Fall 2009

Fault detection and prediction with application to rotating machinery

Gary R. Halligan

Department:

FAULT DETECTION AND PREDICTION WITH APPLICATION TO ROTATING MACHINERY

by

GARY HALLIGAN

A THESIS

Presented to the Faculty of the Graduate School of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE in ELECTRICAL ENGINEERING

2009

Approved by
Jagannathan Sarangapani, Advisor
Maciej Zawodniok
R. Joe Stanley

PUBLICATION THESIS OPTION

This thesis consists of the following two articles that have been submitted for publication as follows:

Paper 1, pages 9-32, Gary R. Halligan and S. Jagannathan, “PCA-based Fault Isolation and Prognosis with Application to Pump,” has been submitted to International Journal of Advanced Manufacturing Technology.

Paper 2, pages 33-76, Gary R. Halligan, Balaje T. Thumati, and S. Jagannathan, “A Novel Fault Detection and Prediction Scheme in Discrete-time Using a Nonlinear Observer and Artificial Immune System as an Online Approximator,” intended for submission to IEEE Trans. on Control System Technology.

ABSTRACT

In this thesis, the detection and prediction of faults in rotating machinery is undertaken and presented in two papers. In the first paper, Principal Component Analysis (PCA), a well known data-driven dimension reduction technique, is applied to data for normal operation and four fault conditions from a one-half horsepower centrifugal water pump. Fault isolation in this scheme is done by observing the location of the data points in the Principal Component domain, and the time to failure (TTF) is calculated by applying statistical regression on the resulting PC scores. The application of the proposed scheme demonstrated that PCA was able to detect and isolate all four faults. Additionally, the TTF calculation for the impeller failure was found to yield satisfactory results.

On the other hand, in the second paper, the fault detection and failure prediction are done by using a model based approach which utilizes a nonlinear observer consisting of an online approximator in discrete-time (OLAD) and a robust adaptive term. Once a fault has been detected, both the OLAD and the robust adaptive term are initiated and the OLAD then utilizes its update law to learn the unknown dynamics of the encountered fault. While in similar applications it is common to use neural networks to be used for the OLAD, in this paper an Artificial Immune System (AIS) is used for the OLAD. The proposed approach was verified through implementation on data from an axial piston pump. The scheme was able to satisfactorily detect and learn both an incipient piston wear fault and an abrupt sensor failure.

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Jagannathan Sarangapani, for his patience, guidance, and support. I also would like to thank Dr. Maciej Zawodniok and Dr. R. Joe Stanley for serving on my masters committee.

I would also like to thank my parents, Larry and Nancy, my brothers, Matt and Greg, as well as the rest of my friends and family for their love and support. Their understanding, encouragement, and constant support during my education have been invaluable. I would also like to thank God for providing me these opportunities and the courage to pursue them.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

1. INTRODUCTION

Many types of machines are currently being used in modern industrial processes all of them susceptible to many different failure modes. These failure modes can have consequences ranging from a mild inconvenience to life-threatening situations. Additionally, every fault occurrence has an associated cost. This cost include the parts and labor for repairing or replacing the failed system component, as well as the cost incurred by the production line shutdown, and possibly the repair of the collateral damage. Because of the associated risks and costs of failures, early detection, isolation, and prediction of faults is important.

1.1 OVERVIEW OF FDP METHODOLOGIES

Typically, the fault detection and prediction (FDP) schemes currently in use belong in one of two categories: data-based, or model-based. Both categories operate on the same two fundamental concepts:

- 1) Thresholds can be established to distinguish between healthy and faulty data; an example is shown in Fig. 1.1 where Var 1 and Var 2 are two of the system parameters, and
- 2) The faults encountered by the system will be distinct, i.e. separable ; as shown in Fig. 1.2 where Var 1 and Var 2 are two of the system parameters, N is the normal operation region and F1-F4 are four known fault types.

The key difference between the two categories is the approach used to establish the thresholds required for heath determination as well as fault classification.

Data-based schemes such as FFT, wavelets, or PCA which is used in this thesis, require the collection of data to determine the ranges each system parameter can take under normal condition. Based on those ranges, thresholds can be set beyond which the data is deemed to be faulty. However, this does not mean the fault has been identified. It simply means that the system has encountered a fault, but which fault has yet to be determined. If data typical for a particular fault is available, then it is possible to identify thresholds corresponding to the particular fault classes. If the faulty data falls within the boundaries of a known fault class, the data is then determined to be of that class. If the data does not fall within the boundaries of any known fault class, the data based methodology will be able to detect, but not isolate, the fault being experienced. Additionally, if two faults are indistinguishable, a data based methodology will frequently incorrectly classify both faults as a single fault.

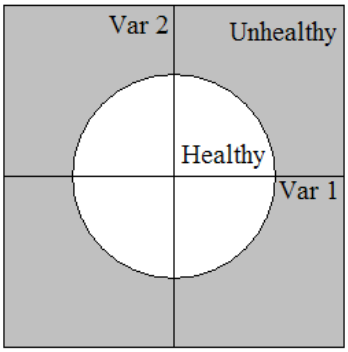

Fig 1.1 Fault Detection Example

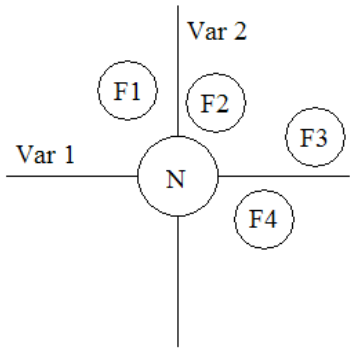

Fig 1.2 Fault Isolation Example

A major disadvantage of such methodologies is the created model's dependence on the data provided to create the model. Also data –based schemes require the collection of data that is representative of all modes of healthy operation as well as each fault type in each of the healthy operation modes. If this is not done, the performance of the data-based methodology will suffer. For example, if a fault has not been represented in all operation modes, then a known fault can be classified incorrectly or not at all if the system is operating in a mode in which the given fault has not been represented. Another example is if all healthy operation modes have not been accounted for in the model, the algorithm may classify a data from a healthy operation mode as faulty. As a result of the dependence of the model on the data, data collection for such a scheme can be quite costly in terms of storage space, as well as the time and money invested.

In contrast, model-based methodologies require a suitable mathematical model of the system. The variation of the modeled output can then be used for establishing a suitable fault detection threshold. This means that data is not needed for the selection of the fault detection threshold. In model-based methodologies, an observer observes the system parameters and compares them to the model to create a residual. The residual is

then compared to the fault detection threshold determined by the model. Because no data is required, fault detection and learning can be done online without any offline training. Additionally, no data is required for fault isolation. However, as in the data-based methodologies, isolation requires that a given fault has been experienced before. Otherwise, the fault can only be detected. Additional, one of the disadvantages of model-based methodologies is the fact that models can be quite complex, computationally expensive, or difficult to obtain.

1.2 ORGANIZATION OF THE THESIS

In this thesis, fault detection and prediction for rotating machinery is undertaken by utilizing schemes in a novel approach that has not been presented in existing literature. This thesis is presented in two papers. Their relation to each other is illustrated in Fig. 1.3. The common theme of these papers is the detection and prediction of faults on rotating machinery, or, more specifically, pumps.

In the first paper, a data-based method called Principal Component Analysis (PCA) is used to detect four different fault types (impeller failure, seal failure, sensor failure, and a filter clog) which were seeded on a centrifugal water pump test bed. PCA is a multivariate statistical analysis technique which is typically used for dimension reduction, fault detection, and fault isolation [1-7]. PCA uses eigenanalysis to determine the relationships between different system parameters. Using PCA for fault detection and isolation assumes that the relationships between the variables will change as the relative system health changes.

In contrast, the second paper implements model-based approach utilizing the system dynamic equations of a 10.5cc axial piston pump and an Artificial Immune System (AIS) as an online approximator in discrete time (OLAD) for the detection and

learning, in real time, of two different fault types (piston wear and sensor failure). The AIS used in this paper is inspired by biological immune systems, such as the human immune system.

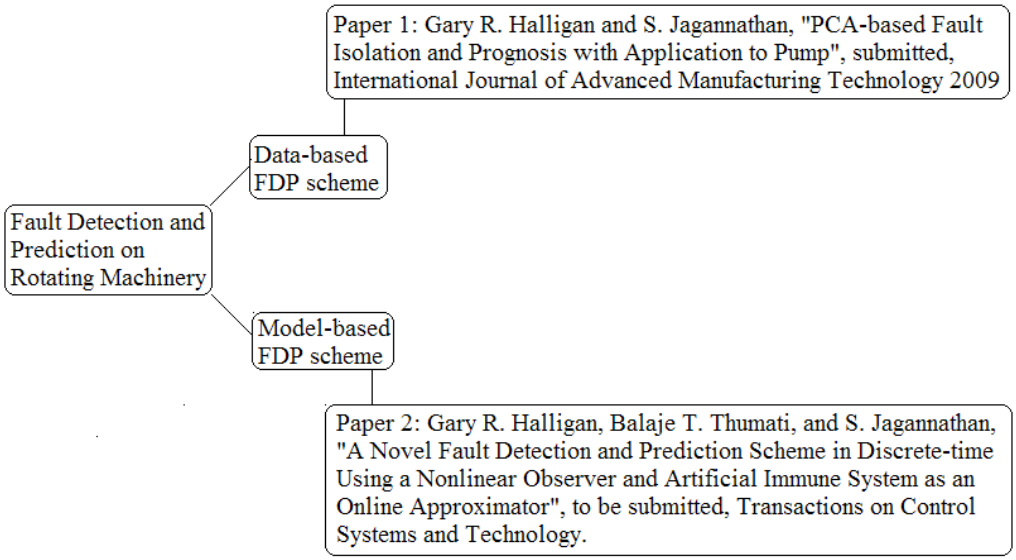

Fig 1.3 Thesis Outline

A biological immune system protects the body from invasion by antigens such as viruses or bacteria. To do this, the immune system has two primary types of white blood cells floating in the blood stream, T-cells and B-cells. When a T-cell encounters another cell, it determines if it is an antigen or not. If the cell is not an antigen, the T-cell moves on. If, however, the cell is an antigen, the T-cells will swarm the antigen in an attempt to destroy the antigen [8]. In addition to the search and destroy behavior, the T-cell also triggers the B-cells to start generating antibodies to fight the invasion [8]. The B-cells effectively “remember” all antigens that have been encountered previously. In the event the immune system encounters a new antigen, the B-cells compare the new antigen to the

previously encountered ones. The B-cell will then release the antibody that was effective on the most similar previous antigen. Over time, the B-cell learns how to fight the new antigen and stores that information in memory [8].

In the second paper of this thesis, the T-cells are determining whether or not each data point is faulty, which is the counterpart to being an antigen in the biological system [8]. When a data point is considered faulty, the OLAD, or B-Cells, is triggered. Each subsequent data point is then treated as an antigen would in a biological system. With each data point the OLAD learns the fault.

1.3 CONTRIBUTIONS OF THE THESIS

This thesis provides contributions to the field of fault detection and prediction. PCA was extended to the area of fault prognosis. In Paper 1, the proposed methodology uses only the Principal Component scores, also called PC scores, and basic regression to isolate and predict faults. The PCs need to be determined offline. However, once the PCs have been determined, they can be implemented in real time for fault detection and determining which know fault the system is likely experiencing. Using the knowledge of the likely fault, a time to failure calculation is achieved utilizing least squares regression on a combination of appropriate PCs. Additionally, past literature [2-8] depicted that PCA typically applied to detect faults in processes and was rarely applied to an individual machine. When it was applied to a single machine [6-8], it was typically used in combination with other algorithms such as wavelets or other statistical features such as kurtosis and skewness are introduced. For the approach proposed in this thesis, the PCA was applied to a single centrifugal water pump.

While in literature the AIS has been mainly used as offline for classification and pattern recognition [8-18], in Paper 2 of this thesis, the AIS is used as an OLAD for the detection and learning of the fault in real time. Additionally, the AIS-based methods in literature [11, 13-18] are data driven and require offline training. In contrast, the methodology proposed here uses the AIS is tuned online without a priori training. The proposed methodology was verified in simulation on a two degree of freedom manipulator, and a hydraulic 10.5cc axial piston pump. It was further verified through the application to data from the pump mentioned above.

1.4 REFERENCES

[1] Doan X. Tien, Khiang-Wee Lim, and Liu Jun, "Comparative Study of PCA Approaches in Process Monitoring and Fault Detection", Proc. of the 30th Annual Conference of the IEEE Industrial Electronics Society, vol. 3, pp. 2594-2599, 2004.

[2] G.O. Beale and J.H. Kim, "Detection and Classification of Faults Affecting Maneuverability of Underwater Vehicles", Proc. of the 29th Annual Conference of the IEEE Industrial Electronics Society, vol. 1, pp. 261-267, 2003.

[3] Jie Zhang, Elaine Martin, and A. Julian Morris, "Fault Detection and Classification through Multivariate Statistical Techniques", Proceedings of the American Control Conference, vol.1, pp. 751-755 vol. 1, 1995.

[4] E.B. Martin, A.J. Morris, J. Zhang, "Process performance monitoring using multivariate statistical process control", IEE Proceedings on Control Theory and Applications, vol. 143 issue 2, pp. 132-144, 1996

[5] Feng Wang, Junyi Cao, Binggang Cao, "Nonlinear Feature Fusion Scheme Based on Kernel PCA for Machine Condition Monitoring", Proc. of the International Conference on Mechatronics and Automation, pp. 624-629, Aug 5-8, 2007.

[6] Feng wang, Bo cheng, Binggang cao, "Machine condition monitoring by nonlinear feature fusion based on kernel principal component analysis with genetic algorithm", Proc. of the Third International Conference on Natural Computation, vol. 2, pp. 665-670, Aug 24-27, 2007.

[7] A. Lachouri, K. Baiche, R. Djeghader, N. Doghmane, and S. Ouhtati, "Analyze and Fault Diagnosis by Multi-scale PCA", Proc. of the 3rd International Conference on Information and Communication Technologies: From Theory to Applications, pp.1-6, April 7-11 2008.

[8] C. C. Luh and W. C. Cheng, "Non-linear system identification using an artificial immune system", *Proc. Instn. Mech. Engrs.*, vol. 215, part I, pp. 569- 585, 2001.

[9] D. Dasgupta, *Artificial Immune Systems and Their Applications*, Springer, NY, 1998.

[10] J. Timmis, "Artifical immune systems- today and tomorrow", *Nat. Comput.*, vol. 6, pp.1-19, 2007.

[11] D. Dasgupta," Artificial neural networks and artificial immune systems: similarities and differences, *Proc. of the IEEE International Conference on Systems, Man, and Cybernetics*, Orlando, FL, USA, vol. 1,pp.873-878, 1997.

[12] D. Dasgupta, "Advances in artificial immune systems", *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 40-49, 2006.

[13] J. Gomez, F. Gonzalez, and D. Dasgupta, "An immuno-fuzzy approach to anomaly detection", *The 12th IEEE International Conference on Fuzzy Systems (FUZZ)*, pp. 1219-1224, 2003.

[14] C. C. Luh and W. C. Cheng, " Immune model-based fault diagnosis", *Mathematics and Computers in Simulation*, vol. 67, pp. 515-539, 2005.

[15] L. Shulin, Z. Jiazhong, S. Wengang, and H. Wenhu, "Negative-selection algorithm based approach for fault diagnosis of rotary machinery", *Proc. of the American Control Conference*, Anchorage, AK, USA, pp. 3955-3960, 2002.

[16] I. Aydin, M. Karakose, and E. Akin, "Artificial immune based support vector machine algorithm for fault diagnosis of induction motors", *International Aegean Conference on Electrical Machines and Power Electronics (ACEMP)*, pp. 217-221, 2007.

[17] P. J. Costa Branco, J. A. Dente, and R. Vilela Mendes,"Using Immunology Principles for Fault Detection", *IEEE Trans. on Industrial Electronics*, vol. 50, no. 2, pp.362-373, 2003.

[18] R. R. Sumar, A. A. R. Coelho, and L. dos S. Coelho,"Use of an artificial immune network optimization approach to tune the parameters of a discrete variable structure controller", *Expert Systems with Applications*, vol. 36, pp. 5009–5015, 2009.

PAPER

1. PCA-Based Fault Isolation and Prognosis with Application to Water Pump[1]

Gary R. Halligan and S. Jagannathan

Abstract *— In this paper, the use of Linear and Kernel PCA for fault isolation and prognosis is explored in contrast to the traditional usage of PCA for detection and isolation. Vector projection and statistical analysis were utilized to isolate and predict faults in the PCA domain. Linear and Kernel PCA were applied to data collected from experiments on a one-half horsepower centrifugal water pump both for normal and faulty operation consisting of the four fault scenarios: Impeller failure, seal failure, inlet pressure sensor failure, and a filter clog. Upon close observation of the behavior of the principal component scores, it was determined that the linear PCA does not adequately isolate and predict the failures due to the nonlinear nature of the pump and the inherent linearity assumption of Linear PCA. Therefore, Kernel PCA, utilizing a Gaussian kernel was applied to the same data sets. Analysis of the behavior shows that the principal component scores gained from the Kernel PCA were more suitable than linear PCA.*

Keywords*:* Principal Component Analysis, Fault Isolation, Fault Prognosis, Rotating Machinery

[1]This work was supported in part by NSF I/UCRC Intelligent Maintenance Systems Center grant.

Gary Halligan is with the Department of Computer and Electrical Engineering, Missouri University of Science and Technology, Rolla, MO 65409 USA (phone: 573-341-6405; e-mail: halligan@mst.edu).

S. Jagannathan is with the Department of Computer and Electrical Engineering, Missouri University of Science and Technology, Rolla, MO 65409 USA (email: sarangap@mst.edu).

I. Introduction

Principal Component Analysis (PCA) is a multivariate statistical analysis technique which is very commonly used for dimension reduction, fault detection and isolation. PCA constructs a model from the eigenvalues and eigenvectors of the covariance matrix of the data to be analyzed. Dimension reduction is achieved by having the user define the minimum percentage of variance the PCA model must account for and retaining as few eigenvectors, also called PCs, as possible while still reaching the threshold. The fault detection and diagnosis comes from observing the relationships between the variables and monitoring any changes in these relationships due to the occurrence of faults in the system.

