DEVELOPMENT OF A SUPERVISORY TOOL FOR FAULT DETECTION AND DIAGNOSIS OF DC ELECTRIC POWER SYSTEMS WITH THE APPLICATION OF DEEP SPACE VEHICLES

by

MARC A. CARBONE

Submitted in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Department of Electrical Computer and Systems Engineering

CASE WESTERN RESERVE UNIVERSITY

January, 2021

,

CASE WESTERN RESERVE UNIVERSITY

SCHOOL OF GRADUATE STUDIES

We hereby approve the dissertation[1] of

MARC A. CARBONE

candidate for the degree of

Doctor of Philosophy

Committee Chair

Dr. Kenneth A. Loparo

Committee Member

Dr. Vira Chankong

Committee Member

Dr. Kalmesh Mathur

Committee Member

Dr. Farhad Kaffashi

Date of Defense

August, 17 2020

[1]We certify that written approval has been obtained

for any proprietary material contained therein.

,

In memory of Jeanette Kulifay

Contents

List of Tables

List of Figures

Acknowledgments

There are many people who contributed to the completion of this work. First and foremost, I would like to thank my advisor Dr. Kenneth Loparo for his endless encouragement throughout my time at CWRU. His guidance was instrumental in developing my knowledge and skills. For this I am extremely grateful.

Next I would like to thank James Soeder, Ray Beach, Linda Taylor, Karin Bozak, and Bob Scheidegger for giving me the opportunity to be a part of their team and supporting my research at NASA GRC. Jeff Csank also deserves special thanks for providing an unbelievable amount of support and advice over the years.

This dissertation could not be made possible without the patience and contributions of the APC development team. Jeff Follo, Brian Tomko, Matthew Granger, Matthew Muscatello, George Thomas, Kevin Rak, Joe Newman, and Steven Walker have given endless amounts of technical assistance, particularly with software design and implementation. Also thanks to Marcus Horning for capturing experimental data from the hardware testbed.

Amir Sajadi deserves special thanks for his countless reviews and invaluable advice. I would also like to thank Jordan Murray and Tim Dever from CWRU.

Lastly, I owe a huge amount of gratitude to my family, Francie, Marcel, Genna, Sarah, and Michael for the support they provided during my studies.

Development of a Supervisory Tool for Fault Detection and Diagnosis of DC Electric Power Systems with the Application of Deep Space Vehicles

Abstract

by

MARC CARBONE

This dissertation formulates the problem of fault detection and diagnosis of DC electric power systems for the application of autonomous spacecraft. The ability to accurately identify and isolate failures in the electrical power system is critical to ensure the reliability of a spacecraft. This problem becomes more pronounced during deep space missions that lack the ability to monitor from ground control. The current state of electrical power system fault supervision is insufficient to guarantee highly reliable and robust operation. To solve this issue, a combination of model-based and rules-based techniques are used in a hierarchical framework to improve the diagnostic performance of the spacecraft electrical power system. Noise, disturbances, and modeling errors are considered in the design of the method. Practical considerations related to the hardware and software are discussed for the flight application. A wide array of failure types are simulated in a series of experiments to assess the functionality of the design. The experiments showed that the methods used improved the diagnostic capability of the autonomous system while taking into account the limitations attributed to flight software requirements. The significance of this study is to provide a framework capable of advanced diagnostics of an electrical power system with little to no interaction from a human operator.

Chapter 1

Introduction

This chapter provides an overview of the challenges in autonomous fault detection and diagnosis in feedback control systems. In particular, fault detection and diagnosis of DC electric power systems with the application of autonomous spacecraft is discussed. Lastly, the motivations, objectives, and contributions of this dissertation are provided.

1.1 Fault Detection and Diagnosis of Feedback Control Systems

A feedback system (see Fig. 1.1) is defined as a plant in which the output is fed back into the input to form an error signal that controls the plant [1]. Although feedback control techniques enhance the system performance, it is still vulnerable to faults. A fault can be defined as an unpermitted deviation of at least one characteristic property or parameter

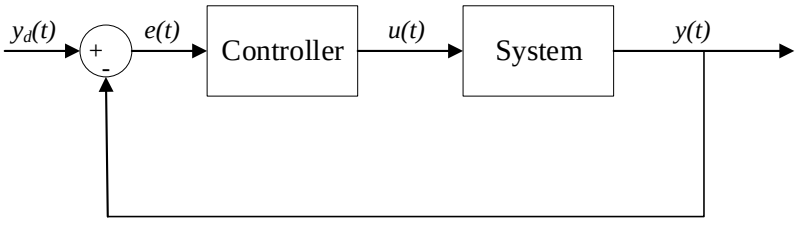


Figure 1.1: Feedback control system

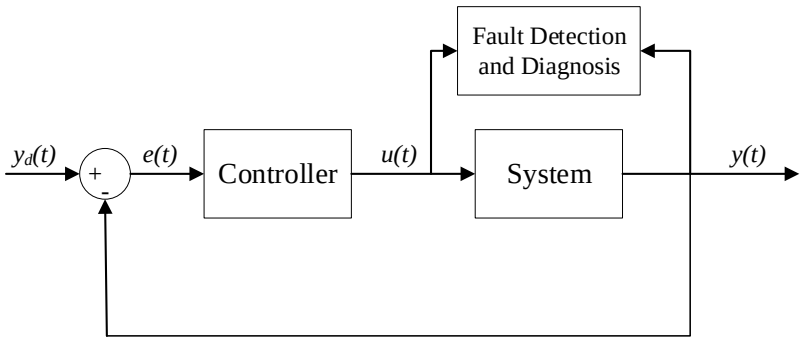


Figure 1.2: Fault detection and diagnosis of a feedback control system

of the system from the acceptable/usual/standard condition [2]. Faults in the sensors, actuators, or controller can be amplified by closed loop control, leading to damage of the plant and creating a dangerous environment for humans near it. Therefore, there is a demand for improved supervisory functions responsible for identifying faults and taking action to maintain operation and avoid further damage [3].

The process of identifying faults takes places in two steps. First, *fault detection* which is defined as the binary decision to recognize that a fault occurred. Second, *fault diagnosis* which is the task of finding the cause and location of the fault. The quick and accurate diagnosis of the faulty components facilitates decision making for the the optimal corrective actions. These decisions can prevent component deterioration and performance degradation. Fig. 1.2 depicts fault detection and diagnosis (FDD) integrated with a feedback control system.

