Data-Driven Fault Diagnostic Methods for Mechanical Systems Based on Koopman Operator

A THESIS SUBMITTED TO THE FACULTY OF
THE UNIVERSITY OF MINNESOTA BY

Alexandru Robert Nichifor

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE IN MECHANICAL ENGINEERING

August 2021

COPYRIGHT PAGE

ACKNOWLEDGEMENTS

Notably, I would like to thank my advisor Dr. Yongzhi Qu. His encouragement and patience inspired me to persevere through this considerable life goal. With his guidance and cooperation, I was able to achieve more than imagined.

With great appreciation, I would like to thank my defense committee, Dr. Allison Hoxie and Dr. Michael Pluimer. Their support throughout my academic career has been remarkable.

A special thanks to my parents, Florin and Roxana Nichifor, for instilling in me a determination and value of hard work. Their love and support throughout my life made me the person I am today.

Finally, I would like to thank my soon to be wife, Alissa Kirby. Her constant energy and encouragement gave me the confidence to achieve my wildest dreams. Throughout any hardships or times of struggle, she is a beaming light of hope and joy.

ABSTRACT

The Koopman operator is a linear evolution operator that can represent non-linear dynamical systems. Traditionally, there are several methods of analyzing linear dynamical systems. However, when introducing the non-linearity, the analysis of those systems become exponentially more problematic. The method of linear representations of non-linear dynamical systems is becoming more pertinent as data-driven systems are becoming more abundant. The novel use of the Koopman operator in this paper is to extract and classify significant parameters of a non-linear dynamic mechanical system. This thesis proposes two distinct methods of using the Koopman operator for fault diagnosis. The first method proposes a model to extract key features from a dynamic system and feed the features to a neural network to classify the existence of a fault. The second method uses parameters derived from the Koopman operator to create a prediction model with healthy data. This prediction model is then used to predict future system dynamics for a measured time evolution and compare that with direct measurements when future dynamics become available. The two methods are then tested via two separate case studies and the results are discussed.

Table of Contents

List of Tables

List of Figures

1. INTRODUCTION

The world around us can be modeled by dynamical systems including climate change, sociology, ecology, neuroscience, mechanical systems, etc. A dynamical system evolves over some time intervals which in turn can be modeled as either continuous or discrete time. Continuous time systems are what we experience in "real time" having a time interval that is infinitesimally small, while discrete time systems evolve over a particular time step. The principal requirement to begin modeling a dynamical system is asking the question: what is it that we are modeling over time? Typically, the answer is a state space model or a differential equation. This state space can be used to fully describe a dynamical system using state variables. In mechanical systems, possible state variables could be coordinates for position and the velocity between position coordinates. Having both position and velocity of a mechanical system is enough to define a state space which evolves over time. Further, dynamical systems can be categorized as linear or nonlinear systems. Linear systems are based on either additivity or homogeneity principles, where if given a system input, we can accurately assume a proportional system output. Introducing nonlinearity to a system greatly increases the complexity of the given system, causing the dynamics to appear chaotic and subsequently increasing the difficulty of understanding the system. Linear system can be fully characterized by spectral decomposition and other well-known analytical techniques. Mathematically, a continuous time dynamical system is considered in the form of:

$$\frac{d}{dt}\boldsymbol{x}(t) = f(\boldsymbol{x}(t)) \tag{1}$$

where x is the state of the system, t is the system at a time, and f is a function which defines how the system changes over time. The derivative in the above equation implies that the time taken between system changes is infinitesimal. In a discrete case a dynamical system is represented as:

$$\boldsymbol{x}_{t+1} = F(\boldsymbol{x}_t) \tag{2}$$

where x is again the state of the system and F is the function that describes the system changes from one time step to the next. This discrete time case is the more general formulation of describing dynamical systems.

Dynamical systems have been studied for hundreds of years starting with Henri Poincaré and his work on the chaotic motion of celestial bodies [1]. As the modeling of dynamical systems describe the world around us, it has become a focus of modern engineering. Recent efforts put towards the comprehension of modern dynamical systems include the understanding of the full evolution of physical systems, future state prediction, and optimizing system efficiency. Understanding how a system fully evolves over time is perhaps the underlying goal in recent studies of dynamical systems. Being able to parameterize any given system would be crucial to the analysis and solutions of governing equations. Accurately predicting dynamics has also been a goal of many studies in the past. For example, in metrology, future state predictions are being used regularly to predict the weather or natural phenomena [2]. However, much like in metrology, an issue occurs when attempting to predict the trajectories of a dynamical system too far ahead in time. The accuracy of these future state predictions is exponentially decreased over increasing time periods. The demand for system efficiency has long been regulated by the knowledge we have of the system itself. In industrial settings, any machine downtime due to unexpected maintenance or emergency machine faults can be detrimental to a company's productivity and budget. Unforeseen degradation to a machine can be anticipated by Prognostic Health Management (PHM). PHM takes into consideration system diagnostics as well prognostic methods. The diagnostic process involves detecting and isolating faults of a system in real time. Prognostic methods are an extension of diagnosis by attempting to identify the Remaining Useful Life (RUL) of a system. Prognostics use prediction methods based on historical conditions of a system to predict a future state.

Two leading challenges when it comes to achieving the goals mentioned above are nonlinearity and unknown dynamics. A significant impediment when it comes to fully understanding a system is that the majority of systems in real life are non-linear. Once a nonlinearity is introduced into a dynamical system, the system itself becomes extremely complex to analyze. This is due to the fact that when a nonlinearity is present, the system output is no longer directly proportional to the system input. Unlike linear systems, where there is an abundance of analytical methods used to fully describe the dynamics of a system [3], handling nonlinear system is considered a challenge and existing methods are limited. Another obstacle that modern dynamical systems are facing is the unknown parameters of a system one is attempting to understand or model. In many cases, there is a gap of knowledge between the dynamic system in question and known governing equations. Without this knowledge, we are left to look for patterns in these dynamical systems which in some cases lack accuracy.

This paper will introduce a Koopman operator based method for obtaining features of non-linear dynamical systems to be used in machine diagnostics. We will map a non-linear system to a linear system using the Koopman operator, solve for this operator using spectral decomposition, then use the results as parameters for system diagnostics. The rest of the paper is organized as follows: Section 2 will review recent literature on diagnostic methods. Section 3 presents the fundamentals of the Koopman operator and Dynamic Mode Decomposition as well as describing two proposed methods of analysis of dynamical systems. Sections 4 discusses the results from the proposed methods. Finally, section 5 concludes the thesis.

2. LITERATURE REVIEW

2.1 Introduction to Diagnostic Modeling

This literature review will cover various current procedures of diagnostic methods used for Prognostic Health Management (PHM). For the remainder of this review, we will refer to a system as a mechanical hardware system instead of a computer software of a simulated system. The process of diagnosis is initiated by creating an algorithm, or model, that is able to distinguish between a system that is operating routinely and a system that might be experiencing a fault. Furthermore, if the system is not operating at a desired standard, the algorithm can detect the location and type of fault experienced by the system. In recent literature, there are two main categories of diagnosing methods, and both include qualitative and quantitative methodologies. The first category is model based, or physics-based, methods which incorporates Fault Detection and Isolation (FDI) and includes hardware and software redundancies. The second category of diagnostic methods is Data Driven Modeling (DDM), which uses methods from machine learning and statistical analysis. The first step in any diagnostic method, regardless if it is model-based or data-driven, is to understand the system by taking system measurements. Then, using any desired method of feature extraction, a fault diagnosis algorithm can be created to detect and isolate any fault within a system.

2.2 Model Based

Model based diagnosis originated from a subfield of engineering known as dynamics and control engineering. Control engineering is based on control theory, which relies on creating an optimized algorithm which takes a combination of system inputs, to obtain a desired system output. The ability to create a model for diagnostics is heavily reliant on the physical parameters of a system. Creating this model can either look at separate components or the entire system. The FDI method of model-based diagnosis falls in the field of control engineering and evaluates the physical system itself. FDI involves hardware and information redundancies, where a parameter that is redundant to how the system operates is introduced and analyzed. The benefit of introducing a redundant aspect to a system is to improve the safety and reliability of the system itself. By having a duplicate component, for example a sensor, when the original component is to fail, the duplicate can continue normal system operations and provide information of a where a fault possibly occurred. Information redundancy is used to analyze a code created to detect residual errors. This

is done by creating an algorithm that models an observable healthy system to its parallel simulated system. The higher the residual error is between the observed system to the healthy simulated system, the higher the chance of a fault existing in the system. The residuals extracted here can be considered as features of the system used to create the diagnostic model. Approaching the analytical redundancies of model-based diagnostics can be further broken down into two main stages: residual generation and residual evaluation.

2.2.1 Residual Generation

The first stage of FDI analysis is to generate a fault threshold or indicator. This can be done by using the inputs and outputs of the observed system. Some methods previously used for residual generation include parity, observers, and parameter estimation. The residual between the true system output and the modeled output, if the system is healthy, will be statistically zero. However, if a number above zero is present, then a fault might have occurred. For this method to be accurately considered, the residuals generated must be independent of the system state of operation. Figure 1 below shows a general diagram of how residuals are generated.

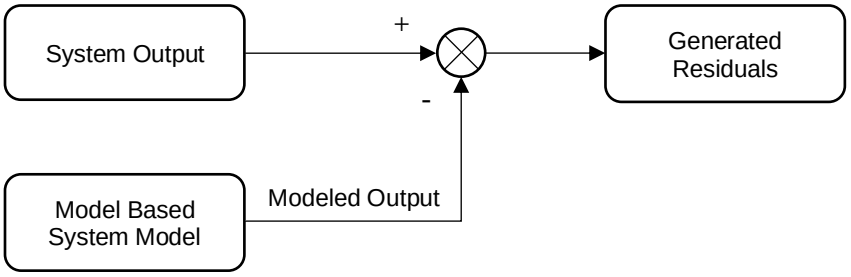

Figure 1: Diagram of residual generation

By comparing the system outputs with the outputs from the created model, residuals are created by a number of different methods. Parity or constancy relations is a method of generating the necessary residuals. This can be done using mathematical equations to create a simulated model which when compared to the original model should return values close to zero. This method was tested on a DC motor to test predetermined controlled faults and the methods ability to accurately detect and isolate them [4]. A parity model was also created for fault detection of an actuator where the relationship between the system inputs and outputs were modeled to optimize a transformation matrix [5]. Observer methods are based on estimated algorithms instead of mathematical equations.

A common observer algorithm is known as the Kalman filter. This algorithm in line with several others, are able to estimate possible outputs of a system depending on the system inputs. A recent method of observer based residual generation was designed for linear differential algebraic equations (DAE) where a numerical example is examined of a robot manipulator [6]. Parameter estimation is yet another method that is able to generate residuals. This method compares the physical components of a system as parameters for the simulated model. Parameter estimation has been used for continuous time models by theoretically modeling the desired parameters to physical state-equations or state dependencies such as resistance quantities [7]. In general, the knowledge of exact system parameters is not easily acquired, therefore this method is often used in industry to obtain an estimate of the parameters. A review was conducted of methods used for residual generation and the most common are observer-based modeling and parameter estimation in [8].

2.2.2 Residual Evaluation

Once the residuals are generated, the second stage of FDI analysis is evaluation of the established residuals. In general, this stage is used to decide whether or not a fault has occurred by assessing the residual threshold. This stage can be further broken down into three main steps: *residual processing*, *threshold identification*, and *decision logic*. In ideal cases, the residual calculated is zero indicating that there is no fault in the system. For any other value, this stage of residual evaluation is able decide the presence of a fault. Figure 2 below shows the steps taken to detect a fault.

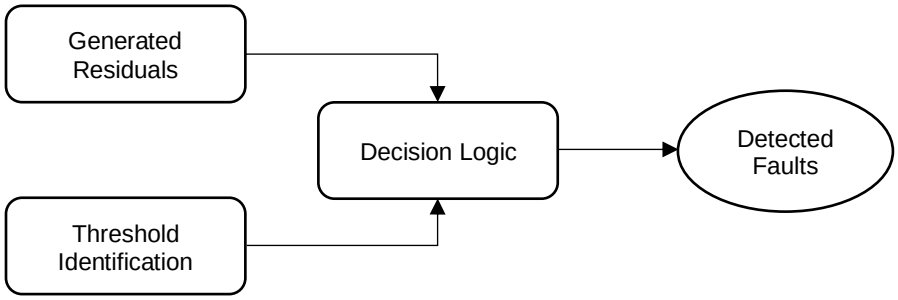

Figure 2: Diagram of residual evaulation to detect a fault.

After processing the residuals and identifying a threshold for evaluation, both values are then input into a method of decision logic to detect a fault. Either norm-based methods or statistical methods are used to process the generated residuals depending on whether the system is

respectively deterministic or stochastic [9]. For deterministic systems, the most common norm-based method used is L_2-norm. The L_2-norm method lives in a Euclidean space and measures the length of a vector by calculated the square root of the sum of squares of two coordinates. This method was used in [10] for fault diagnosis for open-circuit faults of induction motor drives. When dealing with a stochastic system, statistical methods are preferred such as root-mean-square (RMS) tests, regression, or hypothesis testing. Most commonly, regression analysis methods are used when analyzing stochastic systems [11]. The second step of residual evaluation and arguably the most important step is threshold identification. Regardless of the type of system that is being diagnosed, choosing an optimized threshold is important in the accuracy of fault detection. If a low threshold is selected, some residuals can be misdiagnosed as faults and generate "false alarms". On the other hand, if a high threshold is selected, some small faults could go undetected and cause "missing alarms". A threshold, or limit monitoring, involves a minimum and maximum value that is not be exceeded. When identifying a threshold, certain aspects of the system must be taken into consideration such as the bounds of measured system inputs or outputs, the dynamics of the residual generator, and the uncertainties of the created system model [12]. The final step is the logic used to make the decision of the existence of a fault. The simplest and most common method of decision logic is to simply compare the residuals generated to the threshold chosen. The output of the logic is typically binary, i.e., a fault exists, or no fault has been detected. The overarching goal of residual evaluation and the previously talked about methods is to compare the information imported from the output residuals and further conclude whether or not a fault exists.