There are many types of PCA including: linear, or conventional, Moving PCA [1], Adaptive PCA [1], Exponentially Weighted PCA [1], and kernel PCA [2]. As with any analysis technique, each of these techniques have advantages and disadvantages, many of which are discussed at length in [1]. While conventional PCA is well suited for application to a linear system, its greatest weaknesses are assumptions that the data is of a Gaussian distribution and relationships among the variables are linear [3] in nature. As a result, the conventional PCA is ill-suited for processes with multiple operation modes and non-linear systems. An additional weakness of conventional PCA is its inability to recognize outliers [4]. In short, Moving PCA, Adaptive PCA and Exponentially Weighted PCA are all less sensitive to outliers than linear PCA and are better suited for applications with multiple operating conditions provided the additional computational complexity can be accommodated [1]. In [5], the authors suggest that faults based on the behavior of the PCs can be isolated by identifying the out of control PCs. The isolation

mentioned in [5] requires observing the changes in the variables feeding the out of control PCs and combining that information with the expertise of the operator to relate the observed behavior to a given fault.

On the other hand, Wang et al. [6], aimed at detecting the failure of a rolling bearing on an electric motor. The measurements taken on the test apparatus were vibration of the bearing housing and acoustic signal to monitor the noise generated by the apparatus. In addition to these readings, eight additional non-dimensional statistical features were extracted. These statistical features consisted of: Shape, Kurtosis, Crest, Skewness, Impulse, Second-order moment, Clearance, and K Factor. Both linear (conventional) and kernel PCA, utilizing a Gaussian kernel, were applied to the data and the results were compared. Additionally, both variations of PCA were applied to just the vibration and acoustic signals, and again to the vibration and acoustic signals including the eight statistical features. It was found that due to the nonlinear nature of the machinery in the experiment, the kernel PCA performed better than the linear PCA both with and without the statistical features. It was also found that the analysis which included the statistical features performed better than without the statistical features.

In Wang et al. [2], the aim was again to detect the failure of rolling bearings on an electric motor. The acoustic and vibration signals were recorded and the same eight statistical features were calculated. In contrast to [6], a genetic algorithm was included in the analysis. Both linear and kernel PCA were applied to the data with and without the genetic algorithm. It was found that incorporating the genetic algorithm with PCA improved the performance of both versions of PCA, under the same conditions, kernel PCA performed better than linear PCA.

Lachouri et. al [7] used a Daubechies-1 wavelets for analysis and processing of vibration signals to detect the failure of rolling bearings. The use of the wavelet was intended to decompose the signals into approximations. Then, multi-scale PCA was applied to each of the resulting matrices. The authors in [7] concluded that the results obtained were satisfactory and the method ensured a good and accurate diagnosis. However, this method is more mathematically complex than the method proposed in this paper. In this paper, PCA is applied directly to the data and the FDP determination was applied to the resulting PC scores.

To the best knowledge of the author's knowledge, there has not been any work done in extending PCA into the area of fault prognosis. In this paper, the proposed methodology uses only the PC scores and basic regression algorithms to isolate and predict faults. Once the PCA components and associated thresholds are identified offline, the approach can be implemented in real-time to detect abnormal data and determine which known fault is the most likely to be occurring. Once the type of fault has been identified, the time to failure can then be estimated online using least squares regression from a combination of appropriate PCA components.

In addition past literature [1-7] indicated that PCA was typically applied to processes to aid in the detection, and rarely applied to individual machines. In the few cases of the application of PCA to individual pieces of machinery [2, 6, 7], PCA was used in conjunction with other approaches such as wavelets, or genetic algorithms, or by simply introducing statistical features for instance: kurtosis and skewness. In contrast to the proposed approach applied the PCA directly, and without utilizing any other schemes,

to data from a single piece of rotating machinery, a centrifugal water pump. The isolation and prognosis are verified using experimental data.

Therefore the main contributions of this paper include the extension of the PCA-based schemes for fault isolation and prognosis, and the application of the PCA-based detection, isolation and prognosis scheme onto a rotating machine.

The remainder of the paper is organized as follows: Section II presents the methodology of both linear and kernel PCA, Section III discusses a case study involving a centrifugal water pump. Conclusions are given in Section IV, and finally the references are included in Section V.

II. PCA Methodology

A. Linear PCA

The first step in PCA is the normalization of the data. Standardization simply means that the data is scaled so that it has a mean of zero and a variance of one. The standardization is defined as

$$X = (X_0 - \alpha U)S^{-1} \tag{1}$$

where X_0 is a M by N matrix of raw data, α is an M x 1 matrix with all entries equal to one, U is a 1 x N matrix with each column entry being the mean of the corresponding column in X_0 and S is an N x N diagonal matrix containing the standard deviations of the columns in X_0. Since S is a diagonal matrix, S^{-1} is a diagonal matrix with the inverse of the standard deviations. PCA calculates principal components, or PCs, that are thought to contain the essence of the data. These PCs describe varying amounts of variance observed in the data. The dimension reduction feature mentioned earlier comes into play

when the user wishes to account for only a percentage of the variance. The first step in the calculation of the PCs is the calculation the covariance matrix, *C,* of the data matrix X which is given by

$$C = (M-1)^{-1} X^T X \tag{2}$$

The PCs mentioned above are the eigenvectors associated with the eigenvalues of the covariance matrix. The eigenvalues are equal to the variance explained by each of the PCs [8]. This means that the relative magnitudes of the eigenvalues are a measure of the relative importance of the corresponding PCs. In other words the PCs associated with larger eigenvalues are more important, for dimension reduction applications, than the PCs associated with smaller eigenvalues [9]. In order to determine which PCs must be retained for dimension reduction approaches, the user must establish a threshold for the minimum amount of variance that the model must account for. Next, the eigenvalues, λ's, must be sorted in the descending order: $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N$. The PCs to be retained can be determined by utilizing

$$\frac{\left(\sum_{i=1}^{j} \lambda_i\right)}{\left(\sum_{k=1}^{N} \lambda_k\right)} * 100 \geq thc\,, \;\; j < N \tag{3}$$

where *thc* is a user defined threshold expressed as a percentage. Now the user should retain only the PCs which were associated in the first term of (3). By reordering the eignenvalues, it is guaranteed that the minimum number of PCs are retained while still reaching the minimum variance threshold.

The retained PCs, expressed as column vectors, are then placed into a principal component loading matrix *P*. The standardized data is then mapped into a new

coordinate system which is defined by the PCs [8] by multiplying the standardized data matrix, X, with the principal component loading matrix P. The resulting matrix, T, is called the principal component score matrix. The PC score matrix can be calculated using

$$T = XP \tag{4}$$

PCA reduces the number of data dimensions through the construction of orthogonal principal components that are weighted, linear combinations of the original variables as

$$X = \sum_{j=1}^{a} t_j p_j^T + E \tag{5}$$

where p_j is the jth vector in the principal component loading matrix, P, t_j is the jth vector in the principal component score matrix T, E is the residual error, and a is the number of principal components retained [1, 2].

From (5) it is clear that the error and the number of PCs retained have an inverse relationship. In other words, as the number of retained PCs increases, the error will decrease. However, eliminating even one PC will introduce errors. Therefore, the only way to not have any residual error is to retain all of the PCs. When PCA is being used for dimension reduction, retaining all of the PCs would defeat the point of doing the PCA. However, for the methodology proposed in this paper, all of the PCs will be retained. It will be shown in Section III that retaining all PCs aids fault isolation.

The PCA is now complete, as the high dimensional data has been projected onto a lower dimensional subspace. However, the model established by PCA only yields a framework in which data samples can be compared. There are many ways in which the

comparison can be carried out. However, Hotelling's T^2 and the squared prediction error Q statistics are typical [1] metrics. The Hotelling's metric is defined by

$$T^2 = (x_i - \bar{x})C^{-1}(x_i - \bar{x})^T \quad (6)$$

where $\bar{x}$ is the sample mean.

The Squared Prediction Error (SPE) for the ith sample data x_i is defined as

$$SPE_i = e_i e_i^T \quad (7)$$

where $e_i = (X_i - \sum_{j=1}^{k} t_j p_j^T)$ and k is the number of retained PCs.

These two quantities, Hotelling's T^2 and SPE, are widely used to detect and diagnose the malfunctions. Both statistics observe at the relationships between the variables and detect any changes. If the T^2 statistic of a data entry exceeds the user defined maximum threshold, then there is an increased chance that at least one fault is present. Unfortunately, by setting a threshold the user must determine what level of false alarms or missed detections is acceptable [5]. If the level is set too high, there will be instances when a fault is present without crossing the threshold. Conversely, if the threshold is set too low, the threshold may be crossed even when no fault is present in the system rendering false alarms.

From the discussion above it is clear that for each PC the variables with the largest coefficients, or loadings, have the most influence where the score will be plotted. Additionally, the PCs which most clearly demonstrate abnormal behavior likely contain information indicating the reason why the nonlinear system has left the normal operation

region. By observing the abnormal PCs and utilizing prior knowledge of the system, an operator may be able to relate the changes to a particular malfunction [5].

When the number of retained PCs is relatively small, the PC plots can be used as visual aids. If the process variables are highly correlated, a fault would likely cause the process measurements to move in a specific direction. This movement of PCs in a specific direction would aid fault classification or isolation [4].

Although the PC/SPE chart can be helpful for detecting abnormal operation, such charts are not always able to give exact causes for the deviations [4, 10]. In short, traditional PCA is useful for fault detection when the relationships among variables are linear in nature but not for fault diagnosis. Additionally, linear PCA is ill-suited for application to non-linear systems as it cannot adequately model nonlinear dynamics.

B. Kernel PCA

Kernel PCA is a nonlinear generalization of PCA. Kernel PCA maps the data into a higher dimensional space using nonlinear mapping and computing dot products in the feature space. The kernel PCA algorithm has two main steps. The first step is to linearize the data utilizing the kernel for nonlinear mapping from the input space to a higher-dimensional feature space [9]. The mapping Φ is defined implicitly and takes the form of the dot product mentioned earlier [9]. The dot product, defined in terms of the kernel function is given by

$$K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$$

The most commonly used kernels are Gaussian kernels of the form demonstrated by

$$K(x, y) = \exp\left(- \|x - y\|^2 / (2\sigma^2) \right) \tag{8}$$

If the nonlinear function mappings of the input variables to the PC domain are unknown, a Gaussian kernel is the logical choice due to the well-known nonlinear approximation properties of the Gaussian functions. If the mappings are known, however, another kernel that more closely matches the nonlinear function may be preferable. The mapped data set in the feature space is given by $\Phi = \{\Phi_i : i = 1, \cdots, n\}$, where $\Phi_i = \Phi(x_i)$. The second step of kernel PCA is to apply PCA on the centered mapped data given by

$$\tilde{\Phi} = (\Phi - \gamma U) = \Phi(x_i) - \left(\frac{1}{n}\right)\sum_{i=1}^{n} \Phi(x_i) \tag{9}$$

where γ is an M x 1 matrix of ones, *U* is a 1 x N matrix with each column entry being the mean of the appropriate column in Φ. Then apply PCA by calculating the covariance matrix of the centered mapped data mapped data given by

$$C = \frac{1}{n-1}\tilde{\Phi}^T \bullet \tilde{\Phi} = \frac{1}{n-1}\sum_{i=1}^{n} \tilde{\Phi}(x_i)^T \tilde{\Phi}(x_i) \tag{9}$$

where *n* is the number of dimensions of the feature space and performing the eigen decomposition. The eigenvectors, ν, of C are given by

$$\mu\nu = C\nu \tag{10}$$

where μ is the matrix of eigenvalues and *C* is the covariance matrix from (9). Rearranging (10) and substituting the third term of (9) into (10) yields

$$\nu = \sum_{i=1}^{n} \tilde{\Phi}(x_i)\left(\frac{1}{\mu(n-1)}\tilde{\Phi}(x_i)^T \nu\right) = \sum_{i=1}^{n} \alpha_i \tilde{\Phi}(x_i) \tag{11}$$

where $\alpha_i = \frac{\left(\tilde{\Phi}(x_i)\nu\right)}{(\mu(n-1))}$. Taking the dot product with ν results in

$$\tilde{\Phi}(x_j) \cdot \nu = \sum_{i=1}^{n} \alpha_i \tilde{\Phi}(x_i) \cdot \tilde{\Phi}(x_j) = \sum_{i=1}^{n} \alpha_i \tilde{K}_{ij} \tag{12}$$

Rearranging the definition for α_i from (11) and utilizing the first and third terms in (12), implies that

$$\mu(n-1)\alpha_j = \sum_{i=1}^{n} \alpha_i \tilde{K}_{ij} \tag{13}$$

and therefore

$$\tilde{K}\alpha = \tilde{\mu}\alpha \tag{14}$$

where $\alpha = (\alpha_1, \ldots, \alpha_n)$ and $\tilde{\mu} = \mu(n-1)$. In other words, α, is an eigenvector of $\tilde{K}$. The derivation in this section is based on the work in [9].

The user again defines a threshold for percentage of variance that must be accounted for by the PCA model. As before, the equation (3) will determine the PCs necessary to reach the threshold. Retaining only those PCs necessary to reach the threshold will reduce the dimensionality. The data is then multiplied by the retained PCs to project the data into the PC domain. However, it is important to note that not much work on isolation and prognosis was accomplished using PCs. The fault isolation and prognosis schemes proposed next can then be applied to the resulting PC scores.

C. Isolation and Prognosis

For both the linear and kernel PCA methodologies described in the previous section, data from the normal operation of the nonlinear system, for instance a pump, was utilized to establish the PCA model.

The PC scores, i.e. the projection into the PC domain, of data from four different fault types (e.g. impeller failure, seal failure, inlet pressure sensor failure, and a filter clog) were calculated. Additionally, the PC scores were plotted for the data transitioning

from normal operation to impeller failure. The first step in prognosis is the isolation of the fault. To achieve fault isolation, each data point of unknown classification, represented by U in Figure 1, was compared to the 95% confidence interval of the normal data, represented by N. If the point falls within the confidence interval, the point is then assumed to be normal.

However, if the point did not fall into the 95% confidence interval of the normal data the point was then compared to each of the known fault classes, represented by F1, F2, and F3. This was done by calculating the angle between the point to be classified and the mean vector of each of the known fault classes, θ. The smaller the angle is, the more aligned the data point is with the given fault classification.

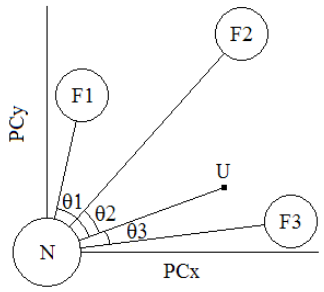

Figure 1: Illustration of the Angle Based Isolation

The values of the angles between the point and each fault class were then compared. To avoid class assignment due to insufficient knowledge, this angle calculation and comparison was done for each of the PC pairings, represented by PCx and PCy in Figure 1. The data point was then assigned to the fault class which contained the most minima. The assumption made here is that the data which is transitioning from normal operation to a given fault will follow a roughly linear trajectory as illustrated by

Figure 2. This assignment merely asserts that, of the known faults, the system is most likely trending towards the assigned fault. It is worth noting here that all of the analysis is being applied directly to the PC scores without any SVMs, NNs, etc., which is typical in the current/past literature. Consequently, the complexity of the proposed scheme is low.

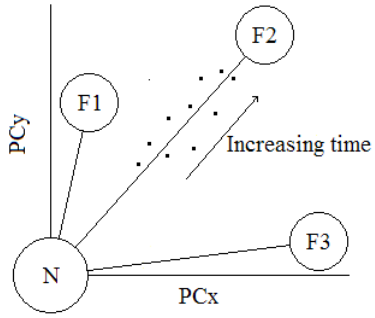

Figure 2: Illustration of the Prognosis Assumption

The basis which was used to calculate the Estimated Time to Failure (ETTF) for linear and kernel PCA was the product of PC1 and PC3, and the product of PC2 and PC4, respectively. Both products correspond to the passivity of the system. It is worth noting that the passivity [11] is the time derivative of the stored energy of the system as a function of time. In a stable system, the derivative should be decreasing over time. Conversely, for an unstable system the derivative should increase over time.

While the PC scores do not have any units, the physical meaning of the scores may be inferred by analyzing the units of the inputs with the largest loading. The failure threshold, i.e. the threshold for determining the presence of the failure, was assigned to be slightly larger than the maximum value observed in the transition data. The data during the seeding of the fault changes abruptly over time. This makes fault prognosis nearly

impossible. Therefore, the following analysis was conducted on data after the fault was seeded in order to determine the time to failure.

Next, for each data point a parabolic data trajectory expressed in (15) was calculated using least squares regression by utilizing the current data point and all previous points.

$$y = at^2 + bt + c \tag{15}$$

Solving (15) for *t* yields the well known quadratic formula:

$$t_{failure} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{16}$$

Solving (16) will yield two values for $t_{failure}$. The larger of the two values yielded by (16) is time index for the occurrence of failure. ETTF can be found by simply subtracting the index for the current sample from $t_{failure}$ as

$$t_{remaining} = t_{failure} - t_{current} \tag{17}$$

III. Case Study

Both the linear and kernel PCA were applied to data from the centrifugal water pump shown in Figure 3. The data consisted of: 125 hours of normal operation, 40 hours of impeller failure, 20 hours of seal failure, 78 hours of filter clog, 9 hours of impeller fault seeding, and six instances of inlet pressure sensor failure (about one hour each).

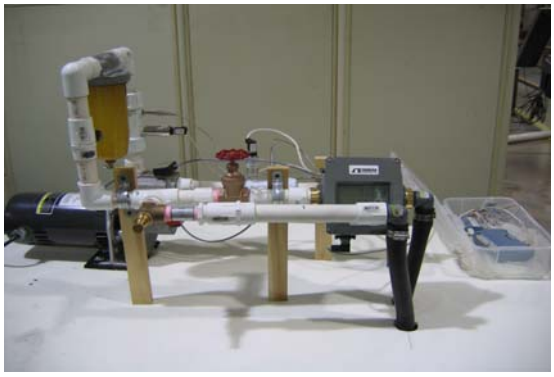

Figure 3: Centrifugal Water Pump Test Bed

A. Linear PCA

The result of plotting the normal, fault, and fault seeding data with respect to PC1 and PC2 is shown in Figure 4. The assumed linear trajectory the data would take towards each fault is also included in Figure 4. The sensor failures shown in plots throughout the remainder of this paper are multiple failures of the same sensor. Additionally, between the first and second instances of sensor failure, modifications were made to the system to allow the pump to operate in two operation modes. Previously, the pump had only one operation mode. The modifications resulted in the difference in location of the data/PC scores for the first sensor failure and the additional sensor failures. Also, the pump was toggling between the two possible operation modes, thus resulting in two clusters of data points.

Figure 4 illustrates that the PCA model can distinguish between each classification of data. However, the prognosis based solely on the PC1 and PC2 scores would be problematic since the data for the seal failure and sensor failures lay close to the trajectory for the impeller failure. For example, the difference in trajectory between first sensor failure and impeller failure is approximately one degree.