For automatic processes, Isermann [4] defines the following functions needed for supervision:

(a) *monitoring*: measurable variables are checked in regard to tolerances, and alarms are generated for the operator.

(b) *automatic protection*: in the case of a dangerous process state, the monitoring function automatically initiates an appropriate counteraction.

(c) *supervision with fault diagnosis*: based on the measured variables, features are calculated, symptoms are generated via change detection, a fault diagnosis is performed and

decisions for counteractions are made.

Traditionally, methods (a) and (b) are sufficient for the overall protection of the process; however, to set the tolerances compromises must be made to obtain fast fault detection and diagnosis. These methods may also cause unnecessary alarms due to normal fluctuations in the variables. Limit checking is often used in fault detection due to its ease of implementation. This works especially well if the process operates in a steady state. For complex processes where the operating point changes rapidly, the task of fault detection and diagnosis becomes more complex. Additionally, closed loop control algorithms can mask changes in the process through feedback actions. Therefore changes cannot be detected from the output signals as long as the manipulated process inputs remain in the normal range. Thus, feedback systems hinder the early detection of process faults.

Although the classical limit-checking supervision methods are easy to implement and reliable, they alone are insufficient for safety critical processes. The main disadvantage is that they are only able to react after a relatively large change in a variable, such as a large abrupt fault or a long-lasting gradually increasing fault. In addition, detailed fault diagnosis using a set of rules is usually not possible because they are unable to capture the nuanced dynamics of the fault. Thus, the advanced methods of supervision and fault diagnosis in (c) are needed to perform the following objectives:

- Early detection of small faults with abrupt or incipient behavior
- Diagnosis of faults in the actuator, process components, or sensors
- Detection of faults in closed loops
- Supervision of processes in transient states
- Process condition based maintenance and repair
- Fault management and fault-tolerant control

Fig. 1.3 shows a detailed view of automatic process supervision. In particular, it shows how supervision with fault diagnosis can be integrated with the classical methods of monitoring and automatic protection. Supervision with fault diagnosis attempts to use all of the

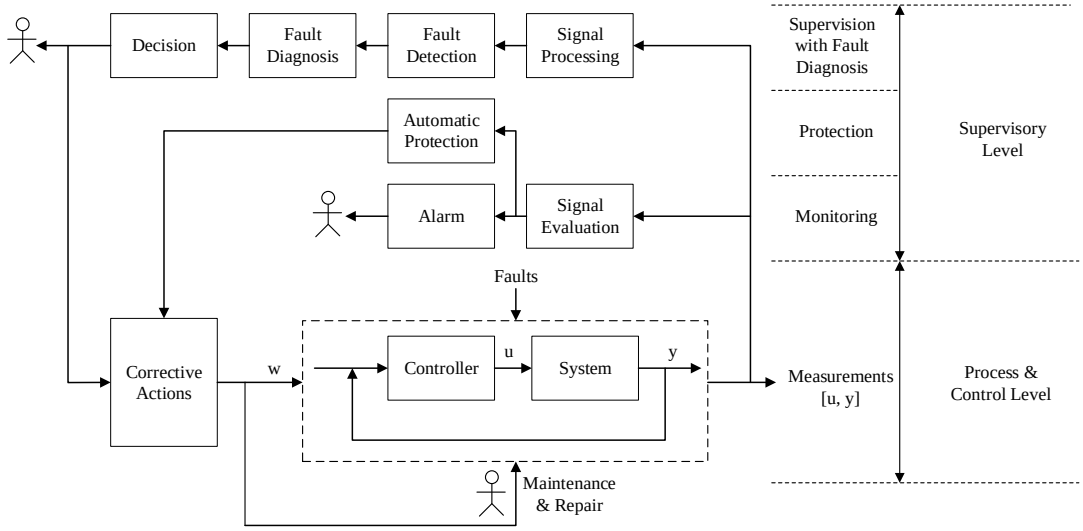


Figure 1.3: Flow diagram of process supervision methods in the loop, proposed by Isermann

available measurements and their mathematical relationships to generate more information about the process. Both output signals $y(t)$ and input signals $u(t)$ are valuable sources of information when analyzing the static and dynamic behavior of the plant with or without faults. Detailed fault diagnostic information can improve the decision making ability of the corrective actions function. This method is able to carry out recovery actions through a stop or change in operation, reconfiguration, or provide recommendations for maintenance and repair by the system operator. The advanced methods of process supervision are means to improve both the reliability and robustness of the feedback control system.

Due to the wide variety of potential fault types, more advanced techniques are necessary to correctly diagnose and react to faults in a timely manner. Therefore it is useful to identify the manner in which a fault may appear in the system. A fault may be classified into one of the following three categories:

(a) Abrupt (sudden) faults

(b) Incipient (slowly developing) faults, i.e. drift faults

(c) Intermittent faults

Abrupt faults appear when a large change in a parameter or behavior has occurred

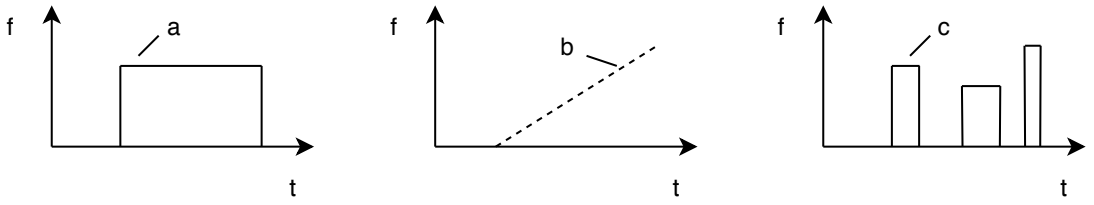


Figure 1.4: Fault types based on time dependency: a) abrupt, b) incipient, c) intermittent

rapidly. Often abrupt faults occur in systems where failures need to be detected early enough to prevent catastrophic damage or collapse of the entire system by fast corrective action. *Incipient* faults however, are commonly associated with maintenance problems where early detection of worn equipment is required. In this situation faults are typically small, and difficult to detect, but the detection time is of less importance. Lastly, *intermittent* faults appear and disappear over time, making detection and diagnosis additionally challenging. Fig. 1.4 shows the time dependency of a fault, f, for each type.