2.2.3 Summary of Reviewed Model Based Methods

Overall, the model-based method of fault diagnosis is heavily based on the system inputs and outputs. The most common model-based method falls into the category of Fault Detection and Isolation (FDI). Creating a model using hardware redundancies is often much simpler than the analyzing information redundancies. However, the information needed to build a model using hardware redundancies is not always readily available. Analyzing information-based redundancies is broken down into two main stages: the generation of residuals and the evaluation of residuals. In general, combining the two stages with an analytical model, system faults are detected. A summary of methods discussed for residual generation and evaluation is found below in Table 1.

Table 1: Summary of residual generation and residual evaluation methods

Residual Generation	Residual Evaluation
Parity Equations	Processing Residuals
Observer Based	Threshold Identification
Parameter Estimation	Decision Logic

Table 1 shows the several methods of generation and evaluation of residuals that were covered, all concluding with detecting system faults. The most common methods of residual generation are observer based and parameter estimation methods. Likewise, the steps of residual evaluation are largely identified as processing the generated residuals, identifying a threshold, and a decision tool used to determine if a fault is present. An overview of model-based system diagnosis is shown in Figure 3 below.

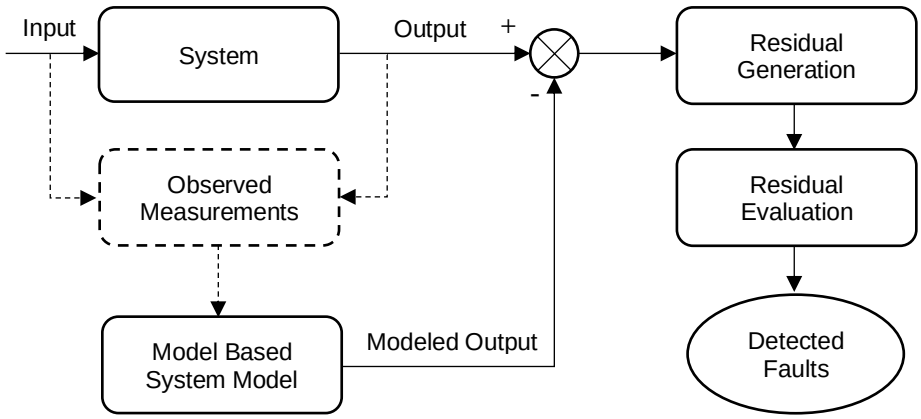

Figure 3: Model Based Diagnostic Fault Detection

Figure 3 shows an analytical approach to the model-based method of fault diagnosis. First, the knowledge of the system inputs and outputs are required to create a model-based system model using observed measurements. The outputs generated from the model are compared with original system outputs to obtain their residuals. Several methods of generation and evaluation of residuals were covered all concluding with detecting system faults.

2.3 Data-Driven Modeling

With the abundance of data in the modern technological world also comes new methods of modeling different systems. Data-driven modeling has become a new field of interest across many disciplines in science and engineering, such as industrial processing, the gas and oil industry, fluid dynamics, and even bioengineering [13]. Rather than having prior knowledge of a system, as is required for model-based diagnosis, data-based diagnosis requires current and historical system data to be used as system measurements. Data-driven modeling typically relies on either machine learning algorithms or statistical methods of analysis. Machine learning methods includes the application of Neural Networks (NN), Decision Trees, Support Vector Machines (SVM), etc. Conversely, statistical methods comprise of, but not limited to, Principal Component Analysis (PCA) and Bayesian Networks (BN). The purpose of data-driven modeling in diagnostic purposes are identical to that of the model-based modeling. The overarching goal is to detect a fault in a system. However, the data-driven method requires little knowledge of the system itself and relies solely on the data taken from the system. To accurately diagnose a system with a data-driven method, there is a requirement of both healthy state data and faulty data. This fault can either be naturally occurring or be in a controlled environment. The following sections will introduce two subcategories of data-driven modeling: Machine learning and statistical methods.

2.3.1 Machine Learning

A dominant field of computer science in recent literature has been the topic of machine learning. By developing some sets of rules, an output can be estimated from any given input. Many advancements have been made to create algorithms that can estimate such outputs. A common method is known as a neural network, which is a set of algorithms that is able to identify relationships between data. Data-driven neural networks can be labeled as supervised and unsupervised learning. Supervised neural networks require a knowledge of both an input and a desired output, while unsupervised learning is used when only knowledge of input variables exists [14]. Once the data is collected, it is broken down into a majority training data and a minority testing data. Training data is used to fit the system parameters by finding connections and patterns that accurately create a neural network model. The remainder of the data is used to test this neural network to then classify existing parameters. The overall goal of neural networks is to map every system input to a desired output as accurately as possible. In recent literature, an Artificial Neural Network (ANN) was used to diagnose faults of winding a generator [15]. This study used two

ANNs to first detect a fault occurrence and then to classify the faults. Another method of creating a model is by Support Vector Machines. SVM is a strong classification tool when dealing with two dimensions. If the given system is in a high dimensional state, the data can be projected or mapped to a lower dimension using various kernel functions. Once training data is determined in two dimensions, SVM is able to characterize system data into two classification groups. Such a mapping was done to create a multi-parameter manufacturing quality prediction model [16]. A combination of ANN and SVM can be used for a data-driven modeling method where ANN is used to detect faults while SVM is used to isolate or classify these faults. Regardless of the different methods of training the acquired data, a general algorithm of data-driven modeling using machine learning is shown below in Figure 4.

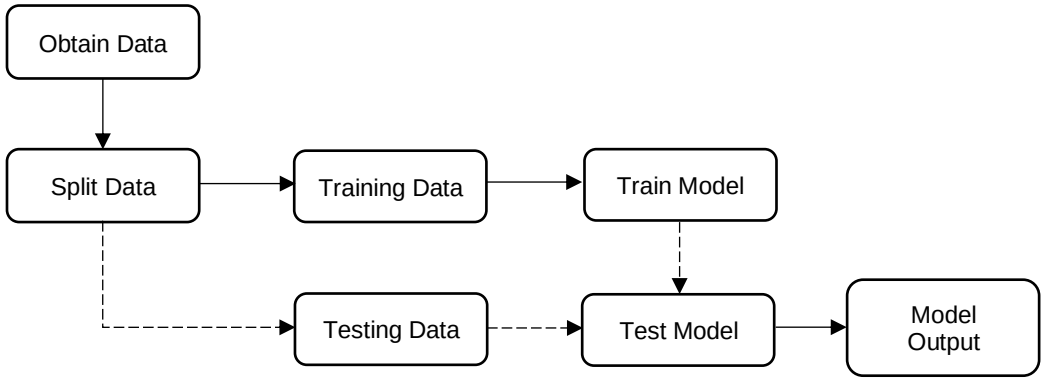

Figure 4: Basic flow of machine learning algorithms.

Once the data is obtained it is split into mostly training data which is in turn used to train a model. Afterwards, testing data is used test the model's accuracy. If the model provides expected outputs, the model is considered accurate. However, if the outputs are not as expected, the method returns to the beginning and trains the model using different algorithms or patterns.

2.3.2 Statistical Analysis

Other data-driven methods of building a model have been derived from existing statistical techniques. A largely used statistical analysis tool has been Principle Component Analysis (PCA). This method has been widely used for linear systems of high dimensions. PCA methods perform analysis on multivariate systems and is able to reduce dimensionality by mapping high dimensional data to a low dimensional space. Once this data is in the low dimensional space, it is then classified

as principal components of the preliminary large data set [17]. A setback of using PCA analysis is that it solely works for linear systems, and most systems are inherently non-linear. However, methods have been recently developed to extend PCA to nonlinear systems by combing the dimensionality reduction with an auto-associative neural network [18]. Another statistical method of creating a data driven model is using a Bayesian Network (BN). A BN uses probabilistic methods to create a model of a represented system via a directed acyclic graph [19]. Every input variable is considered a node which is later assigned some probabilistic value based on its preceding nodes [17]. In recent literature, a Bayesian Network has been used to incorporate a probabilistic boundary for fault detection models [20]. The statical methods of data-driven modeling incorporates existing statistical methods to be able to either create a model or diagnose a fault in that model.

2.3.3 Summary of Data-Driven Modeling

Recent research efforts for data-driven modeling have grown with the increasing ease of data collection. The simplicity of implementing sensors in a system and simply collected data is an overarching advantage of creating a model with the data-driven approach. Some methods reviewed included neural networks, SVM, PCA and Bayesian Networks to create a model. Once the model is created and the output value predicted, similar steps to residual evaluation are taken to determine the existence of a fault. By comparing a modeled state to the healthy system state, these methods can detect faults by isolating parameters that fall out a given range.

2.4 Hybrid Modeling

A third, less common yet evolving category is a combination of the preceding methods. For optimization purposes, model-based methods or FDI have been used for the evaluation of residuals, while aspects from data-driven modeling are used to create a model and generate residuals. For example, a combined diagnostic method has been used in the automotive industry, specifically on antilock braking systems [21]. FDI was used to create a physical model of the vehicle's parameters, and then SVM is used on the data to discover any parameters which could not be found using FDI. Another example of a hybrid modeling was found in [22] where a different combination of SVM and FDI was used to diagnose and detect faults of an HVAC system. The author created a simulated model and evaluated the residual values between the healthy and simulated models using FDI techniques and later classified the faults using SVM. PCA was used

in combination with Artificial Neural Networks (ANN) to both create a model and then classify any faults based on optimized threshold values from the neural network [23]. In this example, the author created a model, optimized a threshold, and performed fault detection and diagnosis on HVAC systems by combining known methods. Hybrid models for diagnosis typically use different methods based on their ideal advantages in separate portions of the process. This creates an optimized method of diagnosis that is customizable to the need of the specific system.

2.5 Summary of Diagnostic Methods

A number of different methods for both modeling a system and diagnosing a fault have been reviewed in the previous section. The variety of methods was broken down into three categories: model based, data-driven, and hybrid methods. Model based methods of diagnosis are heavily reliant on a deep understanding of the system itself and tend to be limited to the complexity of the system. These methods include a general outline of a two-stage process: residual generation and residual evaluation. The residual generation portion of the process covered methods such as parity equations, observed based methods, and parameter estimation, while the residual evaluation portion covered processing residuals, threshold identification and decision logic. Data-driven modeling is based on the existence of sensors on a system that is able to produce constant data. This section was divided into two categories: machine learning methods and statistical methods. Machine learning methods included neural networks and SVM algorithms while the statistical methods covered PCA and Bayesian Networks. This review covered only common methodologies and ideas of both models based or data-driven methods. There is an abundance of methods that were not covered by this literature review. A more comprehensive review of both model-based and data driven methods of fault diagnosis is reviewed by [24].

In this paper, a Koopman operator based diagnostics is proposed which aims to build a data driven model from dynamic data to model and diagnose a dynamical system. Dynamic Mode Decomposition is used to identify parameters used to create a representative model that reflects the underlying physics using spectral decomposition of the Koopman Operator. A detailed methodology is presented in the following section.

3. METHODOLOGY

This section will be split into two parts. The first part 3.1 will cover the theory behind the methodology proposed in this thesis and includes subsections on the Koopman operator and Dynamic Mode Decomposition. The second part 3.2 will cover two distinct proposed approaches and contains subsections of both methods used for fault diagnostics of dynamical systems.

3.1 Background of Methodology

As covered in the literature review in the previous section, there are a number of methods for performing fault diagnosis. Two new approaches are proposed based on features extracted from the Koopman operator and Dynamic Mode Decomposition (DMD) is used as a solver. The modeling process is inertly data-driven, while the latter portion of methods use residual generation and evaluation.

3.1.1 The Koopman Operator

Dynamical systems are typically non-linear systems which are, in practice, difficult to analyze. The Koopman operator is a linear operator that can help construct a linear relationship for the "temporal" dynamics of a non-linear system [25]. This is done based on the assumptions that if we "lift" the states of the system to a high dimensional observable space with all non-linear terms captured, it is possible to express the original non-linear dynamics with a linear model. Therefore, using the Koopman operator for dynamic system analysis, one can get a linear representation of the system and extract key features which further enables the prediction and control of the original non-linear system.

Traditionally, discrete time dynamics are found in the state space and are modeled as $x_{n+1} = F(x_n)$, where F is a time evolution operator that brings the system from one time to the next [25]. Figure 5 is an illustration of the "lifting" of the dynamics from the state space to the observable space. In this new space, the states x_k are represented by functions of the state, or observables, $g(x_k)$. For a discrete time-based system, the dynamics can be illustrated as $g(x_{k+1}) = g \circ F(x_k) = Kg(x_k)$, where K is the Koopman operator which evolves the observables forward in time. The number of states depicted by "n" are typically much lesser than the number

of observables depicted by “k”. In the state space, the dynamics are non-linear and finite dimensional. Conversely in the observable space, the dynamics are now linear but augmented to higher dimensions.

Once the Koopman operator is obtained, spectral decomposition is performed to extract its eigenvalues and eigenvectors. Then, the Koopman modes and mode amplitudes are defined, which in turn can be used to reconstruct the original system states. In this paper, the modes and mode amplitudes can then be considered as features of the original state.

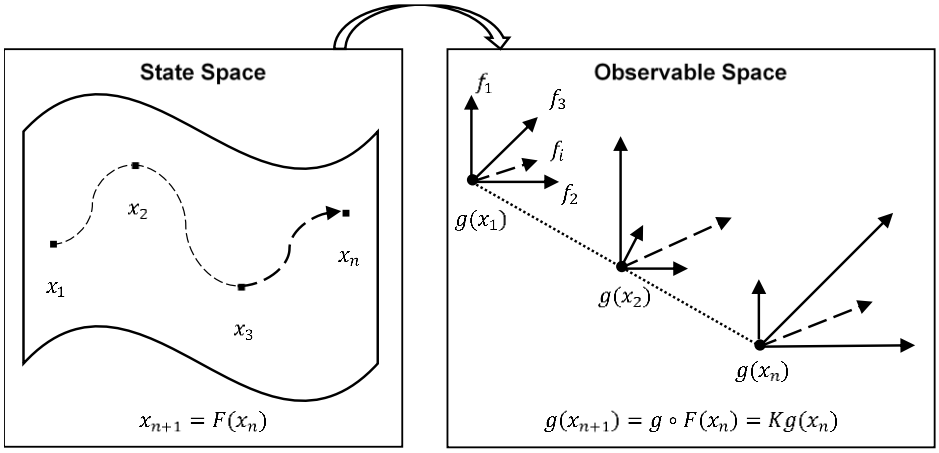

Figure 5: An illustraion of how the Koopman operator behaves in an observable space.

