Chapter 2
System Identification of Data from Rotating Machinery Using Deep Learning Network Training

R. Wetherington and M. Kasprzyk

Abstract Deep learning algorithms are a powerful tool that can be used in the identification of unique features and behavior of data, yet applying these algorithms in practice can require innovative solutions for different application domains. This paper discusses a simple approach for mapping single-channel data (like that produced by accelerometers and proximity probes) from a rotating machine, , such that it can be used for the training of a simple convolutional neural network. The resulting trained neural network can accurately identify data features to its source with useful accuracy. This method could be used to identify known unique operating system characteristics for prognostic and diagnostic assessment. This paper discusses an evaluation of this approach using real and simulated data, and the results are presented herein.

Keywords AI networks · Neural networks · Rotating machine · Sensor · Mapping data · Diagnostic

Introduction

Developmental testing of rotating machinery can involve dozens of sensor channels and high-speed sampling of the sensor values. Sample rates of tens of thousands of samples per second are not uncommon. The sinusoidal nature of these data present the option of remapping it from 1D to 2D using an X-Y projection method, such as what is used in orbit plots [1]. The projected 2D figures exhibit impressive consistency that are often unique to the source. When the signal-to-noise ratio is high, these signals often present distinguishable features in the data. Therefore, it becomes possible to process the information with a simple convolutional neural network designed for character recognition. The resulting trained AI network can then accurately identify data features to its source to allow insight into the mechanical and design properties that cause the features.

This paper evaluates this approach using data acquired from a six-disc mechanical fault simulator (MFS), as well as from a software-based signal simulator. The MFS sensor data were acquired with a National Instruments data acquisition system and analyzed with MATLAB and its Deep Learning Toolbox,[2]. The trained and validated neural networks were benchmarked with an independent test data set.

Background and Methodology

Dynamic test engineers use orbit plots to analyze the behavior of rotors under operating conditions. They typically do this using a pair of proximity probes placed 90 degrees apart with respect to the rotation axis. Data are digitally acquired at high rates and projected as a scatter plot with the data from one channel projecting along the X-axis and the other channel projecting along the Y-axis. The resulting imagery is interpreted for rotor behavior.

A single axis of sensors are sometimes used to monitor clearances, or general-purpose accelerometers are used for vibration monitoring. In these cases, a secondary set of channels, placed 90 degrees with respect to the first set, is not available. If the rotating article is behaving as a symmetrically mounted rigid body, then a synthesized secondary channel can be approximated from the primary sensor to facilitate the creation of a 2D map. There are two straightforward methods for simulating a second sensor channel positioned 90 degrees from the first sensor.

R. Wetherington · M. Kasprzyk
Oak Ridge National Laboratory, Oak Ridge, TN 37830
e-mail: wetheringtgr@ornl.gov; kasprzykmr@ornl.gov

Thomas Matarazzo et al. (eds.), *Data Science in Engineering, Vol. 11*, Conference Proceedings of the Society for Experimental Mechanics Series, https://doi.org/10.13052/97887-438-0156-6_2

The first method is to take the primary data and duplicate it for the second sensor data, then shift the number of samples to represent 90 degrees of rotation. This is referred to as the “shifted-data” method. This approach is computationally efficient but has two drawbacks. The first is that the fidelity of the shift is limited to discrete samples. Thus, the angular precision is unlikely to produce a perfect 90-degree shift because the sample rate multiplied by 1/4 may not equate to 90 degrees of samples. This error is somewhat minimized by higher sample rates. The second deficiency is for cases in which the rotational speed is changing. The shifted number of samples will change periodically, and the result is a significant change to the resulting orbit plot.

The second method of simulating the second sensor channel is to use the derivative of the primary channel data. This approach works well for signals that are not overly sampled at a high rate because the differential will not be as noticeable. The prominent advantage is speed variations in the test article cause minimal effect, to the differential. While the derivative utilizes negligible processing power, and the difference calculation where the delta amount between each sample is calculated can be a suitable substitute. MATLAB has such a function called “diff”, used for synthesizing the second channel of data, and it is referred to as the “derivative-diff” method.