1.2 Fault Detection and Diagnosis in Electrical Power Systems

An application of a feedback control system with the need for fault detection and diagnosis is the electric power system. Major blackouts caused by faults have illustrated the need for improved fault supervision [5]. Failures in the sensors, actuators, components, and controller can cause the power system to deviate from its safe operating condition, leading to damaged equipment and losses in load. Advancing the state of fault supervision in power systems provides more reliability for critical loads.

The origins of modern power system control dates back to the mid-twentieth century. As power systems grew larger and more complex, the need for a total control system became necessary. The controller design in [6] proposes a system made up of automatic functions, and human functions. Real-time computer control for power systems is introduced in [7], where a multi-level approach is used to distribute processing power by assigning tasks to subprocesses, each with its own control system. Then the various subsystems are coordinated to obtain the best solution to the overall control problem. This approach may be referred to

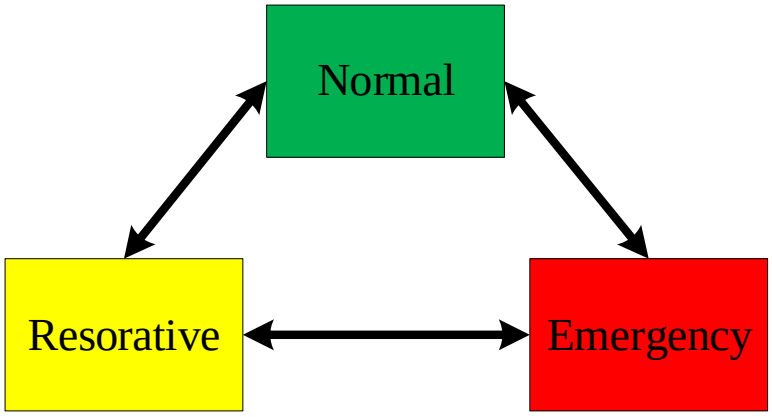


Figure 1.5: Power system operation 3 state diagram, proposed by Dy-Liacco

as *hierarchical control*. This techniques allows us to reduce the scope of problems that may be too difficult to solve, limiting the need for instrumentation and communication within the power system, and increase reliability by eliminating the possibility of a single point of failure.

The electrical operation of the generation and transmission system is classically decomposed into three operating states. These states are shown in Fig. 1.5. The normal operating state (also known as the preventative state) requires that all of the load demand is satisfied and all components are functioning withing their operating constraints. These constraints may include electrical, mechanical, and thermal limits. The control objective in the normal state is to continue indefinitely without interruption and at minimum cost. Also, it is implicit that the electrical system will not go unstable for minor or routine disturbances. Therefore the control system must be able to take defensive action during environmental changes, evaluate the effect of uncertainties, and also take action to prevent a reduction in service. Next the emergency state occurs when one or more component operating limits are violated. The control objective in this state is to relieve the system stress and prevent further degradation, while satisfying the greatest possible load demand. Lastly, the restorative state requires that all operating constraints are met, but some load demand is lost. This state usually takes place after an emergency. The control objective in the restorative mode is to transition from partial to total load satisfaction in minimum time.

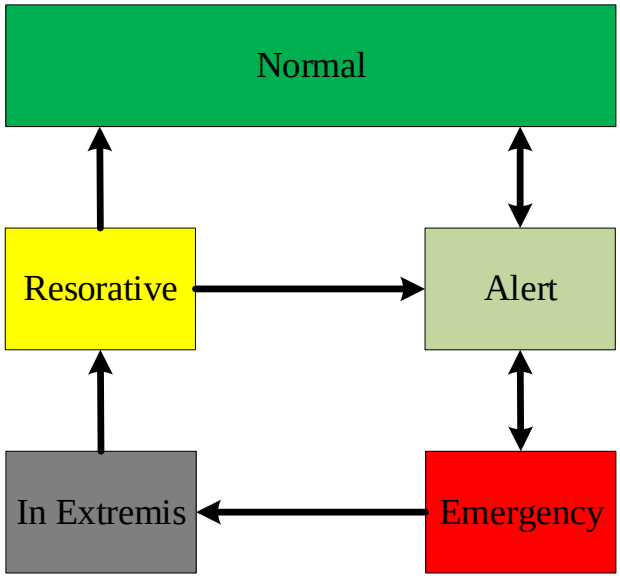


Figure 1.6: Power system operation 5 state diagram, proposed by Fink

Other developments in power system control have considered a five state paradigm [8], [9] as shown in Fig. 1.6. This structure introduces the concept of the alert state, where power generation and delivery meet the load demand but some of the components will violate their operational constraints if a credible contingency occurs. It also adds the in-extremis state, where power generation and delivery do not meet the demand and load shedding is required. This state can lead to a total blackout without the use of corrective actions.

Improved methods of fault detection and diagnosis can be used to enhance system reliability and performance. This can be done through the early detection of faults and the use corrective actions to maintain the power system in the normal state. For high-risk processes such as spacecraft, maintaining safe and reliable power system operation is of critical value.

1.3 Application to Deep Space Vehicles

The United States National Aeronautics and Space Administration (NASA) along with domestic and international partners are working to solve the challenges of deep space explo-

ration [10]. The increased distance between Earth-based mission control and the spacecraft will increase communication delays. For example, communications from Earth to a Mars-based spacecraft are expected to take between 6 to 44 minutes round trip [11]. Even crewed missions are unable to carry out all of the functions performed by mission control today due to the increasing complexity of the spacecraft. In order to solve these problems, higher levels of vehicle autonomy will be necessary to carry out the functions of deep space vehicles that have previously been done from the ground.

The electrical power system (EPS) of a spacecraft is one of the processes that will require improved automation in order to function in the deep space environment. The primary function of an electric power system is the reliable generation, transmission, and distribution of electric power to meet a randomly variable demand [12]. The same requirements are amplified for the electric power systems onboard spacecraft which may contain high cost components and potentially risk human lives. A critical aspect of spacecraft power system control is the ability to detect and diagnose faults quickly and accurately in order to prevent catastrophic blackouts on the spacecraft. Current efforts in fault detection and diagnosis for spacecraft EPS relies on monitoring, automatic protection and close monitoring from ground support to diagnose any anomaly in the system. Typically the limits of single variables are monitored and the operators are alarmed. In order to make EPS supervision a part of the onboard autonomous controller several enhancements will have to be made to the existing approach. The current methods of supervision can be improved significantly by taking into account the information hidden in all measurements and automatic actions to keep the system in operation [13].

