MODEL- AND DATA-DRIVEN APPROACHES TO FAULT DETECTION AND ISOLATION IN COMPLEX SYSTEMS

By

Hamed Khorasgani

Dissertation

Submitted to the Faculty of the

Graduate School of Vanderbilt University in

partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

in

Electrical Engineering

January 31, 2018

Nashville, Tennessee

Approved:

Gautam Biswas, Ph.D.

Akram Aldroubi, Ph.D.

Don Wilkes, Ph.D.

Gabor Karsai, Ph.D.

Xenofon Koutsoukos, Ph.D.

ACKNOWLEDGMENTS

I gratefully acknowledge the support provided by Scott Poll, Mark Shirley, Peter Berg, and other members of the LADEE Mission Ops team from NASA Ames in acquiring, analyzing, and interpreting the LADEE Telemetry data. I would also like to thank Freddie Adom, and Glenn Mohon in the Department of General Services at metropolitan government of Nashville for their help in implementing the experimental test in the Lentz Public Health Center hot water system, and analyzing the results.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 Motivation

Complex engineering systems pervade every aspect of our daily lives. The need for increased performance, safety, and reliability of these systems provide the motivation for developing system health monitoring methodologies for these systems. For example, systems health monitoring is the central component of abnormal event management (AEM) in process engineering. Early fault detection and isolation during system operations can help system operators take timely actions to prevent abnormal event progression and to reduce down time and productivity loss. In extreme cases, timely fault detection and isolation can lead to avoidance of catastrophic situations, thus preserving system safety and reliability. Faults and abnormal events cost the petrochemical industries an estimated 20 billion dollars every year, and AEM is rated as the number one problem in the industry [176].

For space missions, safety and reliability are even more critical. Because of the complex structure of spacecraft and the unpredictability of the space environment, it is practically impossible to eliminate anomalies and fault occurrences by design, even when we employ state of the art reliability methods to design subsystems. Robust fault detection and isolation enables accompanying fault tolerant control units to react in a timely manner, thus reducing the possibility of damage and loss of the mission [70]. Therefore, designing and developing integrated systems health management (ISHM) methodologies have been a long-standing area of research in major industries as well as NASA [159].

Due to the advances in networking technology, along with the proliferation of complex cyber-physical systems (CPS), and the availability of inexpensive sensors and processors, we have witnessed a shift in focus from centralized to more distributed diagnosers within the fault detection and isolation (FDI) community in recent years. State of the art health

monitoring approaches have to be applied to systems with large number of distributed components that exhibit complex, hybrid behaviors. Hybrid system behaviors are represented as continuous behaviors interspersed with discrete mode transitions [172]. For such systems, the system diagnostics models are not easy to develop, and keep updated during the system life-cycle. Therefore, reliable models of these systems are not always available. Even when models are available, they are often incomplete and plagued by uncertainties in tracking system behavior.

To address these issues, it is important to develop diagnosis methods that are robust to model uncertainties and measurement noise. In situations where models are not very accurate, their imperfections and incompleteness can be overcome by supplementing them with additional operational data from the system. A robust monitoring algorithm can detect and isolate faults in the presence of noise and uncertainties in the measurements and system model. On the other hand, the lack of available models, or models of poor quality may indicate the need to move toward data-driven approaches for diagnosis. In this thesis, we will design and develop diagnosers that handle a number of these challenges.

1.2 Challenges

Fault detection and isolation (FDI) in complex systems with large number of components can be a complex and challenging task. In this section, we review the challenges in robust model-based FDI with extensions to hybrid and distributed systems. We also study challenges that appear in data-driven diagnosis methods.

1.2.1 Robust model-based FDI

To avoid high false alarm rates (FAR) and high missed alarm rates (MAR) due to the uncertainties in the model, disturbances generated in the environment, and measurement noise, it is necessary to design robust diagnosis algorithms. However, designing a robust FDI approach is challenging for the following reasons.

- **Quantifying the effects of noise and uncertainty on residuals.** It is not trivial to quantify the effects of noise and uncertainties on fault detection and isolation performance, especially for complex, nonlinear systems. The distributions of noise and uncertainties are generally unknown, and even when they are known, the nonlinearity in the system makes it challenging to estimate their effects on the system's measurements and the residuals.

- **Generating robust residuals.** In many cases, noise, uncertainties and system faults have similar effects on system behavior. Therefore, it is challenging to reduce the effects of noise and uncertainties on the detectors while making sure they are sufficiently sensitive to faults.

- **Residual selection.** The total number of residuals for fault detection and isolation grows exponentially as the number of measurements [5, 167]. Since the sensitivity of the residuals to faults and uncertainties can vary, different sets of residuals provide different performance for fault detection and isolation. Moreover, for a given behavior trajectory, the performance of residuals can vary from one operating region to another for the system. Because of the large number of possible residual candidates and their variable performance in different operating regions of the system, especially when the system is nonlinear, it is challenging to select a set of residuals with the acceptable performance for the behavior trajectory over the entire operating regions for a system.

1.2.2 Distributed model-based FDI

For reliability and practical reasons, developing distributed diagnosers may be favored for FDI in complex systems. However, designing a distributed FDI approach has the following challenges over and above the ones we listed for continuous systems.

- **FDI among subsystems.** It may be hard to differentiate the change in behavior due

to a fault that occurs in a specific subsystem from change in behavior caused by a fault in neighboring subsystems. This can increase false alarm rates and decrease the reliability of the health monitoring algorithms that are employed.

- **Global accuracy with minimum communication.** Transferring the collected sensor information from one subsystem to another can be expensive and error prone. On the other hand, each subsystem typically has access to few local measurements. Fewer measurements in general means fewer redundancies, which decreases the chances for unique fault isolation. In addition, this can increase missed alarm rates when compared to centralized approaches. It is a challenge to solve the problem of ensuring that each subsystem diagnoser provides the correct results while keeping the communication between subsystems to a minimum.

- **Limited information from neighboring subsystems.** Subsystems of a complex system are designed by different manufacturers, who may not be willing to pass along all of their knowledge of the subsystems to the system integrator for intellectual property reasons. It is challenging to design globally correct diagnosers when limited information is available of the interactions between subsystems.

1.2.3 Model-based FDI in hybrid systems

Many complex systems, such as automobiles [172], and aircraft [41] are hybrid in nature, where continuous behavior evolution is interspersed with discrete mode changes. The discrete mode changes make diagnosis of hybrid systems much more challenging than diagnosis of continuous systems for the following reasons.

- **Number of modes and possible trajectories.** The total number of modes in a hybrid system is exponential in the number of discrete switches. Therefore, it is computationally intractable to pre-enumerate all the operational modes of a complex system and to design diagnosers that provide correct results for each mode. Tracking all

the possible trajectories of a hybrid system is even more challenging, since the complexity grows exponentially as the number of modes. The complexity becomes even worse, when one considers trajectories that include possible faults in the system.

- **Distinguishing mode changes and faults.** It is challenging to differentiate the change in behavior due to a fault from change in behavior caused by a mode transition in hybrid systems. In fact, a mode change can be detected as a fault and a fault can make mode estimation more complicated.

1.2.4 Data-driven anomaly detection

Most of health monitoring methods rely on a system behavior model. When these models are not available for diagnosis or the available models are incomplete, data-driven methods can be an alternate approach to link faults to measurement deviations. However, designing data-driven diagnosers is challenging for the following reasons.

- **Incomplete and corrupt data.** Available data can be noisy or it may be corrupted during collection and storage. The sampling rate may vary between the different measurements and data may be lost during transmission and storage. Moreover, in the real world, we usually do not have access to training data to learn the system's normal behaviors in all of the different operating modes.

- **High dimensionality and irrelevant variables.** High dimensionality of the datasets in complex systems is another challenge for anomaly detection. High dimensionality in the data increases the required time and space for processing the data, and can mask the effect of faults. Typically, many variables in the dataset will be redundant or irrelevant to a particular fault. The irrelevant variables may hurt anomaly detection by acting as noise and hiding effects on the relevant variables. Redundant measurements may artificially enhance some effects and, therefore, decrease the effect of others, making some faults hard to detect [119].

1.3 Contributions

This thesis research addresses a number of challenges in model- and data-driven analyses that were discussed in Section 1.2. Overall, the contributions of this thesis research can be primarily categorized into: 1) model-based diagnosis contributions, and 2) data-driven diagnosis contributions. Our contributions to the model-based diagnosis are summarized below.

1.3.1 Robust model-based FDI

Most diagnosis approaches perform residual generation and residual selection simultaneously [140, 154]. However, since our focus is on choosing an optimal set of residuals, our method generates the entire set of residuals and then applies a methodology to select a subset of residuals that has sufficient robustness to noise and uncertainties. For residual generation we use the fault diagnosis toolbox developed by Frisk and Krysander[1] [68]. The contributions of our work in residual selection are

- **Quantifying residual performance.**

    1. *Derivative-based sensitivity analysis approach.* We use derivative-based sensitivity analysis [144] to define two quantitative measures, the detectability and isolability ratios, to evaluate and quantify the performance of individual residuals in fault detection and isolation in the presence of uncertainties. Unlike distinguishability measures [50], detectability and isolability ratios can be used for nonlinear dynamic systems with multiplicative faults.

    2. *Global sensitivity analysis approach.* The derivative-based approach is computationally efficient. However, the derivative-based approach only determines the effect of uncertainties at the single point at which the derivative is constructed. For linear and smooth nonlinear systems, the effect of uncertainties in

[1] https://faultdiagnosistoolbox.github.io/

other operation points can be easily determined by extrapolation. For stiff nonlinear system this can lead to a significant error. To overcome to this problem, we have developed a global sensitivity analysis method [175] to define global detectability and global isolability ratios to quantify residual performance.

- **Residual selection.** We propose two algorithms to select subsets of residuals that fulfill specified performances for fault detection and isolation.

  1. *Off-line Residual Selection:* when the system's trajectory is available, our algorithm divides this trajectory into regions, such that the set of residuals that have sensitivity values above a pre-specified threshold remain the same in a region, but vary across the different regions. The selection of a minimum number of residuals that meet the robustness and sensitivity criteria is formulated as a binary integer linear programming (BILP) optimization problem [182].

  2. *Dynamic Residual Selection:* for the cases where the system's trajectory is unknown, an efficient dynamic residual selection algorithm is proposed. This algorithm removes residuals when their performance drop below the threshold. They are then replaced by residuals that provide the highest performance ratios in the current region. This guarantees the required performance is maintained for any trajectory.

1.3.2 Distributed model-based diagnosis

We develop two approaches to address the problem of distributed fault detection and isolation in large scale systems with several subsystems. In method one, we generate the residuals first and then select a set of residuals for each subsystem that guarantee full diagnosability and minimum communication of measurements among subsystems. We formulate this problem as a BILP optimization problem. Note that this algorithm addresses

the first two challenges presented in subsection 1.2.2. The contributions of this approach compared to previous work are

- **Minimum communication.** Unlike previous work [150] that only guarantees minimal shared variables among subsystems, our algorithm guarantees that the subsystems share the minimum number of measurements, implying that we minimize the communication of measurement streams across subsystems of the global system. This is important because transmitting the data to other subsystems can be costly in large scale systems.

- **Easy to extend to robust distributed diagnosis.** Generating all the residuals beforehand, makes robustness analysis and, therefore, robust distributed residual selection possible. We can compute the detectability and isolability ratio for each residual and select the residuals with acceptable robustness performance.

The second method instead of generating all the residuals, works directly with the system equations and uses a matching algorithm to find a minimal set of measurements and equations from neighboring subsystems that generate enough redundancies to make all the faults diagnosable. The contributions of this approach are as follows.

- **Computational efficiency.** There are polynomial solutions for the matching algorithm [179]. Therefore, the second algorithm is computationally efficient.

- **No need for the global model.** This algorithm only searches among neighboring subsystems and, therefore, does not need to use the global model in the design process of the supervisory system. This makes the algorithm more practical, especially for the complex systems.

1.3.3 Model-based diagnosis for hybrid systems

We develop an algorithm for fault detection and isolation in hybrid systems. Our proposed approach consists of two parallel algorithms; 1) mode detection, and, 2) fault detection and isolation in each operating mode. The contributions of our approach are

- **Mode detection in the presence of faults.** We extend previous work on mode detection in hybrid systems [43], and propose an algorithm that, if it is structurally feasible [58, 173], selects a minimal set of equations to solve the mode identification problem in nonlinear hybrid systems in the presence of system faults. By detecting the operating mode, the diagnoser unit does not need to track all the possible trajectories which is a primary challenge in hybrid diagnosis. Moreover, mode detection in the presence of faults addresses the challenge of distinguishing mode changes from faults.

- **On-line residual generation.** Our approach does not need to pre-compile the residuals for every possible mode of the hybrid system which can be computationally intractable. Therefore, it does not have to pre-enumerate all the possible modes which is exponential in number of discrete variables in the model. Instead, our approach updates the diagnoser when the system switches to a new operating mode.

The second part explores data-driven methods to address the monitoring problem when the system model may be incomplete, outdated, or unavailable. Our contributions to data-driven diagnosis is summarized below.

1.3.4 Unsupervised anomaly detection

We review the literature and develop anomaly detection methods that include the following steps: 1) data pre-processing, and generation of the feature space for anomaly detection, 2) applying a clustering algorithm and determining regions of nominal behavior, and by extension outliers and anomalous data points, 3) associating significant features

with the outlier groups and consultation with experts in order to identify and characterize special modes of operation as well as anomalous behavior of the system. Our contributions to the unsupervised anomaly detection are:

- **Anomaly detection in time series data.** In this work, we extend previous work [124] to detect anomalies in one-off long term space missions. Toward this end, we derive a set of objects from a curated set of time series data. Our approach divides the time series representing the entire mission trajectory into segments based on the application. Each segment represents a data object. In our case study each segment represents a dark or light period on the mission time line. The time interval width (the window size) is variable depending on the period of each dark or light interval, therefore, each object can have a different number of samples.

  In the next step, we apply a wavelet transform [24] to each time series signal to extract a set of discrete features for each object. The wavelet transform captures the time-frequency characteristics of signal waveforms, i.e., it captures the frequency characteristics of the signal at different time intervals in the signal. Note that selecting few wavelet coefficients as the features for each object also helps to mitigate the effects of noise in the anomaly detection process.

- **Feature selection.** Unlike [131], our approach does not require prior knowledge about the system parameters and variables. Therefore, our approach is more general and can easily be extended to anomaly detection in time series datasets in different domains. A simple solution to the problem of possible redundant features in an unknown dataset is to remove any variable that has high correlation with another variable. However, the redundant variables can represent redundancies in the system and, therefore, they may provide critical information for anomaly detection.

  To capture this information, we add a residual to the dataset after removing each variable. The irrelevant variables also hurt anomaly detection by acting as noise

and hiding effects of the relevant variables. In this work, we consider the minimum number of waveforms or variables that represents an specific percentage of the total variance as the relevant variables. Using this approach, we also automatically remove the residuals that are generated from identical variables and do not contribute any meaningful information.

- **Defining significant features.** Our anomaly detection approach combines unsupervised learning methods with human-expert support to analyze initial anomalies detected in the data. Therefore, unlike other clustering-based anomaly detection methods (see [114] for example), our approach can distinguish faulty events from special operating modes. To facilitate an expert's task in identifying anomalies, we define *significant features*, as the set of features that best differentiate each anomalous group from the nominal groups. Our experience in anomaly detection in lunar atmosphere and dust environment explorer (LADEE) spacecraft, show these features help the human experts to better understand and characterize the anomalous situation as potential faults, or special modes of operation.

1.4 Organization of this Dissertation

The rest of this dissertation is organized as follows. Chapter II reviews model-based fault detection and isolation approaches in complex dynamic systems. This chapter reviews previous work in the robust model-based FDI, distributed model-based fault diagnosis, and model-based fault detection and isolation in hybrid systems. Chapter III presents our proposed approach for robust diagnosis in nonlinear dynamic systems in the presence of noise and model uncertainties. Chapter IV formulates the distributed fault detection and isolation problem and presents our algorithms for distributed diagnosis. Chapter V discusses fault detection and isolation for hybrid systems and presents our approach to diagnosis of hybrid systems.

Chapter VI reviews data-driven anomaly detection techniques. The data-driven anomaly

detection approaches are categorized based on their input data labels into three main groups: 1) supervised, 2) semi-supervised, and 3) unsupervised. The unsupervised anomaly detection approaches are more widely applicable to the real world problems and, therefore, are discussed in more detail. This chapter presents different information theory measures and clustering algorithms that can be used in an unsupervised environment. Moreover, it presents feature learning and feature selection methods for unsupervised anomaly detection. Chapter VII presents our approach to data-driven diagnosis, i.e., the anomaly detection problem, and then discusses our contributions in this topic. Chapter VIII summarizes the contributions of this research, and presents the future work.

CHAPTER II

BACKGROUND ON MODEL-BASED FAULT DETECTION AND ISOLATION (FDI)

A residual is an analytical redundancy relation between known variables in the system such as parameters of the system, process measurements, and inputs. In model based approaches, to detect a fault f, we need a residual sensitive to the fault and, to isolate a fault f_i from another fault f_j we require a residual sensitive to f_i and at the same time insensitive to f_j [62]. There are three main approaches for residual generation 1) observer-based [3, 63], 2) identification [91, 92], and 3) parity equations and analytical redundancy relations (ARR) [73, 163]. Residual generation involves eliminating unknown variables from a set of equations till a relation is established between measured variables and fault parameters. Observer-based approaches use different linear or nonlinear observers to estimate the unknown variables. They typically reconstruct measurements of the system with the aid of an observer using a mathematical model of the system and makes the decision on possible faults in the system on the basis of the analytical redundancy thus being created. Even though there are a number of different design procedures, the core of the diagnostic approach is always observers or estimation filters such as Kalman filters or particle filters [63].

On the other hand, identification approaches perform fault detection and isolation by on-line parameter estimation. Identification approaches are especially useful for fault identification [61, 141]. In theory, little attention has been paid to the identification approaches. This is probably due to the fact that the existing parameter estimation theory can readily be applied to fault diagnosis without major modifications [63]. The parity equations and analytical redundancy relations approaches derive a residual from a set of overdetermined equations by developing a computation sequence for computing and eliminating

the unknown variables. The equivalence between observer-based approaches and parity space approaches has been proven for linear systems [127] and nonlinear state-affine systems [98]. There are several algorithms to extract analytical redundancies from the system model. Pulido and Alonso [147] use possible conflicts to find minimal redundant sets, Krysander et al. [106] designed an algorithm for finding minimally overdetermined sets (MSOs) of constraints and Travé-Massuyes et al. [173] proposed an algorithm to generate ARRs from linear system models. The equivalence between PCs, MSOs and ARRs from a structural diagnosability point of view has been shown in the literature [5, 22]. In this work, we focus on the MSO-algorithm [106] as a particular general technique to extract analytical redundancies in system models.

Krysander et al. [106] use the Dulmage Mendelsohn (DM) decomposition [46] to design an algorithm for finding minimally overdetermined sets of constraints. Consider a dynamic system model as a bipartite graph where the set of equations, E, and the set of variables, V, represent the two disjoint vertex sets and each variable $v \in V$ connects to equation $e \in E$ if v appears in e in the system dynamic model. The DM decomposition uses a bipartite matching algorithm [179] to partition the vertices of the bipartite graph, variables and equations, into subsets, with the property that two adjacent vertices belong to the same subset if and only if they are paired with each other in a perfect matching of the graph. The perfect matchings represent the exactly determined part of the system. Application of the DM decomposition also extracts under determined, and over determined parts of the system model [59].

The over determined part introduces redundancy in the system and can be used for residual generation for fault detection and isolation. Frisk and Nyberg [69] argued that minimal over determined sets use fewer parameters from the system model and fewer measurements from the sensors and, therefore, a residual generated from a minimal structurally overdetermined (MSO) set tends to be more robust against model uncertainties. A residual can be

derived from a MSO by developing a computation sequence for computing and eliminating the unknown variables till a relation is established between known variables and fault parameters. When there are no implicit equations and algebraic loops in the equation set, eliminating unknown variables is straight forward.

However, for nonlinear systems with non-invertible equations, algebraic loops, and implicit equations generating a residual by eliminating the unknown variables is challenging, and in some cases not possible in a closed form. Zhang, et al. [186] used Ritt's algorithm to eliminate unknown variables in nonlinear dynamic systems. Their approach generates residuals in derivative causality and second and higher order time derivatives of inputs and output measurements may need to be estimated, and that can be error prone. Dustegor, et al. [47] used a matching algorithm to derive computational sequences to solve for the unknowns. Svard and Nyberg [166] extended the approach to dynamic equations and included both differentiation and integration in the same solver to generate the maximize number of possible residuals.

2.1 Robust Fault Detection and Isolation

Residuals represent redundancies in the system equations, and they can form the basis for fault detection and isolation. Ideally, each residual should compute to a value of zero in the fault-free case, and residuals sensitive to a fault become nonzero when the fault occurs. Therefore, a residual r_a sensitive to fault f_i can be monitored to detect f_i. Note that r_a could also be sensitive to other faults in the system (say f_j for example). To isolate fault f_i from fault f_j in residual-based approach, we need a residual, r_b, sensitive to f_i and insensitive to f_j. When r_a becomes nonzero we detect f_i and f_j as possible fault candidates, and when r_b becomes nonzero we can isolate f_i from f_j. Due to model uncertainties and measurement noise, a residual may deviate from zero even in the fault-free case. Therefore, noise and uncertainty can make fault detection and fault isolation tasks much more challenging.

To make a diagnoser robust to noise and uncertainties, typically hypothesis tests are

used to determine if a a residual deviation is statistically significant. A common type of hypothesis test is the use of a threshold, where the threshold can be established using more or less sophisticated methods, such as maximum likelihood estimators, cumulative sum control chart (CUSUM) [11] or Z-test [18]. The decision logic detects and isolates the possible faults based on the hypothesis test outputs. When the effects of noises and uncertainties are significant, we need to design conservative hypothesis tests to avoid false alarms. However, using conservative hypothesis tests could increase detection time or even makes a set of faults undetectable.

To avoid this problem, geometric [129], unknown input observer [31, 177], parity space [74], eigenstructure [6], and frequency domain [178] approaches have been developed to generate residuals that are perfectly decoupled from uncertainties and disturbances. Frisk and Nyberg [69] presented sufficient condition for perfect decoupling in linear systems. When perfect decoupling is not possible, optimization methods such as the H_∞ optimization and linear matrix inequality (LMI) approaches have been used to design robust residuals that minimizing the effect of uncertainties and disturbances [190]. Zhong et al [189] apply an observer-based residual generation approach for fault detection in LTI systems with additive faults and uncertainties. To achieve robust performance, the authors propose to design the observer gain matrix and the residual weighting matrix in a way that maximizes the effect of faults and minimizes the effect of disturbances on the residuals at the same time. The solution of the optimization problem is then presented via a LMI formulation.

These decoupling and optimization methods generally apply to linear time invariant (LTI) systems with additive and abrupt faults. However, most real systems are nonlinear and time varying with possible multiplicative faults and LTI and additive fault assumptions limit the applicability of decoupling and optimization methods to real problems. To extend the scope of robust model based diagnosis, several robust fault detection methodologies for specific classes of nonlinear systems have been developed (e.g., [64, 100]). However, these

approaches are only applicable to specific classes of nonlinear systems and they don't provide a general robust residual generation solution. Moreover, the performance of residuals can vary from one operating region to another. Therefore, designing robust residuals with acceptable performance for any given trajectory is not feasible. To address robust FDI in nonlinear systems, we quantify the performance of residuals in fault detection and isolation over the known operating region of the system. We then select a set of residuals that meet the performance requirements for each operating region.

There are two main approaches to quantify residual performance for fault detection and isolation in the presence of uncertainty: 1) stochastic methods, 2) sensitivity analysis methods. In the stochastic approaches, the distance between a residual probability distribution in the presence of fault and in the fault free case is the measure to quantify the performance of residual for detecting the fault. Note that a distance measure does not have to have the strict definition of distance in metric space. For example, a distance measure may not satisfy triangle inequality or it could be asymmetrical. Several distance measures have been developed for stochastic models [9], among which Kullback-Liebler (KL) divergence is one of the more prevalent measures used for fault detection [10].

Erikson et al. [50] have derived a measure called *distinguishability*, to evaluate fault diagnosability performance for linear discrete-time dynamic systems. The approach uses the KL divergence measure along with the stochastic characterizations of the different fault modes and provides a quantified measure for fault detectability and isolability in a linear dynamic system for a given set of sensors with associated noise distributions. In [51], Erikson et al. propose an on-line sequential test selection strategy where the detectability performance of each residual is quantified using the distinguishability measure. Erikson and Sundstrom [52] used distinguishability to optimize the design of residuals. The stochastic approach provides fairly accurate estimation of residual performance. However, those methods usually need faults and uncertainties to be characterized by probability distribution functions (PDFs) and they are computationally expensive. Moreover, it is challenging to

apply them to nonlinear systems as well as situations where the faults and uncertainties are multiplicative. To overcome the computational problems, several researchers have adopted sensitivity analysis methods to evaluate the performance of residuals.

Djeziri et al. [42], estimated the effects of uncertainties on residuals using sensitivity analysis to design adaptive thresholds. Perez et al. [143] used sensitivity analysis to localize leakage in water distribution networks. Blesaa et al. [19] proposed a sensor placement algorithm to achieve maximum leak detectability and isolability in water distribution networks and defined a measure, *robustness percentage index*, to evaluate the robustness of the methodology. In this work, we use sensitivity analysis to define *detectability ratio* and *isolability ratio* measures that quantify the performance of residuals in nonlinear dynamic systems. In addition to residual performance quantification, detectability and isolability ratios are used to define dynamic systems operating regions, in order to select a set of residuals that meet pre-specified detectability and isolability performance in each region. Moreover, we define global detectability and global isolability ratios for stiff nonlinear dynamic systems with measurement noise.

2.1.1 Summary

Several methods such as unknown input observer and parity space approaches have been developed to generate residuals that are perfectly decoupled from uncertainties and disturbances in LTI systems. When perfect decoupling is not possible, optimization methods such as the H_∞ optimization and LMI approaches have been used to design robust residuals for these systems. For nonlinear systems there is no general approach to decouple uncertainties and disturbances. Moreover, the performance of residuals can vary from one operating region to another. Therefore, designing robust residuals with acceptable performance for any given trajectory is not feasible. To address robust FDI in nonlinear systems, we use sensitivity analysis to quantify the performance of residuals in fault detection and

isolation. We then select a set of residuals with acceptable performance for each operating region.

2.2 Distributed Fault Detection and Isolation

Traditional approaches develop centralized diagnosers for complex systems, e.g., the aircraft diagnostic and maintenance systems (ADMS) used on modern aircraft systems [6, 162]. However, as the complexity and size of systems, such as aircraft, automobiles, power plants, and manufacturing processes, have grown, distributed approaches to fault detection and isolation in large dynamic systems with many subsystems have become important [110, 112, 161]. Safety-critical systems must detect and isolate faults quickly and reliably to enable effective safety maneuvers and fault tolerant control so as not to endanger operations and human lives [44]. Transferring all of the collected sensor information to a central fault detection and isolation unit can be expensive and error prone. Centralized diagnosers may also be less reliable because they create a single point of failure.

Networking delays can also affect the timeliness of diagnosis decisions [57]. Transmission delays not only increase detection time, but can also affect the order of detection, which can further affect diagnostic accuracy. Detection time is important for the safe and reliable operation of safety-critical systems. Faster fault detection and isolation enables accompanying fault tolerant control units to react in a timely manner, thus reducing damage and down time of systems. The computational intractability of building centralized diagnosers for the large systems is another important reason to develop distributed solutions for FDI problems. Also, from practical view point, subsystems of a complex system are designed by different manufacturers, who may not be willing to pass along all of their knowledge of the subsystems to the system integrator for intellectual property reasons. Therefore, a centralized approach to fault detection and isolation problem may not be possible nor is it desirable in many cases.

In the literature, there has been considerable effort in developing distributed fault detection and isolation in the context of discrete event systems. In the most simple case, a group of distributed fault detection and isolation approaches consider each subsystem as a node that could be "OK" or "faulty" without attributing any dynamics to the system behavior. This approach is especially useful in wireless sensor networks [153] and computer networks [149]. In these systems each node has a specified operating range and can observe the other nodes performances within its operating range. When a node detects an inconsistency between its performance and a neighboring node in the network, for example, two sensors monitor the environment temperature differently, there could be a fault in the neighboring node or the node itself may be faulty.

To detect faulty nodes, Blough et al. [20] used a majority vote among the neighboring sensors to determine the status of each node. In [153], each node sends a signal to the coordinator (manager) whenever there is a state change. The coordinator uses the collected information to detect subsystem failures. In [97] each node identifies its own status to be either "OK" or "faulty" and the claim is then supported or reverted by its neighbors as they also evaluate the node behavior. Rish et al. [149] proposed an adaptive real-time distributed diagnosis approach for computer networks. Their approach uses an information-theoretic approach for test selection that speeds up real-time diagnosis by minimizing set of measurements, while maintaining high diagnosability.

In many real systems, distributed fault detection and isolation is more complicated. In these systems a subsystem has several components and a fault could occur in a sensor, actuator or other components in the subsystem. Therefore, it is not enough to simply declare a subsystem "OK" or "faulty" and we have to isolate the faulty components inside each subsystem. Deb et al. [40] proposed to design a local diagnoser for each subsystem where each local diagnoser considers all the inputs to the subsystem as potential faults. When a subsystem is fault free, all the inputs to the subsystem would be considered fault free and when a local diagnoser detects a fault in its subsystem, all the outputs from that subsystem

would be marked as faulty. Each subsystem receives update from its neighboring subsystems as soon as there is a change in their status. In this approach, it is possible that some faults stay undetectable or the distributed approach fails to isolate some fault pairs.

Roychoudhury et al. [150] developed an algorithm for dynamic systems to search for the minimal number of additional external measurements to add to each local diagnoser in order to make all the faults locally detectable and isolable. Their algorithm only guarantees minimal number of measurements and it does not guarantee the optimum solution. Daigle et al. [35] used the same approach for distributed fault detection in mobile robots. Bregon et al. [23] used breadth-first search to find the minimum number of measurements that we need to add to each subsystem to make all the faults globally detectable and isolable. The algorithm guarantees optimum solution, however, it is exponential in terms of the cardinality number of the system measurements. To address this problem, the authors proposed a greedy search algorithm which is computationally efficient but does not guarantee optimality of the solution.

Like centralized approaches, study of robustness of the distributed methods is critical and has to be considered. However, there are few researchers who have investigated robust distributed approaches. A common approach to address uncertainties in distributed fault detection is to design an adaptive threshold for each local diagnosis subsystem [187, 188]. Ferrari et al. [56, 57] proposed a distributed fault detection and identification approach where each subsystem uses an adaptive approximator [55] to estimate the dynamic of its neighbors and then the neighboring subsystems use a consensus-based estimator mechanism [139] to improve the detection and identify the fault. Because of the adaptive approximation, their approach is robust to uncertainties. However, they do not propose any algorithm to partition the system or to select shared variables between the subsystems.

2.2.1 Summary

Most of the distributed fault detection and isolation algorithms have been developed in the context of discrete event systems. Roychoudhury et al. [150] proposed an approach based on quantitative fault detection and qualitative fault isolation for distributed FDI in dynamic systems. Their algorithm produces globally correct diagnosis results with minimal exchange of information among the subsystems. However, it does not guarantee that the exchange of information among subsystems is globally minimum. To achieve robust FDI, the authors apply a hypothesis test to ensure observed deviations are statically significant. Relying on hypothesis tests to achieve robustness without designing robust residuals can increase miss alarm rates and detection time.

Ferrari et al. [56, 57] proposed a robust distributed fault detection and identification approach. However, their approach does not propose any algorithm to partition the system or to determine the minimum required shared variables between the subsystems. In this work, we propose an algorithm based on residual selection for distributed diagnosis. It is straight forward to extend our approach to robust distributed diagnosis by considering residuals robustness performance in the selection process. Our algorithm provides globally correct diagnosis results and guarantees that the subsystems share the minimum number of measurements, implying that we minimize the communication of measurement streams across subsystems of the global system.

2.3 Fault Detection and Isolation in Hybrid Systems

In the real world, many complex systems, such as automobiles [172], aircraft [41] and spacecraft [16, 36] exhibit hybrid behaviors, where continuous behavior evolution is interspersed by discrete changes that occur at points in time. Therefore, hybrid systems exhibit continuous behaviors with discrete mode transitions that may be attributed to configuration changes in the system or to simplifying assumptions, where complex nonlinear behaviors are replaced by a sequence of simpler piecewise linear forms [136]. As a result, diagnosis

of hybrid systems present a much bigger challenge than diagnosis of continuous systems. Tracking mode changes in hybrid observers can be a challenging task [130]. It becomes even more challenging in the presence of faults, for multiple reasons: (i) faults cause unknown changes in a system model, therefore, tracking mode changes in the presence of faults is difficult; and (ii) it may be hard to differentiate the change in behavior due to a fault from change in behavior caused by a mode transition.

Moreover, it is computationally intractable to directly extend continuous system diagnosis by pre-enumerating all the operation modes of the system, and generating residuals for each mode to track and isolate faults. State estimation [85], parity equations [33], and fault signatures based on temporal causal graphs [137] are some of the approaches that researchers have adapted to address the problem of fault detection and isolation in hybrid systems. Mode detection becomes an essential step when extending continuous system diagnosis to hybrid system diagnosis. Domlan et al [43] have presented sufficient conditions for mode detectability of linear hybrid systems. However, the mode detection problem becomes challenging when hybrid systems exhibit nonlinear behaviors in one or more of their operating modes.

A common approach to addressing fault detection and isolation in hybrid systems is based on state estimation approaches, where multiple-model estimation schemes are employed to track the likely hybrid trajectories of the system [2, 78, 174]. However, tracking all the possible trajectories of a hybrid system is computationally intractable, since the complexity grows exponentially as the number of modes. The complexity becomes even worse, when one considers trajectories that include possible faults in the system. Adaptive multiple-model-estimation has been proposed as a possible solution to this problem [116, 117, 118]. The idea is to track a subset of modes that are most likely at a given step time. An alternative approach estimates discrete modes and continuous state variables simultaneously (e.g., see [89] and [65]). This approach is relatively efficient, however,

because of low probability of fault modes, the standard estimation methods may fail to detect them. Williams and Hofbaur [85] developed an estimation based approach for hybrid diagnosis which only tracks a set of highly likely paths.

Cocquempot et al [34] extended parity equations and the analytical redundancy approach for mode detection in hybrid systems. They used ARRs to detect the current operating mode, and then apply a proper set of residuals to isolate faults in each operating mode. In [33] the authors derived necessary and sufficient conditions for discernable modes under no fault conditions, however, they do not present required conditions for isolating mode transitions from faults. Moreover, as discussed above, pre-enumeration of all the possible modes to monitor system behavior is computationally expensive, and in some cases infeasible. Bayoudh et al [12] introduced parameterized ARRs to account for different modes of a class of hybrid systems. They considered faults as new modes in the system, and used the ARRs to track mode transitions. Low et al [121, 122] developed the concept of global ARRs (GARRs) for fault detection in hybrid systems. GARRs are analytical redundancy relations between continuous and discrete known parameters that are valid over all of the hybrid system operating modes. Levy et al. [113] proposed an integrated approach which combines GARRs and a discrete monitoring approach [155] for mode detection.

To avoid pre-enumeration of all the system modes and maintain the detection performance, Narasimhan and Biswas [137] used the hybrid bond graph (HBG) representation [136] that captures discrete switches in model configuration at the component-level. The system mode is defined by the state of all of the discrete switches associated with the system. In their work, they assumed accurate mode tracking under nominal conditions, and incremental on-line generation of models for a particular mode after a mode transition [146, 151]. This reduces the complexity of tracking operating modes after fault detection. However, they had to implement a roll back function to account for the fact that a fault may be detected only after a certain number of unknown mode transitions after the actual fault occurrence, and then a quick roll forward to catch up with the current measurements. In

addition, their approach had to recompute the fault signatures after every predicted mode transition which adds to the computational complexity of the algorithm. Bregon et al. [21] extended the PCs approach for continuous systems to decompose the hybrid systems to small subsystems. This increases the efficiency of the algorithm by avoiding causality reassignment for the complete bond graph model.

2.3.1 Summary

State estimation [85], parity equations [33], and fault signatures based on temporal causal graphs [137] are some of the approaches that researchers have adapted to address the problem of fault detection and isolation in hybrid systems. Mode detection becomes an essential step when extending continuous system diagnosis to hybrid system diagnosis. Moreover, pre-enumeration of all the possible modes in the design stage is computationally expensive, and in some cases infeasible. We propose a new approach to the problem of mode detection and fault detection and isolation in hybrid systems in this work. Unlike previous work [137], our algorithm can detect the operating mode even in the presence of faults and, therefore, it does not require a backward search for the correct sequences of mode transitions after a fault. Moreover, our approach does not pre-compile the residuals for every possible mode and, therefore, it does not have to pre-enumerate all the possible modes. Instead, it updates the diagnoser when the system switches to a new operating mode. These make our approach feasible for complex systems.

2.4 Conclusion

We have reviewed the state of the art for three fundamental research problems in the area of model-based diagnosis; 1) robust fault detection and isolation, 2) distributed fault detection and isolation, and 3) fault detection and isolation in hybrid systems. We have also listed some of the limitations in these approaches. It is critical for diagnosis methods to provide accurate and efficient solutions that can be applied for diagnosis in nonlinear

complex systems. Therefore, we develop solutions to address some of the limitations of the existent methods in these three different problem areas in the next chapters.

CHAPTER III

ROBUST FAULT DETECTION AND ISOLATION

A number of residual generation methods have been developed for robust model-based fault detection and isolation (FDI). However, the robust residual generation methods are only applicable to specific classes of systems and don't provide a general robust FDI solution. Moreover, design-time algorithms are not tuned to optimize performance for different operating regions of system behavior. Our proposed approach for robust fault detection and isolation has three main steps: 1) residual generation, 2) quantifying the performance of the residuals over the known operating regions of the system, and 3) residual selection to meet the performance requirement (see Figure 1). For residual generation, we use the fault diagnosis toolbox developed by Frisk and Krysander [68]. The toolbox generates a set of residuals given a system model and set of measurements. The basic algorithms implemented in the toolbox are presented in [67, 106, 107, 166]. To quantify the performance of the residuals, we need to define measures of sensitivity and robustness. We also need efficient algorithms to select a set of residuals that meet the performance requirements as system behavior transitions between operating regions. The rest of this chapter is organized as follows. Section 3.1 presents the background concepts and definitions. Section 3.2 presents the problem formulation. Section 3.3 develops the sensitivity analysis approach to define the performance measures: *detectability ratio*, *isolability ratio*, *global detectability ratio*, and *global isolability ratio*. Section 3.4 defines fault detectability and isolability in the presence of noise and uncertainties in the system. Section 3.5 presents our algorithms for residual selection. Section 3.6 presents the case studies, and Section 3.7 presents the conclusions of this chapter.

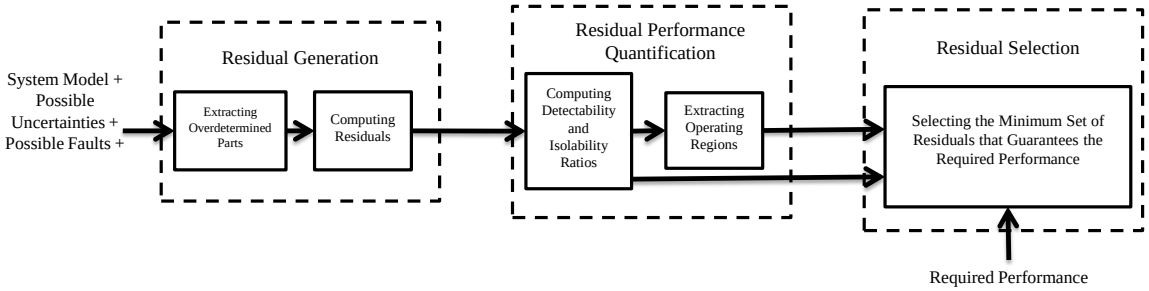

Figure 1: Automated residual generation and selection for robust FDI.

3.1 Background

This section reviews the background concepts and presents relevant definitions. We start with a general definition of a nonlinear system model, and introduce the basic concepts associated with residual generation and robust residual selection for diagnosis of dynamic systems.

3.1.1 System Representation

Model-based approaches to dynamic systems diagnosis use a system model, S, that can be defined as:

Definition 1 (System model)**.** *A system model S is defined as a four-tuple: (V, E, F,* Δ*), where V is the set of variables, E is the set of equations, F is the set of system faults and* Δ *is the set of system uncertainties.*

Adopting a state space equation representation, the general model of nonlinear dynamic systems takes the form:

$$\begin{aligned} \dot{x} &= g(\theta_n, x, u, \Delta, F), \\ y &= h(\theta_n, x, u, \Delta, F), \end{aligned} \tag{1}$$

where $x \subset V$ represents the set of state variables, $\dot{x} \subset V$ is the corresponding set of state variable (time) derivatives, $u \subset V$ are the actuator signals, $y \subset V$ is the set of measured signals in the system and $\theta_n \subset V$ is the set of system parameters. $g \subset E$ and $h \subset E$ represent

the set of system equations, Δ is the set of parameter uncertainties, and F is the set of possible faults in the system.

In this work, we model faults and disturbances as deviations of system parameters from their nominal values [66]. Faults and uncertainties can have additive and multiplicative effects. Therefore, each component parameter in the system may vary from its nominally specified value due to uncertainties and fault occurrences. Consider component i with parameter value θ_i. We can represent θ_i as: $\theta_i = \theta_{ni}(1+\delta_{\theta i})(1+f_{\theta i})$, where θ_{ni} is the presumed nominal value of the component parameter, and $f_{\theta i} \in F$ and $\delta_{\theta i} \in \Delta$ represent faults and uncertainties associated with the component. As a running example for this chapter, consider the following dynamic system

$$\begin{aligned}
\dot{x}_1 &= -(1+\delta_1)x_1 + u_1 + f_1 \\
\dot{x}_2 &= x_1 - 2(1+\delta_2)x_2 + u_2 \\
y_1 &= (1+\delta_3)x_1 \\
y_2 &= (1+f_2)x_2 \\
\dot{x}_1 &= \frac{dx_1}{dt} \\
\dot{x}_2 &= \frac{dx_2}{dt},
\end{aligned} \tag{2}$$

where x_1 and x_2 are the system state variables, u_1 and u_2 represent the inputs to the system, y_1 and y_2 are the system measurements, δ_1 and δ_2 represent system uncertainties, δ_3 represents a sensor uncertainty, f_1 is an actuator fault and f_2 is a sensor fault. Figure 2 illustrates the system model in block diagram form.

3.1.2 Residuals

Residuals are the basis for model-based fault detection and isolation. A residual represents as an analytical redundancy relation (ARR) in the system. In this work, we define a residuals as

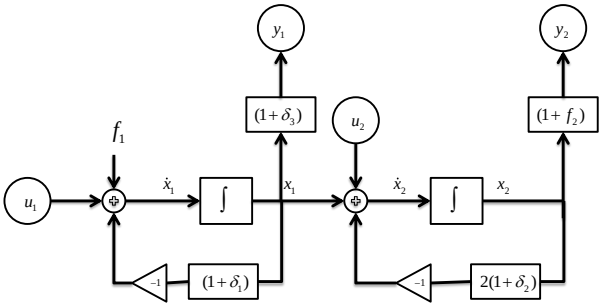

Figure 2: Simple example diagram.

Definition 2. *(Residual) A residual r is an analytical redundancy relation between nominal parameters of the system, process measurements, and inputs. We model residuals for the nonlinear dynamic system (1) as a nonlinear relationship between known nominal parameters of the system, process measurements, and known inputs in the following manner:*

$$\begin{aligned} \dot{z} &= \hat{g}(z, \theta_n, u, y) \\ r(z, \theta_n, u, y) &= 0, \end{aligned} \tag{3}$$

where z represents the internal dynamics of the residual and $\hat{g}$ and r represent the dynamic model of the residual.

To generate a residuals, the system has to be redundant or overdetermined. Structurally overdetermined (SO) sets and minimal structurally overdetermined (MSO) sets are defined in the next subsection.

3.1.3 Minimal Structurally Overdetermined Sets of Equations

Frisk and Nyberg [69] argued that minimal residuals use fewer parameters from the system model and fewer measurements from the sensors and therefore, tend to be more robust against model uncertainties. Moreover, MSOs are well suited for automatic residual generation. Krysander et al. [106] developed an algorithm for finding the set of MSOs

given the set of system equations. This algorithm is used in the diagnosis toolbox [68] for residual generation. Next, we formally define structurally overdetermined and MSO sets.

Definition 3. *(Structural Overdetermined Set) Consider a subset of system equations and its associated variables:* (E^i, V^i). *This set of equations is structurally overdetermined (SO) if the cardinality of the set* E^i *is greater than the cardinality of set of unknown variables* $V^i_{unknown} \subseteq V^i$, *i.e.* $|E| > |V^i_{unknown}|$.

Consider the running example in equation (2). There are four unknown variables, x_1, $\dot{x}_1$, x_2, and $\dot{x}_2$, and six equations. Therefore, this system presents a SO set:

$$\begin{aligned} e_1 &: \dot{x}_1 + x_1 - u_1 = 0 \\ e_2 &: \dot{x}_2 - x_1 + 2x_2 - u_2 = 0 \\ e_3 &: y_1 - x_1 = 0 \\ e_4 &: y_2 - x_2 = 0 \\ e_5 &: \dot{x}_1 - \frac{dx_1}{dt} = 0 \\ e_6 &: \dot{x}_2 - \frac{dx_2}{dt} = 0. \end{aligned} \tag{4}$$

Definition 4. *(Minimal Structurally Overdetermined Set) A set of over determined equations is minimal structurally overdetermined (MSO) if it has no subset of equations that is structurally overdetermined.*

Consider the SO set in equation (4). e_1, e_3, and e_5 represent a minimal set of over determined equations:

$$\begin{aligned} e_1 &: \dot{x}_1 + x_1 - u_1 = 0 \\ e_3 &: y_1 - x_1 = 0 \\ e_5 &: \dot{x}_1 - \frac{dx_1}{dt} = 0. \end{aligned} \tag{5}$$

Note that faults and uncertainties are not mentioned in *Definition* 4. However, in addition

to equations, and variables, each MSO can have a set of faults and uncertainties. For example, the MSO in equation (70) can be represented in a more general way as $MSO_{70} = (E_{70}, V_{70}, M_{70}, F_{70}, \Delta_{70})$, where $E_{70} = \{e_1, e_3, e_5\}$ is the set of equations, $V_{70} = \{\dot{x}_1, x_1\}$ is the set of unknown variables, $M_{70} = \{u_1, y_1\}$ is the set of measurements (known variables), $F_{70} = \{f_1\}$ is the set of faults and $\Delta_{70} = \{\delta_1, \delta_3\}$ is the set of uncertainties in the MSO. For the sake of brevity and simplification we simply say a specific equation, variable, measurement, fault, or uncertainty is a member of a MSO. For example, we say $f_1 \in MSO_{70}$.

3.1.4 Fault Detectability and Fault Isolability

MSOs represent the redundancies in the system and can be used for fault detectability and isolability analysis [107]. A detectable fault is defined as:

Definition 5. *(Detectable fault) A fault $f \in F$ is detectable in system S if there is a minimal structurally overdetermined set MSO_i in the system, such that $f \in MSO_i$.*

Consider *Definition* 5 and MSO_{70}. Fault f_1 is detectable because $f_1 \in MSO_{70}$. In a similar way, fault isolability is defined as:

Definition 6. *(Isolable fault) A fault $f_i \in F$ is isolable from fault $f_j \in F$ if there exists a minimal structurally overdetermined set MSO_i in the system S, such that $f_i \in MSO_i$ and $f_j \notin MSO_i$.*

Consider *Definition* 6 and MSO_{70}. Fault f_1 is isolable from fault f_2 because $f_1 \in MSO_{70}$ and $f_2 \notin MSO_{70}$.

3.1.5 Residual Generation

To generate an explicit mathematical residual equation from an MSO we have to apply a computational sequence to eliminate the unknown variables. A residual is in integral

(derivate) causality if and only if its computational structure or equation includes only integral (derivative) forms. Equation (86) represents a residual in integral causality. To generate a residual from MSO_{70} in derivative causality, we start with (70), use e_3 to substitute $x_1 = y_1$ in e_1 and generate a residual:

$$r_d = \dot{y}_1 + y_1 - u_1. \tag{6}$$

Otherwise, we can use e_1 to estimate x_1 and generate a residual in integral causality

$$\begin{aligned} \dot{z}_1 &= -z_1 + u_1 \\ r_i &= y_1 - z_1. \end{aligned} \tag{7}$$

In this work, an exhaustive approach for generating residuals is adopted so that the minimum set of residuals can be picked to meet specified detectability and isolability criteria. As we mentioned earlier, we use the fault diagnosis toolbox [68] to generate the set of residuals. Given the set of system equations, the toolbox uses an efficient algorithm designed by Krysander et al. [106] for finding minimally overdetermined sets of constraints to generate the minimal structurally overdetermined (MSO) sets. It then applies the algorithm developed by Svard and Nyberg [166] to generate residuals from the generated MSO sets.

3.1.6 Residual Selection

Most diagnosis approaches reported in the literature perform residual generation and residual selection simultaneously [140, 154]. However, since our focus is on choosing an optimal set of residuals, our method generates the entire set of residuals and then applies a methodology to select a minimum subset of residuals with sufficient sensitivity to the faults of interest, and robustness to known uncertainties. The total number of residual candidates

for fault detection and isolation grows exponentially as the number of measurements increase [5, 167]. Using the model form in (1), the total number of redundancies introduced into the system model is equal to the number of measurements, l_y. Theoretically, each MSO can include from one to l_y measurements. Therefore, the total number of MSOs, N_{MSO} is proportional to all possible combinations of the measurements:

$$N_{MSO} \propto \sum_{i=1}^{l_y} \binom{l_y}{i} = 2^{l_y} \tag{8}$$

In addition, by assuming different causality types (integral, differential), each MSO can be used to generate several residuals [67]. Further, linear and non-linear combinations of residuals generate new residuals. Therefore, a large number of residual subsets can be generated to detect faults. Given the uncertainties in the system model, the sensitivity of these residuals to each uncertain parameter has to be computed to determine the residuals that provide the required performance. The sensitivity of the residuals to the faults and uncertainties are not the same. Therefore, different sets of residuals can have different performance values for fault detection and isolation. Moreover, for a given behavior trajectory, the performance of residuals can vary from one operating region to another for the system. To achieve the required performance for the entire trajectory our algorithm needs to find a set of residuals that satisfies the pre-specified performance thresholds across all of the operating regions. To simultaneously minimize computational costs, the algorithm selects a residual set that have minimal cardinality.

Note that this is equivalent to finding an optimal solution for the set covering problem, which is known to be NP-hard [105] and, therefore, any algorithm for finding a set of residuals with minimal cardinality and required performance will have exponential computational complexity. Svard et al. [167] used a greedy search strategy to select a minimal set of residuals that can detect and isolate a set of faults. Roychoudhury et al. [150] developed a greedy heuristic algorithm for selecting a set of measurements for each distributed

fault diagnosis unit and compared the results to an optimal solution generated by exhaustive search. Greedy algorithms are computationally efficient but they are not guaranteed to find the optimal solution. Sarrate et al. [157] have employed a binary integer programming approach for optimal sensor placement for fault detection and isolation. In this work, the residual selection problem is formulated as a BILP problem. The approach efficiently picks a minimum number of residuals that guarantees the required detectability and isolability performances for all the operating modes.

3.2 Problem Formulation

In this chapter, the detectability ratio of each fault of interest, f_i, given a residual r, $D(f_i|r)$, is defined as a measure that captures the performance of residual r in detecting the fault f_i. The isolability ratio for a residual, r, for pairs of faults f_i and f_j, $f_i \neq f_j$, $I(f_i, f_j|r)$ is also defined as a measure that represents the performance of residual r in isolating f_i from f_j. Using these measures and minimum specified detectability and isolability thresholds provided by the system designers and operators, the set of qualified residuals for fault detection and isolation at a time interval, T_a, are defined as

Definition 7 (Set of qualified residuals for detecting fault f_i)**.** *The set of residuals that meet the specified criterion to detect each fault f_i over a time interval T_a, is the set of residuals whose detectability ratios of fault f_i is greater than the specified detectability ratio threshold during the time interval T_a.*

Definition 8 (Set of qualified residuals for isolating fault f_i from fault f_j)**.** *The set of qualified residuals for isolating fault f_i from fault f_j over a time interval T_a is the set of residuals whose isolability ratios for (f_i, f_j) is greater than a pre-specified isolability ratio for the time interval T_a.*

The system operating regions are defined based on the qualified residuals for fault detection and isolation.

Definition 9 (System operating regions)**.** *The operating regions for a system is defined as the longest time interval starting at t_1 and ending at t_2 such that for each pair of faults f_i and f_j in the system, the set of qualified residuals for detecting f_i and f_j, and also the set of qualified residuals for isolating f_i from f_j do not change in that time interval.*

Using definition 9, a new system operating region is defined when the set of residuals that meet the pre-specified performance criteria for detection and isolation change. As system behavior evolves dynamically, it is possible that the system may switch back to a previous region of operation, i.e., it may switch back to a set of residuals that were used earlier. To maintain the pre-specified detectability and isolability performances with minimum on-line computation as the regions of operation change, our algorithm chooses a minimum subset of residuals that meet the detectability and isolability performance criteria over the entire behavior trajectory. The objective is to find a subset of residuals with minimum cardinality number that fulfills a specified diagnosability performance for all the operating regions of the system. The selected residual set can then be invoked on-line for the fault detection and isolation tasks. The offline residual set selection problem is presented as follows.

The residual selection problem for known system behavior trajectory: let $\mathscr{F} = \{f_1, f_2, \ldots, f_n\}$ denote the set of faults, and $\mathscr{R} = \{r_1, r_2, \ldots, r_l\}$ represent the entire set of residuals given the system model and the set of measurements made on the system. It is assumed that this set of measurements is sufficient to detect and uniquely isolate all of the faults in $\mathscr{F}$. Our goal is to develop an algorithm that selects a subset of residuals R^* which guarantees the pre-specified detectability ratio for each fault $f_j \in \mathscr{F}$ and pre-specified isolability ratio for each pair of faults f_j and f_k at all the operating regions, M, and includes a minimum number of residuals. More formally, the residual selection problem can be defined as:

$$
\begin{aligned}
\min \quad & R^* \subseteq \mathscr{R} \\
\text{s.t.} \quad & \\
& \forall m_i \in M : \\
& \forall f_i \& f_j \in \mathscr{F} : \\
& \exists r_i \& r_j \in R^* : \\
& D(f_i | r_i) > D_{req}, \\
& I(f_i, f_j | r_j) > I_{req},
\end{aligned}
\tag{9}
$$

where D_{req} and I_{req} are minimum detectability and isolability required performances, respectively. In the next section, we develop our method to quantify the diagnosability performance of each residual.