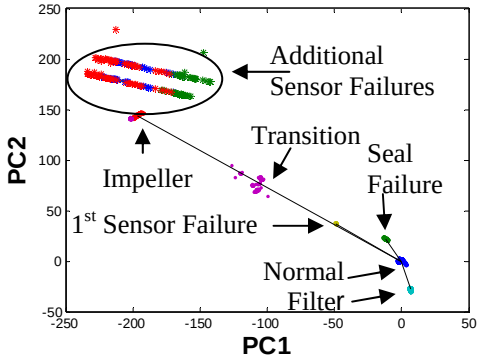

Figure 4: Score Plots for PC1 and PC2 with mean vector

Analyzing the loadings of the different sensors in Table 1, the inlet pressure sensor has the largest loading in PC1. Additionally, by looking at the readings of each sensor over time it was determined that an impeller failure manifests itself in a reduction of inlet pressure. This explains how the two failures could lie roughly on the same trajectory with respect to the origin.

Table 1: Linear Principal Component Coefficients

	PC1	PC2	PC3	PC4
Inlet Pressure	0.6393	-0.1930	0.2259	-0.7092
Outlet Pressure	-0.4434	-0.4359	0.7825	-0.0318
Lateral Accel	-0.5796	0.4388	-0.1115	-0.6775
Vertical Accel	0.2422	0.7617	0.5694	0.1924

By comparison, Figure 5 shows the data plotted with respect to PC1 and PC3. In this case, the data classes have significantly larger angles between them. This increase in the angular distance facilitates the fault isolation.

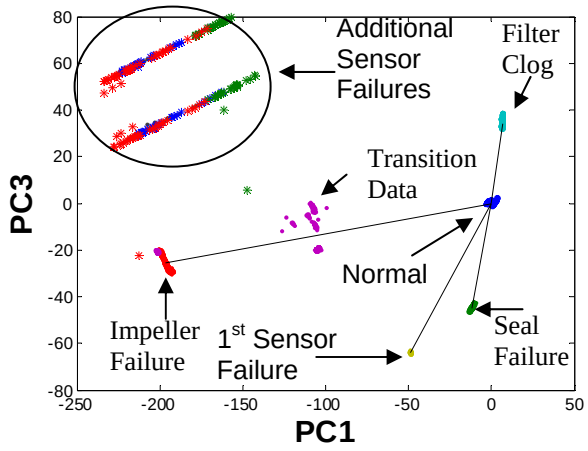

Figure 5: Score Plots for PC1 and PC3 with Mean Vector

To verify the methodology for the fault isolation, the second instance of the inlet sensor failure, the first instance in which the pump had multiple operation modes, was used as a reference for the other four instances of the inlet pressure sensor failure. When the angle based isolation algorithm was applied, it correctly classified all of the data points from the four sensor failures. The results of the isolation are expressed in Table 2.

Table 2: Results of Angle Based Isolation

Sensor Failure	Number of Data Points	Classified as Sensor Failure	Percent Accuracy
Failure 3	72	72	100
Failure 4	59	59	100
Failure 5	58	58	100
Failure 6	73	73	100

As stated earlier, the product of PC1 and PC3, or the passivity, of the system was utilized for the prognosis of the impeller failure. From Table 1, the inputs with the

largest loading in PC1 and PC3 in the linear case are the inlet and outlet pressures respectively.

The normalized product is shown in Figure 6. Since the system is failing, an upward trend is expected. Instead the analysis shown in Figure 6 yields a downward trend. A pump is not a linear system and the downward trend is an indication that the nonlinear dynamics that are not adequately accounted for in the model established by linear PCA. Therefore, applying linear PCA to such a system will result in large errors in the results. Hence, kernel PCA is applied. The results of the two methodologies are contrasted in the subsequent section.

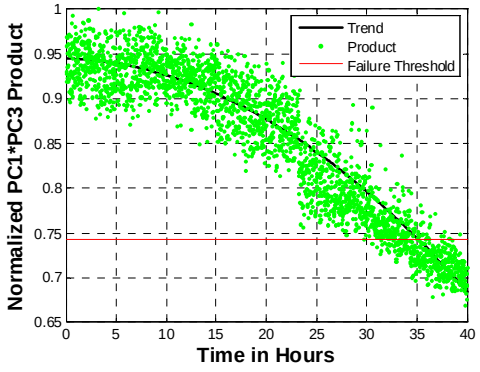

Figure 6: Normalized Product of PC1 and PC3 Following Failure Seeding

Figure 7 shows the normalized PC1 times PC3 product averaged over a sliding window of 30 minutes. The plot in Figure 8 shows the ETTF based on the averaged product from Figure 7. Initially, the prognosis attempts struggle due to the limited number of data points and weak downward trend. At approximately 15 hours the downward trend becomes more pronounced since the fault progressed. At that time the trending yields a more accurate time to failure prediction.

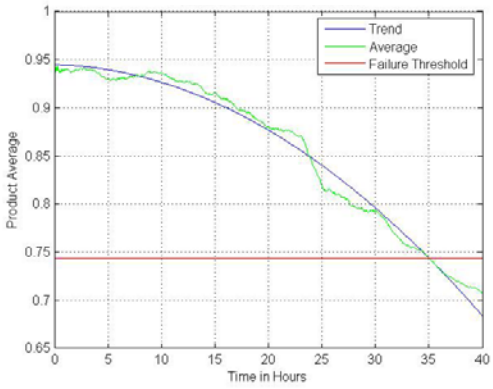

Figure 7: Product of PC1 and PC3 Averaged over a 0.5 Hour Sliding Window

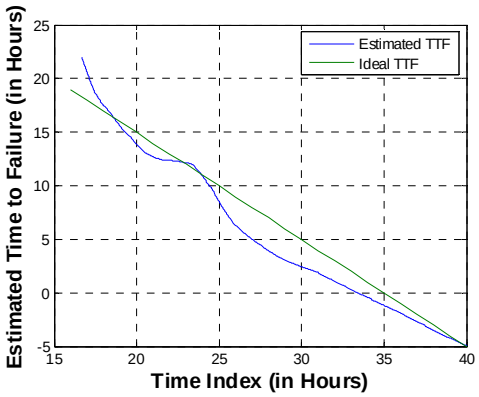

Figure 8: ETTF vs. Time

For the time between 16 hours and 40 hours the ETTF follows the ideal time to failure. The maximum error in the ETTF after the 16 hour mark is approximately three hours occurring at the time of 28 hours. Additionally, the ETTF crosses zero at about 33 hours. As mentioned earlier, the system has nonlinear dynamics which are not adequately accounted for in the model established from linear PCA. Hence, the large error in ETTF is observed.

B. Kernel PCA

In this section, the test bed data is analyzed using kernel PCA. A Gaussian kernel was selected due to its generic nature. Figures 9 and 10 respectively show PC1 and PC2 plotted with respect to PC4. By observing the leading terms of the PCs in Table 1 and Table 3, it can be seen that Figures 9 and 10 are effective plotting kernel PCA equivalent of the PCs shown in Figures 4 and 5.

Table 3: Kernel Principal Component Coefficients

	PC1	PC2	PC3	PC4
Inlet Pressure	0.2747	-0.6060	0.1926	-0.7213
Outlet Pressure	-0.009	-0.7747	-0.0951	0.6251
Lateral Accel	0.0120	-0.0518	-0.9757	-0.2125
Vertical Accel	0.9615	0.1730	-0.0429	0.2093

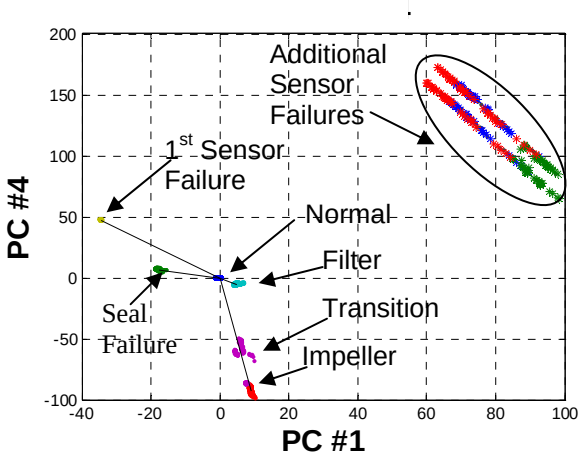

Figure 9: Score Plots for PC1 and PC4 with Mean Vector

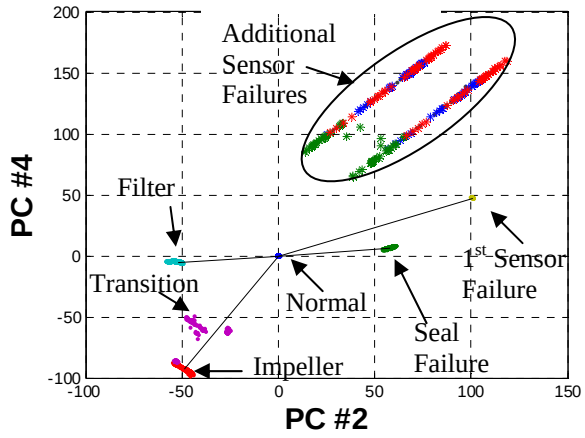

Figure 10: Score Plots for PC2 and PC4 with Mean Vector

In both plots, the faults were better separated from each other in comparison to when the linear PCA was utilized since the kernel PCA included the nonlinear dynamics.

In the linear case, the first instance of inlet pressure sensor failure was used as a reference for the other instances. The isolation algorithm was applied to the PC scores resulting from the kernel PCA. The end result of the isolation algorithm was in agreement with the results from earlier, in Table 2.

Table 4: Results of Angle Based Isolation

Sensor Failure	Number of Data Points	Classified as Sensor Failure	Percent Accuracy
Failure 3	72	72	100
Failure 4	59	59	100
Failure 5	58	58	100
Failure 6	73	73	100

In the linear case, the passivity was used for the prognosis of the impeller failure. One noticeable difference is the upward trend over time of the product. This is

more in line with expectations and indicates that the use of kernel PCA is preferable to the use of linear PCA. Following the methodology from subsection III.A, the prognosis results are obtained and shown in Figures 11 and 12.

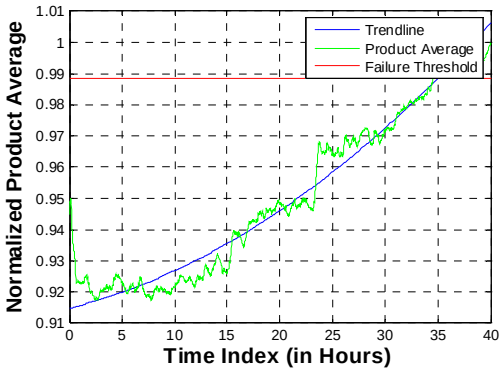

Figure 11: Product of PC2 and PC4 Averaged over a 0.5 Hour Sliding Window

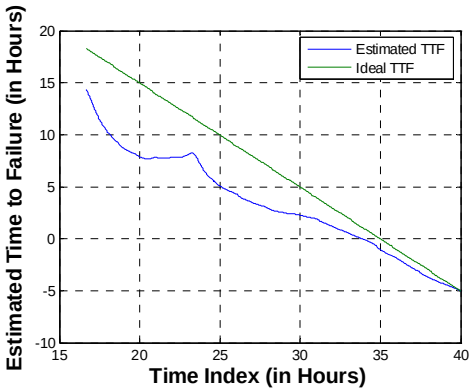

Figure 12: ETTF vs. Time

While the general shape of the ETTF is similar to that of the linear case, before approximately 30 hours, the linear PCA ETTF approximation performed better than the kernel PCA approximation. In the linear case the maximum error in estimation was about 3 hours at time of 28 hours. In the kernel PCA case, the maximum error is 7.2

hours and it occurs at 20 hours. However, after reaching the 30 hour point, the approximation from the kernel PCA was closer to the "ideal" time to failure line than the linear PCA approximation. The actual failure occurred at 35 hours while the approximation of the ETTF, from the linear PCA and kernel PCA cross zero at 33 and 34 hours, respectively.

IV. Conclusions

This paper introduces a new PCA-based method for fault prognosis while PCA is typically used for dimension reduction, fault detection and isolation. The water pump's nonlinear dynamics are not adequately modeled by linear PCA. By applying the Gaussian kernel, the fault isolation and prognosis results were improved by approximately 10 percent for the proposed case study of a water pump.

V. References

[1] Doan X. Tien, Khiang-Wee Lim, and Liu Jun, "Comparative Study of PCA Approaches in Process Monitoring and Fault Detection", Proc. of the 30th Annual Conference of the IEEE Industrial Electronics Society, vol. 3, pp. 2594-2599, 2004.

[2] Feng wang, Bo cheng, Binggang cao, "Machine condition monitoring by nonlinear feature fusion based on kernel principal component analysis with genetic algorithm", Proc. of the Third International Conference on Natural Computation, vol. 2, pp. 665-670, Aug 24-27, 2007.

[3] G.O. Beale and J.H. Kim, "Detection and Classification of Faults Affecting Maneuverability of Underwater Vehicles", Proc. of the 29th Annual Conference of the IEEE Industrial Electronics Society, vol. 1, pp. 261-267, 2003.

[4] Jie Zhang, Elaine Martin, and A. Julian Morris, "Fault Detection and Classification through Multivariate Statistical Techniques", Proceedings of the American Control Conference, vol.1, pp. 751-755 vol. 1, 1995.

[5] E.B. Martin, A.J. Morris, J. Zhang, "Process performance monitoring using multivariate statistical process control", IEE Proceedings on Control Theory and Applications, vol. 143 issue 2, pp. 132-144, 1996

[6] Feng Wang, Junyi Cao, Binggang Cao, "Nonlinear Feature Fusion Scheme Based on Kernel PCA for Machine Condition Monitoring", Proc. of the International Conference on Mechatronics and Automation, pp. 624-629, Aug 5-8, 2007.

[7] A. Lachouri, K. Baiche, R. Djeghader, N. Doghmane, and S. Ouhtati, "Analyze and Fault Diagnosis by Multi-scale PCA", Proc. of the 3rd International Conference on Information and Communication Technologies: From Theory to Applications, pp.1-6, April 7-11 2008.

[8] Balaje Thumati, Neha Bassi, Jeffery Birt, and Jag Sarangapani, "A Neural Network Model Based Approach to Detect Seal and Impeller Failures in Centrifugal Pumps", IMS 2.1 Project *Report 2*, June 2007.

[9] George Stefatos and Ben Hamza, "Statistical Process Control Using Kernel PCA", Proc. of the Mediterranean Conference on Control and Automation, pp.1-6, 2007.

[10] Yang Yinghua, Lu Ningyun, Wang Fuli, MA Liling. "A new fault detection and diagnosis method based on principal component analysis in multivariate continuous processes", in Proceedings of the 4th World Congress on Intelligent Control and Automation, vol. 4, pp.3156-3160, 2002.

[11] F.Lewis, S. Jagannathan, A. Yesilderek, "Neural network control of robot manipulators and nonlinear systems", Taylor and Francis, UK, 1999

PAPER

2. A Novel Fault Detection and Prediction Scheme in Discrete-time Using a Nonlinear Observer and Artificial Immune System as an Online Approximator

Gary R. Halligan, Balaje T. Thumati, and S. Jagannathan

Abstract — In this paper, an observer-based fault detection and prediction (FDP) scheme using artificial immune system (AIS) as an online approximator is introduced for a class of nonlinear discrete-time systems. Traditionally, AIS was considered as an offline tool for fault detection in an ad hoc manner. However, in this paper, the AIS utilized as an online approximator in discrete-time (OLAD) is considered while its parameters are tuned online. A nonlinear observer comprising of the AIS and a robust adaptive term is used for detecting faults in the given nonlinear system. A fault is detected by comparing the residual against *apriori* chosen threshold, which is obtained by comparing the output of the nonlinear estimator to that of the given system. Upon detection, the AIS and the robust adaptive term are initiated in the observer, where the AIS parameters are tuned online using a suitable update law for learning the unknown fault dynamics. Additionally, this update law is used to estimate the time-to-failure (TTF), which is considered as a first step for prognostics On the other hand, the robust term, which is a function of the AIS parameter vector, is used to deliver asymptotic convergence of the residual unlike bounded

Gary R. Halligan (halligan@mst.edu), Balaje T. Thumati (e-mail: bttr74@mst.edu), and S. Jagannathan (e-mail: sarangap@mst.edu) are with the Department of Electrical & Computer Engineering, Missouri University of Science and Technology (formerly University of Missouri–Rolla), Rolla, MO, 65401, USA. Research supported in part by NSF I/UCRC on Intelligent Maintenance Systems and Intelligent Systems Center.

stability in other schemes. The performance of the proposed FDP scheme is first demonstrated on a two-link robot arm and an axial piston pump in simulation and subsequently on an axial piston pump test bed.

I. INTRODUCTION

Modern engineering systems require early fault detection and warning system to render safe and reliable service. Therefore, numerous efforts have been under taken in addressing the problem of fault detection and prediction (FDP). Due to the presence of noise and system uncertainties, the problem of fault detection (FD) is complex thus requiring robustness. The commonly used FD methods include quantitative or model-based [1] and qualitative or data-driven based techniques [2]. The qualitative based techniques are found to be expensive [1] due to the need for large quantities of data and are dependent upon region of operation. However, quantitative methods require a suitable representation of the nonlinear discrete-time systems. Typically, an observer is utilized to represent the nonlinear system.

In the past literature, FD efforts are limited to linear systems [1-5], by using a sliding mode observer [3], geometric approach [4], and parity relations [2] etc. Typically, in the observer based approach, a residual is generated by comparing the observer output with that of the actual system. Moreover, a fault is detected by comparing the generated residual against *apriori* chosen threshold. However, selection of the threshold is a challenging task due to the presence of uncertainties, but an analytical procedure has been developed to identify thresholds [5] analytically.

In the recent years, with better understanding of nonlinear control system theory,

the techniques proposed for linear systems have been extended to nonlinear systems. Such schemes include the sliding mode observers [6], geometric approach [7], adaptive and diagonal observers [5, 8, 9] and so on. A recent survey on the various FD schemes for nonlinear systems can be found in [10]. Another aspect that is of interest to the FD community is the stability and the robustness of the FD schemes. Recently, various FD schemes [5- 9] have been proven to be stable. However, most of the developments are in continuous-time and not much has been accomplished in the discrete-time.

The time-to-failure determination (TTF) is another important feature, which in general is unavailable in the previously reported schemes [3-9], since TTF is the first step for prognostics assessment. Some TTF schemes like the data-driven approaches [11-13], assume a specific degradation model which has been found to be limited to the particular system or material type under consideration. Another scheme [14] employs a deterministic polynomial and a probabilistic method for prognosis by assuming that certain parameters are affected by the fault. While others [15] use a black box approach using neural network (NN) on the failure data. All these schemes [11-15] while being data-driven address only TTF prediction, require offline training and do not offer performance guarantees. Therefore, it is envisioned that a unified FDP scheme will be necessary to alert an impending failure and provide the remaining useful life.