1.4 Motivation of This Thesis

There have been several efforts in electrical power system fault detection and diagnosis for aerospace electric power systems by industry experts. An overview of this research can be concluded in [14], [15], [16], [17]. These works express the need for more advanced diagnostic and supervision capabilities in the electrical power system. Recent studies in FDD for aerospace vehicle electric power systems include the Bayesian approach [18], [19],

interacting multiple model technique [20], [21], optimization based methods [22], constraint suspension approach [23], Kalman filter based methods [24]. A majority of these studies consider a narrow range of faults, and only the case of single faults. Additionally, much of the research does not address the interactions between monitoring, automatic protection and supervision.

Substantial progress in field of FDD has been made for the application of terrestrial microgrids. The need for more reliable microgrids is illustrated in [25], [26]. Many works in terrestrial electric power system fault diagnosis have been conducted in both industry and academia. Recent progress in this field includes, inference engine/expert systems [27], [28], [29], Bayesian approaches [30], Kalman filter based methods [31], [32], and estimation based approaches [33]. Much of the existing research in terrestrial microgrids is focused on AC systems. In addition, studies that consider DC microgrids lack focus on system automation due to the availability of human operators. Furthermore, the Consortium for Electric Reliability Technology cites the integration of control actions with process supervision a one of the main challenges for the future electric power systems [34]. Lastly, a major issue in the field of fault detection and diagnosis is the gap in in development between industrial application and academic research. One reason for this is that advanced techniques often require more upfront effort to design and are more difficult to implement.

The motivation for this research is to implement a comprehensive online tool for monitoring, automatic protection, and supervision for fault detection and diagnosis for the application of electric power systems for spacecraft. This work will build off of the state of the art techniques in the literature and addresses the shortcomings of the current research. This dissertation will address the issues mentioned in this section and provide a design for a practical fault detection and diagnosis method that expands the current capability of spacecraft electric power system supervision. This tool will utilize state of the art techniques in order to expand the number of faults and the range of fault types that can be diagnosed. In addition, this work will advance the current state of autonomy for FDD in spacecraft by greatly reducing the need for human intervention within the power system. Much of the work in power system supervision is focused on the application of a single methodology or approach. This research attempts to provide an improved solution

using the strengths of multiple FDD methods in order to capture the best features found in industry and academia.

1.5 Objectives of This Thesis

This work is designed to address the technical challenges of electrical power system autonomous control for exploratory spacecraft.

The objective of this dissertation is to develop supervisory software that is capable of autonomously detecting and diagnosing faults within a DC power system onboard a spacecraft. The purpose of this software is to maintain the power system in the normal/preventative state as much as possible, then in the event of a fault, to safe the system and service as much load as possible until a repair can be made. In order to implement the appropriate corrective action, fault detection and diagnosis must be performed quickly and accurately.

The findings of this dissertation will offer recommendations to the challenges associated with designing a fault detection and diagnosis tool for spacecraft. It will provide insight into the nature of algorithms that should be used to detect and diagnose certain types of faults in a timely manner, and abide by the constraints of the hardware and communication equipment. The study will be conducted using models, simulation, and a hardware testbed. This research uses simulations of the electrical power system in the MatLab/Simulink environment, developed by PC Krause and Associates.

This work is part of the development effort of an Autonomous Power Controller (APC) under the Advanced Explorations Systems (AES) Modular Power Systems (AMPS) project at NASA GRC. AES is focused on developing vehicle level autonomy, where subsystems such as the electrical power system are controlled in a hierarchical manner and pass information to the higher level vehicle system manager (VSM). Several previous efforts in the development of the APC have been published. An autonomous communication interface for load planning was developed in [35]. Software agents were introduced for autonomous spacecraft control in [36]. The framework for autonomous control of a spacecraft is discussed in [37] and [38]. An overview of the current structure of the APC and VSM is presented in [39] and [10] respectively.

1.6 Contributions of this Thesis

This research consists of the development of an online supervision tool for electric power systems onboard autonomous spacecraft. The tool contains the following contributions to the field of fault detection and diagnosis:

1. The integration of heritage limit-checking and model-based techniques.
2. The ability to detect and diagnose a wide variety of electrical power system faults including sensors, actuators, communications and power electronics for the autonomous spacecraft application.
3. A method to detect and diagnose multiple faults in any sequence until the system is fully degraded or unable to be analyzed.
4. A technique for active probing of the power system to distinguish between similar fault types.

1.7 Organization of this Thesis

The remainder of this dissertation is organized as follows:

Chapter 2 presents metrics for the design of successful fault detection and diagnosis. Then it gives an overview of rule-based and model-based techniques and discusses the merits of both methods.

Chapter 3 discusses the development of the DC electric power system model and a description of the fault types that will be considered in this study. Next the hierarchical control paradigm is given, with a focus on the FDD methods used at each level. Lastly, the coordination of all three hierarchical levels is detailed.

Chapter 4 analyzes the performance and results of the fault detection and diagnosis tool using a series of functional tests.

Chapter 5 provides the conclusion of this dissertation and suggestions for future areas of research.

Chapter 2

Theoretical Background

This chapter introduces the metrics for successful fault detection and diagnosis. Then an overview of rules-based and model-based techniques are provided. Lastly, the strengths and weaknesses of each method are discussed.

2.1 Metrics for Successful Fault Detection and Diagnosis

It is useful to define a set of desirable properties for fault detection and diagnosis in order to determine which techniques are appropriate for a particular system. The metrics used in this work are adopted from [40]. In general, no one diagnostic algorithm will have all of the useful characteristics. Therefore it is important to understand these differences in order to select the appropriate FDD method(s) for a particular system. When an abnormality occurs, the fault diagnostic classifier should provide a set of hypothesis (faults) that explain the abnormality. To this end, two metrics can be used:

- *Completeness* can be defined as the requirement of a fault diagnostic classifier to provide the correct fault within the hypothesis set.
- *Resolution* is defined as the requirement for the hypothesis set to be as minimal as possible.