The basis of this approach is best illustrated in Figure 1, wherein two sensors are located 90 degrees apart along the rotational axis of a rotating body, where ? defines the angle and direction of rotation. If the body is rigid and simple, then the signal picked up by the two sensors will be separated by 90 degrees and will each be a sinusoid. One sensor can be approximated by the sine of the angle of rotation and the other by the cosine of the angle, wherein the cosine is also the first derivative of the sine. Therefore, as shown the figure below, the sensor at 12 o’clock can be approximated by taking the derivative of the sensor at the 3 o’clock, and for a sampled system, shifting it by the number of samples associated with a 1/4 rotation of the body.

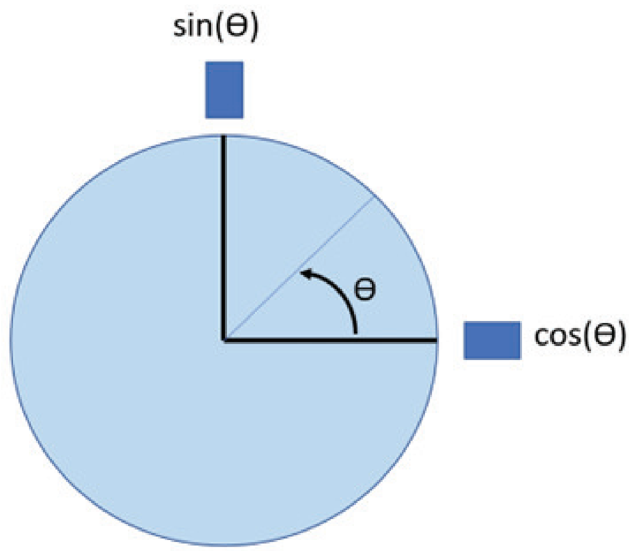


Fig. 1 Relationship of sensor position of a rotating body. Recall that the cosine is the derivative of the sine, which is also a 1/4 of the samples for one revolution.

Mechanical Fault Simulator

Figure 2 shows a mechanical fault simulator (MFS), including two main components: the driver and shaft assembly. A 250W (1/3 hp) induction motor drives the shaft from 0–66.5 Hz (0–4,000 rpm) via a lightweight aluminum precision flexible shaft coupling. The AISI 1060 shaft is supported by grease-lubricated clamp-on collar ball bearings with a nominal support distance of 0.7 m (28.5 in) and a shaft diameter of 12.7 mm (0.5 in.). Six 6061 aluminum balance discs are secured to the shaft, each with two keyless bushings. Each disc includes two balancing planes with 70 equally spaced bolt holes along the face of the disc for installing balancing weights. Eddy current proximity probes are in the horizontal and vertical orientations measuring the displacement at each disc. The acquisition system collected data at a sample rate of 16,384 samples/s.

The intent of this test rig was to create distinctly different sensor signatures and evaluate the performance of the deep learning algorithm. Therefore, data were collected at different mode shapes of the machine. When a rotating machine passes