3.3 Quantifying Residual Performance

A residual, $r(y,u) = 0$, captures nominal system behavior. Ideally, each residual should be zero in the fault-free case and residuals sensitive to a fault become nonzero when it occurs. Due to uncertainties and noise, the residuals may deviate from zero even in the fault-free case, and this complicates the fault detection and isolation task. Depending on the relationships between faults and uncertainty magnitudes, the likelihood of detecting and isolating a fault of a specified magnitude in the presence of uncertainties can vary. If we can quantify these relations, we can use them to select the best set of residuals for fault detection and isolation. In this section we use two approaches to develop quantitative measures of residual performance for fault detection and isolation in the presence of noise and uncertainties: 1) derivative-based sensitivity analysis, and 2) global sensitivity analysis.

3.3.1 Derivative-based sensitivity analysis approach

Sensitivity analysis evaluates how model behaviors are affected by changes in model parameters [160]. In reality, faults and uncertainties result in deviations in the system variables from their nominal values, producing non zero residual values. Unlike previous work [42], where the focus is on computing the sensitivity of the residuals with respect to model uncertainties, we start by computing the derivative of state and measured variables with respect to all parameters by applying the chain rule for derivative computation. Consider a general nonlinear dynamic system model (1). We assume that the system has l_x state variables, and l_y measurements and $x \in R^{l_x}$ is the state vector and $y \in R^{l_y}$ is the measurement vector. The chain rule [144] applied to the state and output equations produces:

$$\begin{aligned} \dot{p}_\psi &= \frac{\partial g}{\partial x} p_\psi + \frac{\partial g}{\partial \psi} \\ q_\psi &= \frac{\partial h}{\partial x} p_\psi + \frac{\partial h}{\partial \psi}, \end{aligned} \tag{10}$$

where $\frac{\partial g}{\partial x} \in R^{l_x \times l_x}$, $\frac{\partial g}{\partial \psi} \in R^{l_x}$, $\frac{\partial h}{\partial x} \in R^{l_y \times l_x}$, and $\frac{\partial h}{\partial \psi} \in R^{l_y}$. $p_\psi = \frac{\partial x}{\partial \psi} \in R^{l_x}$, and $q_\psi = \frac{\partial y}{\partial \psi} \in R^{l_y}$ represent the sensitivity of state variables and measurements to the parameter ψ. ψ can be a fault, f_i, or the uncertainty associated with a parameter, δ_j.

Consider the running example represented by equation (2). We can present the dynamics associated with $p_{1\delta_1} = \frac{\partial x_1}{\partial \delta_1}$ and $p_{2\delta_1} = \frac{\partial x_2}{\partial \delta_1}$ for nominal behavior as:

$$\begin{aligned} \dot{p}_{1\delta_1} &= -p_{1\delta_1} - x_1 \\ \dot{p}_{2\delta_1} &= p_{1\delta_1} - 2p_{2\delta_1} \end{aligned} \tag{11}$$

Using equation (2) and the chain rule we can derive the sensitivity of known system variables to δ_1 as a function of $p_{1\delta_1}$ and $p_{2\delta_1}$:

$$\begin{aligned} \frac{\partial u_1}{\partial \delta_1} &= 0 & \frac{\partial u_2}{\partial \delta_1} &= 0 \\ \frac{\partial y_1}{\partial \delta_1} &= p_{1\delta_1} & \frac{\partial y_2}{\partial \delta_1} &= p_{2\delta_1}. \end{aligned} \tag{12}$$

Note that u_1 and u_2 are external control signals that does not depend on system incarcerates, implying $\frac{\partial u_1}{\partial \delta_1} = \frac{\partial u_2}{\partial \delta_1} = 0$. We can present the state equations associated with the sensitivity of system variables to fault f_1 as:

$$\begin{aligned} \dot{p}_{1f_1} &= -p_{1f_1} + 1 \\ \dot{p}_{2f_1} &= p_{1f_1} - 2p_{2f_1} \\ \frac{\partial y_1}{\partial f_1} &= p_{1f_1} \\ \frac{\partial y_2}{\partial f_1} &= p_{2f_1}, \end{aligned} \tag{13}$$

where $p_{1f_1} = \frac{\partial x_1}{\partial f_1}$ and $p_{2f_1} = \frac{\partial x_2}{\partial f_1}$. Sensitivity of system variables to other uncertainties and faults can be derived in a similar manner but for the lack of space we do not list all of them in this section.

Sensitivity analysis of residuals: Consider the residual model given by equation (86). We assume that the residual has l_z state variables, and $z \in R^{l_z}$ is the residual state vector. We can use the sensitivity of the system variables to uncertainties and faults derived in the previous section to compute the sensitivity of the residual to faults and uncertainties, represented by ψ, as:

$$\begin{aligned} \dot{\hat{p}}_\psi &= \frac{\partial \hat{g}}{\partial z}\hat{p}_\psi + \frac{\partial \hat{g}}{\partial y}q_\psi \\ \frac{\partial r}{\partial \psi} &= \frac{\partial r}{\partial z}\hat{p}_\psi + \frac{\partial r}{\partial y}q_\psi, \end{aligned} \tag{14}$$

where $\frac{\partial \hat{g}}{\partial z} \in R^{l_z \times l_z}$, $\frac{\partial \hat{g}}{\partial y} \in R^{l_z \times l_y}$, $\frac{\partial r}{\partial z} \in R^{l_z}$, $\frac{\partial r}{\partial y} \in R^{l_y}$ and $\hat{p}_\psi = \frac{\partial z}{\partial \psi} \in R^{l_z}$. For the running example, the fault diagnosis toolbox generates four MSOs and a residual from each MSO. The four residuals are

$$\begin{aligned} \dot{z}_{11} &= -z_{11} + u_1 & \dot{z}_{22} &= y_1 - 2z_{22} + u_2 \\ r_1 &= y_1 - z_{11} & r_2 &= y_2 - z_{22} \end{aligned} \tag{15}$$

$$
\begin{aligned}
\dot{z}_{31} &= -z_{31} + u_1 \\
\dot{z}_{32} &= z_{31} - 2z_{32} + u_2 \qquad\qquad r_4 = y_1 - \dot{y}_2 - 2y_2 + u_2 \\
r_3 &= y_2 - z_{32}
\end{aligned} \tag{16}
$$

The first three residuals are in integral causality and the last one is in derivative causality. As an example, the sensitivity of the first residual r_1 to δ_1 is derived as:

$$
\frac{\partial r_1}{\partial \delta_1} = p_{1\delta_1}, \tag{17}
$$

where $p_{1\delta_1}$ dynamic equation is presented in (11).

3.3.1.1 Detectability Ratio

We use sensitivity analysis to compute the effects of uncertainties and faults on the residuals. It is assumed that the fault magnitude and uncertainties are small and approximately constant over time. Ideally, in no fault situations all residuals will have a value of zero. A first order linear approximation of the residual with respect to the set of faults F and uncertainties Δ is given by:

$$
r(y,u) \approx \sum_{i=1}^{l_f} \frac{\partial r}{\partial f_i} f_i + \sum_{j=1}^{l_\delta} \frac{\partial r}{\partial \delta_j} \delta_j, \tag{18}
$$

where l_f is the cardinality of F and l_δ is the cardinality of the set of uncertainties, Δ. The partial derivatives $\frac{\partial r}{\partial f_i}$ and $\frac{\partial r}{\partial \delta_j}$ are computed using (14). Note that if a residual r is not sensitive to a fault or uncertainty then the corresponding partial derivative is zero.

When quantifying the detectability performance of a residual r with respect to a fault f_i the relative effect of the fault is compared to the total effect of the uncertainties Δ. Since the actual magnitudes of the fault and uncertainties are unknown, the maximum values of the magnitude of uncertainties and minimum magnitudes of a fault f_i are used for the

calculations. This gives us the worst case scenario of the difficulty in detecting a fault. We define a quantitative measure of detectability performance as follows.

Definition 10. *(Detectability Ratio) Given a dynamic system (1) the detectability ratio of a fault f_i for a residual r presented in (86) is defined as:*

$$D(f_i|r) = \frac{\left|\frac{\partial r}{\partial f_i}\min(f_i)\right|}{\left|\frac{\partial r}{\partial f_i}\min(f_i)\right| + \sum_{j=1}^{l_\delta}\left|\frac{\partial r}{\partial \delta_j}\max(\delta_j)\right|}, \tag{19}$$

If $\frac{\partial r}{\partial f_i} = 0$ *then* $D(f_i|r) = 0$.

Note that we assume that each fault f_i and uncertainty δ_i have known lower and upper bounds magnitudes:

$$\begin{aligned} \min(f_i) \leq |f_i| \leq \max(f_i) \\ \min(\delta_i) \leq |\delta_i| \leq \max(\delta_i). \end{aligned} \tag{20}$$

If the maximum magnitude of a fault f_i is unknown, then $\max(f_i) = \infty$. The detectability ratio has a value in the interval $[0, 1]$, where 0 corresponds to the situation where the residual is not sensitive to the fault f_i and 1 if there are no uncertainties affecting the residual's ability to detect the fault. For example, consider $u_1 = sin(t)$, $u_2 = cos(t)$, with a maximum amplitude of 1% for each parameter and sensor uncertainty, and $\min(f_2) = 1$, in the running example in equation (2). The detectability ratio of f_2 given residual r_d in equation (6) is $D(f_2|r) = 0$. This is not surprising because r_d is an analytical redundancy relationship between u_1, y_1 and $\dot{y}_1$ and, therefore, it is not sensitive to the second sensor fault. If the effect of a fault, f, is larger than the total effect of all the uncertainties then $D(f|r) > 0.5$, which means that the fault is detectable.

3.3.1.2 Hybrid Residuals

Sensitivity of system measurements to δ_1 and f_1 are presented in equation (12) and (13) respectively. We derive sensitivity of system variables to δ_2 and δ_3 in a similar manner as

follows.

$$\begin{aligned} \dot{p}_{2\delta_2} &= -2p_{2\delta_2} - 2x_2 \\ \frac{\partial y_1}{\partial \delta_2} &= 0 \\ \frac{\partial y_2}{\partial f_1} &= p_{2\delta_2}, \end{aligned} \tag{21}$$

where $p_{2\delta_2} = \frac{\partial x_2}{\partial \delta_2}$.

$$\begin{aligned} \frac{\partial y_1}{\partial \delta_3} &= x_1 \\ \frac{\partial y_2}{\partial \delta_3} &= 0. \end{aligned} \tag{22}$$

Using equations (12), (13), (21), and (22), we can derive the detectability ratios of fault f_1 for the four residuals generated by the fault diagnosis toolbox in equation (15) and equation (16) as:

$$\begin{aligned} D(f_1|r_1) &= \frac{\left|p_{1f_1}\min(f_1)\right|}{\left|p_{1f_1}\min(f_1)\right| + \left|p_{1\delta_1}\max(\delta_1)\right| + \left|x_1\max(\delta_3)\right|} \\ D(f_1|r_2) &= 0 \\ D(f_1|r_3) &= \frac{\left|p_{2f_1}\min(f_1)\right|}{\left|p_{2f_1}\min(f_1)\right| + \left|p_{2\delta_1}\max(\delta_1)\right| + \left|p_{2\delta_2}\max(\delta_2)\right|} \\ D(f_1|r_4) &= 0. \end{aligned} \tag{23}$$

r_1 and r_3 are sensitive to fault f_1 and, therefore, we can use each of them to detect f_1. However, these residuals do not represent the same detectability performances. r_1 is not sensitive to δ_2, and its sensitivity to δ_3 is a function of the state variable x_1. On the other hand, r_3 is not sensitive to δ_3.

A residual with higher detectability ratio is less likely to produce false alarms, is more likely to report the fault and also because of higher sensitivity it can detect smaller fault magnitudes. As we can see in equation (23), the detectability ratio of residuals vary in different operating regions. It is possible to track the detectability ratio of residuals, and choose the best residual for each of the operating regions of the system. If we use the following hybrid residual for the system we will have a residual with maximum sensitivity

to f_1 and robustness to the uncertainties.

$$r = \begin{cases} r_1 & if \quad D(f|r_1) \geq D(f|r_3) \\ r_3 & otherwise. \end{cases} \tag{24}$$

3.3.1.3 Isolability Ratio

In typical diagnosis applications, fault detection is followed by fault isolation. To isolate a fault f_i from the other faults in the system, we need a residual that is sensitive to fault f_i but invariant or robust to other faults and uncertainties. However, the effects of faults and uncertainties in the residuals are unknown, and we have to, like before, estimate them using sensitivity analysis. Making the single fault assumption, we consider that only one of the possible faults occurs, and we want to quantify the performance of each of the residuals to isolate that fault from the others. The magnitudes of the possible faults and uncertainties are unknown, therefore, to quantify the performance of residual r to isolate fault f_i from another fault f_j, the minimum magnitude of f_i and the maximum magnitude of f_j and uncertainties are considered. In other words, the other fault f_j is treated as an uncertainty. Then, the isolability ratio is defined, using equation (18) as a quantitative measure of isolation performance as follows.

Definition 11. *(Isolability Ratio) Given the dynamic system (1) the isolability ratio for fault f_i from another fault f_j, using residual r is defined as:*

$$I(f_i, f_j|r) = \frac{\left|\frac{\partial r}{\partial f_i}\min(f_i)\right|}{\left|\frac{\partial r}{\partial f_i}\min(f_i)\right| + \left|\frac{\partial r}{\partial f_j}\max(f_j)\right| + \sum_{k=1}^{l_\delta}\left|\frac{\partial r}{\partial \delta_k}\max(\delta_k)\right|}, \tag{25}$$

$I(f_i, f_j|r) > 0.5$ implies that the effect of a fault f_i on r is always larger than the total effects of the fault f_j and the combined uncertainties present in the system. Therefore, we can use r to isolate f_i from f_j. Note that if r is sensitive to f_j and we do not know the maximum magnitude of f_j, i.e. $\max(f_j) = \infty$, then r cannot be used to isolate f_i from f_j

and $I(f_i, f_j|r) = 0$. The isolability ratio is simply a generalization of the detectability ratio. In fact, the detectability ratio of a fault f_i is the isolability ratio of f_i from the no fault case.

In this subsection, we used a derivative-based sensitivity analysis to quantify the performance of residuals. The derivative-based approach is the most common method to perform sensitivity analysis. The derivative-based approach is easy to understand intuitively and computationally efficient to implement. However, the derivative-based approach only determines the effect of uncertainties at the single point at which the derivative is constructed. For linear systems, the effect of uncertainties in other operation points can be easily determined by extrapolation. For nonlinear system this can lead to a significant error. Moreover, the derivative-based approaches perform well when the there is no noise in the measured signals. To overcome to these problems, we investigate the application of a global sensitivity analysis method [175] which is based on exploring the uncertainty space in multiple points in the next subsection. In comparison with the derivative-based approach, global sensitivity approach increases the computational complexity but generates more robust results for residual quantification.

3.3.2 Global sensitivity analysis approach

The global sensitivity analysis approach computes the effects of the faults or uncertainties, represented by ψ, on residual, r, as:

$$S_{\psi}^{r} = \frac{Var(r) - E[Var(r|\psi)]}{Var(r)} \tag{26}$$

$Var(r) - E[Var(r|\psi)]$ is the expected reduction in the variance of residual, r, when parameter, ψ is fixed. S_{ψ}^{r} is called the first order sensitivity index and is always between 0 and 1 [175]. Several researchers have used global sensitivity analysis to introduce different measures for quantifying uncertainties in dynamic systems. Iman and Hora [90] defined *uncertainty importance* of a parameter X_i, I_i, as the expected reduction in the variance of

model output, Y, when X_i is fixed:

$$I_i = \sqrt{Var(Y) - E[Var(Y|X_i)]} \tag{27}$$

Sankararaman et al. [156] used *total effect* index, S_i^T, to rank the parameters uncertainty effects in prognostics.

$$S_i^T = 1 - \frac{E[Var(Y|X_i)]}{Var(Y)}, \tag{28}$$

Note that in the derivative-based approach, the sensitivity of a residual with respect to a fault or uncertainty was independent of the other faults and uncertainties. However, the global sensitivity analysis considers the total effects of different sources of uncertainty and generates more accurate results. Moreover, by using derivative-based sensitivity analysis, we had to assume that the fault magnitude and uncertainties are small and approximately constant over time. To overcome to these problems, we will use global sensitivity analysis to define the global detectability ratio and global isolability ratio and develop robust fault detection and isolation methods with high performance in the presence of noise and uncertainty.

Consider the running example represented by equation (2) and residual r_1 represented by equation (15). We consider the scenario where the initial state is $X_0 = (2, 1.5)$, each uncertainty has normal distribution with zero mean and 0.1 standard deviation; $\delta_i = N(0, 0.1) : i \in \{1,2,3\}$, and there is no fault in the system; $f_1 = f_2 = 0$. Figure 3 represents Scatterplots of r_1 versus uncertainties δ_1, δ_2 and δ_3. Figure 3 shows r_1 is sensitive to δ_1 and δ_3 but is not sensitive to δ_2. We can use Figure 2 to explain the observation. Residual r_1 represents the redundancy relationship between u_1 and y_1 and it is expected to be insensitive to δ_2. In this work, we are interested to study the effect of faults and uncertainties in the residuals and develop reliable measures to quantify the sensitivity of residuals to uncertainties and faults for robust residual selection. The effects of the faults or uncertainties on residuals can be computed using global sensitivity analysis. For residual r_1 in the running example,

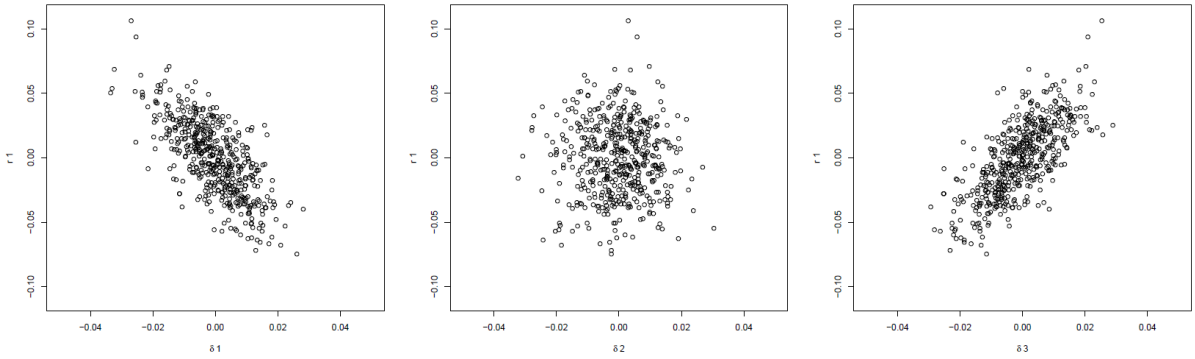

Figure 3: Scatter-plots of residual r_1 versus δ_1, δ_2, δ_3 when $f_1 = f_2 = 0$.

we can use equation (26) to derive the sensitivity of r_1 to δ_1, δ_2 and δ_3:

$$\begin{aligned} S_{\delta_1}^{r_1} &= 0.509 & S_{\delta_3}^{r_1} &= 0.516 \\ S_{\delta_2}^{r_1} &= 0 \end{aligned} \tag{29}$$

These results validate our observation in the scatter-plot in Figure 3.

Consider the case where in addition to the uncertainties, there is a fault $f_1 = N(1, 0.1)$ in the running example. Figure 4 shows the scatter-plots of residual r_1 versus δ_1, δ_2, δ_3 and f_1 in this scenario. Figure 4 shows that r_1 is very sensitive to f_1 and can be used to detect it. We can confirm this observation by computing first order sensitivity index of residual r_1 with respect to f_1 and uncertainties:

$$\begin{aligned} S_{\delta_1}^{r_1} &= 0.029 & S_{\delta_3}^{r_1} &= 0.075 \\ S_{\delta_2}^{r_1} &= 0 & S_{f_1}^{r_1} &= 0.92 \end{aligned} \tag{30}$$

Note that in the presence of fault $S_{\delta_1}^{r_1}$ and $S_{\delta_3}^{r_1}$ decrease significantly. This is an important difference between using global sensitivity analysis versus derivative-base analysis. In the derivative-base approach, the sensitivity of a residual with respect to a fault or uncertainty was independent from the other faults and uncertainties. However, the global sensitivity

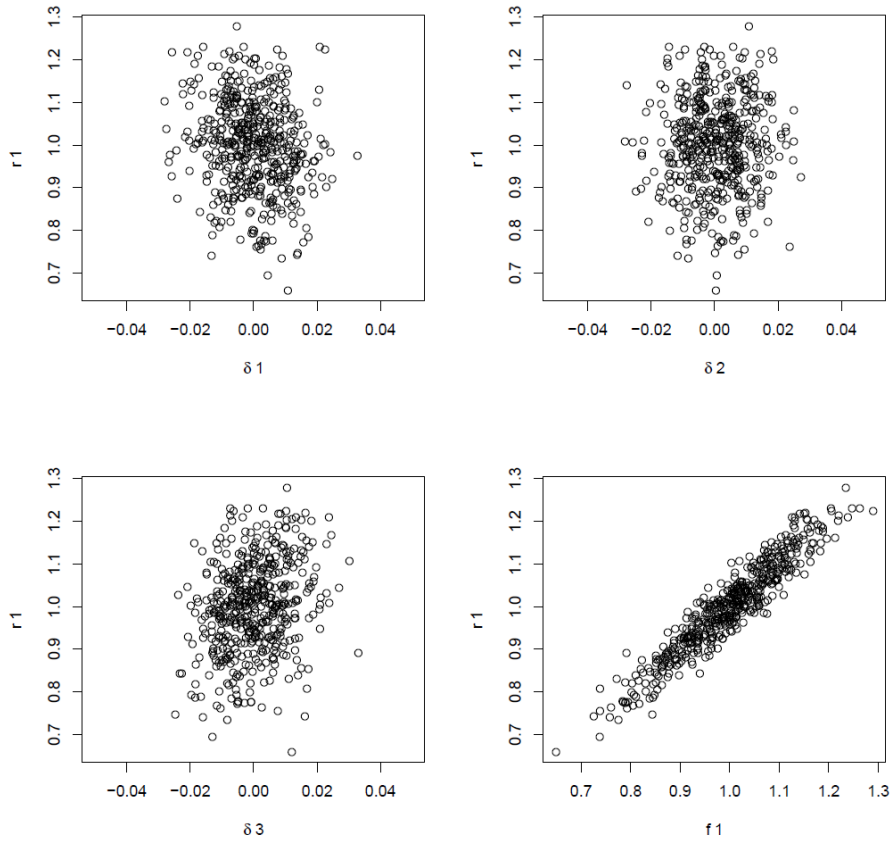

Figure 4: Scatter-plots of residual r_1 versus δ_1, δ_2, δ_3 and f_1.

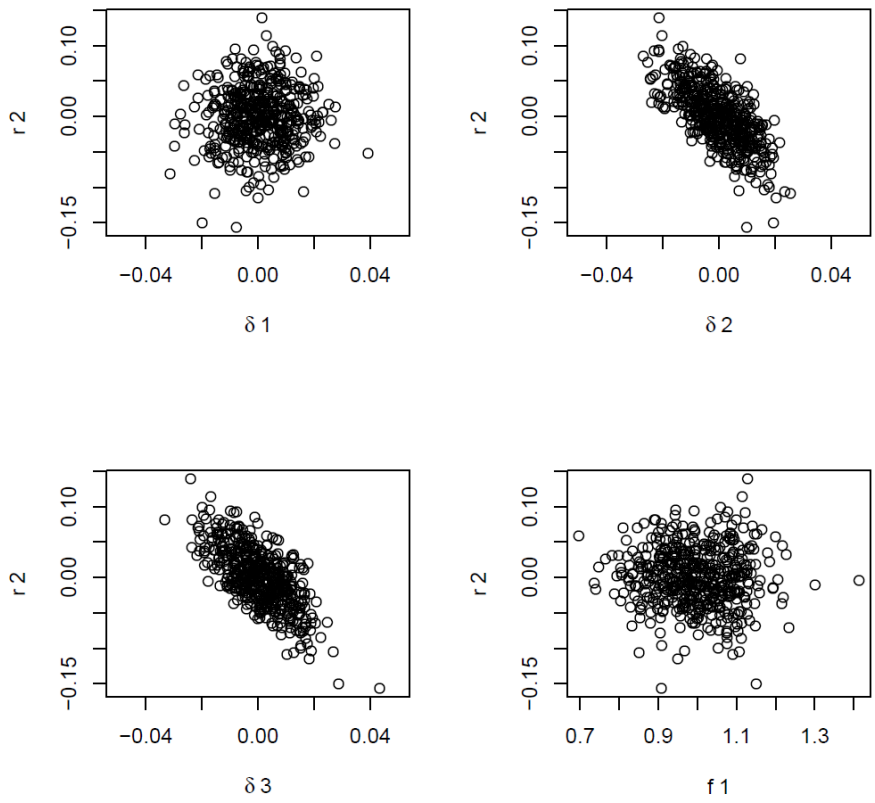

Figure 5: Scatter-plots of residual r_2 versus δ_1, δ_2, δ_3 and f_1.

analysis considers the total effects of different sources of uncertainty and generates more accurate results.

In addition to r_1, the fault diagnosis toolbox generates three more residuals for the running example (see equations (15) and (16)). Figure 5 represents the scatter-plots of residual r_2 versus δ_1, δ_2, δ_3 and f_1. We can see from the scatter-plots that r_2 is not sensitive to f_1 and can not be used to detect this fault. We validate this observation by computing sensitivity indexes of residual r_2 with respect to the fault and uncertainties:

$$\begin{aligned} S^{r_2}_{\delta_1} &= 0 & S^{r_2}_{\delta_3} &= 0.45 \\ S^{r_2}_{\delta_2} &= 0.43 & S^{r_2}_{f_1} &= 0 \end{aligned} \tag{31}$$

3.3.2.1 Global Detectability Ratio

In the previous subsection, we used derivative-based sensitivity analysis to quantify the detectability performance of a residual r with respect to a fault f_i. We compared the effect of the fault to the total effect of the uncertainties Δ. Using derivative-based, we had to make several limiting assumptions. To develop a more general approach, we use global sensitivity analysis to define the global detectability ratio as follows.

Definition 12. *(Global Detectability Ratio) Given a dynamic system (1) the global detectability ratio of a fault f_i for a residual r (86) is defined as:*

$$GD(f_i|r) = S^r_{f_i}, \tag{32}$$

where $\forall f_j \neq f_i \implies f_j = 0$.

In fact, to compute the global detectability ratio of a fault f_i for a residual r, we assume that f_i is the only possible fault in the system and consider the global sensitivity of r to f_i as $GD(f_i|r)$.

If $S^r_{f_i} = 0$ then $GD(f_i|r) = 0$. The global detectability ratio has a value in the interval $[0,1]$, where 0 corresponds to the situation where the residual is not sensitive to the fault f_i and 1 if there are no uncertainties affecting the residual's ability to detect the fault. For example, consider the running example in equation (2) with a $\delta_i = N(0,0.1)$ for $i \in \{1,2,3\}$, and $f_1 = N(1,0.1)$. The detectability ratio of f_1 given each of the residuals in equations (15) and (16) are

$$\begin{aligned} GD(f_1|r_1) &= 0.92 & GD(f_1|r_3) &= 0.92 \\ GD(f_1|r_2) &= 0 & GD(f_1|r_4) &= 0 \end{aligned} \tag{33}$$

This is not surprising because r_2 and r_4 are analytical redundancy relationships between y_1, u_2 and y_2 and, therefore, they are not sensitive to the actuator fault, f_1. On the other hand,

r_1 and r_3 perform equally well in detecting f_1. This is because we considered the same distribution for δ_2 and δ_3.

3.3.2.2 Global Isolability Ratio

Like the previous subsection, we follow fault detection by fault isolation. To isolate a fault f_i from fault f_j in the system, we need a residual that is sensitive to fault f_i but invariant or robust to f_j and uncertainties. However, the effects of faults and uncertainties in the residuals are unknown, and we have to, like before, estimate them using sensitivity analysis. Making the single fault assumption, we consider that only one of the possible faults occurs, and we want to quantify the performance of each of the residuals to isolate that fault from the others. The size and time-variant behavior of the possible faults and uncertainties are unknown, therefore, to quantify the performance of residual r to isolate fault f_i from another fault f_j, the distribution of f_i, f_j and the uncertainties in the system are considered. Then, the global isolability ratio is defined, as a quantitative measure of isolation performance as follows.

Definition 13. *(Global Isolability Ratio) Given the dynamic system (1) the global isolability ratio for fault f_i from another fault f_j, using residual r is defined as:*

$$GI(f_i, f_j|r) = S^r_{f_i}, \tag{34}$$

where $\forall f_k \neq f_i$ and $f_k \neq f_j \implies f_k = 0$.

To compute the global isolability ratio of f_i from f_j for a residual r, we assume that f_i and f_j are the only possible faults in the system and consider the global sensitivity of r to f_i as $GI(f_i, f_j|r)$.

$GI(f_i, f_j|r) > 0.5$ implies that the effect of a fault f_i on r is always larger than the total effects of the fault f_j and the combined uncertainties present in the system. Therefore, we can use r to isolate f_i from f_j. The isolability ratio is simply a generalization of the

detectability ratio. In fact, the detectability ratio of a fault f_i is the isolability ratio of f_i from the no fault case. Consider the running example. The global isolability ratio of fault f_1 from fault $f_2 = N(1, 0.1)$ for each of the generated residuals is as follows.

$$\begin{aligned} GI(f_1, f_2|r_1) &= 0.88 & GI(f_1, f_2|r_3) &= 0.37 \\ GI(f_1, f_2|r_2) &= 0 & GI(f_1, f_2|r_4) &= 0 \end{aligned} \tag{35}$$

We can see that r_1 is the only reliable residual to isolate f_1 from f_2.

3.4 Fault Detectability and Fault Isolability in the Presence of Noise and Uncertainty

As we mentioned earlier, residuals represent redundancies in the system equations, and they can form the basis for fault detection and isolation. Ideally, each residual should compute to a value of zero in the fault-free case, and residuals sensitive to a fault become nonzero when the fault occurs. In subsection 3.1.4, we used a structural approach to define fault detectability and fault isolability. Due to model uncertainties and measurement noise, the residual may deviate from zero even in the fault-free case. Therefore, we go beyond previous work [22, 107], and define fault detectability and isolability taking into account noise and uncertainties in the system using detectability and isolability ratios.

Definition 14. *(Detectable fault) A fault $f \in F$ is detectable in system S if there is a residual r, such that $D(f|r) > 0.5$.*

In this definition, we consider a fault detectable if the effect of fault on the residual is greater than the effects of noise and uncertainties on the residual. In this case, we can design a proper threshold to distinguish normal operation from faulty scenarios. Note that in *Definition* 21, we can replace detectability ratio with global detectability ratio based on the application. Fault isolability in the presence of noise and uncertainties is defined as:

Definition 15. *(Isolable fault) A fault $f_i \in F$ is isolable from fault $f_j \in F$ if there exists residual r, such that $I(f_i, f_j|r) > 0.5$.*

In this definition, we consider a fault f_i isolable from fault f_j, if there is a residual r where the effect of f_i on r is greater than the effects of f_j, noise and uncertainties in the model on r. Therefore, we can design a threshold in a way that the residual passes the threshold only when f_i occurs. Note that in *Definition* 23, we can replace isolability ratio with global isolability ratio based on the application.

3.5 Residual selection problem

In this section, our approach for selecting a subset of residuals for fault detection and isolation is presented.

3.5.1 Off-line Residual Selection

The total number of residuals that can be derived given a system model and a set of measurements grows exponentially as a function of the number of measurements (see equation (8)). Therefore, computing the detectability and isolability ratios of all the residuals on-line, is computationally expensive and in many cases infeasible. When the system's behavior trajectory is known beforehand, the detectability and isolability ratios of the residuals can be computed off-line and the minimum number of residuals that guarantees the specified required performance can be selected. This method minimizes the on-line computational cost.

The residual selection problem was presented in (9). As discussed, the residual selection problem is equivalent to a set covering problem, and, there is no polynomial algorithm for deriving the minimum set of residuals that meet the detectability and isolability performance constraints. However, formulating the problem as a binary integer linear programming (BILP) problem at least provides us with a systematic approach for finding the

minimal number of residuals that satisfy the required performance criteria. There are several tools[1] available for solving BILP optimization problems in an efficient manner.

Definition 16. *(Binary Integer Linear Programming Problem (BILP)) A Binary integer linear programming problem is a special case of an integer linear programming (ILP) optimization problem in which some or all the unknown variables to be solved for are required to be binary, and the constraints in the problem and the objective function, like ILP, are linear.*

The mathematical formulation of BILP is as follows.

$$\begin{aligned} &\min c^T x \\ &Ax \leq b, \\ &\exists x_b \subset x, \\ &\forall x_k \in x_b \Rightarrow x_k \in \{0,1\}. \end{aligned} \tag{36}$$

Vector c is the cost weights associated with the variables, x that have to be minimized using the linear optimization approach. Matrix A and vector b define the linear constraints imposed on the optimization problem, $Ax \leq b$, and x_b represents a subset of the variable set, x, that are binary valued [182].

To formulate problem (9) as a BILP problem a binary variable x_i for residual r_i is defined as follows.

$$x_i = \begin{cases} 1 & if \quad r_i \in R^* \\ 0 & if \quad r_i \notin R^*, \end{cases} \tag{37}$$

where R^* is the answer to problem (9). To minimize the number of residuals the cost vector is considered $c = I_{l_r * 1}$, where l_r is the number of residuals in $\mathscr{R}$. $c = I_{l_r * 1}$ implies that the optimization problem considers the same weight for each residual, and minimizes the total number of residuals. In the running example, there are four residual candidates

[1] For example, see http://www.mathworks.com/help/optim/ug/mixed-integer-linear-programming-algorithms.html in the Matlab™linear integer programming toolbox.

(see equations (15) and (16)) and, therefore, $c^T = [1\ 1\ 1\ 1\ ]$. Assume that for fault f_l a set of residuals $R_l = \{r_{l1}, r_{l2}, ..., r_{lg}\}$ has detectability ratio above the minimum required detectability ratio for operating region m. To have an acceptable detectability performance for f_l in this operating region at least one of these residuals has to be in the final set R^*. Mathematically speaking, the following constraint has to be considered.

$$\sum_{i=1}^{i=g} -x_{li} \leq -1. \tag{38}$$

To solve the isolability problem, the same procedure is followed. Consider a case that $R_{f_m f_n} = \{r_{mn1}, r_{mn2}, ..., r_{mnh}\}$ is the set of residuals with isolability ratio of fault f_m from fault f_n above the minimum isolability requirement in an operating region m. To have acceptable isolability performance of f_m from f_n in this operating region the following constraint has to be considered.

$$\sum_{i=1}^{i=h} -x_{mni} \leq -1, \tag{39}$$

where x_{mni} is the binary variable associated with residual r_{mni}. Therefore, when there are p faults and q operating regions in the system p^2q constraints have to be considered in the optimization problem. The goal is to minimize the number of selected residuals while satisfying these constraints at the same time.

As an example, consider the running example when the minimum required detectability ratio and minimum required isolability ratio are $D_{req} = I_{req} = 0.8$, and the inputs to the system are $u_1 = sin(0.2t)$ and $u_2 = cos(0.2t)$. Using *Definition* 9, five operating regions are detected for the system in time interval $[0s\ 10s]$, and the optimal residual selection problem can be formally represented as:

$$\min \sum_{i=1}^{i=4} x_i \quad \text{Subject to :}$$

$$-\begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \leq \begin{pmatrix} -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \end{pmatrix} \tag{40}$$

$$\forall i \in \{1,2,3,4\} \quad \Rightarrow \quad x_i \in \{0,1\}.$$

Matlab function bintprog is used to solve this optimization problem and obtained the result: $x = [1,1,0,0]$. This means our diagnostic algorithm minimally needs r_1 and r_2 to achieve the required performance for the given trajectory. These two residuals can be used for on-line residual selection approach. For example, to detect f_1 the following on-line residual selection approach can be used.

$$r = \begin{cases} r_1 & if \qquad D(f_1|r_1) \geq D(f_1|r_2) \\ r_2 & otherwise. \end{cases} \tag{41}$$

The same approach can be applied to select residuals to detect f_2, isolate f_1 from f_2 and isolate f_2 from f_1 using $D(f_2)$, $I(f_1, f_2)$ and $I(f_2, f_1)$, respectively.

3.5.2 Dynamic Residual Selection

The approach discussed in the previous subsection assumed the system behavior trajectory is known, and used a BILP approach to derive the minimum number of residuals that guarantee the required performance across the different operating regions of system behavior. In this subsection, an algorithm for residual selection when the system trajectory is not known beforehand is proposed.

Algorithm 1 Dynamic Residual Selection

input: $\mathscr{R}$, $\mathscr{R}^*(k)$, S, T_r, k
output: $\mathscr{R}^*(k+1)$
if $k = 0$ **then**
  for each $f_i \in F$ **do**
    $\mathscr{R}^*(i,i,1) \leftarrow \max D(\mathscr{R}(i,i,0))$
  end for
  for each $f_j \& f_k \in F$ **do**
    $\mathscr{R}^*(i,j,1) \leftarrow \max I(\mathscr{R}(i,j,0))$
  end for
else
  for each $f_i \in F$ **do**
    if $D(\mathscr{R}^*(i,i,k)) < T_r$ **then**
      $\mathscr{R}^*(i,i,k) \leftarrow \max D(\mathscr{R}(i,i,k))$
    end if
  end for
  for each $f_j \& f_j \in F \& i \neq j$ **do**
    if $I(\mathscr{R}^*(i,j,k)) < T_r$ **then**
      $\mathscr{R}^*(i,j,k) \leftarrow \max I(\mathscr{R}(i,j,k))$
    end if
  end for
end if

The algorithm achieves the required detectability and isolability performances by switching to a new set of residuals, when one or more residuals performances drop below the

threshold. The algorithm starts by computing the detectability and isolability ratio of all the residuals at the start of system operation, it selects the best residual to detect each fault and the best residual to isolate each fault from every other fault, taking into account their detectability and isolability ratios at the initial point. The algorithm keeps track of the selected residuals on-line and if a residual's performance drops below the required performance, the algorithm recomputes the performances of all the residuals at that point and replaces the residuals, whose performance has fallen below the pre-specified threshold, by a subset that provides the highest performance gain at this point.

In Algorithm 1, $\mathscr{R}^*(i,i,k)$ is the residual that our algorithm uses to detect fault f_i, but at sample time k its performance falls below the pre-specified threshold, T_r. $\max D(\mathscr{R}(i,i,k))$ is the residual in $\mathscr{R}$ that provides maximum detectability ratio to detect f_i at sample time k. Similarly, $\mathscr{R}^*(i,j,k)$ was the selected residual to isolate fault f_i from fault f_j at sample time k. $\max I(\mathscr{R}(i,j,k))$ represents the new residual in $\mathscr{R}$ with maximum isolability ratio for faults f_i and f_j at sample time k. Dynamic residual selection tracks the detectability and isolability ratios of the selected residuals on-line, and replaces the residuals that do not perform well with a new set of residuals that provide the maximum increase in performance. This approach helps us achieve the required robustness to uncertainties and sensitivity to the faults in each region of system operation but there is no guarantee that the set of residuals being used is minimum. The advantage of this method is that the algorithm does not need to know the trajectory of the system behavior beforehand, and the method guarantees the required robustness in fault detection and isolation for all the system trajectories. Note that in both off-line residual selection algorithm and dynamic residual selection algorithm, we can replace detectability ratio and isolability ratio with global detectability ratio and global isolability ratio respectively.

3.6 Case Studies

3.6.1 The Reverse Osmosis System

The Advanced Water Recovery System (AWRS) is a part of the Advanced Life Support (ALS) system, which was designed and built at the NASA Johnson Space Center for long duration manned missions. During the mission, the AWRS converts wastewater in micro-gravity conditions to potable water for the astronauts. The block diagram of the AWRS is shown in Figure 6. The Biological Water Processing System (BWP) is designed to remove organic impurities, whereas the Reverse Osmosis System (RO) is designed to remove inorganic impurities from the wastewater. After these two stages, about 85% of the water is clean enough to be fed to the Post Processing System that employs ultraviolet light treatment to remove microbial matter. The remaining 15% containing mostly sludge is sent through the Air Evaporation System (AES), which uses an evaporation and condensation processes to recover the water. In this work, our residual selection method is applied to the RO system, whose behavior includes complex nonlinearities.

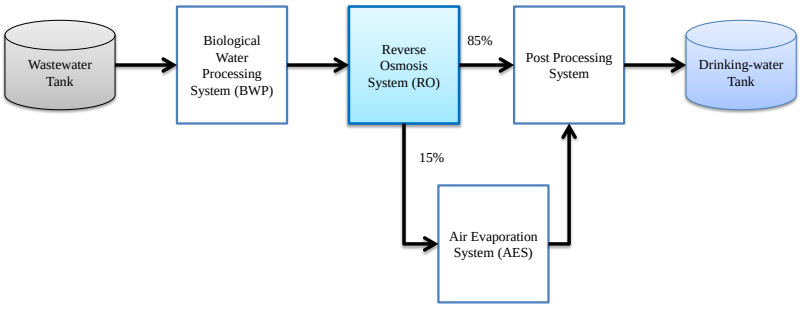

Figure 6: Advanced Water Recovery System (AWRS) [17].

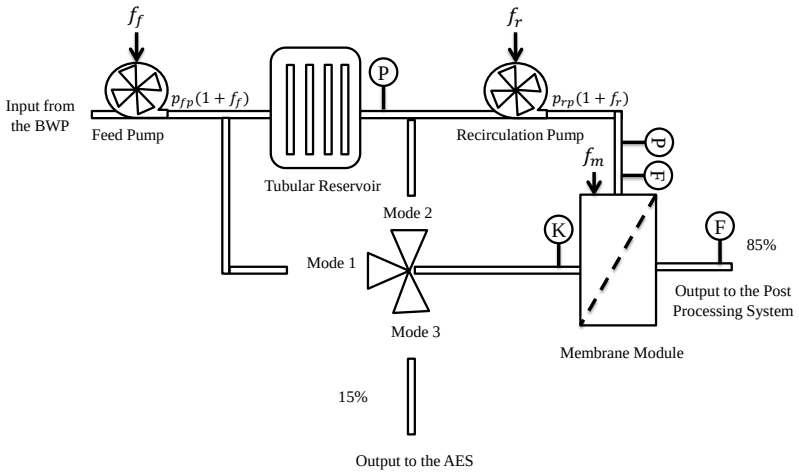

Figure 7: Reverse Osmosis System (RO).

3.6.1.1 The state space model for the RO system

The RO system, shown in Figure 7, operates in three modes that are controlled by a three-way valve. In this chapter, a diagnoser is developed for the first mode of operation, where the water circulates in the longer loop. In Chapter V, we develop a hybrid diagnoser for all the operating modes of the system. The first mode of the RO system can be modeled by a sixth order lumped parameter state space model, with state variables: f_{fp}, the volume flow rate generated by the pump, p_{tr}, the pressure of the fluid in the tubular reservoir, f_{rp}, the volume flow rate due to the recirculation pump, p_{memb} that represents the pressure of fluid at the membrane through which the clean water passes (but leaves the impurities behind), and two abstract variables, e_{Cbrine} and e_{Ck} that capture the dynamics of the impurities in the fluid as it circulates through the primary RO loop.

The feed pump pushes the partially purified water from the BWP into the main loop of the RO system at a nominal pressure p_{fp}. The rate of change of the volume flow rate, f_{fp} is given by: $\dot{f}_{fp} = \frac{\Delta p_{fp}}{I_{fp}}$, where Δp_{fp} is drop in pressure of the fluid across the feed pump and I_{fp} represents the inertia of the rotating elements of the feed pump. Taking into account the pump internal resistance to flow, R_{fp}, its uncertainty, $\delta_{R_{fp}}$, and the efficiency decrease in the feed pump, which is modeled by a multiplicative factor f_f, the pressure drop can be

computed, $\Delta p_{fp} = p_{fp}(1-f_f) - R_{fp}(1+\delta_{R_{fp}})f_{fp} - p_{tr}$ and the first state equation can be derived as:

$$\dot{f}_{fp} = \frac{1}{I_{fp}}(-R_{fp}(1+\delta_{R_{fp}})f_{fp} - p_{tr} + p_{fp}(1-f_f)). \tag{42}$$

The tubular reservoir with capacity value, C_{tr}, acts as a storage capacity that helps the system to keep water circulation rate steady. The net volume flow rate to the tubular reservoir, f_{tr}, is equal to the algebraic sum of the volume flow rates into and out of the tubular reservoir. The flow in is f_{fp} plus the flow from the membrane module that can be computed as the pressure difference between the membrane and the tubular reservoir, $p_{memb} - p_{tr}$, over the resistance of the pipe from the membrane module to the tubular reservoir, $R_{return_l}(1+\delta_{return})$, where δ_{return} is the uncertainty associated with the pipe resistance. The output volume flow rate from the tubular reservoir is equal to the recirculation pump volume flow rate, f_{rp}. Using these the net volume flow rate to the tubular reservoir can be computed as $f_{tr} = f_{fp} + \frac{p_{memb}-p_{tr}}{R_{return_l}(1+\delta_{return})} - f_{rp}$. Considering the uncertainty associated with the tubular reservoir capacitance, $\delta_{C_{tr}}$, the second state equation can be derived as:

$$\dot{p}_{Ctr} = \frac{1}{C_{tr}(1+\delta_{C_{tr}})}(f_{fp} + \frac{p_{memb} - p_{tr}}{R_{return_l}(1+\delta_{return})} - f_{rp}). \tag{43}$$

The recirculation pump boosts the liquid pressure by p_{rp}. The rate of change of pump's fluid flow rate, $\dot{f}_{rp}$, is given by ($\dot{f}_{rp} = \frac{\Delta p_{rp}}{I_{rp}}$), where Δp_{rp} represents drop in the fluid pressure inside the pump and I_{rp} represents the inertia of the rotating elements of the pump. The pump's internal resistance includes uncertainties, and is represented as $R_{rp}(1+\delta_{R_{rp}})$. The efficiency decrease in the recirculation pump, f_r, is the second fault parameter in the RO system. The pressure at the pump output can be computed as a function of the membrane module pressure, p_{memb} and the pressure drop in the pipe from the pump to the membrane module, $R_{forward}(1+\delta_{forward})f_{rp}$, where $R_{forward}(1+\delta_{forward})$ represents resistance of the pipe from the recirculation pump to the membrane (with the nominal value $R_{forward}$ and associated uncertainty $\delta_{forward}$). From these components, the third state equation is

derived as:

$$\dot{f}_{rp} = \frac{1}{I_{rp}}(-R_{rp}(1+\delta_{R_{rp}})f_{rp} - R_{forward}(1+\delta_{forward})f_{rp} - p_{memb} + p_{rp}(1-f_r)). \quad (44)$$

The membrane is a key component for removing particulate matter from the water in the RO system. The recirculation pump pushes the input water at high pressure onto the membrane. The purified water that comes out of the other side of the membrane is fed to the Post Processing System, and the remaining water recirculates in the RO primary loop. As more and more water passes through the membrane, the water remaining in the loop has an increased concentration of impurities. At the same time particulate matter that collects on the membrane, increases its resistance to flow. The membrane chamber can be modeled as a combination of a capacity, C_{memb}, and a resistance, R_{memb}. The rate of membrane pressure variation, $\dot{p}_{memb}$, is given by ($\dot{p}_{memb} = \frac{f_{memb}}{C_{memb}}$), where f_{memb} is the net volume flow rate to the membrane.

The uncertainty associated with C_{memb} is modeled with an unknown parameter, $\delta_{C_{memb}}$, and the membrane capacitance including this uncertainty is represented as: $C_{memb}(1+\delta_{C_{memb}})$. In previous work, Carl et al. [27] empirically derived membrane resistance: $R_{memb} = 0.202(4.137 * 10^{11}(\frac{e_{Ck}-12000}{165} + 29))$. Note that R_{memb} increases as the impurities in the water, and, therefore, the water conductivity, e_{Ck}, increases. The membrane clogging factor f_m is the third fault in the system, implying that its resistance is higher than nominal, i.e., $R_{memb}(1+f_m)$. We compute the net volume flow rate to the membrane, f_{memb}, as an algebraic sum of the input volume flow rate from the recirculation pump, f_{rp}, and the output volume flow rates to the post processing system, $f_{out} = \frac{p_{memb}}{R_{memb}(1+f_m)}$, and the return volume flow rates to the tubular reservoir, $\frac{p_{pmemb}-p_{tr}}{R_{return_l}(1+\delta_{return})}$. Using this the fourth state equation is derived,

Table 1: RO System Parameters and Inputs

Parameter	Name	Value and unit (SI)
Feed pump inertance	I_{fp}	$0.1\ N.s^2/m^5$
Recirculation pump inertance	I_{rp}	$2\ N.s^2/m^5$
Feed pump energy dissipation	R_{fp}	$0.1\ N./m^5$
Recirculation pump energy dissipation	R_{rp}	$0.1\ N./m^5$
Conductivity capacitor	C_k	$565\ m^5/N$
Hydraulic resistance	$R_{forward}$	$70\ N./m^5$
Capacitance of the tubular reservoir	C_{tr}	$1.5\ m^5/N$
Hydraulic resistance (long loop)	R_{return_l}	$15\ N./m^5$
Hydraulic resistance (short loop)	R_{return_s}	$8\ N./m^5$
Hydraulic resistance	$R_{return_{ASE}}$	$5\ N./m^5$
Capacitance of the membrane module	C_{memb}	$0.6\ m^5/N$
Brine capacitor	C_{brine}	$8\ m^5/N$
Feed pump nominal pressure	p_{fp}	$1\ N./m^2$
Recirculation pump nominal pressure	p_{rp}	$160\ N./m^2$

$$\dot{p}_{memb} = \frac{1}{C_{memb}(1+\delta_{C_{memb}})}(f_{rp} - \frac{p_{memb}}{R_{memb}(1+f_m)} - \frac{p_{memb}-p_{tr}}{R_{return_l}(1+\delta_{return})}). \tag{45}$$

To complete the dynamic state space model, the conductivity of the fluid is represented as a state variable, making the assumption that the conductivity of the water increases every cycle through the flow loop, with the increase being proportional to the flow of liquid out of the membrane. This generates the two last state equations:

$$\begin{aligned} \dot{e}_{Cbrine} &= \frac{p_{memb}-p_{tr}}{1.667*10^{-8}C_{brine}R_{return_l}} \\ \dot{e}_{Ck} &= \frac{f_{rp}}{C_k}(6e_{Cbrine}+0.1)/(1.667*10^{-8}), \end{aligned} \tag{46}$$

where C_{brine} and C_k are conductivity parameters and are represented in Table 1. The system inputs are the feed pump pressure, p_{fp} and the recirculation pump pressure, p_{rp}. More details of the RO modeling scheme are presented in [17, 27, 169].

There are five sensors in the system. These sensors measure $y_1 = p_{tr}$, $y_2 = p_{mem}$, $y_3 = f_{out}$, $y_4 = e_{Cbrine}$, and $y_5 = e_{Ck}$. The system inputs, p_{fp} and p_{rp}, are assumed to be known as well. The RO system's parameters and input signals in this case study are presented in Table 1.

3.6.1.2 Residual generation and residual selection for the RO system

Figure 8 shows the schematic for residual generation and selection in the RO system case study. Running the fault diagnosis toolbox [68], produces 380 residuals for the RO system. For efficiency, our residual selection algorithm selects a minimum subset of residuals with an acceptable robustness performance. As it is shown in Figure 8, the residual selection algorithm requires 1) the set of residuals, 2) the minimum required robustness performance, 3) the system trajectory, 4) the upper bounds of parameter uncertainties, 5) the lower and the upper bounds of system faults, as the inputs.

For this case study, the minimum required detectability and isolability ratios are considered to be 0.8. To select the residuals, the algorithm needs to know the system trajectory. In this case study the system inputs are $p_{fp} = 1N./m^2$ and $p_{rp} = 160N./m^2$ during the simulation time $T = [0s\ 10800s]$. The minimum and maximum fault amplitude are 25% and 100% deviation, respectively. For example, when the efficiency decrease occurs in the feed pump, the pump pressure, which is equal to 1 $N./m^2$ in the nominal operation, is expected to be $0 \leq p_{fp} \leq 0.75$. For our experiment, a 1% maximum uncertainty is assumed for all uncertain parameters. Therefore, $0.099 \leq R_{fp} \leq 0.101$ and $1.485 \leq C_{tr} \leq 1.515$. Our algorithm selects three residuals that guarantee the required detectability and isolability performance for the given trajectory.

$$r_1 = y_2 - y_3 * 0.202(4.137 * 10^{11}(\frac{y_5 - 12000}{165} + 29)). \tag{47}$$

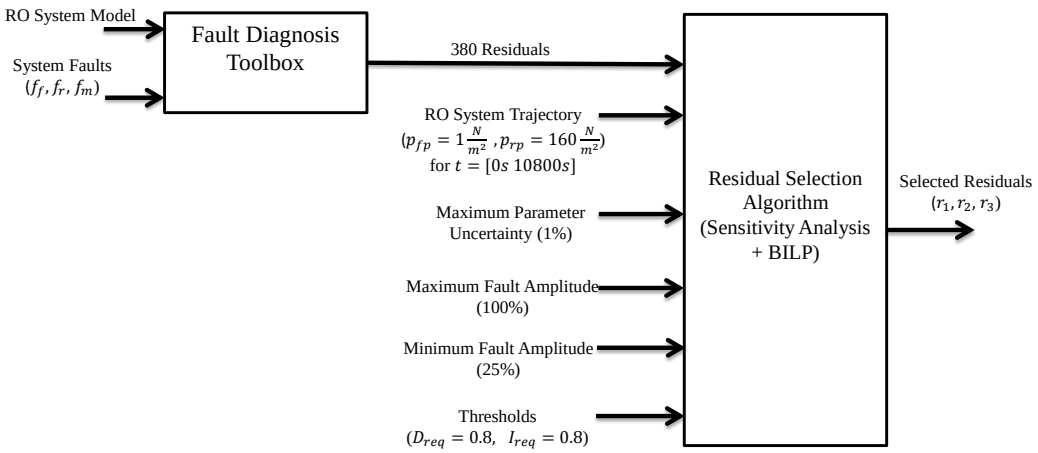

Figure 8: Residual generation and selection for the RO system.

$$
\begin{aligned}
z_{21} = (\dot{y_2} + \frac{y_2}{0.202(4.137 * 10^{11}(\frac{y_5 - 12000}{165} + 29))} + \frac{y_2}{R_{return_l}} + \\
\frac{y_2 - z_{22}}{C_{memb} R_{forward}}))(\frac{1}{\frac{1}{C_{memb} R_{forward}} + \frac{1}{R_{return_l}}}) \\
\dot{z}_{22} = \frac{1}{I_{rp}}(u_2 - R_{rp} z_{22} - \frac{1}{R_{forward}}(z_{21} + z_{22} - y_2)) \\
r_2 = y_1 - z_{21}.
\end{aligned}
\tag{48}
$$

$$
\begin{aligned}
z_{31} = (R_{forward})(C_{memb}\dot{y_2} + \frac{y_2}{0.202(4.137 * 10^{11}(\frac{y_5 - 12000}{165} + 29))} \\
+ \frac{y_2 - z_{32}}{R_{return_l}} - \frac{z_{32} - y_2}{R_{forward}}) \\
\dot{z}_{32} = \frac{1}{z_2} * (z_3 + \frac{y_2 - z_{32}}{R_{return_l}} - \frac{1}{R_{forward}}(z_2 + z_{31} - y_2)) \\
\dot{z}_3 = \frac{1}{I_{fp}}(-R_{fp} z_3 - z_2 + u_1) \\
r_3 = y_1 - z_{31}.
\end{aligned}
\tag{49}
$$

The detectability ratios of the system faults f_m, f_r, and f_f using the selected residuals r_1, r_2 and r_3 are shown in Figure 9 for the assumed behavior trajectory in the phase 1 operation of the RO system. The detectability ratio of fault f_m is above the pre-specified performance

threshold, $D_{req} = 0.8$, for residual r_1. The detectability ratio of fault f_r is above the pre-specified threshold for residual r_2. Finally, r_3 provides the required performance to detect f_f.

