they are modeled as a single trainable parameter. Following the algorithm in Section 3 the parameters were trained till convergence. The modeled and trained bearing parameters are given in Table 1. It can be observed that the algorithm is able to calculate exact parameters with less than 1% error.

Table 1: Modeled and learned parameters of ball bearing

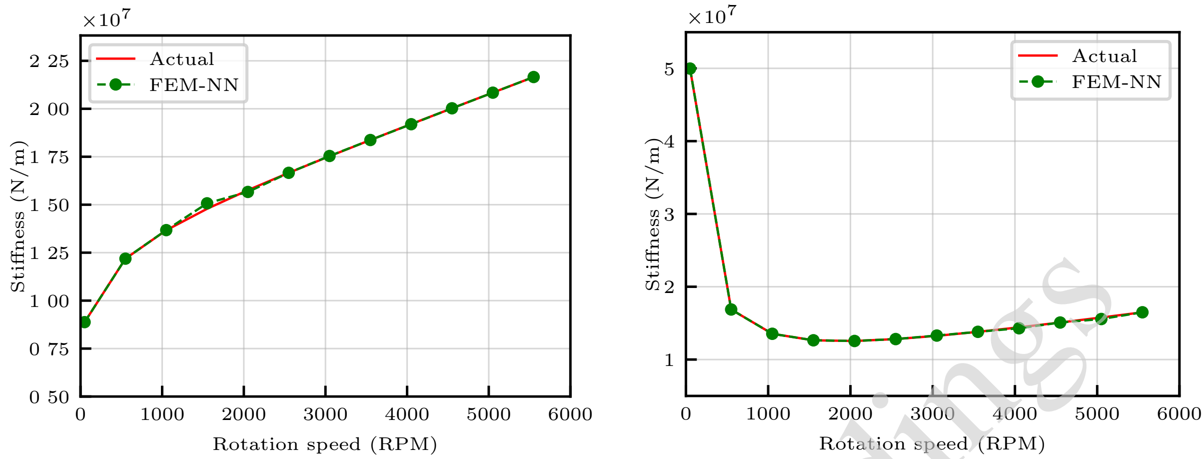
Bearing parameter	Modeled	Learned	Percentage error
k_{xx}(N/m)	464.16883×10^5	466.80013×10^5	0.56
k_{yy} (N/m)	1009.06269×10^5	1001.13943×10^5	0.78



(a) Prediction of stiffness k_{xx}

(b) Prediction of stiffness k_{yy}

Figure 2: Learning stiffness for different initial guesses



(a) Prediction of stiffness k_{xx} (b) Prediction of stiffness k_{yy}

Figure 3: Speed dependent bearing stiffness of a hydrostatic bearing actual vs learned

In Figure 2 robustness of the algorithm is shown for different initial guess for the stiffness values. Initial guesses from 1 to 1×10^8 were used. One can observe that for any initial guesses the algorithm converges to the actual values.

4.2 Flexible shaft supported on hydrostatic bearing

In a second example, we consider a flexible shaft with two discs supported on hydrostatic bearing. The dynamic bearing coefficients of a hydrostatic bearing depends on the dimensionless number called Sommerfeld number S

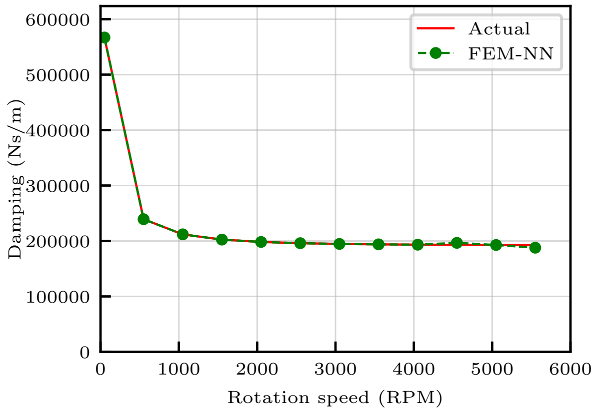
$$S = (\frac{\mu \Omega R L}{W})(\frac{R}{c_r})^2(\frac{L}{D})^2 \tag{8}$$

where W is the static load, μ is the lubricant viscosity, Ω is the journal rotating speed which is equal to the rotational speed of the shaft ω, D is the bearing bore, R is the journal radius, L is the bearing length and c_r is the bearing radial clearance. So, for a given application they are functions of rotating speed Ω.