Discrete-time development is important due to the stability problems incurred in the direct conversion of the continuous time FD schemes to discrete-time [16]. Recent developments in discrete-time include [16, 17], where a FD scheme is introduced by using the persistent of excitation (PE) condition. Since it is very difficult to verify or guarantee PE, in our previous work [18], a FD scheme using linearly parameterized online

approximators is introduced by relaxing the PE requirement. However, bounded stability of all the signals is demonstrated similar to the case of fault detection algorithms in continuous-time.

In contrast, in this paper a FDP scheme for nonlinear discrete-time systems with guarantees of asymptotic stability is introduced by using an observer. To best of our knowledge not many FDP schemes in discrete-time render asymptotic stability. However, in [19], asymptotic stability of a continuous time FD scheme for robotic systems with specific actuator faults is undertaken. The FD scheme proposed in this paper comprises of a nonlinear observer, which is used for detecting faults in the given system. Additionally, the nonlinear observer comprises of an online approximator in discrete-time (OLAD) and a robust adaptive term generated by the OLAD parameter vector. The OLAD and the robust adaptive term are initiated only after the detection of a fault. Moreover, a fault is detected by comparing the generated residual against *apriori* chosen threshold. The residual is generated by comparing the outputs of the nonlinear system with that of the observer. By using a suitable update law, the parameters of the OLAD are tuned online to learn the unknown fault dynamics. Additionally, the robust adaptive term is used to guarantee the asymptotic convergence of the residual and the parameter estimation errors after the occurrence of the fault and in the presence of the uncertainties.

Most of the previously proposed FD scheme [5, 9, 16-18] uses neural networks or fuzzy systems as online approximators. However, in this paper, we use an artificial immune system (AIS) as the OLAD since biological immune systems detect external virus and protect the human body. Conventionally, AIS has been considered as an offline tool for applications such as classification, pattern recognition and detection. Additionally, offline

data based training schemes are proposed to obtain AIS [20-30] parameters. However, in this paper a new online adaptive law is introduced for tuning the AIS parameter vector online while demonstrating that the AIS is an online approximator.

In general, AIS draws inspiration from the biological immune system. In the event of a disease causing antigen (such as virus, bacteria etc.) attacking the human body, the immune system detects the foreign bodies and responds to the antigen by releasing suitable antibodies. Based on the affinity between the released antibody and the antigen, the disease causing antigen is destroyed. Moreover, the immune system memorizes the type of antibodies utilized to kill the antigen, so that in future attacks it ensures a quick release of antibody to overcome the antigen. The inherent advantage of the immune system in detecting anomalies makes it as a natural candidate for system identification [21], FD [25-29] and control [30] when compared to neural networks (NNs) which are derived from neurological system.

However, existing AIS-based methods [22, 25-30] are data driven, ad hoc and require extensive offline training to tune the AIS parameter vector. Therefore, in this paper, AIS is used as an OLAD, which is a part of the nonlinear FD observer. Moreover, the AIS parameter vector is tuned online without any *apriori* offline training. Moreover, mathematically, the asymptotic convergence of the residual and the parameter estimation errors of the FD scheme after the occurrence of the fault is shown by using Lyapunov analysis.

Using the parameter update, mathematically a method is proposed to derive the TTF by projecting the current value of the parameter to its limit provided the limiting parameter value is defined by the designer. This process is iteratively performed to

continuously predict TTF up to the failure threshold beyond which the system is considered unsafe. For most practical systems, the unknown parameters could be tied to physical entities thus making the parameter-based TTF determination very useful. Alternatively, the state trajectories from the FD estimator can be utilized for TTF determination due to asymptotic convergence. Finally, simulation examples and experimental results are presented to show the performance of the proposed FDP scheme.

The important contribution of this paper is the asymptotic stability of the FD scheme for nonlinear discrete time systems using the robust adaptive term and the AIS as an OLAD. Addition of the robust adaptive term complicates the stability analysis whereas the Lyapunov proof is still offered. In addition, the time to failure determination is introduced by using the AIS parameter vector. Finally, the online fault detection and prediction is verified on an experimental test bed.

This paper is organized as follows: Section II provides background on the AIS. Section III introduces the system under investigation whereas Section IV explains the FD scheme and the stability analysis. Section V introduces the prediction scheme whereas Section VI provides simulation results and Section VII explains the experimental results. In Section VIII conclusions and future work are given.

II. Artificial Immune System as Function Approximators

In biological organisms, the function of the immune system is to protect the body from invasion by foreign objects, called antigens. This is done by lymphocytes, which comprises of the two main types of white blood cells: T-cells and B-cells. There are two classes of T-cells: killer T-cells and helper T-cells. When an infection is detected, the killer T-cells destroys the infected cells whereas the helper T-cells assist in engulfing and

destroying the invading pathogens. In addition, the helper T-cells stimulates B-cells to produce clones of antibodies to attack the pathogen. The B-cells fine tunes the antibodies to increase their affinities to the antigen being encountered. The higher the affinity is, the stronger the immune response will be. Additionally, more antibodies will be released to mitigate the antigen. Antibodies with highest affinity are retained while a feedback is provided to the T-cells to store in memory the type of antibody required for a particular antigen. This would help in mitigating future attacks by the similar antigen. Interested readers for further reading could refer to [20].

Based on this understanding, a recent work on AIS can be found in [21, 23] wherein the AIS is utilized to solve engineering problems. For instance, in [24], AIS is used for identification of nonlinear systems. In this method, an offline data based training scheme is proposed for the nonlinear system identification. However, an interesting contribution is the definition of a mathematical equation to describe the function of the AIS for system identification

$$\hat{f}(x_j) = \sum_{i=1}^{N} m_{ij} a_i = \frac{\sum_{i=1}^{N} a_i e^{-\beta_i d_{ij}}}{\sum_{i=1}^{N} e^{-\beta_i d_{ij}}} \tag{1}$$

where $i = 1, \ldots, N$ is the number of antibodies, $j = 1, \ldots, N_s$ is the number of data sets, m_{ij} is the ij^{th} affinity function, β_i is the shape parameter, a_i is the appropriate immune response, $d_{ij} = \|x_j - p_i\|$ is the Euclidean distance between the j^{th} antigen epitope vector (x_j) and the i^{th} antibody receptor vector p_i.

For engineering problems, the artificial immune response considers the unknown data as antigen, and the output is the net response of all the antibodies (i.e., output of

equation (1)). Therefore, by calculating the error between the estimated and the actual value, the parameters of the AIS function are updated. However, the AIS training is an iterative process and is performed offline. Therefore, in this paper, a new online tuning mechanism is proposed to tune the parameters of the AIS online by using adaptive control techniques. Moreover, we use the same mathematical equation as that given in [24] to describe the function of the AIS and exploit the function approximation property.

To guarantee that the AIS scheme could be utilized for approximating any unknown function over the compact set, in the following theorem we show that indeed an AIS possesses function approximation properties. This enables the AIS to be an OLAD similar to an artificial neural network, fuzzy logic and other online approximators. However, AIS is preferred for FD due to its natural affinity of detecting and preventing antigen attacks when compared to the other online approximators.

Theorem 1: For every continuous smooth function f , every AIS basis function ϕ, every probability measure σ , and every $n_a \geq 1$, there exists a linear combination of AIS functions $\hat{f}_a(x)$, such that

$$\sum_B \left(f(x) - \hat{f}_a(x) \right)^2 \sigma \leq \frac{(2C)^2}{n_a} \tag{2}$$

where $C > 0$, B is a compact set, and n_a is the number of antibodies or the size of the AIS function.

Proof: Follow steps similar to [31].

As shown in this theorem, the use of AIS in approximating unknown functions is valid. Therefore, similar to neural networks, the unknown function ($f(x)$) and the estimate of AIS could be written as

$$f(x) = \overline{a}^T \varphi(x) + \varepsilon(k),$$

$$\hat{f}_a(x) = \overline{a}^T(k)\varphi(x), \quad \hat{f}_a \in \Re^{n\times 1} \tag{3}$$

where $\overline{a} \in \Re^{l\times n}$ is the unknown ideal immune response, $\varepsilon(k)$ is the approximation error and bounded by a known constant, i.e., $\|\varepsilon(k)\| \le \varepsilon_a$. Also, $\overline{a}(k) \in \Re^{l\times n}$ is a matrix of estimated immune response, $\varphi(x) \in \Re^{l\times 1}$ is the basis function which is given by

$$\varphi(x) = \left[\frac{e^{-\beta_1 d_1}}{\sum_{i=1}^{l} e^{-\beta_i d_{ij}}}, ..., \frac{e^{-\beta_l d_l}}{\sum_{i=1}^{l} e^{-\beta_i d_{ij}}} \right]^T$$

where β_i , $\forall i = 1, ..., l$ is a positive randomly chosen shape parameter, $d_i = \|x - p_i\|$, where $x \in \Re^{n\times 1}$ is the input to the AIS basis function, $p_i \in \Re^{n\times 1}$, $\forall i = 1, ..., l$ is a randomly chosen constant vector. With this understanding on AIS, we next proceed with the discussion on the system under investigation.

III. Problem Statement

Consider the following general class of nonlinear discrete-time systems described by

$$x(k+1) = \omega(x(k), u(k)) + \eta(x(k), u(k)) + h(x(k), u(k)) \tag{4}$$

where $x \in \Re^n$ is the system state vector, $u \in \Re^m$ is the control input vector, $\omega : \Re^n \times \Re^m \to \Re^n$, $\eta : \Re^n \times \Re^m \to \Re^n$, $h : \Re^n \times \Re^m \to \Re^n$ are smooth vector fields. The term $\omega(x(k), u(k))$ represents the known nonlinear system dynamics while $\eta(x(k), u(k))$ is the system uncertainty. The

unknown fault function $h(x(k),u(k)) = \Pi(k-k_0)\,f(x(k),u(k))$ with $f(x(k),u(k))$ representing the unknown fault dynamics while $\Pi(k-k_0)$ being a $n \times n$ square matrix function representing the time profiles of faults, and $k_0 \geq 0$ is the initial time.

Typically, the time profile of the faults are modeled by

$$\Pi(k-k_0) = diag(\Omega_1(k-k_0), \Omega_2(k-k_0), ..., \Omega_n(k-k_0))$$

where

$$\Omega_i(\tau) = \begin{cases} 0, & \text{if } \tau < 0 \\ 1 - e^{-\bar{\kappa}_i \tau}, & \text{if } \tau \geq 0 \end{cases} \quad \text{for } i = 1, 2, ...n \tag{5}$$

and $\bar{\kappa}_i > 0$ is an unknown constant that represents the rate at which the fault in the corresponding state x_i occurs. The term $\Omega_i(\tau)$ approaches a step function when $\bar{\kappa}_i$ is large, which in turn represents an abrupt fault whereas a small value of $\bar{\kappa}_i$ implies incipient faults. It is important to understand that the exponential time profile is only used to classify the faults as incipient or abrupt. However, $f(x(k),u(k))$ represents the magnitude and the type of the fault. Since the fault function is expressed as a nonlinear function of the system states and the inputs, therefore, it represents a wide range of faults that can potentially occur in a given system. For example, such faults could be a piston wear in a compressor or an actuator fault.

Remark 1: The known nominal dynamics in (4) is in nonaffine form. However, for affine systems, the known nominal dynamics could be written as $\omega(x(k),u(k)) = \omega_f(x(k)) + \omega_g(x(k))u(k)$, where, $\omega_f \in \Re^{n\times 1}$ and $\omega_g \in \Re^{n\times m}$ are known smooth functions. However, the system uncertainty and the faults still be expressed in nonaffine form and are functions of the system states and the input. It is important to note that the following discussion for

nonaffine systems is also applicable to affine systems.

Remark 2: Modeling faults using the above time profile is quite common in the FD literature [32], and is used extensively by researchers [5, 9, 16-18].

Before proceeding any further, we propose the following assumption.

Assumption 1: The modeling uncertainty is unstructured and bounded [5, 9, 16-18] above satisfying $\|\eta(x(k),u(k))\| \leq \eta_M,\ \forall (x,u) \in (\chi \times U)$ where $\eta_M \geq 0$ is a known constant.

Remark 3: The uncertainties have to be bounded above in order to identify faults from system uncertainties.

In certain previously reported FD schemes [3, 8], the system uncertainty is assumed to structured, which helps to simplify the development of the FD scheme. In other schemes [1-3], structured faults are assumed, which also simplifies the development of the FD scheme. However, such assumptions are not considered in this paper.

In this paper, we consider a general framework for nonlinear systems with unknown system uncertainty. However, this complicates the design of a FD scheme, but is still undertaken in this paper. In the next section, the fault detection scheme is introduced by using a novel nonlinear observer using AIS as the online approximator. Additionally, using Lyapunov theory, the asymptotic performance of the proposed FD scheme is shown.

IV. Fault Detection Scheme

In this FD scheme, a nonlinear observer is designed to monitor and detect faults in the given system described in (4). It is essential to understand that the purpose of the FD observer is not to estimate the system states [16, 17] whereas to obtain residual for the purpose of detection.

A. Observer Dynamics

Consider the nonlinear observer described by

$$\hat{x}(k+1) = A_0\hat{x}(k) + \omega(x(k), u(k)) + \hat{h}(x(k), u(k); \hat{\theta}(k)) - A_0 x(k) + v(k) \tag{6}$$

where $\hat{x} \in \Re^n$ is the estimated state vector, A_0 is a constant $n \times n$ design matrix chosen by the user, $\hat{h} : \Re^n \times \Re^m \times \Re^{q \times n} \to \Re^n$ is the online approximator in discrete-time (OLAD) [18], $\hat{\theta} \in \Re^{q \times n}$ is a set of adjustable immune system parameters, and $v(k)$ is the robust adaptive term, which is to be defined later. Prior to the fault, the initial values for the estimated model (6) are taken as $\hat{x}(0) = x(0)$, $\hat{\theta}(0) = \hat{\theta}_0$, so that $\hat{h}(x, u, \hat{\theta}_0) = 0$ for all $x \in \chi$ and $u \in U$. Typically, the commonly used OLAD's are neural networks, fuzzy systems etc. However, in this paper, we consider AIS as an OLAD. Therefore, the AIS based OLAD is defined by using (3) as

$$\hat{h}(z, \hat{\theta}) = \hat{\theta}^T(k)\varphi(z) \tag{7}$$

where $z = [x, u]^T$ is the input vector, $\hat{\theta}(k) \in \Re^{l \times n}$ is a tunable immune system response, and $\varphi(z)$ is the AIS basis function as defined in (3).

Remark 4: Upon the detection of the fault, the OLAD and the robust adaptive term are initiated.

Now define the detection residual or state estimation error as $e = x - \hat{x}$. Then from (4) and (6) prior to the fault the residual dynamics are given by

$$e(k+1) = A_0 e(k) + \eta(x(k), u(k)) \tag{8}$$

In order to detect faults in the given system, the residual is compared against a known threshold via a dead-zone operator. The selection of the threshold is a challenging

task; however a mathematical procedure is developed for selecting it by using (8). It is important to note that by using a threshold, the robustness of the fault detection scheme can be improved [1, 2, 5, 16-18].

Prior to the fault, the residual, $e(k)$, remains within the threshold. However, in the event of a fault, the residual increases and crosses the threshold and therefore a fault is declared active. We define the threshold operator as $D[\cdot]$

$$D[e(k)] = \begin{cases} 0, & \text{if } \|e(k)\| \le \rho \\ e(k), & \text{if } \|e(k)\| > \rho \end{cases} \tag{9}$$

where $\rho > 0$ is the threshold. The selection of the dead-zone size ρ clearly provides a tradeoff between reducing the possibility of false alarms (robustness) and improving the sensitivity of the faults. The selection of an appropriate value for ρ is addressed next.

B. Fault Detection Threshold Selection

A suitable threshold is selected by solving the residual dynamics (8) through standard linear control theory as $e(k) = \sum_{j=0}^{k-1} A_0^{k-1-j} \eta(x(k), u(k))$. Since the matrix A_0 is stable with its poles chosen inside the unit disc, there exists two positive constants μ and β_c such that the Frobenius norm [33] $\|A_0^k\| \le \beta_c \mu^k \le 1$. Therefore, $\|e(k)\| \le \beta \eta_M \frac{(1-\mu^k)}{(1-\mu)}$, where $\beta = \beta_c \mu$. This implies that the threshold can be selected as a constant value $\rho = \frac{\beta \eta_M}{(1-\mu)}$ or a time-varying function as $\rho = \frac{\beta \eta_M (1-\mu^k)}{(1-\mu)}$. As a consequence, the residual $e(k)$ remains within the threshold for all $k \ge k_0$ and the OLAD and the robust adaptive terms stay at zero.

The dead-zone operator is utilized to turn the OLAD and robust adaptive terms online. Prior to the fault, *i.e.*, $\|e(k)\| \le \rho$, $\hat{\theta}(k) = \begin{bmatrix} 0 & . & . & . & 0 \\ . & . & . & . & . \\ . & . & . & . & . \\ 0 & . & . & . & 0 \end{bmatrix}_{l \times n}$, $0 \le k \le T$, and $v(k) = [0, 0, ..., 0]^T$.

This means $\hat{h}(x(k), u(k); \hat{\theta}(k)) = [0, 0, ..., 0]^T$, in the time interval $0 \le k \le T$ prior to a state or output fault.

When the residual exceeds the detection threshold, i.e., $\|e(k)\| > \rho$, a fault is declared active and the OLAD schemes that generate, $\hat{h}(.)$ is initiated. A standard delta-based parameter tuning algorithm [34] can be utilized whereas it is slower in convergence. To overcome this problem, the following parameter update law is used

$$\hat{\theta}(k+1) = \hat{\theta}(k) + \alpha\varphi(k)\mathrm{D}[e^T(k+1)] - \gamma \left\| I - \alpha\varphi(k)\varphi^T(k) \right\| \hat{\theta}(k) \tag{10}$$

is proposed where $\alpha > 0$ is the learning rate, $\gamma > 0$ is the adaptation rate, and $\varphi(k)$ is the OLAD basis function. Now using Theorem 1 and equation (3), we rewrite the fault dynamics in (4) as

$$h(x(k), u(k)) = \theta^T \varphi(x(k), u(k)) + \varepsilon_1(k) \tag{11}$$

where $\theta \in \Re^{l \times n}$ is the target parameter matrix such that the approximation error $\varepsilon_1(k)$ is bounded above and $\varphi(x(k), u(k))$ is the known basis function of the AIS. By appropriate selection of the antibodies in the AIS scheme, the approximation error can be decreased. The output of the OLAD is given by

$$\hat{h}(x(k), u(k); \hat{\theta}(k)) = \hat{\theta}(k)^T \varphi(x(k), u(k)) \tag{12}$$

where $\hat{\theta}(k) \in \Re^{l \times n}$ is the estimated AIS parameter matrix.