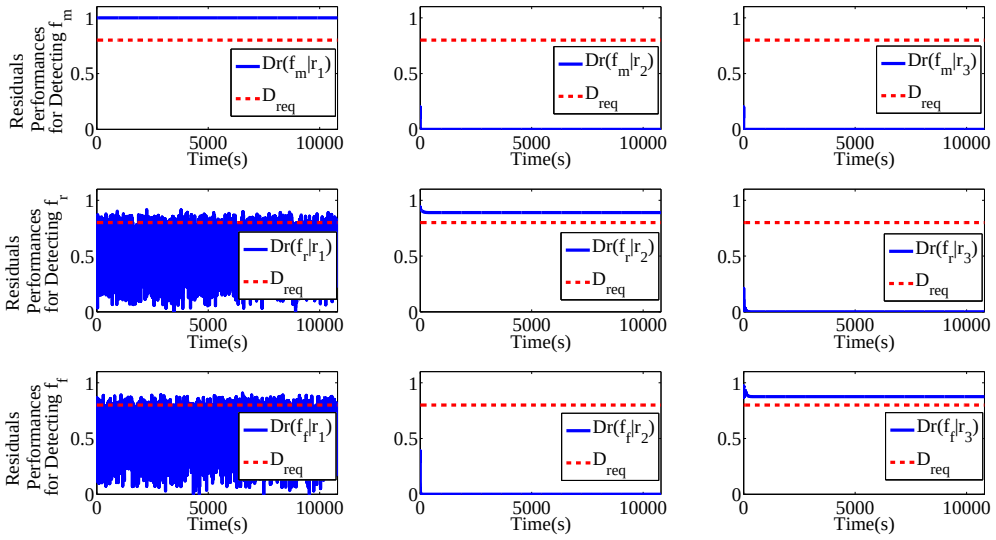

Figure 9: Detectability ratios of faults f_f, f_m and f_r using r_1, r_2 and r_3 .

Figure 10 shows the performances of the selected residuals for fault isolation. The plots with blue background represent the isolability ratios that are above the pre-specified performance threshold, $I_{req} = 0.8$, for the given trajectory. The isolability ratios of fault f_m from faults f_f and f_r are above the pre-specified threshold, I_{req}, using r_1. r_2 can be used to isolate f_r from f_f and f_m, and r_3 is a reliable candidate to isolate f_f from f_m and f_r robustly. In this example, the choice of the three residuals is easily justified. The first fault f_m can be detected and isolated from f_r and f_f using residual r_1. The second fault, f_r can be detected and isolated from the two other faults using r_2. Finally, r_3 provides the required performance to detect and isolate fault f_f.

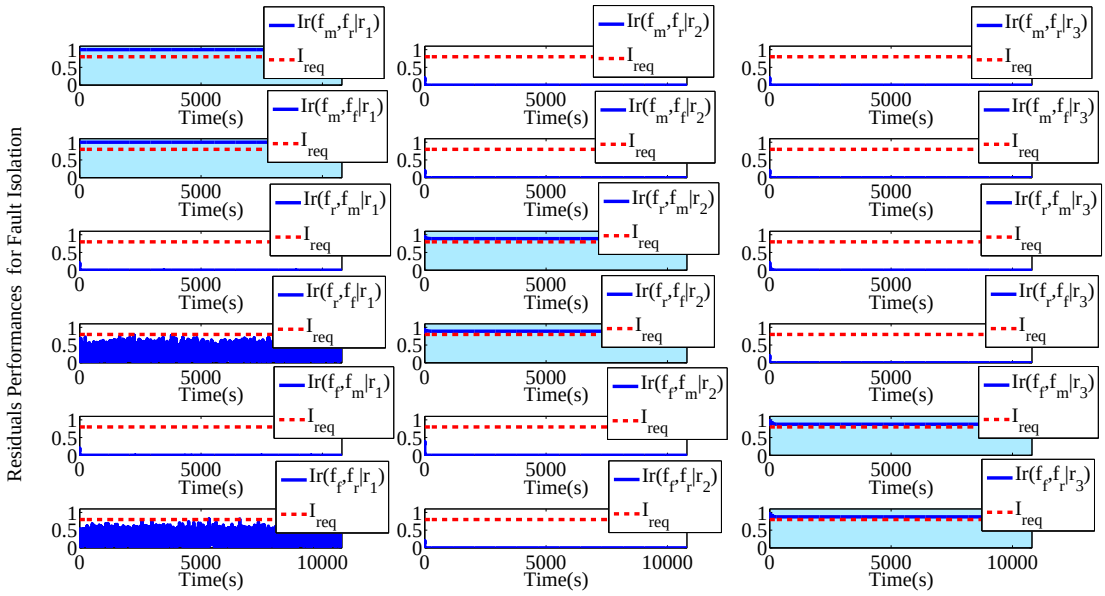

Figure 10: Isolability ratios of the system faults using r_1, r_2 and r_3.

In this case study, three hundred and eighty residuals have been generated, but three of them are enough to achieve the robustness performance. Higher detection and isolation ratio thresholds may have required a greater number of residuals. Similarly, increasing the levels of uncertainty in the model parameters may have required a higher number of residuals. For example, if we consider the maximum uncertainty in the parameters 5% instead of 1%, the algorithm selects five residuals to achieve the same robustness performance.

3.6.1.3 Fault detection and isolation in the RO system

In this section, the derived residuals that meet our detectability and isolability criteria are applied for on-line diagnosis. In realistic situations, statistical tests are employed to accommodate for modeling errors and measurement noise and reduce the false alarm rate for detection and isolation. In previous work, Biswas et al. [18] developed a Z-test that has produced high accuracy in detection tasks, while ensuring low false alarm rates. The effectiveness of the residual selection scheme combined with a Z-test based fault detector in detecting and isolating abrupt faults is demonstrated through different fault detection and

Table 2: Fault Detection and Isolation Performance

Fault	Fault Magnitude	Detection Time	Isolation Time
f_m	0.25	$0.1s$	$0.1s$
f_r	0.25	$0.3s$	$0.3s$
f_f	0.25	$0.4s$	$0.4s$

isolation scenarios in the RO system. For all of our detection and isolation experiments, the confidence level for the Z-test is 95%.

In the first scenario, an abrupt membrane clogging fault $f_m = 0.25$ occurs at $t = 5400s$. Figure 11 shows that r_1, which has a high detectability ratio for f_m, immediately changes as f_m occurs, but the other residuals as expected from the performance analysis do not react to the fault. In another scenario, an abrupt actuator fault, $f_r = 0.25$, occurs at $t = 5400s$. Figure 11 shows that residual r_2, which has a high detectability ratio for f_r, jumps to a non zero value, but the other residuals do not react in any significant way. Finally, consider the case where an abrupt fault $f_f = 0.25$ occurs at $t = 5400s$. Figure 11 shows the expected result. Note that because of uncertainties in the system the residuals have a non zero value, even when there is no fault in the system. Table 2 shows the fault detection and isolation performance for each scenario.

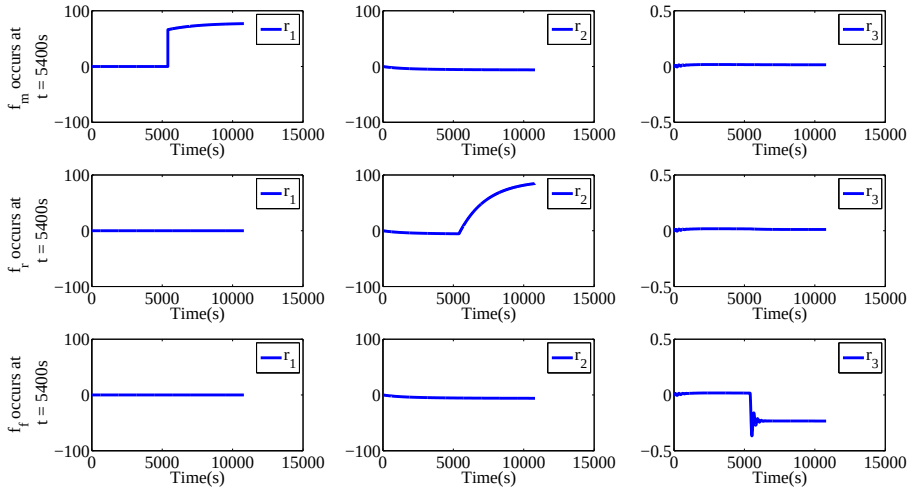

Figure 11: Residuals outputs in each fault scenario.

3.6.2 The Hot Water System

The hot water system is designed to provide hot water and energy for the heating system of Lentz Public Health Center, Nashville, TN. The system consists of three identical variable speed drive pumps with nominal pressure, $p_n = 51.15psi$, nominal rotational speed $\omega_n = 1750rpm$ and nominal flow rate $q_n = 200gpm$. The goal is to provide hot water with a required pressure to each of the building three floors. The operators can adjust the pump speeds to control the water pressure. There are two differential pressure sensors in the system. The first sensor is located at the output of the hot water system and measures the differential pressure in the in the entire system, p_h. The second one is located in the hydraulically most remote point of the building heating water piping system and measures the differential pressure in the third floor, p_t.

The hot water system also has three sensors that measure the rotational speed of each pump. The sampling rate is 1 sample per minute. In our experiment, we record the sensor values for 10000 minutes. To validate our diagnosis algorithm, we create an artificial leak by slightly opening a pressure relief valve in the system 7200 minutes after the start of the experiment. We closed the valve after 100 minutes, so the system transits to normal operation. The block diagram of the hot water system is shown in Figure 12 .

Table 3: Hot water System Parameters and Uncertainty Distributions

Parameter	Name	Value
Pump nominal pressure	p_n	51.51 psi
Pump nominal rotational speed	ω_n	1750 rpm
Pump nominal flow rate	q_n	200 gpm
Units Constant	K	20
Viscosity compensation factor	V	1
Specific gravity	S	1
Pipe resistance (first floor)	R_a	0.7165458 $Lohms$
Resistance uncertainty (first floor)	δ_{R_a}	$N(0, 6.250977e-08)$
Pipe resistance (third floor)	R_b	0.7028942 $Lohms$
Resistance uncertainty (third floor)	δ_{R_b}	$N(0, 6.452614e-08)$

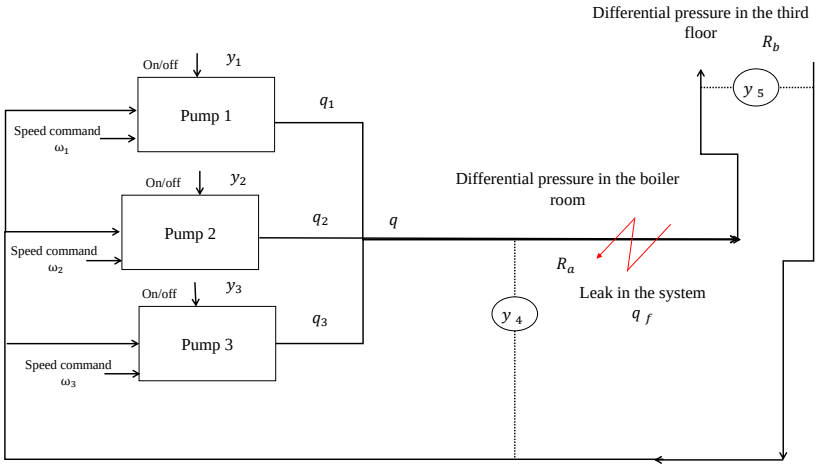

Figure 12: Hot water system.

Using the affinity laws [76], we can compute pump i flow rate at any given rotational speed, ω_i, as follows.

$$q_i(\omega_i) = (\frac{\omega_i}{\omega_n})q_n. \tag{50}$$

As we mentioned above, each pump has a tachometer to measure its rotational speed. We

represent these measurements as

$$\begin{aligned} y_1 &= \omega_1 + \delta_1 \\ y_2 &= \omega_2 + \delta_2 \\ y_3 &= \omega_3 + \delta_3, \end{aligned} \tag{51}$$

where y_i represents measurement i and δ_i represents noise in the measurement. Figure 13 shows the first three measurements in the hot water system. We can see that during our experiment, only one of the pumps is operating at each moment.

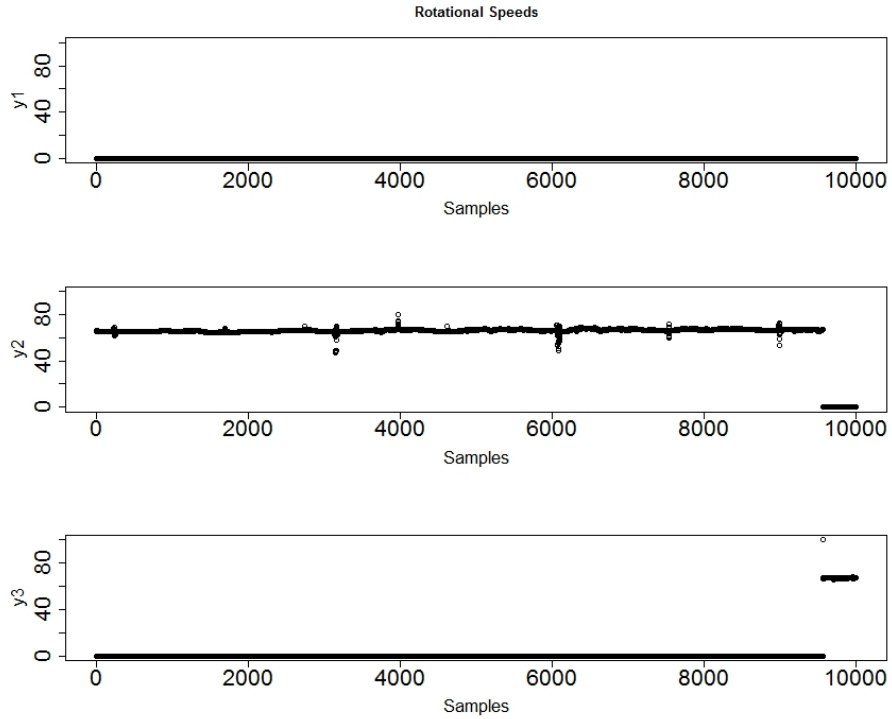

Figure 13: $y_{1:3}$ **in the hot water system.**

The hot water system is a closed loop system. This means the flow rate at each given point in the loop is equal to the total flow rate generated by the pumps.

$$q = q_1 + q_2 + q_3, \tag{52}$$

where q is the system flow rate. Using the Lohm laws, we can derive the following equation

$$KV\sqrt{\frac{p_h}{S}} = R_a(1+\delta_{R_a})q \tag{53}$$

where K is a constant to correct different units of measure. In this case study, we represents pressures in *psi* and flow rates in gpm and, therefore, K is equal 20. S is specific gravity, V is viscosity correction factor, R_a is the total pipe resistance outside of the hot water system, and δ_{R_a} represents the uncertainty in R_a. As we mentioned earlier, p_h is measured by a sensor in the hot water system. We represent the fourth measurement, y_4, and the noise in the sensor, δ_5, as

$$y_4 = p_h + \delta_4. \tag{54}$$

A possible leak in the system decreases the flow rate. However, when there is no fault in the system, the flow rate stay the same in the entire loop and the flow rate in the third floor is equal to the flow rate in the hot water system.

$$q_t = q, \tag{55}$$

where q_t is the flow rate in the third floor. The Lohm laws can be applied to derive the following equation for the third floor.

$$KV\sqrt{\frac{p_t}{S}} = R_b(1+\delta_{R_b})q_t \tag{56}$$

where R_b is the pipe resistance in the third floor, and δ_{R_b} represents the resistance uncertainty. We assume the uncertainties have normal distribution and use the nominal data from Lentz Public Health Center to estimate pipe resistances and uncertainty distributions. Table 3 represents the parameters. p_t is the last measurement in the system.

$$y_5 = p_t + \delta_5, \tag{57}$$

where y_5 is the sensor value and δ_5 represents the noise. Figure 14 shows the last two measurements in the system.

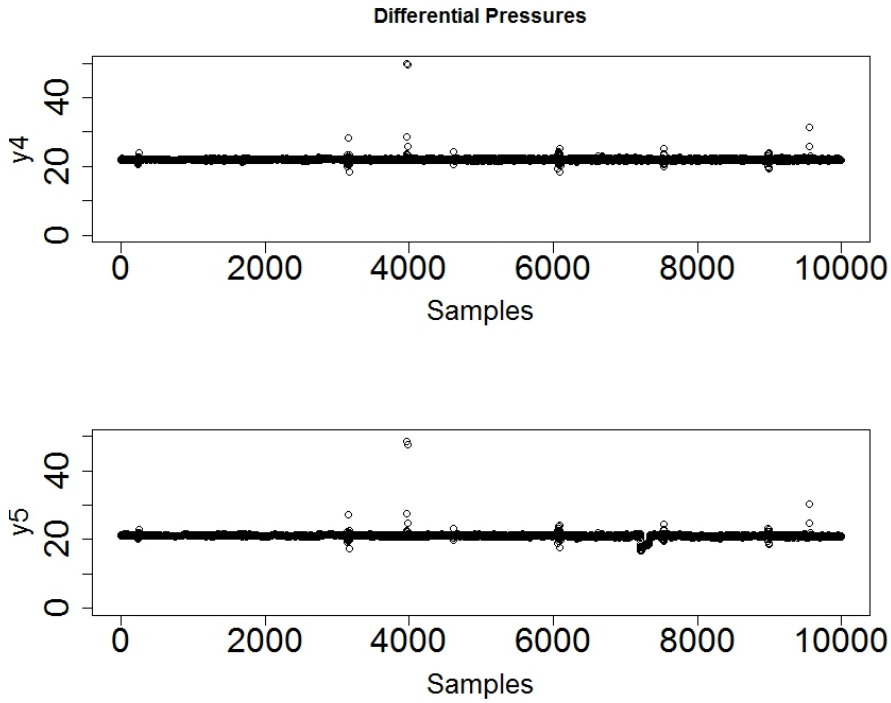

Figure 14: $y_{4:5}$ **in the hot water system.**

To detect the leak in the hot water system, we use equations (50) to (56) to derive the following residuals.

$$\begin{aligned} r_1 &= KV\left(\frac{1}{R_a}\sqrt{\frac{y_4}{S}} - \frac{1}{R_b}\sqrt{\frac{y_5}{S}}\right) \\ r_2 &= \left(\sum_{i=1}^{3}\left(\frac{y_i}{\omega_n}\right)q_n\right) - \frac{KV}{R_a}\sqrt{\frac{y_4}{S}} \\ r_3 &= \left(\sum_{i=1}^{3}\left(\frac{y_i}{\omega_n}\right)q_n\right) - \frac{KV}{R_b}\sqrt{\frac{y_5}{S}} \end{aligned} \tag{58}$$

To analyze the detectability performance of each residual, we use the nominal and faulty data from Lentz public health center to compute the global detectability ratio of

each residual.

$$GD(q_t|r_1) = 0.8847921$$
$$GD(q_t|r_2) = 0.2660716 \tag{59}$$
$$GD(q_t|r_3) = 0.4135846$$

Therefore, only r_1 can be used to reliably detect q_f. Note that to compute global detectability ratio, we do not need to estimate fault and uncertainties distributions. We only need to compute variance of each residual for normal and faulty data.

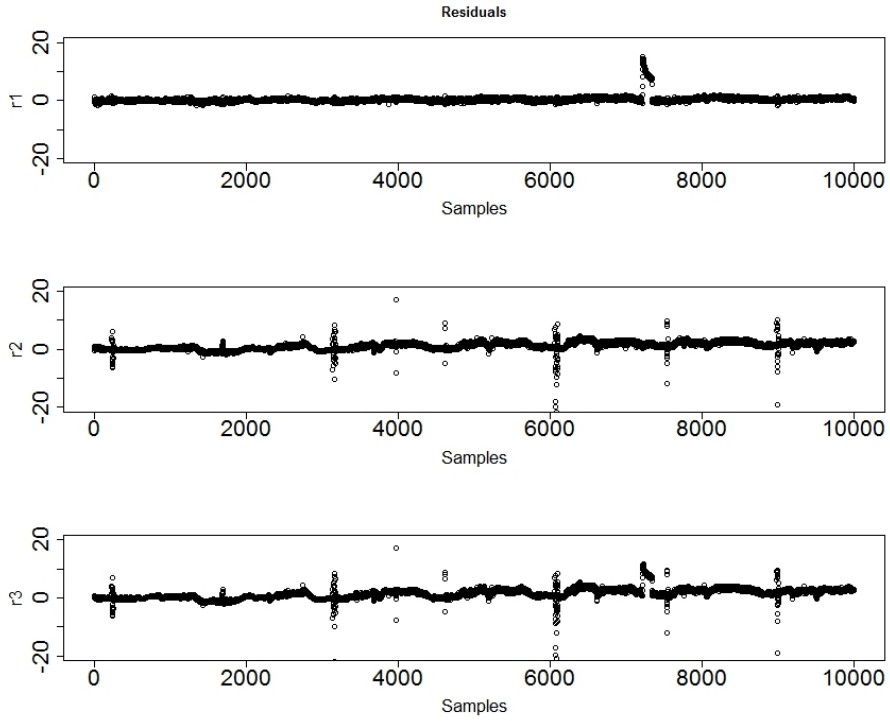

Figure 15: Residuals in the hot water system.

The leak occurs at sample point 7200 and lasts till sample point 7300. Figure 15 shows the residuals. We can see that r_1 is very sensitive to the leak and r_2 is not sensitive to the leak. r_3 is the most interesting case. Figure 15 shows that r_3 is sensitive to the leak. However, the global detectability ratio of q_t given r_3 is less than 0.5 and therefore, we do not expect r_3 to detect q_t. Figure 15 shows that r_3 is also sensitive to noise and uncertainties

Table 4: Fault Detection Performance

Fault	False Positive Rate	False Negative Rate Time
r_1	0%	0%
r_2	100%	1.27%
r_3	37%	0.76%

in the system and therefore, it is not reliable to detect q_t. We apply a Z-test with 95% confidence level to detect the leak. Figure 16 shows the output of the Z-test for each residual.

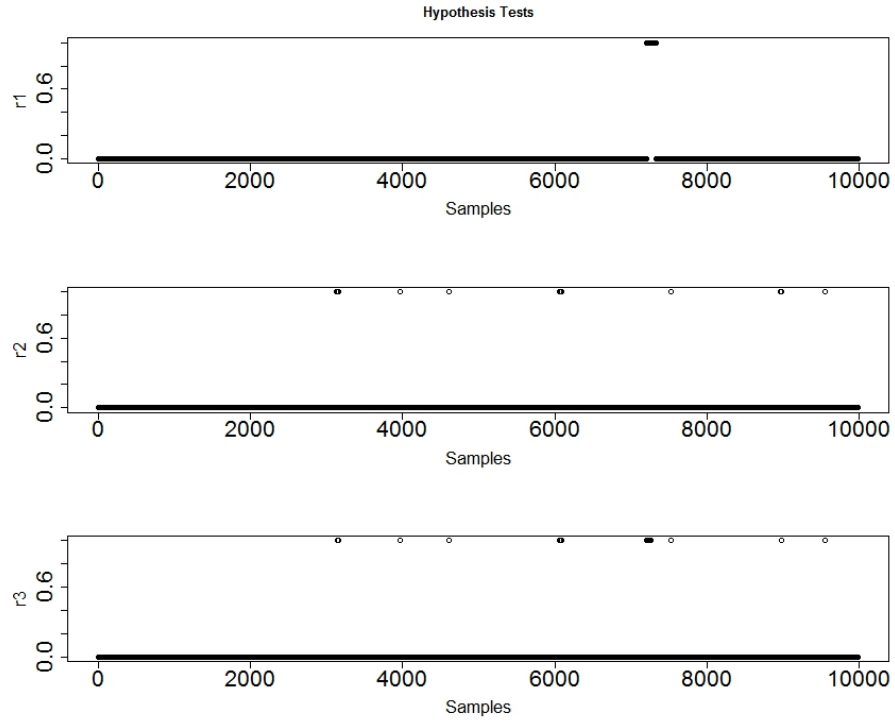

Figure 16: Hypothesis test outputs for the hot water system residuals.

Table 4 shows fault detection performance for each residual. r_1 detects the fault with no false alarm. r_2 has 100% false positive rate which means it does not detect the fault. r_3 only detects 37% of faulty sample points.

3.7 Conclusions

In this chapter, a general framework for automated residual generation and robust residual selection for nonlinear dynamic systems has been developed. Detectability and isolability measures were defined to quantify the performance of generated residuals in nonlinear systems using derivative-based sensitivity analysis. The derivative-based approach is the most common method to perform sensitivity analysis. The derivative-based approach is easy to understand and computationally efficient. However, the derivative-based approach only determines the effect of uncertainties at the single point at which the derivative is constructed. For linear and smooth nonlinear systems, the effect of uncertainties in other operation points can be easily determined by extrapolation. For stiff nonlinear system this can lead to a significant error. To overcome to this problem, we defined global detectability and global isolability ratios using global sensitivity analysis. In comparison with the derivative-based approach, global sensitivity approach increases the computational complexity but generates more robust measures for residual selection.

The robust residual selection problem was formulated as a binary integer linear programming (BILP) optimization problem to select a subset of residuals to achieve the prespecified performances. Unlike our previous work [104], the algorithm developed in this work guarantees the number of selected residuals is minimum. This minimizes the computational cost of on-line fault detection and isolation. Moreover, the algorithm identifies the regions of operation automatically. For the cases where the trajectory is unknown, the dynamic residual selection algorithm can be applied. Developing an automated robust residual evaluation and selection algorithm for nonlinear systems that fulfills the required performance over multiple operating regions is novel, and has not been discussed in the literature. Our experimental studies demonstrate the effectiveness of our approach.

CHAPTER IV

DISTRIBUTED FAULT DETECTION AND ISOLATION

The increasing complexity and size of cyber-physical systems (e.g., aircraft, manufacturing processes, and power generation plants) is making it hard to develop centralized diagnosers that are reliable and efficient. On the other hand, advances in networking technology, along with availability of inexpensive sensors and processors, is causing a shift in focus from centralized to more distributed diagnosers. This chapter develops two structural diagnosis approaches; 1) MSO-based, and 2) equation-based for distributed fault detection and isolation (FDI). The first approach is based on Minimal Structurally Overdetermined (MSO) set selection. Each MSO represents an analytical redundancy relation in the system and can be used for residual generation.Even though we do not discuss robustness in this chapter, it is straight forward to extend our first approach to robust distributed diagnosis by considering residuals robustness performance in the selection process as discussed in Chapter III.

The fist method provides globally correct diagnosis results and guarantees that the subsystems share the minimum number of measurements, implying that we minimize the communication of measurement streams across subsystems of the global system. However, the total number of residuals is exponential in terms of the system measurements. This increases the computational cost of the solution. To avoid the computational complexities of dealing with a large number of residuals, we develop another distributed diagnosis method based on system equations in this chapter. The second algorithm is computationally efficient. Moreover, it does not use the global model in the design process of the supervisory system. This makes the algorithm suitable for large, complex systems where global systems models are likely to be unavailable or unknown. We compare the diagnosis performances and the computational costs of the proposed algorithms and clarify the pros and cons of

each method. We then demonstrate through a case study the results obtained from each of the proposed methods.

The rest of this chapter is organized as follows. Section 4.1 presents basic definitions and the running example of the chapter. Our first method for distributed diagnosis is presented in Section 4.2. Section 4.3 presents our second approach for distributed diagnosis. Section 4.4 compares computational complexities of the proposed methods. Section 4.5 presents the case study and Section 4.6 presents the summary and conclusions of the chapter and points out the advantages and disadvantages of each approach.

4.1 Basic Definitions and Running Example

This section introduces the basic concepts associated with distributed diagnosis of dynamic systems. The system model S is defined as follows.

Definition 17 (System model)**.** *A system model S is a four-tuple: (V, M, E, F), where V is the set of variables, M is the set of measurements, E is the set of equations and F is the set of system faults.*

We use a four tank system example to describe the nature of the problem, and use this system as a running example to discuss the algorithms for distributed diagnosis presented in this chapter. Fig. 17 illustrates the four tank system model. We assume each tank, and the outlet pipe to its right, constitute a subsystem. Therefore, this system has four subsystems. Two of the subsystems, 1 and 3, also have inflows into their tanks. We assume the subsystems are disjoint, i.e., they have no overlapping components. Associated with each subsystem are a set of measurements that are shown as encircled variables in the figure.

More generally, we assume the system, S has n pre-defined subsystems, $S_1, S_2, ...S_n$. Each subsystem model is defined as:

Definition 18 (Subsystem model)**.** *A subsystem model of system model S, S_i $(1 \leq i \leq k)$*

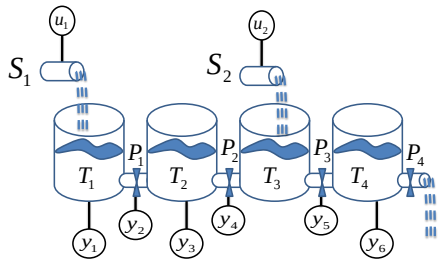

Figure 17: Running example: Four Tank System.

is also a four-tuple: (V_i, M_i, E_i, F_i), where $V_i \subseteq V$, $M_i \subseteq M$, $E_i \subseteq E$ and $F_i \subseteq F$. Also, $S_1 \cup S_2 \cup ....S_k = S$.

For illustration, the first subsystem in our running example is described by the following set of equations:

$$\begin{aligned} e_1 &: \dot{p}_1 = \frac{1}{C_{T1} + f_1}(q_{in1} - q_1) & e_4 &: q_{in1} = u_1 \\ e_2 &: q_1 = \frac{p_1 - p_2}{R_{P1} + f_2} & e_5 &: p_1 = y_1 \\ e_3 &: p_1 = \int \dot{p}_1 \, dt & e_6 &: q_1 = y_2. \end{aligned} \tag{60}$$

Therefore, $E_1 = \{e_1, e_2, e_3, e_4, e_5, e_6\}$ defines the set of equations, $V_1 = \{\dot{p}_1, p_1, p_2, q_{in1}, q_1\}$ defines the set of subsystem unknown variables, $M_1 = \{u_1, y_1, y_2\}$ defines the set of subsystem known variables (measurements), and $F_1 = \{f_1, f_2\}$ defines the set of faults associated with this subsystem model. We assume the system parameters, (C_{T1}, and R_{P1} in the first subsystem), are known.

Similarly, the second subsystem model is defined by the following equations:

$$\begin{aligned} e_7 &: \dot{p}_2 = \frac{1}{C_{T2} + f_3}(q_1 - q_2) & & \\ e_8 &: q_2 = \frac{p_2 - p_3}{R_{P2} + f_4} & e_{10} &: p_2 = y_3 \\ e_9 &: p_2 = \int \dot{p}_2 \, dt & e_{11} &: q_2 = y_4. \end{aligned} \tag{61}$$

For this subsystem the set of equations is $E_2 = \{e_7, e_8, e_9, e_{10}, e_{11}\}$, the set of variable is

$V_2 = \{\dot{p}_2, p_2, p_3, q_1, q_2\}$, the set of measurements is $M_2 = \{y_2, y_4\}$, and $F_2 = \{f_3, f_4\}$ is the set of faults. We assume there are no overlapping components among the subsystems. However, the subsystems may share variables at their interface, e.g., liquid flowrate at outlet of pipe = liquid flowrate at input to connected tank.

Definition 19 (First Order Connected Subsystems)**.** *Two subsystems, S_i and S_j are defined to be first order connected if and only if they have at least one shared variable.*

In the running example, subsystems S_1 and S_2 are first order connected and their shared variables are $V_1 \cap V_2 = \{p_2, , q_1\}$. The two other subsystems in the running example are:

$$\begin{aligned} e_{12} &: \dot{p}_3 = \frac{1}{C_{T3}}(q_{in2} + q_2 - q_3) \\ e_{13} &: q_3 = \frac{p_3 - p_4}{R_{P3} + f_5} \\ e_{14} &: p_3 = \int \dot{p}_3 \, dt \end{aligned} \qquad \begin{aligned} e_{15} &: q_{in2} = u_2 \\ e_{16} &: q_3 = y_5. \end{aligned} \tag{62}$$

$$\begin{aligned} e_{17} &: \dot{p}_4 = \frac{1}{C_{T4} + f_6}(q_3 - q_4) \\ e_{18} &: q_4 = \frac{p_4}{R_{P4}} \end{aligned} \qquad \begin{aligned} e_{19} &: p_4 = \int \dot{p}_4 \, dt \\ e_{20} &: p_4 = y_6. \end{aligned} \tag{63}$$

In more general terms, i_{th} order connected subsystem models are defined as follows.

Definition 20 (i_{th} Order Connected Subsystems)**.** *Two subsystems, S_k and S_j are defined to be ith order connected if and only if there exists a subsystem model S_m that is $(i-1)$th order connected to S_k, and is first-order connected to S_j, or S_m is $(i-1)$th order connected to S_j, and is first-order connected to S_k .*

For example in the four tank system, S_1 and S_3 are second order connected because both of them are first order connected to S_2. In this chapter, we use MSO sets [106] as the primary conceptual approach for fault detection and isolation. The formal definitions of Structurally Overdetermined (SO) and MSO sets are presented in definition 3 and definition 4. Consider subsystem S_1 of the four tank system in equation (60). Using the fault

diagnosis toolbox [68] , we can compute the only minimal structurally overdetermined set in this subsystem as $MSO_{11} = (E_{11}, V_{11}, M_{11}, F_{11})$, where $E_{11} = \{e_1, e_3, e_4, e_5, e_6\}$, $V_{11} = \{\dot{p}_1, p_1, q_{in1}, q_1\}$, $M_{11} = \{u_1, y_1, y_2\}$ and $F_{11} = \{f_1\}$. MSOs represent the redundancies in the system and can form the basis for fault detection and isolation. Global and local fault detectability are defined as:

Definition 21. *(Globally detectable fault) A fault $f \in F$ is globally detectable in system S if there is a minimal structurally overdetermined set MSO_i in the system, such that $f \in MSO_i$.*

Definition 22. *(Locally detectable fault) A fault $f \in F_i$ is locally detectable in subsystem S_i if there is a minimal structurally overdetermined set MSO_i in the subsystem that $f \in MSO_i$.*

Consider *Definition* 22 and equation (60). Fault f_1 is locally detectable because $f_1 \in MSO_{11}$ but f_2 is not locally detectable since there is no MSO in this subsystem that includes f_2. To detect f_2 locally, the diagnosis subsystem needs to include additional measurements. Global and local fault isolability are defined as:

Definition 23. *(Globally isolable fault) A fault $f_i \in F$ is globally isolable from fault $f_j \in F$ if there exists a minimal structurally overdetermined set MSO_i in the system S, such that $f_i \in MSO_i$ and $f_j \notin MSO_i$.*

Definition 24. *(Locally isolable fault) A fault $f_i \in F_i$ is locally isolable from fault $f_j \in F$ if there exists a minimal structurally overdetermined set MSO_i in subsystem S_i, such that $f_i \in MSO_i$ and $f_j \notin MSO_i$.*

Note that if a fault f_i is locally detectable in a subsystem S_i, it is globally detectable too, and if a fault f_i is locally isolable from a fault f_j, it is globally isolable from f_j as well. In the next section, we present the MSO-based distributed FDI approach. Section 4.3 presents the equation-based FDI method.

4.2 MSO-based Distributed Fault Detection and Isolation

4.2.1 Problem Formulation

Designing a set of distributed diagnosers that together have the same diagnosability as a centralized diagnoser is the focus of our work in this chapter. In the ideal case, each subsystem includes sufficient redundancies, such that its set of MSOs is sufficient to detect and isolate all of its faults, F_i uniquely and unambiguously. In that case, we can associate an independent diagnoser D_i with each subsystem $S_i; 1 \leq i \leq k$, and each diagnoser operates with no centralized control, and no exchange of information with other diagnosers. If the independence among diagnosers does not hold, then the subsystems need to communicate some of their measurements to other subsystems to detect and isolate the faults. To address this problem in an efficient way, we derive an integrated approach to select a set of MSOs for each subsystem that guarantee full diagnosability and minimum exchange of measurements among subsystems.

Given subsystems, $S_i; 1 \leq i \leq k$, with a set of local fault candidates, F_i, such that $\bigcup_{i=1}{}^{k} F_i = F$. We may need to augment each subsystem with additional measurements that are typically acquired from the (nearest) neighbors of the subsystem, such that all of the faults associated with the extended model of this subsystem are detectable and isolable. In the worst case, all of the measurements from another subsystem may have to be included to make the current subsystem diagnosable. When such a situation occurs, we say the two subsystems are merged and represented by a common diagnoser, therefore, the total number of independent distributed diagnosers may be less than k. Each MSO is sensitive to a set of faults and, therefore, can be used to detect and isolate them from the other faults in the system. For each subsystem S_i, our goal is to find a minimal set of MSOs that provide maximum detectability and isolability to that subsystem. A set of MSOs is minimal if there is no subset of MSOs that provides the same detectability and isolability. To achieve distributed fault diagnosis, we also want each subsystem to use the minimum number of

measurements from the other subsystems. In other words, we want to minimize communication or the amount of data (measurements) to be transmitted between the subsystems. More formally, the problem for designing a diagnoser for a particular subsystem S_i can be described as follows:

Consider $\mathscr{MSO} = \{MSO_1, MSO_2, \ldots, MSO_r\}$ as the set of possible MSOs for the subsystem S_i. We need to develop an algorithm to select a minimal subset of $\mathscr{MSO}$ that guarantees maximal structural detectability and isolability for faults F_i associated with the subsystem, and include a minimum number of measurements from the other subsystems in the system to assure the equivalence of local and global diagnosability , i.e.,

$$\begin{aligned} \forall S_i; \quad & 1 \leq i \leq k \\ Select \quad & MSO_{Si} \subset \mathscr{MSO} \\ s.t. \quad & \min_{M_o \subseteq M} |M_o| \\ & D_i(M_i \cup M_o) = D_i(M), \\ & I_i(M_i \cup M_o) = I_i(M), \end{aligned} \tag{64}$$

where M_o represents the set of measurement we need to communicate to the subsystem S_i along with the set of measurements, M_i associated with the subsystem S_i. M represents the set of all measurements in the system. For a given set of measurements, X, $D_i(X)$ represents the set of detectable faults in F_i, and $I_i(X)$ represents the set of isolable faults in F_i from the system faults, F. In the next subsection we formulate the problem as a BILP problem. Formulating the problem as a BILP, enables us to use a number of well-developed tools like branch and bound algorithms [109] and branch and cut algorithms [134] to solve the problem. However, much like integer linear programming, the general BILP solution is exponential

4.2.2 MSOs Selection for Distributed Fault Detection Using Global Model

In this subsection, we present our algorithm to select a minimal set of residuals for each subsystem of a system whose global model is available as a set of equations. In the next subsection, we modify this algorithm to make it applicable to much larger systems, where a compiled global model is not available. For the situation in which the global model is known, M in equation (64) is the set of all system measurements. Assume we have l measurements in the system: $M = \{m_1, m_2, ..., m_l\}$. The measurements imply redundancies in the system model that form the basis for generating MSOs. Let us assume we can generate r MSOs given M: $\mathscr{MSO} = \{MSO_1, MSO_2, \ldots, MSO_r\}$.

Our goal is to design an algorithm that selects $\mathscr{MSO}_i \subseteq \mathscr{MSO}$ in a way that we add a minimum number of measurements $M_o \subseteq M, M_i \cap M_o = \emptyset$, i.e., measurements from the system not belonging to subsystem i, to a subsystem to make all its faults globally diagnosable. Note that this is equivalent to the set covering problem and, therefore, any algorithm for finding the minimal measurements is exponential, in general. Roychoudhury et al [150] have adopted heuristic search methods for solving this problem. Their approach for designing subsystem diagnosers used the Temporal Causal Graph (TCG) approach. In this work, we formulate the search for minimal sensors as a BILP problem. The general formulation of BILP is presented in (36).

To formulate the problem (64) as a BILP problem we define a binary variable $x(k)$: $1 \leq k \leq l$, for measurement m_k in the system as follows:

$$x(k) = \begin{cases} 1 & if \quad m_k \in M_i \cup M_o \\ 0 & if \quad m_k \notin M_i \cup M_o, \end{cases} \tag{65}$$

where M_o is the answer to problem (64). We also define $x(k+l)$: $1 \leq k \leq r$, for MSO MSO_k

in the system as follows.

$$x(k+l) = \begin{cases} 1 & if \quad MSO_k \in \mathscr{MSO}_i \\ 0 & if \quad MSO_k \notin \mathscr{MSO}_i. \end{cases} \tag{66}$$

To minimize the number of measurements from the other subsystems, we develop the following cost function c as:

$$c(k) = \begin{cases} 0 & if \quad m_k \in M_i \\ 1 & if \quad m_k \in M \backslash M_i \\ 0 & if \quad l < k \leq l+r, \end{cases} \tag{67}$$

where l is the number of system measurements and r is the number of MSOs in the system. Using the fault diagnosis toolbox [68], 165 MSOs are generated for the running example, the four tank system. Since there are 8 measurements in the system c is a vector with 173 elements for this example.

Consider subsystem S_i with local faults F_i and the set of system faults, F. Each local fault $f_j \in F_i$ has to be locally detectable. Given definition 22, we can guarantee local detectability of all the faults $f_j \in F_i$ with the following constraints in the optimization problem (36).

$$A(j,k) = \begin{cases} 0 & if \quad k < l \\ -1 & if \quad f_j \in MSO_{k-l} \\ 0 & otherwise. \end{cases} \tag{68}$$

Note that l is the number of measurements in the system. By considering $b(j) = -1$ for $1 \leq j \leq g$, where g is the number of faults in F_i, we make sure that we have selected at least one MSO to detect each fault. To address isolability requirement we follow the same

procedure. To isolate $f_j \in F_i$ from any other fault in system, i.e., $f_h \in F$ we need to have:

$$A(j+g,k) = \begin{cases} 0 & k < l \\ -1 & f_j \in MSO_{k-l}, f_h \notin MSO_{k-l} \\ 0 & otherwise. \end{cases} \tag{69}$$

Setting $b(j) = -1$ for $g < j \leq g*h$, where h is the number of faults in the system, $h = |F|$, we make sure that there is at least one MSO to isolate each of the subsystem faults from the other faults in the system.

In addition to the constraints that guarantee maximum detectability and isolability for the distributed diagnosis system, we need a set of constraints that capture the relationship between the measurements and MSOs in the distributed diagnosis system. Using a MSO is equivalent to using the measurements that are included in the MSO, and we need to include this in the optimization problem. For example, consider MSO_{11}, it has three measurements $M_{11} = \{u_1, y_1, y_2\}$. Using MSO_{11} in a local diagnosis subsystem means we need to communicate these measurement streams to that subsystem to achieve global diagnosability for the faults that belong to that subsystem. The following equation represents this constraint.

$$-x(1) - x(2) - x(3) + |M_1|x(7) \leq 0, \tag{70}$$

where $|M_1| = 3$ is the cardinality number of M_1 and $x(1)$, $x(2)$, $x(3)$ and $x(7)$ are binary variables that are 1 if we use u_1, y_1, y_2 and MSO_{11} in the diagnosis system and are zero otherwise. This constraint implies that if we use MSO_{11}: $x(7) = 1$, its associated measurements are used by the subsystem too: $x(1) = x(2) = x(3) = 1$.

Equation (71) represents these set of constraints in A matrix.

$$A(j+g*h,k) = \begin{cases} -1 & if \quad m_k \in MSO_j \\ |M_j| & if \quad k = j + |M| \\ 0 & otherwise, \end{cases} \tag{71}$$

where $|M_j|$ is the cardinality number of set of measurements in MSO_j and $|M|$ is the cardinality number of set of all the measurements in the system. Setting $b(j) = 0$ for $g*h < j \leq g*h+r$, where r is the number of MSOs in the system. The optimization problem takes into account the relationship between measurements and MSOs. For the running example we generated 165 MSOs, there are also 3 measurements in the subsystem 1, and 8 measurements for the entire system. Similarly, subsystem 1 has two faults of interest, and the goal is to be able to isolate them from any of the 6 faults in the complete system. Therefore, to solve the optimization problem (36) for subsystem 1, matrix A has 177 rows (equal to the number of constraints: 2 constraints to guarantee the local detectability of f_1 and f_2, 10 constraints to guarantee the local isolability of f_1 and f_2 from the other faults, and 165 constraints to capture the relationship between the MSOs and the measurements) and 173 columns (equal to the number of binary variables: 8 for the measurements and 165 for the MSOs) and b is a vector with 177 elements (equal to the number of constraints).

Table 5 shows the set of measurements that we need to add for each of the subsystem diagnosers to achieve maximum possible detectability and isolability using our proposed algorithm. To find the optimum measurements, we solved the optimization problem (36) for each subsystem. Considering the expanded measurement set, the schematic of the four

Table 5: Set of augmented measurements to each subsystem model

Subsystem	Set of augmented measurements
S_1	y_3
S_2	u_2, y_2, y_6
S_3	y_4, y_6
S_4	y_5

tank system with the four distributed diagnosers is shown in Fig. 18. The figure shows the complete set of measurements required by the four subsystem diagnosers to achieve global detectability and isolability for the set of faults they contain. For example subsystem 1

includes three measurements $M_1 = \{u_1, y_1, y_2\}$, and to achieve global diagnosability for its faults, y_3 must be communicated to its diagnoser from subsystem 2. Subsystem 2 is the only subsystem that shares a variable with a second order connected subsystem, all the other subsystems only need to communicate with their first order connected subsystems. Note that communicated measurements typically will incur additional cost and may lower reliability of the system diagnoser. But keeping them to a minimum (see results in Table 5) reduces that cost and uncertainty, while maintaining global diagnosability.

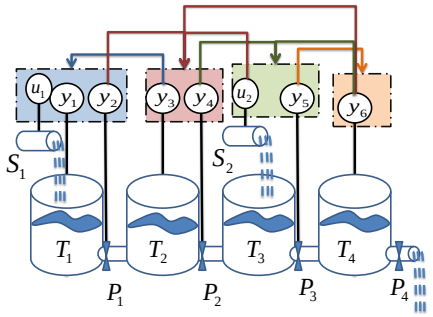

Figure 18: Distributed diagnosis subsystems.

A common way to validate a distributed fault detection and isolation approach is to compare the result with the maximum global detectability and isolability. Adopting the exoneration assumption, Table 21 shows the detectability and isolability performance of the centralized approach. An X in the table shows that the fault in the row and the fault in the column are not isolable from each other. An X in the first column (NF) means the fault in the corresponding row is not isolable from NF (No Fault) or simply it is not detectable. Table 21 shows that with a centralized approach we can detect and isolate all the faults. However, Table 7 shows that using the original subsystems for distributed diagnosis does not provide the same results as the centralized global diagnoser. In fact, only f_1 can be detected and isolated from the other faults. Using the augmented subsystems in Table 5 (Figure 18) we achieve the same performance as the global diagnoser as shown in Table 8. This demonstrates that the distributed approach can achieve the same performance with the

Table 6: Fault isolability table for running example using centralized approach

	NF	f_1	f_2	f_3	f_4	f_5	f_6
f_1		X					
f_2			X				
f_3				X			
f_4					X		
f_5						X	
f_6							X

Table 7: Fault isolability table for running example using distributed approach for the original subsystems

	NF	f_1	f_2	f_3	f_4	f_5	f_6
f_1		X					
f_2	X		X	X	X	X	X
f_3	X		X	X	X	X	X
f_4	X		X	X	X	X	X
f_5	X		X	X	X	X	X
f_6	X		X	X	X	X	X

centralized approach for fault detection and isolation in the running example.

In general, the worst case scenario for a system with strongly connected subsystems (i.e., all subsystems are connected to each other) will typically require a large number of measurements from other subsystems to be communicated to each subsystem diagnoser. In those situations, subsystem diagnosers just get rid of the single point of failure, but each subsystem diagnoser may require a large number of measurements to be communicated to it from all of the other subsystems. In our example, the four tank system model included 165 MSOs, which means for each subsystem there was 2^{165} different MSO candidate sets. This creates a very large search space (in general the search space is exponential in the number of MSOs, and generating all MSOs is in itself an exponential problem. This justifies the formulation of the problem as a BILP problem that provides efficient tools, like the *bintprog* function in Matlab™(see earlier footnote), to solve it. However, given the exponential nature of the solution, this method will not scale up for larger systems, even

Table 8: Fault isolability table for running example using distributed approach for the augmented subsystems

	NF	f_1	f_2	f_3	f_4	f_5	f_6
f_1		X					
f_2			X				
f_3				X			
f_4					X		
f_5						X	
f_6							X

if the subsystem diagnoser design is performed off-line. In addition to the computational complexity, the availability of global models for large, complex systems is unlikely because of the issues discussed in the beginning of this chapter. To overcome this problem, we sacrifice minimality of the solution to some extent, and propose an incremental algorithm for designing the subsystem diagnosers.

4.2.3 MSOs Selection for Distributed Fault Detection Using Neighboring Subsystems

The proposed approach in the previous section used the global model of the system to generate the residuals, and then derived the subsystem diagnosers using the BILP algorithm run on the global MSO set. In this section, we achieve global diagnosability of a subsystem diagnoser by incrementally adding a minimum number of measurements from the neighbors of this subsystem till the global diagnosability property is established. The algorithm starts with the set of equations for the subsystem whose diagnoser is being designed, and if global diagnosability is not achieved using this model, it expands to include equation sets that correspond to the models of its immediate neighbors. If global diagnosability is achieved, the algorithm terminates, otherwise the algorithm expands to use the next higher order of neighbors and repeats the search for minimal MSOs to achieve complete diagnosability. The process of including successively higher order neighbors is shown in Fig. 19. In the worst case, this process continues, till the complete set of system

equations are required to generate all possible MSOs, and establish global diagnosability for the subsystem. Therefore, it is guaranteed that the method has the same diagnosability performance as the best centralized diagnoser for the same set of measurements.

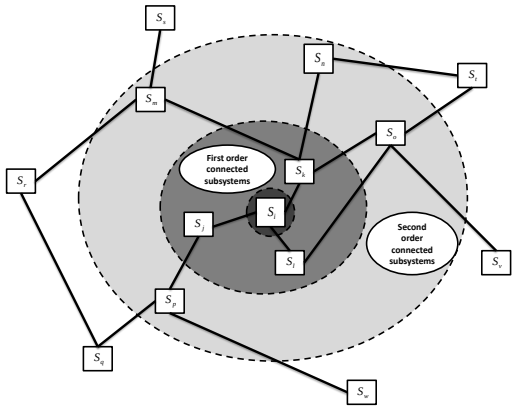

Figure 19: Expanding the search environment to the higher order connected subsystems.

Algorithm 2 describes the algorithm for our proposed method. Consider the running

Algorithm 2 Incremental Algorithm

for each $S_i \in S$ **do**
  $SS = S_i$
  $j = 0$
  while $D_i(SS) \neq D_i(S)$ or $I_i(SS) \neq I_i(S)$ **do**
    $j = j + 1$
    $SS = SS \cup (j\text{th order connected subsystems of } S_i)$
    Generate all the MSOs for SS
    Use equation (67) to compute cost function for SS
    Use equations (68), (69), and (71) to generate A matrix for SS
    Generate vector b for SS
    Use $bintprog(c,A,b)$ to solve the problem and compute $D_i(SS)$ and $I_i(SS)$
  end while
end for

example. To design the diagnosis system for the first subsystem, we start with its set of

equations and we can only generate one MSO which is not enough to detect subsystem faults and isolate them from the system faults. We then augment the subsystem model with the model from its nearest neighbor subsystem 2, and generate the set of MSOs for the augmented model. The total number of MSOs for the augmented subsystem (Subsystem 1 + subsystem 2) is 11 which leads to 2^{11} MSO set candidates which is much smaller than 2^{165} candidates. Solving the optimization problem presented in this section gives the same result with the global method for this subsystem, but the computation time is reduced significantly. Using the same approach for every subsystem, the set of measurements that we need to transfer to each subsystem of the running example are presented in Table 9.

Table 9: Set of augmented measurements to each subsystem model

Subsystem	Set of augmented measurements
S_1	y_3
S_2	u_2, y_2, y_5
S_3	y_4, y_6
S_4	y_5

Fig.20 shows that for the four tank case study, all the subsystems share measurements with their first order connected subsystems. This provides a practical advantage to this algorithm because usually the subsystems with shared variables are physically closer to each other (corresponding to our definition of nearest neighbors) and, therefore, we do not need to transfer data over long distances, which, as discussed earlier, can be costly and error-prone. Table 10 shows that this distributed diagnosis system provides the same diagnosability performance as the centralized diagnosis method. The proposed algorithm provides the maximum possible detectability and isolability that can be achieved. The advantage of this algorithm is that not only we do not need a global model for detecting and isolating the faults, but also we do not use the global model in the design process of the supervisory system. This makes the approach suitable for large, complex systems, such as

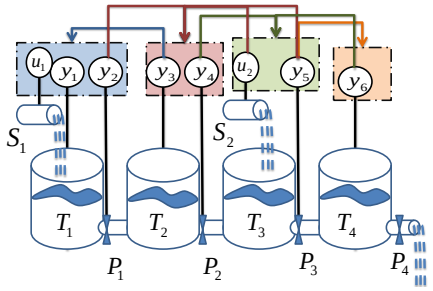

Figure 20: Distributed diagnosis subsystems using incremental algorithm.

Table 10: Fault isolability table for running example using the incremental algorithm

	NF	f_1	f_2	f_3	f_4	f_5	f_6
f_1		X					
f_2			X				
f_3				X			
f_4					X		
f_5						X	
f_6							X

aircraft and power plants where the global systems models are likely to be unavailable or unknown.

4.3 Equation-based Distributed Fault Detection and Isolation

In the algorithm developed in the last section, the total number of MSOs is an exponential function of the system measurements. This makes the problem computationally very expensive. To avoid this computational complexity, we propose a distributed diagnosis method that works directly with the system equations. To recap from earlier work [59], the Dulmage-Mendelsohn (DM) decomposition decomposes a system model into three parts: (1) under determined, (2) exactly determined and (3) over determined. The over determined part introduces redundancy in the system and can be used for fault detection and isolation. Fig. 21 shows the DM decomposition for the first subsystem of the running example. This subsystem model has a just determined part (${S_1}^0$) and an over determined part (${S_1}^+$).

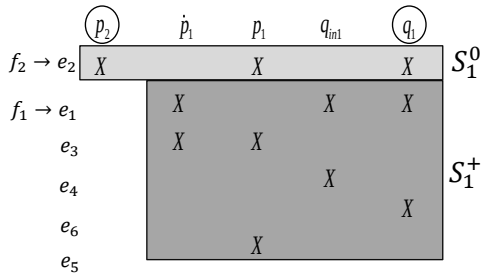

Figure 21: DM decomposition of the first subsystem model.