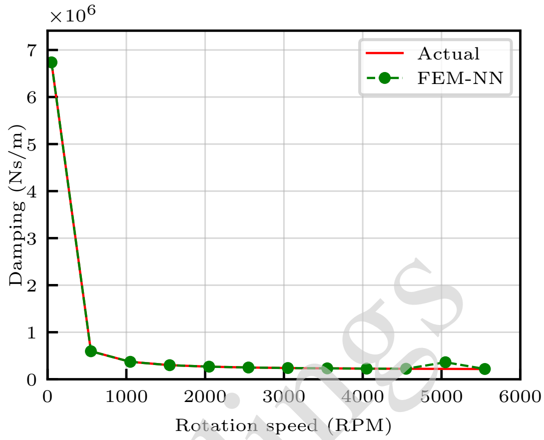
$$K = f(\Omega, \theta) \tag{9}$$

In this example we model the unknown bearing parameter as a neural network which takes the rotating speed as the input. Similar to the previous example we model only the x- and y- direction stiffnesses. Even though the cross-coupling stiffnesses and damping coefficients are relevant in a hydrostatic bearing they are assumed to be known in this study. More details regarding the assumptions are given in Section 5. Three unbalance configurations were considered for the data generation. For each unbalance configuration the vibration data, system matrices without the x- and y- direction stiffness from bearings and force vectors are generated starting from a speed of 100 rpm upto 6000 rpm with an interval of 10. Following the algorithm in Section 3 the neural network was trained and the model was created to find bearing coefficients from rotating speed. The results of bearing coefficients in x and y directions are given in Figures 3 ans 4.

It can be observed that the model was able to learn the relation between rotational speed and coefficients very well. Excellent agreement between modeled and estimated parameters can be seen. The error between predicted and estimated values are again less than 1% for both stiffnesses (0.002% for $x-$ diretion and 0.001% for $y-$ direction respectively). Compared to $x-$ direction stiffness, the $y-$ direction stiffness function is more complex to learn. Nevertheless, the proposed algorithm captured both the functions precisely. The



(a) Prediction of damping coefficient c_{xx}



(b) Prediction of damping coefficient c_{yy}

Figure 4: Speed dependent damping coefficient of a hydrostatic bearing actual vs learned

results for damping coefficients are also predicted accurately. The prediction matches with the modeled with an error of 0.005% in $x-$ direction and 0.054% in $y-$ direction.

4.2.1 Predicting vibration from estimated coefficients

As discussed in the Section 1, the online detection of such parameters can help xDTs to predict the future behavior of the device. In our example, vibration of the system at different nodes can be calculated after obtaining the bearing coefficients. A schematic diagram to enable such an xDT is given in Figure 5.

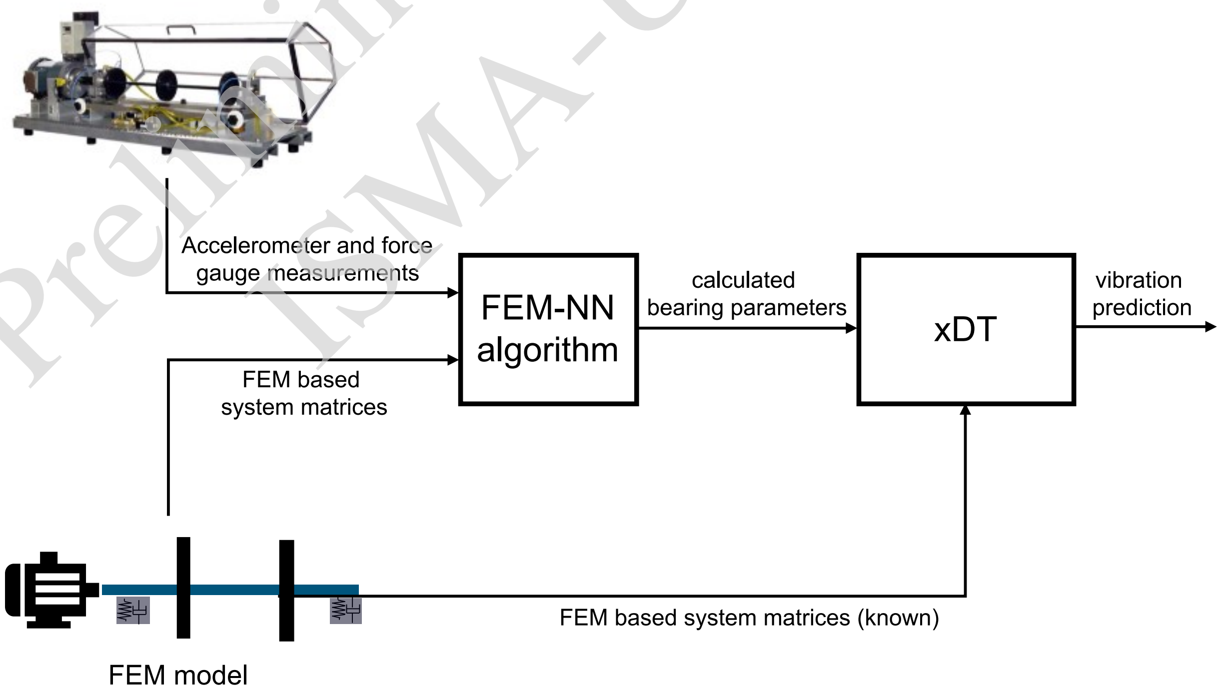
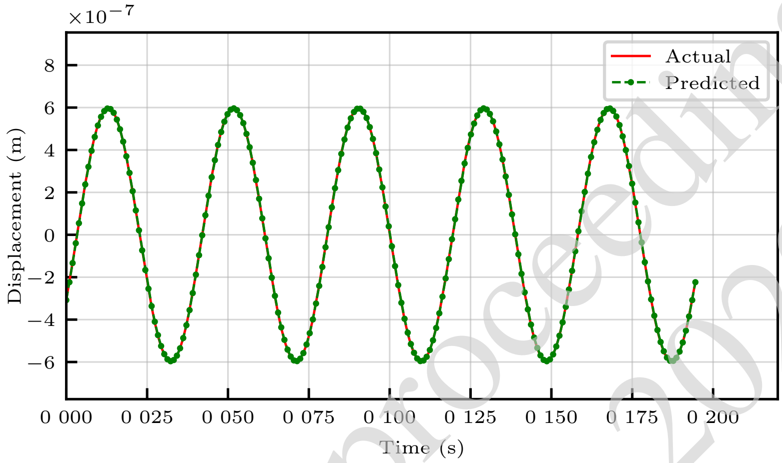