3.1.2 Dynamic Mode Decomposition

The purpose of dynamic mode decomposition (DMD) [26] in this paper is to extract the Koopman modes and mode amplitudes. DMD utilizes singular value decomposition (SVD) so that dimensionality reduction can be applied to a high dimensional system. DMD at its core is a data-driven linear regression problem. Consider a data set of vectors $\{x_1, x_2, \dots, x_n\}$ where every vector $x_i \in \mathbb{R}^n$ and each vector is an evolution forward in time. Each vector x_i is referred to as a snapshot in time. A collection of such snapshots forms a high dimensional matrix $\boldsymbol{X} = [x_1, x_2, \dots, x_n] \in \mathbb{R}^{m\,x\,n}$. ($m$ denotes the number of states and m denotes the number of states.) For a discrete-time dynamical system, the data can be modeled with an unknown linear operator *A*, in the form of $Ax_i = x_{i+1}$. This operator A is a linear approximation of the dynamics of a nonlinear dynamical system [27]. The following algorithm is summarized based on [27]. From this model, a high

dimensional future state matrix can be defined as $\boldsymbol{Y} = [y_1, y_2, \dots, y_n]$, where $y_i = x_{i+1}$. A set of snapshot pairs of these high dimensional matrices is defined as $\{(x_i, y_i)\}_{i=1}^{n}$. The linear operator *A* represents the relationship between the two high dimensional snapshot matrices:

$$Y \approx AX. \quad (3)$$

DMD utilizes dimensionality reduction via SVD to approximate the best-fit linear operator A. The following steps will show how the DMD algorithm extracts the operator *A* and extracts its modes and mode amplitudes.

STEP 1: Once the data is separated into the snapshot pairs compute the SVD of X [28]:

$$X = U\Sigma V^* \quad (4)$$

where $U \in \mathbb{C}^{n\,x\,r}$, $\Sigma \in \mathbb{R}^{r\,x\,r}$ is a diagonal matrix, $V \in \mathbb{C}^{m\,x\,r}$ and r is low rank approximation of X.

STEP 2: A low rank approximation of A is defined as $\tilde{A}$:

$$\tilde{A} \triangleq U^* \mathrm{Y} \mathrm{V} \Sigma^{-1} \quad (5)$$

STEP 3: Compute spectral decomposition of $\tilde{A}$:

$$\tilde{A}w = \lambda w \quad (6)$$

where w is a matrix containing the eigenvectors of $\tilde{A}$ and λ is a diagonal matrix containing the eigenvalues of $\tilde{A}$.

STEP 4: Using the eigenvectors, w, DMD modes can be reconstructed to describe the high dimensional operator A.

$$\varphi = YV\Sigma^{-1}w \quad (7)$$

STEP 5: Compute the mode amplitudes:

$$b = (w\lambda)^{-1}x_1 \quad (8)$$

The eigenvalues and modes of the dynamical system shown in equation (3) are advantageous in describing the system dynamics. DMD modes are vectors that are spatial representations of the dynamics and the eigenvalues respective to their modes are growth or decay rates. Mode amplitudes define the significance that each mode has on the system. Figure 6 below summarizes the dynamic mode decomposition algorithm presented in this section.

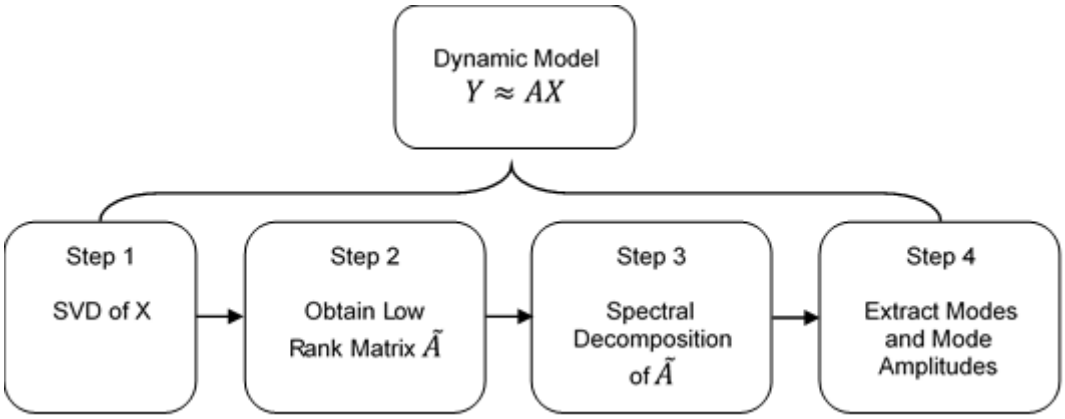

Figure 6: Dynamic Mode Decomposition (DMD)

Dynamic mode decomposition can be used as a solver for the Koopman operator and the Koopman modes and mode amplitudes. Once these features are extracted, they are fit to a neural network used to extract key features from the original system dynamics [29].

The Koopman operator requires a rich enough data set. In practice, data is often limited by the number of sensors, or states, of many dynamical systems. Although limited, the data still contains essential information to describe the non-linear dynamics. However, the data resides in a low dimensional space. In order to unfold the complex information from the original space, one can augment the data using a "dictionary" [30]. This dictionary can be considered as a group of functions that map the original data to coordinates in a new space. This method is also known as a kernel in machine learning. Various methods of augmenting the state space include a Gaussian

kernel, Hermitian polynomials, and radial basis functions or RBFs. The case studies later used in this paper will use an RBF kernel to create a dictionary of approximated functions of the state space based on the distance between fixed data points and the centers of clusters found from a k-means algorithm. Specifically, a thin-plate spline radial basis function kernel is used for data interpolation and smoothing, which grants the solution of the Koopman operator a more accurate approximation.

3.2 The Proposed Approaches

Here, an overview of two proposed methods for analysis of dynamic systems using the Koopman Operator are presented. Both methods can be classified as fault detection tools. The first uses a neural network to build and test a model while the second uses statistical analysis methods to test a prediction model. The two methods begin by sharing the same feature extraction process. This extraction process is initialized by collecting raw sensor data from a non-linear dynamical system. This data consists of multiple sensors or state variables in the state space. To be able to use the Koopman operator, the data must be mapped to an observable space before any type of analysis. This mapping process can augment the data to a higher dimensional space using a kernel function. The data is then rearranged to form snapshot pairs, X and Y, which represent the current system state and a future system state respectively. These snapshot pairs are then used to create the Koopman operator model.

$$Y = K(X) \tag{9}$$

Next, either traditional DMD or extended DMD, based on the mapping method used from the state space to the observable space, is performed to solve for the Koopman operator as well it's spectral properties. From the extracted Koopman eigenvalues and eigenvectors, Koopman modes and mode amplitudes are defined as key features of the original non-linear dynamical system. This process of feature extraction is illustrated below in Figure 7. From this point, the two methods differ and are described more thoroughly in the following subsections. The overall goal of both methods is to create an accurate model from healthy data, test it against experimentally or simulated faulty data, and be able to determine if a fault is present.

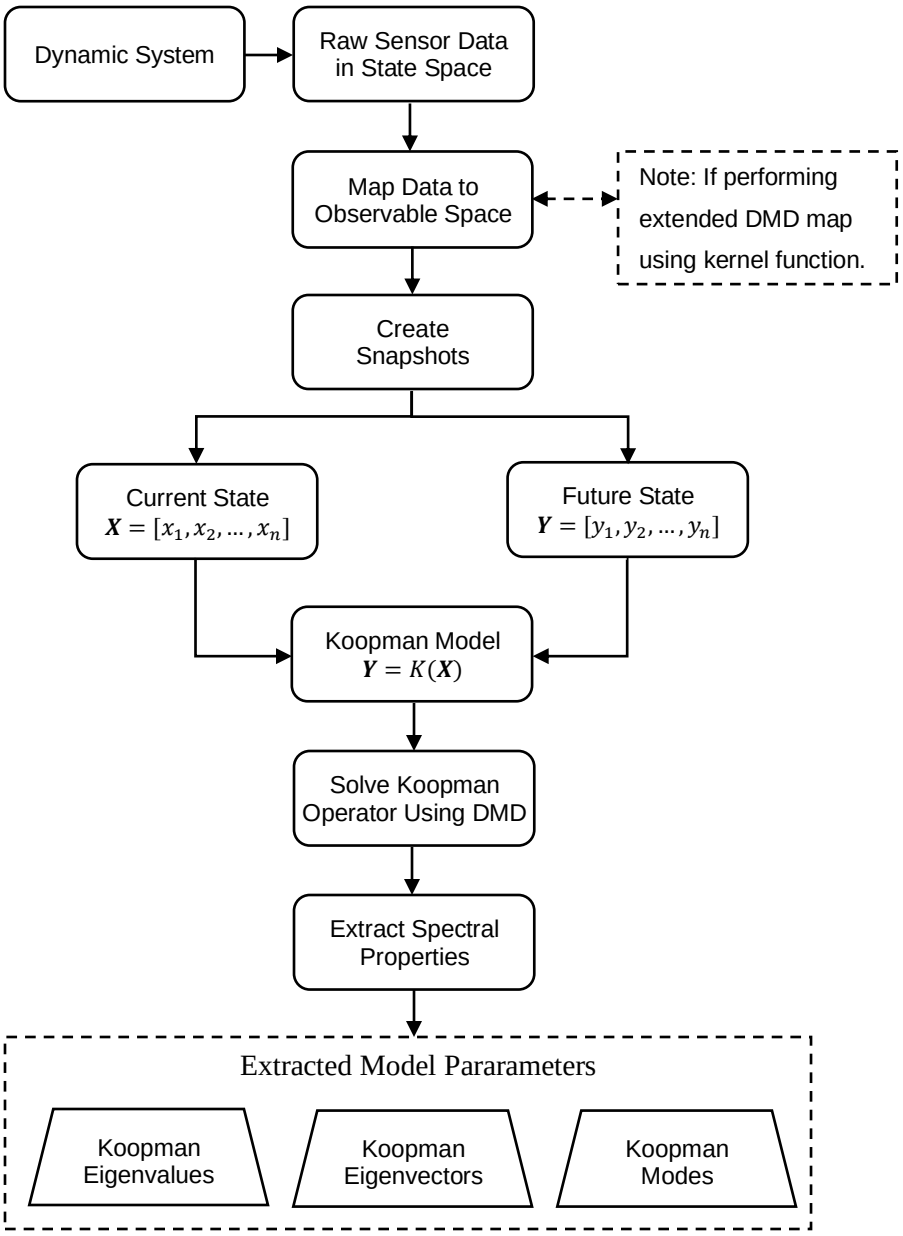

Figure 7: The common feature extraction process shared by both proposed methods

Once the model parameters are extracted, the two methods diverge from their similarities. The first method uses only the extracted mode amplitudes while the second uses the Koopman eigenvalues and the Koopman modes. The following methods are mainly data driven meaning the initial data used comes from a healthy state and is compared to data that has experienced some type of fault. These methods both rely on the assumption that both healthy and faulty system states exit and there is an abundance of data for each.

3.2.1 Method 1: Mode Amplitude

The first method of analysis feeds the mode amplitudes obtained from DMD into a neural network model and the results are used to determine the accuracy of fault detection. Once the mode amplitudes are extracted, they are treated as features of the dynamical system and are imported into a simple neural network. An overview of this method is presented below in Figure 8.