Fig. 21 represents the set of equations in the just determined part and the over determined part of S_1 and the set of unknown variables in each equation. The shared variables are shown as encircled variables in the figure. In this section, we assume every fault parameter is included in exactly one equation. This is not a restricting assumption because if we have more than a fault in an equation we can consider the other faults as new variables and then add new equations for each of these new variables making the variable equal to the fault. Therefore, after making the appropriate changes, for each fault f there is one and only one equation e_f associated to this fault. Given that, the local detectability can be defined as:

Definition 25. *(Locally detectable) A fault $f \in F_i$ is locally detectable in subsystem S_i if $e_f \in {S_i}^+$, where ${S_i}^+$ is the over-determined part of subsystem S_i.*

Note that *Definition* 25 is equivalent to *Definition* 22. Consider *Definition* 25 and Fig. 21. Fault f_1 is locally detectable because $e_1 \in {S_1}^+$ but f_2 is not locally detectable since $e_2 \notin {S_1}^+$. To expand the overdetermined part and make f_2 detectable, the diagnosis subsystem needs to have at least one additional equation. We define such a subsystem as

Definition 26. *(Augmented subsystem) Given subsystem S_i and a set of equations, E_k, the augmented subsystem model $S_{iE_k} = (S_i|E_k)$ is (V_{iE_k}, M_{iE_k}, E_{iE_k}, F_{iE_k}), where V_{iE_k} is the union of V_i and the unknown variables that appear in E_k, M_{iE_k} is the union of M_i and the known*

variables that appear in E_k, E_{iE_k} is the union of E_i and E_k and F_{iE_k} is the union of F_i and the possible faults associated with E_k.

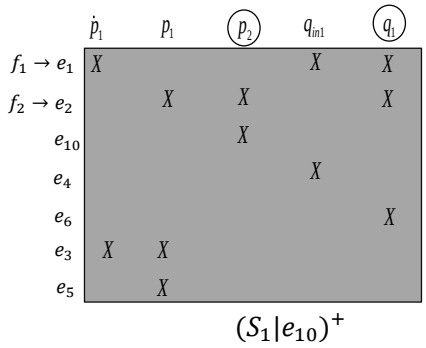

Figure 22: DM decomposition of $S_{1e10} = (S_1|e_{10})$.

Consider the running example. $S_{1e10} = (S_1|e_{10}) = (V_{1e10}, M_{1e10}, E_{1e10}, F_{1e10})$, where $V_{1e10} = \{\dot{p}_1, p_1, p_2, q_{in1}, q_1\}$, $E_{1e10} = \{e_1, e_2, e_3, e_4, e_5, e_6, e_{10}\}$, $M_{1e10} = \{u_1, y_1, y_2, y_3\}$ and $F_{1e10} = \{f_1, f_2\}$. Note that e_{10} did not add any new unknown variables or faults to the subsystem model. Fig. 22 represents the DM decomposition of the augmented subsystem, S_{1e10}. This figure shows that $e_2 \in {S_{1e10}}^+$, and, therefore, f_2 is locally detectable for the augmented subsystem S_{1e10}. We re-define locally isolable faults using the equation-based approach as:

Definition 27. *(Locally isolable) A fault $f_i \in F_i$ is locally isolable from fault $f_j \in F$ if $e_{f_i} \in (S_i \backslash e_{f_j})^+$, where $(S_i \backslash e_{f_j})^+$ is the over-determined part of subsystem S_i without equation e_{f_j}.*

Note that *Definition* 27 is equivalent to *Definition* 24. Fig. 23 shows DM decomposition of the $S_{1e10} \backslash e_1$. e_2 is in the overdetermined part of the augmented subsystem model, therefore, f_2 is locally isolable from f_1 in the augmented subsystem.

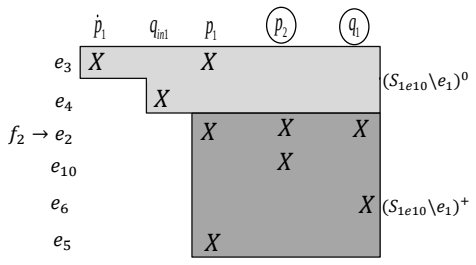

Figure 23: DM decomposition of $S_{1e10}\backslash e_1$.

4.3.1 Problem formulation

We formulate the problem and equation-based solution approach for designing a distributed diagnoser for a system, S made up of a number of subsystems, $S_1, S_2, \cdots S_n$. As we mentioned in the previous section, there is no overlap of components among the subsystems, however, the subsystems may share variables at their interface. In the ideal case, each subsystem includes a sufficient number of measured variables, such that the ensuing redundancy is sufficient to detect and isolate all of its faults F_i locally. In that case, we can associate an independent diagnoser D_i with each subsystem $S_i; 1 \leq i \leq n$, and each diagnoser operates with no centralized control, and no exchange of information with other diagnosers. If the independence among diagnosers does not hold, then we have to consider the following additional cases:

1. $f_k \in F_i$ is not locally detectable.

2. $f_l \in F_i$, and $f_m \in F_i$ are not locally isolable from each other.

3. $f_n \in F_i$ is not locally isolable from $f_o \in F_j$ and $f_o \notin F_i$.

Designing distributed diagnosers that account for these three scenarios is the focus of our work in this section. After addressing each of these situations, we derive an integrated approach to distributed FDI, and derive algorithms that apply to complex, dynamic systems made up of a number of subsystems.

Given subsystems, $S_i; 1 \leq i \leq n$, associated with each subsystem there is a set of equations E_i and a set of local fault candidates, F_i, such that $\bigcup_{i=1}{}^{n} F_i = F$. We may need to augment each subsystem with additional equations that are typically acquired from the neighbors of the subsystem, such that all of the faults associated with the extended model of this subsystem are detectable and isolable. In the worst case, all of the equations from a neighboring subsystem may have to be included to make the current subsystem diagnosable. For each subsystem S_i, our goal is to find minimal sets of equations from the neighboring subsystems that provide complete detectability and isolability to that subsystem. A set of equations is minimal if there is no subset of equations that provides the same detectability and isolability. More formally, the problem for designing a diagnoser for a particular subsystem S_i can be described as follows.

Consider $\mathcal{N}_{Si} = \{S_1, S_2, \ldots, S_l\}$ as the set of neighboring subsystems to subsystem S_i. To address the three situations mentioned above, we need to develop an algorithm to find a minimal equation set E_o in $\mathcal{N}_{Si}$ that guarantees maximal structural detectability and isolability for subsystems faults F_i, i.e.,

$$\begin{aligned} \min_{E_o \subseteq E_l} \quad & |E_o| \\ & D(S_i|E_o) = D(S_i|E_l), \\ & I(S_i|E_o) = I(S_i|E_l), \end{aligned} \tag{72}$$

where E_l represents the set of all the equations in $\mathcal{N}_{Si}$, D represent the set of detectable faults in F_i, and I represents the set of isolable faults in F_i from the system faults F. Consider the first subsystem of the running example S_1, e_{10} makes f_1 and f_2 detectable and isolable from all the other faults in the system. Therefore, $A_1 = \{e_{10}\}$ is a minimal solution to the problem. In this section, we present a method to make all the faults in a subsystem

locally detectable (situation (1) above). We also discuss the solution to the fault isolability problem (situation (2) above), and prove that if we address the first situation, the third situation is automatically taken care of.

4.3.2 Maximum Detectability

Consider subsystem 1 whose equations are listed in (60). The DM decomposition of this subsystem is shown in Fig. 21. f_2 is in the just determined part of the subsystem, therefore, the fault is not locally detectable. However, p_2 is a shared variable with subsystem 2. Therefore, we could find an equation from subsystem 2, e_{10}, to make f_2 locally detectable in the augmented subsystem, S_{1e10}, (see Fig. 22). Adding measurement equation e_{10} made p_2 known and, therefore, the subsystem overdetermined. Note that we cannot make a variable that only appears in subsystem 1 ($\dot{p}_1$ for example) known by adding equations from other subsystems. Therefore, our ability to increase fault diagnosability is limited to the shared variables in the subsystem. More formally, we can prove the following theorem.

Theorem 1. *Consider local subsystem model* $S_i = \{V_i, M_i, E_i, F_i\}$ *and* $V_{ishared} \subset V_i$ *the set of shared variables in the subsystem. If a fault* $f \in F_i$ *is not locally detectable in a new subsystem* $S_j = \{V_i - V_{ishared}, M_i \cup V_{ishared}, E_i, F_i\}$ *where all the shared variables are known,* f *is not globally detectable.*

Proof. If e_f stays in the just determined part or under determined part of the subsystem when all the shared variables have became known, there is no addition equation in the system that can make any of the variables in e_f known and, therefore, moves the equation to the over determined part of the structural decomposition. □

Therefore, the maximum detectability performance that we can achieve in each subsystem cannot be more than the detectability performance when all the shared variables are known. Using Theorem 1, we develop Algorithm 3 and Algorithm 4 to find an upper

bound for the number of detectable faults and isolable fault pairs in each subsystem without requiring any information from the neighboring subsystems respectively.

Algorithm 3 Detectable-Faults

input: $V_{ishared}$
input: $S_i = \{V_i, M_i, E_i, F_i\}$
Let DF be $\{\}$
Let S_{DF} be $\{V_i - V_{ishared}, M_i \cup V_{ishared}, E_i, F_i\}$
for each $f \in F_i$ **do**
  if $f \in (S_{DF})^+$ **then**
    $DF = DF \cup \{f\}$
  end if
end for
return DF

Algorithm 4 Isolable-Faults

input: $V_{ishared}$
input: $S_i = \{V_i, M_i, E_i, F_i\}$
Let IF be $\{\}$
Let S_{IF} be $\{V_i - V_{ishared}, M_i \cup V_{ishared}, E_i, F_i\}$
for each $f_j \in F_i$ **do**
  for each $f_k \in F_i$ **do**
    if $f_i \in (S_{IF} \setminus ef_k)^+$ **then**
      $IF = IF \cup \{f_j, f_k\}$
    end if
  end for
end for
return DF

In this work, we are interested in finding a minimal set of shared variables to achieve the maximum detectability performance for each subsystem. Adopting the following strategy, we can find a minimal set of shared variables that guarantees maximum detectability.

- We assume all the shared variables are known. If a fault is not locally detectable when

all the shared variables are known, we remove that fault from the list of detectable faults (see Algorithm 3).

- We move each shared variable form the list of known variables to the unknown variables, and examine the list of detectable faults. If removing the shared variable from the known variables decreases the number of faults in the list of detectable faults, we reveres our action and add the variable to the list of minimal required shared variables. Otherwise, we do not need to know this shared variable.

Algorithm 5 presents our method to find a minimal set of required share variables. We initialize the algorithm with the subsystem model and the set of shared variables (for subsystem 1, p_2 and q_1 are unknown shared variable) and it provides a minimal subset of shared variables in the subsystem that makes all the faults detectable. For subsystem 1, $V_{1m} = \{p_2\}$ is a possible answer.

Algorithm 5 Minimal-Shared-Variables

```
1: input: S_i = {V_i, M_i, E_i, F_i}
2: Let V_ishared be the set of shared variables in S_i
3: DF = Detectable-Faults(V_ishared, S_i)
4: Let V_im be V_ishared
5: for each v_is ∈ V_im do
6:     Let V_im be V_im/v_is
7:     if Detectable-Faults(V_im, S_i) not equal DF then
8:         V_im = V_im ∪ {v_is}
9:     end if
10: end for
11: return V_im
```

In this subsection, we developed an algorithm to find a minimal set of unknown shared variables, that if transfered from neighboring subsystems, can provide maximum detectability performance. Note that all the shared unknown variables are not necessarily measured in the neighboring subsystems. However, in some cases it is possible to transfer a set of

equations from the neighboring subsystems that can be used with the equations in the subsystem to compute the unknown variables. In the next subsection, we present our proposed approach to find a minimal set of equations from the neighboring subsystems in order to compute the minimal set of required shared unknown variables and provide maximum possible fault detectability.

4.3.3 Equation-based Fault Detection Approach

Given a minimal set of required shared variables, we present our proposed approach to find a minimal set of equations from the neighboring subsystems in order to achieve maximum possible fault detectability. We illustrate the procedure by solving this problem for subsystem 2 of the case study, and then generalize this approach by developing a general algorithm to solve this problem. Consider subsystem 2 presented in equation (61). The corresponding structural decomposition of this subsystem is shown in Fig. 24.

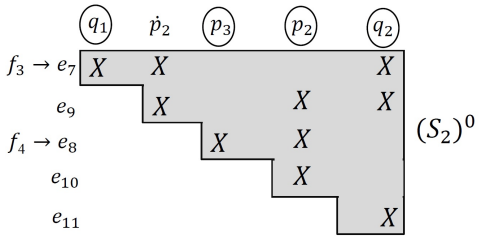

Figure 24: DM decomposition of S_2.

This subsystem is just determined, therefore, none of the faults are locally detectable. However, q_1 and p_2 are shared variables with subsystem 1, and q_3 and p_3 are shared variables with subsystem 3. Algorithm 5 finds $V_{1m} = \{q_1, p_3\}$ as a minimal set of shared unknown variables, that if transfered from neighboring subsystems, can provide maximum detectability performance. Therefore, to make f_3 and f_4 locally detectable, we have to find equations from the neighboring subsystems to make q_1 and p_3 known.

To find a minimal set of just determined equations that includes q_1,we start with all

equations in S_1 that have q_1. These equations are e_1, e_2, and e_6 as it is shown in Fig. 25. Then for the additional variables in each equation that is not already in S_2^0 we need to add

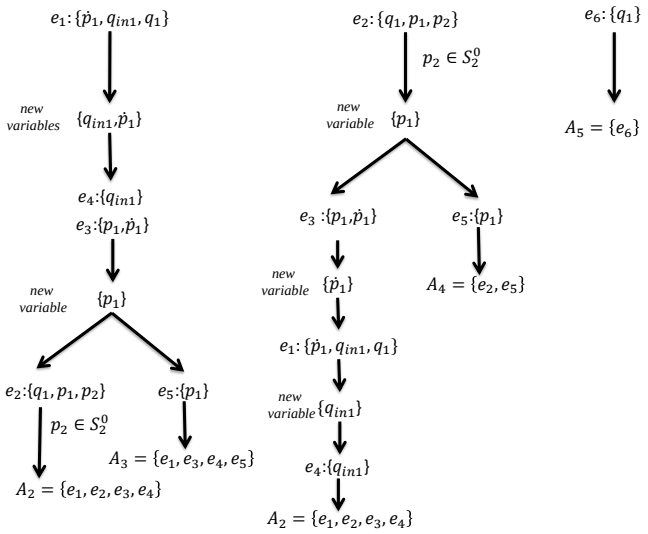

Figure 25: Finding the minimal sets of equations in S_1 to compute q_1.

other equations. For e_1 we need to add two new equations one with q_{in1} and the other one with $\dot{p}_1$. Finally, we need to add a new equation with p_1 and since $p_2 \in {S_2}^0$ we do not have to consider it.

To find the other minimal sets we keep adding the relative equations to the other sets using the same approach described above. As it is shown in Fig. 25, by adding equations to the system we eventually achieve four sets of minimal constraints: $A_2 = \{e_1, e_2, e_3, e_4\}$, $A_3 = \{e_1, e_3, e_4, e_5\}$, $A_4 = \{e_2, e_5\}$, and $A_5 = \{e_6\}$. Fig. 25 represents a matching algorithm. In fact, we applied a matching algorithm to find a minimal set of equations from neighboring subsystems for computing each required shared variable. The general form of our algorithm is presented as Algorithm 6. If we initialize the algorithm with the set of unknown variables (in Fig. 25, q_1 is the unknown variable) it provides a set of complete matching of variables and equations in the neighboring subsystems that includes the unknown variables. Fig. 26 shows that augmenting A_2 with S_2 makes f_3 detectable. To make

Algorithm 6 Count-Matchings

input: current matching $\mathscr{M}$
input: sets of determined variables $\mathscr{D}$ and undetermined variables $\mathscr{U}$, set of equations E
if $U = \emptyset$ **then**
  return $\mathscr{M}$ as a feasible (minimal) matching.
end if
for each $x \in \mathscr{U}$ **do**
  for each $e \in E$ which can determine x **do**
    Let $\mathscr{M}'$ be $\mathscr{M} \cup \{e \to x\}$
    Let $\mathscr{D}'$ be $\mathscr{D} \cup \{x\}$.
    Let $\mathscr{U}'$ be $\mathscr{U} \setminus \{x\}$.
    Let E' be $E \setminus \{e\}$.
    Add all the undetermined variables of y to $\mathscr{U}'$.
    COUNT-MATCHINGS$(\mathscr{M}', \mathscr{D}', \mathscr{U}', E')$
  end for
end for

f_4 locally detectable as well, we need to use Algorithm 6 to find a minimal set of equations in the neighboring subsystems that includes p_3 and augment $S_2|A_2$ with those equations.

Subsystem 2 is just determined but a subsystem can have an under-determined part as well. For example, consider subsystem S_3 in equation (62). The DM decomposition of this subsystem model is shown in Fig. 27. f_5 is in the underdetermined part of the structure. q_{in2} and q_3 are in the just determined part of the system and we can compute them using e_{15} and e_{16}, respectively. However, to compute the other four variables in the subsystem, p_3, q_2, $\dot{p}_3$, and p_4, we only have three constraints, which makes complete matching between constraint and variables impossible. To make this part of the subsystem just determined, we need to augment a set of equations from the neighboring subsystems.

Unlike previous work [103] where we had to develop a new algorithm for subsystems with under-determined parts, Algorithm 5 automatically takes care of subsystems with under-determined sections. Using algorithm 5 gives us $V_{m3} = \{q_2, q_3\}$ as a minimal set of required shared variables to make f_5 detectable. Having the set of required shared variables,

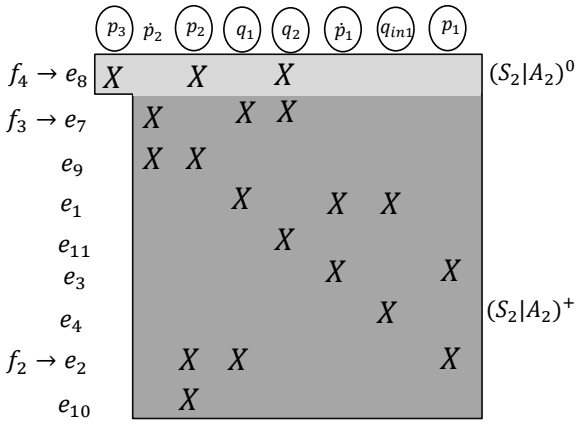

Figure 26: DM decomposition of $(S_2|A_2)$.

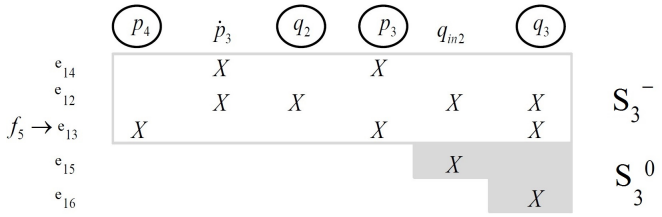

Figure 27: DM decomposition of S_3.

Algorithm 6 gives us $A_6 = \{e_{11}, e_{17}, e_{18}, e_{19}\}$ as a minimal sets of equations from neighboring subsystems that we can augment to S_3 to make the f_5 locally detectable. Fig. 28 shows DM decomposition of $(S_3|A_6)$.

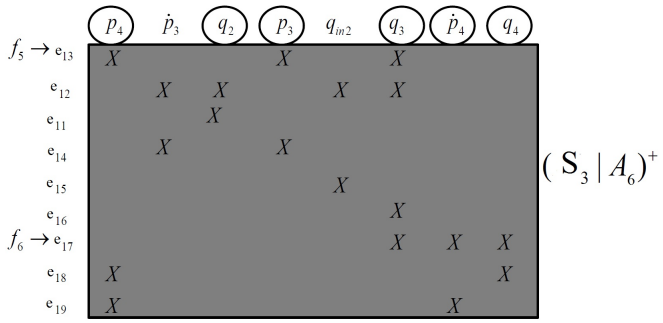

Figure 28: DM decomposition of $(S_3|A_6)$.

In some cases, it is possible that an augmented minimal set, A_i, also adds a set of faults

F_{A_i} to the subsystem model S_i. These faults can be sensor faults or faults in other equations. The following theorem states that these faults are locally detectable in subsystem model S_i.

Theorem 2. *Consider local subsystem model $S_i = \{V_i, M_i, E_i, F_i\}$ and E_k a set of minimal equations that makes set of faults F_i detectable in the augmented subsystem $(M_i|C_{augments}) = \{V_j, C_j, F_j\}$, then the set of faults F_j in the augmented subsystem $(S_i|E_k)$ are locally detectable.*

Proof. The proof of this theorem is straight forward, since the minimal set makes a part of the system that includes the fault overdetermined, the set itself should be in the overdetermined part as well. This means the associated faults in the set are detectable. □

For example, f_6 is locally detectable in $(S_3|A_6)$. Therefore, as long as we are focused on fault detection the augmented faults do not cause any problem. The fault detection algorithm is summarized as Algorithm 7 below.

Algorithm 7 Detectability

1: **input:** subsystem S_i
2: **input:** subsystem model neighbors $\mathcal{N}_{Si}$
3: $\mathscr{M} = \{\}$
4: $V_d =$ set of determined variables in S_i
5: **if** $\forall f \in F_i$ therefore $e_f \in {S_i}^+$ **then**
6: **return**
7: **end if**
8: $U =$ Minimal-Shared-Variables(S_i)
9: $D = V_d \setminus U$
10: $E_d =$ Count-Matchings ($\mathscr{M}$, D, U, $\mathcal{N}_{Si}$)

4.3.4 Equation-based Fault Isolation Approach

In this subsection we assume the set of minimal equations to make all the faults locally detectable have been derived based on the method presented in the previous subsections.

It is clear that the locally detectable faults in each subsystem are locally isolable from the faults in the other subsystems.

Theorem 3. *Consider local subsystem $S_i = \{V_i, M_i, E_i, F_i\}$ if $f_i \in F_i$ is locally detectable in S_i, then f_i is locally isolable form f_j if $f_j \notin F_i$.*

Proof. Since f_i is detectable we have $e_{f_i} \in {S_i}^+$ and since $f_j \notin F_i$ we can say $e_{f_j} \notin {S_i}^+$. Therefore, ${S_i}^+ = (S_i \backslash e_{f_j})^+$ and $e_{f_i} \in (S_i \backslash e_{f_j})^+$ □

For example, in $(S_3|A_6)$, f_5 is isolable from f_1, f_2, f_3, and f_4 because they are not in the augmented subsystem and f_5 is detectable in this augmented subsystem. Considering Theorem 3, it is straight forward to address the isolability problem. For each fault $f_i \in F_i$, we remove the associated equation e_{f_i} from E_i and all the neighboring subsystems. Then we use Algorithm 7 to make all the remaining faults in F_i detectable. For example, consider $(S_3|A_6)$. To make f_5 isolable from f_6, we remove e_{17} from $(S_3|A_6)$ and S_4. DM decomposition of $(S_3|A_6)\backslash e_{17}$ is shown in Fig 29. Applying algorithm 7 to $S_4 \backslash e_7$ gives us

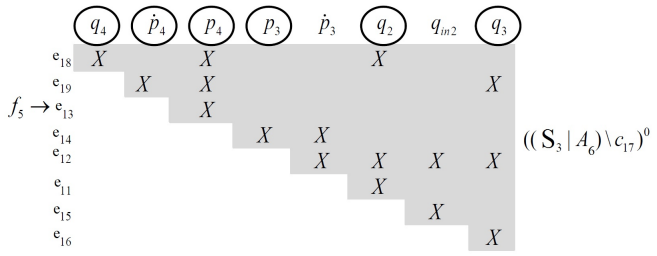

Figure 29: DM decomposition of $(S_3|A_6)\backslash e_{17}$.

$\{e_{20}\}$ as a minimal set that can make f_5 detectable. Therefore, we can say the augmented subsystem $(S_3|A_6 \cup \{e_{20}\})$ will detect f_5 and isolate it from all the other faults in the global system S. Algorithm 8 summarizes the method discussed above.

Much like Algorithm 2, our proposed approach considers the first order neighboring

Algorithm 8 Diagnosability

input: subsystem model S_i
input: subsystem model neighbors $\mathcal{N}_{Si}$
$E_d =$ Detectability(S_i, $\mathcal{N}_{Si}$)
$S_i = (S_i|E_d)$
for each $f \in S_i$ **do**
    $\bar{S}_i = S_i \setminus (f$ and $e_f)$
    $E_d =$ Detectability($\bar{S}_i$, $\mathcal{N}_{Si}$)
    $S_i = (S_i|E_d)$
end for

subsystems of subsystem S_i and augment minimal constraints from them to maximize diagnosability. If the set of first order neighboring subsystems does not have required redundancies to achieve maximum diagnosability we have to expand the search process to the next higher order of neighborings subsystems. The process of including successively higher order neighbors is shown in Fig. 19. The expansion process will stop when the distributed approach achieves maximum diagnosability. Therefore, it is guaranteed that the method has the same diagnosability performance as the best centralized diagnoser for the same set of measurements. In the case that no independent subsystem diagnosers can be derived using our distributed approach, the solution gradually expands to include all subsystems and eventually derives the centralized diagnoser. Algorithm 9 summarizes this approach.

Algorithm 9 DistributedDiagnosis

input: subsystem S_i
input: subsystem model neighbors $\mathcal{N}_{Si}$
Let $V_{ishared}$ be the set of shared variables in S_i
$DF =$ Detectable-Faults($V_{ishared}$, S_i)
$IF =$ Isolable-Faults($V_{ishared}$, S_i)
$E_o =$ Diagnosability(S_i, $\mathcal{N}_{Si}$)
if $D(S_i|E_o) = DF$ and $I(S_i|E_o) = IF$ **then**
    return
end if
$\mathcal{N}_{Si} = \mathcal{N}_{Si} \cup$ (neighboring subsystems of $\mathcal{N}_{Si}$)
DistributedDiagnosis(S_i, $\mathcal{N}_{Si}$)

Table 11 shows the set of equations that each subsystem in the running example needs from its neighbors to achieve maximum possible detectability and isolability using the equation-based approach. Like previous section, to validate our distributed fault detection

Table 11: Set of augmented constraints to each subsystem model

Subsystem	Set of augmented equations
S_1	e_{10}
S_2	$e_6, e_{12}, e_{14}, e_{15}, e_{16}$
S_3	$e_{11}, e_{17}, e_{18}, e_{19}$
S_4	e_{16}

and isolation approach, we compare the result with the maximum global detectability and isolability. Table 21 shows that with a global diagnostic method we can detect and isolate all the faults. Using the augmented subsystems in Table 11 we will have the same performance as the global model as shown in Table 12. This demonstrates that the distributed

Table 12: Fault isolability table for running example using equation-based distributed approach for the augmented subsystems

	NF	f_1	f_2	f_3	f_4	f_5	f_6
f_1		X					
f_2			X				
f_3				X			
f_4					X		
f_5						X	
f_6							X

approach has the same performance with the centralized approach for fault detection and isolation in the running example.

4.4 Time Complexity

Algorithm 6 is the only exponential step in Algorithm 7 and, therefore, the time complexity of Algorithm 7 is mostly governed by this algorithm. In the worst case scenario, Algorithm 6 has to consider all the permutations of the equations to find a solution for the required unknown variable. Therefore, theoretically this algorithm has $O(|\mathscr{U}| \times |E_N|!)$ time complexity, where $|\mathscr{U}|$ is the number of required unknown variables in the subsystem and $|E_N|$ is the number of equations in the neighboring subsystems. Algorithm 8 calls Algorithm 7 for every fault in the subsystem. Therefore, Algorithm 8 has $O(|F_i| \times |\mathscr{U}| \times |E_N|!)$ time complexity for subsystem i, where $|F_i|$ is the number of faults in the subsystem. Note, that in the case that no globally accurate diagnoser can be derived using neighboring subsystems, the solution gradually expands to include all subsystems. Therefore, the time complexity of our proposed method in Algorithm 9 for subsystem i is $O(|F_i| \times |\mathscr{U}| \times |E|!)$, where $|E|$ is total number of equations in the system.

In practice, Algorithm 6 finds the answer much faster. For example, consider Fig. 25 where algorithm 6 is searching for a set of equations to solve q_1. As soon as the algorithm reaches to an equation which does not have the required unknown variable, the algorithm discards that equation and, therefore, avoids enumerating the rest of the candidate equations in that branch. To achieve even faster solutions, we can sort the equations by the number of their unknown variables before the search. In this way, the algorithm starts with equations with fewer unknown variables and, therefore, has to expand fewer branches in average. For example, consider the case where the equations are sorted in Fig. 25. In this scenario, the algorithm starts with e_6 which has no unknown variable and that is the solution. Note that this step does not improve the worst case scenario, but reduces the average time significantly.

The equation-based solution is exponential in terms of number of equations in the system. The MSO-based solution is exponential in terms of number of MSOs in the system. The total number of MSO sets for fault detection and isolation grows exponentially as the

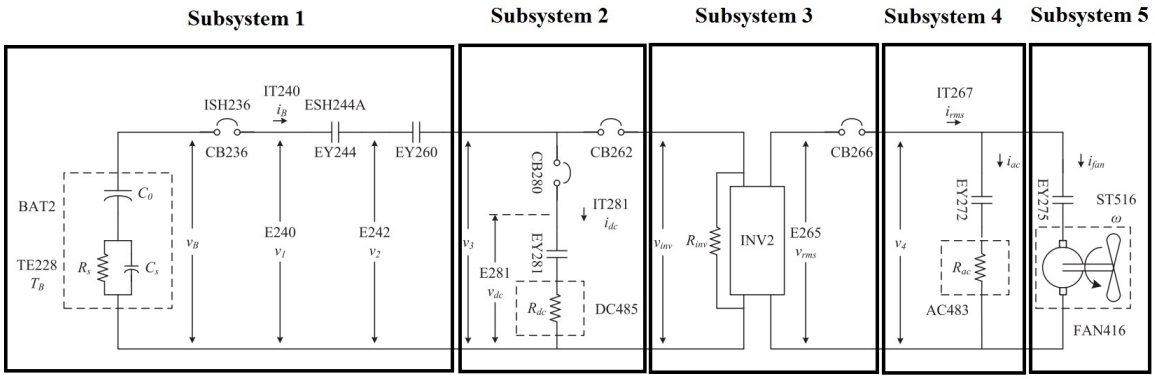

Figure 30: ADAPT-Lite subsystems [36].

number of measurements increase [5]. Consider definition 40. The total number of redundancies introduced into the system model is equal to the number of measurements, $|M|$. Theoretically, each MSO can include from one to $|M|$ measurements. Therefore, the total number of MSOs, N_{MSO} is proportional to all possible combinations of the measurements:

$$N_{MSO} \propto \sum_{i=1}^{|M|} \binom{|M|}{i} = 2^{|M|} \tag{73}$$

In general, there are much more MSOs in a system than equations. For example, the running example in this paper has 20 equations and the fault diagnosis toolbox generated 165 MSOs for this system. Therefore, we expect the equation-based approach to solve the problem in a more efficient way. In the next section, we demonstrate computational advantage of the equation-based method through a case study.

4.5 Case study

ADAPT-Lite system is designed to emulate the operation of unmanned aircraft electrical power systems [36]. The system has five subsystems: 1) the battery, 2) the direct current (DC) electric load, 3) the inverter, 4) the alternative current (AC) electric load, and 5) the electric fan (see Fig 30). The system has seven measurements: y_{E240}, y_{E242}, and y_{E281} represent DC voltage measurements in the system, y_{IT240} represents the battery current,

y_{E265} represents the inverter AC output voltage, y_{IT267} is the inverter AC output current, and y_{ST516} is the fan rotational speed. We consider six faults in the system: f_{E240} and f_{E242} are the sensor faults in y_{E240}, and y_{E242}, respectively, f_{dc} represents the fault in the DC load, f_{INV} models the inverter fault, f_{ac} represents the fault in the AC load, and f_{fan} is the fan fault. The ADAPT-Lite system has several circuit breakers (CB236, CB262, CB266, and CB280), and relays (EY244, EY260, EY281, EY272, and EY275) and, therefore, is a hybrid system. In Chapter V, we discuss diagnosis in hybrid systems. In this chapter, we focus on distributed diagnosis. Therefore, we assume all the circuit breakers and relays are on and there is no mode change in the system. The set of equations in each subsystem are presented as follows.

Subsystem 1 (battery): the set of equations in the first subsystem are

$$\begin{aligned}
e_{a1} &: \dot{v}_0 = \frac{1}{C_0}(-i_B) \\
e_{a2} &: v_0 = \int \dot{v}_0 dt \\
e_{a3} &: \dot{v}_s = \frac{1}{C_s}(i_B R_s - v_s) \\
e_{a4} &: v_s = \int \dot{v}_s dt \\
e_{a5} &: v_B = v_0 - v_s \\
e_{a6} &: v_1 = v_B \\
e_{a7} &: v_2 = v_1 \\
e_{a8} &: v_3 = v_2 \\
e_{a9} &: y_{E240} = v_1 + f_{E240} \\
e_{a10} &: y_{E242} = v_2 + f_{E242} \\
e_{a11} &: y_{IT240} = i_B
\end{aligned} \tag{74}$$

where $V_{a1} = \{\dot{v}_0, v_0, i_B, \dot{v}_s, v_s, v_B, v_1, v_2, v_3\}$ is the set of unknown variables in this subsystem, the set of measurements is $M_{a1} = \{y_{E240}, y_{E242}, y_{IT240}\}$, $F_{a1} = \{f_{E240}, f_{E242}\}$ represents subsystem faults, and C_0, C_s and R_s are the parameters in the subsystem. The battery is directly connected to the second subsystem (DC load).

Subsystem 2 (DC load): the DC load is modeled by an electric resistance, R_{dc}. The set of equations for this subsystem are

$$\begin{aligned} e_{a12} &: v_3 = v_{dc} \\ e_{a13} &: i_B = i_{dc} + i_{inv} \\ e_{a14} &: i_{dc} = f_{dc}\frac{v_{dc}}{R_{dc}} \\ e_{a15} &: y_{E281} = v_{dc} \end{aligned} \tag{75}$$

where $V_{a2} = \{v_3, v_{dc}, i_B, i_{dc}, i_{inv}\}$ is the set of unknown variables in the subsystem, $M_{a2} = \{y_{E281}\}$ represents the set of subsystem measurements, $F_{a2} = \{f_{dc}\}$ is the set of faults associated with this subsystem, and R_{dc} is the only parameter in the subsystem. Subsystems 1 and 2 are first order connected and their shared variables are $V_{a1} \cap V_2 = \{v_3, i_B\}$.

Subsystem 3 (inverter): the inverter converts DC power to AC. When there is no fault in the subsystem and the input voltage, v_{in}, is above $18V$, the output voltage, v_{rms}, stays at $120V$. R_{inv} represents the internal resistance in the inverter and e is the inverter efficiency rate. The set of equation for the subsystem are

$$\begin{aligned} e_{a16} &: v_{in} = v_{dc} \\ e_{a17} &: v_{rms} = 120.(v_{in} > 18).f_{INV} \\ e_{a18} &: i_{inv} = \frac{v_{rms}.i_{rms}}{e.v_{in}} + \frac{v_{inv}}{R_{inv}} \\ e_{a19} &: y_{E265} = v_{rms} \end{aligned} \tag{76}$$

where $V_{a3} = \{v_{in}, v_{dc}, v_{rms}, i_{inv}, i_{rms}\}$ defines the set of subsystem unknown variables, $M_{a3} = \{y_{E265}\}$ is the set of subsystem measurements, $F_{a3} = \{f_{INV}\}$ defines the set of subsystem

faults, and e and R_{inv} are the subsystem parameters. Subsystems 2 and 3 are first order connected and their shared variables are $V_{a2} \cap V_{a3} = \{v_{dc}, i_{inv}\}$. Subsystems 1 and 3 are second order connected because they have no shared variable and they are both first order connected to the second subsystem.

Subsystem 4 (AC load): like the DC load, the AC load is modeled with an electric resistance, R_{ac}. The set of equations for this subsystem are

$$\begin{aligned} e_{a20} &: v_4 = v_{rms} \\ e_{a21} &: i_{ac} = f_{ac}\frac{v_4}{R_{ac}} \\ e_{a22} &: i_{rms} = \frac{1}{\sqrt{2}}|\sqrt{2}i_{fan}(cos\phi + jsin\phi) + \sqrt{2}i_{ac}| \\ e_{a23} &: y_{IT267} = i_{rms} \end{aligned} \tag{77}$$

where $V_{a4} = \{v_4, v_{rms}, i_{ac}, i_{rms}, i_{fan}\}$ represents the set of subsystem unknown variables, $M_{a4} = \{y_{IT267}\}$ represents the set of measurements in the subsystem, $F_{a4} = \{f_{ac}\}$ is the set of subsystem faults, and R_{ac} and ϕ are the parameters. Subsystems 3 and 4 are first order connected and their shared variable is $V_{a3} \cap V_{a4} = \{v_{rms}\}$.

Subsystem 5 (electric fan): the fan rotational speed, ω, is a function of fan current, i_{fan}. The last subsystem equations are

$$\begin{aligned} e_{a24} &: i_{fan} = f_{fan}\frac{v_4}{R_{fan}} \\ e_{a25} &: \dot{\omega} = \frac{1}{J_{fan}}(\frac{i_{fan}}{B_{fan}} - \omega) \\ e_{a26} &: \omega = \int \dot{\omega} dt \\ e_{a27} &: y_{ST516} = \omega \end{aligned} \tag{78}$$

where $V_{a5} = \{i_{fan}, v_4, \dot{\omega}, \omega, i_{fan}\}$ is set of unknown variables in the subsystem, $M_{a5} = \{y_{ST516}\}$ represents the set of subsystem measurements, and $F_{a5} = \{f_{fan}\}$ is the set of subsystem faults. Fan electrical resistance, R_{fan}, fan inertial, J_{fan}, and fan mechanical

resistance, B_{fan}, are the parameters. Subsystems 4 and 5 are first order connected and $V_{a4} \cap V_{a5} = \{i_{fan}, v_4\}$ is the set of shared variables among these subsystems. More details of the ADAPT-Lite system are presented in [36]. In this case study, we use the ADAPT-Lite system to represent the application of our proposed distributed diagnosis methods.

4.5.1 MSO-based Method Using Global Model

For the ADAPT system the fault diagnosis toolbox generates 258 MSOs. To find the optimum measurements, the global MSOs selection algorithm solves an optimization problem for each subsystem. Table 13 shows the set of measurements that we need to add for each of the subsystem diagnosers to achieve maximum possible detectability and isolability using our global MSOs selection algorithm. In the first subsystem all the faults are locally detectable and isolable, and, therefore, this subsystem does not require any measurement from the other subsystems. For each other subsystem, we have to transfer exactly one measurement to achieve maximum diagnosability.

Table 13: Set of augmented measurements to each ADAPT subsystem using global method

Subsystem	Set of augmented measurements
Subsystem 1	-
Subsystem 2	y_{IT267}
Subsystem 3	y_{E281}
Subsystem 4	y_{ST516}
Subsystem 5	y_{E265}

Table 14 shows the set of MSOs for each local diagnoser. Not that the global MSOs selection method only minimizes the number of shared variables, but the subsystems may require equations from the other subsystems. For example, the first subsystem in ADAPT does not require any additional measurement to locally detect and isolate its faults, however, as we can see in Table 14, this subsystem requires several equations from the other

subsystems. The total time for MSO generation, and solving the optimization problems to

Table 14: Set of MSOs for each local diagnoser using global method

Subsystem	Set of MSOs
1	$MSO_{a11} = \{e_{a8}, e_{a10}, e_{a11}, e_{a12}, e_{a13}, e_{a14}, e_{a16}, e_{a17}, e_{a18}, e_{a20}, e_{a21}, e_{a22}, e_{a24}\}$ $MSO_{a12} = \{e_{a7}, e_{a9}, e_{a10}\}$ $MSO_{a13} = \{e_{a7}, e_{a8}, e_{a9}, e_{a11}, e_{a12}, e_{a13}, e_{a14}, , e_{a16}, \ldots$ $e_{a17}, e_{a18}, e_{a20}, e_{a21}, e_{a22}, e_{a24}\}$
2	$MSO_{a21} = \{e_{a1}, e_{a2}, e_{a3}, e_{a4}, e_{a5}, e_{a6}, e_{a7}, e_{a8}, e_{a12}, e_{a13}, e_{a14}, e_{a15}, e_{a16}, \ldots$ $e_{a18}, e_{a20}, e_{a21}, e_{a22}, e_{a23}, e_{a24}\}$ $MSO_{a22} = \{e_{a1}, e_{a2}, e_{a3}, e_{a4}, e_{a5}, e_{a6}, e_{a7}, e_{a8}, e_{a12}, e_{a13}, e_{a14}, e_{a15}, e_{a16}, \ldots$ $e_{a17}, e_{a18}, e_{a23}\}$
3	$MSO_{a31} = \{e_{a15}, e_{a16}, e_{a17}, e_{a19}\}$
4	$MSO_{a41} = \{e_{a21}, e_{a22}, e_{a23}, e_{a24}, e_{a25}, e_{a26}, e_{a27}\}$ $MSO_{a42} = \{e_{a1}, e_{a2}, e_{a3}, e_{a4}, e_{a5}, e_{a6}, e_{a7}, e_{a8}, e_{a12}, e_{a13}, e_{a14}, e_{a16}, \ldots$ $e_{a17}, e_{a18}, e_{a20}, e_{a21}, e_{a22}, e_{a23}, e_{a25}, e_{a26}, e_{a27}\}$
5	$MSO_{a51} = \{e_{a19}, e_{a20}, e_{a24}, e_{a25}, e_{a26}, e_{a27}\}$

find a set of MSOs for each subsystem with minimum shared variables was 118s, where the experiment was run on a desktop with a the Intel Core i7-4790 processor (3.60 GHz).

4.5.2 MSO-based Method Using Neighboring Subsystems

Table 15: Set of augmented measurements to each subsystem model using neighboring subsystems

Subsystem	Set of augmented measurements
Subsystem 1	–
Subsystem 2	y_{E265}, y_{IT267}
Subsystem 3	y_{E281}
Subsystem 4	y_{E265}, y_{ST516}
Subsystem 5	y_{E265}

Table 16: Set of MSOs for each local diagnoser using neighboring subsystems

Subsystem	Set of MSOs
1	$MSO_{b11} = \{e_{a1}, e_{a2}, e_{a3}, e_{a4}, e_{a5}, e_{a6}, e_{a9}, e_{a11}\}$ $MSO_{b12} = \{e_{a7}, e_{a9}, e_{a10}\}$ $MSO_{b13} = \{e_{a1}, e_{a2}, e_{a3}, e_{a4}, e_{a5}, e_{a6}, e_{a7}, , e_{a10}, e_{a11}\}$
2	$MSO_{b21} = \{e_{a1}, e_{a2}, e_{a3}, e_{a4}, e_{a5}, e_{a6}, e_{a7}, e_{a8}, e_{a12}, e_{a13}, e_{a14}, e_{a15}, e_{a16}, e_{a18}, e_{a19}, e_{a23}\}$
3	$MSO_{b31} = \{e_{a15}, e_{a16}, e_{a17}, e_{a19}\}$
4	$MSO_{b41} = \{e_{a21}, e_{a22}, e_{a23}, e_{a24}, e_{a25}, e_{a26}, e_{a27}\}$ $MSO_{b42} = \{e_{a19}, e_{a20}, e_{a21}, e_{a22}, e_{a23}, e_{a25}, e_{a26}, e_{a27}\}$
5	$MSO_{b51} = \{e_{a19}, e_{a20}, e_{a24}, e_{a25}, e_{a26}, e_{a27}\}$

In the previous subsection we used the global model of the ADAPT system to generate the MSOs, and then selected the MSOs for each subsystem diagnosers using the BILP algorithm run on the global MSO set. In this section, we achieve global diagnosability of a subsystem diagnoser by incrementally adding a minimum number of measurements from the neighbors of this subsystem till the global diagnosability property is established. Compared to the global method, the computational complexity is much lower in this approach. For example, to design the diagnosis system for the first subsystem, we start with its set of equations and the fault diagnosis toolbox generates only 3 MSOs for this subsystem which are enough to detect and isolate all the subsystem faults. Therefore, we do not need to consider other subsystems. Using the same approach for every subsystem, the set of measurements that we need to transfer to each ADAPT subsystem are presented in Table 15.

In some cases, considering the first order neighboring subsystem is not enough to detect and isolate all the faults and the algorithm has to expend to the higher order neighbors. For example, for subsystem 2, the algorithm can not find any solution when it considers the first order neighbors (subsystem 1 and subsystem 3). Therefore, it extends the search

to the second order neighboring subsystems (subsystem 4). For these four subsystems, the fault diagnosis toolbox generates 44 MSOs, and the algorithm selects 1 MSO to detect and isolate the subsystem fault. We can see the advantage of using neighboring subsystems over the global approach in this example because the computational cost of solving the set covering problem for 44 MSOs is significantly less than 258 MSOs. Using the same processor, the total time for this method was 2.9s. This is significantly less than the total time for the global method. However, as we mentioned earlier, this algorithm does not guarantee global minimum communication among subsystems. In the global method subsystem 2 only requires y_{IT267} to be transfered from the other subsystems. However, in this approach it requires y_{E265}, and y_{IT267}. Table 16 shows the set of MSOs for each local diagnoser using the neighboring subsystems. Compared to the global method (see Table 14), the MSOs in this approach tend to have fewer number of equations.

4.5.3 Equation-based Distributed Diagnosis

Instead of generating all the MSOs and selecting a subset of MSOs for each local diagnoser, Algorithm 9 finds a minimal set of equations from neighboring subsystems that guarantees complete diagnosability performance. Table 17 shows the set of equations that

Table 17: Set of augmented equations and measurements to each subsystem model using equation-based approach

Subsystem	Augmented equations	Augmented measurements
Subsystem 1	-	-
Subsystem 2	e_{a11}, e_{a16}, e_{a18}, e_{a19}, e_{a23}	y_{IT240}, y_{E265}, y_{IT267}
Subsystem 3	e_{a15}	y_{E281}
Subsystem 4	e_{a19}, e_{a25}, e_{a26}, e_{a27}	y_{E265}, y_{ST516}
Subsystem 5	e_{a19}, e_{a20}	y_{E265}

we need to augment from neighboring subsystems to each local diagnoser to achieve maximum possible detectability and isolability using equation-based approach.

Table 18: Set of MSOs for each local diagnoser using equation-based approach

Subsystem	Set of MSOs
1	$MSO_{c11} = \{e_{a1}, e_{a2}, e_{a3}, e_{a4}, e_{a5}, e_{a6}, e_{a9}, e_{a11}\}$ $MSO_{c12} = \{e_{a7}, e_{a9}, e_{a10}\}$ $MSO_{c13} = \{e_{a1}, e_{a2}, e_{a3}, e_{a4}, e_{a5}, e_{a6}, e_{a7}, , e_{a10}, e_{a11}\}$
2	$MSO_{c21} = \{e_{a11}, e_{a13}, e_{a14}, e_{a15}, e_{a16}, e_{a18}, e_{a19}, e_{a23}\}$
3	$MSO_{c31} = \{e_{a15}, e_{a16}, e_{a17}, e_{a18}, e_{a19}\}$
4	$MSO_{c41} = \{e_{a19}, e_{a20}, e_{a21}, e_{a22}, e_{a23}, e_{a25}, e_{a26}, e_{a27}\}$
5	$MSO_{c51} = \{e_{a19}, e_{a20}, e_{a24}, e_{a25}, e_{a26}, e_{a27}\}$

Table 17 also represents the set of measurements that we need to transfer to each ADAPT subsystem. Note that like the previous subsection, our algorithm in this subsection does not guarantee global minimum communication. For example, subsystem 2 requires three measurements from other subsystems (see Table 17). However, we can see in Table 11 that the complete diagnosability is achievable by only one additional measurement. To detect and isolate faults in each subsystem, we use the subsystem equations and the set of augmented equations to generate MOSs. Table 18 shows the set of MSOs for each local diagnoser using the equation-based method. We used the same desktop (Intel Core i7-4790 processor, 3.60 GHz) to run this experiment and the total time was $0.32s$. This shows the computational advantage of this method.

4.6 Discussion and Conclusions

Two structural distributed diagnosis methodologies are presented in this chapter. The proposed algorithms provide the maximum possible detectability and isolability that can be achieved for a system given a set of measurements. Unlike previous work, such as [23, 35] our proposed methods directly work with system equations, and therefore, do not need to use the temporal response and event ordering in the diagnosis, all of which are derived properties, and, therefore, require additional computation. Using a purely structural approach, reduces the overall diagnosability of the system for the given set of measurements.

However, it also reduces the number of assumptions we need to make about the fault characteristics, order of events in the diagnoses subsystems (which can be error-prone), and we do not have to analyze in detail the subsystem dynamics.

We proposed two algorithms for MSOs selection for the distributed diagnosis. The first algorithm guarantees that the subsystems share the minimum number of measurements, implying that we minimize the communication of measurement streams across subsystems of the global system. This is important because sending the data to other subsystems is costly in large scale systems. On the other hand, the second algorithm does not need to use the global model in the design process of the supervisory system. This makes the algorithm more practical, specially for the complex systems. However, the second algorithm does not guarantee that the number of shared variables among the subsystems are globally minimum.

The MSO-based methods generate the MSOs sets and select a subset of MSOs with minimum required shared variables. The total number of MSOs is exponential in terms of the system measurements and selecting a subset of MSOs with minimum shared measurements is equivalent to the set covering problem. Therefore, these algorithms have high computational cost specially for large-scale systems. Instead of selecting a subset of MSOs from the generated MSOs for each local diagnoser, the equation-based algorithm finds a minimal set of equations from neighboring subsystems that guarantees maximum possible detectability and isolability that can be achieved. The number of equations is significantly smaller than the number of MSOs. This makes the approach very feasible for large-scale complex systems.

CHAPTER V

FAULT DETECTION AND ISOLATION IN HYBRID SYSTEMS

Hybrid systems are characterized by continuous behaviors that are interspersed with discrete mode changes in the system, making the analysis of behaviors quite complex. Developing feasible approaches for on-line monitoring, fault detection, and fault isolation of complex hybrid and embedded systems such as automobiles, aircraft, power plants, and manufacturing processes, is essential in securing their safe, reliable, and efficient operation. Frequent changes in the operational modes of these systems because of operators actions such as changing gears in an automobile, or environmental changes, such as driving on a wet or icy road make the fault detection and isolation task in these systems challenging. It is important to detect and isolate faults in all the operating modes, and at the same time, not mistake a mode change as a fault in the system. The MSO approach has been used extensively for designing model based fault detection and isolation schemes for complex systems [106, 168]. We extend this approach by working with equations that contain continuous and discrete variables to describe hybrid systems behavior. Using the mixed continuous-discrete equations, we define Hybrid Minimal Structurally Overdetermined (HMSO) sets for fault detection and isolation in hybrid systems.

In this chapter, we adopt a structural approach to developing a mode detection algorithm. We address the mode detection problem in hybrid systems as the first step in diagnoser design. The proposed method uses analytic redundancy methods to detect the operating mode of the system. The mode detection unit, once designed can track the operating mode even in the presence of system faults. When the operating mode is detected, the corresponding diagnosis methodology efficiently picks a set of HMSOs that guarantee complete fault diagnosability in the current mode. We develop two solutions for the HMSO selection problem. The first solution finds a subset of HMSOs with minimum cardinality

that satisfies a pre-specified diagnosability performance. Note that this is equivalent to finding an optimal solution for the set covering problem, which is known to be NP-hard [105] and, therefore, any algorithm for finding a set of HMSOs with minimal cardinality and required diagnosability performance will have exponential computational complexity. We formulate this problem as a binary integer linear programming (BILP) problem.

To reduce the computational complexity, we also develop a greedy search algorithm to find a minimal set of HMSOs that guarantee complete fault detectability and isolability in the current mode. Therefore, the selected HMSOs set may not have the minimum cardinality number. A larger number of HMSOs increases the residual generation computational cost but this increase in the computational complexity is negligible compared to the computational complexity of finding an optimal solution for the set covering problem. The selected HMSO set can then be used for residual generation. The proposed structural approach does not require pre-enumeration of all possible modes in the diagnoser design step. Therefore, our approach is feasible for hybrid systems with large number of switching elements, implying that the system can have large number of operating modes. We demonstrate its effectiveness using a case study on a Reverse Osmosis (RO) subsystem of an Advances Life Support System (ALS) for long duration manned space missions. Important challenges that can affect the success of our approach include the need for sufficiently detailed hybrid models that capture nominal and faulty behavior, and a sufficient number of sensors to make simultaneous mode change and fault detection and isolation possible.

The rest of this chapter is organized as follows. A formal definition of hybrid systems and the running example, a four-tank system, is presented in Section 5.1. The problems of mode detection and FDI for hybrid systems and our general approach to address these problems are formally described in Section 5.2. Section 5.3 presents our algorithm for mode detection. The fault diagnosis approach in hybrid systems is presented in Section 5.4. Section 5.5 presents the case study and Section 5.6 presents the conclusions of the chapter.

5.1 Hybrid Systems

5.1.1 Hybrid systems modeling

A number of different approaches have been used for modeling hybrid systems. These include hybrid automata, hybrid I/O automata, mixed logic-dynamic systems, piecewise affine systems, and hybrid bond graphs. Heemels et al [82] have proved equivalences among several hybrid system modeling approaches. In this work, we model hybrid systems as a set of equations that contain both continuous and discrete variables, much like the mixed logic-dynamic approach proposed by Bemporad and Morari [13]. Formally, we define a hybrid system H as follows:

Definition 28 (Hybrid system model)**.** *A hybrid system model H is a tuple: (X, Σ, T, E, M), where X represents the set of continuous variables in the system, $\Sigma = \Sigma_1 \cup \Sigma_2$ represents the set of discrete variables; Σ_1 are variables whose values are defined by controlled mode transitions, i.e., signals that are generated external to the system, e.g., by a controller; Σ_2 are a set of variables whose values are defined by autonomous mode transitions, i.e., they are based on values associated with continuous variables in the system; $T : X \rightarrow \Sigma_2$ represents the set of conditions on continuous variables that define autonomous mode transitions; and $E : X \times \Sigma \rightarrow X$ represents the set of equations that define the hybrid system behavior. The total number of modes in the hybrid system, M is exponential in the number of discrete variables, Σ, i.e., $M = 2^{|\Sigma|}$.*

This definition adopts the Mosterman and Biswas [136] approach to model mode transition functions. E generalizes this approach, and adopts the mixed logic-dynamics [13] form that combines continuous and discrete variables to model hybrid systems. Since the number of modes in a hybrid system is an exponential function of its discrete variables, any approach that requires pre-computing all the possible modes of operation is computationally intractable. We extend the hybrid model H to support diagnosis by considering measurements and faults in the system. Thus the hybrid system model for diagnosis, H_d, is defined as:

Definition 29 (Hybrid system model for diagnosis)**.** *A hybrid system model for diagnosis is $H_d = H \cup Y \cup Z \cup F$, where Y is a set of continuous measurements that are made on the system, i.e., $Y = \Phi(X)$; Z represents discrete measurements, where $Z \subseteq \Sigma$; and F is the set of fault parameters that are of diagnostic interest.*

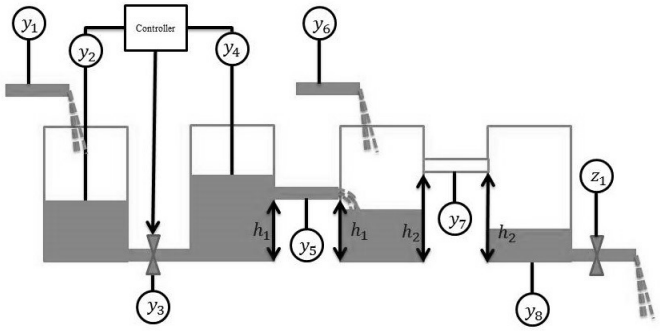

Figure 31: Running example: Hybrid Four Tank System.