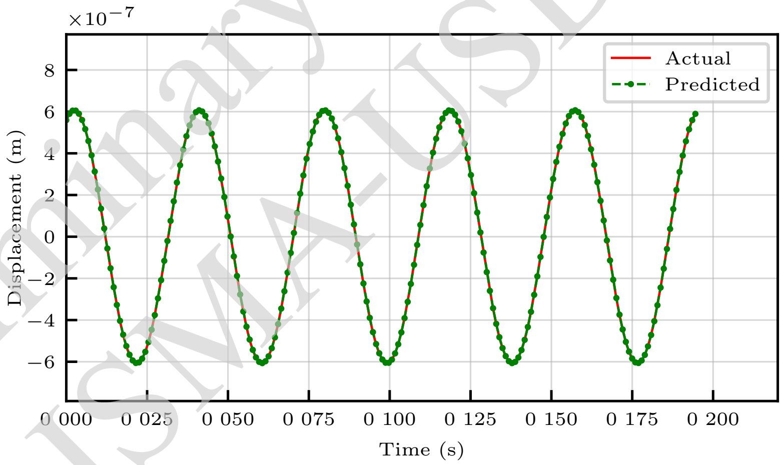
Figure 5: Schematic diagram of an xDT of a rotordynamic system

In such an xDT, the actual force and vibration data can be acquired using different sensors like force gauges and accelerometer. The known parts of the system are modeled using a FEM software. The unknwon bearing coefficients can be calculated using the FEM-NN algorithm. By using the calculated coefficients and known parts of the FEM, xDT can predict the future vibration of the system.

Here we calculated the x and y direction vibration at the disk location for the Example 4.2 at the speed of 1500 rpm. The actual vibration and xDT predicted vibration are plotted in Figure 6. It is to be noted that, here we do not use real data. Hence the actual vibration is also the vibration from the FEM software but using the actual bearing coefficients.



(a) x-direction displacement of the disc node



(b) y-direction displacement of the disc node

Figure 6: Predicted vs actual displacement of the disc node of the rotordynamic system

5 Conclusion and Outlook

We introduced a parameter identification algorithm for the model updating of rotordynamic system bearing parameters based on the FEM-NN hybrid method. The presented algorithm is tested on simulated data to identify constant and speed-dependent bearing parameters. The test resulted in the accurate estimation of unknown parameters. It is found that the method has advantages over conventional methods in accurately determining the bearing parameters. The difficulties in traditional algorithms, such as matrix inversion, preconditioning, etc., are not experienced in the proposed algorithm. Another advantage is that the time-domain calculations are avoided by using the neural network in the frequency domain. The proposed method can be extended to address the non-linear behavior of bearings against different input parameters like the viscosity of oil, temperature, etc., as neural networks successfully learn non-linear functions. In the present work, we only considered speed as the independent variable, but in reality, it can depend on more parameters.