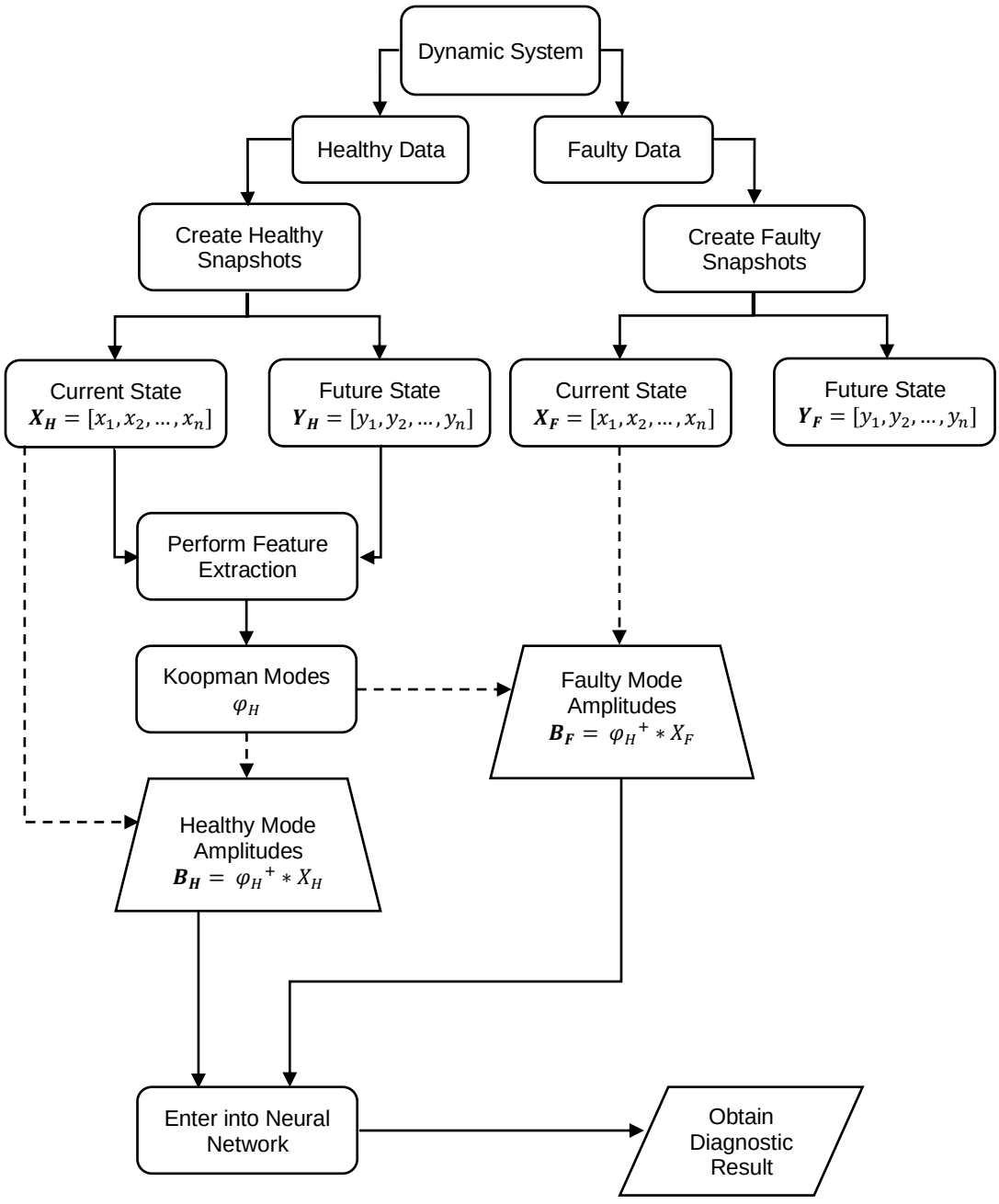

Figure 8: Illustration of proposed method for fault diagnosis using Koopman mode amplitudes

The Koopman mode amplitude approach for fault diagnosis begins identical to the feature extraction methodology. First, healthy data is mapped to an observable space and healthy current state and future state snapshots are created $[X_H, Y_H]$, where the subscript, H, denotes a healthy system state. Next, the feature extraction process begins by creating a Koopman operator model, $Y_H = K * X_H$. DMD is then used to solve for the Koopman operator and in turn performs spectral decomposition of the operator to extract the Koopman modes. Once the Koopman modes are extracted, the healthy mode amplitudes are found by

$$\boldsymbol{B_H} = {\varphi_H}^+ * X_H \quad (10)$$

where φ_H denotes the healthy Koopman Modes, and B_H denotes the healthy mode amplitudes. After the desired healthy mode amplitudes are extracted, the faulty data is similarly mapped to an observable space and faulty current state and future state snapshots are created $[X_F, Y_F]$, where the subscript, F, denotes a faulty system state. Next, the faulty mode amplitudes are found by

$$\boldsymbol{B_F} = {\varphi_H}^+ * X_F \quad (11)$$

where φ_H again denotes the healthy Koopman modes, and B_F denotes the faulty mode amplitudes. The independent variable in this new model is the current state snapshot matrix of either a healthy system state or a faulty system state.

After both healthy and faulty mode amplitudes are extracted, they are fed into a neural network. The joined mode amplitude data is split into a majority training and a minority testing data. The majority is then used to train a classifier model and then the remainder of the data is used to test its accuracy.

3.2.2 Method 2: Prediction Model

The second method of analysis is building a prediction model via regression, which creates a dynamic function over time. This method uses the modes extracted from DMD as well as the Koopman eigenvalues to create a prediction model. The model outputs predicted future states of a dynamical system. Once the desired outputs are obtained, the model is tested for accuracy by known statistical methods. An overview of this methods is presented in Figure 9 below.

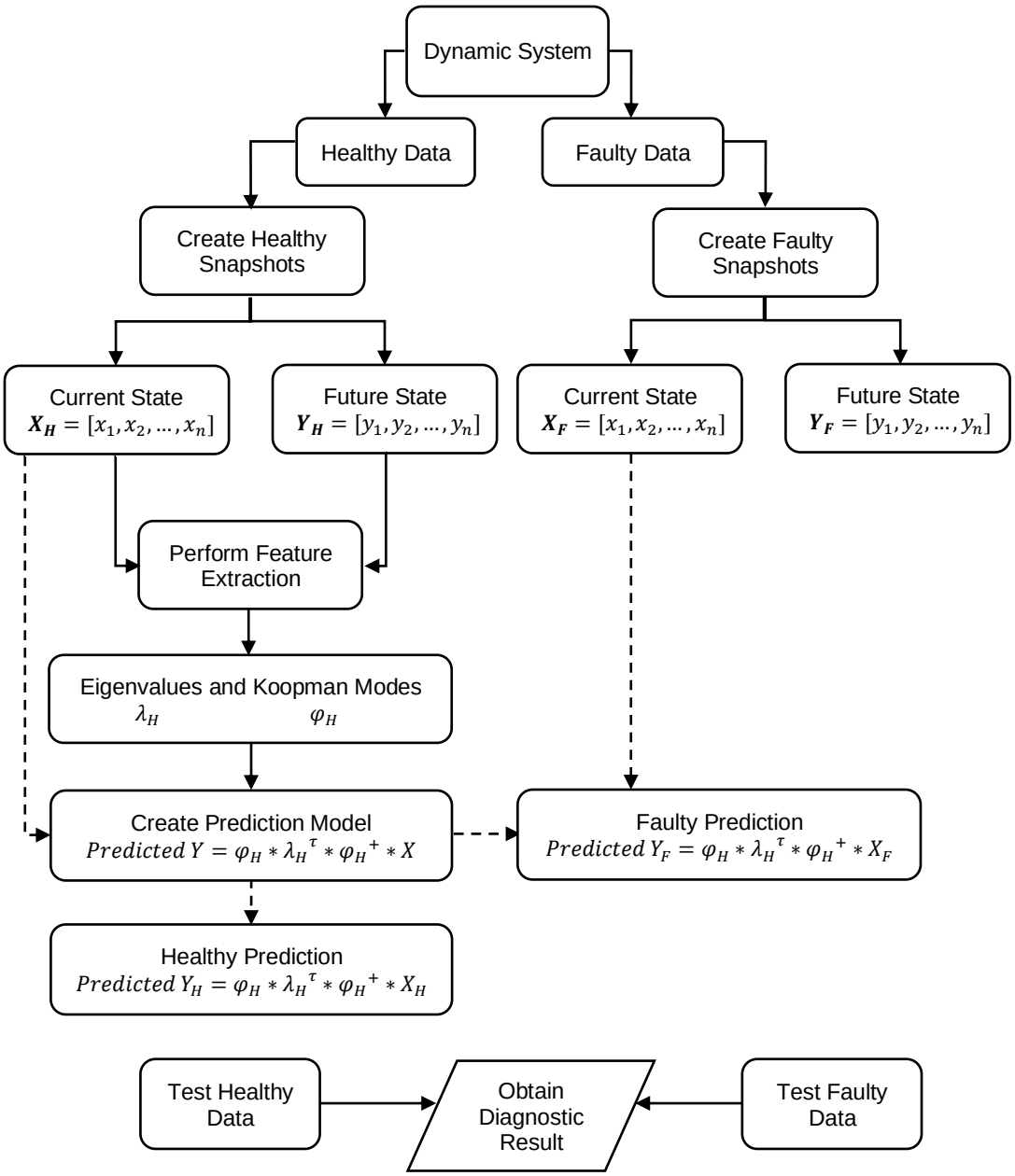

Figure 9: Illustration of the proposed method for fault diagnosis by creating a prediction model

The proposed prediction model approach is based on Koopman eigenvalues and Koopman modes. This method of fault diagnosis begins identically the previous method and the feature extraction methodology. The process is initialized by creating current state and future state snapshots from the healthy data, $[X_H, Y_H]$. Next, the healthy Koopman operator model is created and solved via DMD. The spectral decomposition portion of DMD extracted the Koopman eigenvalues and the Koopman modes to be used as features for the prediction model. This model

can be used to recreate the original dynamics to some time-step forward. The equation for this model is written as:

$$(y_\tau) = \varphi \lambda^\tau \varphi^+ x_k \quad (12)$$

where x_k is the original snapshot data, y_τ is the predicted future state data, φ and λ are the Koopman eigenmodes and eigenvalues respectively and finally τ indicates the time-step chosen for prediction. Any snapshot, y_k, can be inserted into the model above to produce a prediction τ steps away from the original data. As seen in Figure [9] above, when the current state of the faulty data set is imported in Equation (12) the result is the predicted future state of the faulty data set. The output, y_τ, from the prediction model and the original data can be compared using any valid known statistical methods to test the accuracy of the model. This prediction model, acts as a dynamic function over time. It describes how the system will evolve over time. Much like how time is the independent variable in the derivative model, τ, is the only independent variable in this prediction model. By altering τ in the above model, the prediction can be expanded over a desired number of time steps. The goal of this method is not only to detect a possible fault in the system, but also to predict the possible occurrence of a fault before it occurs.

The previously proposed methods aim to detect and diagnose faults of dynamical systems. Both methods share the same feature extraction process and differ only in methods of residual generation and extraction. The following section will apply both of the proposed methods to two separate case studies and present the results of using these methods.

4. CASE STUDIES & RESULTS

4.1 Experimental Setup

The dynamical system used in the following two case studies was a spur gearbox containing one set of gears. The experimental gearbox was installed in an electronically closed transmission test rig as shown in Figure 10. On either side of the gearbox, there exists two 45 kW Siemens servo motors. In this set up, motor 1 acts as the driving motor with flexible configuration modes for variable torque and speed. The flexibility of the system allows for motor 2 to be a load motor acting as a generator.

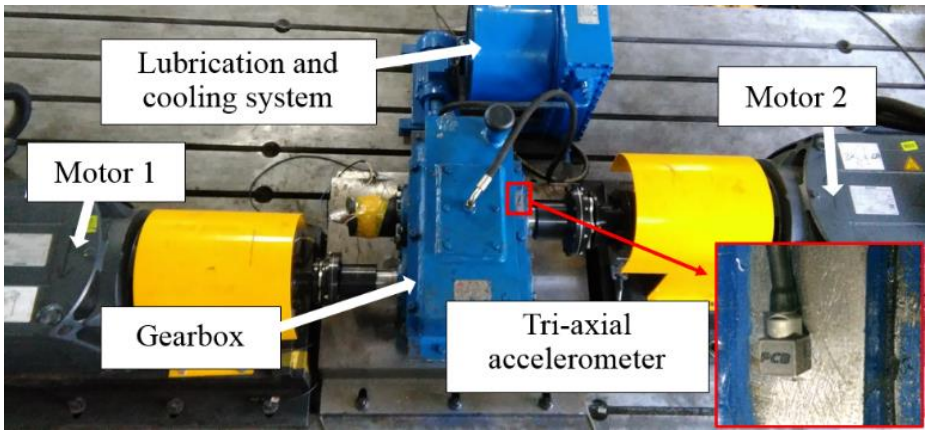

Figure 10: Experimental test rig for both case studies.

The data was collected under multiple speeds and torque combinations. The speeds ranged from 1000 RPM to 3600 RPM with increasing intervals. At each interval, the torque was varied from 50 Newtons to 500 Newtons. In both case studies, data under 1000 RPM and 200 N of torque are used. Vibration data is collected using a triaxial accelerometer at a sampling rate of 20,480 Hz. The data collected, or the states of the system, is of the x, y, and z directions respectively defined as the horizontal, axial, and vertical directions. The motion control of both motors and the Data Acquisition System (DAQ) is illustrated in Figure 11 below.

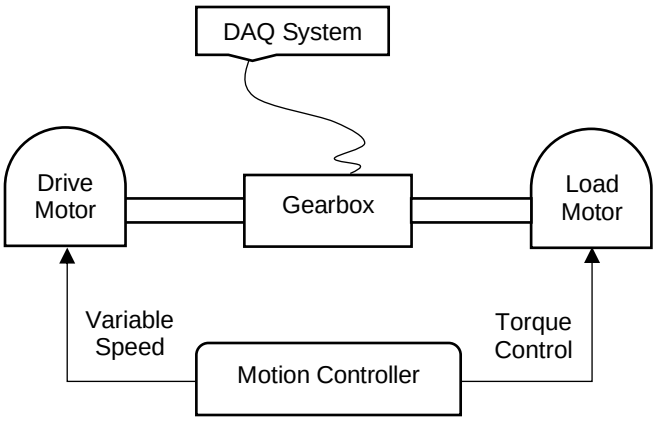

Figure 11: Controller and data aquisition set up

The gearbox itself is a one stage gearbox with spur gears that exhibit a gear reduction ratio of 1.8:1. The driving gear connected to motor 1 has 40 teeth while the driven gear on the side of motor 2 has 72 teeth. The gear sets used for the case studies are the same size expressed by the module length of 3 mm. Therefore, their pitch diameter, pressure angle, and tooth width are all equal. Their base circle diameter differs by the same speed reduction ratio of 1.8:1. The tooth width for both gears is 85 mm. A Computer Aided Design (CAD) model of the two gears inside of the gearbox is shown in Figure 12.

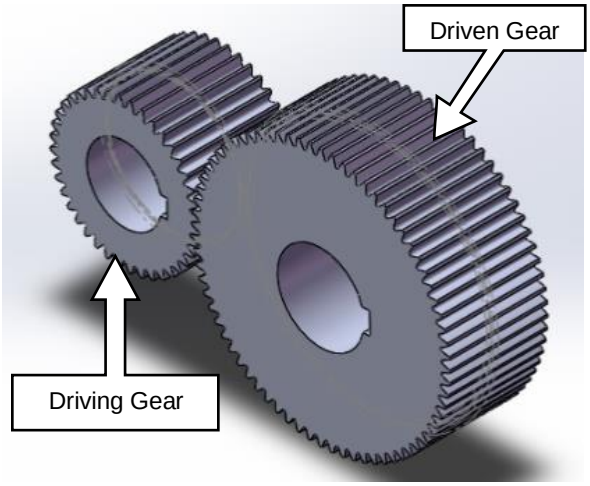

Figure 12: 3D model of the tested spur gears.