We use a configured four tank system, shown in Fig. 31, as a running example throughout this chapter to illustrate the hybrid diagnosis problem, and to develop our structural approach for hybrid systems diagnosis. Tanks 1 and 3 have inflows that we assume are know, and represented by measurements y_1 and y_6, respectively. There are connecting pipes between adjacent tanks. For Tanks 1 and 4 these pipes are located at the bottom of the tanks. For Tanks 2 and 3, the pipes are located at known heights, h_1 and h_2, respectively. The flow through the connecting pipe between Tanks 1 and 2 is controlled by an on/off valve, whose state is controlled by an external signal, generated by a controller. The controller function depends on the pressure in Tanks 1 and 2, which are measured values, y_2 and y_4, respectively. The flow through this valve is a measured variable, y_3.

For the valve on the outlet pipe of Tank 4, we assume the discrete on/off state of the valve is determined by the known external control signal z_1. The direction and amount of flow between Tanks 2 and 3, and between Tanks 3 and 4, depend on the liquid levels in the three tanks. Of these three pressures, only the pressures in Tanks 2 and 4 are measured,

and represented by the variables y_4 and y_8, respectively. In addition, the flow through the two connecting pipes are also measured, and represented by y_5 and y_7, respectively. All of the continuous and discrete measurements are shown as encircled variables in the figure. The equations, transition conditions, and the output variables corresponding to the hybrid system diagnosis model for our running example are listed below. $E = \{e_i | 1 \leq i \leq 21\}$ defines the set of equations, $T = \{t_{1:5}\}$ is the set of transitions, $X = \{\dot{p}_{1:4}, p_{1:4}, q_{1:4}, q_{in_{1,3}}\}$ defines the set of continuous variables, $\Sigma = \{\sigma_{1:6}\}$ is the set of discrete variables, $Y = \{y_{1:8}\}$ defines the set of continuous measurements, $Z = \{z_1\}$ is the set of discrete measurements, and $F = \{f_i | 1 \leq i \leq 6\}$ represents the set of faults associated with the hybrid system model. The system has six binary variables and, therefore, 2^6 modes.

$$
\begin{aligned}
&e_1 : \dot{p}_1 = \frac{1}{C_{T1}}(q_{in_1} - q_1 - f_1) && e_7 : \dot{p}_3 = \frac{1}{C_{T3}}(q_{in_3} + q_2 - q_3)\\
&e_2 : p_1 = \int \dot{p}_1\, dt && e_8 : p_3 = \int \dot{p}_3\, dt\\
&e_3 : q_1 = \sigma_1 \frac{p_1 - p_2}{R_{P1} + f_2} && e_9 : q_3 = \frac{\sigma_4 p_3 - \sigma_5 p_4}{R_{P3} + f_5}\\
&t_1 : \sigma_1 = \begin{cases} 1 & p_1 \geq p_2 \\ 0 & p_1 < p_2 \end{cases} && t_4 : \sigma_4 = \begin{cases} 1 & p_3 \geq \rho g h_2 \\ 0 & p_3 < \rho g h_2 \end{cases}\\
&e_4 : \dot{p}_2 = \frac{1}{C_{T2}}(q_1 - q_2 - f_3) && t_5 : \sigma_5 = \begin{cases} 1 & p_4 \geq \rho g h_2 \\ 0 & p_4 < \rho g h_2 \end{cases}\\
&e_5 : p_2 = \int \dot{p}_2\, dt && e_{10} : \dot{p}_4 = \frac{1}{C_{T4}}(q_3 - q_4 - f_6)\\
&e_6 : q_2 = \frac{\sigma_2 p_2 - \sigma_3 p_3}{R_{P2} + f_4} && e_{11} : q_4 = \sigma_6 \frac{p_4}{R_{P4}}\\
&t_2 : \sigma_2 = \begin{cases} 1 & p_2 \geq \rho g h_1 \\ 0 & p_2 < \rho g h_1 \end{cases} && e_{12} : p_4 = \int \dot{p}_4\, dt\\
&t_3 : \sigma_3 = \begin{cases} 1 & p_3 \geq \rho g h_1 \\ 0 & p_3 < \rho g h_1 \end{cases}
\end{aligned}
\tag{79}
$$

$$
\begin{aligned}
e_{13} &: q_{in_1} = y_1 & \qquad e_{18} &: q_{in_3} = y_6 \\
e_{14} &: p_1 = y_2 & \qquad e_{19} &: q_3 = y_7 \\
e_{15} &: q_1 = y_3 & \qquad e_{20} &: p_4 = y_8 \\
e_{16} &: p_2 = y_4 & \qquad e_{21} &: \sigma_6 = z_1 . \\
e_{17} &: q_2 = y_5 &
\end{aligned}
\tag{80}
$$

In the equations, p_i represents the pressure in Tank i, and q_i represents the flow through the connecting pipe associated with the adjoining tanks. q_{in_i} represents the inflow into Tank i, and the on/off state of valve i is represented by $\sigma_i = 1$ (on), and $\sigma_i = 0$ (off). The y_is represent continuous measurements, and z_ks represent discrete measurements. The capacity of Tank i is represented as C_{Ti}, and the resistance of the connecting pipe to the right of a tank is represented by R_{Pi}. The fault parameters are modeled by f_i. f_1 represents a leak in Tank 1, f_2 represents a clog in the connecting pipe to the right of Tank 1, f_3 represents a leak in Tank 2, f_4 represents a clog in the connecting pipe to the right of Tank 2, f_5 represents a clog in the connecting pipe to the right of Tank 3, and f_6 represents a leak in Tank 4. Other parameters required by the model include the density of the liquid, ρ, and the gravitational constant, g. The height of the pipes, h_1 and h_2, are assumed to be constant and known.

5.1.2 Mode detection and mode observability in hybrid systems

This subsection introduces the basic concepts and definitions associated with structural mode detection and fault detection and isolation in hybrid systems. In this work, we extend the DM decomposition approach [59] for mode detection. To detect the operating mode of the system, we have to know the value of all of its discrete variables (e.g., $\Sigma = \{\sigma_{1:6}\}$ in the running example). To compute a discrete variable σ_i we need a subset of determined transition and behavior equations (sets T and E) that include σ_i and a sequential ordering for computing σ_i. For example, to compute σ_1 in our running example (see (79) and (80)),

we can use e_{14} and e_{16} to compute p_1, and p_2, respectively and substitute for p_1, and p_2, in t_1 to compute σ_1.

We define the notion of detectable modes in hybrid systems in the presence of faults, by introducing the concepts of structural determined (SD) sets and minimal structural determined (MSD) sets as follows.

Definition 30. *(Structural Determined Set in Hybrid Systems) Consider a set of equations and transitions and their associated continuous variables, discrete variables, and faults:* (E,T,X,Σ,F). *This set of equations and transitions is structurally determined (SD) if the cardinality of the set E plus cardinality of T is greater than or equal to the sum of the cardinalities of the sets X,* Σ*, and F , i.e.* $|E|+|T| \geq |X|+|\Sigma|+|F|$.

Definition 31. *(Minimal Structural Determined Set in Hybrid Systems) A set of structurally determined equations is minimal structurally determined (MSD) if it has no subset of structurally determined equations.*

Consider the four tank system represented by equations (79) and (80). A minimal structurally determined set in this system is $MSD_1 = (E_1,T_1,X_1,\Sigma_1,F_1)$, where $E_1 = \{e_{14},e_{16}\}$, $T_1 = \{t_1\}$, $X_1 = \{p_1,p_2\}$, $\Sigma_1 = \{\sigma_1\}$, and $F_1 = \{\}$. For the sake of brevity, in the rest of the chapter we simply say a specific equation, transition, variable, or fault is a member of a MSD (e.g., $\sigma_1 \in MSD_1$). MSDs represent solvable set of variables in the system and can be used for mode detection. We define a detectable discrete variable in a hybrid system as follows.

Definition 32. *(Detectable discrete variable in hybrid systems) A discrete variable* $\sigma \in \Sigma$ *is detectable for diagnostic hybrid system,* H_d*, if there is a minimal structurally determined set* MSD_i *in the system, such that* $\sigma \in MSD_i$.

For example, discrete variable σ_1 in equation (79) is detectable because $\sigma_1 \in MSD_1$.

Babaali and Egerstedt [7] defined mode observability based on continuous variables trajectories in different modes. In this work, we define a mode observable diagnostic hybrid system model as

Definition 33 (Mode Observable Diagnostic Hybrid system)**.** *A hybrid system $H_d = (X, \Sigma, T, E, M, Y, Z, F)$ is mode observable if all the discrete variables $\sigma_i \in \Sigma$ are detectable (i.e., they are directly observed, or their values are computable given H_d).*

5.1.3 Fault detection and isolation in hybrid systems

Mode detection is an integral part of hybrid system diagnosis. We derive a mode detection scheme based on the structural properties of the system equations, i.e., the MSD approach, making the assumption that the hybrid system is mode observable. In addition, we assume faults and mode changes do not occur at the same time and a fault is detected and isolated in the same mode in which it initially occurs. Therefore, our approach to fault detection and isolation, requires synchronous solution of the mode identification, fault detection, and fault isolation tasks. We describe our diagnosis algorithm in greater detail in the next section.

We formally define the concepts of Hybrid Structurally Overdetermined (HSO) and Hybrid Minimal Structurally Overdetermined (HMSO) sets for hybrid system diagnosis below.

Definition 34. *(Hybrid Structural Overdetermined Set) Consider a set of equations and its associated continuous variables, discrete variables, and faults: (E, X, Σ, F). This set of equations is a set of hybrid structurally overdetermined (HSO) if the cardinality of the set E is greater than the cardinality of set X, i.e. $|E| > |X|$ and all the $\sigma \in \Sigma$ are detectable.*

Definition 35. *(Hybrid Minimal Structurally Overdetermined Set) A HSO is hybrid minimal structurally overdetermined (HMSO) if it has no subset of hybrid structurally overdetermined equations.*

For example, consider the four tank system in equations (79) and (80). A hybrid minimal structurally overdetermined set in this system is $HMSO_1 = (E_2, X_2, \Sigma_2, F_2)$, where $E_2 = \{e_1, e_2,\ e_{13}, e_{14}, e_{15}\}$, $X_2 = \{\dot{p}_1, p_1, q_1, q_{in_1}\}$, $\Sigma_1 = \{\}$, and $F_1 = \{f_1\}$. Note that $HMSO_1$ does not include a discrete variable. For the sake of brevity and simplification we simply say a specific equation, variable, measurement, or fault is a member of a HMSO in the rest of the chapter. For example, we say $f_1 \in HMSO_1$. HMSOs represent the redundancies in the hybrid system and can be used for fault detection and isolation. We define fault detectability in hybrid systems as follows.

Definition 36. *(Detectable fault in hybrid systems) A fault $f \in F$ is detectable in hybrid system H_d if there is a hybrid minimal structurally overdetermined set $HMSO_i$ in the system, such that $f \in HMSO_i$.*

For example, consider fault f_1 in the running example in equation (79). f_1 is detectable because $f_1 \in HMSO_1$.

Definition 37. *(Isolable faults in hybrid systems) A fault $f_i \in F$ is isolable from fault $f_j \in F$ if there exists a hybrid minimal structurally overdetermined set $HMSO_i$ in the system H_d, such that $f_i \in HMSO_i$ and $f_j \notin HMSO_i$.*

As an example f_1 is isolable from f_2 because $f_1 \in HMSO_1$ and $f_2 \notin HMSO_1$.

5.2 Problem Formulation and Solution

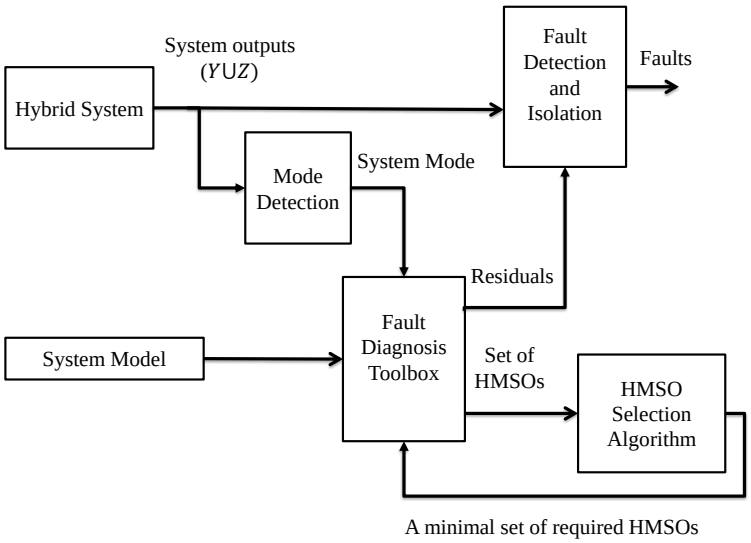

Figure 32: Fault Detection and Isolation in Hybrid Systems.

Fig.32 represents our diagnosis approach for hybrid systems. Our approach has two main steps: 1) mode detection, and 2) fault detection and isolation. To detect the operating mode of a hybrid system, we have to compute the value of its discrete variables. Therefore, finding a minimal structural determined (MSD) set to detect each discrete variable is the first problem we have to address in this chapter. We assume the hybrid system is mode observable and develop an algorithm to find a MSD to detect each discrete variable in the presence of possible faults.

Definition 38. *(Mode detection problem): Let* Σ *denote the set of discrete variables in a diagnostic hybrid system,* H_d*. Our goal is to develop an algorithm that selects a minimal structurally determined set* MSD_i *for each* $\sigma_i \in \Sigma$ *such that* $\sigma_i \in MSD_i$*. More formally, the mode detection problem for hybrid systems can be defined as:*

$$\begin{aligned} &\forall \sigma_i \in \Sigma \quad \textit{find } MSD_i, \\ &\textit{such that} \quad \sigma_i \in MSD_i. \end{aligned} \tag{81}$$

The mode detection unit in Fig.32 continuously tracks the operating mode of the system. When the hybrid system transitions to a new mode, our diagnosis approach generates a new set of residuals to support fault detection and isolation in that mode. Note that if the hybrid system revisits a mode that has been active earlier, we can simply revive a cashed set of residuals that was generated for this mode. When our mode detection algorithm reports a new mode of operation, we apply the fault diagnosis toolbox [68] to generate the entire set of HMSOs for the operating mode. We then select a minimal set of HMSOs for fault detection and isolation in this mode. Developing an algorithm to select a minimal set of HMSOs that guarantees complete fault detectability and isolability in each operating mode of hybrid systems is the second problem we address in this chapter.

Assuming the modes are detectable, we can define fault detection and isolation problem in hybrid systems as follows. Each HMSO is sensitive to a set of faults and, therefore, can be used to detect and isolate them from the other faults in the hybrid system. Given the hybrid system model for diagnosis, H_d, with a set of faults F, we assume that the set of HMSOs in each mode, m, $HMSO_m = \{HMSO_{m_1}, HMSO_{m_2}, \ldots, HMSO_{m_r}\}$, is sufficient to detect and uniquely isolate all of the faults in that mode, $F_m \subset F$. Note that the set of fault candidates is not necessarily the same in all the operating modes. Our goal is to develop an algorithm that selects a minimal subset of HMSOs, $HMSO_m^*$ for each mode, m, which guarantees the fault detectability for each fault $f_i \in F_m$ and fault isolability for each pair of faults $f_j \in F_m$ and $f_k \in F_m$. More formally, for each operation mode we define *minimal HMSO set* as a minimal set of HMSOs that can be used to achieve the complete structural diagnosability performance.

Definition 39. *(Minimal HMSO set for operation mode m): $HMSO_m^* \subset HMSO_m$ is a minimal set of HMSOs for diagnosis the faults in mode m, F_m, if $HMSO_m^*$ fulfills the following*

diagnosability performances

$$\begin{aligned} &\forall f_i \& f_j \in F_m : \\ &\exists HMSO_k \in HMSO_m^* : \\ &f_i \in HMSO_k, \\ &f_j \notin HMSO_k, \end{aligned} \tag{82}$$

and all proper subsets of $HMSO_m^*$ *do not.*

As discussed earlier, and illustrated in Fig.32, we use the fault diagnosis toolbox [68] to derive the set of residuals from the selected HMSOs, $HMSO_m^*$, in each operating mode. The derived residuals are used for FDI in the mode. Note that HMSO selection and residual generation are only required when the system transits to a mode for the first time. The generated residuals for each visited mode can be saved and used when the system returns to the mode later during system operations. Algorithm 10 summarizes this approach. In the next sections we present each step of the algorithm in greater detail.

Algorithm 10 Fault Detection and Isolation

1: **input:** (Y, Z)
2: Detect the current mode
3: Generate the model in the current mode
4: Generate the set of HMSOs for the current mode
5: Find a minimal HMSO set for FDI in the current mode
6: Genrate a residual from each HMSO
7: Apply the residuals for FDI

5.3 Mode Detection Algorithm

In this section, we present our proposed approach to finding a minimal set of constraints for detecting each discrete mode change during the hybrid system operation. We illustrate the procedure by solving this problem for the running example, and then generalize this

approach and present an algorithm that solves this problem. The tank system has six discrete variables. To detect each $\sigma_i \in \Sigma$, we have to find a determined set of equations and transitions (MSD) that include σ_i. We illustrate this for σ_1 by starting with all of the system equations and transitions in H_d that have σ_1. These equations and transitions are e_3, and t_1 as shown in Fig. 33. Then for the additional variables and faults in each equation or transition we need to add other equations so that they may be replaced to generate an equation with only one unknown variable. For t_1 we need to add two new constraints: one with p_1 and the other with p_2. e_{14} and e_{16} represent measurement equations and p_1 and p_2 are the only unknown variables in these equations, respectively. Therefore, these equations do not add any new variable to t_1, and $E_1 = \{e_{14}, e_{16}\}$ plus t_1 is a minimal structurally determined set that makes σ_1 detectable.

Figure 33: Detecting σ_1.

However, as it is shown in Fig. 33 there is no additional set of equations that we can add to e_3 to generate a structurally determined set. More formally, Fig. 33 depicts a matching algorithm whose general form is presented as Algorithm 11.If we initialize the algorithm with the set of unknown variables and faults, $\mathscr{U}$, (in this example p_1 and p_2 are the unknown variables) it returns a set of complete matching of variables and equations in the subsystem that includes the unknown variables. We use Algorithm 11 to find a MSD set to detect each discrete variable in the running example as it is shown in Fig. 34. Note that we used the same matching algorithm in Chapter IV (see Algorithm 6) to find minimal equations in the neighboring subsystems. Algorithm 12 summarizes the procedure. Table 19 shows the set of constrains in each MSD.

Algorithm 11 Count-Matchings

input: current matching $\mathscr{M}$
input: sets of determined variables $\mathscr{D}$ and undetermined variables $\mathscr{U}$
if $U = \emptyset$ **then**
    return $\mathscr{M}$ as a feasible (minimal) matching.
end if
for each $x \in \mathscr{U}$ **do**
    for each y which can determine x **do**
        Let $\mathscr{M}'$ be $\mathscr{M} \cup \{x \to y\}$
        Let $\mathscr{D}'$ be $\mathscr{D} \cup \{x\}$.
        Let $\mathscr{U}'$ be $\mathscr{U} \setminus \{x\}$.
        Add all the undetermined variables of y to $\mathscr{U}'$.
        COUNT-MATCHINGS$(\mathscr{M}', \mathscr{D}', \mathscr{U}')$
    end for
end for

Algorithm 12 Mode Detection

input: $(X, \Sigma, E, T, Y, Z, M, F)$
for each $\sigma \in \Sigma$ **do**
    C_σ = the set of equations and transitions that include σ
    for each $c \in C_\sigma$ **do**
        Let $\mathscr{M}$ be (c, σ)
        Let $\mathscr{D}$ be σ and $\mathscr{U}$ be the rest of variables in c
        MSD_σ = Count-Matchings($\mathscr{M}$, $\mathscr{D}$, $\mathscr{U}$)
        if $MSD_\sigma \neq \emptyset$ **then**
            select MSD_σ as a feasible MSD for σ.
        end if
    end for
end for

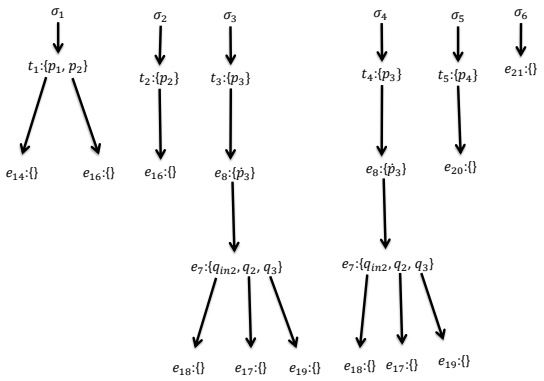

Figure 34: Detecting σ_1 - σ_6.

Table 19: Set of selected MSDs

HMSOs	Set of equations	discrete variable
MSD_1	t_1, e_{14}, e_{16}	σ_1
MSD_2	t_2, e_{16}	σ_2
MSD_3	t_3, e_8, e_7, e_{18}, e_{17}, e_{19}	σ_3
MSD_4	t_4, e_8, e_7, e_{18}, e_{17}, e_{19}	σ_4
MSD_5	t_5, e_{20}	σ_5
MSD_6	e_{21}	σ_6

To detect each discrete variable, we assign a reverse causality to the graphs shown in Fig. 34 to solve the MSDs for the discrete variables using the known variables. Equations (83) and (84) show the solutions for the discrete variables.

$$\sigma_1 = \begin{cases} 1 & y_2 \geq y_4 \\ 0 & y_2 < y_4 \end{cases}$$

$$\sigma_2 = \begin{cases} 1 & y_4 \geq \rho g h_1 \\ 0 & y_4 < \rho g h_1 \end{cases} \tag{83}$$

$$\sigma_3 = \begin{cases} 1 & \int \frac{1}{C_{T3}}(y_6 + y_5 - y_7) \geq \rho g h_1 \\ 0 & \int \frac{1}{C_{T3}}(y_6 + y_5 - y_7) < \rho g h_1 \end{cases}$$

$$\sigma_4 = \begin{cases} 1 & \int \frac{1}{C_{T3}}(y_6 + y_5 - y_7) \geq \rho g h_2 \\ \\ 0 & \int \frac{1}{C_{T3}}(y_6 + y_5 - y_7) < \rho g h_2 \end{cases}$$

$$\sigma_5 = \begin{cases} 1 & y_8 \geq \rho g h_2 \\ \\ 0 & y_8 < \rho g h_2 \end{cases} \tag{84}$$

$$\sigma_6 = z_1.$$

5.4 Fault Detection and Isolation Algorithm

In this section, we present our algorithm to select a minimal HMSOs set that guarantees complete diagnosability in each operating mode. We then discuss residual generation and the fault detection and isolation algorithm for hybrid systems.

5.4.1 Selecting a minimal HMSO set for FDI

The FDI algorithm assumes that the discrete variables values have been computed and the hybrid system mode is known. Therefore, we can remove the discrete variables from the list of unknown variables. Assume we generate r HMSOs for hybrid system diagnosis model, H_d, in mode m, $HMSO_m = \{HMSO_1^m, HMSO_2^m, \ldots, HMSO_r^m\}$, given its set of measurements $Y \cup Z$. Our goal is to design a greedy search algorithm that selects a minimal set of residuals $HMSO_m^* \subseteq HMSO_m$ in a way that makes all the system faults in this mode, F_m, structurally diagnosable. Algorithm 13 sorts the HMSOs by number of equations in the first setp (line 5). Because of the sorting step the greedy search selects HMSOs with the smallest number of equations. The smaller the number of equations that make up an HMSO, the smaller are the number of the system model parameters in this HMSO (this is a heuristic), and, therefore, the HSMO is likely to be more robust against model uncertainties. Moreover, the computational complexity of solving for unknown variables and

deriving residuals from HMSOs with fewer equations is likely to be smaller than HMSOs that include more equations (note, this is again a heuristic).

Algorithm 13 HMSO-Selection

input: set of selected HMSOs for the current mode $HMSO_m^*$
input: set of HMSOs in the current mode $HMSO_m$
input: set of undetectable faults UD
input: set of pair of faults that are not isolable UI
sort $HMSO_m$ by number of equations
if $UD = \emptyset$ and $UI = \emptyset$ **then**
    return $HMSO_m^*$ as a minimal HMSO set.
end if
$HMSO_{new}$= Find-HMSO($HMSO_m$,$HMSO_m^*$,UD,UI)
for each $f \in UD$ if $f \in HMSO_{new}$ **do**
    Let UD be $UD \setminus \{f\}$
end for
for each $(f_i, f_j) \in UD$ if $f_i \in HMSO_{new}$ and $f_j \notin HMSO_{new}$ **do**
    Let UI be $UI \setminus \{(f_i, f_j)\}$
end for
Let $HMSO_m^*$ be $HMSO_m^* \cup \{HMSO_{new}\}$.
Let $HMSO_m$ be $HMSO_m \setminus \{HMSO_{new}\}$.
HMSO-SELECTION($HMSO_m$, $HMSO_m^*$, UD, UI)

When the system transits to a mode m for the first time, the set of selected HMSOs for this mode, $HMSO_m^*$, is empty, and therefore, no fault is detectable or isolable in this mode. At each step, function Find-HMSO adds a HMSO candidate, $HMSO_{new}$, to $HMSO_m^*$ that makes at least a fault detectable, if not isolable from other faults that can occur in this mode. The algorithm keeps adding HMSOs to $HMSO_m^*$ till all the faults are detectable and isolable. The selected $HMSO_m^*$ are used to generate the set of residuals for this mode. The residuals for each visited mode will be saved to be used when the system returns to the mode.

As an example consider the case where discrete variables in the running example are $\Sigma = \{1,0,0,1,1,0\}$. This is the running example's mode number 38 when we number the system's 64 modes from 0 to 63. In this operating mode, the system equations are as

Algorithm 14 Find-HMSO

1: **input:** set of selected HMSOs for the current mode $HMSO_m^*$
2: **input:** set of HMSOs in the current mode $HMSO_m$
3: **input:** set of undetectable faults UD
4: **input:** set of pair of faults that are not isolable UI
5: **for each** $HMSO \in HMSO_m$ if ($\exists f \in HMSO$ and $f \in UD$) or ($\exists f_i \in HMSO$ and $\exists f_j \notin HMSO$ and $(f_i, f_j) \in UI$) **do**
6: return $HMSO$ as the selected HMSO
7: **end for**

follows.

$$
\begin{aligned}
e_1 &: \dot{p}_1 = \frac{1}{C_{T1} + f_1}(q_{in_1} - q_1) & e_{10} &: \dot{p}_4 = \frac{1}{C_{T4} + f_6}(q_3 - q_4) \\
e_2 &: p_1 = \int \dot{p}_1 \, dt & e_{11} &: q_4 = 0 \\
e_3 &: q_1 = \frac{p_1 - p_2}{R_{P1} + f_2} & e_{12} &: p_4 = \int \dot{p}_4 \, dt \\
e_4 &: \dot{p}_2 = \frac{1}{C_{T2} + f_3}(q_1 - q_2) & e_{13} &: q_{in_1} = y_1 \\
e_5 &: p_2 = \int \dot{p}_2 \, dt & e_{14} &: p_1 = y_2 \\
e_6 &: q_2 = 0 & e_{15} &: q_1 = y_3 \\
e_7 &: \dot{p}_3 = \frac{1}{C_{T3}}(q_{in_2} + q_2 - q_3) & e_{16} &: p_2 = y_4 \\
e_8 &: p_3 = \int \dot{p}_3 \, dt & e_{17} &: q_2 = y_5 \\
e_9 &: q_3 = \frac{p_3 - p_4}{R_{P3} + f_5} & e_{18} &: q_{in_2} = y_6 \\
& & e_{19} &: q_3 = y_7 \\
& & e_{20} &: p_4 = y_8
\end{aligned}
\tag{85}
$$

The set of system faults in this operating mode is $F_{38} = \{f_1, f_2, f_3, f_5, f_6\}$. Note that in this operation mode f_4 is not among the system faults. In fact, when $\sigma_2 = 0$ the valve flow $q_2 = 0$ independent of the resistance in the pipe, R_{R_2}. In this situations, a fault in R_{R_2} does not have any effect on the system dynamics. To detect f_4 the diagnoser should wait for a mode change in the hybrid system or adopt an active fault diagnosis approach [135].

Running the fault diagnosis toolbox, produces 47 HMSOs for the running example in this operating mode and, therefore, $HMSO_{38} = \{HMSO_1^{38}, HMSO_2^{38}, ..., HMSO_{47}^{38}\}$. Table 20 shows the selected HMSOs and their sets of equations and faults in mode 38. To implement

Table 20: Set of selected HMSOs for FDI in mode 1010

HMSOs	Set of equations	Set of faults
$HMSO_1^{38}$	$e_3, e_{14}, e_{15}, e_{16}$	f_2
$HMSO_2^{38}$	$e_{10}, e_{11}, e_{12}, e_{19}, e_{20}$	f_6
$HMSO_3^{38}$	$e_4, e_5, e_{15}, e_{16}, e_{17}$	f_3
$HMSO_4^{38}$	$e_1, e_2, e_{13}, e_{14}, e_{15}$	f_1
$HMSO_5^{38}$	$e_7, e_8, e_9, e_{17}, e_{18}, e_{19}, e_{20}$	f_5

the diagnosis approach for detecting and isolating the faults, we generate residuals in the manner shown next.

5.4.2 Generating residuals for hybrid systems

We developed Algorithm 13 to select a minimal set of HMSOs for fault detection and isolation. When the system operating mode is detected and, therefore, the discrete variables in the hybrid system are known, we can use the fault diagnosis toolbox to generate a residual from each HMSO. Consider the set of HMSOs in Table 20. The residual generated by fault diagnosis toolbox from each HMSO is presented as follows.

$$
\begin{aligned}
HMSO_1^{38} &: r_1 = y_3 - \frac{y_2 - y_4}{R_{P1}} \qquad HMSO_3^{38} : r_3 = y_4 - \frac{1}{C_{T_2}} \int y_3 - y_5 \\
HMSO_2^{38} &: r_2 = y_8 - \frac{1}{C_{T_4}} \int y_7 \qquad HMSO_4^{38} : r_4 = y_2 - \frac{1}{C_{T_1}} \int y_1 - y_3 \\
HMSO_5^{38} &: r_5 = y_8 + R_{P_3} y_7 - \frac{1}{C_{T_3}} \int y_6 + y_5 - y_7
\end{aligned}
\tag{86}
$$

To show how this set of five residuals is enough to detect and isolate all of the system faults in the current operation mode we present the residuals for detecting individual faults

Table 21: Selected residuals for FDI in mode 38.

	Detection	f_1	f_2	f_3	f_4	f_5	f_6
f_1	r_4	X	r_4	r_4	r_4	r_4	r_4
f_2	r_1	r_1	X	r_1	r_1	r_1	r_1
f_3	r_3	r_3	r_3	X	r_3	r_3	r_3
f_5	r_5	r_5	r_5	r_5	r_5	X	r_5
f_6	r_2	r_2	r_2	r_2	r_2	r_2	X

and isolating pairs faults in Table 21. The detection column lists the residuals that can be used to detect system faults, and the residuals in row f_i and column f_j can be applied to isolate fault f_i from fault f_j.

5.4.3 Designing fault diagnosers

As it is shown in Fig 32, the fault detection and isolation unit applies the generated residuals to detect and isolate system faults in each operation mode. To apply the residuals for FDI, we need to go beyond structural analysis, take into account parameter values, and also consider sensor noise and modeling uncertainties in the system. Residuals represent redundancies in the system equations, and they can form the basis for fault detection and isolation. Ideally, each residual should compute to a value of zero in the fault-free case, and residuals sensitive to a fault become nonzero when the fault occurs. In practice, due to model uncertainties and measurement noise, a residual may deviate from zero when no faults have occurred.

To address this problem, we apply a Z-test [18] to determine where the change in the residual value, r, is statistically significant. We consider the last N_2 residual values to compute the mean value of residual distribution (assumed to be a normal distribution):

$$\mu_r(k) = \frac{1}{N_2} \sum_{i=k-N_2+1}^{k} r(i). \tag{87}$$

The last N_1 samples (typically, $N_1 \gg N_2$) to compute the variance:

$$\sigma_r^2(k) = \frac{1}{k-N_1-1} \sum_{i=k-N_1+1}^{k} (r(i) - \mu_r(k))^2. \tag{88}$$

The confidence level for the Z-test, α, determines the bounds, z_-, and z_+, and, therefore, the sensitivity of the residuals.

$$P(z_- < (r(k) - \mu_r(k)) < z_+) = 1 - \alpha. \tag{89}$$

The Z-test is implemented as follows:

$$\begin{aligned} z_- < r(k) - \mu_r(k) < z_+ &\rightarrow \text{NF} \\ \text{Otherwise} &\rightarrow \text{Fault} \end{aligned} \tag{90}$$

5.5 Case Study

5.5.1 The RO system hybrid model

The RO system, shown in Figure 7, operates in three modes controlled by a three-way valve. In Chapter III we developed a robust FDI for the first operating mode (valve setting 1). In this chapter we apply our hybrid diagnosis method for mode detection and FDI in the RO system. In the first mode of operation, the water circulates in the longer loop. The accumulation of impurities in the membrane increases membrane resistance, R_{memb}, which decreases the output flow rate, q_{out}. After a specific period of time, the system switches to the secondary mode (valve setting 2). In this mode, the recirculation pump circulates the water in a smaller secondary loop, which increases its flow rate. As a result, the output flow rate increases compared to the primary loop. As clean water leaves the RO system, the concentration of brine, e_{Cbrine}, in the residual water increases. High concentration of brine leads to increases in R_{memb}, which decreases the system performance significantly. Again after a predetermined time interval the system switches to the purge mode (valve setting 3).

In this mode the recirculation pump is turned off, and concentrated brine is pushed out to the AES subsystem.

The RO system can be modeled as a hybrid system, with continuous state variables: q_{fp}, the volume flow rate generated by the feed pump, p_{tr}, the pressure of the fluid in the tubular reservoir, q_{rp}, the volume flow rate due to the recirculation pump, p_{memb} that represents the pressure of fluid at the membrane through which the clean water passes (but leaves the impurities behind), and two abstract variables, e_{Cbrine} and e_{Ck} that capture the dynamics of the impurities in the fluid, and discrete variables: σ_1 and σ_2 where $\sigma_1 = 1, \sigma_2 = 0$ means the system is in the first mode, $\sigma_1 = 0, \sigma_2 = 1$ means the system is in the second mode, and $\sigma_1 = \sigma_2 = 0$ means the system is in the third mode. In the original design and experiments conducted with a prototype RO system, operating times in each mode were fixed by the system operators. However, to demonstrate our mode detection algorithm, we assume the transition times are unknown, and apply our algorithm to detect mode transitions.

The feed pump pushes the partially purified water from the BWP into the main loop of the RO system at a nominal pressure p_{fp}. The rate of change of the volume flow rate, q_{fp} is given by: $\dot{q}_{fp} = \frac{\Delta p_{fp}}{I_{fp}}$, where Δp_{fp} is drop in pressure of the fluid across the feed pump and I_{fp} represents the inertia of the rotating elements of the feed pump. Taking into account the pump internal resistance to flow, R_{fp}, and the efficiency decrease in the feed pump, which is modeled by a multiplicative factor f_f, the pressure drop can be computed, $\Delta p_{fp} = p_{fp}(1 - f_f) - R_{fp} q_{fp} - p_{tr}$ and the first equation of the RO system, e_{RO1}, can be derived as:

$$e_{RO1} : \dot{q}_{fp} = \frac{1}{I_{fp}}(-R_{fp} q_{fp} - p_{tr} + p_{fp}(1 - f_f)). \tag{91}$$

Note that this equation is independent of the system operating mode.

The tubular reservoir with capacity value, C_{tr}, acts as a storage capacity that helps the system to keep water circulation rate steady. The net volume flow rate to the tubular reservoir, q_{tr}, is equal to the algebraic sum of the volume flow rates into and out of the tubular reservoir. The flow in is q_{fp} plus the flow from the membrane module. This flow

rate can be computed as the pressure difference between the membrane and the tubular reservoir, $p_{memb} - p_{tr}$, over the resistance of the pipe from the membrane module to the tubular reservoir in the long loop, R_{return_l}, in the first mode. In the second and third modes there is no flow from the membrane toward the tubular reservoir. The output volume flow rate from the tubular reservoir is equal to the recirculation pump volume flow rate, q_{rp} in the first and third modes and equal to q_{rp} minus the pressure difference between the membrane and the tubular reservoir, $p_{memb} - p_{tr}$, over the resistance of the pipe from the membrane module to the tubular reservoir in the short loop, R_{return_s}. Using these the net volume flow rate to the tubular reservoir can be computed as $q_{tr} = q_{fp} + \sigma_1 \frac{p_{memb}-p_{tr}}{R_{return_l}} - q_{rp} + \sigma_2 \frac{p_{memb}-p_{tr}}{R_{return_s}}$. Therefore, the second equation can be derived as:

$$e_{RO2} : \dot{p}_{tr} = \frac{1}{C_{tr}}(q_{fp} + \sigma_1 \frac{p_{memb} - p_{tr}}{R_{return_l}} - q_{rp} + \sigma_2 \frac{p_{memb} - p_{tr}}{R_{return_s}}). \tag{92}$$

The recirculation pump boosts the liquid pressure to p_{rp} in the first and second modes. In these modes, the rate of change of pump's fluid flow rate, $\dot{q}_{rp}$, is given by ($\dot{q}_{rp} = \frac{\Delta p_{rp}}{I_{rp}}$), where Δp_{rp} represents drop in the fluid pressure inside the pump and I_{rp} represents the inertia of the rotating elements of the pump. The pump's internal resistance is represented as R_{rp}. The efficiency decrease in the recirculation pump, f_r, is the second fault parameter in the RO system. The pressure at the pump output can be computed as a function of the membrane module pressure, p_{memb} and the pressure drop in the pipe from the pump to the membrane module, $R_{forward} q_{rp}$, where $R_{forward}$ represents resistance of the pipe from the recirculation pump to the membrane. From these components, the third state equation in the first two modes is derived as:

$$\dot{q}_{rp} = \frac{1}{I_{rp}}(-R_{rp} q_{rp} - R_{forward} q_{rp} - p_{memb} + p_{rp}(1 - f_r)). \tag{93}$$

In the third mode, the recirculation pump is turned off, therefore, q_{rp} is not a state variable

and can be computed as $q_{rp} = \frac{p_{tr} - p_{memb}}{R_{forward}}$. We can write q_{rp} in general as:

$$e_{RO3} : q_{rp} = \frac{\sigma_1 + \sigma_2}{I_{rp}} \int (-R_{rp} q_{rp} - R_{forward} q_{rp} - p_{memb} + p_{rp}(1 - f_r)) + (1 - \sigma_1 - \sigma_2) \frac{p_{tr} - p_{memb}}{R_{forward}} \tag{94}$$

The membrane is a key component for removing particulate matter from the water in the RO system. The purified water that comes out of the membrane is fed to the Post Processing System, and the remaining water recirculates in a RO loop or goes to the AES based on the operation mode. As more and more water passes through the membrane, the water remaining in the loop has an increased concentration of impurities. At the same time particulate matter that collects on the membrane, increases its resistance to flow. The membrane chamber can be modeled as a combination of a capacity, C_{memb}, and a resistance, R_{memb}. The rate of membrane pressure variation, $\dot{p}_{memb}$, is given by ($\dot{p}_{memb} = \frac{q_{memb}}{C_{memb}}$), where q_{memb} is the net volume flow rate to the membrane.

In previous work, Carl et al. [27] empirically derived the dynamic value for the membrane resistance, $R_{memb} = 0.202(4.137 * 10^{11}(\frac{e_{Ck} - 12000}{165} + 29))$. Note that R_{memb} increases as the impurities in the water, and, therefore, the water conductivity, e_{Ck}, increases. The membrane clogging factor f_m is the third fault in the system, implying that its resistance is higher than nominal, i.e., $R_{memb}(1 + f_m)$. The net volume flow rate to the membrane, q_{memb}, is an algebraic sum of the input volume flow rate from the recirculation pump, q_{rp}, and the output volume flow rates to the post processing system, $q_{out} = \frac{p_{memb}}{R_{memb}(1+f_m)}$, and the return volume flow rates. The return flow rate to the flow is equal to $\frac{p_{pmemb} - p_{tr}}{R_{return_l}}$ in the first mode, equal to $\frac{p_{pmemb} - p_{tr}}{R_{return_s}}$ in the second mode, and equal to $\frac{p_{pmemb}}{R_{return_{ASE}}}$, where $R_{return_{ASE}}$ is the resistance of the pipe from the membrane to the ASE. Using this the fourth equation is derived,

$$
\begin{aligned}
e_{RO4} : \dot{p}_{memb} = \frac{1}{C_{memb}}(q_{rp} - \frac{p_{memb}}{R_{memb}(1+f_m)} \\
-\sigma_1 \frac{p_{memb} - p_{tr}}{R_{return_l}} - \sigma_2 \frac{p_{memb} - p_{tr}}{R_{return_s}} \\
-(1-\sigma_1-\sigma_2)\frac{p_{pmemb}}{R_{return_{ASE}}}).
\end{aligned} \tag{95}
$$

To complete the model, the conductivity of the fluid is represented as a state variable, making the assumption that the conductivity of the water increases every cycle through the flow loop, with the increase being proportional to the flow of liquid out of the membrane. This generates the two last state equations:

$$
\begin{aligned}
e_{RO5} : \dot{e}_{Cbrine} = \frac{1}{1.667 * 10^{-8} C_{brine}}(\sigma_1 \frac{p_{memb} - p_{tr}}{R_{return_l}} \\
+\sigma_2 \frac{p_{memb} - p_{tr}}{R_{return_s}} + (1-\sigma_1-\sigma_2)\frac{p_{memb}}{R_{return_{ASE}}}) \\
e_{RO6} : \dot{e}_{Ck} = \frac{q_{rp}}{C_k}(6e_{Cbrine} + 0.1)/(1.667 * 10^{-8}),
\end{aligned} \tag{96}
$$

where C_{brine} and C_k are conductivity parameters and are represented in Table 1. The system inputs are the feed pump pressure, p_{fp} and the recirculation pump pressure, p_{rp}. More details of the RO modeling scheme are presented in [17, 27].

There are five sensors in the system. These sensors measure the following variables in the system.

$$
\begin{aligned}
e_{RO7} : p_{tr} = y_1 \qquad & \qquad e_{RO10} : e_{Cbrine} = y_4 \\
e_{RO8} : p_{mem} = y_2 \qquad & \qquad e_{RO11} : e_{Ck} = y_5 \\
e_{RO9} : q_{fp} = y_3 \qquad &
\end{aligned} \tag{97}
$$

The system inputs, p_{fp} and p_{rp}, are assumed to be known in this case study.

$$
e_{RO12} : p_{fp} = u_1 \qquad e_{RO13} : p_{rp} = u_2 \tag{98}
$$

To complete the set of constrains, we present the differential constrains and the dynamic value for the membrane resistance as follows.

$$\begin{aligned} e_{RO14} : q_{fp} &= \int \dot{q}_{fp} & e_{RO17} : p_{memb} &= \int \dot{p}_{memb} \\ e_{RO15} : p_{tr} &= \int \dot{p}_{tr} & e_{RO18} : e_{Cbrine} &= \int \dot{e}_{Cbrine} \\ e_{RO16} : q_{rp} &= \int \dot{q}_{rp} & e_{RO19} : e_{Ck} &= \int \dot{e}_{Ck} \end{aligned} \tag{99}$$

$$e_{RO20} : R_{memb} = 0.202(4.137 * 10^{11}(\frac{e_{Ck} - 12000}{165} + 29)). \tag{100}$$

The RO system's parameters and input signals in this case study are presented in Table 1. In the experiments, for the case study, we assume 1% uncertainty in the system parameters and Gaussian measurement noise, with variance = 0.01 $\times$ mean value of the signal.

5.5.2 Mode detection for the RO system

Employing Algorithm 12 we extract $MSD_{RO1} = (E_{RO1}, T_{RO1}, X_{RO1}, \Sigma_{RO1}, F_{RO1})$, where $E_{RO1} = \{e_{RO2}, e_{RO5}, e_{RO6}, e_{RO7}, e_{RO8}, e_{RO9}, e_{RO10}, e_{RO11}, e_{RO15}, e_{RO18}, e_{RO19}\}$, $T_{RO1} = \{\}$, $X_{RO1} = \{p_{mem}, p_{tr}, \dot{p}_{tr}, e_{Cbrine}, \dot{e}_{Cbrine}, e_{Ck}, \dot{e}_{Ck}, q_{rp}, q_{fp}\}$, $\Sigma_{RO1} = \{\sigma_1, \sigma_2\}$, and $F_{RO1} = \{\}$, to detect each discrete variable in the RO system.

By substituting the known variables in the set of determined equations in MSD_{RO1}, and after some algebraic manipulations, we reach to the following linear equation to solve for σ_1 and σ_2 :

$$\begin{aligned} a_1\sigma_1 + b_1\sigma_2 &= c_1 \\ a_2\sigma_1 + b_2\sigma_2 &= c_2, \end{aligned} \tag{101}$$

where $a_1 = \frac{y_2 - y_1}{R_{return_l}}$, $b_1 = \frac{y_2 - y_1}{R_{return_s}}$, $c_1 = C_{tr}\dot{y}_1 - y_3 + \frac{\dot{y}_5 C_k 1.667 * 10^{-8}}{6e_{Cbrine} + 0.1}$, $a_2 = \frac{y_2 - y_1}{R_{return_l}} - \frac{y_2}{R_{return_{ASE}}}$, $b_2 = \frac{y_2 - y_1}{R_{return_s}} - \frac{y_2}{R_{return_{ASE}}}$, $c_2 = (1.667 * 10^{-8} C_{brine})\dot{y}_4 - \frac{y_2}{R_{return_{ASE}}}$. Therefore, σ_1 and σ_2 can

be computed as

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \end{bmatrix} = \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \end{bmatrix}^{-1} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \tag{102}$$

To show the performance of our proposed mode detection approach, we perform a simulation study where the RO system operates for $1000s$ and switches mode every $33s$. The system starts from mode 1, switches to modes 2 after $33s$, switches to mode 3 after $66s$, and switches back to mode 1 at $t = 100s$. Figure 35 shows that equation (102) can perfectly estimate the discrete variables, and therefore, the operating mode of the system. To overcome the effects of measurement noise we applied a simple low pass filter.

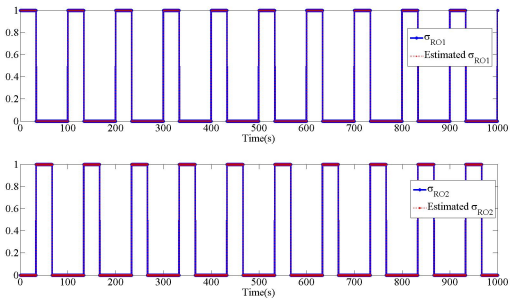

Figure 35: Mode detection in the RO system in no fault (NF) scenario.

Note that Algorithm 12 guarantees that mode detectability is independent of the system faults. However, to show a possible fault cannot affect the mode detection performance, we consider another experiment, where an abrupt fault $f_f = 0.5$ occurs at $t = 510s$. Figure 36 shows that f_f does not affect mode detection performance.

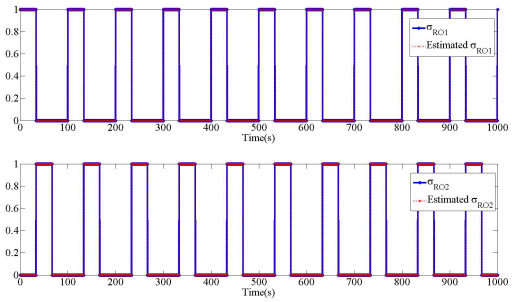

Figure 36: Mode detection in the RO system in the presence of f_m.

5.5.3 Fault Detection and Isolation in the RO system

As it is shown in Figure 32 the proposed FDI approach for hybrid systems has the following steps; 1) mode detection, 2) HMSO generation for the detected mode, 3) selecting a minimal set of HMSOs for fault detection, 4) generating residuals from the selected HMSOs, 5) applying the generated residuals for FDI. We developed algorithms for mode detection, and selecting a minimal set of HMSOs (steps 1 and 3) in this chapter, for HMSO generation and residual generation (steps 2 and 4) we use the fault diagnosis toolbox [68]. We discussed step 5 in subsection 5.4.3.

Table 22: Set of selected HMSOs for FDI in the RO system

HMSOs	Set of equations	Set of faults	Mode
$HMSO_{RO1_1}$	$e_{RO1}, e_{RO7}, e_{RO9}, e_{RO12}, e_{RO14}$	f_f	1
$HMSO_{RO1_2}$	$e_{RO3}, e_{RO6}, e_{RO8}, e_{RO10}, e_{RO11}, e_{RO13}, e_{RO16}, e_{RO19}$	f_r	1
$HMSO_{RO1_3}$	$e_{RO4}, e_{RO6}, e_{RO7}, e_{RO8}, e_{RO10}, e_{RO11}, e_{RO17}, e_{RO19}, e_{RO20}$	f_m	1
$HMSO_{RO2_1}$	$e_{RO1}, e_{RO7}, e_{RO9}, e_{RO12}, e_{RO14}$	f_f	2
$HMSO_{RO2_2}$	$e_{RO3}, e_{RO6}, e_{RO8}, e_{RO10}, e_{RO11}, e_{RO13}, e_{RO16}, e_{RO19}$	f_r	2
$HMSO_{RO2_3}$	$e_{RO4}, e_{RO6}, e_{RO7}, e_{RO8}, e_{RO10}, e_{RO11}, e_{RO17}, e_{RO19}, e_{RO20}$	f_m	2
$HMSO_{RO3_1}$	$e_{RO1}, e_{RO7}, e_{RO9}, e_{RO12}, e_{RO14}$	f_f	3
$HMSO_{RO3_2}$	$e_{RO3}, e_{RO4}, e_{RO7}, e_{RO8}, e_{RO11}, e_{RO17}, e_{RO20}$	f_m	3

In the first mode this system has three possible faults $F = \{f_f, f_r, f_m\}$. In this mode, the fault diagnosis toolbox generates 98 HMSOs and Algorithm 13 selects 3 HMSOs,

Table 23: Selected Residuals for Fault Detection and Isolation

Selected residual	First mode	Second mode	Third mode
Detecting f_f	r_{RO1_1}	r_{RO2_1}	r_{RO3_1}
Detecting f_r	r_{RO1_2}	r_{RO2_2}	–
Detecting f_m	r_{RO1_3}	r_{RO2_3}	r_{RO3_2}
Isolating f_f from f_r	r_{RO1_1}	r_{RO2_1}	–
Isolating f_f from f_m	r_{RO1_1}	r_{RO2_1}	r_{RO3_1}
Isolating f_r from f_f	r_{RO1_2}	r_{RO2_2}	–
Isolating f_r from f_m	r_{RO1_2}	r_{RO2_2}	–
Isolating f_m from f_f	r_{RO1_3}	r_{RO2_3}	r_{RO3_2}
Isolating f_m from f_r	r_{RO1_3}	r_{RO2_3}	–

$HMSO_{RO1_1}$, $HMSO_{RO1_2}$, and $HMSO_{RO1_3}$, as a minimal set of HMSOs that can detect and isolate the faults. The fault diagnosis toolbox generates a residual from each HMSO (r_{RO1_1}, r_{RO1_2}, and r_{RO1_3}). The RO system has three possible faults, $F = \{f_f, f_r, f_m\}$, in mode 2. The fault diagnosis toolbox generates 84 HMOSs and Algorithm 13 selects 3 HMSOs, $HMSO_{RO2_1}$, $HMSO_{RO2_2}$, and $HMSO_{RO2_3}$, as a minimal set of HMSOs that can detect and isolate the faults in the second mode. The fault diagnosis toolbox generates 3 residuals, r_{RO2_1}, r_{RO2_2}, and r_{RO2_3}, for this mode. In the third mode the RO system only has two possible faults, $F = \{f_f, f_m\}$ because the recirculation pump is turned off. In this mode, the fault diagnosis toolbox generates 59 HMSOs and Algorithm 13 selects 2 HMSOs, $HMSO_{RO3_1}$, and $HMSO_{RO3_2}$, as a minimal set of HMSOs that can detect and isolate the faults. Table 22 represents the selected HMSOs and their associated faults in each mode. The fault diagnosis toolbox generates r_{RO3_1}, and r_{RO3_2} for the third mode. The set of residuals for each operating mode is available on GitHub[1].

To detect and isolate the RO system's faults we need to use a new set of residuals in each operating mode. Table 23 shows the possible residuals for FDI in each operating mode. To design a fault diagnoser for hybrid systems, we need to define hybrid residuals. For example, to detect f_f we have to use r_{RO1_2} or r_{RO1_3} when the system is in the first operation

[1] https://rosystemresiduals.github.io/

mode, and we have to use r_{RO2_2} or r_{RO2_3} when the system is in the second operation mode, and r_{RO3_2} is the only choice when the system is in the third mode. Therefore, a possible candidate to detect f_f is

$$r_{f_f} = \begin{cases} r_{RO1_2} & if \quad \sigma_1 = 1 \\ r_{RO2_2} & if \quad \sigma_2 = 1 \\ r_{RO3_2} & otherwise. \end{cases} \tag{103}$$

To isolate f_f from f_r and f_m we use the following residuals.

$$r_{f_f|f_r} = \begin{cases} r_{RO1_2} & if \quad \sigma_1 = 1 \\ r_{RO2_2} & if \quad \sigma_2 = 1 \end{cases} \tag{104}$$

$$r_{f_f|f_m} = \begin{cases} r_{RO1_3} & if \quad \sigma_1 = 1 \\ r_{RO2_3} & if \quad \sigma_2 = 1 \\ r_{RO3_2} & otherwise. \end{cases} \tag{105}$$

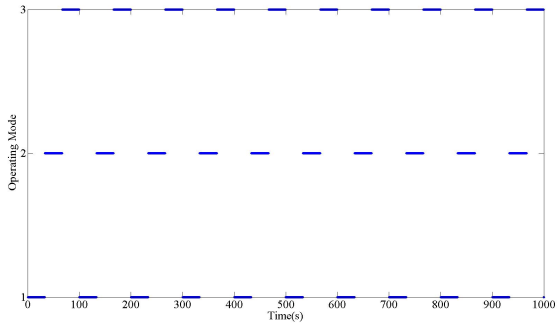

Figure 37: Operating modes in the RO system.

In this case study we assume that the system goes through the cycle shown in Figure 35. Figure 37 shows the operating modes of the RO system. To show FDI performance

we consider an abrupt efficiency decrease in the feed pump $f_f = 0.5$ occurs at $t = 510s$. Figure 38 shows the state variables in this case study.