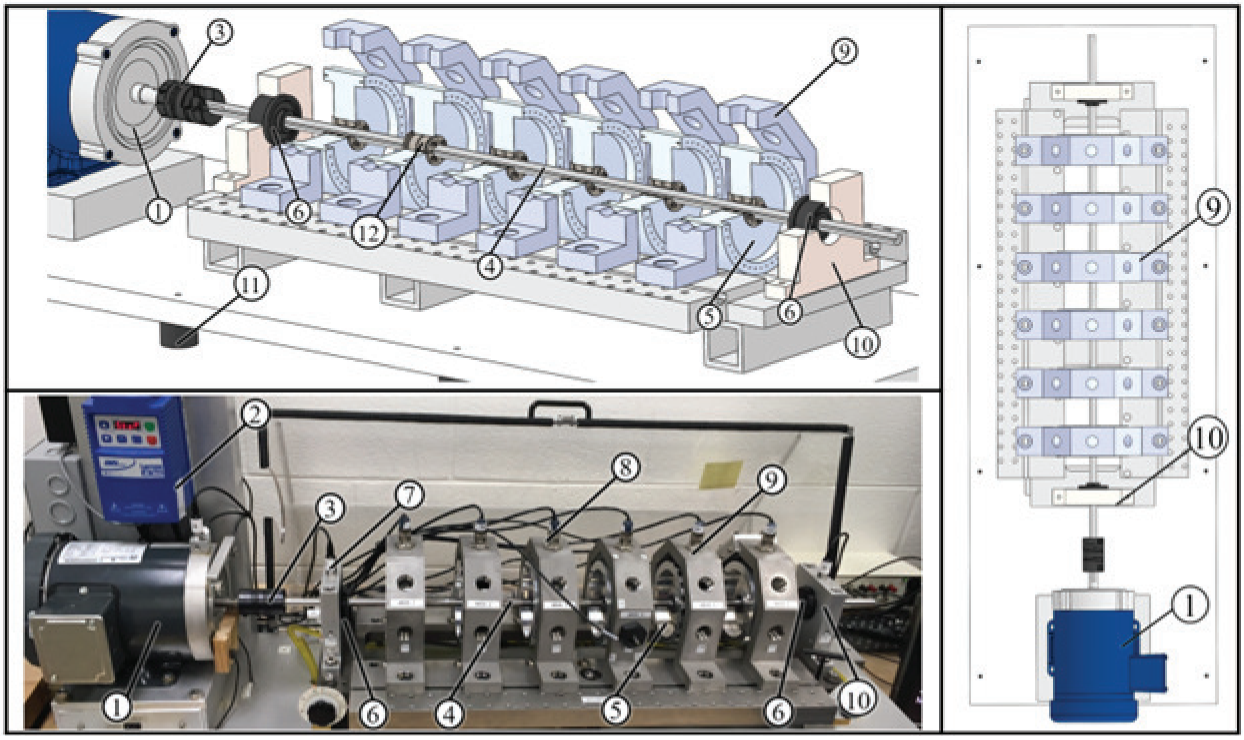


Fig. 2 Mechanical fault simulator (MFS): (1) motor, (2) VFD, (3) coupling, (4) shaft, (5) balancing wheel –qty 6, (6) ball bearing, (7) accelerometer, (8) proximity probe, (9) proximity probe mount (10) bearing pedestal, (11) rubber support, (12) keyless bushing.

through a natural frequency, the rotor operates in a different mode (i.e., the motion along the shaft is different) and thus a different signature will be obtained. For this test rig, the first natural frequency is a cylindrical (i.e., bounce) mode, the second natural frequency is a conical mode, and the third natural frequency is the first bending mode. (See Figure 1. 3 in ref [3], for a visual of the physical mode shapes.)

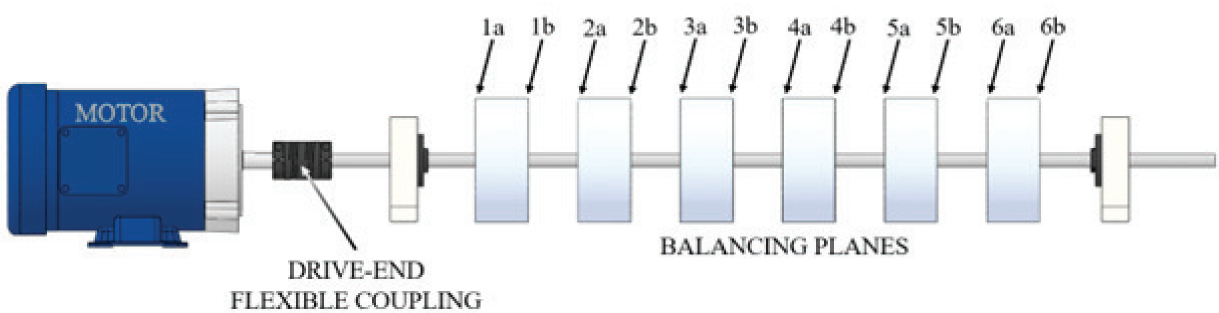


Fig. 3 Cross-sectional view of the rotor assembly, including the balancing planes (1a–6b), coupler, and driver. Each balancing wheel has two balancing planes spaced by approximately 2 in. where 2-56 size bolts were installed to balance the machine.