4.2 Method 1: Mode Amplitude Model

The mode amplitude method of fault diagnosis is based off the knowledge of healthy and faulty data sets. Healthy snapshots $[X_H, Y_H]$ were created from the normal operating conditions of the test rig. The snapshots were then mapped directly to an observable space. Since no augmentation was used in the mapping process, exact DMD is performed to solve for the Koopman operator model, $Y_H = K * X_H$. The solver outputs the spectral decomposition of the Koopman operator yielding the Koopman eigenvalues, eigenvectors, and modes. For this particular method, only the Koopman modes were used to create a mode amplitude model via, $\boldsymbol{B} = {\varphi_H}^+ * X$. This model takes the inverse of the healthy Koopman modes and returns either healthy or faulty mode amplitudes depending on the input X. Faulty snapshots $[X_F, Y_F]$ were created and X_F was input into the mode amplitude model to extract the faulty mode amplitudes. Once both healthy and faulty mode amplitudes were extracted, they are split into 80 percent training data and 20 percent testing data to be processed through a neural network. The specific neural network in this proposed method contains three nodes for the input layer, 100 nodes for the first hidden layer, 10 nodes for the second hidden layer, and returns a single node in the output layer. The testing data is then passed back through the neural network and returns a percent accuracy of the model itself. If the model returns a satisfactory percent accuracy, then it can be used to diagnose faults of the dynamical system.

Next, the same process is followed but instead of directly mapping the state variables to the observable space, the state variables are mapped through a Radial Basis Function (RBF) kernel to a high dimensional space. Specifically, the RBF kernel used in this application is thin-plate splines (TPS). The purpose of this RBF kernel is to spatially map any point in the state space to a functional or observable space. TPS interpolates the distance between a specific point and the center of clusters found via the k-means algorithm. Once the data is augmented, extended DMD is used to solve for the Koopman operator model, and the process continues. Figure 13 below shows the process of fault diagnosis via the mode amplitude method described above.

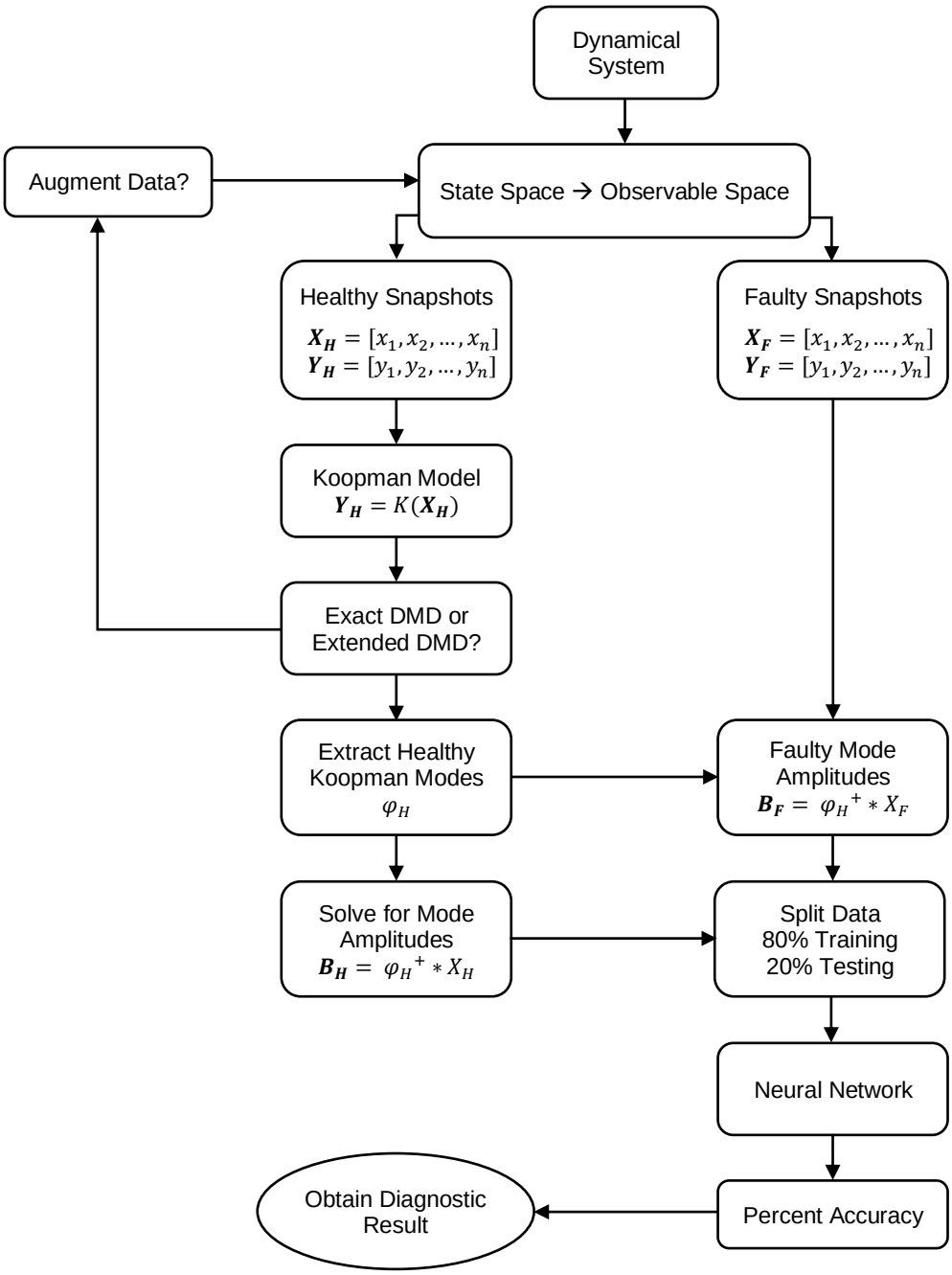

Figure 13: Detailed proccess of proposed method 1

The following sub-sections apply the mode amplitude method of fault diagnosis via two separate case studies. The first case study focuses on two system states, healthy and faulty, where healthy is normal operating conditions and faulty has some simulated level of pitting. The second case study focuses on a healthy system state as well as four other simulated levels of pitting with

increasing levels of pitting. The following case studies were tested and analyzed using the computational software MATLAB and programming language of Python.

4.2.1 Case Study 1: Healthy vs. Faulty

This case study used the gearbox test rig to compare a healthy gear, with no signs of ware or pitting, to a gear with controlled pitting on one of its teeth. First, the experiment ran under healthy conditions and vibration signals were collected into the DAQ system via the triaxial accelerometer. Next, the large pitting conditions are created on a single tooth as shown in Figure 14. The pitting was processed using electrical discharge machining to erode approximately 0.5 mm away from the face of the gear tooth. The operating conditions for both healthy and faulty data sets are a speed of 1000 rpm and 200 N of torque.

Figure 14: Single pitted gear tooth

The data that was collected for both healthy and faulty conditions was comprised of system states, or vibration data, in three directions: x, y, and z. The vibration measurements in three directions are categorized as three separate state variables. Without augmentation, the three state variables are directly considered as observables in the observable space. However, when augmenting the data, the state variables are mapped to a higher dimensional observable space, where the number of observables is larger than the original number of state variables.

The feature extraction method was similar for both methods. First, current and future state snapshots were created for both healthy and faulty system states. Then, a comparison of non-augmented and augmented data sets was performed for each of the following methods. Depending on whether the data set is augmented, either DMD or extended DMD was used to solve for the Koopman operator and extract significant system parameters such as Koopman eigenvalues, eigenvectors, or modes.

First, it is necessary to examine the original raw data. Below in Figure 15, the healthy and faulty vibration data is shown in the three directions taken by the tri-axial accelerometer. This is only a small sample taken from the entire data set. The full data set contains over six hundred thousand samples.

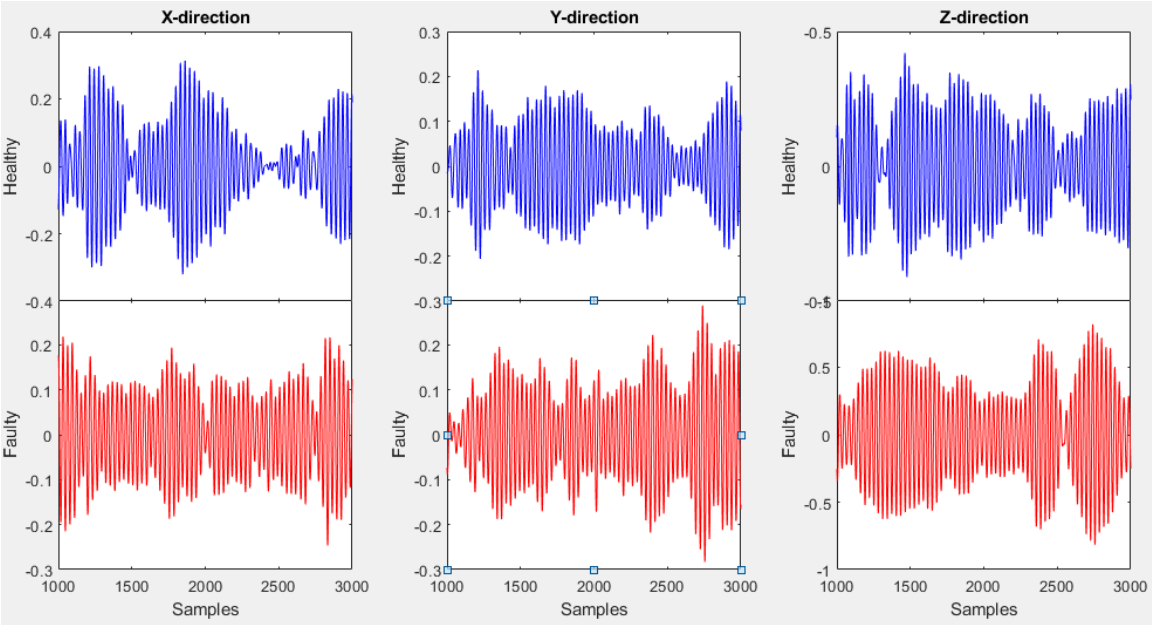

Figure 15: Raw vibration data

By simply analyzing the raw data above in each direction, it is difficult to diagnose which data sets are healthy and which are faulty. The diagnosis between these faults is the goal of this proposed approach. This data will be compared before and after an augmentation process, where both exact DMD and extended DMD is used to solve for the Koopman operator model. Initially, exact DMD is executed on the data, where no augmentation is performed. The existing state variables are directly mapped to the observable space. Once, the Koopman modes are extracted, they are used to create the mode amplitude model, $\boldsymbol{B} = {\varphi_H}^+ * X$. Using this model, both healthy

and faulty mode amplitudes are extracted. Every 1000 snapshots are averaged then split into 80 percent training and 20 percent testing data to be entered into a neural network. Figure 16 below shows the three mode amplitudes used to train the neural network.

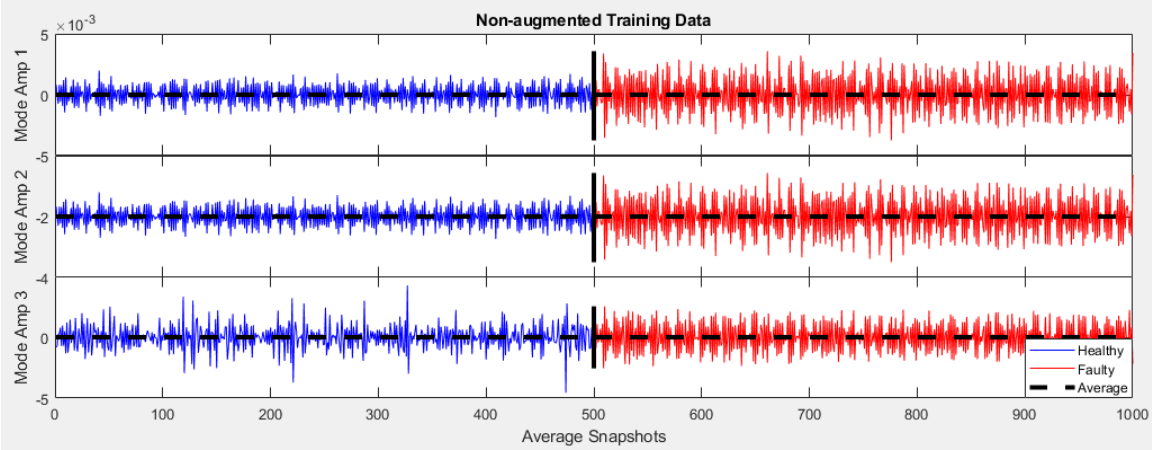

Figure 16: Healthy and faulty non-augmented neural network training data set

The neural network uses the training data to learn a mode amplitude model by attempting to distinguish between the healthy and faulty data sets. Once the model is trained, the neural network uses the remainder 20% of the data to test the proposed model. To evaluate the results of the tested model, a confusion matrix was created to determine the accuracy of the model. Table 2 below shows 10 trials were performed and an average percent accuracy was taken. It was determined that the non-augmented mode amplitude model was 72.44% accurate.

Table 2: Percent accuracy of the non-augmented mode amplitude model over two data sets

Trial	*Non-Augmented*
1	71.48%
2	71.85%
3	73.95%
4	74.79%
5	71.85%
6	73.11%
7	72.69%
8	71.01%
9	71.43%
10	72.27%
Average	**72.44%**

Next, the same process was completed using the RBF thin plate kernel to augment the original tri-axial state variables. The data was augmented from three state variables to ten observables. Again, from the original data it is difficult to diagnose a fault. However, when averaging the snapshots and splitting the data into training and testing, the difference between the healthy and faulty data becomes more noticeable as seen in Figure 17. The figure below shows all ten of the mode augmented mode amplitudes averaged every 1000 snapshots.