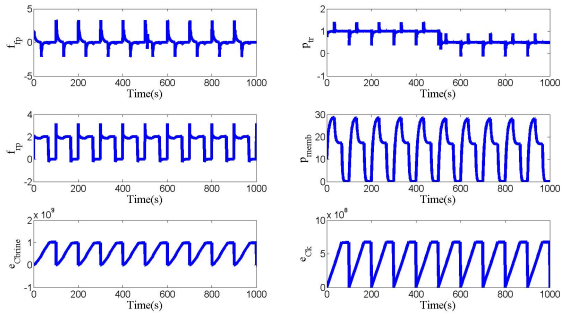

Figure 38: Continuous state variables in the RO system.

Figure 39 shows the hybrid residuals, r_{f_f}, $r_{f_f|f_r}$ and $r_{f_f|f_m}$ can successfully detect and isolate f_f. For all of our detection and isolation experiments, the confidence level for the Z-test is 95%.

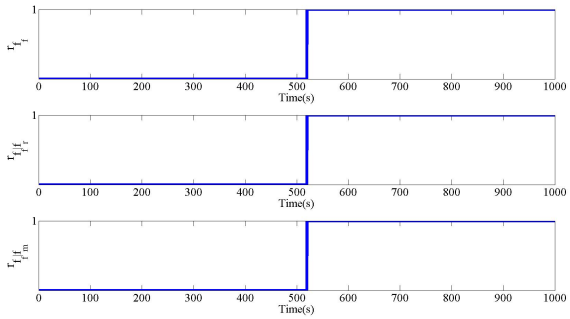

Figure 39: f_f detection and isolation.

5.6 Summary and Conclusions

A new approach to the problem of fault detection and isolation in hybrid systems is presented in this chapter. Our proposed approach consists of two algorithms; 1- mode detection, and, 2- fault detection and isolation in each operating mode. The contribution of

this work is that we do not need to pre-compile the MSOs and residual for every possible mode of the hybrid system, which can be computationally intractable. Therefore, our algorithm does not have to pre-enumerate all the possible modes, which is exponential in number of discrete variables in the model. Instead, our approach updates the diagnoser when the system switches to a new operating mode. These make the approach feasible for fault diagnosis in complex systems, where there are large number of modes.

Unlike previous work [102], where we formulated HMSO selection as an exponential problem in term of the number of HMSOs in each mode, Algorithm 13 selects a minimal set of residuals with $O(l_{fm}^2 r_m)$ time complexity, where l_{fm} is the number of faults in mode m, and r_m is the number of HMSOs in that mode. Note that in this work we have adopted a greedy search approach and, therefore, our algorithm does not guarantee the minimum number of HMSOs for each mode. However, in the running example and in the case study the algorithm selected the minimum number of HMSOs and in general, we expect the number of selected HMSOs to be close to the optimal solution.

It is expected that solving mode detection and fault detection and isolation as two interrelated but separated problems requires more redundancies and therefore, more measurements, compared to the approaches that address both problems simultaneously such as [137] and [65]. This is because we have to extract sets of just determined equations for mode identification which are independent of the system faults. However, the recent developments in manufacturing inexpensive and efficient sensors and processors make our approach feasible for complex, modern systems.

In this work, we used mixed causality (integral and derivative) to solve for discrete variables for mode detection, and to generate residuals for fault detection and isolation. Using mixed causality improves mode detection and FDI performances. Limiting our approach to integral causality can reduce the number of solvable equations and this may make a set of discrete variables or a set of system faults undetectable [67]. However, in the presence of measurement noise, derivative computation can be error-prone. In that case, restricting

mode detection and residual generation to integral causality naturally increases the robustness of the approach.

To handle issues of robustness in a more general way, we have to consider the effect of model uncertainty and sensor noise in mode detection and FDI. In Chapter III, we used sensitivity analysis to define a detectability ratio measure and a isolability ratio measure to quantify the performance of a residual in fault detection and isolation. In future work, we will extend the previous work in robust residual selection to hybrid systems. Toward this end, we will develop a quantitative measure for robust mode detection and apply detectability ratio and isolability ratio measures for residual selection for robust FDI in each operating mode.

CHAPTER VI

BACKGROUND ON DATA-DRIVEN DIAGNOSIS AND ANOMALY DETECTION

Traditional approaches to fault detection typically rely on a model that defines nominal behavior of a system, or on human expertise that characterizes the parameters or thresholds that separate nominal from anomalous behavior. However, for complex systems it is often expensive to generate system models. On the other hand, models generated by human experts, who are required to provide a fairly exhaustive set of relations between faults and observed measurements, are likely to be incomplete. In the real world these models are not always available, and are often incomplete, and sometimes erroneous. Moreover, it is hard to maintain the accuracy of these models during a system's life-cycle. This could lead to misclassifying faults or missing faulty behavior. In some situations, faulty and anomalous situations may be unknown because of a lack of sufficient experience in some of the operational regions of the system. In such situations, data-driven approaches for diagnosis become the key to protecting system safety and integrity [124].

Data-driven diagnosis algorithms aim to detect and isolate system faults by operating exclusively on measured data without detailed system knowledge. Data-driven diagnosis approaches typically detect deviations in a process time series measurement as a fault and then use a classifying technique for fault isolation [39]. This approach performs well for systems with smooth nominal trajectories but for hybrid systems or nonlinear systems with stiff behaviors a discrete change in a measurement may not represent a fault. A more general approach is to find patterns in operational data that do not conform to expected systems behavior. These patterns, typically called anomalies or outliers, are represented by single data points or a small group of data points that appear to be sufficiently different from the rest of the data that make up the operational behaviors of the system [28]. For

dynamic systems, anomalies can represent unexpected changes in the systems behavior, and correspond to faults in system operation and component failures.

Anomaly detection is a key issue in securing safe and reliable operation of advanced complex systems such as spacecraft [16], aircraft [115, 125] and power systems [15]. Several surveys and review articles have categorized data-driven anomaly detection approaches based on different aspects of the problem [28, 84, 145]. A common approach is to divide data-driven anomaly detection approaches into three main groups: 1) supervised, 2) semi-supervised, and 3) unsupervised. The supervised anomaly detection approaches assume the training data is available for both normal and anomalous operations. The semi-supervised methods only need normal data for the training and the unsupervised methods do not use any training data.

Supervised anomaly detection approaches are based on classification methods. A typical approach to develop a supervised anomaly detector is to consider normal operation modes and anomalies as different classes and design a classifier that differentiates between the two sets of data. The classifier can then label new data points as nominal or anomalous. Ma et al [123] compared the performances of several classifiers such as linear discriminant analysis (LDA), naive Bayes (NB), and decision tree (DT) for anomaly detection in uncertain data streams. The learning step in the classification approaches is done off-line and, therefore, the on-line implementation is computationally efficient. However, having access to labeled data specially for anomalous operations is expensive and difficult. Abe et al [1] proposed to artificially generate anomalous data to address this problem. However, in the real world, new and unknown types of anomalous behaviors may arise. Therefore, we have to consider semi-supervised, and unsupervised methods for anomaly detection.

6.1 Semi-supervised Anomaly Detection

Semi-supervised anomaly detection approaches are more general than supervised techniques; the only labeled data they need corresponds to the data that represents nominal

behavior. Then they apply similarity-based measures to label data points that are not close to the nominal regions as anomalies [28]. These methods are also called novelty detection methods in the literature [95]. A common approach in semi-supervised anomaly detection is to apply a classification algorithm, such as neural networks to learn the normal operation class or the multiple normal operation classes using available nominal data from a system. In the implementation phase, each new data point is provided as an input to the classifier. If the classifier accepts the input, it is normal and if the classifier rejects an input, it is labeled as an anomaly.

Support vector machines and other kernel based learning methods can be used to learn complex regions [158]. Ratsch et al [152] applied a one-class support vector machine (I-SVM) to learn the boundaries of region that contains the normal training data objects by finding a set of separating hyperplanes. They formulated the problem as a constraint quadratic optimization problem. For each test instance, if the test instance falls within the learnt region, it is considered as normal, else it is declared as an anomaly. Barbara et al [8] applied a naive Bayes classifier for novel network intrusion detection. They considered an extra class in addition to the classes in the training data to represent new attacks. The naive Bayes network estimates the posterior probability of observing class label data, given a test data instances. The class label with largest posterior is chosen as the predicted class for the given test instance. The likelihood of observing the test instance given a class, and the classes prior probabilities, are estimated from the training data set.

Several neural networks based methods have been proposed for semi-supervised anomaly detection [128]. Hawkins et al. [80] applied replicator neural networks (RNN) for anomaly detection on a large multivariate dataset. The RNN is a three layers perceptron neural-networks with the same number of input and output neurons. The number of neurons in the input and output layers is equal to the number of features in the dataset. The RNN is trained with normal data in a way that reproduces the input points at the output layer with the minimum reconstruction error. The hidden layer has a smaller number of nodes than

the input and output layers and is designed to extract normal operating features. In the implementation stage, any test point that has large reconstruction errors is considered as an anomaly. Wulsin et el [183] applied deep belief nets (DBNs) which is a type of multi-layer generative neural networks with the ability to model high-dimensional data, for anomaly detection. Like [128], they used reconstruction error for each test object as the anomaly score. DBNs can learn higher-level features which can improve classification accuracy. However, their loss function is non-convex, therefore, there is no guarantee that the global minimum is achieved [49].

The computational complexity of semi-supervised anomaly detection techniques depends on the classification algorithm used, however, like the supervised methods, the testing phase is usually very fast since it uses a pre-computed model [28]. However, several factors can make semi-supervised anomaly detection methods challenging. Defining every possible region of nominal behavior for a complex system that has many operating modes may be difficult and sometimes, computationally intractable. Furthermore, as systems operate under different environmental and operational conditions, and as the systems age, their nominal behaviors may keep drifting and evolving, and current nominal behavior may not be indicative of future nominal behaviors. And last, training data labeled as nominal for every operating mode of the system may be hard to come by. Further, even if the data is labeled, noise and corruption may distort the differences between nominal and anomalous behavior. Therefore, unsupervised anomaly detection may be the only practical choice in many real world applications.

6.2 Unsupervised Anomaly Detection

There are several techniques that can be used for anomaly detection in unlabeled data. For example, histogram based approaches use a simple non-parametric statistical technique for unsupervised anomaly detection. To use histogram for unsupervised anomaly detection in an univariate dataset, we can build a histogram based on the different values taken by

that feature and assign an anomaly score to each object based on the height (frequency) of the bin in which it falls. For multivariate data, a basic technique is to construct feature-wise histograms. The anomaly score for each feature of each object can be calculated as the height of the bin that contains the feature value. The per-feature anomaly scores are aggregated to obtain an overall anomaly score for the object. The size of bins is the key parameter in histogram based anomaly detection. If the bins are small, many normal test instances will fall in empty or rare bins, resulting in high false alarm rates (FAR). If the bins are large, many outliers will fall in frequent bins, resulting in high missed alarm rates (MAR). Thus an important challenge is to determine an optimal size of the bins to achieve low FAR and low MAR [28]. This becomes more challenging for high dimension data. More sophisticated unsupervised anomaly detection approaches, such as information theory techniques, and clustering based methods can be used to address this problem.

6.2.1 Information theory techniques

Information theory techniques can be applied to unsupervised anomaly detection. These techniques assume anomalies in data induce irregularities in the information content of the data set and information theoretic measures can be used to detect the anomalies. For example, Threepak and Watcharapupong [170] use the fact that entropy level of anomaly requests are usually higher than entropy level of legitimate behavior requests on the web and declare all the request strings with relative entropy values more than 2σ away from the average relative entropy as high-risk requests. Lee and Xiang [111] propose several information-theoretic measures for anomaly detection:

Entropy: For a dataset X where each data item belongs to a class $x \in C_x$, the entropy of X relative to this $|C_x|$-wised classification is defined as:

$$H(X) = \sum_{x \in C_x} P(x) log \frac{1}{P(x)} \tag{106}$$

where $P(x)$ is the probability of $x \in X$. The entropy value is smaller when the data is more

redundant. For example, if all data items are identical, the entropy is zero. Therefore, entropy can be used as a measure of regularity for anomaly detection.

Conditional entropy: The conditional entropy of X given Y is defined as:

$$H(X|Y) = \sum_{x,y \in C_x, C_y} P(x,y) log \frac{1}{P(x|y)} \tag{107}$$

where $P(x,y)$ is the join's probability of $x \in X$ and $y \in Y$ and $P(x|y)$ is the conditional probability of x given y. For anomaly detection, we can use conditional entropy as a measure of regularity of sequential dependencies. When the conditional entropy takes on smaller values, the data is more sequentially dependent and, therefore, more predictable.

Arackaparambil et al [4] argue that monitoring conditional entropy, which presents the dependency of one feature on another, makes the job of masking the effects of attack on network traffic harder and therefore, is more reliable than using entropy in attack detection. Attackers typically mask their attacks on network traffic by mimicking the normal distribution of traffic features in the packets they introduce. Maintaining dependencies between any pair of features is more challenging than just maintaining the distribution of features independently. Therefore, monitoring conditional entropy makes the attacker's job harder and improves the reliability of the monitoring approach.

Relative entropy: The relative entropy between two probability distributions $p(x)$ and $q(x)$ defined over the same $x \in C_x$ is defined as

$$relentropy(p|q) = \sum_{x \in C_x} p(x) log \frac{p(x)}{q(x)} \tag{108}$$

Relative entropy measures the distance of the regularities between two datasets. For example, if the two dataset have the same distribution, $p = q$, then the relative entropy is zero, indicating that the two datasets have the same regularity.

Relative conditional entropy: The relative conditional entropy between two conditional probability distributions $p(x|y)$ and $q(x|y)$ defined over the same $x \in C_x$ and $y \in C_y$ is defined

as

$$relconentropy(p|q) = \sum_{x \in C_x} p(x,y) log \frac{p(x|y)}{q(x|y)} \tag{109}$$

The relative conditional entropy represents sequential dependencies between distributions. The last measure is information gain.

Information gain: The information gain of a feature f on dataset X is

$$Gain(X, f) = H(X) - \sum_{v \in Value(f)} \frac{|X_v|}{|X|} H(X_v) \tag{110}$$

where $Value(f)$ is all the possible values of feature f, and X_v is a subset of X where f is equal to v. This measure can be used to evaluate the features in anomaly detection. Ham and Choi [77] used information gain to rank the features for malware detection.

Other complexity measures such as Kolmogorov complexity and Complexity-invariant distance have been used for anomaly detection. Keogh et al [101] developed compression-based dissimilarity measure (CDM) motivated by Kolmogorov complexity and used a divide and conquer algorithm to find anomalous section of data with the least similarity to the global sequence.

6.2.2 Clustering methods

Clustering is a process of partitioning a set of objects into clusters such that objects in the same cluster are more similar to each other than objects in different clusters according to some defined criteria [126]. Li et al [115] used a density-based clustering approach to detect anomalous flights based on onboard-recorded flight data. In previous work [16] we used a hierarchical clustering to identify anomalies in a spacecraft telemetry data. This section reviews three main clustering methods for anomaly detection.

K-means clustering: k-means are partitioning clustering algorithms. Partitioning clustering algorithms are based on specifying an initial number of groups, and iteratively reallocating objects among groups till convergence. The k-means algorithm assigns each object

to the cluster with the nearest centroid. The centroid is the average of all the objects in the cluster. The k-means algorithm can be expressed as the following optimization problem:

$$\begin{aligned} &\text{Find} \quad U\&Z \\ &\text{s.t.} \\ &\min \quad P(U,Z) = \sum_{l=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{m} u_{i,l}\, d(x_{i,j}, z_{l,j}) \\ &\qquad \sum_{l=1}^{k} u_{i,l} = 1, 0 \le i \le n, \qquad\qquad u_{i,l} \in \{0,1\} \end{aligned} \tag{111}$$

where U is an $n \times k$ partition matrix, Z is a set of k vectors representing the centroids of the k clusters, n is the number of objects, and m is the number of features in the dataset. $u_{i,l} \in U$ is a binary variable that is equal to 1 when object i belongs to cluster l, $d(x_{i,j}, z_{l,j})$ represents the distance of feature j of object i from the center of cluster l, $z_{l,j} \in Z$.

Several algorithms have been developed to search for the optimal partition of the k-means clustering in the literature [87]. Forgy [60] proposed Algorithm 15 to solve problem

Algorithm 15 K-means

```
input: O, k
output: C
generate k random centroids Z
assign each object o_i ∈ O to the cluster with the closest center
while no movement of an object has occurred do
    compute the centroids with the current arrangement of objects
    assign each object o_i ∈ O to its closest center
end while
```

(111). The running time for this algorithm is $O(nkd_i)$, where d_i is the number of iterations in the optimization problem. Algorithm 15 is easy to implement and computationally efficient. However, the drawback of this algorithm is that different initial centroids might lead to different local optima of the clustering results.

K-Means clustering is computationally efficient, and therefore, easy to implement for

large datasets. Moreover, the algorithm is intuitive and easy to understand. However, K-means clustering requires the number of cluster as an input to the algorithm and has no notion of outliers. Therefore, each object has to be assigned to one of the clusters even if it does not belong to any of them. This can degrade anomaly detection performance by assigning few outliers to each normal cluster. In that situation, the outliers pull the normal cluster centroid towards them, and make it harder to detect anomalies.

Density-based clustering algorithms consider a region in which the density of data objects exceeds a specified threshold a cluster. Low density regions represent clusters of noise or clusters of outliers. These methods can be used for anomaly detection by assuming normal data instances belong to clusters in the data with high density, while anomalies belong to the low density regions. Density-based methods can discover arbitrary-shaped clusters. Moreover, they have advantages in dealing with large datasets with uneven clusters and noise [126]. In this section, we review two common density-based algorithm, density-based spatial clustering of applications with noise (DBSCAN) [54] and shared nearest neighbor (SNN) [53].

DBSCAN considers an object o a core object if there are at least $MinObj$ objects within its ε distance. $MinObj$ and ε are input parameters to the algorithm. An object q is directly reachable from a core object o if q is within ε distance from o; $dis(o,q) \leq \varepsilon$. An object q is reachable from a core object o if there is a path $p_1,...,p_n$ with $p_1 = o$ and $p_n = q$, where each p_{i+1} is directly reachable from p_i (all the objects on the path must be core objects, with the possible exception of q). If an object is reachable from any object of the cluster, it is part of the cluster as well. All objects not reachable from any other object are outliers. The key idea of the DBSCAN algorithm is that, for each object of a cluster, the neighborhood of a given radius has to contain at least a minimum number of objects. The DBSCAN algorithm identifies the directly reachable objects of each object using the ε threshold and if the directly reachable objects are more than $MinObj$, it marks the object as a core object with a new cluster that forms a new set of reachable objects. At the end,

the composition of the clusters is verified in order to check if there exist clusters that can be merged together [54]. Algorithm 16 represents pseudo code for DBSCAN.

Algorithm 16 DBSCAN

input: O, ε, $MinObj$
output: C
$C = \{\}$
for each $O_i \in O$ **do**
  if O_i is not visited **then**
    mark O_i as visited
    NeighborsOfObj = Neighbor(O_i, ε)
    if sizeof(NeighborsOfObj) $\geq MinObj$ **then**
      Generate a new cluster $C_i \in C$
      C_i = ExpandCluster(C_i, O_i, NeighborsOfObj, ε, $MinObj$)
    end if
  end if
end for

Algorithm 17 ExpandCluster

input: C_i, O_i, NeighborsOfObj, ε, $MinObj$
output: C_i
for each $O_j \in$ NeighborsOfObj **do**
  if O_j is not visited **then**
    mark O_j as visited
    NeighborsOfObj$'$ = Neighbor(O_j, ε)
    if sizeof(NeighborsOfObj$'$) $\geq MinObj$ **then**
      NeighborPts = NeighborsOfObj $\cup$ NeighborsOfObj$'$
    end if
  end if
  if O_j is not a member of any cluster **then**
    add O_j **to** C_i
  end if
end for

The time complexity of DBSCAN is mostly governed by the Neighbor function (see Algorithm 18), which in the worst case (when ε is large) has $O(n)$ complexity, where n is

Algorithm 18 Neighbor

1: **input:** O_i, ε
2: **output:** NeighborsOfObj
3: return all points within O_i ε-neighborhood (including O_i)

the number of objects . Therefore, the complexity of Algorithm 16 is $O(n^2)$. DBSCAN does not require the number of clusters as an input. However, as it is shown in algorithm 16, it has two inputs: ε and $MinObj$. A common approach to determine ε is to compute the distance of k nearest neighbors of each object point for some k determined by the user and select ε where a sharp change is observed [71]. The proper approach for selecting $MinObj$ can be different based on the dataset. Gaonkar and Sawant [71] proposed to consider

$$MinObj = \frac{1}{n}\sum_{i=1}^{i=n} p_i, \tag{112}$$

where p_i is the number of objects in ε distance of object i. Therefore, $MinObj$ is determined as a function of ε.

Unlike K-means clustering that is designed to discover spherical clusters, DBSCAN is capable of finding clusters of different shapes and sizes. However, it fails to find clusters with different densities. The SNN algorithm, is a modified version of DBSCAN to address this problem. The main difference between this algorithm and DBSCAN is that SNN defines the similarity between two objects by the number of nearest neighbors that they share. In fact, shared nearest neighbors is the similarity measure. Using this similarity measure in the SNN algorithm, the density is defined as the sum of the similarities of the nearest neighbors of an object. Objects with high density become core objects, while objects with low density represent outliers. All remainder objects that are strongly similar to a specific core object will represent a new clusters. Like DBSCAN, SNN runs in $O(n^2)$ time [53].

Hierarchical clustering: There are two general approaches for hierarchical clustering:

1) agglomerative (bottom-up), 2) divisive (top down). The agglomerative approach considers each observation as a cluster, and pairs of clusters are merged as it moves up the hierarchy. The divisive approach considers all the observations as an unit cluster, and splits are performed recursively as it moves down the hierarchy. Divisive clustering with an exhaustive search is exponential and, therefore, the divisive methods have been largely ignored in the literature because of limitations due to their computational complexity [99].

An agglomerative hierarchical clustering algorithm is presented in Algorithm 19. First, a dissimilarity matrix $D_{nn} = dist(O_i, O_j), 1 \leq i, j \leq n$ is created. Several metric distances have been used for the dissimilarity matrix in the literature. The most common ones are the Euclidean distance, Manhattan distance, maximum distance and Mahalanobis distance [45]. Consider two clusters X and Y, where X and Y have $|X|$, and $|Y|$ number of objects, respectively. The agglomerative hierarchical clustering algorithm defines the mean distance between the elements of X and Y as the distance between the clusters:

$$dist(X,Y) = \frac{1}{|X||Y|} \sum_{O_i \in X} \sum_{O_j \in Y} dist(O_i, O_j), \tag{113}$$

and combines the two clusters that are the closest to each other into a higher-level cluster at each step. To calculate the distance between the new joined cluster $X \cup Y$ and a cluster, Z, the algorithm uses the proportional averaging of $dist(X,Z)$ and $dist(Y,Z)$:

$$dist(X \cup Y, Z) = \frac{|X|dist(X,Z) + |Y|dist(Y,Z)}{|X| + |Y|}. \tag{114}$$

The algorithm saves the distances between merged clusters in a distance vector, d_v. Algorithm 19 runs in $O(n^2 logn)$ time.

After generating dendrograms by hierarchical clustering we have to choose the level at which to cut the dendrogram. Therefore, like k-means clustering, we establish the number of clusters or groups in hierarchical clustering. Determining the number of clusters in the dataset is a challenging problem. Several approaches have been proposed to determine the

Algorithm 19 Clustering

input: O
output: C , d_v
for each $O_i, O_j \in O$ **do**
    $D(i,j) \leftarrow dist(O_i, O_j)$
end for
$C \leftarrow O$
while $|C| > 1$ **do**
    for each $X, Y \in C$ **do**
        if dist(X,Y) is equal to min distance in C **then**
            merge X and Y
            add $dist(X,Y)$ to d_v
        end if
    end for
end while

number of clusters in a dataset [132, 184]. Typically these methods use different criteria based on within cluster distances [79, 88], the ratio of between cluster distances to within cluster distances [26], or information theory [165] to find the number of clusters in a data set.

Milligan and Cooper [132] compared 30 different methods for finding the number of clusters in a dataset and concluded that Calinski and Harabasz method [26] generally shows the best performance. Calinski and Harabasz select the number of clusters that maximize the following ratio

$$CH(g) = \frac{B(g)/(g-1)}{W(g)/(n-g)}, \tag{115}$$

where g is the number of clusters in the dataset, n is the number of objects, $B(g)$ is between the clusters sum square error and $W(g)$ is within clusters sum square errors. $B(g)$ is calculated as the trace of between-group dispersion matrix B [184]:

$$\begin{aligned} B &= \sum_{m=1}^{g} n_m (\bar{f}_m - \bar{f})(\bar{f}_m - \bar{f})' \\ \bar{f} &= \frac{1}{n} \sum_{i=1}^{n} f_i, \end{aligned} \tag{116}$$

where n_m is the number of objects in group m, f_i is the set of features for object i, and $\bar{f}_m$ is the mean of features for the set of objects in group m. $W(g)$ is calculated as the trace of within-groups dispersion matrix W [184]:

$$W = \sum_{m=1}^{g} \sum_{i=1}^{n_m} (f_{mi} - \bar{f}_m)(f_{mi} - \bar{f}_m)', \tag{117}$$

where f_{mi} is the set of features for object i in group m.

When the measures are sensitive to the amplitude of the input signals, such as Euclidean distance between voltage and current variables, it is necessary to standardize the objects before clustering. Milligan and Cooper [133] performed an experimental study of seven standardization methods for clustering plus no standardization at all and concluded the approaches which standardize by division by the range of the variable give superior performance in recovering the cluster structure in the presence of noise. These approaches use the range of each feature to standardize it. A common method to standardize variable, v, is

$$v_s = \frac{v - \min(v)}{\max(v) - \min(v)}, \tag{118}$$

where $0 \leq v_s \leq 1$.

Clustering algorithms generally generate a small number of clusters from a big dataset. Therefore, these methods simplify the data analyzing process significantly. However, the performance of a clustering based technique is highly dependent on the effectiveness of the clustering algorithm in capturing the clusters structures in the dataset and a successful clustering method for a specific dataset does not necessarily performs well for others. Moreover, the computational complexity of some of the clustering algorithms could be a problem for large datasets.

6.3 Feature Learning and Feature Selection

To perform anomaly detection we are usually confronted with very high dimensional data. High dimensionality can be a problem in anomaly detection for several reasons; 1) it significantly increases the time and space requirements for processing the data, 2) anomaly detection techniques such as classification or clustering, that are analytically or computationally manageable in low dimensional spaces may become completely intractable in spaces of several hundred or thousand dimensions [25], 3) Usually many features are redundant or irrelevant in high dimensional data. The irrelevant features may hurt anomaly detection by acting as noise and hiding the relevant features. The redundant features are generally of no help for anomaly detection as well. They may reduce the effect of noise but at the same time they artificially embolden some features and, therefore, decrease the effect of others in the detection [119].

To overcome this problem, two general strategies have been used: 1) feature learning, 2) feature selection. Given the input data, D, with the set of features, m, and the performance target, c_t, a feature learning procedure generates a new set of features, h, with a lower dimension than m that can optimally characterize c_t. On the other hand, a feature selection algorithm selects a subset of features, $h \subset m$, that can optimally characterize c_t. Among feature learning algorithms, Principal Components Analysis (PCA) is the most widely used [14]. It learns a set of orthogonal bases in the directions where the data has the greatest variances. The method is easy to understand and the features are decorrelated. PCA has been widely used to reduce dimensionality for clustering [115, 120]. However, the performance of PCA in feature learning for clustering is not justified because the principal components with largest eigenvalues do not necessarily provide the best separation between subgroups [29]. Note that PCA based feature learning approach is a special case of a more general auto-encoder learning approaches.

The auto-encoder is a common feature learning approach which starts by explicitly defining a feature extracting function, f, in a specific parametrized closed form. This

function is called the encoder and will allow the straightforward and efficient computation of new feature vectors, $h = f(m;\theta)$, from the original features, m. The decoder function, g, maps the new feature space back into the input space, $r = g(h;\theta)$. The set of parameters, θ, of the encoder and decoder are learned simultaneously to minimize the reconstruction error $c_t = L(m,r)$, where L is a measure of the discrepancy between m and its reconstruction, r. PCA is a linear auto-encoder (linear encoder and decoder) with squared reconstruction error. Various versions of auto-encoders such as sparse auto-encoders, denoising auto-encoders and deep auto-encoders are proposed in the literature [14].

The main drawback of feature learning is that each new feature is a function of the full set of original features. This makes interpretation of the anomaly detection results harder. To overcome this problem, feature selection techniques can be applied. The feature selection is a data reduction process performed with the selection of subset of features that capture the relevant aspects of system operations. When we have domain knowledge about the data we can select a set of "ad hoc" features. For example, in the previous work [16] since our focus was on the power generation and distribution systems of a spacecraft, we selected voltage an current waveforms of the solar array panels, the battery, and the electrical loads in the spacecraft. Therefore, each object was made up of a set of voltage and current variables where each variable was a time series.

More sophisticated approaches have been proposed for feature selection in the literature. Huang et al [86] proposed to update the basic K-means optimization problem as follows to automatically weights variables based on their importance in K-means clustering.

$$P(U,Z,W) = \sum_{l=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{m} u_{i,l} w_j^{\beta} d(x_{i,j}, z_{l,j})$$

$$\sum_{l=1}^{k} u_{i,l} = 1, \quad u_{i,l} \in \{0,1\}, \quad 0 \le i \le n \tag{119}$$

$$\sum_{j=1}^{m} w_j = 1, \quad 0 \le w_j \le 1,$$

where k is the number of clusters, n is the number of objects, and m is the number of features in the dataset. $u_{i,l} \in U$ is a binary variable that is equal to 1 when object i belongs to cluster l, $d(x_{i,j}, z_{l,j})$ represents the distance of feature j of object i from the center of cluster l, $z_{l,j} \in Z$, $\beta > 1$ is a constant and $w_j \in W$ is the weight of feature j. The optimization algorithm assigns smaller weights to the features with higher within cluster distances. Therefore, noise variables can be identified with their small weights. Moreover, the weighting process reduces the effects of noise variables on the clustering result. They called the new algorithm W-k-means. W-k-means updates variables weights in each step based on the current clusters. The generated weights can be used for variable selection and data reduction.

Jing et al [96] considered an additional term based on information theory in the optimization cost function to avoid sparsity problem of high dimensional data. The updated optimization problem is called entropy weighting k-Means (EWKM) and is presented as

$$P(U,Z,W) = \sum_{l=1}^{k}(\sum_{i=1}^{n}\sum_{j=1}^{m} u_{i,l} w_{j,l} d(x_{i,j}, z_{l,j}) + \gamma \sum_{j=1}^{m} w_{j,l} log w_{j,l})$$
$$\sum_{l=1}^{k} u_{i,l} = 1, \quad u_{i,l} \in \{0,1\}, \quad 0 \leq i \leq n$$
$$\sum_{j=1}^{m} w_{j,l} = 1, \quad 0 \leq w_{j,l} \leq 1, \quad 0 \leq l \leq k \tag{120}$$

where $w_{j,l}$ is the weight of feature j in cluster l. Unlike objective function (119) in which a weight is assigned to a feature for the entire data set, objective function (120) assigns a weight to each feature for each cluster. The weight entropy term in the objective function, $\sum_{j=1}^{m} w_{j,l} log w_{j,l}$, is considered to stimulate more features to contribute to the identification of clusters and avoid selecting few features for each cluster with very high weights. Selecting a large value for γ leads to a relatively even distribution of weights across the features. On the other hand, a small γ leads to a more uneven distribution of weights, giving more

discrimination between features. Chen et al [32] extend objective function (120) to generate weights for the given feature groups in addition to individual features. The new clustering algorithm is called FG-k-means. When we have several groups of features this approach can be used to compare the performance of feature groups in clustering. de Amorim [38] provides an extensive survey for feature weighting based on K-means algorithm.

K-means clustering assumes the data points in each cluster are modeled as lying within a sphere around the cluster centroid. A sphere has the same radius in each dimension. Moreover, K-means algorithm models clusters by the position of their centroids, therefore, it implicitly assumes all clusters have the same radius. These assumptions are hardly the case in anomaly detection. When these assumptions are violated, K-means can behave in a non-intuitive way, even when clusters are visually identifiable [148]. Witten and Tibshirani [181] developed a general framework for feature selection that is applicable to both K-means and hierarchical clustering. They argue that weighting features in many clustering methods can be expressed as a general optimization problem of the form

$$\max \sum_{j=1}^{m} w_j f_j(X_j, \Theta) \tag{121}$$

where $X_j \in R^n$ denotes feature j with weight w_j. Θ represents clustering parameters. For example for K-means clustering Θ is a partition of the observations into K disjoint sets and for hierarchical clustering Θ is dissimilarity matrix. f_j is some function that involves only the j^{th} feature of the data. For K-means clustering f_j is between cluster Euclidean distance for feature j, and for hierarchical clustering f_j is total Euclidean distance between all the objects for feature j. For hierarchical clustering the algorithm re-weight the dissimilarity matrix. Since this method involves computations on a $n^2 \times m$ matrix, it has the potential to become quite slow if the number of objects, n, and the number of features, m, are large.

So far we have considered the feature selection methods which perform clustering and feature weighting simultaneously. These methods combine the feature subset search and

the clustering by assigning small weights to irrelevant features and, therefore, reducing their importance in the clustering algorithm. A common drawback of these techniques is that they have a higher risk of over-fitting. An alternative approach is to assess the relevance of features separately. Typically, a feature relevance score is calculated, and low-scoring features are removed. Afterwards, the set of selected features can be presented as input to any clustering algorithm. These algorithms usually scale to high-dimensional datasets easier, they are computationally more efficient, and they are less dependent to the clustering algorithm. As a result, feature selection needs to be performed only once, and then different clustering methods can be used. Peng et al [142] developed a method based on mutual information to select a subset of features that have maximum relevance to clusters and minimum redundancy. Finding the set of features that guarantees maximum mutual information with the clusters (Max-Dependency) is computationally expensive, therefore, they select a set of features with maximum average mutual information (Max-Relevance) instead. They defined the relevance between the features and clusters, D, as

$$D = \frac{1}{|X_s|} \sum_{x_i \in X_s} I(x_i, c), \tag{122}$$

where X_s is the set of selected features, c is the clusters, and $I(x_i, c)$ represents the mutual information of feature x_i and c.

Considering Max-Relevance as the only criterion can lead to a highly redundant set of features. To address this problem, they combine Max-Relevance with a minimal redundancy criterion (Min-Redundancy). To quantify feature redundancy, the authors defined features redundancy, R, as

$$R = \frac{1}{|X_s|^2} \sum_{x_i, x_j \in X_s} I(x_i, x_j), \tag{123}$$

The Max-relevance, and min-redundancy algorithm selects a subset of features X_s that maximizes $\Phi = D - R$. The Min-Redundancy term penalizes features with high mutual information. Note that this algorithm is still dependent on a clustering algorithm because we

have to have the clusters c. However, we expect less dependency than the previous approach since the feature selection procedure is not integrated into the clustering algorithm. Moreover, unlike K-means and hierarchical clustering based feature selection methods, the problem of redundant features has been dealt with in this method.

He et al [81] proposed a filter method for unsupervised feature selection. Their approach evaluates the importance of a feature by its power of locality preservation or Lapacian score. They construct a nearest neighbor graph of the objects, and select those features that represent the graph structure. Their approach has the following steps:

- Construct a graph G where each object in the data-set is a node. Two nodes are connected in the graph if one of them is among the k nearest neighbors of the other one. This step most likely will leave all the anomalies completely isolated.
- If node i is connected to node j, $S_{ij} = e^{\frac{-||x_i - x_j||}{t}}$, where t is a constant parameter. Otherwise $S_{ij} = 0$.
- Consider matrix S. They define matrix D and matrix L as follows. $D = diag(SI_{n\times 1})$, and $L = D - S$, where matrix L is often called the graph Laplacian. Let $f_r = [f_{r1}, ..., f_{nr}]$ represent the value of feature r in each object. To remove the mean from the samples, they define:

$$\hat{f}_r = f_r - \frac{f_r^T DI}{I^T DI} I \tag{124}$$

- The Laplacian score of feature r is

$$L_r = \frac{\hat{f}_r^T L \hat{f}_r}{\hat{f}_r^T D \hat{f}_r} \tag{125}$$

The features with lower Laplacian score, are the more important features. Therefore, we can consider the features with high Laplacian score as the irrelevant features. Using this

method implies that a feature which is significantly different with the rest of the features is irrelevant.

6.4 Summary and Discussion

Anomaly detection methods, especially those based on unsupervised learning, focus on finding previously undiscovered faults, or new faults that may arise because of new extensions to system behavior, aging, and system degradation. Training data labeled as nominal for every operating mode of the system may be hard to come by. Therefore, unsupervised anomaly detection is the only practical choice in many real world problems. However, it is not trivial to detect anomalies in an unsupervised manner. The large number of data points makes the unsupervised anomaly detection even more challenging. Clustering algorithms simplify the problem by grouping the data points into a small number of clusters. K-means clustering methods tend to generate spherical clusters and, therefore, are not the best choices for anomaly detection applications where the goal is to identify a small number of outliers. Hierarchical clustering does not assume any prior knowledge about the number of clusters in the dataset. Agglomerative hierarchical clustering uses dissimilarity matrix to merges the two most similar clusters at each step. We expect the anomalies to be more dissimilar to other observations and, therefore, be more resistant to be merged with normal clusters [72]. Density-based clustering algorithms require the minimum number of objects within a specific distance as the inputs and automatically determine the number of clusters and the outliers. Moreover, they are capable to find clusters of different shapes, size, and density.

To address high dimensionality problem, two general strategies have been proposed: 1) feature learning, 2) feature selection. Feature learning methods generate a completely new set of features which makes it harder for the experts to interpret the anomaly detection results. Among the feature selection techniques, W-k-means, EWKM, and FG-k-means, are various versions of K-means clustering, and, therefore, have disadvantages in anomaly

detection applications. Witten and Tibshirani [181] method which can be used in hierarchical clustering framework, is a promising alternative. For large datasets where this approach is computationally intractable, mutual information based feature selection approaches can be applied. He et al [81] proposed a filter method for unsupervised feature selection. Their method selects a subset of features that represent the general structure of the dataset. Using this method implies that a feature which is significantly different with the rest of the features is irrelevant.

CHAPTER VII

DATA-DRIVEN DIAGNOSIS AND ANOMALY DETECTION

As engineered systems have become more complex, self-monitoring, self-diagnosis, and adaptability to maintain operability and safety have become focus areas for research and development. Typical goals of such self-diagnosis approaches are the detection and isolation of faults, identifying and analyzing the effects of degradation and wear, and providing fault-tolerant and fault-adaptive control [30]. As we have discussed in the previous chapters, the majority of projects dealing with monitoring and diagnosis applications rely on models created using physical principles or by human experts. However, these models are not always available, and are often incomplete, and sometimes even erroneous. Moreover, it is hard to maintain the accuracy of these models during a system's life-cycle. More recently, data-driven alternatives have emerged that exploit the large amounts of operational data collected from systems to better understand system operations under nominal and faulty conditions [185]. The longer-term goal is to develop Cyber-Physical Systems (CPSs) [138] that can monitor their own behavior, recognize unusual situations, and inform operators, who can then modify system operations to ensure safety and ability to complete a mission. In some situations, this information can also help to plan maintenance tasks. Systems experts and engineers can use the information gleaned from this data to update operational procedures, increase autonomy of the system, and even redesign future versions of the system.

In this chapter, we take on the challenges of developing a data-driven scheme for anomaly detection. As a case study, we analyze telemetry data that was generated by NASA's Lunar Atmosphere and Dust Environment Explorer (LADEE) spacecraft[1], a robotic

[1] see https://www.nasa.gov/mission_pages/ladee/main/index.html

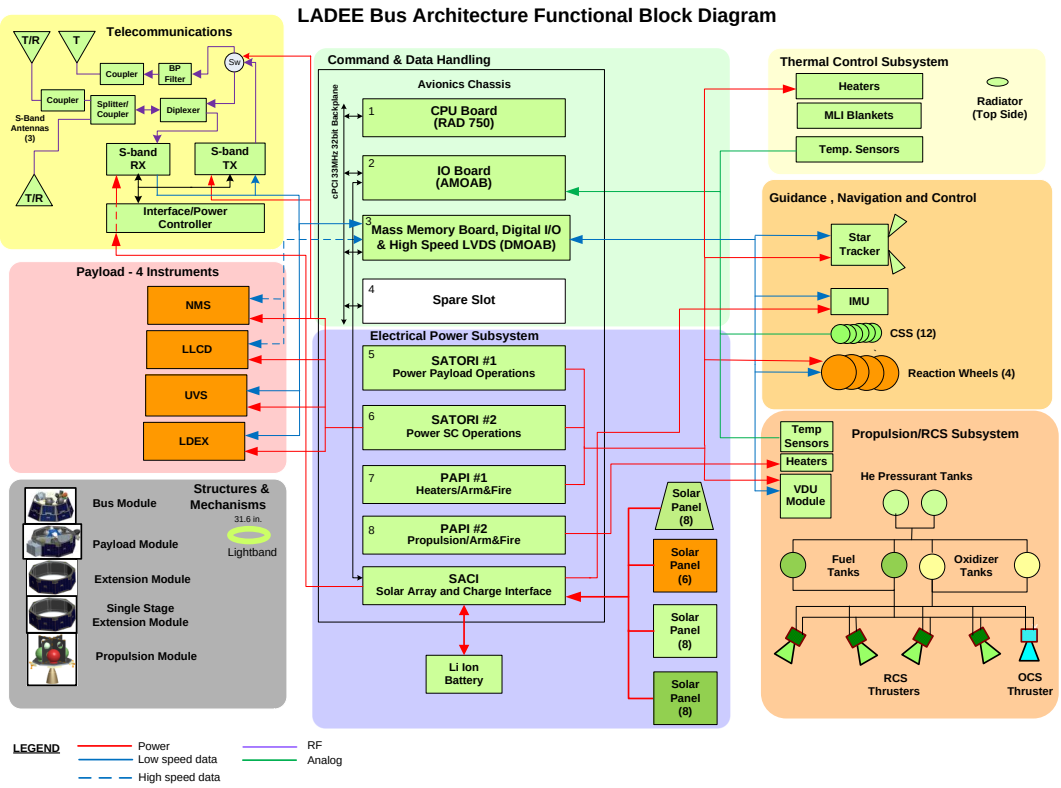

Figure 40: Block diagram of the LADEE spacecraft (image credit: NASA/ARC).

mission that orbited the moon to gather detailed information about the structure and composition of the thin lunar atmosphere, and determine whether dust is lofted into the lunar sky [83]. The LADEE system block diagram[2], shown in Figure 40, shows the four primary subsystems of the spacecraft: (1) the Integrated Avionics system, (2) the Propulsion system, (3) the Attitude Control system (ACS)), and (4) the Electrical Power Subsystem (EPS). Using the lessons learned from this case study, our overall goal is to develop a general data-driven monitoring approach for telemetry (i.e., streaming time series) data for purposes of health monitoring, which includes fault and anomaly detection, prognosis, and performance analysis of the monitored system.

Our primary focus in this chapter is on developing unsupervised methods for data-driven anomaly detection in complex systems. We want our solution to be viable for future

[2](see https://directory.eoportal.org/web/eoportal/satellite-missions/l/ladee)

long-duration space missions. For these missions that often operate in environments that are not completely known, it makes sense that we learn about system operations and anomalies that occur by collecting and analyzing data during the mission, and then using the knowledge gained to develop fault detectors and isolation mechanisms that make it easier to keep the system operational and safe as the mission progresses. To enable the operators to use our approach, we develop an anomaly detection toolbox that can be used for future missions without significant modifications. The results of our anomaly detection approach can be used for designing on-line fault detectors for system health monitoring.

When dealing with one-off space missions, one may not have access to a lot of historical data on spacecraft operations from previous missions to characterize faults and errors, that may form the basis for detecting and analyzing faults during the current mission. On the other hand, these missions are often long in duration, and it is possible to collect and analyze telemetry data from early in the mission to discover and characterize anomalies that may occur during spacecraft operations. Characterizing anomalies can help the mission specialists to come up with corrective actions or change the mission plan to avoid adverse incidents. Alternately, discovering the root causes may influence the design of future spacecraft to avoid such anomalies.

For long duration spacecraft missions, the spacecraft may operate in multiple modes linked to maneuvering the spacecraft and initiating a variety of science experiments. We have developed a multi-step unsupervised learning method to distinguish normal operating modes from the anomalies or faults. Figure 41 illustrates this process. Typically, a large majority of the time segments of the telemetry data will represent nominal operations of the spacecraft, but a small subset may represent anomalous and faulty behaviors. We hypothesize that the clusters or groups that contain a large number of the time segments represent nominal operations, whereas outliers (single time segments) and smaller groups may represent anomalous situations. In previous work, researchers have developed classifier or supervised methods for characterizing known faults (e.g., [125]) and semi-supervised and

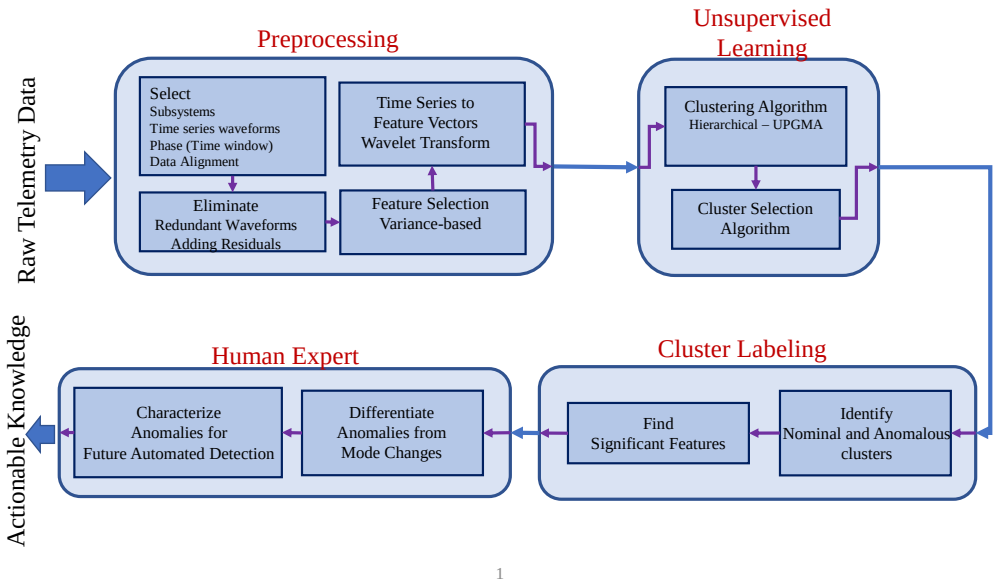

Figure 41: Unsupervised learning method anomaly and mode detection

unsupervised methods for discovering and characterizing unknown faults and anomalies (e.g., [115]). We propose a mixed method that combines unsupervised learning and expert analysis for anomaly detection in robotic space missions. Figures 41 represents our four step approach: 1) data pre-processing; 2) unsupervised learning using a clustering approach; 3) identifying outlier groups and deriving the significant features that characterize each outlier group from the nominal: and 4) expert characterization of the anomalous groups. We describe each step in greater detail in this chapter.

The rest of this chapter is organized as follows. Section 7.1 presents the per-processing steps. Section 7.2 represents our unsupervised learning method using a hierarchical clustering approach. Section 7.3 presents our approach for identifying outlier clusters and deriving the significant features that characterize each outlier cluster from the nominal. Section 7.4 shows the application of our methodology to telemetry data from the Electric Power System (EPS) of the LADEE spacecraft. Section 7.5 extends the analysis to include the EPS and

the Guidance, Navigation and Control (GNC) units of the spacecraft. Section 7.6 presents the summary and conclusions of the chapter.

7.1 Pre-processing

7.1.1 Standardization

The feature selection algorithms and the clustering algorithms are sensitive to the amplitude of the features. The features can have a large range of amplitudes. For example, in the LADEE dataset the temperature variables are typically larger than the electrical current measurements. This can bias feature selection and clustering algorithms. To avoid this problem, we standardize the variables as the first step. Milligan and Cooper [133] performed an experimental study of seven standardization methods for clustering and concluded the approaches which standardize by division by the range of the variable give superior performance in recovering the cluster structure in the presence of noise. These approaches use the range of each feature to standardize it. We use the method presented in equation (118) to standardize each signal in the dataset.

7.1.2 Defining the objects

In this work, anomaly detection is applied to telemetry data streamed to earth stations from different subsystems of the spacecraft. We derive a set of objects from a curated version of the time series data. Each object is defined by a set of signals, and each signal is extracted from a longer time series signal representing the variable waveform over the entire mission. We start with each time series waveforms that captures the relevant aspects of system operations represented as a variable; therefore, each data object is represented by a set of variables, $V = \{v_1, v_2, ..., v_m\}$, and each variable is a time series made up of k_j samples, $1 \leq j \leq m$. For example, in the first case study, our focus is on the EPS, so the set of variables includes electrical measurements from components such as solar array panels, the battery, and the electrical loads. In the second case study, we extend the analysis to

include GNC subsystems of the LADEE spacecraft. Therefore, the set of variables also includes measurements related to the GNC subsystems such as the reaction wheels, and the star trackers.

Our approach divides the time series representing the entire mission trajectory into segments, i.e., $O = \{O_1, O_2, ..., O_n\}$, and each segment represents an object of interest on the mission time line. We can adopt different approaches to define the objects. In the fist case study, we assume the windows have the same number of samples and use an empirical approach to derive the window size. In the second case study, we derive the time interval width of each object (window size) by considering the mission phases (see Figure 42). In the beginning of the mission the spacecraft has access to the sunlight constantly. When the spacecraft enters the lunar orbit it experiences dark and light intervals periodically. To select the time interval for each object we adopt the following strategy. In the earth orbital phase, each time window (object) is an hour long. During the lunar orbital phase, each period of dark or light is selected as an object. This strategy has been chosen based on our expectation that the signals in the dataset follow different patterns during the light periods and the dark periods.

7.1.3 Re-sampling

In the second case study, we define the object windows using dark and light periods. Since measurement sampling rates vary, each window can have a different number of samples. This makes the comparison between the objects and, therefore, detecting the anomalous objects complicated. We apply a simple sampling approach in order to have the same number of samples in each object. For discrete variables, we estimate the value of signal in each sample time t_k by using the closest available sample point. For example consider the case where t_a is the time of the last sample point before t_k and t_b is the time of the first available sample point after t_k. We estimate the value of discrete variable, v_i, at sample

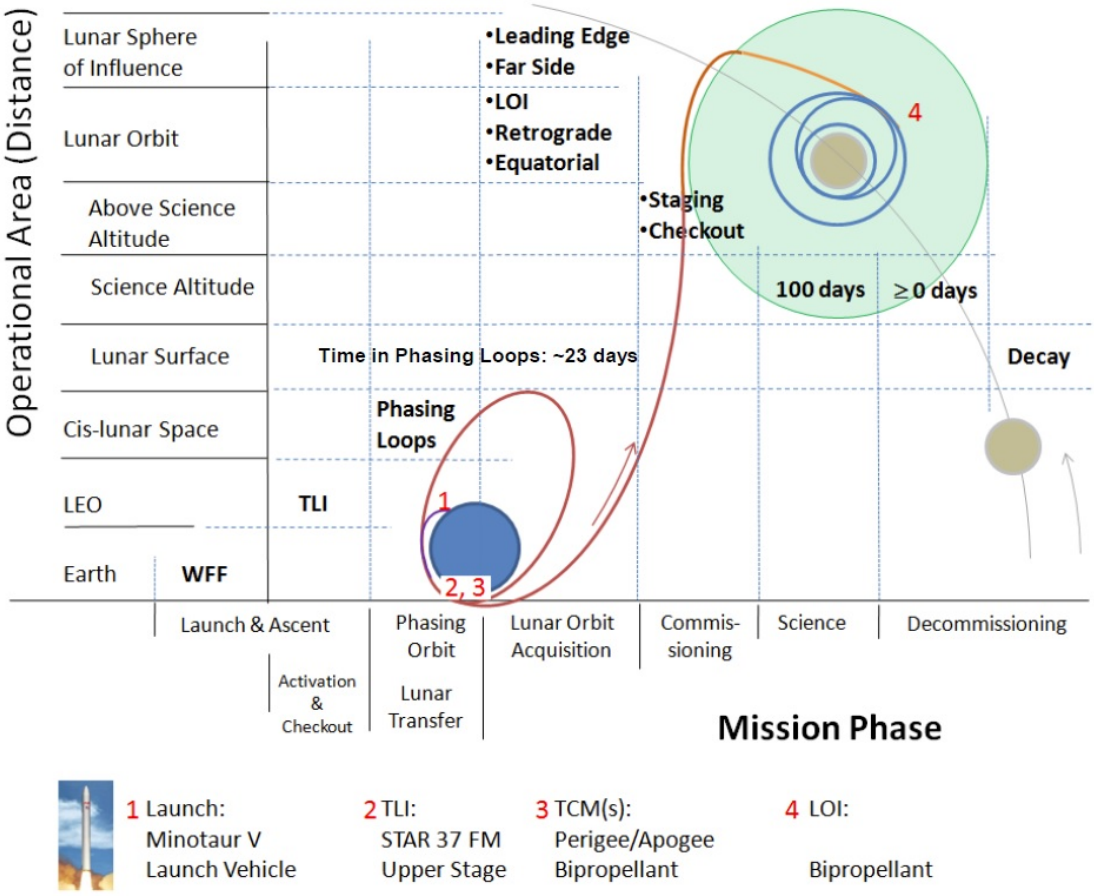

Figure 42: LADEE mission phases [48]

point t_k as

$$v_i(t_k) = \begin{cases} v_i(t_a) & if \quad t_b - t_k \geq t_k - t_a \\ v_i(t_b) & otherwise. \end{cases} \tag{126}$$

For continuous variables, we consider a weighted average of previous and next sample point as the value of the signal. For example, consider the case where we are interested to estimate the value of a continuous signal, v_j at time t_k. If the value of v_j is not measured at t_k, we estimate v_j at time t_k as

$$v_j(t_k) = \frac{t_k - t_a}{t_b - t_a} v_j(t_a) + \frac{t_b - t_k}{t_b - t_a} v_j(t_b), \tag{127}$$

where t_a is the sampling point before t_k and t_b is the sample point after t_k. In the second case study, we select 64 sample points for each object.

7.1.4 Feature selection

In the feature selection step, our goal is to address two main challenges in the data-driven anomaly detection: 1) redundant variables, 2) irrelevant variables.

7.1.4.1 Redundant variables

Redundant variables are very common in datasets. There are different reasons for redundant variables. For example, several sensors may measure the same variable, a single measurement may be recorded with different names in the dataset, or two variables could be highly correlated. Redundant measurements can make anomaly detection challenging. They may artificially enhance some effects and, therefore, decrease the effect of others, making some faults hard to detect. On the other hand, they can represent redundancies in the dataset and, therefore, they can be valuable resources for anomaly detection.

Whitley et al [180] compute squared correlation coefficients for pairs of variables, and eliminate one of the pair if the coefficients values are large. In this work, we use the

absolute value of *Pearson correlation coefficient* as a measure for redundant variables. Pearson correlation coefficient of two measurement vectors, v_j and v_k is computed as the ratio of their covariance over the product of their standard deviations.

$$r_{jk} = \frac{E[(v_j - \mu_{v_j})(v_k - \mu_{v_k})]}{\sigma_{v_j}\sigma_{v_k}}, \tag{128}$$

where $E(x)$ and σ_x represent the expectation value and the standard deviation of x respectively. A simple solution is to remove any variable that has high correlation with another variable. However, as we mentioned earlier, the redundant variables can represent redundancies in the system and, therefore, they may provide critical information for anomaly detection. To capture this information, we add a residual to the dataset after removing each variable.