Additionally, each response signature was affected by the balancing discs, which had the three following configurations: (a) no alterations, (b) a single strip of aluminum tape, simulating a jump in amplitude once per revolution, and (c) 24 engraved lines simulating a drop in signal 24 times per revolution (Figure 4). The MFS provides 16 channels of data, including a tachometer, three accelerometers, and 12 proximity sensors. Three of the proximity sensors were tested at three different speeds, including 523 RPM (8.7 Hz), 1,300 RPM (21.6 Hz), and 4,386 RPM (73.1 Hz). The response described above for each type of balance wheel is shown in Table 1 below.

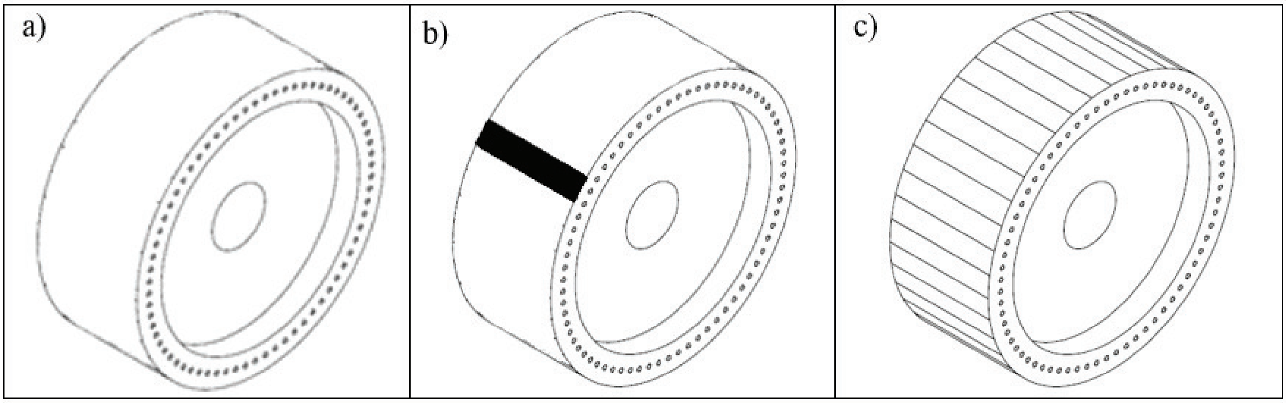
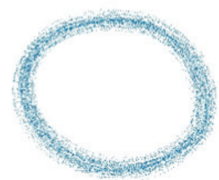
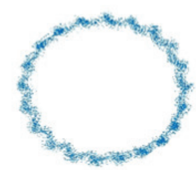
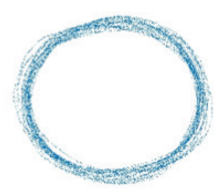
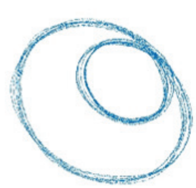
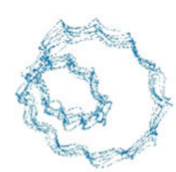


Fig. 4 Each balancing disc was black anodized in three patterns including blocking the signal of the proxy probe: (a) no distortion: third disc; (b) single piece of aluminum tape: fourth disc; (c) 40 inscribed lines: first and sixth disc.

Table 1 Orbit plots for the three different balance disc configurations at each natural frequency.

Natural frequency	No Distortion	Single strip	40 engraved lines
1st			
2nd			
3rd			

Data Preparation Process

Orbit plots were made using 10,000 data points, an image 368 x 598 px, and a plot character size of 15. These plots were reduced to 100 x 100 px PNG files, in gray scale with no axes annotation. One hundred deep learning iterations were used to obtain a comprehensive evaluation of the network training, and each deep learning session used a randomly chosen set of orbit plots for training. Of these orbit plots a few were set aside for testing the neural network. Each iteration was scored for training and testing accuracy. The stochastic gradient descent with momentum (SGDM) algorithm was used in all test cases for the simple convolutional neural network.