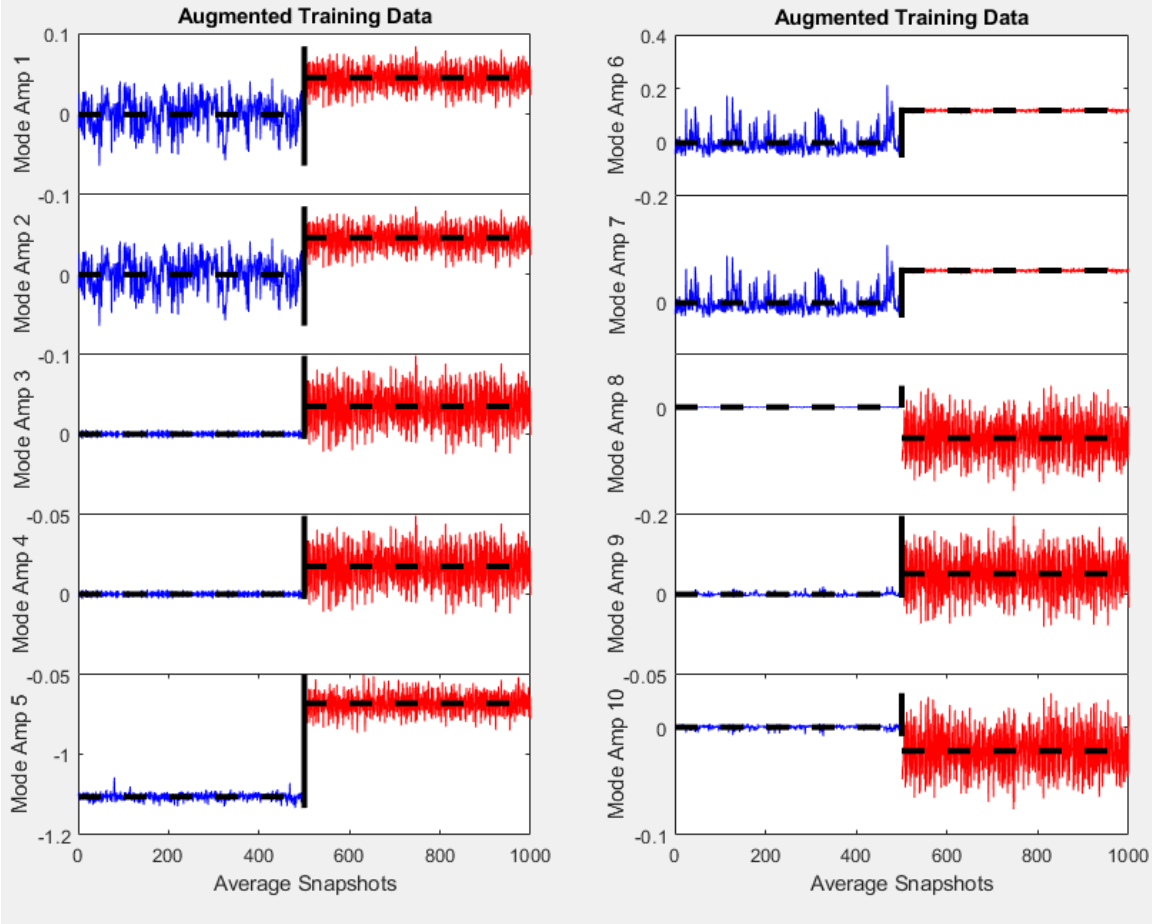

Figure 17: Healthy and faulty augmented neural network training data set

The data above from Figure 17 is used to train a new model in a neural network with the same characteristics. Similar to the non-augmented case, the neural network trains the model, the remainder 20 percent of the data is used to test the accuracy of the model. Table 3 below again shows 10 trials were performed and an average percent accuracy was recorded. However, this time,

the confusion matrix results in a 99.91% accuracy in determining the difference of healthy versus faulty data.

Table 3: Percent accuracy of augmented mode amplitude model over two data sets

Trail	*Augmented*
1	99.56%
2	100.00%
3	100.00%
4	100.00%
5	100.00%
6	100.00%
7	99.56%
8	100.00%
9	100.00%
10	100.00%
Average	**99.91%**

This case study used a healthy data set and a large fault data set to create a model which attempted to diagnose a fault in the dynamic system. The results above suggest that when the data is augmented, the mode amplitude model is approximately 99.91% accurate in determining whether or not a fault is present. However, when non-augmenting the data sets, the accuracy of the model decreases by 27.47%. A summary of these results is shown at the end of this section.

4.2.2 Case Study 2: Multiple Levels of Pitting

The second case study used the same test rig and experimental set up as before. Again, the healthy state was tested first, and vibration data was collected using the tri-axial accelerometer. This case study compares a healthy state and four simulated levels of fault as shown in Figure 18 below. Once more, the operating conditions for the healthy and faulty data sets are a speed of 1000 rpm and 200 N of torque.

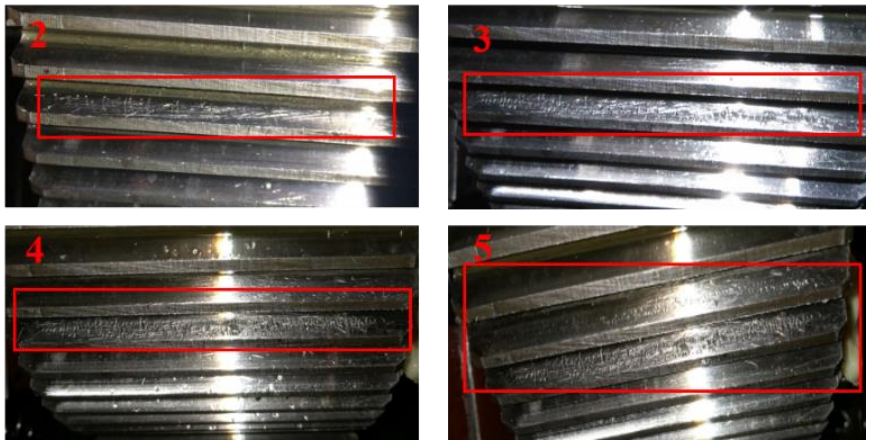

Figure 18: Four levels of simulated fault.

These pitting faults were manually created using a drill bit while the gear remained on the test rig. The first test case examined is a healthy state not shown in Figure 18. The second test case features light scratches on the surface of a single gear tooth. Test cases three and four both slightly increase the number of scratches. The final test case number five depicts the same scratches as test case four but also includes scratches on a second gear tooth. These four levels of fault are much smaller than the previous case study examined. The same process is followed from Figure 13, except there are now four levels of fault instead of one. Therefore, this case study will have five total data sets. The same comparison was made between non-augmenting and augmenting the test cases, including the same neural network setup.

Once again, the process begins with creating healthy snapshots from the test rig running under normal operating conditions. First, the snapshots were non-augmented and mapped directly to an observable space. By using DMD to solve the Koopman operator model (9), spectral decomposition was performed yielding the Koopman modes. The mode amplitude model, (10) was created using the healthy Koopman modes. For this case study, the mode amplitude model was now tested using the four levels of fault. The resulting mode amplitudes from all five data sets were averaged every 1000 snapshots then split into 80% training and 20% testing data to be entered into a neural network with the same configuration as case study 1. The non-augmented training data set from the first mode amplitude is shown below in Figure 19.

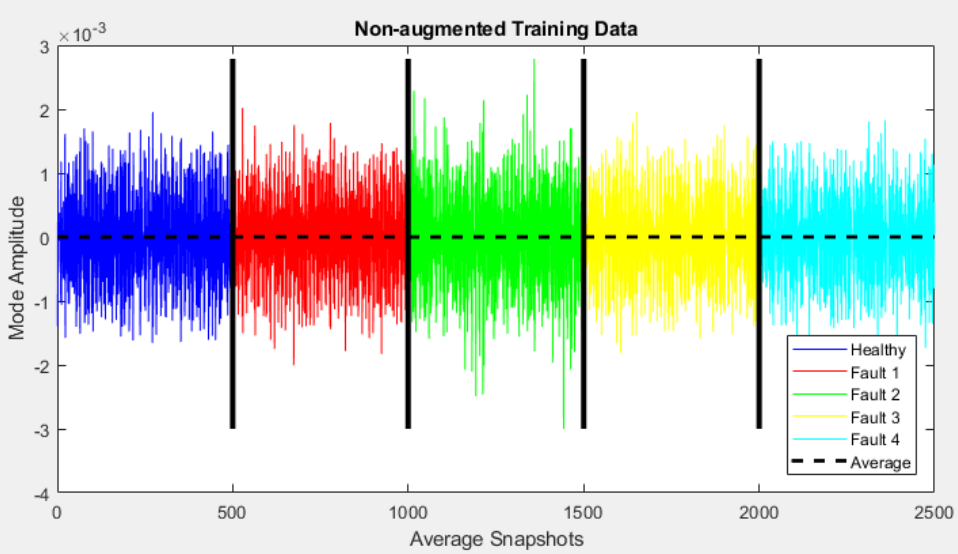

Figure 19: Healthy and four levels of fault non-augmented neural network training data set

The training data above is used in a neural network to attempts to distinguish between the five data sets and learn a mode amplitude model. The remainder 20% of the data not used in the training process was used to test the proposed model. Again, a confusion matrix was created to determine the accuracy of the tested model. Table 4 below shows an average of 30% percent accuracy taken over 10 trials.

Table 4: Percent accuracy of the non-augmented mode amplitude model over five data sets

Trial	*Non-Augmented*
1	40.00%
2	20.00%
3	20.00%
4	20.00%
5	40.00%
6	20.00%
7	40.00%
8	40.00%
9	40.00%
10	20.00%
Average	**30.00%**

Afterwards, the same augmenting process as in the first case study was performed using the RBF thin plate kernel on the original tri-axial state variables. The augmentation of the three state variables were mapped to a higher dimensional space with ten observables. When averaging every 1000 snapshots and splitting the data into training and testing data, the difference between the levels of pitting became more visible. Figure 20 below shows the first mode amplitude of all five augmented data sets.

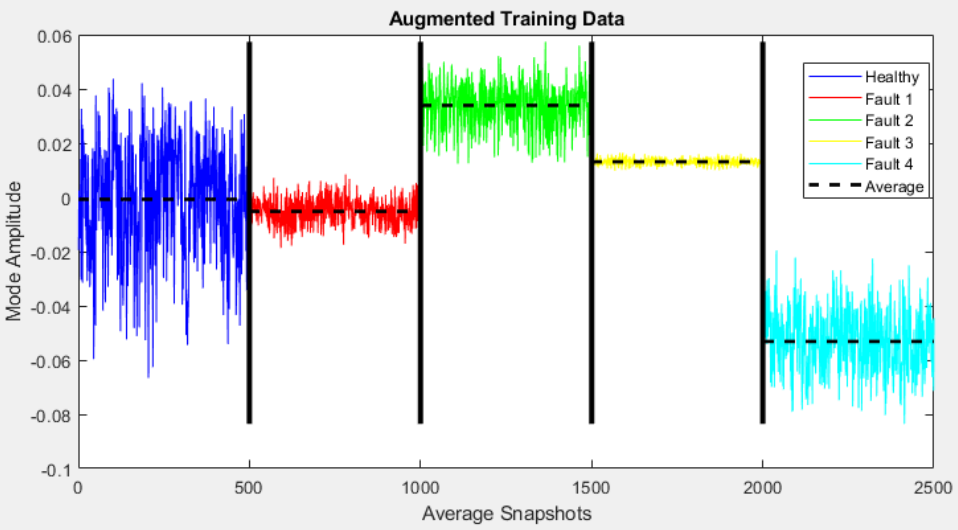

Figure 20: Healthy and four levels of fault augmented neural network training data set

The five training data sets from Figure 20 were used to train a new neural network model. As before, the remainder 20% of the data was used to test the accuracy of the created model under these new conditions. Table 5 below shows an average of 99.33% accuracy in determining the difference of the five levels of fault.

Table 5: Percent accuracy of the augmented mode amplitude model over five data sets

Trial	*Augmented*
1	99.44%
2	99.26%
3	99.26%
4	99.26%
5	99.26%
6	99.44%
7	99.26%
8	99.26%
9	99.44%
10	99.44%
Average	**99.33%**

4.2.3 Method 1 Summary

The proposed mode amplitude model method of fault diagnosis was applied over two separate case studies. Both case studies compared the accuracy of the model when the original state variables were either non-augmented or augmented. Table 6 below is an overview of the analysis performed on the proposed method.

Table 6: Average percent accuracy of the mode amplitude model

Case Study	*Non-augmented*	*Augmented*
Healthy vs. Faulty	72.44%	99.91%
Multiple Levels of Pitting	30.00%	99.33%

The results from Table 6 show that augmenting the original state variables is crucial to the accuracy of the mode amplitude model. By definition, the Koopman operator itself exists in a high dimensional space. The augmentation process allows for the complex information from the non-linear dynamic system to unfold from the original state space into a higher dimensional observable space. When the state variables are augmented to a larger number of observables, the Koopman operator model provides a more accurate representation of the dynamics. By augmenting the first case study, “Healthy vs. Faulty”, the accuracy of the model increases by 27.47%. When augmenting the second case study, “Multiple Levels of Pitting”, the model accuracy is greatly increased by 69.33%. The lack of accuracy of the non-augmented data set for the second case study originates

from the experiment itself. The faults under the second case study were much smaller than the single fault of the first case study. Since the first case study contained such a large fault, the model more accurately differentiated between the healthy and faulty data sets. Regardless of the initial non-augmented data in either case study, performing the augmentation process greatly increases the accuracy of the model in determining the existence of a fault.