Based on these definitions, there is a clear trade-off between completeness and resolution. Adjusting the performance of one term or the other will have an effect on the overall accuracy

of the diagnostic classifier. Other quantifiable metrics issue trad-offs in the performance of fault detection and diagnosis. They are defined as follows.

(a) **Quick Detection and Diagnosis**

It is important for a diagnostic system to respond quickly to faults in order to prevent further damage. For highly severe faults the controller may need to respond within micro-seconds to successfully safe the system. Less critical faults may allow for longer duration reaction times. The challenge in detecting high speed faults is that the controller becomes more sensitive to noise, leading to more false alarms. Thus there is a trade-off between quick and robust diagnosis.

(b) **Isolability**

Isolability can be defined as the ability to distinguish between different faults in the system. In an ideal system, the classifier will only identify faults that are currently present. However,the availability of measured data, level of noise, modeling uncertainties, and disturbances all contribute to challenge of correctly diagnosing the true fault. Not to mention the design of the diagnostic system. A common controls problem in this field is being able to achieve high isolability while still rejecting modeling uncertainties and disturbances. It is up to the designer of diagnostic system to determine what level of each features need to be considered the most.

(c) **Robustness**

As mentioned before, the diagnostic system should be robust to noise, uncertainty, and disturbances in order minimize the number of false alarms during normal operation. Increasing robustness often comes at the cost of slower detection and diagnosis times, as well as less isolability. This results in the need for a balance of the three desirable qualities.

(d) **Classification Error Estimate**

In practical applications it is sometimes useful for the diagnostic system to provide some measure of confidence of a particular fault diagnosis. This feature may help the

controller in making restorative actions based on the level of confidence that the correct fault(s) have been identified.

(e) **Adaptability**

Plants that may change or evolve over time create new issues for fault detection and diagnosis. For the application of a space power system components such as distribution lines, solar arrays, and batteries will degrade over the many years of service. In addition, operational control methods may change over time due to faults and disturbances. It is important that the diagnostic system can adapt to all of these changes in order to continue monitoring for faults.

(f) **Model Complexity**

The level of detail of the models needed for fault diagnosis is worth noting. Developing practical models that accurately capture the behavior of the system is an important feature. For development, models should be made as simple as possible.

(g) **Computational Efficiency**

In order to create diagnostic systems that can run in real-time, the processor limits and storage requirements should be met based on the available computing power of the controller. Diagnostic systems should aim to reduce computational complexity and data storage.

(h) **Multiple Fault Diagnosis**

The ability to detect and diagnose multiple faults is an important but challenging task. The main difficulty is due to the interacting nature of many faults. Hence the single fault patterns used previously may not be useful in classifying instances of multiple faults. Moreover, developing scenarios for each case of multiple faults grows combinatorially. The number of fault combinations quickly grows, even for small systems.

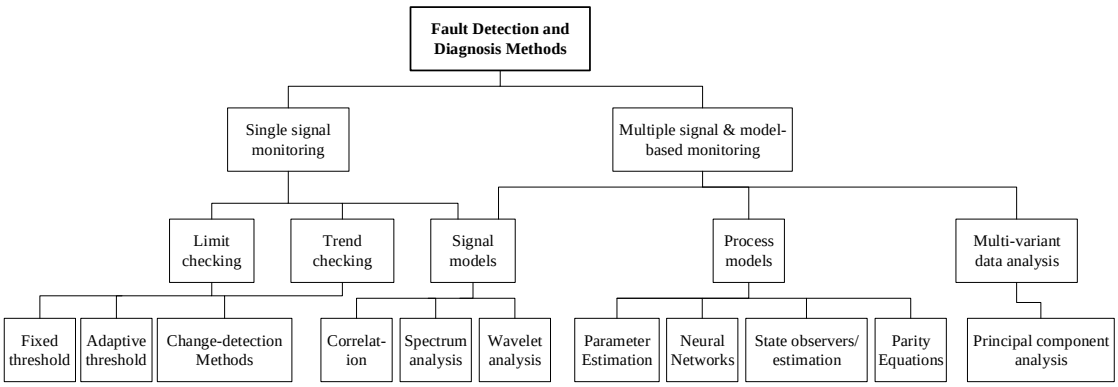


Figure 2.1: Classification of fault detection and diagnosis methods

2.2 Fault Detection and Diagnosis Methods Overview

The increase in process automation over the past several decades has led to several approaches to fault detection and diagnosis. Fig. 2.1 displays some common techniques for handling this problem. Each method can be grouped by those using single or multiple signals. Single signal monitoring consists of limit checking and trend checking, while multiple signal monitoring includes both process models and multi-variant data analysis. Signal models such as spectrum or wavelet analysis, however, can be applied to both single and multiple signal applications. The scope of all fault detection and diagnosis methods is too broad to cover within this work in detail. More information about each technique can be found in [13]. For the application of an autonomous spacecraft power system, this effort focuses on limit checking and process models to manage fault detection and diagnosis. The motivation for using these methods ranges from the capability of the flight computer, to the heritage of fault monitoring onboard spacecraft, to the modeling capability of a DC electric power system. An overview of the techniques used in this effort is discussed in detail in the following section.

2.3 Model-Free Methods

Model-free methods of fault detection are the most common techniques found in industry due to the simplicity and reliability of the techniques. Gertler [41] describes some of the