For example, it is possible that two variables have similar patterns during the normal operation, but follow different trajectories in the fault modes. Typically, the majority of data points represent nominal operations, therefore, we expect the variables to be highly correlated. However, the difference between their trajectories during the anomalous points can help us to detect these anomalies. Our goal is to generate a residual that captures possible dissimilarities between the variables with high correlation coefficients. Consider two measurements v_j and v_k with correlation coefficient, r_{jk}, above the minimum threshold for redundant variables, $r_{threshold}$. After removing v_k from the set of variables we add residual, Res_{jk}, to the dataset.

$$Res_{jk} = v_j - \frac{v_j.v_k}{v_k.v_k}v_k, \tag{129}$$

where $v_j.v_k$ represents inner product of v_j and v_k. When v_i and v_j are identical, $Res_{jk} = 0$.

7.1.4.2 Irrelevant variables

The irrelevant variables may hurt anomaly detection by acting as noise and hiding effects of the relevant variables. To remove irrelevant variables, Peng et al [142] developed an approach to select a subset of features that have maximum relevance to the clusters. In this work, the set of clusters are not determined. Therefore, we have to adopt an unsupervised method to remove irrelevant features. Whitley et al [180] argued that the variables with small standard deviations contribute no significant information and therefore, are irrelevant. In this work, we consider the minimum number of waveforms or variables that represents an specific percentage of the total variance as the relevant variables. Let $V = \{v_1, v_2, ..., v_a\}$ denotes the set of variables plus the added residuals. We select a subset of variables $V_s = \{v_{s1}, ..., v_{sb}\}$ that represent a minimum required percentage, I_t, of the total sum of variable variances, i.e.,

$$\begin{aligned} \min \quad & V_s \subseteq V \\ \text{s.t.} \quad & \sum_{i=1}^{b} (\sigma_{v_{si}})^2 > I_t \sum_{j=1}^{a} (\sigma_{v_j})^2 \end{aligned} \tag{130}$$

Using this approach, we retain features with the highest information content. Therefore, we automatically remove the residuals that are generated from identical variables and do not contribute any meaningful information.

7.1.5 Feature extraction

High dimensionality of the datasets in complex systems is another challenge for anomaly detection. High dimensionality in the data increases the required time and space for processing the data. For example, in the first case study, we have 1512 objects and each object has 380 samples for each feature. In the second case study, we have 877 objects in the earth orbital phase and 4556 objects in the lunar orbital phase and each object includes 64 sample points for each selected feature. We use the *feature reduction* step to convert each

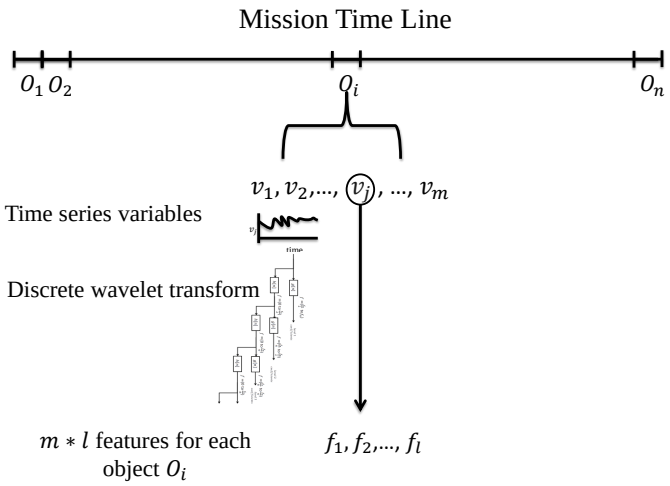

Figure 43: Data pre-processing and feature extraction

time series signal to a set of discrete features whose values are derived by applying the wavelet transform [24] to the signal. Figure 43 illustrates the feature extraction process on the time line for each signal segment. The wavelet transform captures the time-frequency characteristics of signal waveforms, and, in this process it can also be used to capture the frequency characteristics of the signal at different time intervals in the signal. The wavelet transform that we describe in greater next, is illustrated in Figure 44.

We employ the Haar discrete wavelet transform (DWT) [164] to extract the time-frequency characteristics of the signals at specific intervals, and, in this process, compress the signal waveforms. Computing the Haar wavelet coefficients is equivalent to passing the signal through a series of shifted and cascaded low- and high-pass filters that decomposes the signal into high and low frequency bands, which are then down-sampled to capture the local time-frequency characteristics of the signal. Figure 44 shows the first three levels of the computation for an input example, $o = [3, 4, -1, 5, 6, 57]$. The computational scheme requires the number of discrete samples in the signal to be a power of 2. Therefore, DWT algorithms extend the signals of other sizes using different signal extension methods such as zero-padding.

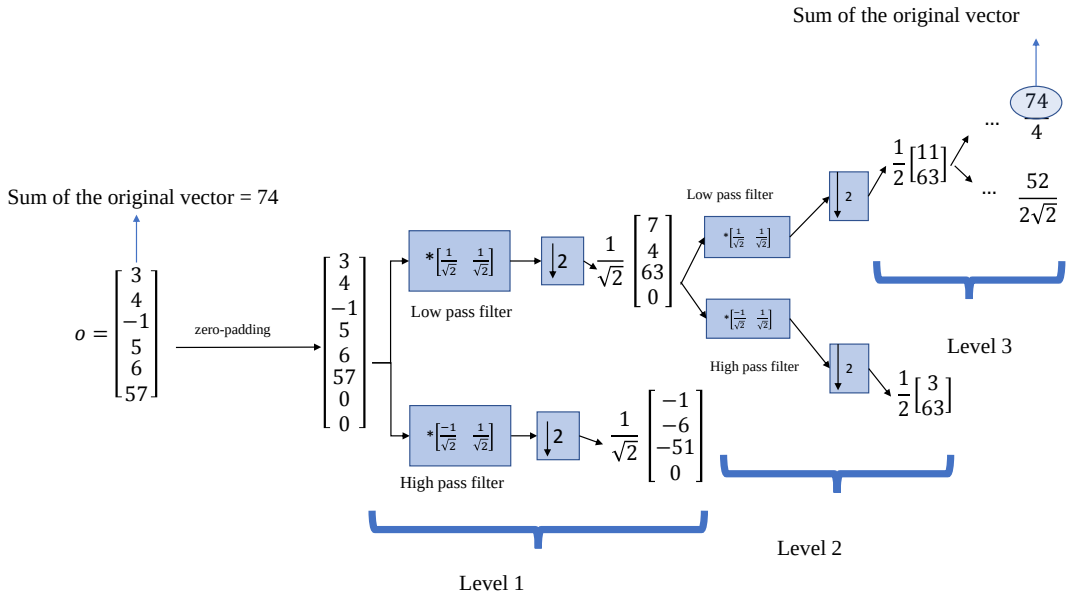

Figure 44: Haar Discrete Wavelet Transform (DWT)

In the first case study, each time window has 380 sample points, and, we use zero-padding to extend each window to 512 samples. In the second case study, in the re-sampling step we make sure that the number of samples in each object is a power of 2. Therefore, signal extension was not required. Based on the application, we select the DWT coefficients at a specific level as the features that define the signal. Therefore, the set of features extracted for each object, O_i, is a vector $f_i \in R^{m*l}$, where m represents the number of selected time series signals, and l is the number of coefficients extracted for each signal by the Haar transformation. In fact, each data object is represented by a set of $m * l$ features. In the second case study, each object has 64 samples and we consider level 4 DWT coefficients as the features that define the signal. Therefore, we represent each signal in a data object with 8 features which reduces computational cost compare to 64 original sample points.

7.2 Unsupervised Learning

For unsupervised learning in step 2, we have applied a hierarchical clustering algorithm [94]. We adopt the Euclidean distance as the metric to compute the dissimilarity matrix.

For clustering, we run a generic UPGMA (Unweighted Pair Group Method with Arithmetic Mean), agglomerative (bottom-up) hierarchical clustering algorithm [37], and represent the order of cluster formation as a dendrogram. UPGMA defines the distance between the clusters as the mean distance between the cluster's objects. At each step, two existing clusters that are the closest to each other are merged into a higher-level cluster. The algorithm saves the distances between merged clusters in a distance vector, d_v. To achieve computational efficiency in the next step of merging, the algorithm calculates the distance between the new cluster and all other existing clusters, using the proportional averaging (see Algorithm 19).

After generating dendrograms by hierarchical clustering, we have to choose the level at which to cut the dendrogram. One of the advantages of generating dendrograms by hierarchical clustering is that we can apply a number of heuristic methods to choose the level at which to cut the dendrogram, and, in this process establish the number of clusters or groups in the data set. Several approaches have been proposed for determining the number of clusters in a data set (e.g., [132, 184]). In the first case study, we select the number of clusters based on a metric derived from the distances between successive cluster formations in the dendrogram. The distance level ($y-$axis on the dendrogram) at which the clusters are partitioned is defined by a distance threshold (say, d_t) to define a distinct grouping of clusters. Therefore, by increasing or decreasing d_t, we can decrease or increase the number of clusters considered.

Our approach to selecting the value d_t ensures that the clusters or groupings formed are unambiguous and stable, i.e., small changes in d_t do not result in large changes in the number of clusters generated. Toward this end, we apply a Z-test [18] to determine where the change in the distance vector, d_v is statistically significant. In the second case study, we select the number of groups based on two common methodes; 1) Calinski and Harabasz method [26] and 2) Krzanowski and Lai method [108]. Calinski and Harabasz method selects the number of clusters that maximizes the between the clusters distance over

within clusters distance ratio (see (115)). Krzanowski and Lai method selects the number of clusters where there is a dramatic decrease in within clusters distances.

7.3 Cluster Labeling

In order to gain a deeper understanding of anomalies, and to devise methods for detecting anomalies, we have to identify the clusters that represent anomalies and understand the root causes that differentiate them from nominal operations. We consider large groups derived from the clustering algorithm to be nominal (this corresponds to the assumption that the system operates normally most of the time). Singletons and smaller groups that are sufficiently distant from the nominal groups are labeled as outliers or anomalies. As discussed earlier, spacecraft missions are complex, and they may involve multiple phases and operational modes, corresponding to trajectory maneuvers and conducting of scientific experiments, over the duration of the mission. In reality, some of the smaller clusters initially labeled as anomalies or outliers may correspond to special modes of operation, and, therefore, are not of interest in discovering discrepant and faulty behavior. Therefore, an additional challenge we face in this work is separating the special modes of operation from truly anomalous behaviors.

We have developed an approach to extract additional cues to identify special operating modes. We map the objects constituting the smaller clusters back onto the mission timeline, and look for continuous or discrete signals that may explain the differences in system behavior in the small clusters from the nominal operation. For example, the reaction wheels are activated to correct the attitude of the spacecraft, and this can be detected by a switch turning on to supply power to the reaction wheels. The activation of the reaction wheels increases the overall load currents in the power system, but since this increase can be primarily correlated with the switches being commanded on, the experts labeled the outlier group corresponding to this phenomena as a special operational mode rather than

an anomaly. In this section, we generalize this approach to detect other special modes of operation.

Other groups of data objects not be explained by observed mode changes in spacecraft operations, are then presented to human experts for further characterization. These may turn out to be additional special modes that are not easily interpreted from the switching signals, or they may represent anomalous behaviors that are linked to faults in the system. We define *significant features* to formally generate additional cues for the experts in order to help them distinguish anomalies from the special modes of operation. The significant features are defined as follows.

Definition 40 (Significant features)**.** *Significant features are a single feature or a set of features that best distinguish an outlier group from nominal operations of a system.*

To facilitate identification of anomalies and special modes, we pick significant features that best differentiate each small cluster from the labeled nominal groups. These features help our human experts better understand and characterize the outlier clusters as potential faults, or special modes of operation. Different methods, such as variance decomposition [75] and information gain measures applied to decision trees [93] can be applied to extract significant features for each outlier cluster. In our work, we developed a simple *Euclidean distance based method* to extract significant features: The distance measure between normal operation group, a, and an outlier group, b for signal variable, j is computed as:

$$D^j_{ab} = \sqrt{\sum_{i=1}^{l} (\frac{E[o^j_{ai}] - E[o^j_{bi}]}{E[o^j_{ai}]})^2}, \tag{131}$$

where $E[o^j_{ai}]$ represents the mean value of the ith level coefficient of signal j in group a. When summed over all m variables, the total distance between the normal operation group, a and an outlier group, b is computed as:

$$D_{ab} = \sqrt{\sum_{j=1}^{m} (D^j_{ab})^2}, \tag{132}$$

We define the importance of each time series waveform v_j in distinguishing an outlier group, b, from normal operations, a, $I_{ab}(v_j)$, as the ratio of D^j_{ab} to D_{ab}, i.e.,

$$I_{ab}(v_j) = \frac{D^j_{ab}}{D_{ab}} \tag{133}$$

The importance of a set of variables, $V_k = \{v_1, v_2, ..., v_k\}$ in distinguishing b from normal operation, a, is defined as:

$$I_{ab}(V_k) = \sqrt{\sum_{i=1}^{k} (I_{ab}(v_i))^2}. \tag{134}$$

Let $V = \{v_1, v_2, ..., v_m\}$ denote the set of variables. We select a subset of variables V_b to guarantee a minimum required importance, I_r, in distinguishing b from normal operation with minimum cardinality, i.e.,

$$\begin{aligned} \min \quad & V_b \subseteq V \\ \text{s.t.} \quad & I_{ab}(V_b) > I_r \end{aligned} \tag{135}$$

Once the significant features have been established and ranked, this information is presented to the human expert to further characterize the anomalous group. After study, the expert may establish that this group represents a true anomaly, i.e., unexpected or aberrant behavior, or otherwise it is a mode of operation that we could not characterize.

7.4 Case study 1: The Electrical Subsystem

The data used for this chapter was telemetry data from the LADEE lunar mission directed by the NASA Ames Research Center. This mission lasted for 223 days from launch till the spacecraft was intentionally crashed onto the moon's surface. The telemetry data we analyzed contained $2,949$ time-series waveforms that represented variables from the different subsystems of the spacecraft. The sampling rates for the waveforms differed between subsystems, and they also differed during the different phases of the mission. Overall, the data set contained $1,894,285,525$ samples, which was about 14 GB of data. For the first

case study, we focused on the Electric Power System (EPS) of the spacecraft. This subsystem includes 265 time series variables. From the 265 variables, we selected 34 continuous voltage and current variables for analysis. Of these 7 represented voltage variables; this included the battery voltage, solar panel voltages, and load voltages. 27 were current measurements, such as, battery, solar panel, and load currents. 67 variables were binary-valued, and they helped us interpret the different modes of operation of the EPS.

For our analysis, we subdivided the 34 continuous voltage and current telemetry waveforms into 1512 windows, with each window corresponding to a data object. Each time window contained 380 samples. As discussed earlier, the sampling rate of the recorded data was not constant, therefore, a time window represented anywhere between 5 minutes to 10 hours of operation. The average window size was 3 hours and 31 minutes. The Haar wavelet transform was applied to each waveform segment to extract a set of 8 wavelet coefficients as distinct features representing that segment. The result was that the set of voltage and current waveforms for each data object were transformed into $34 \times 8 = 272$ features. To apply the discrete Haar transform, each waveform segment had to be represented by a number of samples that were a power of 2, therefore, we padded our waveform represented by 380 samples with $0's$ to make $2^9 = 512$ samples.

The Euclidean distance metric was used to create the dissimilarity matrix of 1512×1512 object pairs. Then we applied the UPGMA hierarchical clustering algorithm (the R function: *hclust*)[3] to generate the dendrogram shown in Figure 45. The dendrogram is a graphical representation of the order in which the objects and groups merge to form larger clusters. Figure 46 represents 1511 distance values at which the objects and groups merged to form larger groups in the dendrogram. This case study was conducted more as a proof-of-concept as opposed to an attempt to exhaustively generate all of the special modes and anomalies. Therefore, we intentionally set a very high confidence bound of

[3] see http://www.R-project.org/

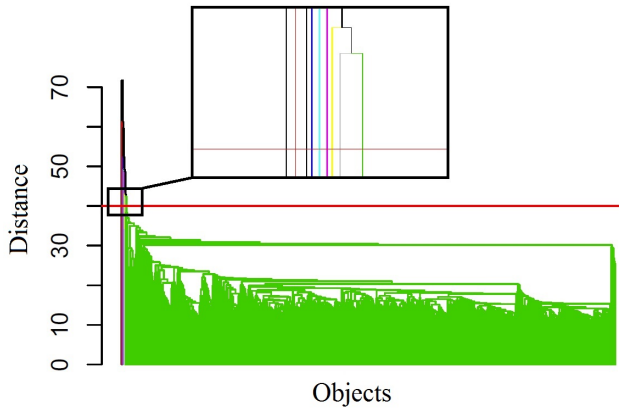

Figure 45: The dendrogram generated by applying the UPGMA hierarchical clustering algorithm. The red line represents the chosen threshold distance for cluster formation. The green section of the dendrogram (the large cluster) represents normal operations, and the outliers and smaller groups are represented by different colors

99.7% to establish the level at which the dendrogram would be cut to establish the number of clusters.

Application of Z-test with 99.7% confidence bound produced the distance threshold and the corresponding red line shown in Figure 46. As expected, this produced one large cluster

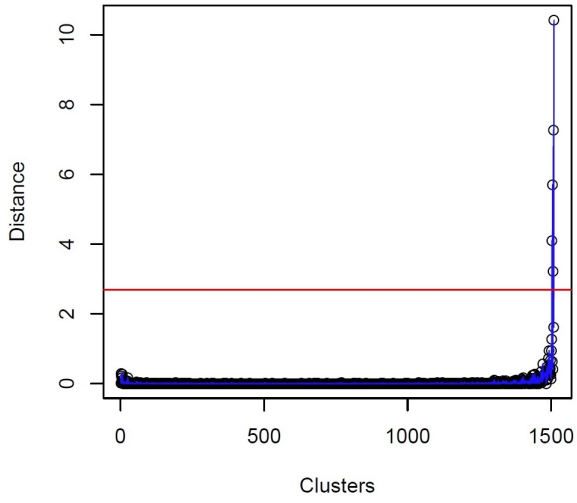

Figure 46: Distance values indicating levels of cluster formation

that we assumed to represent the nominal behavior of the spacecraft. In addition, Figure 45 shows that we generated eight smaller clusters. We studied these groups in greater detail by comparing them against the nominal group to determine if they represented special modes and anomalies. Of the eight groups, three turned out to be modes of operation that were identifiable because of their correspondence to the switching signals. In this chapter, we analyze the other five groups in greater detail. To study the five smaller clusters, we identified the objects corresponding to these clusters on the spacecraft mission timeline. Figure 47 shows these objects as dots on the timeline plot.

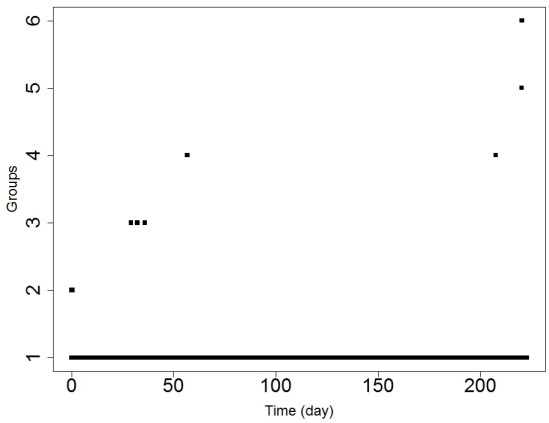

Figure 47: Clusters projected back on the mission timeline

As a first step toward mode characterization, we studied the discrete switch values during these intervals to see if they provided information about special modes of operation. When we were unable to assign a definite interpretation to a cluster, we extracted the significant features that differentiated that cluster from the nominal group. Using equation (135) we selected significant features by setting the threshold, $I_r = 0.9$. The significant features represented an ordered subset of features that contributed the largest amounts to the distance from the mean of the outlier group to the mean of the nominal, and the chosen subset accounted for 90% of the distance between the outlier and nominal group means.

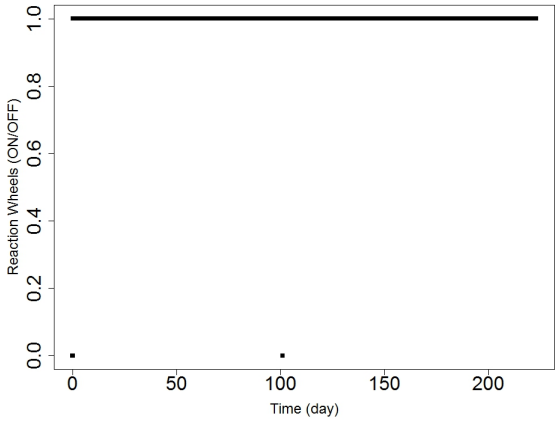

Figure 48: Reaction wheels (OFF=0, ON=1)

Table 24 represents the most important significant features of five of the outlier groups we identified by our clustering approach.

We presented the significant features for each anomalous group and the position at which they occurred on the mission time line to our experts to help us further characterize and classify the special modes and anomalies. In this section, we present our characterization of the five clusters in greater detail using expert input. Our mission experts and specialists from NASA Ames are acknowledged at the beginning of this thesis. Cluster 5 represented the nominal behavior of the spacecraft. Clusters 1-4 are discussed below.

Cluster 2 (the reaction wheels control problem): the behavior represented by this cluster covered two time windows that occurred early in the mission. The behaviors covered 40 and 6 minutes of the mission time line, respectively. Figure 48 shows that the reaction wheels went off twice (corresponding to Reaction Wheels = 0) during the mission. Our experts confirmed after studying the mission operator logs that the reaction wheels only went off once during the mission, and the second zero in the figure was a case of bad data. Figure 49 shows that different currents in the SATORI board[4]#2 were the most significant features for this cluster. SATORI #2 current variable has significantly higher average in cluster 2 than the nominal operation.

[4]The SATORI boards provide power to the Command & Data Handling System.

Table 24: Summary description of the detected modes and anomalies

Group	Detected Mode or Anomaly	Significant Features	Switches
1	Normal operation mode		
2	Anomaly: Reaction wheels	• SATORI #2 HP #2 current • SATORI #2 HP #4 current	• Propulsion heater turned on • Star tracker went off
3	Mode: Lunar orbit insertion	• PAPI #2 HP #7 current • PAPI #2 current	• Pressurant tank heater went on • Valve driver unit went on
4	Anomaly: Laser communication test (during dark phase)	• SATORI #1 Current • Load Current • Battery Current	• Laser communications switch went on
5	Anomaly: Eclipse lasted longer than expected	• Battery Voltage • SATORI #1 Voltage • SATORI #2 Voltage	• Several heaters went on (e.g. Propulsion heater)
6	Mode: Safe	• Battery Voltage • SATORI #1 Voltage • SATORI #2 Voltage	• Several loads (e.g. star tracker) turned off

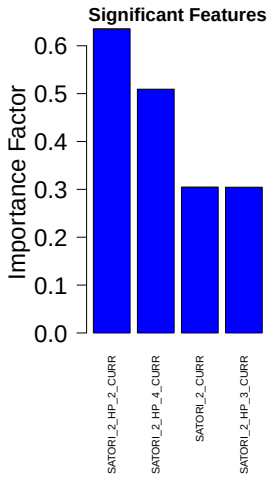

Figure 49: Significant features for cluster 2

The high current values read by the sensors on the SATORI #2 boards indicated that this incident very likely was related to the guidance navigation and control unit. The fact that these currents were related to the three Reaction wheels further confirmed this interpretation. The experts from NASA further substantiated this anomaly as follows. In the first few orbits around the earth, the spacecraft began to spin at a faster rate than was expected, and the reaction wheels were turned off by the control software to avoid a high load current, and, therefore, draining of the battery. This stopped the spacecraft rotations, but, as a consequence, the side of the spacecraft facing away from the sun became much colder than normal. Several heaters went on to prevent the equipment from freezing, and this led to the high currents in a number of units connected through the SATORI board#2.

Cluster 3 (lunar orbit insertion): Figure 47 shows that cluster 3 data objects corresponded to three time intervals that occurred on three different days of the mission. Each time interval was about six minutes long. Two different currents in the Power-switching and Pyro Integration boards (PAPI) board[5]#2 were the significant features that characterized this group. Figure 50 shows that the PAPI #2 high pressure current number 7 during these three time intervals. The high amplitude in the PAPI # 2 propulsion subsystem current

[5]The PAPI boards route power to the Thermal Control, Guidance, Navigation & Control and the Propulsion Subsystems.

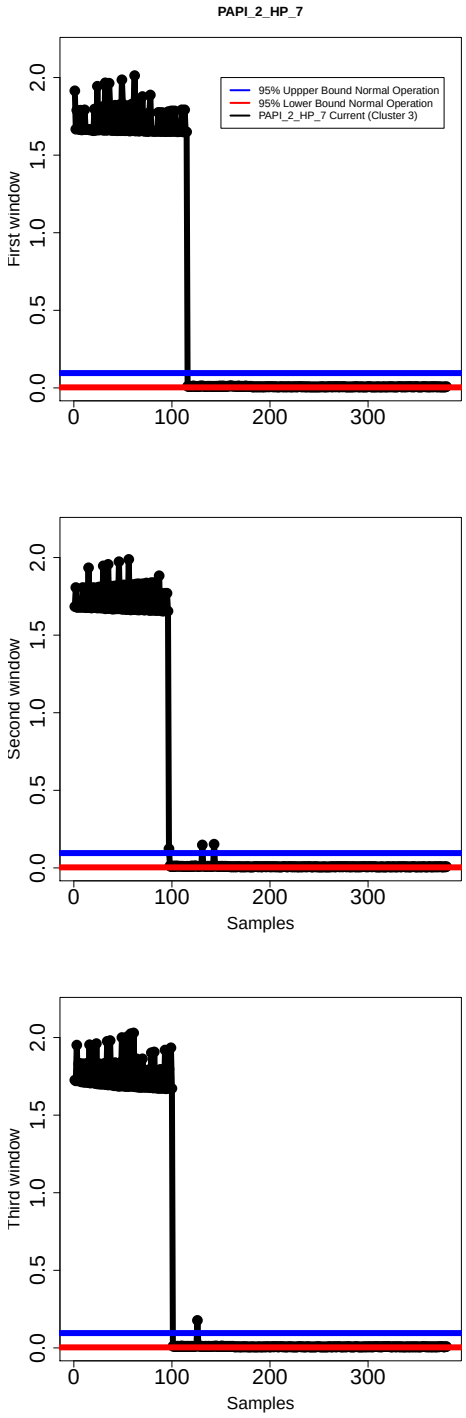

Figure 50: PAPI #2 high pressure current number 7 for cluster 3 objects

was the second significant feature for this cluster. However, unlike cluster 6, the increase in PAPI board #2 current did not occur simultaneously with the battery voltage drop. With the help of our experts, we found out that the valve driver unit, which controls the propulsion subsystem and the pressurant tank heaters, (part of the propulsion subsystem (see Figure 40)) were ON for the three time intervals. This corresponded to a unique behavior, however, our experts confirmed that the behavior was not anomalous. Instead, it represented the lunar orbit insertion process. There were three firings of the propulsion subsystem that occurred to get the spacecraft into lunar orbit and our algorithm successfully grouped them into a single cluster.

Cluster 4 (the laser communication test): this cluster included two time windows, each about 20 minutes in duration. The SATORI #1 current is the most significant feature for this cluster. The load current and battery current are the next two significant features for this cluster. Further analysis showed that the data points in this cluster corresponded to laser communication tests, which were part of the mission plan. The laser communication tests increased load currents significantly. The two time intervals in this cluster also coincided with the occurrence of a new moon, which meant that the solar arrays were not generating any current (values recorded were very close to 0) during this period. The high battery current caused the battery to discharge below acceptable levels, and, therefore, the battery voltage dropped significantly. Our experts characterized this as an anomaly in operations because the laser communications test led to unintended consequences of the battery voltage dropping below specified thresholds.

Cluster 5 (the eclipse): the objects in this cluster extended over a 5 hour time span. It should also be noted that the sampling rate was also significantly lower, because this was the end of the mission. The most significant feature for this group was the battery voltage, which fell below the 95% bounds of normal operation (see Figure 51). The drop in battery voltage led to drops in the SATORI #1 and SATORI #2 voltages. Figure 52 shows SATORI #1 and SATORI #2 voltages were the next set of significant features.

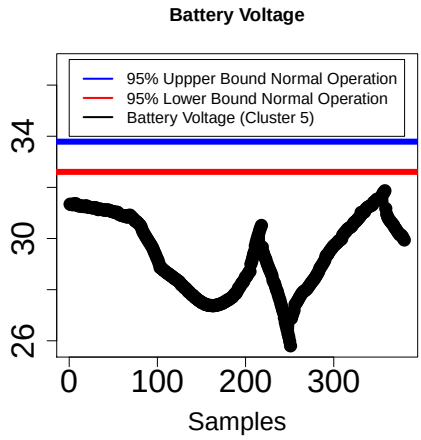

Figure 51: Battery voltage for cluster 5 data points

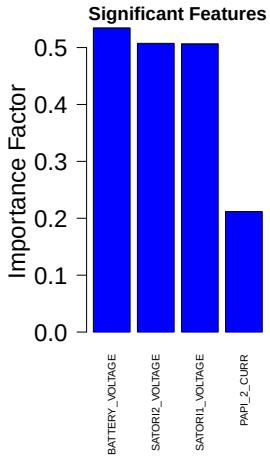

Figure 52: Significant features for cluster 5

The experts confirmed that cluster 5 behavior was directly the consequence of the eclipse that happened at the end of the mission. The solar array current was zero during the first 2 hours of this time interval, and it fluctuated between zero and small non zero values after that. The lack of sunlight caused a significant drop in temperatures (we did not include temperature values in this analysis), and several heaters came on to prevent large temperature drops, which would have affected spacecraft operations. This increased the load current significantly. A simultaneous increase in the load currents and decrease in

the solar array current put an unprecedented load on the battery, which led to large voltage drops in the battery voltage. This likely jeopardized the battery health, therefore, it clearly represented anomalous or unexpected behavior of the EPS.

Cluster 6 (the safe mode): the system went into the safe mode right after the eclipse ended. This mode was about $4\frac{1}{2}$ hours long. The battery voltage was again the most significant feature that distinguished this group from nominal operations. Figure 53 shows battery voltage during this time interval. The remaining significant features for this cluster

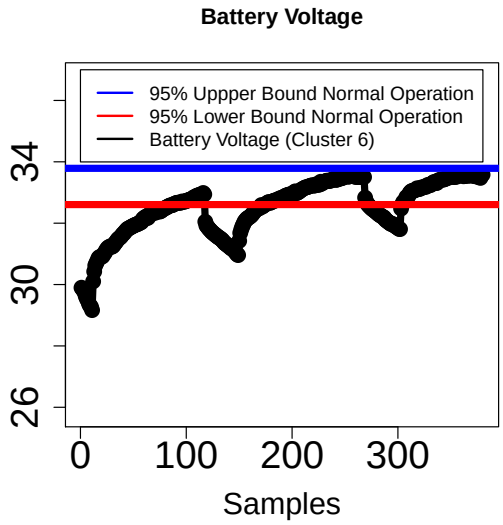

Figure 53: Battery voltage for cluster 6 objects

and their importance factors are presented in Figure 54.

To preserve the health of the battery, several loads were switched off to reduce energy consumption and give the battery a chance to recharge. Figure 53 shows that the battery voltage came back to an acceptable level during this mode. Our experts explained that the data points in this group represented a unique behavior in spacecraft operations. However, they did not classify the behavior to be anomalous, since the spacecraft systems operated exactly as they should have to avoid larger failures and possible loss of the spacecraft power

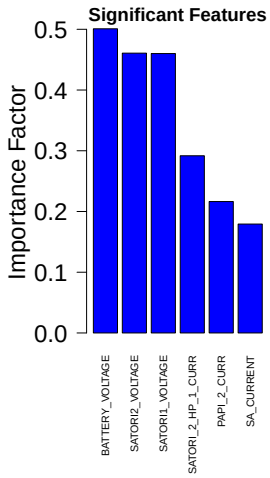

Figure 54: Significant features for cluster 6

system. Hence, this was a special operational mode, i.e., the safe mode to allow for EPS recovery.

7.5 Case Study 2: Anomaly Detection combining telemetry data from the EPS and GNC

In this case study, we include telemetry data from the GNC subsystems in addition to the EPS. For the clustering analysis, we only consider continuous-valued features. In the cluster labeling step we also include binary switches. In the GNC subsystem, we consider variables related to the reaction wheels (RW), and internal measurement unit (IMU). There are 28 variables associated with 4 reaction wheels in the spacecraft and each variable has $3,796,693$ data samples. These measurements include reaction wheel's torques, reaction wheel's rotational speeds, and gyro and reaction wheel's temperatures. The IMU has 4 variables: three accelerations and a temperature. Each variable in the IMU dataset has $692,459$ sample points. As we discussed in the previous case study, the EPS dataset includes 37 voltage and current continuous variables, and 67 binary switches. Each variable in the EPS has $574,687$ data samples. Without standardization, variables with smaller amplitudes have small effect on the clusters. However, these variables may be important

in detecting abnormal behaviors. To avoid neglecting variables with small amplitudes, we standardize all the variables using equation (118).

For our analysis, we divide the mission into two main parts; 1) initial earth orbital phase 2) lunar orbital phase. In the earth orbital phase, the spacecraft is exposed to sun light constantly. Therefore, we do not observe charge and discharge cycles in the electrical variables such as battery voltage. However, in the lunar orbital phase, the spacecraft enters cycles of dark and light and we can observe charge and discharge cycles in electrical variables. The dark cycles and the light cycles represent fundamentally different behavior. Therefore, the experts in NASA recommended to consider dark and light periods in defining objects. For the earth orbital phase, where there is no dark period, we consider each 60 minimizes time interval as an object. This subdivides the earth orbital phase into 877 windows. For the lunar orbital phase, we consider each dark or light interval as an object. This subdivides the lunar phase into 4556 windows where we have 2278 light objects and 2278 dark objects. As discussed earlier, the sampling rate of the recorded data was not constant, therefore, the time windows may have different number of samples. Moreover, EPS, RW, and IMU subsystems have different number of sample points. In order to integrate the subsystems and make the comparison of the objects easier, we re-sampled the waveforms using equation (126) for discrete variables and equation (127) for continuous variables in a way that each object has 64 sample points. Note that 64 is a power of 2, therefore, we did not have to pad the waveforms in the feature reduction step.

Our first steps after data cleansing and alignment, and segmenting the signals into windows, was to apply a feature selection algorithm, which first removed redundant features, added the corresponding residuals and then applied the variance method for feature selection. Table 25 shows a subset of variables with high correlation coefficient. Figure 55 shows battery voltage, SATORI # 1 voltage, and their corresponding residual. We can see that the residual value is close to zero for the entire mission. This means SATORI # 1 voltage always follows the battery voltage.

Table 25: Redundant variables.

First feature	Second feature	correlation coefficient
BATTERY VOLTAGE	SATORI1 VOLTAGE	0.9966
BATTERY VOLTAGE	SATORI2 VOLTAGE	0.9966
GYRO TEMP	MOTOR TEMP	0.998
GYRO TEMP	RW TEMP	0.997
COARSE RATE	GYRO COARSE	0.994
IMU PROP ACC1	IMU PROP ACC2	0.999

For each pair variables, v_i and v_j, with correlation coefficient higher than $r_{ij} > 0.99$, we replace one of the variables, say v_j, with a residual, $Res_{ij} = v_i - \frac{v_i.v_j}{v_j.v_j} v_j$. We then use equation (130) to select a minimum set of variables and residuals from each subsystem that represents 95% of the total variance of the subsystem variables. This reduced the total number of features significantly, reducing the total number of features considered for clustering to 36 features listed in the Table 26. Figure 56 shows that Coarse rate #3 and Gyro coarse #3 demonstrate different behaviors in the beginning of the mission and therefore, their corresponding residual has been selected by our feature selection algorithm (see Table 26).

Note that in this case study only one residual is selected for clustering. This means the other generated residuals (such as the residual corresponding to battery voltage and SATORI #1 current shown in Figure 55) had low variances. Each time window contained 64 samples. The Haar wavelet transform was applied to each waveform segment. We select the output of the low pass filter and the high pass filter in level 4 as the set of distinct features representing each segment. Therefore, 64 samples of each variable in a data object were transformed into $\frac{64}{2^4} \times 2 = 8$ features. As it shown in Table 26, the feature selection algorithm selects 36 variables for this case study, therefore, the total number of features for each object is $36 \times 8 = 288$. As we mentioned earlier, we divide the mission into two main phases; 1) initial earth orbital phase 2) lunar orbital phase. We present the results for our anomaly detection for the two phases of the mission in the following subsections. Note that

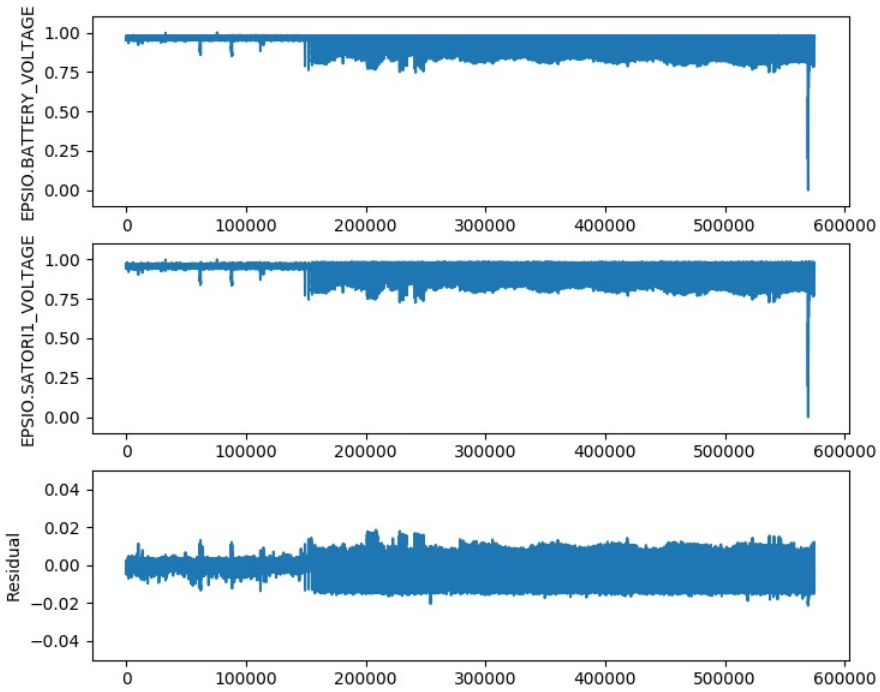

Figure 55: Battery voltage, SATORI #1 voltage, and their corresponding residual

unlike the previous case study, the results of this case study have not been confirmed by the experts in NASA.

7.5.1 Earth orbital phase

We use the Euclidean distance metric to create the dissimilarity matrix of 877×877 object pairs for the earth orbital phase. Then we applied the UPGMA hierarchical clustering algorithm to generate the dendrogram shown in Figure 57. After generating dendrograms we have to choose the level at which to cut the dendrogram to generate the clusters. In this case study, we apply Calinski and Harabasz method [26] and Krzanowski and Lai method [108] and select the highest output among these two methods as the number of clusters. We select the maximum number to make sure we capture all the clusters in the dataset. For the earth orbital phase, the Calinski and Harabasz method and Krzanowski and Lai method find 5 and 4 clusters in the dataset respectively. Therefore, we cut the dendrogram to generate

Table 26: Selected features for each subsystem.

EPS	RW	IMU
PCS PAPI 1 current	Reaction wheel 0 speed	Acceleration 0
PAPI 5 voltage	Reaction wheel 1 speed	Acceleration 1
PAPI 5 current	Reaction wheel 2 speed	IMU temperature
PAPI 6 voltage	Reaction wheel 3 speed	
PAPI 6 current	Gyro 0 temperature	
PAPI 7 voltage	Gyro 1 temperature	
PAPI 7 current	Gyro 2 temperature	
PAPI 8 voltage	Gyro 3 temperature	
PAPI 8 current	Reaction wheel 1 torque	
PCS PAPI 1 HP 7 current	Reaction wheel 2 torque	
SATORI 1 HP 3 current	Gyro coarse 0	
SATORI 1 HP 7 current	Gyro coarse 2	
SATORI 2 HP 1 current	Coarse rate 0	
SATORI 2 HP 6 current	Coarse rate 1	
Battery heater temperature 1	Coarse rate 2	
Battery heater temperature 2	Residual (Coarse rate 3 and Gyro coarse 3)	
Solar array current		

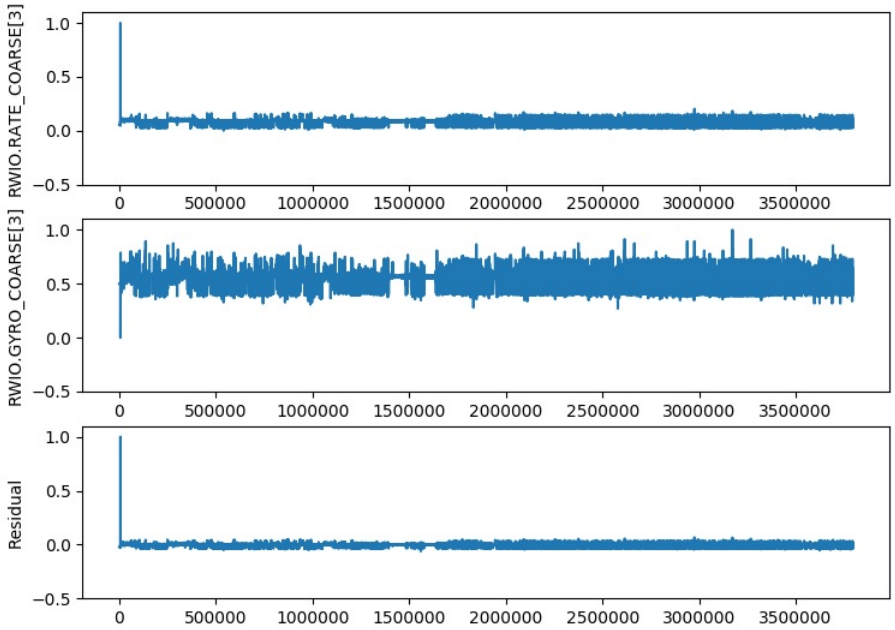

Figure 56: Battery voltage, SATORI #1 voltage, and their corresponding residual

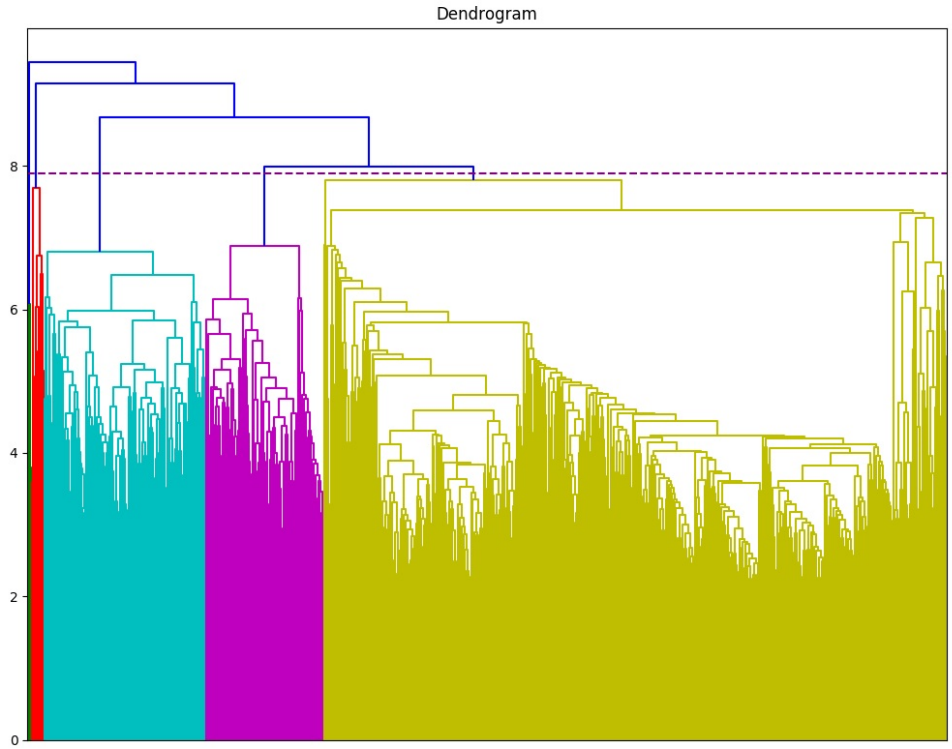

Figure 57: The dendrogram generated by applying the UPGMA hierarchical clustering algorithm to the earth orbital phase of the mission.

5 clusters. Figure 58 shows the clusters for the earth orbital phase. Cluster 5 includes 594

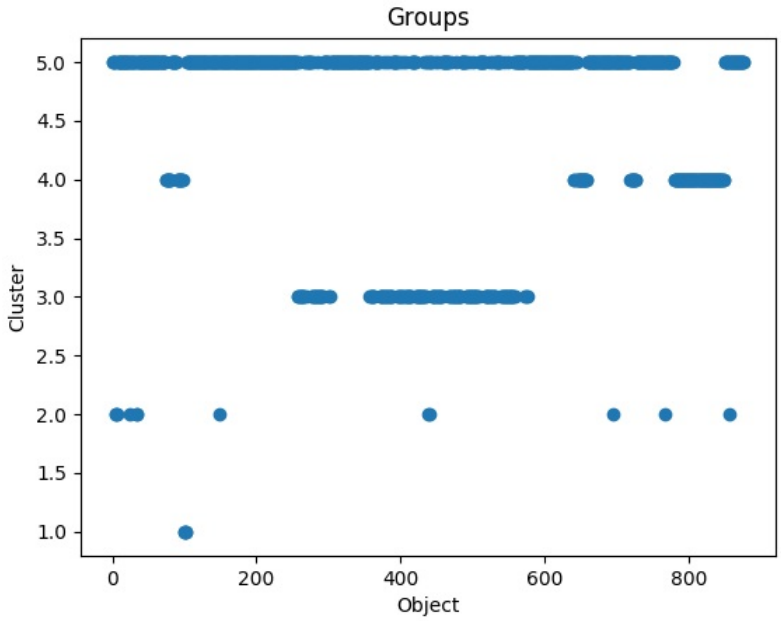

Figure 58: Clusters during the earth orbital phase

time windows and, therefore, it is clear that this cluster represents nominal operations for the earth orbital phase. Clusters 1-4 are discussed below.

Cluster 1: this cluster corresponds to four consequent time intervals. Each time interval in the earth orbital phase is an hour long. The two selected acceleration measurements in the IMU subsystem are the significant features that characterized this cluster. The IMU acceleration measurements during the earth orbital phase are shown in Figure 59. Cluster 1 is marked in this figure. We can see that the accelerations have high amplitude during the cluster. Solar tracker power which is a binary variable is the other significant feature for this cluster. Figure 60 shows the average value of solar tracker power in cluster 5 (normal operation) and cluster 1. We can see that solar tracker has zero average in the beginning of the windows in cluster 1. This means the solar trackers went off during this cluster.

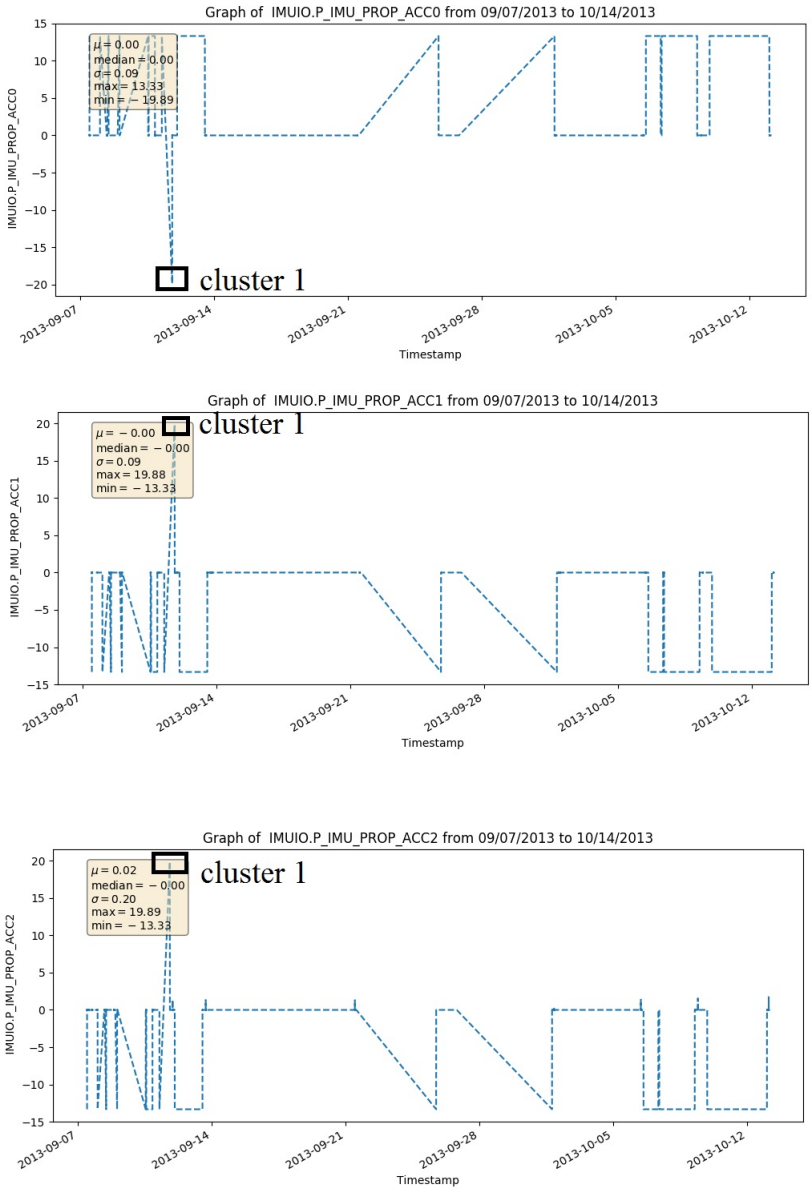

Figure 59: IMU acceleration variables during the earth orbital phase

However, the average is close to one in cluster 5, which means the solar trackers are almost always on in the normal operation. This cluster can represent an anomalous behavior.

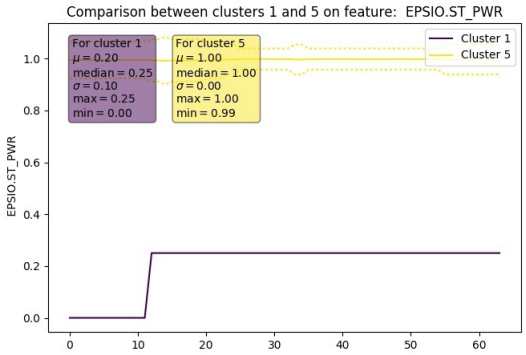

Figure 60: Solar tracker power average for cluster 1 and cluster 5

Cluster 2 includes twelve time intervals that occurred on seven different days during

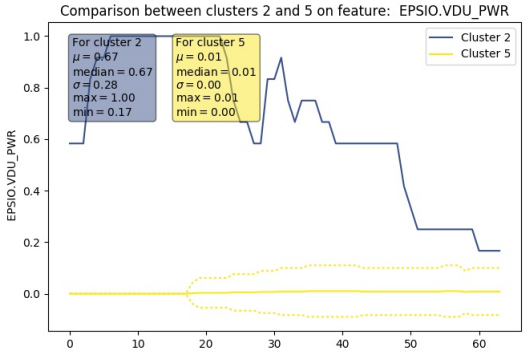

Figure 61: Valve driver unit power average for cluster 2 and cluster 5

the per-orbital phase of the mission. A high pressure current in the SATORI #2 and valve driver unit power switch are the significant features that characterized this cluster. The valve driver unit controls the propulsion subsystem (see Figure 40)). Figure 61 and Figure 62 show the VDU power average and the SATORI #2 HP #6 current average have the identical pattern. Therefore, this cluster can present an operating mode.

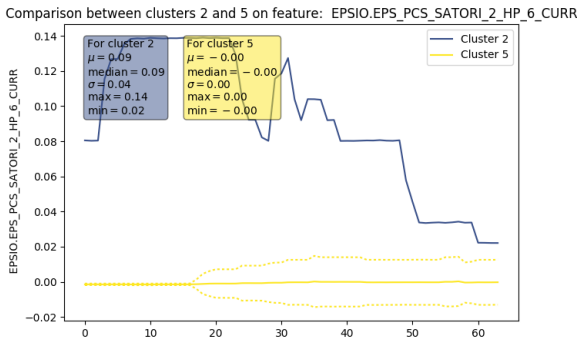

Figure 62: SATORI #2 **HP** #6 **current average for cluster 2 and cluster 5**

Cluster 3 has 154 time windows. SATORI #1 HP #7 current is the significant feature that distinguishes this cluster from the normal operation. Figure 63 shows that SATORI #1 HP #7 current in cluster 3 is significantly lower in average compared to cluster 5. This

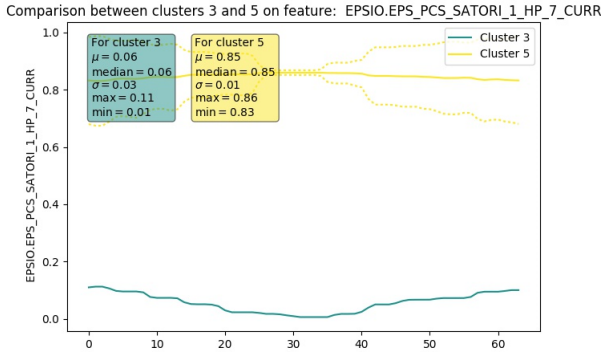

Figure 63: SATORI #1 **HP** #7 **current average for cluster 3 and cluster 5**

cluster seems to represent an operation mode.

Cluster 4 includes 113 objects. The significant features for this cluster are SATORI #2 HP #1 current and solar tracker power. SATORI #2 HP #1 current is significantly lower in this cluster than the normal operation (cluster 5). Moreover, Figure 64 shows that solar tracker power switch average is close to zero for this cluster. This means this switch has been mostly off for the sample points in the cluster. Note that solar tracker power switch

also went off in cluster 2 but it is mostly on in the normal operation. This cluster can present an operating mode corresponding to the solar tracker operation. Table 27 summarizes the

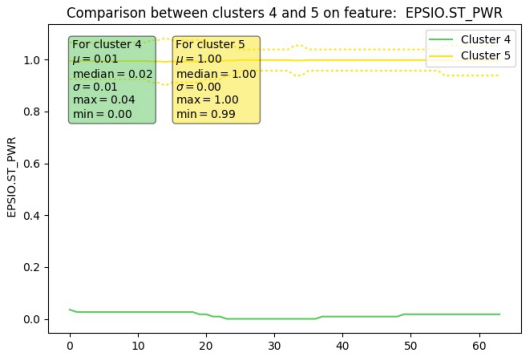

Figure 64: Solar tracker power average for cluster 4 and cluster 5

significant features and our initial assessment of the five clusters in the pre-orbital phase.