Simulated Data

The testing accuracy is defined as the ability of the trained network to recognize data from the same source as training/validation data. Simply put, this is similar to a face identifier but instead of a face, it identifies a simple 1D data sequence from a rotating machine. The unique features of the machine and its characteristics can be used as identification, similar to a fingerprint.

During early testing, it was observed that the training and testing of the neural network did not score above a 95% accuracy rate. It is theorized that systematic common mode data features from the MFS were contained in all the measurements and the training process could not fully differentiate between the channels. Therefore, a signal simulator was developed to produce a secondary set of data specifically for the deep learning process combining a weighted sum of a sinusoid signal and its first three harmonics, random noise, and a glitch generator. The amplitudes of all component signals were randomly set, within constraints, such that the resulting signals were all unique and produced a total of 100 signal combinations and 160 orbit plots for each signal combination.

Results

Three cases were compared to test the accuracy of the neural network to identify the data source. Both methods of 2D mapping were used which included the following cases: (1) using all sixteen channels from the MFS for all three speed conditions, (2) using only the higher-speed case for the MFS and omitting two of the accelerometers and one proximity sensor that provided marginal measurements, and (3) using 100 test cases from the signal simulator. Figure 5 reports the final accuracy results for the three test cases and includes the shifted-data and derivative-diff data mapping methods. The three test cases are the MFS data, MFS reduced data, and simulated data shown in Figure 5.

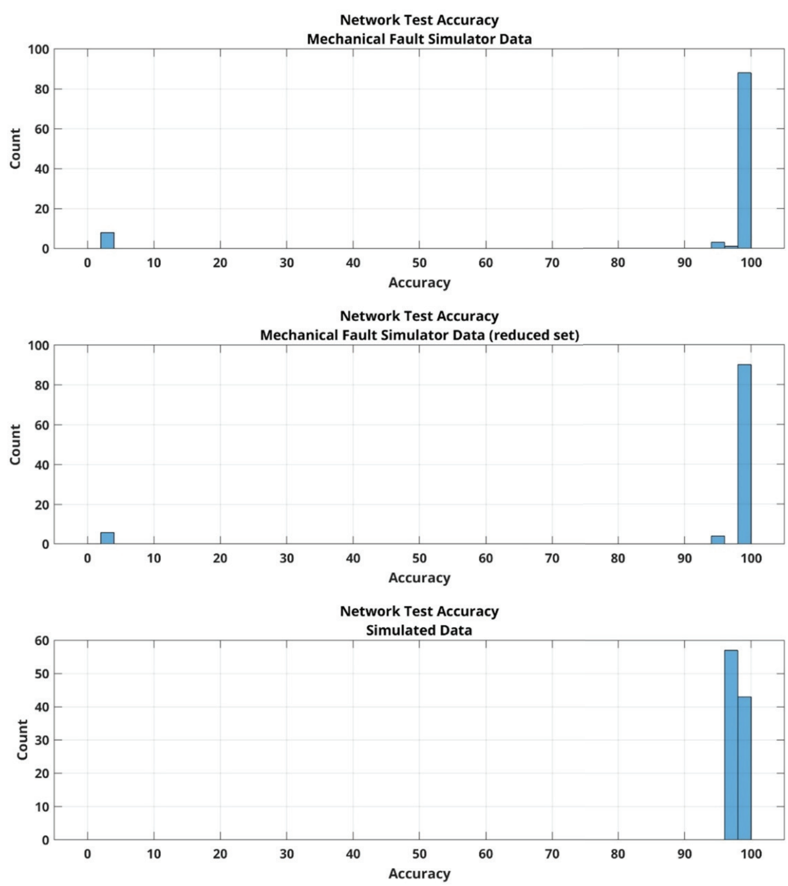


Fig. 5 Trained network accuracy for three test cases.