With this understanding of the proposed observer design, the stability of the

proposed fault detection scheme will be studied next. By using (4) and (6), the residual dynamics after the fault is given by

$e(k+1) = A_0 e(k) + \eta(x(k), u(k)) + h(x(k), u(k)) - \hat{h}(x(k), u(k); \hat{\theta}(k)) - v(k)$ where the robust adaptive term is defined as $v(k) = \dfrac{\hat{\theta}^T(k) B_1}{B_1^T \hat{\theta}(k) \hat{\theta}^T(k) B_1 + c_c}$, with $B_1 \in \Re^{l \times 1}$ is a constant vector and $c_c > 0$ a constant. Next using (11) and (12), the residual dynamics is rewritten as

$$e(k+1) = A_0 e(k) + \tilde{\theta}^T(k) \varphi(x, u) - \frac{\hat{\theta}^T(k) B_1}{B_1^T \hat{\theta}(k) \hat{\theta}^T(k) B_1 + c_c} + \varepsilon(k) \tag{13}$$

where $\varepsilon(k) = \varepsilon_1(k) + \eta(x(k), u(k))$ with the parameter estimation error defined as $\tilde{\theta}(k) = \theta - \hat{\theta}(k)$. Next, add and subtract $\dfrac{\left(\theta^T B_1 - C_1\right)}{B_1^T \hat{\theta}(k) \hat{\theta}^T(k) B_1 + c_c}$, in (13), where $C_1 \in \Re^{n \times 1}$ is a constant vector. The residual dynamics become

$$e(k+1) = A_0 e(k) + \Psi_1(k) + \Psi_2(k) + \varepsilon(k) - \frac{\left(\theta^T B_1 - C_1\right)}{B_1^T \hat{\theta}(k) \hat{\theta}^T(k) B_1 + c_c} \tag{14}$$

where

$\Psi_1(k) = \tilde{\theta}^T(k) \varphi(x, u)$ and $\Psi_2(k) = \dfrac{\left(\tilde{\theta}^T(k) B_1 - C_1\right)}{B_1^T \hat{\theta}(k) \hat{\theta}^T(k) B_1 + c_c}$. Next the following lemma is needed in order to proceed.

Lemma 1: The term, $\varepsilon(k)$, comprising of the approximation error, $\varepsilon_1(k)$, and the system uncertainty, $\eta(x(k), u(k))$ are bounded above according to

$$\varepsilon^T(k) \varepsilon(k) \le \varepsilon_M = d_0 + d_1 \left\| \tilde{\theta}(k) \right\|^2 + d_2 \left\| e(k) \right\| \left\| \tilde{\theta}(k) \right\| + d_3 \left\| e(k) \right\|^2 \tag{15}$$

where $d_0, d_1, d_2,$ and d_3 are computable positive constants.

Proof: Refer to Appendix.

Remark 5: This lemma is necessary similar to the case of continuous-time [38] while such results are not available for discrete-time systems. This result is very mild [35-38] when compared to the case where the approximation error is considered bounded above by a known constant.

Next, the following theorem guarantees the asymptotic stability of the proposed FD scheme after a fault occurs. Additionally, it is clear that prior to the fault the system remains stable for a bounded system uncertainty $\eta(x(k),u(k))$. This is evident from (8) since A_0 has eigen values within the unit disc.

Theorem 2 (FD Observer Performance upon Detection): Let the proposed nonlinear observer in (6) be used to monitor the system given in (4). Let the update law given in (10) be used for tuning the immune response of the AIS based OLAD. In the presence of a fault and bounded system uncertainties, the detection residual, $e(k)$, and the parameter estimation errors $\tilde{\theta}(k)$ are locally asymptotically stable provided:

(a) $$A_{0_{\max}} \leq \sqrt{\frac{(1/5)}{\left(4+20\alpha(2+1/\delta)\varphi_{\max}^2\right)}} \quad , \quad 0<\alpha<1 \tag{16}$$

(b) $$\frac{1-\sqrt{1-C_r}}{\left\|I-\alpha\varphi(k)\varphi^T(k)\right\|} \leq \gamma \leq 1/\left\|I-\alpha\varphi(k)\varphi^T(k)\right\| \tag{17}$$

and

(c) $0<\lambda_s<0.5$, $0<\delta<1$ (18)

where $\|A_0\| \leq A_{0_{\max}}$, $\|\varphi(k)\| \leq \varphi_{\max}$, and $C_r > 0$ is a constant.

Proof: Refer to Appendix.

Remark 6: Theorem 2 guarantees the asymptotic stability of the proposed FD scheme after a fault occurs. In other words, the proposed OLAD will characterize the faults

accurately in comparison with the detection schemes in continuous-time where a bounded residual is demonstrated [5, 9].

In the next section, the prediction scheme is introduced.

V. PREDICTION SCHEME

Thus far, a new FD estimator design using the AIS as online approximator was introduced and its stability was studied rigorously. Now TTF can be determined using the behavior of the immune system parameter trajectories before and after the occurrence of a fault. The following assumption holds in deriving the TTF.

Assumption 2: The parameter vector $\hat{\theta}(k)$ is an estimate of the actual system parameters.

Remark 7: This assumption is satisfied when a system can be expressed as linear in the unknown parameters (LIP). For example, in a mass damper system, or in civil infrastructure such as a bridge, the mass, damping constant and spring constant may be expressed as linear in the unknown parameters. In the event of a fault, system parameters change, and tend to reach their limits. When any one of the parameters exceeds its limit, operation is considered unsafe. TTF is defined as the time elapsed when the first parameter reaches its limit. The TTF can also be analyzed with lower limits.

In this section, to develop an explicit mathematical equation for predicting TTF, we use the parameter update law given in (10). Subsequently, by using this equation, we develop an algorithm for the continuous prediction of TTF iteratively at every time instant. Alternatively, estimated state trajectories can be employed as well if the states can be related to physical quantities. Next, the mathematical equation is presented in the following theorem.

Theorem 3 (Time to Failure): If the system in (4) can be expressed as LIP, the TTF for the ij^{th} system parameter at the k^{th} time instant can be determined using

$$k_{f_{ij}} = \frac{\left| \log\left(\left| \frac{\left(\gamma \left\| I - \alpha\phi\phi^T \right\| \theta_{ij_{\max}} - \alpha \left(\phi e^T \right)_{ij} \right)}{\left(\gamma \left\| I - \alpha\phi\phi^T \right\| \theta_{ij_0} - \alpha \left(\phi e^T \right)_{ij} \right)} \right| \right) \right|}{\left| \log(1 - \gamma \left\| I - \alpha\phi\phi^T \right\|) \right|} + k_{0_{ij}} \tag{19}$$

where $k_{f_{ij}}$ is the TTF, $k_{0_{ij}}$ is the time instant when the prediction starts (bearing in mind that k_{dt} was the initial value, which increases incrementally), $\theta_{ij_{\max}}$ is the maximum value of the system parameter, and θ_{ij_0} is the value of the system parameter at the time instant $k_{0_{ij}}$.

Proof: Refer to [39].

Remark 8: The mathematical equation (19) presents the TTF for the ij^{th} system parameter. In general, for a given system with a parameter vector, the TTF would be $k_{ft} = \min(k_{f_{ij}}), i = 1, 2, \ldots\ldots l$, $j = 1, \ldots\ldots, n$, where $l \times n$ are the number of system parameters. The TTF is defined as the time elapsed when the first parameter reaches its limit. The speed at which the actual parameters approach their target values is dictated by the learning rate or adaptation gain and the design constant in the parameter update law (10). A small value for the learning rate implies slower convergence which further means that the TTF is not as accurate when the learning rate is higher. However, a large value of the learning rate can speed up the convergence. Increasing the learning rate can cause hunting problems which will result in inaccurate prediction of TTF.

Remark 9: Although the proposed prediction scheme is based on the parameter trajectory, estimated system states could also be used for prediction since asymptotic stability is proven. A relationship similar to (19) can be derived for TTF using (6). However, for

brevity, no further discussions on the use of state trajectories for prediction are included in this paper.

Remark 10: The proposed prediction scheme could be applied to unknown systems that satisfy LIP. It could also be applied to systems with partial information that satisfy LIP. Such systems were addressed in Section III.

Figure 1 provides a flow chart of the iterative algorithm to determine TTF (k_{ft}) for each system parameter. The TTF is calculated at each time instant starting when a fault is detected until the system parameter reaches its maximum value (threshold). Therefore, it is logical that the TTF decreases as the parameters approach their corresponding limits. The simulation results presented below will indeed show that the performance of the FDP scheme as indicated in the theorems can be demonstrated in simulation.

By tuning the system parameter estimate ($\hat{\theta}_i(k)$) to update the TTF recursively, the system could be more accurately monitored than would be possible with other methods [13, 14]. In fact, the TTF will not be accurate when the parameter estimate vector is just started. Over time when the parameter vector starts converging to its true values, the TTF prediction starts improving. Additionally, no prior offline training is required to estimate the system parameters, which significantly reduces the burden of collecting data. In the next section, we present some simulation example and later some experimental study to illustrate the proposed FDP scheme.

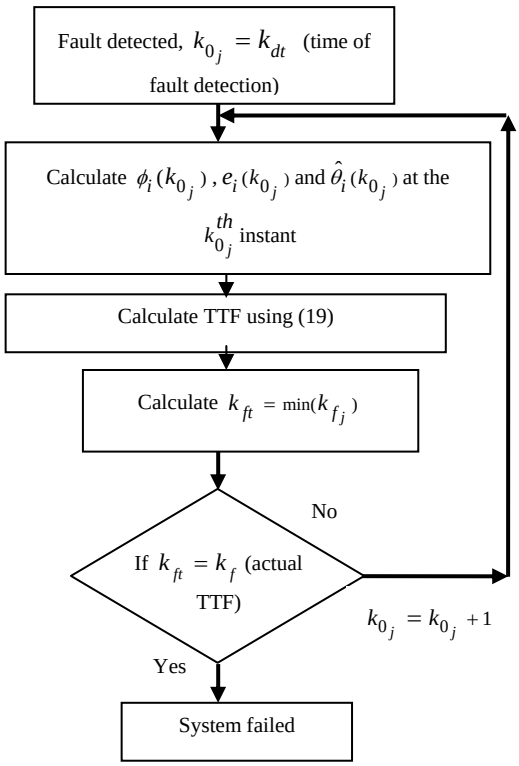

Figure 1: Flow Chart Indicating the TTF Determination.

VI. SIMULATION RESULTS

In this section, two different simulation examples are presented to demonstrate the proposed fault detection scheme. Initially, a two link manipulator is considered followed by an axial piston pump. Subsequently, in the next section, the proposed FDP scheme is verified on a pump test bed.

A. Two link robot manipulator

A schematic of a two degree of freedom manipulator is shown in Fig. 2 and its dynamics model is given below [24]

$$\tau = M(\theta)\ddot{\theta} + V(\theta, \dot{\theta}) + G(\theta) + F(\dot{\theta}) \tag{20}$$

where $\theta = [\theta_1, \theta_2]^T$ is the vector of angular positions and $\dot{\theta} = [\dot{\theta}_1, \dot{\theta}_2]^T$ is the vector of angular velocity of links 1 and 2 respectively. Additionally, $M(\theta)$ is the inertia matrix, $V(\theta, \dot{\theta})$ is the coriolis or centripetal matrix, $G(\theta)$ is the gravity vector, and $F(\dot{\theta})$ is the friction vector. Moreover, τ is a vector of torque input applied to the two link manipulator.

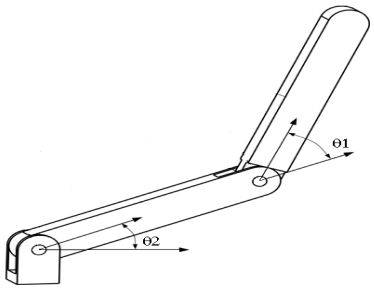

Fig. 2: Schematic of a Two Link Manipulator.

For convenience, we express (20) in the following discrete-time state space form

$$x_1(k+1) = Tx_2(k) + x_1(k)$$

$$x_3(k+1) = Tx_4(k) + x_3(k)$$

$$\begin{bmatrix} x_2(k+1) \\ x_4(k+1) \end{bmatrix} = T\left(M^{-1}\begin{pmatrix} x_1 \\ x_3 \end{pmatrix}\left(\begin{vmatrix} \tau_1 \\ \tau_2 \end{vmatrix} - V\left(\begin{bmatrix} x_1 \\ x_3 \end{bmatrix}, \begin{bmatrix} x_2 \\ x_4 \end{bmatrix}\right) - G\left(\begin{bmatrix} x_1 \\ x_3 \end{bmatrix}\right) - F\left(\begin{bmatrix} x_3 \\ x_4 \end{bmatrix}\right)\right)\right) + \begin{bmatrix} x_2(k) \\ x_4(k) \end{bmatrix}$$

where $x = [x_1, x_2, x_3, x_4]^T$ is the system state vector. We assume an actuator fault, which is expressed as $h(k) = [0, 1.8(1 - e^{-0.05(k-40)})\tau_1(k), 0, 0]^T$. The fault is induced at the 40^{th} second of system operation with a growth rate of 0.05. Moreover, we assume the sampling time for this simulation is taken as $T = 10m\sec$. Additionally, a white noise is introduced in this simulation

with a magnitude of 0.004 units and a constant uncertainty of 0.5 units. To monitor and detect faults in the given system, we use the following FD estimator

$$\hat{x}_1(k+1) = Tx_2(k) + x_1(k) + 0.005(\hat{x}_1(k) - x_1(k))$$

$$x_3(k+1) = Tx_4(k) + x_3(k) + 0.005(\hat{x}_3(k) - x_3(k))$$

$$\begin{bmatrix} x_2(k+1) \\ x_4(k+1) \end{bmatrix} = T\left(M^{-1}\begin{pmatrix} x_1 \\ x_3 \end{pmatrix}\left(\begin{vmatrix} \tau_1 \\ \tau_2 \end{vmatrix} - V\left(\begin{bmatrix} x_1 \\ x_3 \end{bmatrix}, \begin{bmatrix} x_2 \\ x_4 \end{bmatrix}\right) - G\left(\begin{bmatrix} x_1 \\ x_3 \end{bmatrix}\right) - F\left(\begin{bmatrix} x_3 \\ x_4 \end{bmatrix}\right)\right)\right) + \begin{bmatrix} x_2(k) \\ x_4(k) \end{bmatrix} + \begin{vmatrix} 0.005(\hat{x}_2(k) - x_2(k)) \\ 0.005(\hat{x}_4(k) - x_4(k)) \end{vmatrix} \quad (21)$$

where $\hat{x} = [\hat{x}_1, \hat{x}_2, \hat{x}_3, \hat{x}_4]^T$ is the estimated state vector, the OLAD is taken as $\hat{h}(k) = [0, \hat{\theta}(k)\tau_1(k), 0, 0]^T$. Next, using (20) and (21), we generate the norm of the residual as shown in Fig. 3. Since we assumed some disturbances, therefore, we need a threshold to improve the robustness of the proposed fault detection scheme. The threshold is derived by taking $\beta = 1.03$, $\mu = 0.01$, and $\eta_M \approx 0.5$, we have $\rho \approx 0.52$. As seen in Fig. 3, the residual remains within the threshold prior to the fault, however, after the fault occurs, the residual exceeds the threshold. Subsequently, the OLAD and the robust adaptive terms are initiated to learn the unknown fault dynamics. This is evidenced by the fact the residual quickly drops after initiating the OLAD and the robust adaptive term. Additionally, the asymptotic convergence of the residual after the fault is guaranteed as seen in Fig. 3. Therefore, the theoretical results presented in this paper are validated.

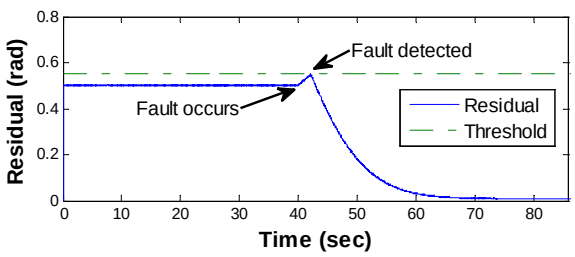

Fig. 3: Residual and the FD Threshold.

Next, the online estimation of the fault magnitude using the proposed OLAD scheme is shown in Fig. 4. As seen in the figure, the online learning is found to be satisfactory. The parameter of the OLAD is tuned online using the update law in (9) with $\alpha = 0.034$ and $\gamma = 0.1$. Using the online estimation of the parameters, we estimate TTF as shown in Fig. 5. From the figure, it's evident that the TTF prediction is satisfactory. However, it is noted that the first few seconds of TTF prediction after the fault detection didn't render reliable results therefore, is not presented. This could be attributed to the random selection in the gains of the weight update law. However, after the 50th second of the system operation, the TTF prediction seems to be reasonable and converges to the actual time of failure, which is 79.43 sec.

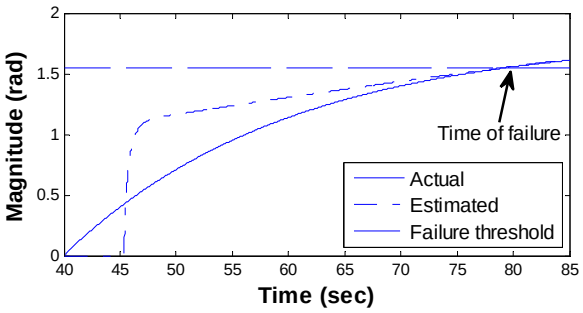

Fig. 4: Online Estimation of the Fault Magnitude.

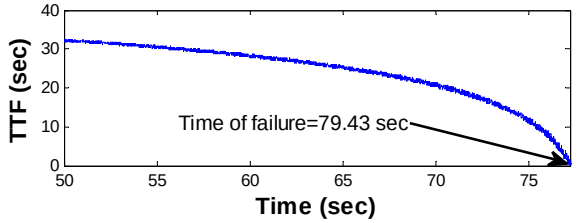

Fig. 5: The TTF Determination Due to the Incipient Actuator Fault.

To show that the proposed scheme is generic, next, an axial piston pump example is considered in simulation.