Table 27: Summary description of the clusters during the earth orbital phase

Cluster	Objects	Significant Features	Mode
1	4 objects	• IMU Acceleration 0 (low) • IMU Acceleration 1 (high) • IMU Acceleration 2 (high)	• Anomaly
2	12 objects	• SATORI 2 HP 6 current (high) • Valve Driver Unit (VDU) Power	• Operation mode
3	154 objects	• SATORI 2 HP 1 current (high)	• Operation mode
4	113 objects	• SATORI 2 HP 1 current (high) • Star Tracker (ST) Power	• Operation mode
5	594 objects	• –	• Normal operation

7.5.2 Lunar orbital phase

The dendrogram for the lunar orbital section of the dataset is shown Figure 65. For the

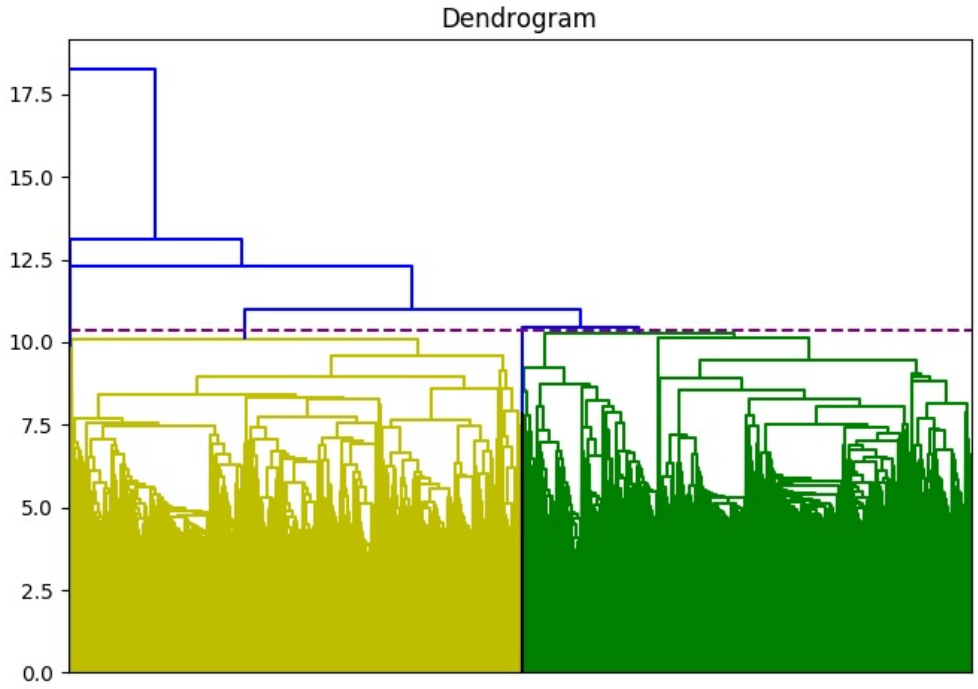

Figure 65: The dendrogram generated by applying the UPGMA hierarchical clustering algorithm. The dashed line represents the chosen threshold distance for cluster formation. The green section of the dendrogram represents normal dark operations, the yellow section represents normal light operations, and the outliers and smaller groups are represented by different colors

lunar orbital phase, the Calinski and Harabasz method finds 7 clusters and the Krzanowski and Lai method finds 9 clusters in the dataset. Therefore, we cut the dendrogram to generate 9 clusters. Figure 66 shows the clusters for the lunar orbital phase. Cluster 7 has 2273 objects and 99.96% of the objects are light windows. Cluster 9 includes 2265 objects and 100% of the objects in this cluster are dark windows. It is clear that cluster 7 represents nominal operations for the light phase and cluster 9 represents nominal operations for the dark phase of the lunar orbital phase of the mission. Clusters 1, 2, 3, and 4 represent the eclipse and dark and light periods after the eclipse when the spacecraft is in the safe mode.

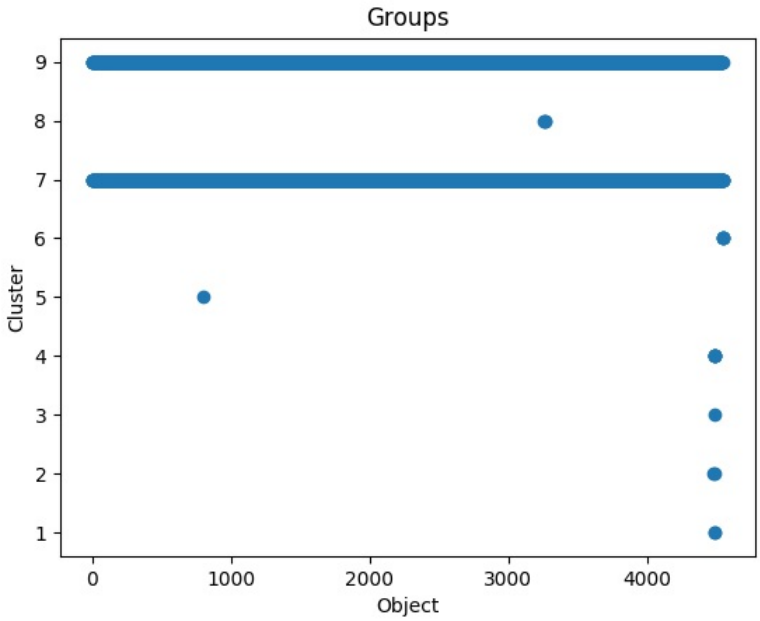

Figure 66: Clusters during the lunar orbital phase

We discussed the eclipse and the safe mode in previous case study in great details. In this section, we discuss the new clusters as follows.

Cluster 5 is a single data object in the light phase (object number 1678 among 5433 objects in the mission including the earth orbital phase and the lunar orbital phase). Note that windows in the light phase are represented by even numbers and odd numbers represent windows in the dark phase. Solar array current is the significant feature in cluster 5. This cluster represents missing data in the dataset. Figure 67 shows the average solar current for cluster 5 and cluster 7. The data is missing in the second half of the time window in cluster 5. The re-sampling step uses linear interpolation to estimate the solar current which obviously does not follows the actual pattern. This cluster represents an anomaly due to missing data.

Cluster 6 includes four consequent dark windows (objects 5423, 5425, 5427 and 5429) at the end of the mission. Figure 68 shows the average IMU temperature in this cluster is significantly higher than the normal dark periods. High SATORI #1 HP #7 current is

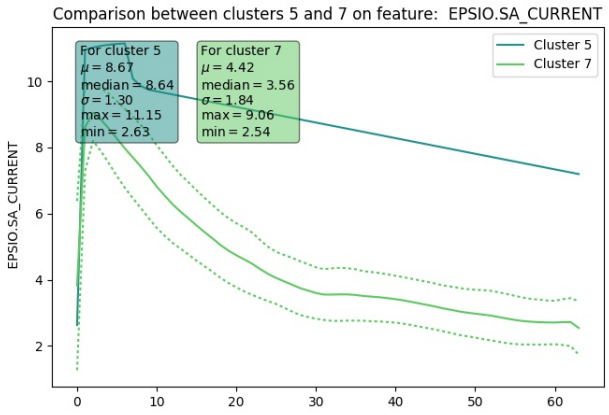

Figure 67: Solar current average for cluster 5 and cluster 7

another significant feature in this cluster. This cluster may represent a special operating condition at the end of the mission before the spacecraft was crashed onto the surface of the moon.

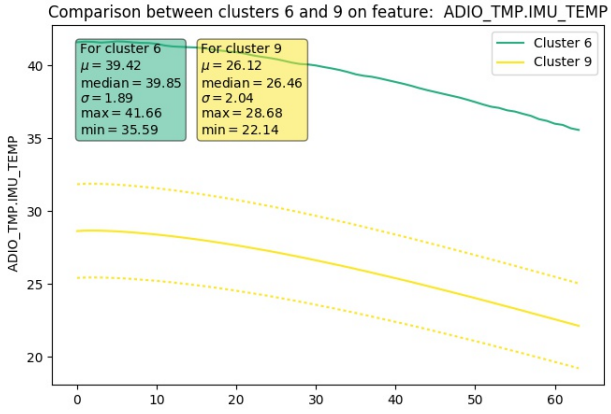

Figure 68: IMU temperature average for cluster 6 and cluster 9

Cluster 8 has three dark windows. Figure 69 shows the average solar array current in this cluster is significantly higher than the normal dark periods at the second half of the time windows. We investigated this cluster and we believe it represents an error in labeling dark and light periods. In the objects of this cluster, the end points of the time windows

are several minutes later than when the dark periods actually ended. Therefore, each dark window has few minutes of light with high solar array current. Table 28 shows the clusters,

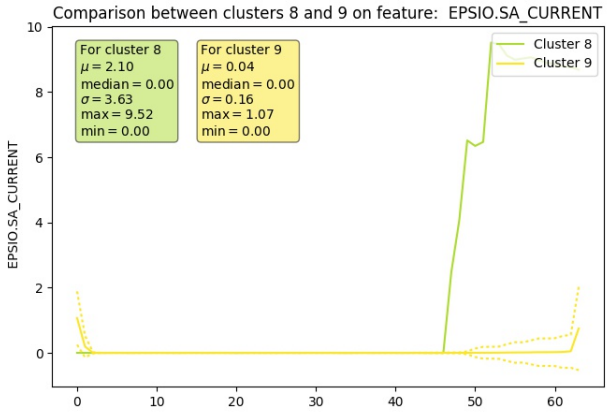

Figure 69: Solar array current average for cluster 8 and cluster 9

their objects and the significant features for each cluster for the lunar orbital phase.

7.6 Summary and Discussion

In this chapter, we have developed a mixed anomaly detection method that applies an unsupervised learning method combined with human-expert support to analyze telemetry data from spacecraft missions. We have described the various steps of the method from the data pre-processing, generation of the feature space, applying a clustering algorithm, determining nominal and outlier processes, associating significant features with the outlier groups, to the consultation with experts resulting in the identification and characterization of special modes of operation as well as anomalous behavior of the system. As the first case study, we applied our approach to analyzing telemetry data from the Electric Power System (EPS) of a recent lunar mission called LADEE. This case study provided interesting results. We were successful in working with mission experts to identify a set of special modes as well as some anomalies that occurred during the mission. The use of significant features

Table 28: Summary description of the clusters during the lunar orbital phase

Cluster	Objects	Significant Features	Detected Mode or Anomaly
1	2 objects (100% light)	• Reaction wheel # 2 temperature (low) • Propulsion red heater	Safe Mode (light windows)
2	2 objects (100% dark)	• Gyro # 1 temperature • Propulsion red heater	Eclipse and a dark window during the safe mode
3	1 objects (100% dark)	• Propulsion red heater • Oxid tank #1 heater	safe mode (dark window)
4	5 object (60% light, 40% light)	• SATORI #1 HP #7 • Oxid tank #2 heater	Safe mode
5	1 object (100% light)	• Solar current	Missing data
6	4 objects (100% dark)	• IMU temperature • SATORI #1 HP #7	End of the mission
7	2273 objects (99.96% light)		Normal light
8	3 objects (100% dark)	• Solar current	Dark period error
9	2265 objects (100% dark)		Normal dark

as well as the projection of the outlier data groups back onto the mission timeline greatly facilitated the mission experts' tasks of identifying and characterizing the special modes and anomalies.

In the second case study, we extended the analysis by including telemetry data from two GNC subsystems in addition to the EPS. To integrate signals from different subsystems, we re-sampled the signal variables in all the subsystems with an unified rate. Moreover, we applied a feature selection method to remove redundant and irrelevant features. The feature selection step helps us to have reasonable computational complexity while including more subsystems in our analysis. This approach shows great promise in generalizing to complex cyber physical systems (CPSs), where well-developed models of the system are not readily available, therefore, operational data has to be used to understand and evaluate system operations, and detect anomalies and outlier behaviors.

CHAPTER VIII

RESEARCH CONTRIBUTIONS AND FUTURE WORK

In this chapter, we review the research contributions and present future work.

8.1 Summary and Research Contributions

One of the primary contributions of this thesis research was the development of derivative-based sensitivity analysis methods using the concepts of detectability and isolability ratios to quantify residuals fault diagnosis performance in the presence of noise and uncertainty. This contribution was discussed in Chapter III of the thesis. Derivative-based sensitivity analysis is easy to implement and computationally efficient and it can be used to generate accurate results for linear and smooth nonlinear systems. We defined global detectability and global isolability ratios to quantify diagnosis performance in stiff nonlinear systems. In comparison with the derivative-based approach, global sensitivity approach increases the computational complexity but generates more robust results for residual quantification.

We combined these measures with a residual selection algorithm to find a residual set that meets pre-specified diagnostic criteria. When the system's trajectory is available, our algorithm divides this trajectory into regions, such that the set of residuals that have sensitivity values above a pre-specified threshold remain the same in a region, but vary across the different regions. The selection of a minimum number of residuals that meet the robustness and sensitivity criteria over all the operating regions is formulated as a BILP optimization problem. For the cases where the system's trajectory is unknown, an efficient dynamic residual selection algorithm is proposed. This algorithm removes residuals when their performance drop below the threshold. They are then replaced by residuals that provide the highest performance ratios in the current region. This guarantees the required performance is maintained for any trajectory.

A second contribution of this thesis in the model-based diagnosis domain was developing two general approaches for distributed fault detection and isolation: 1) MSO-based, and 2) equation-based. The first method provides globally correct diagnosis results and guarantees that the subsystems share the minimum number of measurements, implying that we minimize the communication of measurement streams across subsystems of the global system. Moreover, it is straight forward to extend this approach to robust distributed diagnosis by considering residuals robustness performance in the selection process. However, the total number of MSOs is exponential in terms of the system measurements. This increases the computational cost of the solution.

To avoid the computational complexities of dealing with a large number of MSOs, we develop another distributed diagnosis method based on system equations in this chapter. The second algorithm is computationally efficient. Moreover, it does not use the global model in the design process of the supervisory system. This makes the algorithm suitable for large, complex systems where global systems models are likely to be unavailable or unknown. However, it does not guarantee the minimum communication among subsystems. We compared the diagnosis performances and the computational costs of the proposed algorithms. We then demonstrated through a case study the results obtained from each of the proposed methods.

Our next contribution was developing model based methods for diagnosis of hybrid systems. Our method uses analytic redundancy methods to detect the operating mode of the system even in the presence of system faults. We defined hybrid minimal structurally overdetermined (HMSO) sets for hybrid systems. For residual generation, we develop a greedy search algorithm to select a minimal set of HMSOs that guarantee complete diagnosability in each operating mode. We then used standard methods to generate a residual from each selected HMSO. Our proposed structural approach does not require preenumeration of all possible modes in the diagnoser design step. Therefore, our approach is feasible for hybrid systems with large number of switching elements, implying that the

system can have large number of operating modes. We demonstrated the effectiveness of our approach through a case study.

In the area of data-driven diagnosis methods, we have developed anomaly detection methods using unsupervised learning along with human expert input for analyzing telemetry data from long-duration robotic space missions. We have described the various steps of the method from the data standardization, defining the objects, data re-sampling, feature selection, feature reduction, applying a clustering algorithm, determining nominal and outlier processes, associating significant features with the outlier groups, to the consultation with experts resulting in the identification and characterization of special modes of operation as well as anomalous behavior of the system. The case study showed that our proposed approach can generate promising results for one-of space missions, where well-developed models of the system and labeled historical data of previous missions are not available, and, expert inputs can be used to understand and evaluate system operations, and detect anomalies and outlier behaviors.

8.2 Future Work

In this thesis, we developed model-based approaches for robust FDI, distributed diagnosis, and mode detection and FDI in hybrid systems. We also developed a data-driven unsupervised anomaly detection strategy for long-duration robotic space missions. The model-based methods are computationally efficient. Moreover, it is easy to understand and interpret the diagnosis results when we apply these approaches. However, for complex systems, it is expensive and sometimes infeasible to generate an accurate model for the entire system. When reliable models are not available data-driven diagnosis methods can be used as an alternative solution. However, in many cases, the available data-set does not represent all the operating modes of a system. This could include normal and fault modes. Lack of sufficient historical data makes accurate data-driven diagnosis challenging.

When model and data are incomplete, a combined diagnosis approach that uses historical data and the available model equations in an unified framework can generate the best diagnosis results. By combining model-based diagnosis and data-driven anomaly detection methods, we may be able to detect and isolate faults that was not possible with pure model-based or data-driven methods. Several researchers have combined model-based diagnosis with data-driven approaches to achieve better diagnosis performances [171]. Howevere, these combined solutions are typically limited to specific cases and, it is not straightforward to generalize the solutions. Moreover, they often assume complete model and complete historical data are available. In future work, we will develop a general framework to build crossover solutions that integrate techniques from data-driven and model-based communities in a way that the combined solution operates with incomplete models and limited historical data and generates more accurate results.

REFERENCES

[1] N. Abe, B. Zadrozny, and J. Langford. Outlier detection by active learning. In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM Press, New York, NY, USA, 504–509, 2006.

[2] G. A. Ackerson and K. S. Fu. On state estimation in switching environments. *IEEE Trans. Automat. Contr., vol. AC-15, pp. 10-17*, 1970.

[3] E. Alcorta-Garcia and P. Frank. Deterministic nonlinear observer-based approaches to fault diagnosis : A survey. *Control Engineering Practice 5.5*, pages 663–670, 1997.

[4] C. Arackaparambil, S. Bratus, J. Brody, and A. Shubina. Distributed monitoring of conditional entropy for anomaly detection in streams. In Parallel and Distributed Processing, Workshops and Phd Forum (IPDPSW), 2010 IEEE International Symposium on, pp. 1-8. IEEE, 2010.

[5] J. Armengol, A. Bregón, T. Escobet, E. Gelso, M. Krysander, M. Nyberg, X. Olive, B. Pulido, and L. Travé-Massuyès. Minimal structurally overdetermined sets for residual generation: A comparison of alternative approaches. *Proceedings of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, SAFEPROCESS09, 2009.*

[6] M. J. Aslin and G. J. Patton. Central maintenance computer system and fault data handling method. *Patent US 4 943 919, 07 24*, 1990.

[7] M. Babaali and M. Egerstedt. Observability of switched linear systems. In International Workshop on Hybrid Systems: Computation and Control, pp. 48-63. Springer Berlin Heidelberg, 2004.

[8] D. Barbara, N. Wu, and S. Jajodia. Detecting novel network intrusions using bayes estimators. *In Proceedings of the 2001 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics*, pages 1–17, 2001.

[9] M. Basseville. Distance measures for signal processing and pattern recognition. *Signal processing*, 18(4):349–369, 1989.

[10] M. Basseville. On fault detectability and isolability. *European Journal of Control*, 7(6):625–637, 2001.

[11] M. Basseville and I. V. Nikiforov. Detection of abrupt changes: theory and application. *Englewood Cliffs: Prentice Hall Vol. 104*, 1993.

[12] M. Bayoudh, L. Travt'e-Massuy'es, and X. Olive. On-line analytic redundancy relations instantiation guided by component discrete-dynamics for a class of non-linear

hybrid systems. *48th IEEE Conference on Decision and Control, held jointly with the 28th Chinese Control Conference.*, 2009.

[13] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica, 35(3), 407-427*, 1999.

[14] Y. Bengio, A. C. Courville, and P. Vincent. Unsupervised feature learning and deep learning: A review and new perspectives. *CoRR, abs/1206.5538 1*, 2012.

[15] J. Bigham, D. Gamez, and N. Lu. Safeguarding scada systems with anomaly detection. *In International Workshop on Mathematical Methods, Models, and Architectures for Computer Network Security. Springer Berlin Heidelberg*, pages 171–182, 2003.

[16] G. Biswas, H. Khorasgani, G. Stanje, A. Dubey, S. Deb, and S. Ghoshal. An application of data driven anomaly identification to spacecraft telemetry data. *In Annual Conference of the Prognostics and Health Management Society.*, 2016.

[17] G. Biswas, E. J. Manders, J. Ramirez, N. Mahadevan, and S. Abdelwahed. Online model-based diagnosis to support autonomous operation of an advanced life support system. *Habitation: An International Journal for Human Support Research, 10(1), (2004) 21-38.*

[18] G. Biswas, G. Simon, N. Mahadevan, S. Narasimhan, J. Ramirez, and G. Karsai. A robust method for hybrid diagnosis of complex systems. *In Proceedings of the 5th Symposium on Fault Detection, Supervision and Safety for Technical Processes, pp. 1125-1131.*, 2003.

[19] J. Blesaa, F. Nejjari, and R. Sarrate. Robustness analysis of sensor placement for leak detection and location under uncertain operating conditions. *16th International Water Distribution Systems Analysis Conference (WDSA), At Bari (Italy), 2014.*

[20] D. Blough, S. Sullivan, and G. Masson. . fault diagnosis for sparsely interconnected multiprocessor systems. *In Proc. of FTCS-19, 1989, pp.62-69.*

[21] A. Bregon, C. Alonso, G. Biswas, B. Pulido, and N. Moya. Hybrid systems fault diagnosis with possible conflicts. *In Proceedings of the 22nd International Workshop on Principles of Diagnosis, 2011.*, pages 195–202, 2011.

[22] A. Bregon, G. Biswas, B. Pulido, C. Alonso-Gonzalez, and H. Khorasgani. A common framework for compilation techniques applied applied to diagnosis of linear dynamic systems. *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS*, 44(7), July 2014.

[23] A. Bregon, M. Daigle, I. Roychoudhury, G. Biswas, X. Koutsoukos, and B. Pulido.

An event-based distributed diagnosis framework using structural model decomposition. *Artificial Intelligence*, 210:1–35, 2014.

[24] C. S. Burrus, R. A. Gopinath, and H. Guo. *Introduction to wavelets and wavelet transforms: a primer*. Prentice-Hall, Inc., 1997.

[25] D. Cai, C. Zhang, and X. He. Unsupervised feature selection for multi-cluster data. In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 333–342 ACM, 2010.

[26] T. Calinski and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods 3, no. 1*, pages 1–27, 1974.

[27] Z. L. Carl, Joshua D. and G. Biswas. Modeling and simulation semantics for building large-scale multi-domain embedded systems. *ECMS. 2013.*

[28] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *CM computing surveys (CSUR)*, 41(3):15–72, 2009.

[29] W.-C. Chang. On using principal components before separating a mixture of two multivariate normal distributions. *Journal of the Royal Statistical Society, Series C (Applied Statistics): 32:267–275*, 1983.

[30] J. Chen and R. J. Patton. Robust model-based fault diagnosis for dynamic systems. *Springer Publishing Company, Incorporated, 2012.*

[31] J. Chen, R. J. Patton, and H.-Y. Zhang. Design of unknown input observers and robust fault detection filters. *International Journal of Control 63.1 (1996): 85-105.*

[32] X. Chen, Y. Ye, X. Xu, and J. Z. Huang. A feature group weighting method for subspace clustering of high-dimensional data. *Pattern Recognition 45, no. 1, 434–446*, 2012.

[33] V. Cocquempot, T. E. Mezyani, and M. Staroswiecki. Fault detection and isolation for hybrid systems using structured parity residuals. In Control Conference, 5th Asian, vol. 2, pp. 1204-1212. IEEE, 2004.

[34] V. Cocquempot, M. Staroswieck, and T. E. Mezyani. Switching time estimation and fault detection for hybrid system using structured parity residuals. *IFAC Conference Safeprocess, pages 2045-2055, USA*, 2003.

[35] M. J. Daigle, X. D. Koutsoukos, and G. Biswas. Distributed diagnosis in formations of mobile robots. *Robotics, IEEE Transactions*, 23(2):353–369, 2007.

[36] M. J. Daigle, I. Roychoudhury, and A. Bregon. Qualitative event-based diagnosis applied to a spacecraft electrical power distribution system. *Control Engineering Practice 38, (2015): 75-91.*

[37] W. H. Day and H. Edelsbrunner. Efficient algorithms for agglomerative hierarchical clustering methods. *Journal of classification*, 1(1):7–24, 1984.

[38] R. C. de Amorim. A survey on feature weighting based k-means algorithms. *Journal of Classification 33, no. 2, 210–242*, 2016.

[39] I. V. de Bessa, R. M. Palhares, M. F. S. V. D Angelo, and C. E. C. Filho. Data-driven fault detection and isolation scheme for a wind turbine benchmark. *Renewable Energy*, 87:634–645, 2016.

[40] S. Deb, A. Mathur, P. K. Willett, and K. R. Pattipati. Decentralized real-time monitoring and diagnosis. *In Systems, Man, and Cybernetics. IEEE International Conference*, 3:2998–3003, 1998.

[41] P. Derler, E. A. Lee, and A. S. Vincentelli. Modeling cyber–physical systems. *Proceedings of the IEEE*, 100(1):13–28, 2012.

[42] M. A. Djeziri, R. Merzouki, B. O. Bouamama, and G. Dauphin-Tanguy. Robust fault diagnosis by using bond graph approach. *Mechatronics, IEEE/ASME Transactions on, 2007.*

[43] E. A. Domlan, J. Ragot, and D. Maquin. Active mode estimation for switching systems. American Control Conference, pp. 1143-1148. IEEE, 2007.

[44] E. P. Duarte Jr and T. Nanya. A hierarchical adaptive distributed system-level diagnosis algorithm. *Computers, IEEE Transactions*, 47(1):34–45, 1998.

[45] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification*. John Wiley and Sons, 2012.

[46] A. L. Dulmage and N. S. Mendelsohn. Coverings of bipartite graphs. *Canadian Journal of Mathematics*, 10(4):516–534, 1958.

[47] D. Dustegör, V. Cocquempot, and M. Staroswiecki. Structural analysis for residual generation: Towards implementation. *Control Applications, Proceedings of the IEEE International Conference on. Vol. 2. IEEE*, 2004.

[48] R. C. Elphic, G. T. Delory, E. J. Grayzeck, A. Colaprete, M. Horanyi, P. Mahaffy, B. Hine, and J. S. D. Boroson. The lunar atmosphere and dust environment explorer (ladee): T-minus one year and counting. *2012, URL:* http://www.lpi.usra.edu/meetings/leag2012/presentations/Elphic.pdf.

[49] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie. High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning. *Pattern Recognition 58*, pages 121–134, 2016.

[50] D. Eriksson, E. Frisk, and M. Krysander. A method for quantitative fault diagnosability analysis of stochastic linear descriptor models. *Automatica, 2013.*

[51] D. Eriksson, E. Frisk, and M. Krysander. A sequential test selection algorithm for fault isolation. *Proceedings of the 10th European Workshop on Advanced Control and Diagnosis, ACD 2012, Copenhagen, Denmark.*

[52] D. Eriksson and C. Sundström. Sequential residual generator selection for fault detection. *In: Proceedings of 13th European Control Conference (ECC2014), Strasbourg, France, 2014.*

[53] L. Ertoz, M. Steinbach, and V. Kumar. Finding topics in collections of documents: A shared nearest neighbor approach. Clustering and Information Retrieval 11, 83–103, 2003.

[54] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In Kdd, vol. 96, no. 34, pp. 226–231, 1996.

[55] J. Farrell and M. Polycarpou. Adaptive approximation based control: Unifying neural, fuzzy, and traditional adaptive approximation approaches. *Hoboken, NJ: Wiley, 2006*.

[56] R. M. Ferrari, T. Parisini, and M. M. Polycarpou. Distributed fault diagnosis with overlapping decompositions: an adaptive approximation approach. *Automatic Control, IEEE Transactions on, 54(4), 794-799, 2009.*

[57] R. M. Ferrari, T. Parisini, and M. M. Polycarpou. Distributed fault detection and isolation of large-scale discrete-time nonlinear systems: An adaptive approximation approach. *Automatic Control, IEEE Transactions on*, 57(2):275–290, 2012.

[58] V. Flaugergues, V. Cocquempot, M. Bayart, and M. Pengov. Structural analysis for fdi: a modified, invertibility-based canonical decomposition. In *Proceedings of the 20th International Workshop on Principles of Diagnosis, DX09*, pages 59–66, 2009.

[59] V. Flaugergues, V. Cocquempot, M. Bayart, and M. Pengov. Structural analysis for fdi: a modified, invertibility-based canonical decomposition. In *Proceedings of the 20th International Workshop on Principles of Diagnosis, DX09*, pages 59–66. Citeseer, 2009.

[60] E. W. Forgy. Cluster analysis of multivariate data: Efficiency vs. interpretability of classification. Biometrics, 21:768, 1965.

[61] P. M. Frank. Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy: A survey and some new results. *Automatica, 1990.*

[62] P. M. Frank and X. Ding. Frequency domain approach to optimally robust residual

generation and evaluation for model-based fault diagnosis. *Automatica, 1994*.

[63] P. M. Frank and X. Ding. Survey of robust residual generation and evaluation methods in observer-based fault detection systems. *Journal of process control, 7(6)*, pages 403–424, 1997.

[64] G. Franze and D. Famularo. A robust fault detection filter for polynomial nonlinear systems via sum-of-squares decompositions. *Systems and Control Letters 61.8*, pages 839–848, 2012.

[65] D. Freitas, R. Dearden, and F. Hutter. Efficient on-line fault diagnosis for non-linear systems. *Diagnosis by a waiter and a mars explorer. Proceedings of the IEEE, 92(3), 455-468.*, 2004.

[66] E. Frisk. Residual generation for fault diagnosis. *Linkopings Universitet. Department of Computer and Information Science*, 2001.

[67] E. Frisk, A. Bregon, J. Aslund, M. Krysander, B. Pulido, and G. Biswas. Diagnosability analysis considering causal interpretations for differential constraints. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 42(5):1216–1229, 2012.

[68] E. Frisk, M. Krysander, and D. Jung. A toolbox for analysis and design of model based diagnosis systems for large scale models. IFAC World Congress, Toulouse, France, 2017.

[69] E. Frisk and M. Nyberg. A minimal polynomial basis solution to residual generation for fault diagnosis in linear systems. *Automatica 37, no. 9 (2001): 1417-1424.*

[70] Y. Gao, T. Yang, N. Xing, and M. Xu. Fault detection and diagnosis for spacecraft using principal component analysis and support vector machines. *In Industrial Electronics and Applications (ICIEA), 2012 7th IEEE Conference*, pages 1984–1988, 2012.

[71] M. N. Gaonkar and K. Sawant. Autoepsdbscan: Dbscan with eps automatic for large dataset. *International Journal on Advanced Computer Theory and Engineering 2, no. 2, 11-16.*, 2013.

[72] V. Garcia-Font, C. Garrigues, and H. Rifa-Pous. A comparative study of anomaly detection techniques for smart city wireless sensor networks. *Sensors 16, no. 6, 868.*, 2016.

[73] J. Gertler. *Fault detection and diagnosis in engineering systems*. CRC press, 1998.

[74] J. Gertler and D. Singer. A new structural framework for parity equation based failure detection and isolation. *Automatica 26, no. 2*, pages 381–388, 1990.

[75] U. Grömping. Estimators of relative importance in linear regression based on variance decomposition. *The American Statistician.*, 2012.

[76] S. M. Hall. Rules of thumb for chemical engineers. *Butterworth-Heinemann*, 2012.

[77] H.-S. Ham and M.-J. Choi. Analysis of android malware detection performance using machine learning classifiers. pages 490–495, 2013.

[78] P. Hanlon and P. Maybeck. Multiple-model adaptive estimation using a residual correlation kalman filter bank. *IEEE Trans. Aerosp. Electron. Syst., vol. 36, pp. 393-406, Apr*, 2000.

[79] J. A. Hartigan. Clustering algorithms. *New York: Wiley*, 1975.

[80] S. Hawkins, H. He, G. Williams, and R. Baxter. Outlier detection using replicator neural networks. *In International Conference on Data Warehousing and Knowledge Discovery, pp. 170-180. Springer Berlin Heidelberg*, pages 113–123, 2002.

[81] X. He, D. Cai, and P. Niyogi. Laplacian score for feature selection. *In Advances in neural information processing systems*, pages 507–514, 2006.

[82] W. P. Heemels, B. D. Schutter, and A. Bemporad. Equivalence of hybrid dynamical models. *Automatica 37, no. 7, 1085-1091*, 2006.

[83] B. Hine, S. Spremo, M. Turner, and R. Caffrey. The lunar atmosphere and dust environment explorer (ladee) mission. In *IEEE Aerospace Conference*. IEEE, 2010.

[84] V. J. Hodge and J. Austin. A survey of outlier detection methodologies. *Artificial intelligence review 22, no. 2*, pages 85–126, 2004.

[85] M. W. Hofbaur and B. C. Williams. Hybrid estimation of complex systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 34, no. 5*, pages 2178–2191, 2004.

[86] J. Z. Huang, M. K. Ng, H. Rong, and Z. Li. Automated variable weighting in k-means type clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence 27, no. 5, 657–668*, 2005.

[87] Z. Huang. Extensions to the k-means algorithms for clustering large data sets with categorical values. Data Ming and Knowledge Discovery, vol. 2, no. 3, pp. 283-304, 1998.

[88] L. J. Hubert and J. R. Levin. A general statistical framework for assessing categorical clustering in free recall. *Psychological bulletin 83, no. 6*, 1976.

[89] F. Hutter and R. Dearden. Efficient on-line fault diagnosis for non-linear systems. *Proceedings of the 7th International Symposium on Artificial Intelligence, Robotics*

and Automation in Space., 2003.

[90] R. Iman and S. Hora. A robust measure of uncertainty importance for use in fault tree system analysis. *Risk Anal. 10, (1990) 401-406.*

[91] R. Isermann. Fault diagnosis of machines via parameter estimation and knowledge processing– tutorial paper. *Automatica, 29(4)*, pages 815–835, 1993.

[92] R. Isermann. Supervision, fault-detection and fault-diagnosis methods– an introduction. *Control engineering practice, 5*, pages 639–652, 1997.

[93] H. Ishwaran. Variable importance in binary regression trees and forests. *Electronic Journal of Statistics 1: 519-537.*, 2007.

[94] A. K. Jain and R. C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[95] N. Japkowicz, C. Myers, and M. Gluck. A novelty detection approach to classification. In IJCAI, vol. 1, pp. 518-523., 1995.

[96] L. Jing, M. K. Ng, and J. Z. Huang. An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data. *IEEE Transactions on knowledge and data engineering 19, no. 8*, 2007.

[97] C. Jinran, S. Kher, and A. Somani. Distributed fault detection of wireless sensor networks. *Proceedings of the 2006 workshop on Dependability issues in wireless ad hoc networks and sensor networks ACM.*

[98] P. Kabore, M. Staroswiecki, and H. Wang. On parity space and observer-based approaches for fdi in state affine systems. *Proceedings of the 38 Conference on Decision and Control Phoenix, Arizona USA, December*, pages 2910–2911, 1999.

[99] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data. An Introduction to Cluster Analysis*. Wiley-Interscience, New York, 1990.

[100] C. Keliris, M. M. Polycarpou, and T. Parisini. A robust nonlinear observer based approach for distributed fault detection of input output interconnected systems. *Automatica 53*, pages 408–415, 2015.

[101] E. Keogh, S. Lonardi, and C. A. Ratanamahatana. Towards parameter-free data mining. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 206-215. ACM, 2004.

[102] H. Khorasgani and G. Biswas. Structural fault detection and isolation in hybrid systems. *Manuscript submitted for publication in Transactions on Automation Science and Engineering (T-ASE)*, 2017.

[103] H. Khorasgani, D. E. Jung, and G. Biswas. Structural approach for distributed fault detection and isolation. IFAC-PapersOnLine 48, no. 21 : 72-77, 2015.

[104] H. Khorasgani, D. E. Jung, G. Biswas, E. Frisk, and M. Krysander. Off-line robust residual selection using sensitivity analysis. International Workshop on Principles of Diagnosis (DX-14), Graz, Austria, 2014.

[105] B. K. Korte, Bernhard andVygen. Combinatorial optimization. *Heidelberg: Springer-Verlag, 2002*.

[106] M. Krysander, J. Aslund, and M. Nyberg. An efficient algorithm for finding minimal overconstrained subsystems for model-based diagnosis. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 38(1):197–206, 2008.

[107] M. Krysander and E. Frisk. Sensor placement for fault diagnosis. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 38(6):1398–1410, 2008.

[108] W. J. Krzanowski and Y. T. Lai. A criterion for determining the number of groups in a data set using sum of squares clustering. *Biometrics, 44*, pages 23–34, 1988.

[109] A. H. Land and A. G. Doig. An automatic method of solving discrete programming problems. *Econometrica: Journal of the Econometric Society*, pages 497–520, 1960.

[110] P. E. Lanigan, S. Kavulya, and P. Narasimhan. Diagnosis in automotive systems: A survey. *Technical Report CMU-PDL-11-110, Carnegie Mellon University PDL*, 2011.

[111] W. Lee and D. Xiang. Information-theoretic measures for anomaly detection. pages 130–143, 2001.

[112] J. B. Leger, B. Iung, A. Ferro De Beca, and J. Pinoteau. An innovative approach for new distributed maintenance system: application to hydro power plants of the remafex project. *Computers in industry*, 38(2):131–148, 1999.

[113] R. Levy, S. A. Arogeti, and D. Wang. An integrated approach to mode tracking and diagnosis of hybrid systems. *Industrial Electronics, IEEE Transactions on 61, no. 4: 2024-2040.*, 2014.

[114] L. Li, S. Das, R. J. Hansman, R. Palacios, and A. N. Srivastava. Analysis of flight data using clustering techniques for detecting abnormal operations. *Journal of Aerospace Information Systems*, 12(9):587–598, 2015.

[115] L. Li, M. Gariel, R. J. Hansman, and R. Palacios. Anomaly detection in onboard-recorded flight data using cluster analysis. In Digital Avionics Systems Conference (DASC) IEEE/AIAA 30th, pp. 4A4-1, 2011.

[116] X. R. Li. Multiple model estimation with variable structure part ii: Model-set adaption. *IEEE Trans. Automatic Control, vol. 45, pp. 2047-2060, Nov.*, 2000.

[117] X. R. Li and Y. Bar-Shalom. Multiple-model estimation with variable structure. *IEEE Trans. Automat. Contr., vol. 41, pp. 478-493*, 1996.

[118] X. R. Li, X. Zhi, and Y. Zhang. Multiple model estimation with variable structure part iii: Model-group switching algorithm. *IEEE Trans. Aerosp. Electron. Syst., vol. 35, pp. 225-241*, 1999.

[119] Y. Li, M. Dong, and J. Hua. Localized feature selection for clustering. Pattern Recognition Letters 29, no. 1, 10–18, 2008.

[120] J. Liu, J. Zhang, M. Palumbo, and C. Lawrence. Bayesian clustering with variable and transformation selections. Bayesian statistics, no. 7, 249–275, 20083.

[121] C. B. D. W. Low, S. Arogeti, and J. B. Zhang. Causality assignment and model approximation for quantitative hybrid bond graph-based fault diagnosis. in Proc. 17th IFAC World Congr., Seoul, Korea, pp. 10 522-10 527., 2008.

[122] C. B. D. W. Low, S. Arogeti, and J. B. Zhang. Monitoring ability analysis and qualitative fault diagnosis using hybrid bond graph. in Proc. 17th IFAC World Congr., Seoul, Korea, pp. 10 522-10 527., 2008.

[123] J. Ma, L. Sun, H. Wang, Y. Zhang, and U. Aickelin. Supervised anomaly detection in uncertain pseudoperiodic data streams. *ACM Transactions on Internet Technology (TOIT) 16, no. 1*, 2016.

[124] D. L. Mack. *Anomaly Detection from Complex Temporal Sequences in Large Data*. PhD thesis, Vanderbilt University, 2013.

[125] D. L. Mack, G. Biswas, and X. D. Koutsoukos. Learning bayesian network structures to augment aircraft diagnostic reference models. *IEEE Transactions on Automation Science and Engineering 14, no. 1*, pages 358–369, 2017.

[126] T. S. Madhulatha. An overview on clustering methods. *arXiv preprint arXiv:1205.1117*, 2012.

[127] J.-F. Magni and P. Mouyon. On residual generation by observer and parity space approaches. *Automatic Control, IEEE Transactions on*, 39(2):441–447, 1994.

[128] M. Markou and S. Singh. Novelty detection: a review – part 2: neural network based approaches. *Signal Process. 83 (12)*, pages 2499–2521, 2003.

[129] M. A. Massoumnia. A geometric approach to the synthesis of failure detection filters. *Automatic Control, IEEE Transactions on 31.9 (1986): 839-846*.

[130] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan. Interacting multiple model methods in target tracking: a survey. *IEEE transactions on aerospace and electronic systems*, 34(1):103–123, 1998.

[131] Melnyk, A. Banerjee, B. Matthews, and N. Oza. Semi-markov switching vector autoregressive model-based anomaly detection in aviation systems. *arXiv preprint arXiv:1602.06550*, 2016.

[132] G. W. Milligan and M. C. Cooper. An examination of procedures for determining the number of clusters in a data set. *Psychometrika 50.2*, pages 159–179, 1985.

[133] G. W. Milligan and M. C. Cooper. A study of standardization of variables in cluster analysis. *Journal of classification*, 5(2):105–119, 1988.

[134] J. E. Mitchell. Branch-and-cut algorithms for combinatorial optimization problems. *Handbook of applied optimization*, pages 65–77, 2002.

[135] F. d. Mortain, A. Subias, L. Travé-Massuyès, and V. d. Flaugergues. Towards active diagnosis of hybrid systems leveraging multimodel identification and a markov decision process. *IFAC-PapersOnLine 48, no. 21: 171-176*, 2015.

[136] P. J. Mosterman and G. Biswas. A theory of discontinuities in physical system models. Journal of the Franklin Institute, 335(3), 401-439, 1998.

[137] S. Narasimhan and G. Biswas. Model-based diagnosis of hybrid systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part A 37, no. 3, 348-361.*, 2007.

[138] O. Niggemann, G. Biswas, J. S. Kinnebrew, H. Khorasgani, S. Volgmann, and A. Bunte. Data-driven monitoring of cyber-physical systems leveraging on big data and the internet-of-things for diagnosis and control. *26th International Workshop on Principles of Diagnosis, Paris, France*, 2015.

[139] R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. on Automatic Control, 49(9):1520?1533, Sep. 2004.*

[140] R. J. Patton and J. Chen. Observer-based fault detection and isolation: robustness and applications. *Control Engineering Practice 5.5*, pages 671–682, 1997.

[141] R. J. Patton, P. M. Frank, and R. N. Clark. Issues of fault diagnosis for dynamic systems. *Springer Science and Business Media*, 2013.

[142] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence 27, no. 8, 1226–1238*, 2005.

[143] R. Perez, V. Puig, J. Pascual, J. Quevedo, E. Landeros, and A. Peralta. Methodology for leakage isolation using pressure sensitivity analysis in water distribution networks. *Control Engineering Practic, 2011.*

[144] L. R. Petzold and U. M. Ascher. Computer methods for ordinary differential equations and differential-algebraic equations. *Siam, 1998.*

[145] M. A. Pimentel, D. A. Clifton, and L. Tarassenko. A review of novelty detection. *Signal Processing 99*, pages 215–249, 2014.

[146] B. Podgursky, G. Biswas, and X. Koutsoukos. Efficient tracking of behavior in complex hybrid systems via hybrid bond graphs. In *In Proceedings of the Annual Conference of the Prognostics and Health Management Society, PHM10. Portland, OR, USA*, 2010.

[147] B. Pulido and C. A. González. Possible conflicts: a compilation technique for consistency-based diagnosis. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 34.5*, pages 2192–2206, 2004.

[148] Y. P. Raykov, A. Boukouvalas, F. Baig, and M. A. Little. What to do when k-means clustering fails: a simple yet principled alternative algorithm. *PloS one 11, no. 9*, 2016.

[149] I. Rish, M. Brodie, S. Ma, N. Odintsova, A. Beygelzimer, G. Grabarnik, and K. Hernandez. Adaptive diagnosis in distributed systems. *Neural Networks, IEEE Transactions*, 16(5):1088–1109, 2005.

[150] I. Roychoudhury, G. Biswas, and X. Koutsoukos. Designing distributed diagnosers for complex continuous systems. *Automation Science and Engineering, IEEE Transactions on*, 6(2):277–290, 2009.

[151] I. Roychoudhury, M. J. Daigle, G. Biswas, and X. Koutsoukos. Efficient simulation of hybrid systems: A hybrid bond graph approach. *Simulation*, 87(6):467–498, 2011.

[152] G. Rätsch, B. Schölkopf, S. Mika, and K. R. Müller. Svm and boosting: One class. *GMD-Forschungszentrum Informationstechnik*, 2000.

[153] L. B. Ruiz, I. Siqueira, L. B. Oliveira, H. C. Wong, J. M. S. Nogueira, and A. A. F. Loureiro. Fault management in event-driven wireless sensor networks. *MSWiM?04, October 4-6, 2004, Venezia, Italy.*

[154] A. Samantaray, K. Medjaher, B. Bouamama, and M. G. Staroswiecki, Dauphin-Tanguy. Diagnostic bond graphs for online fault detection and isolation. *Simulation Modelling Practice Theory*, 14(3):237Ű262, Apr 2006.

[155] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. C. Teneketzis.

Failure diagnosis using discrete-event models. *IEEE Trans. Control Syst. Technol., vol. 4, no. 2, pp. 105–124*, 1996.

[156] S. Sankararaman and S. Mahadevan. Separating the contributions of variability and parameter uncertainty in probability distributions. *Reliability Engineering and System Safety 112: 187-199.*, 2013.

[157] R. Sarrate, P. Vicenc, E. Teresa, and R. Albert. Optimal sensor placement for model-based fault detection and isolation. pages 2584–2589. 46th IEEE Conference In Decision and Control, 2007.

[158] B. Scholkopf, K.-K. Sung, C. J. Burges, F. Girosi, P. Niyogi, T. Poggio, and V. Vapnik. Comparing support vector machines with gaussian kernels to radial basis function classifiers. *IEEE transactions on Signal Processing 45, no. 11*, pages 2758–2765, 1997.

[159] M. Schwabacher, M. Feather, and L. Markosian. Verification and validation of advanced fault detection, isolation and recovery for a nasa space system. *In Proc. Int. Symp. on Software Reliability Engineering.*, 2008.

[160] R. Serban and A. C. Hindmarsh. Cvodes, the sensitivity-enabled ode solver in sundials. *Proceedings of the 5th International Conference on Multibody Systems, Nonlinear Dynamics and Control, Long Beach, CA, 2005.*

[161] S. K. Shum, J. F. Davis, W. F. Punch III, and B. Chandrasekaran. An expert system approach to malfunction diagnosis in chemical plants. *Computers and chemical engineering*, 12(1):27–36, 1988.

[162] C. Spitzer. Honeywell primus epic aircraft diagnostic and maintenance. *Digital Avionics Handbook*, page 22Ű23, 2007.

[163] M. Staroswiecki and G. Comtet-Varga. Analytical redundancy relations for fault detection and isolation in algebraic dynamic systems. *Automatica 37.5*, 38(6):687–699, 2001.

[164] G. Strang. Wavelet transforms versus fourier transforms. *Bulletin of the American Mathematical Society*, 28(2):288–305, 1993.

[165] C. A. Sugar and G. M. James. Finding the number of clusters in a dataset. *Journal of the American Statistical Association*, 2011.

[166] C. Svärd and M. Nyberg. Residual generators for fault diagnosis using computation sequences with mixed causality applied to automotive systems. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, 40(6), 1310-1328.*, 2010.

[167] C. Svärd, M. Nyberg, E. Frisk, and M. Krysander. Automotive engine fdi by application of an automated model-based and data-driven design methodology. *Control Engineering Practice,*, 2013.

[168] C. Svärd, M. Nyberg, and J. Stoustrup. Automated design of an fdi system for the wind turbine benchmark. *JCSE Journal of Control Science and Engineering*, 2012.

[169] T. Szarka. Structural, behavioral and functional modeling of cyber-physical systems. *PhD diss., Vanderbilt University, 2011.*, 2011.

[170] T. Threepak and A. Watcharapupong. Web attack detection using entropy-based analysis. In Information Networking (ICOIN), 2014 International Conference on, pp. 244-247. IEEE., 2014.

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

[172] F. D. Torrisi and A. Bemporad. Hysdel-a tool for generating computational hybrid models for analysis and synthesis problems. *IEEE transactions on control systems technology 12, no. 2, 235–249*, 2004.

[173] L. Travé-Massuyes, T. Escobet, and X. Olive. Diagnosability analysis based on component-supported analytical redundancy relations. 2006.

[174] J. Tugnait. Detection and estimation for abruptly changing systems. *Automatica, vol. 18, pp. 607-615*, 1982.

[175] A. Van Griensven, T. Meixner, S. Grunwald, T. Bishop, M. Diluzio, and R. Srinivasan. A global sensitivity analysis tool for the parameters of multi-variable catchment models. *Journal of hydrology 324, no. 1 (2006)*, pages 10–23, 2006.

[176] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri. A review of process fault detection and diagnosis: Part i: Quantitative model-based methods. *Computers and chemical engineering 27, no. 3*, pages 293–311, 2003.

[177] N. Viswanadham and R. Srichander. Fault detection using unknown input observers. *Control Theory and Advanced Technology, 3(2)*, pages 91–101, 1987.

[178] N. Viswanadham, J. H. Taylor, and E. C. Luce. A frequency-domain approach to failure detection and isolation with application to ge 21 turbine engine control systems. *Control–Theory and Advanced Technology 3, no. 1 (1987): 45–72.*

[179] D. B. West. *Introduction to graph theory*. Prentice Hall, 2001.

[180] D. C. Whitley, M. G. Ford, and D. J. Livingstone. Unsupervised forward selection: a method for eliminating redundant variables. *ournal of Chemical Information and*

Computer Sciences 40, no. 5 (2000): 1160-1168.

[181] D. M. Witten and R. Tibshirani. A framework for feature selection in clustering. *Journal of the American Statistical Association 105, no. 490, 713-726*, 2010.

[182] L. A. Wolsey and G. L. Nemhauser. *Integer and combinatorial optimization*. John Wiley and Sons, 2014.

[183] D. Wulsin, J. Blanco, R. Mani, and B. Litt. Semi-supervised anomaly detection for eeg waveforms using deep belief nets. *In Machine Learning and Applications (ICMLA), Ninth International Conference on, IEEE*, pages 436–441, 2010.

[184] M. Yan. *Methods of determining the number of clusters in a data set and a new clustering criterion*. PhD thesis, Virginia Polytechnic Institute and State University, 2005.

[185] S. Yin, S. X. Ding, X. Xie, and H. Luo. A review on basic data-driven approaches for industrial process monitoring. *Industrial Electronics, IEEE Transactions on*, 61(11):6418–6428, 2014.

[186] Q. Zhang, M. Basseville, and A. Benveniste. Fault detection and isolation in non-linear dynamic systems: a combined input output and local approach. *Automatica, 34(11)*, pages 1359–1373, 1998.

[187] X. Zhang. Decentralized fault detection for a class of large-scale nonlinear uncertain systems. *American Control Conference (ACC), 2010. IEEE.*, 2010.

[188] X. Zhang and Q. Zhang. Distributed fault detection and isolation in a class of large-scale nonlinear uncertain systems. *In IFAC World Congress (2011, August), Milan, Italy (pp. 4302-4307)*, 2011.

[189] M. Zhong, S. X. Ding, T. Bingyong, T. Jeinsch, and M. Sader. An lmi approach to design robust fault detection observers. *In Intelligent Control and Automation Proceedings of the 4th World Congress on, vol. 4*, pages 2705–2709, 2002.

[190] M. Zhong, S. X. Ding, J. Lam, and H. Wang. An lmi approach to design robust fault detection filter for uncertain lti systems. *Automatica 39, no. 3 (2003): 543-550*.

8.3 Conference Papers

- **Hamed Khorasgani**, Chetan Kulkarni, Gautam Biswas, Jose R. Celaya, and Kai Goebel, Degredation modeling and remaining useful life prediction of electrolytic capacitors under thermal over-stress condition using particle filters In *Annual Conference of the Prognostics and Health Management Society, 2013.*

- **Hamed Khorasgani**, Daniel E. Jung, Gautam Biswas, Erik Frisk, and Mattias Krysander, Robust residual selection for fault detection In *53rd IEEE Conference on Decision and Control, pp. 5764-5769, 2014.*

- **Hamed Khorasgani**, Daniel E. Jung, Gautam Biswas, Erik Frisk, and Mattias Krysander, Off-line robust residual selection using sensitivity analysis In *25th International Workshop on Principles of Diagnosis, Graz, Austria, 2014.*

- **Hamed Khorasgani**, Daniel E. Jung, and Gautam Biswas, Structural approach for distributed fault detection and isolation In *IFAC-PapersOnLine 48, no. 21, 2015.*

- **Hamed Khorasgani**, Daniel E. Jung, and Gautam Biswas, Minimal structurally overdetermined sets selection for distributed fault detection In *26th International Workshop on Principles of Diagnosis, Paris, France, 2015.*

- Daniel E. Jung, **Hamed Khorasgani**, Gautam Biswas, Erik Frisk, and Mattias Krysander, Analysis of fault isolation assumptions when comparing model-based design approaches of diagnosis systems In *IFAC-PapersOnLine 48, no. 21, 2015.*

- Oliver Niggemann, Gautam Biswas, John S. Kinnebrew, **Hamed Khorasgani**, SÃűren Volgmann, and Andreas Bunte, Data-driven monitoring of cyber-physical systems leveraging on big data and the internet-of-things for diagnosis and control In *26th International Workshop on Principles of Diagnosis, Paris, France, 2015.*

- Gautam Biswas, **Hamed Khorasgani**, Gerald Stanje, Abhishek Dubey, Somnath Deb, and Sudipto Ghoshaln, An application of data driven anomaly identification to spacecraft telemetry data In *Prognostics and Health Management Conference, Denver, CO., 2016.*

- **Hamed Khorasgani**, and Gautam Biswas A combined model-based and data-driven approach for monitoring smart buildings (Accepted for presentation In *28th International Workshop on Principles of Diagnosis, Brescia, Italy, 2017.*)

8.4 Journal Papers

- Anibal Bregon, Gautam Biswas, Belarmino Pulido, Carlos Alonso-Gonzalez, and **Hamed Khorasgani**, A common framework for compilation techniques applied to diagnosis of linear dynamic systems In *IEEE Transactions on Systems, Man, and Cybernetics: Systems 44.7, 863-876, 2014.*

- Gautam Biswas, **Hamed Khorasgani**, Gerald Stanje, Abhishek Dubey, Somnath Deb, and Sudipto Ghoshal, An approach to mode and anomaly detection with spacecraft telemetry data In *International Journal of Prognostics and Health Management*, 2016.

- **Hamed Khorasgani**, Gautam Biswas, and Shankar Sankararaman, Methodologies for system-level remaining useful life prediction In *Reliability Engineering and System Safety 154 8-18, 2016.*

- **Hamed Khorasgani**, and Gautam Biswas, Structural Fault Detection and Isolation in Hybrid Systems In *IEEE Transactions on Automation Science and Engineering (T-ASE) 2017.*

8.5 Under review

- **Hamed Khorasgani**, and Gautam Biswas, A Methodology for Monitoring Smart Buildings with Incomplete Models. Submitted to *Applied Soft Computing*

8.6 To be submitted

- **Hamed Khorasgani**, and Gautam Biswas, Robust Fault Detection and Isolation in Nonlinear Dynamic Systems Using Sensitivity Analysis
- **Hamed Khorasgani**, and Gautam Biswas, Structural Methodologies for Distributed Fault Detection and Isolation
- **Hamed Khorasgani**, Ahmed Farahat, Gautam Biswas, Kosta Ristovski and Chetan Gupta, A Framework for Unifying Model-based and Data-driven Fault Diagnosis