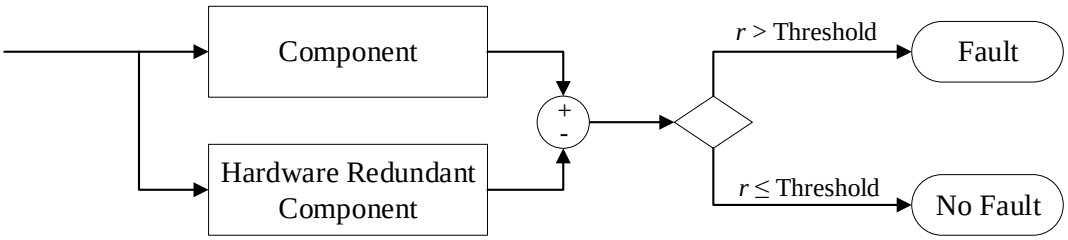


Figure 2.2: Block diagram of physical redundancy

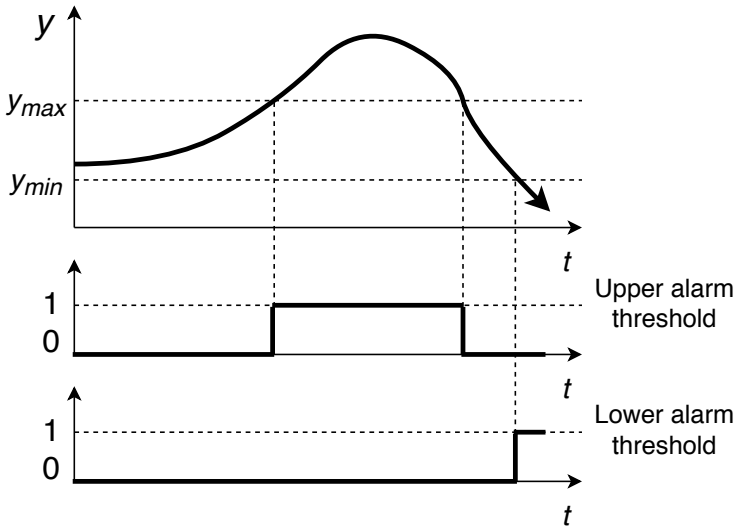


Figure 2.3: Limit checking of signal $y(t)$

common model-free methods such as *physical redundancy*. For physical redundancy multiple instruments (e.g. sensors) are installed to measure the same quantity. Any significant difference between the measurements can indicate a fault, as shown in Fig. 2.2. At least three redundant sensors are needed to create a voting scheme to isolate the faulty sensor. The additional cost and weight can be a large concern for aeronautics and space flight applications. Another method, *limit checking* is commonly found in practice, where measurements are compared by a computer to preset thresholds. For a measured sensor y we can detect faults using the absolute value check

$$y_{min} < y(t) < y_{max} \tag{2.1}$$

where y_{min} and y_{max} are predetermined fault thresholds. Many systems have two levels of

alarms, one to indicate a warning and one for emergency reaction. A limit check can also be applied to the trend $\dot{y}(t)$ of the signal $y(t)$. If the limit values are set small enough, the fault alarm can take place earlier than the absolute value check because the trend provides an indication of the signal progression. The normal state occurs when

$$\dot{y}_{min} < \dot{y}(t) < \dot{y}_{max}. \tag{2.2}$$

While limit checking can be very useful, the method suffers from two serious limitations. First, is that the plant variables may vary widely due to normal input variations, therefore the threshold needs to be set quite conservatively. Second, limit checking methods have very limited ability to specify faults, and the effect of a single process fault may propagate to several plant variables, setting off a confusing number of alarms, sometimes referred to as an "alarm-shower". The large number of alarms can make isolation extremely difficult and under certain circumstances impossible.

Expert Systems are an example of a model free method that uses a series of rules to determine the state of the system [42]. These rules are generated from the symptoms of the detection hardware or software (the alarm system). Normally, a series of **IF** and **THEN** statements are used to diagnose the symptoms and conclude the fault type. Expert systems have been implemented both autonomously and with a human operator in the loop [41]. The rules used in expert systems rely heavily on the operating staff's experience with the physical system. Often, this expertise is captured in rules of thumb which have been observed under emergency conditions. Humans must analyze a combination of data and alarms to diagnose the fault with the aid of the expert system.

The simplicity of design makes expert systems a popular choice for implementation in real-life applications. Generally, expert systems are highly specialized and capture a very narrow domain of problems. The main advantages to using expert systems are ease of development, transparent reasoning, (occasionally) ability to reason under uncertainty, and the ability to produce explanations for the solutions provided [28], [40].

Unfortunately fully autonomous expert systems are only able to diagnose faults that have been explicitly programmed into the algorithm. This drawback makes the design of

expert systems on complex plants bloated and inflexible. In addition, uncertainties such as noise, loss of information, and data synchronization error makes standard expert systems unreliable [43]. In addition, expert systems are only able to react to changes that result in a relatively large change in the features. Under this condition, either a large sudden fault has occurred or a slowly drifting fault has been in the system for a long time. Therefore there is a need to enhance autonomous fault supervision. These requirements must be met to improve performance: (i) early detection of small faults, (ii) diagnosis of faults in actuators and process sensors, (iii) detection of faults in closed loops.

An extension of expert systems, Fuzzy Logic, captures the knowledge of human experts into the expert system by using qualitative and natural language type expressions. In [43], fuzzy relations were used to build an expert system for finding fault in power systems. The fuzzy relations help mitigate the uncertainties mentioned earlier by replacing strict binary logic with qualitative relationships. Fuzzy logic can be a useful tool in "black box" systems where plant interactions are difficult to define, but the overall behavior follows some trend or pattern. For the application of DC power systems, the physics of the system are well known and therefore may not benefit from fuzzy logic techniques.

The disadvantage of these knowledge-based methods is that they only produce useful results for situations that have been explicitly prepared for during the design. This leaves them at risk of failing under unfamiliar alarm patterns or topological changes. In addition they can become unmanageable when faced with multiple failures. The amount of apriori cases in the knowledge-based system would grow exponentially large as the number of possible fault combinations increases.

2.4 Model-Based Methods

Increasing demands for reliability and safety have driven the need for improved supervision of complex processes such as power systems [44]. Process models, estimators, and decision methods make it possible to monitor non-measureable variables such as process states and parameters, as well as predicting signals. These techniques can be used to detect faults earlier and locate them more accurately than conventional limit and trend checks.

Early approaches for fault detection and diagnosis using mathematical models were developed in the mid-twentieth century, see [45], [46], [47], [48], [49], [3] for a review. Among these methods are the fault detection filter [50], the innovation test using a single Kalman filter [51, 52], banks of Kalman filters (or Luenberger observers) [53], [54], the parity space method [55], the parameter estimation approach [56]. A more recent review of fault detection isolation and reconfiguration methods can be found in [2]. Next, a general structure of model based methods is provided and the challenges associated with them are discussed.

In general, model-based methods take place in three steps. The first, is to generate a set of variables or residuals, which are typically zero-mean under the non-faulted mode. It is best if the residuals are insensitive to noise, disturbances, and model uncertainties. The second step is to make the decision of whether a fault has occurred (detection) and the location and type of fault (diagnosis) based on the residual signature. The final step is for the controller to perform an online reconfiguration in response to any detected fault(s).