B. Axial Piston Pump

A discrete-time dynamic representation of the axial piston pump derived in [40] is given as

$$x_i(k+1) = \frac{BT}{C - A_p S_{pi}}\left(Q_{kpi}(k) - Q_{pi}(k) - Q_{lpi}(k)\right) + x_i(k), i = 1,..,9$$

$$x_{10}(k+1) = \frac{TBC}{V_c}\left(Q_p(k) - Q_s(k)\right) + x_{10}(k)$$

$$y(k) = x_{10}(k) \tag{22}$$

where, $x_i(k)$, $i = 1,...,9$ are the system states. Additionally, $[x_1(k),.....,x_9(k)]^T$ is the pressure in the nine pistons, x_{10} is the pump outlet pressure, B is the bulk modulus of the hydraulic fluid, T is the sample timing, V_c is the theoretical volume of flow, and A_p is the piston area. Moreover, S_{pi}, Q_{kpi}, Q_{pi}, Q_{lpi}, and Q_s are the i^{th} piston stroke length, kinematical flow from the piston chamber to the discharge chamber, internal leakage from piston to the case chamber, and the outlet flow of the pump respectively. Additionally, they are obtained using the following equation

$$Q_{kpi}(k) = \frac{\omega \pi d^2 R_p}{4} \tan\beta_c . \sin(\omega k - (i-1)\alpha_p)$$

$$Q_{lpi}(k) = \frac{\pi r h_g^3}{6\mu L}(x_i(k) - P_c)$$

$$Q_{pi}(k) = C_{d1} A_{di} \sqrt{\frac{2|x_i(k) - x_{10}(k)|}{\rho_c}}\, sign(x_i(k) - x_{10}(k)) \;, i = 1,..,9$$

$$Q_p(k) = \sum_{i=1}^{9} Q_{pi}(k)$$

$$Q_s(k) = C_{d2} A_v \sqrt{\frac{2x_{10}(k)}{\rho_c}}$$

$$S_{pi}(k) = R_p \tan\beta_c (1 - \cos(\omega k - (i-1)\alpha_c))$$

where ω is the angular velocity of the pump drive shaft (rad/s), d is the diameter of the piston (m), R_p is the piston pitch radius on barrel, β_c is the angle of swash plate, α_p is the phase delay (rad), r is the radius of piston (m), h_g is the radial clearance between piston and cylinder bore (m), μ is the absolute fluid viscosity (N sec/m^2), L is the length of leakage passage (m), C_{d1} is the flow discharge coefficient of the discharge areas for piston port opening to discharge chamber, ρ_c is the flow density (kg/m^3), A_{di} is the i^{th} discharge area for piston port opening to the discharge chamber in valve plate (m^2), C_{d2} is the discharge coefficient of needle valve orifice, and A_v is the orifice area of the needle valve (m^2). The values of the parameters used in this simulation are taken from [40] and we use a sampling interval of $T = 10m\sec$. To monitor and detect faults in (22), we use the following FD estimator

$$\hat{x}_i(k+1) = \frac{BT}{C - A_p S_{pi}} \left(Q_{\varphi pi}(k) - Q_{pi}(k) - Q_{lpi}(k) \right) + x_i(k) + A_{0_i} (\hat{x}_i(k) - x_i(k)) \ , i = 1,..,9$$

$$\hat{x}_{10}(k+1) = \frac{TBC}{V_c} \left(Q_p(k) - Q_s(k) \right) + x_{10}(k) + A_{0_{10}} (\hat{x}_{10}(k) - x_{10}(k)) \qquad \hat{y}(k) = \hat{x}_{10}(k) \qquad (23)$$

where, $\hat{x}_i(k)$, $i = 1,...,10$ are the estimated system states.

$A_0 = 10^{-4} diag(0.0630, 0.1796, 0.8, 0.0305, 0.1431, 0.1683, 0.1567, 0.1996, 0.1172, 0.0001)$ is the estimator gain matrix. For this simulation, two different faults, i.e., piston wear fault and pressure sensor fault are seeded. First, we discuss the piston wear fault.

B.1) Piston Wear Fault

An incipient piston wear fault described by

$$h(k) = [0,0,0,0,0,0,0,0,0,34\left(1 - e^{-0.02(k-100)}\right)]^T$$

is induced at the 100th minute of system operation. Additionally, a constant uncertainty of 30 units is considered in the simulation. Next to detect the fault online, we generate norm of the residual (i.e., $|e(k)| = |x_{10}(k) - \hat{x}_{10}(k)|$) from (22) and (23) as shown in Fig. 6. Due to the presence of system uncertainties, a threshold is needed to guarantee robustness. Therefore, by taking $\beta = 1.15$, $\mu = 0.01$, and $\eta_M = 30$, we have $\rho \approx 35$, a constant threshold as shown in Fig. 6. From the figure, we see that the fault is detected at 105th minute. After the detection, the OLAD is initiated to learn online the magnitude of the unknown fault dynamics as shown in Fig. 7. Additionally, parameters of the OLAD are tuned online using the update law in (9) with $\alpha = 0.1$ and $\gamma = 0.001$. From the figure, it is observed that the online learning of the fault by the OLAD is satisfactory.

Subsequently, the TTF is determined using the scheme outlined in Section IV and is shown in Fig. 8. From the figure, the initial TTF prediction and the oscillatory behavior in the prediction is attributed to the random selection of the gains. However, as the online estimation of the fault parameter improves, the TTF prediction improves and concurs with the actual time of failure, which 251 min.

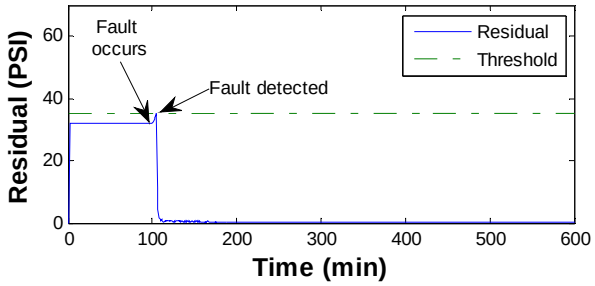

Fig. 6: Residual and the FD Threshold- Piston Wear Fault.

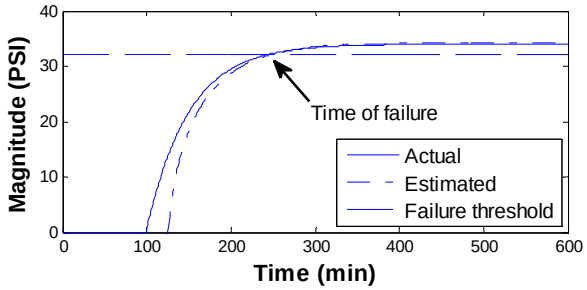

Fig. 7: Online Estimation of the Piston Wear Fault Magnitude.

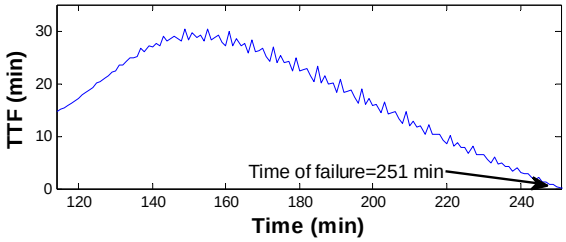

Fig. 8: The TTF Determination Due to the Piston Wear Fault.

B.2) Outlet Pressure Sensor Fault

Next, a pressure sensor fault is induced, which may be due to loose wiring. Mathematically, the fault is described by

$$h_o(k) = \begin{cases} 0 & for k < 100 \text{ min} \\ -0.4(k-99) & \text{for } 100 < k < 300 \text{ min} \\ -1390 & \text{for } k >= 300 \text{ min} \end{cases}$$

For sake of completeness, we assumed a time varying disturbance of 1 unit magnitude. Therefore, we need a threshold to avoid missed or false alarms. Thus by taking $\beta = 1.48$, $\mu = 0.01$ and $\eta_M = 1$, we have $\rho \approx 1.5$. The fault is induced at the 100th minute of system operation. After the fault is initiated, the norm of the residual tends to increase as observed in Fig. 9. Therefore, the fault is detected when the residual exceeds the threshold. Subsequently, the OLAD $\hat{h}_o(k) = \hat{\theta}(k)\varphi(k)$ is initiated to learn online the unknown fault dynamics. Moreover, the OLAD parameter is tuned online using (9) with $\alpha = 0.61$ and $\gamma = 0.001$. Although, the fault begins at 100 minutes, the fault tends to grow and has a sudden increase in the magnitude, which is similar to a step fault. Therefore, we see that the magnitude of the fault changes to a large value, which increases the residual to a large value as seen at around 300 minutes in Fig. 9. However, as the OLAD continues to learn the fault online, eventually the residual converges to zero as seen in the figure.

Next, the online learning of the fault dynamics by the OLAD is given in Fig. 10 and it is found to be satisfactory.

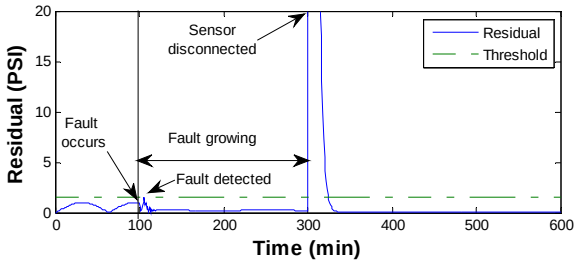

Fig. 9: Residual and the FD Threshold- Output Sensor Fault.

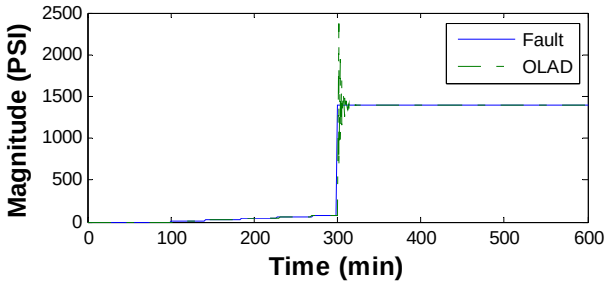

Fig. 10: Evolution of the Pressure Sensor Fault and the OLAD Learning.

Till now, we presented two examples in simulation to verify the proposed scheme. However, in the next section, we verify the proposed FD scheme on an axial piston pump test bed. Additionally, the two faults, i.e., piston wear and pressure sensor faults are induced through accelerated testing as detailed below.

VII. Experimental Results

The performance of the proposed FDP scheme is evaluated on a pump test bed. In addition, the two faults assumed in the simulation are used in the experimental study. The piston wear fault was induced by creating cavitation in the axial piston pump test stand, which shown in Fig. 11. In addition, the sensor fault was due to the loosing wiring. In the test stand shown, we have a 10.5cc variable displacement axial piston pump with nine pistons. On the test stand, the inlet, outlet, and case drain pressures were recorded continuously at 1 kHz using NI cDAQ 9172 hardware. Additionally, the case drain flow, outlet flow, reservoir temperature, case drain temperature, and pump temperature were also recorded.

The estimator model derived in (22) is used again for detecting faults in the pump. Moreover, from the model given in (22), we could see that only the output pressure is

measurable. Therefore, we use the measured outlet pressure for detecting faults in the pump. Before using the data, due to the measurement noise, therefore, to attenuate them, we use a 10^{th} order band-pass pass Butterworth filter with a cut-off frequency of 250 Hz and 300Hz. A snapshot of the raw data and the filtered data for the outlet pressure signal is shown in Figs. 12 and 13 respectively. As seen in Fig. 13, the raw data is filtered using the above defined filter and averaged over a one second fixed time window. Subsequently, the filtered data was used for the verification of the FDP scheme.

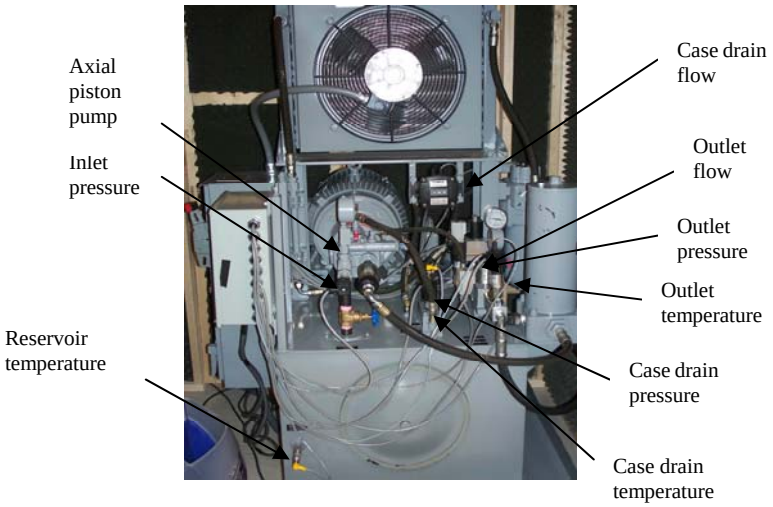

Fig. 11: Picture of the Axial Piston Pump Test Bed.

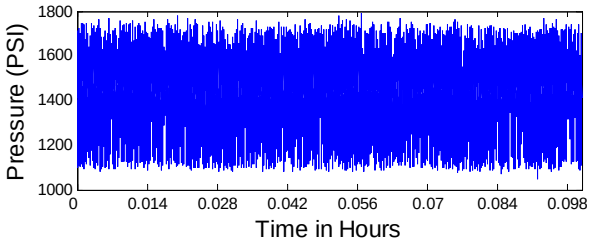

Fig. 12: Raw outlet pressure signal.

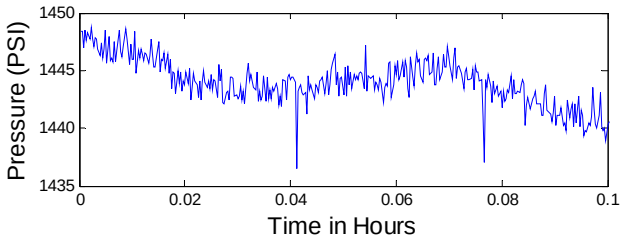

Fig. 13: Processed outlet pressure signal.

Therefore, the FD estimator in (23) is used for monitoring and detecting fault in the pump test bed. The residual generated by comparing the experimental outlet pressure with that of the estimated outlet pressure from the FD estimator is shown in Fig. 14. In this case, the threshold is obtained by taking $\beta = 1.1$, $\mu = 0.01$ and $\eta_M \approx 25$, we have $\rho \approx 28$. As seen in the figure, the residual remains bounded for the healthy system operation. However, as the fault occurs due to the accelerated testing, the residual tends to increase and thus exceeds the threshold. Subsequently, the fault is detected and the OLAD and robust term are initiated.

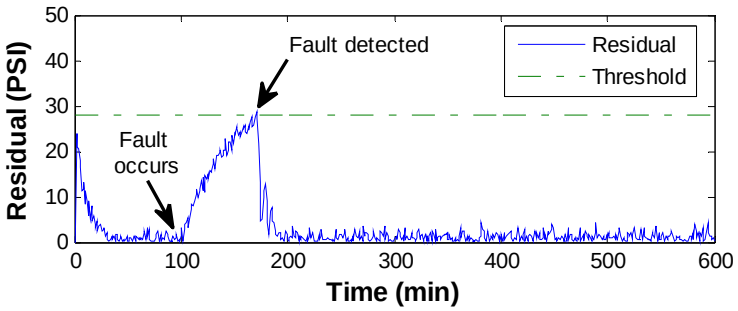

Fig. 14: Residual and the FD Threshold- Piston Wear Fault (experimental results).

Moreover, the OLAD is tuned online using (9) with $\alpha = 0.2$ and $\gamma = 0.03$. From Fig. 15, we could see the satisfactory estimation of the fault magnitude by the OLAD. It is noted that the fluctuations in the magnitude of the OLAD response were reduced to demonstrate the learning. Subsequently, using the online estimation of the fault magnitude, the TTF prediction is determined as shown in Fig. 16. Since the initial online estimation of the fault magnitude was not accurate and also due the random selection in the gains, the TTF prediction was not accurate. However, as the learning improved and approached the actual failure, the TTF prediction was satisfactory. Therefore, in Fig. 16, the TTF prediction is shown only for the last few minutes before the failure.

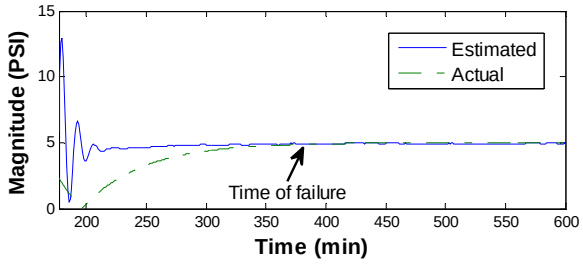

Fig. 15: Online Estimation of the Piston Wear Fault Magnitude (experimental results).

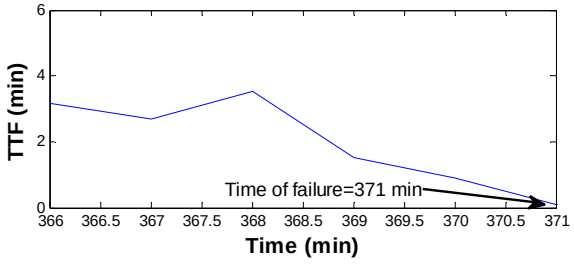

Fig. 16: The TTF Determination Due to the Piston Wear Fault.

In the next case, we assume a pressure sensor fault on the axial piston pump. The norm of the residual used for detecting the fault is shown in Fig. 17. Here again, the residual shown is the difference in the estimated and the experimental outlet pressure. Also, the threshold is obtained by taking $\beta = 1.11$, $\mu = 0.01$ and $\eta_M = 16$, we have a constant FD threshold of $\rho \approx 18$. From the figure, the fault occurs at the 100th min of operation, where, the sensor fault is due to the loosening of the connect pin, and has a unique behavior. The fault grows with time and at around 300 minutes; the connecting pin is detached completely off the sensor. Therefore, we see a sharp increase in the residual as seen in Fig. 17. Although, the OLAD and the robust term were activated the first time the residual exceeded the threshold, however, the residual converges to zero only after the second spike as in the figure.

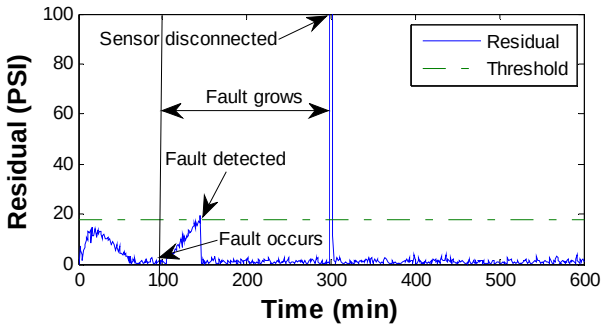

Fig. 17: Residual and the FD Threshold- Pressure Sensor Fault (Experimental Results).

Moreover, the learning of the fault by the OLAD is shown in Fig. 18, and is found to be highly satisfactory. Similar to the previous case, the OLAD is tuned online using (9) with $\alpha = 0.9$ and $\gamma = 1 \times 10^{-9}$.