4.3 Method 2: Prediction Model

The purpose of this method is to recreate the original non-linear system dynamics via a prediction model. Once the prediction model is applied, a statistical analysis method is employed to determine the possible existence of a fault. Similar to method 1, the feature extraction process begins with the creation of healthy snapshots, $[X_H, Y_H]$. After the Koopman operator model is created, it is first solved by exact DMD and performs spectral decomposition of the Koopman operator itself. In addition to the healthy Koopman modes used in method 1, the prediction model (12) also uses the obtained healthy Koopman eigenvalues. This model uses both healthy Koopman eigenvalues and Koopman modes and returns either a healthy or faulty future state snapshot depending on the input X. First, X_H was input into the prediction model to produce a healthy predicted future state snapshot, Y_H. Faulty snapshots $[X_F, Y_F]$ were created and X_F was input into the prediction model to output a faulty predicted future state snapshot, Y_F. The predicted results were then tested against the original future state snapshots via a root mean sum error (RMSE) test. The RMSE tests the Euclidean distance between the original future state snapshots and the predicted future state snapshots. This prediction model is able to predict some time step, τ, forward in time. The results are obtained at varied time steps and again examined via the RMSE test. Next, the same process is followed while augmenting the data sets. The same RBF thin-plate kernel from the first method was used to augment these data sets. Figure 21 below shows a detailed flow chart of the proposed prediction model method.

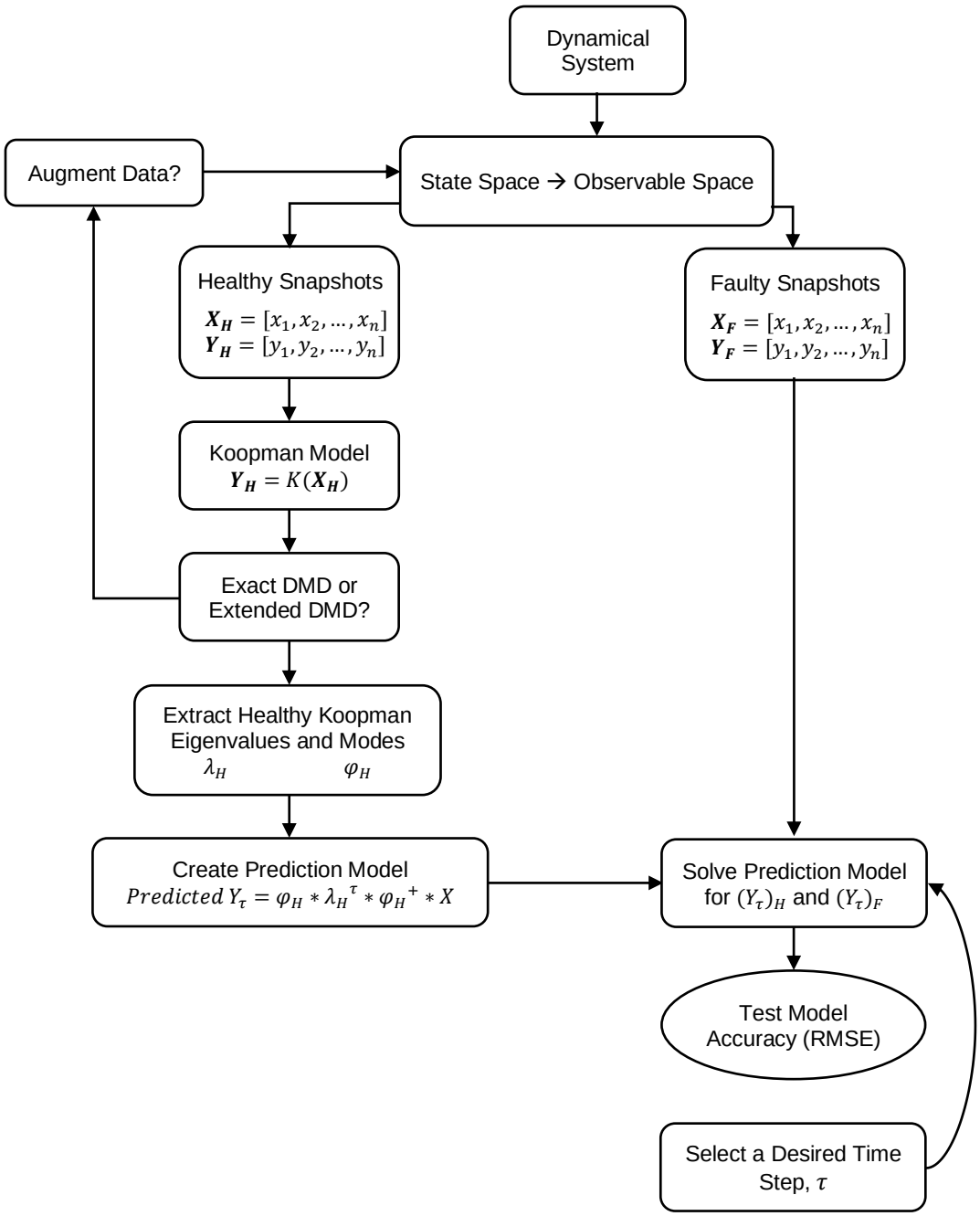

Figure 21: Detailed proccess of proposed method 2

The purpose of this prediction model is to use the solved Koopman parameters from a healthy data set to predict a faulty data set without ever solving for the faulty Koopman parameters. Therefore, if enough data is collected under healthy system conditions, and a fault becomes present in the system itself, it would be possible to determine the residual error between the two system

states. The following two sub-sections apply the prediction model method using the same experimental set up as the two case studies used in method 1.

4.3.1 Case Study 1: Healthy vs. Faulty

The gearbox test rig in this case study first obtained data from a healthy gear and compared it to the same gear with an added level of pitting. Figure 14 from the above section shows the level of pitting created for this case study. The data collected from the section above is also the same data used for applying the prediction model method. The original vibration data for this case is shown in Figure 15 of the previous section. Initially, the healthy vibration data in three directions, state variables, is mapped directly to an observable space. The Koopman operator model is then solved by exact DMD and returns the healthy model parameters of Koopman eigenvalues and Koopman modes. Once these parameters are extracted, they are used to create the prediction model, (12). Using this model, both healthy and faulty future state snapshots are predicted, Y_H and Y_F. The goal of this prediction model will be to accurately predict the faulty data set with the healthy Koopman parameters.

Figure 22: Raw vibration data compared

First, it is necessary to examine the original healthy vibration data overlapped with the original faulty data. Figure 22 above shows a comparison of healthy vs. faulty raw vibration data taken by the triaxial accelerometer. As before, this is a small sample taken from a much larger data set. The figure above shows the future state snapshots of both system states. Then, the Koopman operator model is built from the current and future healthy state snapshots. Through exact DMD,

the Koopman eigenvalues and Koopman modes are extracted as parameters for the prediction model. Once the prediction model is created, it is first tested at a single time step.

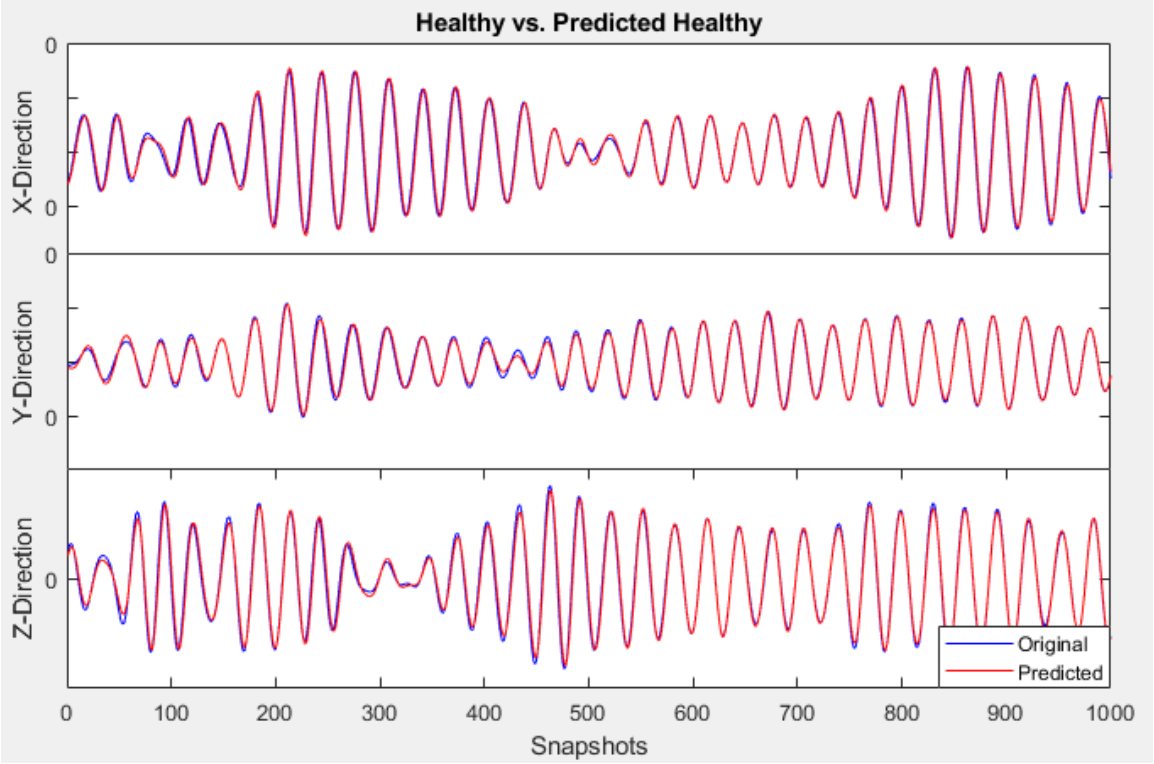

Figure 23: Testing the single step prediction model with healthy data

The single step prediction model in Figure 23 above is tested by importing the healthy current state snapshot, X_H, to output a predicted future state snapshot, Y_H. A statistical comparison of the original future state and the predicted future state is performed via RMSE. The percent error of the non-augmented, single step prediction model is 1.79% when testing the model with the predicted healthy future state.

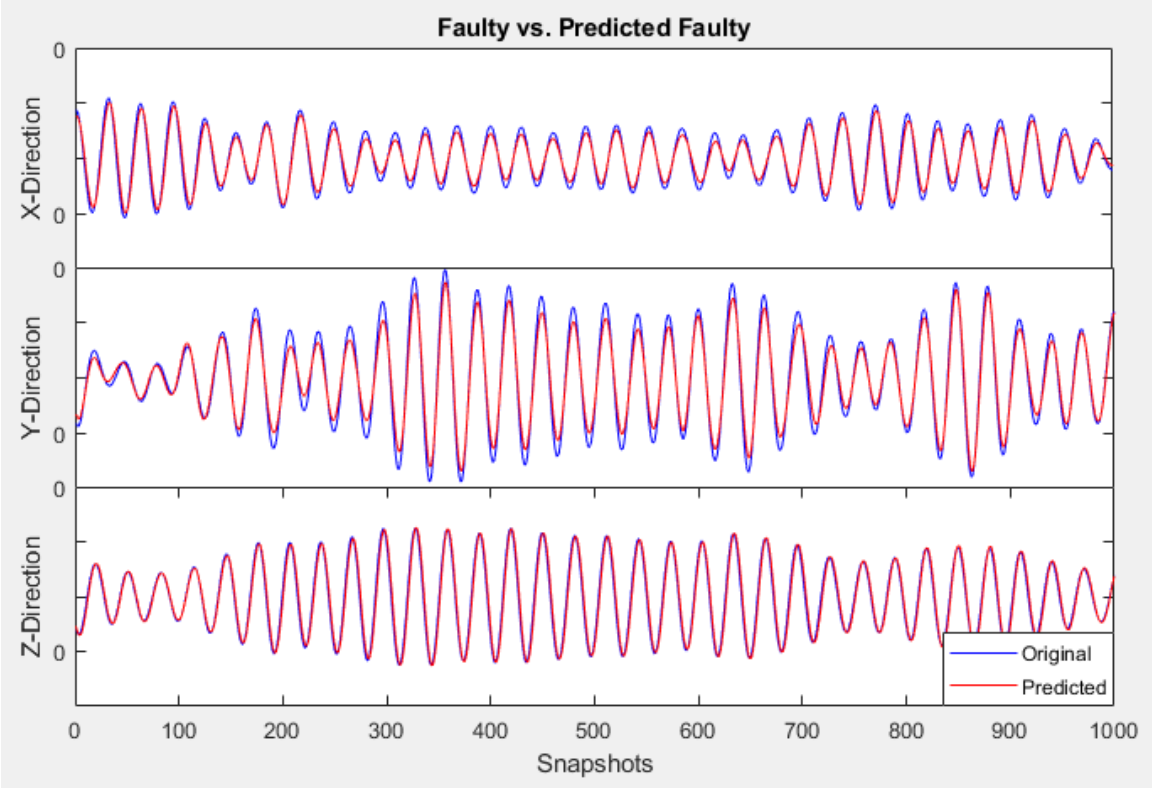

Figure 24: Testing the single step prediction model with faulty data

Figure 24 above shows the non-augmented single step prediction model being tested by importing the faulty current state snapshot, X_F, to output a predicted future state snapshot, Y_H. The percent error found from RMSE of the above analysis is 2.91%.

Next, the same process is repeated with the augmenting of the original state variables with the RBF thin-plate kernel. The Koopman operator model is now solved via extended DMD and new Koopman eigenvalues and Koopman modes are extracted as parameters for the prediction model. As before, the model is tested at a single time step by importing healthy and faulty current state to produce predicted healthy and faulty future state. Testing the predicted healthy future state with the original healthy future state, the RMS percent error was 0.89%. The percent error found from testing the predicted faulty future state, with the original faulty future state was 1.54%.

Table 7: Testing the single step prediction model of healthy and faulty data sets

	Non-augmented	Augmented
Predicted Healthy State RMSE (%)	1.79	0.89
Predicted Faulty State RMSE (%)	2.91	1.54

A summary of the RMSE percent error of the single step prediction model using healthy and faulty data is shown in Table 7 above. The percent error was compared for both non-augmented and augmented data. Since the prediction model itself uses the healthy Koopman eigenvalues and Koopman modes, testing the healthy state percent error was expected to have a lower value than testing the faulty state percent error.