The diagnosis (sometimes referred to as isolation) stage of FDD commonly uses the concept of redundancy. Redundancy can exist in the form of a physical component or a conceptual representation of a part of the system, also known as analytical redundancy. The latter approach tends to be more cost and weight effective, which is largely beneficial in aerospace applications. On the other hand, analytical redundancy methods tend to be more difficult to guarantee robustness in the face of noise, uncertainties, and disturbances.

Most model based fault detection and diagnosis methods rely on analytical redundancy to determine the probability of faults [57], [58]. Analytical redundancy is achieved through use of an explicit mathematical model as well as sensor measurements from the plant. The appeal of analytical redundancy lies in the fact that additional information can be extracted from the existing information without adding physical equipment to the plant. However the challenge in this approach is the need for an accurate mathematical model. The model can be generated from empirical plant data, through system identification, or based on the physical principals of the plant. The success of FDD largely depends on the design of the model. Models may be linear, non-linear, dynamic, static, continuous, or discrete and so on. Frequently, linear discrete-time dynamic models are used [59]. The basic concept of

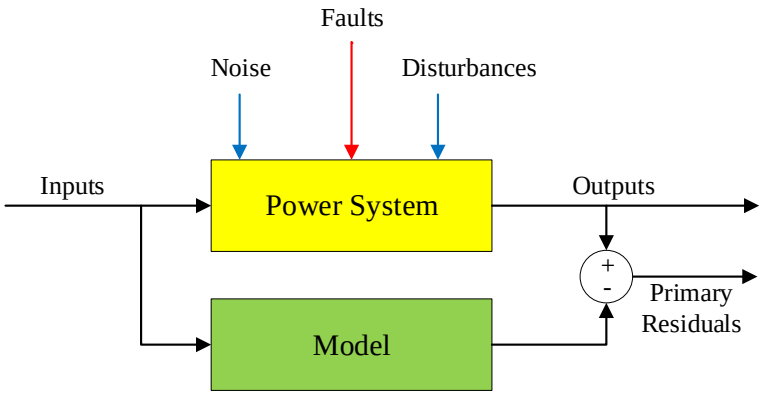


Figure 2.4: Analytical redundancy

model-based FDD is to compare observations from plant to predicted values generated from the mathematical model. A difference between the model estimates and the observed values are indicative of a fault in the system. The difference between measured outputs, y_t and the estimated model outputs, $\hat{y}_t$ define the error of in the system otherwise known as residuals [45].

$$e_t = y_t - \hat{y}_t \tag{2.3}$$

Typically, residuals are zero under normal conditions, and deviate from zero in the presence of a fault. Unwanted noise in measurements and system disturbances can also cause the residuals to become nonzero. These factors must be accounted for in the design of the FDD method. The way a fault propagates into the system equations may be additive or multiplicative. Additive faults may be in the form of plant leaks and sensor biases, whereas multiplicative faults can take be changes in plant parameters such as degradation.

Many model based FDD methods rely on linear discrete-time models. Due to the non-linear nature of many physical plants, models often linearize around a specific operating point. In addition, continuous time plants are commonly represented by their discretized model. A (discrete-time t) state-space model relates the state vector x_t to the input vector

y_t using the known system matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$. The state-space equations are

$$x_{t+1} = \mathbf{A}x_t + \mathbf{B}u_t \tag{2.4}$$

$$y_t = \mathbf{C}x_t \tag{2.5}$$

Introducing additive process faults and additive process noise can be done using vectors $\mathbf{v}_t$ and $\mathbf{w}_t$ each with coefficient matrices $\mathbf{P}$ and $\mathbf{Q}$ respectively. Now the state equations become

$$x_{t+1} = \mathbf{A}x_t + \mathbf{B}u_t + \mathbf{P}v_t + \mathbf{Q}w_t \tag{2.6}$$

Next, let Δu_t and Δy_t be the additive measurement fault (bias or offset) on the input u_t and output y_t. Similarly, for measurement noise add δu_t and δy_t so the measured input $\tilde{u}_t$ and output $\tilde{y}_t$ become

$$\begin{aligned} \tilde{\mathbf{u}}_t &= \mathbf{u}_t + \Delta\mathbf{u}_t + \delta\mathbf{u}_t \\ \tilde{\mathbf{y}}_t &= \mathbf{y}_t + \Delta\mathbf{y}_t + \delta\mathbf{y}_t. \end{aligned} \tag{2.7}$$

The vector $\mathbf{u}_t$ can also represent unmeasured control signals with malfunctions and noise modeled the in a similar manner.

Lastly, matrices $\Delta\mathbf{A}_t$, $\Delta\mathbf{B}_t$, and $\Delta\mathbf{C}_t$ are used to model discrepancies between the true system model $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ and the mathematical model $\hat{\mathbf{A}}$, $\hat{\mathbf{B}}$, and $\hat{\mathbf{C}}$. Model discrepancies may represent multiplicative faults in the system such as a change in network parameter.

$$\begin{aligned} \hat{\mathbf{A}} &= \mathbf{A} + \Delta\mathbf{A}_t \\ \hat{\mathbf{B}} &= \mathbf{B} + \Delta\mathbf{B}_t \\ \hat{\mathbf{C}} &= \mathbf{C} + \Delta\mathbf{C}_t \end{aligned} \tag{2.8}$$

Building the complete mathematical description of the system with all of the possible

faults and noises accounted for can be done by rearranging equations 2.4 - 2.8.

$$x_{t+1} = \hat{\mathbf{A}}x_t + \hat{\mathbf{B}}\tilde{u}_t + \mathbf{P}v_t + \mathbf{Q}w_t \tag{2.9}$$

$$\tilde{y}_t = \hat{\mathbf{C}}x_t. \tag{2.10}$$

Once the mathematical model is defined, the measurements $\tilde{y}_t$ are compared to the model outputs $\hat{y}$. Generating a useful set of residuals is critical to the success of fault diagnosis. A mathematical model of the plant is used to generate the set of residuals in on of three common forms. *Structured residuals* are used when a residual is only sensitive to a particular fault or set of faults resulting in a combination of residuals to indicate a fault (fault code). *Directional residuals* use the direction of the residual vector to indicate each individual fault. Lastly, *diagonal residuals* have each residual signal a particular fault. Several residual generation methods have been designed and have their own strengths and weaknesses. Linear dynamic systems have three common techniques for generating sets of residuals, *parity (direct consistency) relations*, *parity space*, *dynamic observers*. In [60], Gertler proves that these algorithms produce identical results.