For the three cases, the plots show that the deep learning model performed accurately with the separately stored test data that was set aside. There are systematic features common to all the sensor channels so the real data performance could have been improved if it had originated from separate machines. Recall that the MFS data is all from the same rotating device. The simulated data on the other hand is independent of a physical device, hence lacks shared common characteristics associated with a physical device In all situations where the testing data shared common characteristics, the training accuracy reported by the deep learning model was poor. The training accuracy can serve as guidance for when not to use a neural network solution. For completeness, the poorer scoring accuracies are included herein but would most likely not be included in a real-world implementation.

The reduced set excludes the proximity probe channels that were malfunctioning. The simulated data was created by a signal generator model, with random noise and glitch features that were applied in such a way as to make the resulting data unique. The relationship in time and magnitude for the signals, noise and glitch characteristics were all randomly set. Because the simulated data was more unique and did not contain common features, it scored with a better accuracy

When testing the neural network against data it has never seen before, the results had high accuracy, mostly above 90%. For the first plot, there were data with an accuracy below 10%. What this means is that most of the proximity sensor data was accurately identified by the network by 90%, but there were around 3 bad channels that scored very poorly (like 10% accuracy success). For the reduced set, the middle plot, the data from the faulty proximity sensors were removed, which improved the middle plot with an accuracy of 95+% for all channels. Then for the last case with the simulated data, the results obtained were similar to the reduced set. All the simulated cases were accurately identified 96+% of the time.

Conclusion

By using a simple 2D mapping scheme for single-channel cyclic data, convolutional neural network trained with SGDM was able to train yield good data recognition results. The testing accuracy was typically above 94% for data produced from multiple sensors on the same rotating machine. The testing accuracy distribution changed but remained similar when several poorly performing channels were removed. For simulated data, the testing accuracy was above 96% for all test cases and conditions, including displacement measurements from the test rig. The neural network could accurately predict the source of data from a given set of sources at an accuracy of 94% or more. It is believed that the data from the MFS had common mode characteristics, or features, on all channels, thereby making it difficult for the deep learning algorithm to fully differentiate between each channel.

Opportunities for practical applications for this approach include machine diagnostics and performance optimization of rotating machines and engines. Two examples of using this method to diagnose machinery include fault conditions and engine tuning. Fault conditions could be characterized for a class of machines and then neural networks could be trained on the data. The resulting trained and validated neural network could then process and compare newly acquired data from other machines to determine if the systematic characteristics match. For engine tuning, data from an optimally tuned engine could be used to train a neural network to then analyze data acquired on other engines and determine whether optimal tuning was achieved.

References

1. Zhu, X., Hou, D., Zhou, P., Han, Z., Yuan, Y., Zhou, W., Yin, Q., "Rotor Fault Diagnosis Using a Convolutional Neural Network with Symmetrized Dot Pattern Images", vol 138, *Measurement*, 526–535, May 2019
2. *Get Started with Image Classification*. mathworks.com. https://www.mathworks.com/help/deeplearning/gs/create-simple-image-classification-network-using-deep-network-designer.html (accessed October 3, 2024)
3. Glick, Z. C., "The Axisymmetric Harmonic Element Including Gyroscopic Effects: A Complete Derivation", *Rose-Hulman Institute of Technology*, August 2019
4. Jeong, H., Park, S., Woo, S., and Lee, S., "Rotating Machinery Diagnostics using Deep Learning on Orbit Plot Images", *Procedia Manufacturing Rotating Machinery Diagnostics using Deep Learning on Orbit Plot Images*, vol 5, 1107–1118, 2016
5. Caponetto, R., Rizzo, F., Russotti, L., and Xibilia, M. G., "Deep Learning Algorithm for Predictive Maintenance of Rotating Machines Through the Analysis of the Orbits Shape of the Rotor Shaft", *Proceedings of the 1st International Conference on Smart Innovation, Ergonomics and Applied Human Factors (SEAHF). SEAHF 2019. Smart Innovation, Systems and Technologies*, vol 150, 245–250, June 2019