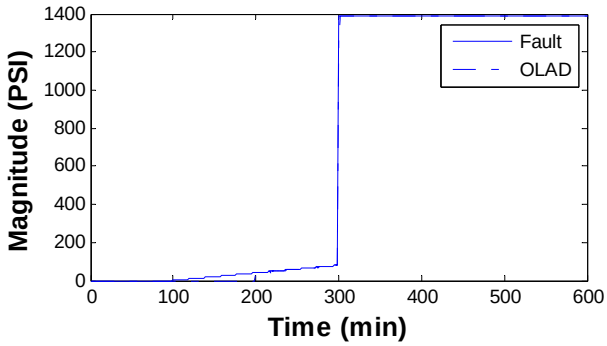

Fig. 18: Evolution of the Pressure Sensor Fault and the OLAD Learning (Experimental Results).

Therefore, from the simulation and experimental verification, one could see that the proposed scheme detects and learns both the incipient and abrupt faults online without any *apriori* offline training. Moreover, the experimental results show the feasibility in the implementing the proposed scheme on an experimental hardware. Therefore, the proposed FDP scheme renders a stable performance both in simulation and in practice.

VIII. Conclusions

In this paper, an online fault detection scheme using a new online approximator using AIS is proposed for a class of nonaffine nonlinear discrete-time systems. An asymptotic estimator is designed to monitor and detect faults in the given system. The scheme could detect both the abrupt and incipient faults. Mathematical asymptotic stability results of the proposed fault detection scheme are derived. Moreover, initially two simulation examples were presented to demonstrate the asymptotic stability and the online learning capabilities of the proposed AIS based FD estimator. Later, the FD scheme was verified on an axial-piston pump test bed. From the experimental results, the FD scheme is

found to successful in detecting and learning online both the incipient and abrupt faults. Therefore, the proposed FD scheme renders asymptotic performance both in simulation and in experiment.

APPENDIX

Proof of Lemma 1: Consider (14) and solving it would render

$$e(k) = A_0^k e(0) + \sum_{j=0}^{k} A_0^{k-j}\left[\tilde{\theta}^T(j)\varphi(j) + \frac{\tilde{\theta}^T(j)B_1 - C_1}{B_1^T\hat{\theta}(j)\hat{\theta}^T(j)B_1 + c_c} + \varepsilon(j) - \frac{\left(\theta^T B_1 - C_1\right)}{B_1^T\hat{\theta}(j)\hat{\theta}^T(j)B_1 + c_c}\right]$$

The above equation is rewritten as

$$\sum_{j=0}^{k} A_0^{k-j}\varepsilon(j) = e(k) - A_0^k e(0) - \sum_{j=0}^{k} A_0^{k-j}\tilde{\theta}^T(j)\varphi(j) - \sum_{j=0}^{k} A_0^{k-j}\frac{\left(\tilde{\theta}^T(j)B_1 - C_1\right)}{B_1^T\hat{\theta}(j)\hat{\theta}^T(j)B_1 + c_c} + \sum_{j=0}^{k} A_0^{k-j}\frac{\left(\theta^T B_1 - C_1\right)}{B_1^T\hat{\theta}(j)\hat{\theta}^T(j)B_1 + c_c}\Bigg]$$

Take the Frobenius norm to render

$$\left\|\sum_{j=0}^{k} A_0^{k-j}\varepsilon(j)\right\| \le \left\|e(k)\right\| + \left\|A_0^k e(0)\right\| + \left\|\sum_{j=0}^{k} A_0^{k-j}\tilde{\theta}^T(j)\varphi(j)\right\|$$

$$+ \left\|\sum_{j=0}^{k} A_0^{k-j}\frac{\tilde{\theta}^T(j)B_1}{B_1^T\hat{\theta}(j)\hat{\theta}^T(j)B_1 + c_c}\right\| + \left\|\sum_{j=0}^{k} A_0^{k-j}\frac{\theta^T B_1}{B_1^T\hat{\theta}(j)\hat{\theta}^T(j)B_1 + c_c}\right\|$$

The summation term in the above equation could be solved using [33] as $\left\|\sum_{j=0}^{k} A_0^{k-j}\tilde{\theta}^T(j)\varphi(j)\right\| \le \frac{\varphi_{\max}\left\|\tilde{\theta}(k)\right\|}{(1 - A_{0_{\max}})}$, where $A_{0_{\max}}$ is the maximum singular value of A_0. Additionally, $A_{0_{\max}} < (1 - A_{0_{\max}})$ if $A_{0_{\max}} \le \lambda_s$, where λ_s is a positive constant, therefore, we have $\frac{\varphi_{\max}\left\|\tilde{\theta}(k)\right\|}{(1 - A_{0_{\max}})} \le \frac{\varphi_{\max}\left\|\tilde{\theta}(k)\right\|}{A_{0_{\max}}}$. Similar result could be derived for the other terms for instance, $\frac{\tilde{\theta}^T(k)B_1}{B_1^T\hat{\theta}(k)\hat{\theta}^T(k)B_1 + c_c} \le \tilde{\theta}^T(k)B_1$, $\frac{\theta^T B_1}{B_1^T\hat{\theta}(k)\hat{\theta}^T(k)B_1 + c_c} \le \theta^T B_1$. The above equation could be rewritten as

$$\frac{\left\|\varepsilon(k)\right\|}{A_{0_{\max}}} \le \left\|e(k)\right\| + \left\|A_0^k e(0)\right\| + \frac{\varphi_{\max}\left\|\tilde{\theta}(k)\right\|}{A_{0_{\max}}} + \frac{B_{1_{\max}}\left\|\tilde{\theta}(k)\right\|}{A_{0_{\max}}} + \frac{B_{1_{\max}}\left\|\theta\right\|}{A_{0_{\max}}}$$

Squaring both side and factoring $A_{0_{\max}}^2$ would give us

$$\left\|\varepsilon(k)\right\|^2 \le A_{0_{\max}}^2 \left(\left\|e(k)\right\| + \left\|A_0^k e(0)\right\| + \frac{\varphi_{\max}\left\|\tilde{\theta}(k)\right\|}{A_{0_{\max}}} + \frac{B_{1_{\max}}\left\|\tilde{\theta}(k)\right\|}{A_{0_{\max}}} + \frac{B_{1_{\max}}\left\|\theta\right\|}{A_{0_{\max}}} \right)^2$$

Take $b_0 = \left\|A_0^k e(0)\right\| + \frac{B_{1_{\max}}\left\|\theta\right\|}{A_{0_{\max}}}$, $b_1 = \varphi_{\max} + B_{1_{\max}}$, then the above equation could be rewritten as

$$\left\|\varepsilon(k)\right\|^2 \le A_{0_{\max}}^2 \left(\left\|e(k)\right\| + b_0 + \frac{b_1\left\|\tilde{\theta}(k)\right\|}{A_{0_{\max}}} \right)^2$$

Expand the term on the right hand side of the above equation, we have

$$\left\|\varepsilon(k)\right\|^2 \le A_{0_{\max}}^2 \left[b_0^2 + \left\|e(k)\right\|^2 + \frac{b_1^2\left\|\tilde{\theta}(k)\right\|^2}{A_{0_{\max}}^2} + \underbrace{2\left\|e(k)\right\| b_0}_{1} + \underbrace{\frac{2b_0 b_1\left\|\tilde{\theta}(k)\right\|}{A_{0_{\max}}}}_{1} + \frac{2b_1\left\|e(k)\right\|\left\|\tilde{\theta}(k)\right\|}{A_{0_{\max}}} \right]$$

Apply Cauchy-Schwarz inequality to terms numbered as 1 in the above equation, and combine similar terms, we would have the following equation

$\left\|\varepsilon(k)\right\|^2 \le 3A_{0_{\max}}^2 b_0^2 + 2A_{0_{\max}}^2\left\|e(k)\right\|^2 + 2b_1^2\left\|\tilde{w}(k)\right\|^2 + 2A_{0_{\max}} b_1\left\|e(k)\right\|\left\|\tilde{\theta}(k)\right\|$ Taking $\overline{d}_0 = 3A_{0_{\max}}^2 b_0^2$, $\overline{d}_1 = 2A_{0_{\max}}^2$, $\overline{d}_2 = 2b_1^2$, and $\overline{d}_3 = 2A_{0_{\max}} b_1$, would reveal equation (15).

Proof of Theorem 2: Consider the Lyapunov function candidate

$$V = e^T(k)e(k) + \frac{1}{\alpha} tr[\tilde{\theta}^T(k)\tilde{\theta}(k)]$$

whose first difference is given by

$$\Delta V = \underbrace{e^T(k+1)e(k+1) - e^T(k)e(k)}_{\Delta V_1} + \underbrace{\frac{1}{\alpha} tr[\tilde{\theta}^T(k+1)\tilde{\theta}(k+1) - \tilde{\theta}^T(k)\tilde{\theta}(k)]}_{\Delta V_2} \qquad \text{(A.1)}$$

Substitute (14) in ΔV_1 of (A.1), therefore, we have

$$\Delta V_1 = \left(A_0 e(k) + \Psi_2(k) + \Psi_1(k) + \varepsilon(k) - \frac{(\theta^T B_1 - C_1)}{B_1^T \hat{\theta}(k)\hat{\theta}^T(k) B_1 + c_c}\right)^T \times$$

$$\left(A_0 e(k) + \Psi_2(k) + \Psi_1(k) + \varepsilon(k) - \frac{(\theta^T B_1 - C_1)}{B_1^T \hat{\theta}(k)\hat{\theta}^T(k) B_1 + c_c}\right) - e^T(k) e(k)$$

Perform some mathematical manipulations to render

$$\Delta V_1 = e^T(k) A_0^T A_0 e(k) + 2e^T(k) A_0^T \Psi_1(k) + 2e^T(k) A_0^T \Psi_2(k) + 2e^T(k) A_0^T \varepsilon(k) - 2\frac{e^T(k) A_0^T (\theta^T B_1 - C_1)}{B_1^T \hat{\theta}(k)\hat{\theta}^T(k) B_1 + c_c} + \Psi_1^T(k)\Psi_1(k)$$

$$-2\frac{\Psi_1^T(k)(\theta^T B_1 - C_1)}{B_1^T \hat{\theta}(k)\hat{\theta}^T(k) B_1 + c_c} + 2\Psi_1^T(k)\Psi_2(k) + 2\Psi_1^T(k)\varepsilon(k) + \Psi_2^T(k)\Psi_2(k) - \frac{2\Psi_2^T(k)(\theta^T B_1 - C_1)}{B_1^T \hat{\theta}(k)\hat{\theta}^T(k) B_1 + c_c}$$

$$-\frac{2\varepsilon^T(k)(\theta^T B_1 - C_1)}{B_1^T \hat{\theta}(k)\hat{\theta}^T(k) B_1 + c_c} + \frac{(\theta^T B_1 - C_1)^T(\theta^T B_1 - C_1)}{\left(B_1^T \hat{\theta}(k)\hat{\theta}^T(k) B_1 + c_c\right)^2} + 2\Psi_2^T(k)\varepsilon(k) + \varepsilon^T(k)\varepsilon(k) - e^T(k) e(k) \qquad \text{(A.2)}$$

Next substitute the parameter update law (10) in ΔV_2 of (A.1), to obtain

$$\Delta V_2 = \frac{1}{\alpha} tr\left\{\left[\left(I - \gamma \left\| I - \alpha\varphi(k)\varphi^T(k)\right\| I\right)\tilde{\theta}(k) - \alpha\varphi(x(k)) e^T(k+1)\right.\right.$$

$$\left.\left.+\gamma \left\| I - \alpha\varphi(k)\varphi^T(k)\right\| \theta)\right]^T \times \left[\left(I - \gamma \left\| I - \alpha\varphi(k)\varphi^T(k)\right\| I\right)\tilde{\theta}(k) - \alpha\varphi(x(k)) e^T(k+1) + \gamma \left\| I - \alpha\varphi(k)\varphi^T(k)\right\| \theta)\right] - \tilde{\theta}^T(k)\tilde{\theta}(k)\right\}$$

After some mathematical manipulation, the above equation could be rewritten as

$$\Delta V_2 = \frac{1}{\alpha} tr\{-2\gamma \left\| I - \alpha\varphi(k)\varphi^T(k)\right\| \tilde{\theta}^T(k)\tilde{\theta}(k) + \gamma^2 \left\| I - \alpha\varphi(k)\varphi^T(k)\right\|^2 \tilde{\theta}^T(k)\tilde{\theta}(k)$$

$$\underbrace{-2\alpha\tilde{\theta}^T(k)\left(I - \gamma \left\| I - \alpha\varphi(k)\varphi^T(k)\right\| I\right)\varphi(k) e^T(k+1)}_{1} \underbrace{+2\gamma \left\| I - \alpha\varphi(k)\varphi^T(k)\right\| \tilde{\theta}^T(k)\left(I - \gamma \left\| I - \alpha\varphi(k)\varphi^T(k)\right\| I\right)\theta}_{1}$$

$$+\alpha^2 e(k+1)\varphi^T(x(k))\varphi(x(k)) e^T(k+1) \underbrace{-2\alpha\gamma \left\| I - \alpha\varphi(k)\varphi^T(k)\right\| r(k+1)\varphi^T(x(k))\theta}_{1} + \gamma^2 \left\| I - \alpha\varphi(k)\varphi^T(k)\right\|^2 \theta^T \theta\}$$

Apply Cauchy-Schwarz inequality ($2ab \le a^2 + b^2$) to terms numbered as 1 in the above equation would reveal

$$\Delta V_2 \le \frac{1}{\alpha} tr\{-2\gamma \left\| I - \alpha\varphi(k)\varphi^T(k) \right\| \tilde{\theta}^T(k)\tilde{\theta}(k) + \gamma^2 \left\| I - \alpha\varphi(k)\varphi^T(k) \right\|^2 . \tilde{\theta}^T(k)\tilde{\theta}(k) + 2\delta\alpha\tilde{\theta}^T(k)\left(I - \gamma \left\| I - \alpha\varphi(k)\varphi^T(k) \right\| I \right).$$

$$\left(I - \gamma \left\| I - \alpha\varphi(k)\varphi^T(k) \right\| I \right)\tilde{\theta}(k) + (2+1/\delta)\gamma^2 \left\| I - \alpha\varphi(k)\varphi^T(k) \right\|^2 \theta^T\theta + \underbrace{(2+1/\delta)\alpha^2 e(k+1)\varphi^T(x(k))\varphi(x(k))e^T(k+1)}_{1} \Bigg\}$$

where $\delta > 0$ is a constant. Next, substitute the residual dynamics (14) to the term numbered as 1 in the above equation and apply the Cauchy-Schwarz inequality $(((a_1 + a_2 + ... + a_n)^T (a_1 + a_2 + ... + a_n) \le n(a_1^T a_1 + a_2^T a_2 + .. + a_n^T a_n)))$ to the same term and perform some mathematical manipulation to render the following equation

$$\Delta V_2 \le \frac{1}{\alpha} tr\Big[-2(2+\delta)\gamma \left\| I - \alpha\varphi(k)\varphi^T(k) \right\| \tilde{\theta}^T(k)\tilde{\theta}(k) + (1+2\delta)\gamma^2 \left\| I - \alpha\varphi(k)\varphi^T(k) \right\|^2 \tilde{\theta}^T(k)\tilde{\theta}(k) + 2\delta\tilde{\theta}^T(k)\tilde{\theta}(k) \Big]$$

$$+5(2+1/\delta)\alpha\varphi^T\varphi e^T(k)A_0^T A_0 e(k) + 5(2+1/\delta)\alpha\varphi^T\varphi\Psi_1^T(k)\Psi_1(k)$$

$$+5(2+1/\delta)\alpha\varphi^T\varphi\varepsilon^T(k)\varepsilon(k) + 5(2+1/\delta)\alpha\varphi^T\varphi \frac{(\theta^T B_1 - C_1)^T(\theta^T B_1 - C_1)}{\left(B_1^T\hat{\theta}(k)\hat{\theta}^T(k)B_1 + c_c \right)^2} + 5(2+1/\delta)\alpha\varphi^T\varphi\Psi_2^T(k)\Psi_2(k)$$

$$+(2+1/\delta)\frac{\gamma^2}{\alpha}\left\| I - \alpha\varphi(k)\varphi^T(k) \right\|^2 tr\left(\theta^T\theta\right) \qquad \text{(A.3)}$$

Next, the overall first difference of the Lyapunov function candidate, $\Delta V = \Delta V_1 + \Delta V_2$, can be obtained from (A.2) and (A.3) as

$$\Delta V \le \Big[e^T(k)k_V^T k_V e(k) + \Psi_1^T(k)\Psi_1(k) + \varepsilon^T(k)\varepsilon(k) + \underbrace{\Psi_2^T(k)\Psi_2(k)}_{1}$$

$$+ \underbrace{\frac{(\theta^T B_1 - C_1)^T(\theta^T B_1 - C_1)}{\left(B_1^T\hat{\theta}(k)\hat{\theta}^T(k)B_1 + c_c \right)^2}}_{2} \Big] - \frac{1}{5}e^T(k)e(k) + \frac{1}{\alpha} tr\Big[-2(2+\delta)\gamma \left\| I - \alpha\varphi(k)\varphi^T(k) \right\| \tilde{\theta}^T(k)\tilde{\theta}(k)$$

$$+(1+2\delta)\gamma^2 \left\| I - \alpha\varphi(k)\varphi^T(k) \right\|^2 \tilde{\theta}^T(k)\tilde{\theta}(k) + 2\delta\tilde{\theta}^T(k)\tilde{\theta}(k) \Big] + 5(2+1/\delta)\alpha\varphi^T\varphi e^T(k)A_0^T A_0 e(k)$$

$$+5(2+1/\delta)\alpha\varphi^T\varphi\varepsilon^T(k)\varepsilon(k)+\underbrace{5(2+1/\delta)\alpha\varphi^T\varphi\frac{(\theta^TB_1-C_1)^T(\theta^TB_1-C_1)}{\left(B_1^T\hat{\theta}(k)\hat{\theta}^T(k)B_1+c_C\right)^2}}_{2}+5(2+1/\delta)\alpha\varphi^T\varphi\Psi_1^T(k)\Psi_1(k)$$

$$+\underbrace{5(2+1/\delta)\alpha\varphi^T\varphi\Psi_2^T(k)\Psi_2(k)}_{1}+(2+1/\delta)\frac{\gamma^2}{\alpha}\left\|I-\alpha\varphi(k)\varphi^T(k)\right\|^2 tr\left(\theta^T\theta\right) \quad \text{(A.4)}$$