Even though the method proved successful in determining the unknown parameters, more research is required for its practical use. For example, the present study only modeled $x-$ and $y-$ direction stiffnesses. Cross-coupling stiffnesses are neglected. It is observed that the training does not converge when modeling all the parameters. This could be either from the difficulty in hyperparameter tuning of the network or from the ill-posedness of the inverse problem. More research is required in this direction to understand the exact reason for the non-convergence.

Further, the present implementation dealt with artificial data created using FEM software. The next step is to attempt parameter identification on real data. Potential difficulties are in creating vibration and force vectors. The force vector can be constructed using the known unbalances attached to the disc for an experimental setup at a lab. The vibration vector can be constructed by measuring vibrations at the bearing locations. An algorithm to construct a full vibration vector from measuring only at bearing locations for systems like that in Figure 1 is already available in [10].

References

[1] F. Tao, H. Zhang, A. Liu, and A. Y. Nee, "Digital twin in industry: State-of-the-art," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2405–2415, 2018.

[2] S. Boschert and R. Rosen, "Digital twin—the simulation aspect," in *Mechatronic futures*. Springer, 2016, pp. 59–74.

[3] D. Hartmann and H. Van der Auweraer, "Digital twins," in *Progress in Industrial Mathematics: Success Stories*. Springer, 2021, pp. 3–17.

[4] O. C. Zienkiewicz, R. L. Taylor, R. L. Taylor, and R. L. Taylor, *The finite element method: solid mechanics*. Butterworth-heinemann, 2000, vol. 2.

[5] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Neural networks for perception*. Elsevier, 1992, pp. 65–93.

[6] D. E. Rumelhart, R. Durbin, R. Golden, and Y. Chauvin, "Backpropagation: The basic theory," *Backpropagation: Theory, architectures and applications*, pp. 1–34, 1995.

[7] B. Newkirk, "Varieties of shaft disturbances due to fluid films in journal bearings," *Transactions of the American Society of Mechanical Engineers*, vol. 78, no. 5, pp. 985–987, 1956.

[8] A. Lees, "Identification of dynamic bearing parameters: a review," *The Shock and Vibration Digest*, vol. 36, no. 2, pp. 99–124, 2004.

[9] Z. Qiu and A. Tieu, "Identification of sixteen force coefficients of two journal bearings from impulse responses," *Wear*, vol. 212, no. 2, pp. 206–212, 1997.

[10] R. Tiwari and V. Chakravarthy, "Simultaneous identification of residual unbalances and bearing dynamic parameters from impulse responses of rotor–bearing systems," *Mechanical systems and signal processing*, vol. 20, no. 7, pp. 1590–1614, 2006.

[11] R. Tiwari, A. Lees, and M. Friswell, "Identification of speed-dependent bearing parameters," *Journal of sound and vibration*, vol. 254, no. 5, pp. 967–986, 2002.

[12] Y. Kang, Z. Shi, H. Zhang, D. Zhen, and F. Gu, "A novel method for the dynamic coefficients identification of journal bearings using kalman filter," *Sensors*, vol. 20, no. 2, p. 565, 2020.

[13] F. Han, X. Guo, and H. Gao, “Bearing parameter identification of rotor-bearing system based on Kriging surrogate model and evolutionary algorithm,” *Journal of Sound and Vibration*, vol. 332, no. 11, pp. 2659–2671, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.jsv.2012.12.025

[14] I. Pavlenko, V. Simonovskiy, V. Ivanov, J. Zajac, and J. Pitel, “Application of artificial neural network for identification of bearing stiffness characteristics in rotor dynamics analysis,” in *Design, Simulation, Manufacturing: The Innovation Exchange*. Springer, 2018, pp. 325–335.

[15] R. E. Meethal, B. Obst, M. Khalil, A. Ghantasala, A. Kodakkal, K.-U. Bletzinger, and R. Wüchner, “Finite element method-enhanced neural network for forward and inverse problems,” *arXiv preprint arXiv:2205.08321*, 2022.

[16] H. Nelson, “A finite rotating shaft element using timoshenko beam theory,” 1980.

[17] R. Timbó, R. Martins, G. Bachmann, F. Rangel, J. Mota, J. Valério, and T. G. Ritto, “Ross - rotordynamic open source software,” *Journal of Open Source Software*, vol. 5, no. 48, p. 2120, 2020. [Online]. Available: https://doi.org/10.21105/joss.02120

[18] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, “Optuna: A next-generation hyperparameter optimization framework,” in *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[19] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[20] M. I. Friswell, J. E. Penny, S. D. Garvey, and A. W. Lees, *Dynamics of rotating machines*. Cambridge university press, 2010.