4.3.2 Summary of Method 2

An analysis was conducted of the RMS percent error of both healthy and faulty data when increasing the time step (τ). The analysis tested both the non-augmented and the augmented data sets. When increasing the time step, the prediction model is able to predict however many future state snapshots in advance. In this analysis, the time step was increased from a single time step to 5, 10, 25, and 50 time steps. For clarity, when τ is equal to 50, the original system dynamics are reconstructed and are predicted 50 snapshots in the future. Again, the RMSE of both healthy and faulty data sets were examined. Figure 25 below, shows the RMSE of non-augmented vs augmented at different time steps.

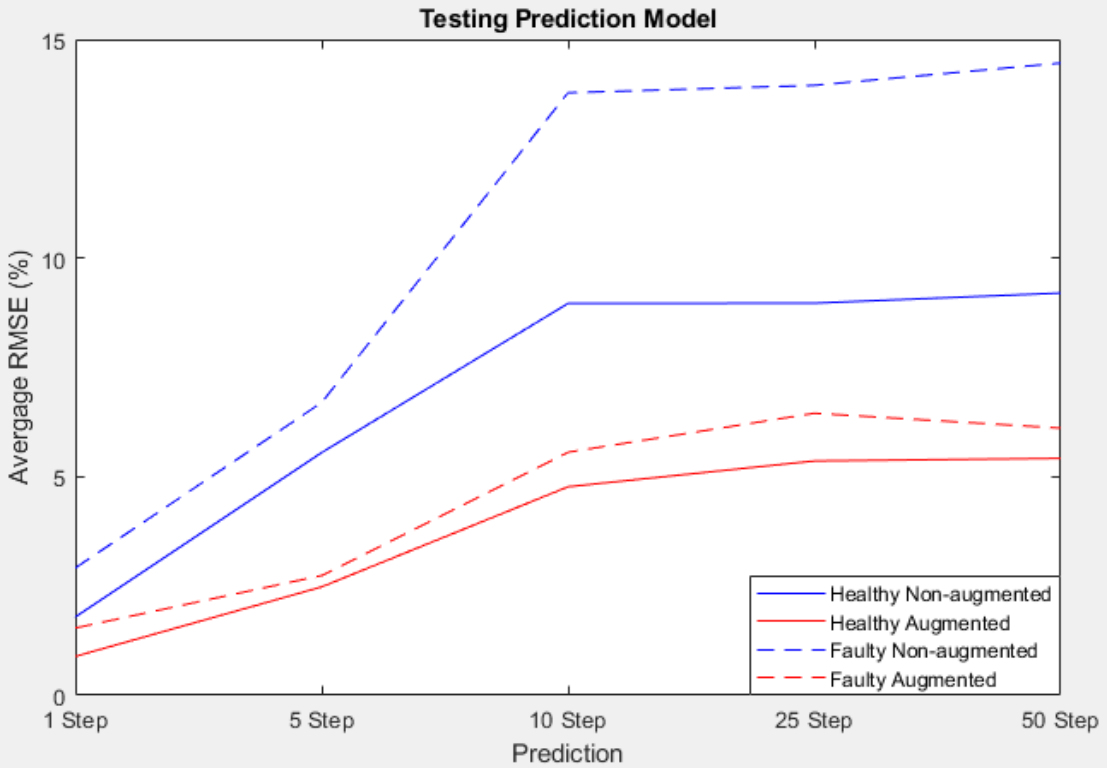

Figure 25: Comparing non-augmented vs augmented data for healthy and faulty RMSE

The figure on the left shows the comparison of non-augmented and augmented data sets when the prediction model tests the healthy future state snapshots. The figure on the right shows the same comparison but the prediction model tests the faulty future state snapshots. The values from Figure 25 are examined in the table below.

Table 8: Comparing non-augmented vs augmented data for healthy and faulty RMSE

	Healthy RMSE (%)	
	Non-augmented	*Augmented*
1 Step	1.79	0.89
5 Step	5.55	2.48
10 Step	8.95	4.77
25 Step	8.96	5.36
50 Step	9.20	5.42

	Faulty RMSE (%)	
	Non-augmented	*Augmented*
1 Step	2.91	1.54
5 Step	6.70	2.74
10 Step	13.78	5.56
25 Step	13.95	6.45
50 Step	14.46	6.11

Table 8 above is a summary of the analysis of the non-augmented vs augmented data sets for the healthy and faulty cases. As expected, the healthy data set provides a lower percent error regardless of an augmentation being performed. As the time step of the prediction model is increased, the percent error increases as well. However, when augmented either data set, the percent error is decreased throughout. In both healthy and faulty cases, the augmented 50 step model has a lower percent error than the non-augmented 10 step model. Overall, two conclusions can be drawn from the application of the prediction method. First, the augmentation of data helps with the accuracy of the predication model. However, it does not help to improve the performance of fault diagnosis. Secondly, the increase in step size benefits the separation of healthy and faulty data and can lead to a higher accuracy of diagnosis decision.

4.3 Discussion of Results

The first proposed method used a derived mode amplitude model to extract key features from the original system dynamics and import them into a neural network. The neural network in turn was able to classify different data sets and a confusion matrix was used to return a percent accuracy of the model. The results from testing this method with the first case study, “healthy vs. faulty”, showed a percent accuracy of 72.44% when using DMD to solve for the Koopman operator parameters. When the original vibration data was augmented, extended DMD was performed to extract the model parameters yielding an increased percent accuracy of 99.91%. The second case study, “multiple levels of pitting”, contained a smaller discrepancy between the original data sets. When performed DMD of the original non-augmented system states, the percent accuracy was decreased to 30% accuracy. However, when again augmenting the state variables and performed extended DMD, the percent accuracy of the second case study was increased to 99.33%. Overall, both case studies showed that augmenting the original states of the nonlinear dynamic system greatly increased the performance of the proposed method.

The second proposed method incorporated a prediction model whose goal was to reconstruct the original system dynamics. The prediction model itself was created from extracted Koopman model parameters via DMD or extended DMD. A time step can be introduced into the model to determine how far away from the original system dynamics the model is able to predict. Once the dynamics of the original system were reconstructed, a statistical analysis method, RMS was implemented to determine the percent error between the original state and the predicted state.

This proposed method was tested by the first case study, “healthy vs. faulty”, containing two different data sets. The Koopman features used in this model were derived solely from the healthy original system state. Results for the prediction model included testing the model against the healthy data set and the faulty data set separately. Testing the healthy data set at a single time step for both the non-augmented and augmented cases resulted in a respective percent error of 1.79% and 0.89%. Again, the non-augmented and augmented cases were tested for the faulty data set and produced a percent error of 2.91% and 1.54%, respectively. In both cases, augmented the data set resulted in a lower percent error. When increasing the time step of the prediction model, the error increased all cases. However, the augmented data always resulted in a lower percent error than the non-augmented counterpart. Although the percent error decreases with augmentation, this process does not create separation between healthy and faulty data. The separation necessary to increase the fault diagnosis capability of this method is done by increasing the time step of the prediction model.

5. CONCLUSION

In this thesis, two proposed methods of hybrid fault diagnosis were presented. Both methods were based on solving the Koopman operator through DMD. The Koopman operator was used as a tool to create a linear model from a nonlinear dynamical system. The spectral decomposition of the Koopman operator yielded parameters which were used to create the models for two separate methods. Two case studies were used to validate the proposed approaches using raw vibration data from a gearbox test rig. The first case study contained two data sets, a healthy data set under normal operating conditions and a data set which contained a large pitting fault. The second case study referred to 5 total data sets, starting with the healthy system state and 4 levels of small increasing faults. The case studies used both DMD and extended DMD to solve for the Koopman operator. The extended DMD solution incorporated an augmentation process of the original vibration data.

The performance of these methods is promising for the prognostic health management (PHM) field. Both methods have the potential to be used for generic industrial applications. The first method has potential to identify faults from non-linear dynamic systems while the second method could be used for prediction and control applications. Future work regarding the proposed methods is to investigate further on the robustness for multilevel fault diagnosis.

To conclude, methods for fault diagnosis and prediction were proposed and applied via case studies. Both methods showed promising results when augmenting the original data sets from a state space to an observable space. However, when the state variables were directly used in the Koopman operator model, the accuracy of the proposed methods decreased significantly. The results from this thesis indicate that augmenting the original data to a high dimensional space can boost prediction performance and help with more accurate diagnosis results.

CITED LITERATURE

[1] Poincare, H. (1892). *New Methods of Celestial Mechanics* (D. Goroff, ed.). Paris, France.

[2] Holton, J. R., & Hakim, G. J. (2012). An introduction to dynamic meteorology: Fifth edition. In *an Introduction to Dynamic Meteorology: Fifth Edition* (Vol. 9780123848666).

[3] Baredar, P., Khare, V., & Nema, S. (2020). Biogas digester plant. *Design and Optimization of Biogas Energy Systems*, 79–155.

[4] Höfling, T., & Isermann, R. (1996). Fault detection based on adaptive parity equations and single-parameter tracking. *Control Engineering Practice*, *4*(10), 1361–1369.

[5] Odendaal, H. M., & Jones, T. (2014). Actuator fault detection and isolation: An optimised parity space approach. *Control Engineering Practice*, *26*(1), 222–232.

[6] Svärd, C., & Nyberg, M. (2008). Observer-based residual generation for linear differential-algebraic equation systems. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, *17*(1 PART 1).

[7] Isermann, R. (1984). Process fault detection based on modeling and estimation methods-A survey. *Automatica*, *20*(4), 387–404.

[8] Simani, S., Fantuzzi, C., & Patton, R. J. (2003). *Model-based Fault Diagnosis in Dynamic Systems Using Identification Techniques*.

[9] Abid, M. (2010). *Fault detection in nonlinear systems: An observer-based approach.* (January).

[10] Espinoza-Trejo, D. R., Campos-Delgado, D. U., Bossio, G., Bárcenas, E., Hernández-Díez, J. E., & Lugo-Cordero, L. F. (2013). Fault diagnosis scheme for open-circuit faults in field-oriented control induction motor drives. *IET Power Electronics*, *6*(5), 869–877.

[11] Peach, N., Basseville, M., & Nikiforov, I. V. (1995). Detection of Abrupt Changes: Theory and Applications. *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, *158*(1), 185.

[12] Ding, S. X. (2013). Model-Based Fault Diagnosis Techniques: Design Schemes, Algorithms, and Tools. In *Journal of Chemical Information and Modeling* (Vol. 53).

[13] Montáns, F. J., Chinesta, F., Gómez-Bombarelli, R., & Kutz, J. N. (2019). Data-driven modeling and learning in science and engineering. *Comptes Rendus - Mecanique*, *347*(11), 845–855.

[14] Han, E. S., & goleman, daniel; boyatzis, Richard; Mckee, A. (2019). International Journal of Advanced Research in Artificial Intelligence. *Journal of Chemical Information and Modeling*, *53*(9), 1689–1699.

[15] Darwish, H. A., Taalab, A. M. I., & Kawady, T. A. (2001). Development and implementation of an ANN-based fault diagnosis scheme for generator winding protection. *IEEE Transactions on Power Delivery*, *16*(2), 208–214.dfd

[16] Bai, Y., Sun, Z., Zeng, B., Long, J., Li, L., de Oliveira, J. V., & Li, C. (2019). A comparison of dimension reduction techniques for support vector machine modeling of multi-parameter manufacturing quality prediction. *Journal of Intelligent Manufacturing*, *30*(5), 2245–2256.

[17] Skliros, C., Esperon Miguez, M., Fakhre, A., & Jennions, I. K. (2019). A review of model based and data driven methods targeting hardware systems diagnostics. *Diagnostyka*, *20*(1), 3–21.

[18] Kramer, M. A. (1992). Autoassociative neural networks. *Computers and Chemical Engineering*, *16*(4), 313–328.

[19] Pearl, J. (2013). Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Interference. In *Journal of Chemical Information and Modeling* (Vol. 53).

[20] Atoui, M. A., Verron, S., & Kobi, A. (2015). Fault detection and diagnosis in a Bayesian network classifier incorporating probabilistic boundary. *IFAC-PapersOnLine*, *28*(21), 670–675.

[21] Zhang, J. (2016). *Model-Based Fault Diagnosis For Automotive Functional Safety*. (January 2016).

[22] Liang, J., & Du, R. (2007). Model-based Fault Detection and Diagnosis of HVAC systems using Support Vector Machine method. *International Journal of Refrigeration*, *30*(6), 1104–1114.

[23] Ahmad, M. W., Mourshed, M., Yuce, B., & Rezgui, Y. (2016). Computational intelligence techniques for HVAC systems: A review. *Building Simulation*, *9*(4), 359–398.

[24] Tidriri, K., Chatti, N., Verron, S., & Tiplica, T. (2016). Bridging data-driven and model-based approaches for process fault diagnosis and health monitoring: A review of researches and future challenges. *Annual Reviews in Control*, *42*(March 2018), 63–81.

[25] Brunton, S. L. (2018). Notes on Koopman Operator Theory. *ArXiv*, (3), 1–37.

[26] Arabai, H. (2019). Introduction to Koopman operator theory of dynamical systems. *Growth Lakeland*, *54*(June), xviii+802.

[27] Williams, M. O., Kevrekidis, I. G., & Rowley, C. W. (2015). A Data–Driven Approximation of the Koopman Operator: Extending Dynamic Mode Decomposition. *Journal of Nonlinear Science*, *25*(6), 1307–1346.

[28] Klema, V. C., & Laub, A. J. (1980). The Singular Value Decomposition: Its Computation and Some Applications. *IEEE Transactions on Automatic Control*, *25*(2), 164–176.

[29] Yeung, E., Kundu, S., & Hodas, N. (2019). Learning deep neural network representations for koopman operators of nonlinear dynamical systems. *Proceedings of the American Control Conference*, *2019-July*, 4832–4839.

[30] Li, Q., Dietrich, F., Bollt, E. M., & Kevrekidis, I. G. (2017). Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the koopman operator. *Chaos*, *27*(10), 1–25.