Consider only terms numbered as 1 in (A.4), we have the following equation

$$\left(1+5(2+1/\delta)\alpha\varphi^T\varphi\right)\Psi_2^T(k)\Psi_2(k)=\left(1+5(2+1/\delta)\alpha\varphi^T\varphi\right).\frac{\left(\tilde{\theta}^T(k)B_1-C_1\right)^T\left(\tilde{\theta}^T(k)B_1-C_1\right)}{\left(B_1^T\hat{\theta}(k)\hat{\theta}^T(k)B_1+c_C\right)^2}$$

$$\le\left(1+5\alpha(2+1/\delta)\varphi^T\varphi\right)\left(\tilde{\theta}^T(k)B_1-C_1\right)^T\left(\tilde{\theta}^T(k)B_1-C_1\right)$$

$$\le 2\left(1+5\alpha(2+1/\delta)\varphi^T\varphi\right)\left[B_1^T\tilde{\theta}(k)\tilde{\theta}^T(k)B_1+C_1^TC_1\right] \quad \text{(A.5)}$$

Next, consider only terms numbered as 2 in (A.4), we have the following equation

$$\left(1+5\alpha(2+1/\delta)\varphi^T\varphi\right)\frac{(\theta^TB_1-C_1)^T(\theta^TB_1-C_1)}{\left(B_1^T\hat{\theta}(k)\hat{\theta}^T(k)B_1+c_C\right)^2}\le$$

$$\left(1+5\alpha(2+1/\delta)\varphi^T\varphi\right)\left[B_1^T\theta\theta^TB_1-2C_1^T\theta^TB_1+C_1^TC_1\right] \quad \text{(A.6)}$$

Consider Lemma 1 and multiply $\left(1+5\alpha(2+1/\delta)\varphi^T\varphi\right)$ throughout (14) and using it along with (A.5) and (A.6) in (A.4), would render the following equation

$$\Delta V\le\left[e^T(k)A_0^TA_0e(k)+\Psi_1^T(k)\Psi_1(k)\right]-\frac{1}{5}e^T(k)e(k)$$

$$+\frac{1}{\alpha}tr\Big[-2(2+\delta)\gamma\left\|I-\alpha\varphi(k)\varphi^T(k)\right\|\tilde{\theta}^T(k)\tilde{\theta}(k)$$

$$+(1+2\delta)\gamma^2\left\|I-\alpha\varphi(k)\varphi^T(k)\right\|^2\tilde{\theta}^T(k)\tilde{\theta}(k)+2\delta\tilde{\theta}^T(k)\tilde{\theta}(k)\Big]$$

$$+\left(1+5\alpha(2+1/\delta)\varphi^T\varphi\right)\varphi^T\varphi e^T(k)A_0^TA_0e(k) \quad +\left(1+5\alpha(2+1/\delta)\varphi^T\varphi\right)\Psi_1^T(k)\Psi_1(k)$$

$$+(2+1/\delta)\frac{\gamma^2}{\alpha}\left\|I-\alpha\varphi(k)\varphi^T(k)\right\|^2 tr\left(\theta^T\theta\right)+2\left(1+5\alpha(2+1/\delta)\varphi^T\varphi\right)\left[B_1^T\tilde{\theta}(k)\tilde{\theta}^T(k)B_1+C_1^TC_1\right]$$

$$+\left(1+5\alpha(2+1/\delta)\varphi^T\varphi\right)\left[B_1^T\theta\theta^TB_1-2C_1^T\theta^TB_1+C_1^TC_1\right]+d_0+d_1\|e(k)\|^2+d_2\left\|\tilde{\theta}(k)\right\|^2+\underbrace{d_3\|e(k)\|\left\|\tilde{\theta}(k)\right\|}_{1} \quad \text{(A.7)}$$

where $d_0=\left(1+5\alpha(2+1/\delta)\varphi^T\varphi\right)\overline{d}_0$, $d_1=\left(1+5\alpha(2+1/\delta)\varphi^T\varphi\right)\overline{d}_1$, $d_2=\left(1+5\alpha(2+1/\delta)\varphi^T\varphi\right)\overline{d}_2$, and $d_3=\left(1+5\alpha(2+1/\delta)\varphi^T\varphi\right)\overline{d}_3$. Apply Cauchy-Schwarz inequality ($ab\le a^2+b^2$) to the term numbered as 1 in (A.7), then, take Frobenius norm in the above equation, therefore, the first difference of the Lyapunov function is given by

$$\Delta V\le-\delta_{1m}\|e(k)\|^2-\delta_{2m}\left\|\tilde{\theta}(k)\right\|^2$$

$$+2\left(1+5\alpha(2+1/\delta)\varphi^T\varphi\right)C_{1_{\max}}^2+\left(1+5\alpha(2+1/\delta)\varphi^T\varphi\right)\left(B_{1_{\max}}^2\theta_{\max}^2-2C_{1_{\min}}\theta_{\min}B_{1_{\min}}+C_{1_{\max}}^2\right)+d_0$$

$$+(2+1/\delta)\frac{\gamma^2}{\alpha}\left\|I-\alpha\varphi(k)\varphi^T(k)\right\|^2 tr\left(\theta^T\theta\right) \quad \text{(A.8)}$$

where $\delta_{1m}=\left(\frac{1}{5}-A_{0_{\max}}^2-5\alpha\varphi_{\max}^2(2+1/\delta)A_{0_{\max}}^2-\frac{3d_1}{2}\right)$ and

$$\delta_{2m}=\left(\frac{2(2+\delta)\gamma}{\alpha}\left\|I-\alpha\varphi(k)\varphi^T(k)\right\|-\frac{(1+2\delta)\gamma^2}{\alpha}\left\|I-\alpha\varphi(k)\varphi^T(k)\right\|^2\right.$$

$$\left.-\varphi_{\max}^2-\frac{2\delta}{\alpha}-5\alpha(2+1/\delta)\varphi_{\max}^4-2\left(1+5\alpha(2+1/\delta)\varphi_{\max}^2\right)B_{1_{\max}}^2-\frac{3d_2}{2}\right).$$ Next, take $B_{1_{\min}}=$

$$\frac{\left(\left(d_0+\frac{(2+1/\delta)\gamma^2}{\alpha}\left\|I-\alpha\varphi(k)\varphi^T(k)\right\|^2\theta_{\max}^2\right)/\left(1+5\alpha(2+1/\delta)\|\varphi(k)\|^2\right)\right)}{2C_{1_{\min}}\theta_{\min}}+\frac{\left(3C_{1_{\max}}^2+\theta_{\max}^2B_{1_{\max}}^2\right)}{2C_{1_{\min}}\theta_{\min}},$$

$B_{1_{\max}}=\frac{\delta}{\sqrt{2\left(1+5\alpha(2+1/\delta)\varphi_{\max}^2\right)}}$, also define $C_r=\frac{\alpha}{(2+1/\delta)}\left(\frac{2\delta}{\alpha}+\varphi_{\max}^2+5\alpha(2+1/\delta)\varphi_{\max}^4+\frac{3d_2}{2}\right.$

$$+2\left(1+5\alpha(2+1/\delta)\varphi_{\max}^{2}\right)B_{1_{\max}}^{2}\Big)$$

then, equation (A.8) could be rewritten as

$$\Delta V \le -\delta_{1m}\|e(k)\|^2 - \delta_{2m}\|\tilde{\theta}(k)\|^2 \qquad \text{(A.9)}$$

As long as the gains in (16)-(18) are satisfied, therefore, $\Delta V \le 0$ in (A.9), which shows stability in the sense of Lyapunov. Hence $e(k)$ and $\tilde{\theta}(k)$ are bounded, provided if $e(k_0)$ and $\tilde{\theta}(k_0)$ are bounded in the compact set *S*. Hence $e(k)$ and $\tilde{\theta}(k)$ converges to zero asymptotically.

REFERENCES

[1] P. M. Frank and L. Keller, "Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy – A survey and some new results", *Automatica*, vol. 26, pp. 459-474, 1990.

[2] J. Chen and R. J. Patton, *Robust Model-based Fault Diagnosis for Dynamic Systems*, Kluwer Academic publishers, MA, USA, 1999.

[3] C. Edwards, S. K. Spurgeon, and R. J. Patton, "Sliding mode observers for fault detection and isolation", *Automatica*, vol. 36, pp. 541-553, 2000.

[4] M. Massoumnia, G. C. Verghese, and A. S. Willsky, "Failure detection and identification", *IEEE Trans. on Automatic Control*, vol. 34, no. 3, pp. 316-322, 1989.

[5] M. A. Demetriou and M. M. Polycarpou, "Incipient fault diagnosis of dynamical systems using online approximators", *IEEE Trans. on Automatic Control*, vol. 43, no. 11, pp. 1612-1617, 1998.

[6] X. G. Yan and C. Edwards,"Nonlinear robust fault reconstruction and estimation using a sliding mode observer", *Automatica*, vol. 43, no. 9, pp. 1605-1614, 2007.

[7] C. De Persis and A. Isidori, "A geometric approach to nonlinear fault detection and isolation", *IEEE Trans. on Automatic Control*, vol. 46, no. 6, pp. 853 – 865, 2001.

[8] S. Narasimhan, P. Vachhani, and R. Rengaswamy, "Nonlinear residual feedback observer for fault diagnosis in nonlinear systems", *Automatica*, vol. 44, no. 9, pp.2222-2229, 2008.

[9] H. A. Talebi, S. Tafazoli, and K. Khorasani, "A recurrent neural-network-based sensor and actuator fault detection and isolation for nonlinear systems with application to the

satellite's attitude control subsystem", *IEEE Trans. on Neural Networks*, vol. 20, no.1, pp. 45- 60 , 2009.

[10] R. Isermann, "Model-based fault-detection and diagnosis–status and applications", *Annual Reviews in Control*, vol. 29, no.1, pp. 71-85, 2005.

[11] J. Luo, M. Namburu, K. Pattipati, L. Qiao, M. Kawamoto, and S, Chigusa, "Model-based prognostic techniques", *AUTOTESTCON 2003: IEEE Systems Readiness Technology Conference*, Anaheim, California, USA, pp. 330-340, 2003.

[12] J. Luo, A. Bixby, K. Pattipati, L. Qiao, M. Kawamoto, and S. Chigusa, "An interacting multiple model approach to model-based prognostics", *IEEE International Conference on Systems, Man and Cybernetics*, Washington, D.C., USA, vol. 1, pp. 189-194, 2003.

[13] E. Phelps, P. Willett, and T. Kirubarajan,"Useful lifetime tracking via the IMM", *Components and System Diagnostics, Prognostics, and Health Management II, Proc. of SPIE*, vol. 4733, pp. 145-156, 2002.

[14] M. J. Roemer and D. M. Ghiocel, "A probabilistic approach to the diagnosis of gas turbine engine faults", *53rd Machinery Prevention Technologies (MFPT) Conference*, Virginia Beach, VA, USA, pp. 325-336, 1999.

[15] Y. Shao and K. Nezu, "Prognosis of remaining bearing life using neural networks", *Proc. of the Institution of Mechanical Engineers, Part I: Journal of Sys. and Control Engg.*, vol. 214, no. 3, pp. 217-230, 2000.

[16] F. Caccavale and L. Villani, "An adaptive observer for fault diagnosis in nonlinear discrete-time systems", *Proc. of the American Control Conference*, Boston, MA, June 30 -July 2, 2004.

[17] F. Cacccavale, F. Pierri, and L. Villani, "Adaptive observer for fault diagnosis in nonlinear discrete-time systems", *ASME Journal of Dyn. Systems, Measurement, and Control*, vol. 130, no. 2, pp. 1-9, 2008.

[18] B. T. Thumati and S. Jagannathan, "An online approximator-based fault detection framework for nonlinear discrete-time systems", *Proc. of 46th IEEE Conference on Decision and Control (CDC)*, New Orleans, LA, USA, pp. 2608-2613, 2007.

[19] M. L. McIntyre, W. E. Dixon, D. M. Dawson, and I. D. Walker, " Fault identification for robot manipulators", *IEEE Trans. on Robotics and Automation*, vol. 21, no. 5, pp. 1028-1034, 2005.

[20] D. Dasgupta, *Artificial Immune Systems and Their Applications*, Springer, NY, 1998.

[21] J. Timmis, “Artifical immune systems- today and tomorrow”, *Nat. Comput.*, vol. 6, pp.1-19, 2007.

[22] D. Dasgupta,” Artificial neural networks and artificial immune systems: similarities and differences, *Proc. of the IEEE International Conference on Systems, Man, and Cybernetics*, Orlando, FL, USA, vol. 1, pp.873-878, 1997.

[23] D. Dasgupta, “Advances in artificial immune systems”, *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 40-49, 2006.

[24] C. C. Luh and W. C. Cheng, “Non-linear system identification using an artificial immune system”, *Proc. Instn. Mech. Engrs.*, vol. 215, part I, pp. 569- 585, 2001.

[25] J. Gomez, F. Gonzalez, and D. Dasgupta, “An immuno-fuzzy approach to anomaly detection”, *The 12th IEEE International Conference on Fuzzy Systems (FUZZ)*, pp. 1219-1224, 2003.

[26] C. C. Luh and W. C. Cheng, “ Immune model-based fault diagnosis”, *Mathematics and Computers in Simulation*, vol. 67, pp. 515-539, 2005.

[27] L. Shulin, Z. Jiazhong, S. Wengang, and H. Wenhu, ”Negative-selection algorithm based approach for fault diagnosis of rotary machinery”, *Proc. of the American Control Conference*, Anchorage, AK, USA, pp. 3955-3960, 2002.

[28] I. Aydin, M. Karakose, and E. Akin, “Artificial immune based support vector machine algorithm for fault diagnosis of induction motors”, *International Aegean Conference on Electrical Machines and Power Electronics (ACEMP)*, pp. 217-221, 2007.

[29] P. J. Costa Branco, J. A. Dente, and R. Vilela Mendes,”Using immunology principles for fault detection”, *IEEE Trans. on Industrial Electronics*, vol. 50, no. 2, pp.362-373, 2003.

[30] R. R. Sumar, A. A. R. Coelho, and L. dos S. Coelho,”Use of an artificial immune network optimization approach to tune the parameters of a discrete variable structure controller”, *Expert Systems with Applications*, vol. 36, pp. 5009–5015, 2009.

[31] A. R. Barrron, “ Universal approximation bounds for superposition of a sigmoidal function”, *IEEE Trans. on Information Theory*, vol. 39, no. 3, pp. 930-945, 1993.

[32] J. Zhang and A. J. Morris, “On-line process fault diagnosis using fuzzy neural networks”, *Intelligent Systems Engineering*, pp. 37-47, 1994.

[33] R. G. Bartle and D. R. Sherbert, *Introduction to Real Analysis*, Wiley, NY, USA, 1982.

[34] S. Jagannathan, *Neural Network Control of Nonlinear Discrete –time Systems*, CRC publications, NY, 2006.

[35] C. M. Kwan, D. M. Dawson, and F. L. Lewis," Robust adaptive control of robots using neural network: global tracking stability", *Proc. of the 34th conference on Decision and Control*, New Orleans, LA, pp. 1846-1851, 1995.

[36] P. M. Patre, W. MacKunis, K. Kaiser, and W. E. Dixon, " Asymptotic tracking for uncertain dynamics systems via a multilayer NN feed forward and RISE feedback control structure", *Proc. of the American Control Conference*, New York City, NY, USA, pp. 5989-5994, 2007.

[37] D. M. Dawson, Z. Qu, and S. Lim," Re- thinking the robust control of robot manipulators", *Proc. of the Conference on Decision and Control (CDC),* Brighton, England, pp. 1043-1045, 1991.

[38] F. L. Lewis, S. Jagannathan, and A. Yesilderek, "*Neural Network Control of Robotics and Nonlinear Systems*", Taylor and Francis, UK, 1999.

[39] B. T. Thumati and S. Jagannathan, " A model based fault detection and prognostic scheme for uncertain nonlinear discrete-time systems", *Proc. of 47th IEEE Conference on Decision and Control (CDC)*, Cancun, Mexico, pp. 392 - 397, 2008.

[40] Z. Li, "*Condition Monitoring of Axial Piston Pump"*, MS Thesis, 2005; http://library2.usask.ca/theses/available/etd-11252005-202705 /unrestricted/EricLithesis 2005NovA.pdf.

2. CONCLUSIONS AND FUTURE WORK

In this thesis, fault detection and prediction for rotating machinery is undertaken by utilizing schemes, which are normally applied to the areas of fault detection, and in ways other than their conventional usage. The first paper proposes a method for the use of PCA for prognosis. Both conventional (linear), PCA and kernel PCA were applied on water pump for the purpose of fault prognosis. The conventional PCA did not adequately model the fault dynamics and the by applying a Gaussian kernel, the results were significantly improved by 10%. This is due to the conventional PCA's inherent assumption that the data to be analyzed is linear. By comparison, Kernel PCA makes no such assumption, and, therefore, is better suited for nonlinear applications. This clearly illustrates a main weakness of data driven schemes: the results are only as good as the data being used to create the model.

In the second paper a new online fault detection schemes is proposed using an AIS as an OLAD for a class of nonaffine nonlinear discrete-time systems. Additionally, an estimator was designed to monitor the system and detect the occurrence of faults. This scheme was able to detect both the abrupt as well as incipient faults. The stability and online learning were demonstrated in two simulated applications: a two degree of freedom manipulator, and an axial piston pump. The FD scheme was later verified with an application to an axial piston pump test bed. From the experimental results, the FD scheme was able to detect and learn both incipient and abrupt faults online. Therefore, the FD scheme proposed in the second paper renders asymptotic stability in both simulation and in hardware.

In conclusion, while both data driven and model driven methodologies have merit. The performance of a data driven methodology is dependent on the suitability of

the data to the type of methodology to be applied (e.g. nonlinear data to be analyzed using linear techniques). In contrast, model based methodologies are more robust to system noise and disturbance as well as multiple operation modes than data based approaches. However, for a model based methodology to work satisfactorily, the model must adequately describe system dynamics. Such equations can be quite complex, computationally expensive, and difficult to obtain.

Future work utilizing in PCA in the context of FDP focus on determining how to evaluate the physical meaning if any of the principal components. Knowledge of the physical meaning of the principle components could provide insights as to which component(s) might be useful for the isolation and prediction of faults.

The use of AIS in the future should extend not only into the detection and learning of faults, but also into areas of fault isolation and prognosis. For such an application, a bank of isolators would need to be accumulated. After isolator bank has been created and an OLAD been placed in the appropriate state variables, isolation should be possible through careful observation of each OLAD behavior and comparison to the behavior modeled by the faults in the isolation bank.

VITA

Gary Ray Halligan was born December 29, 1981 in Springfield, Missouri. He earned his Bachelor of Science in Electrical Engineering in 2004 from the Missouri University of Science and Technology (formerly the University of Missouri-Rolla), and earned his Master of Science degree in Electrical Engineering in 2009, also from Missouri S&T.