In order to make a decision of whether or not a a fault has been detected and/or diagnosed in the system, the residuals must be evaluated. One example of this is robust residual evaluation. This technique uses hypothesis testing to evaluate the residuals with the assumption that the residuals are random variables. This can be as simple as a residual exceeding a threshold limit, but more complicated methods include statistical theories such as the generalized likelihood ratio (GLR) test or the sequential probability ratio test (SPRT).

A serious challenge in developing an online model for fault detection and diagnosis is the sensitivity to noise, modeling errors and disturbances. These three phenomena can obscure the effect of faults and trigger false alarms. Hence, the ability to decouple the model from these three factors is critical in the design of the FDD approach. Some of the major modeling considerations are described below.

(a) **Noise** Both measurement and process noise are present in almost any physical system. A low signal to noise ratio can make it difficult and sometimes impossible to detect and diagnose faults. In addition, noisy signals can often trigger false alarms. Filtering

techniques can be used to reduce the effect of noise. For example, moving averages of the residuals and low-pass filtering can be used to reduce the impacts that noise has on the system. Also, statistical tests have been proven useful in determining that the noise distributions measured by the plant match the theoretical assumptions in the design of the FDD algorithm.

(b) **Disturbances** Disturbances can be described as any change in the system that is not considered a fault. For example, loads changing in power distribution system may be considered as disturbances. Often we can look at disturbances as unmeasurable inputs to the system. With sufficient information about the disturbance (i.e transfer function, state space model, etc.) the residuals can sometimes be decoupled from the disturbances making the FDD scheme insensitive to the disturbances. This will improve the ability to correctly diagnose faults and avoid false alarms. However, the number of disturbances and the outputs of the model may not allow for decoupling of the residuals to the disturbances, in which case, approximate decoupling can be used.

(c) **Modeling Errors** Another challenge in modeling physical systems is that the mathematical models almost never perfectly match the real-world behavior of the system. These differences make fault detection and diagnosis even more challenging. Linearization and other simplifying assumptions can make the method less complex, however, they can cause even greater modeling discrepancies. Reducing the sensitivity of modeling errors often results in also reducing the sensitivity to faults. Certain optimization techniques, such as H-infinity can be used to help mitigate modeling errors.

2.4.1 Dynamic State Estimation (Kalman filter)

Power systems are operated in a hierarchical manner [61]. Faults in the power system are also detected and diagnosed at different levels of the hierarchy. Circuit breakers and relays are used to detect and interrupt dangerous fault currents, while the central controller uses system-wide information to diagnose more subtle failures in the transmission and distribution systems. To compliment this controls paradigm, many of the monitoring and controls applications at the central control level rely on a steady-state model of the system. Ter-

restrial power grids have used models to aid the system operators since the mid-twentieth century in the form of static state estimation (SSE) [62]. Development of the SSE model can be found in [63], [64], and [65].

In reality, power systems never truly operate in the steady-state. Stochastic variations in generation and complex loads cause large uncertainties in the system dynamic characteristics. For the application of spacecraft, generation such as PV arrays and battery storage and a variety of loads (resistive, constant power, AC, etc.) for life support and scientific instruments increase uncertainty and complexity of the electrical behavior. Therefore, the traditional techniques of SSE are unable to capture the dynamic behavior of the operational environment. New tools, such as dynamic state estimation (DSE), are capable of accurately modeling the dynamics of the system states [66]. DSE can be used to help the energy management system in many areas of protection and control [67] [68]. DSE has been used to monitor oscillations and tune power system stabilizers [69], enhance hierarchical and decentralized control [70], [71], improve dependability and reliability of protection systems [72] [73], and enhance the reliability of system models used for dynamics security assessment (DSA) [74].

To estimate the dynamic state vector of the system, a Kalman filter framework can be used. Depending on how the states are propagated over time variants of the Kalman filter are available, such as the extended Kalman filter (EKF), unscented Kalman filter (UKF), ensemble Kalman filter (EnKF), and particle filter (PF). To build an understanding of these methods, a brief review of the Kalman filter is provided below.

A linear multi-input muti-output (MIMO) process with discrete time signals can be modeled in the state space form [13]

$$x_{k+1} = \mathbf{\Phi} x_k + \mathbf{G} u_k \tag{2.11}$$

$$z_k = \mathbf{H} x_k. \tag{2.12}$$

where x_i is an $n \times 1$ vector of state variables (subscript i indicates time instant), u_i is a $p \times 1$ vector of control variables, w_i is a $q \times 1$ vector of random forcing functions, z_i is an

$m \times 1$ vector of output variables (or observables), $\mathbf{\Phi}$ is an $n \times n$ state transition matrix, $\mathbf{G}$ is an $n \times p$ input distribution matrix, and $\mathbf{H}$ is an $m \times n$ output matrix.

Next, a $q \times 1$ vector of random forcing functions w_k and an $m \times 1$ vector of random measurement errors v_k can be added to the model.

$$x_{k+1} = \mathbf{\Phi} x_k + \mathbf{G} u_k + \mathbf{\Gamma} w_k \tag{2.13}$$

$$z_k = \mathbf{H} x_k + v_k \tag{2.14}$$

where $\mathbf{\Gamma}$ is an $n \times q$ noise distribution matrix.

The random vectors w_k and v_k are assumed to be Gaussian and white. Their mean and covariance are

$$E\{w_k\} = \bar{w}_k, \qquad E\{(w_k - \bar{w}_k)(w_j - \bar{w}_j)^T\} = \mathbf{Q}_k \delta_{k,j} \tag{2.15}$$

$$E\{v_k\} = \bar{v}_k, \qquad E\{(v_k - \bar{v}_k)(v_j - \bar{v}_j)^T\} = \mathbf{R}_k \delta_{k,j} \tag{2.16}$$

$$E\{(v_k - \bar{v}_k)(w_j - \bar{w}_j)^T\} = 0 \tag{2.17}$$

where $E\{\cdot\}$ is the expectation operator and $\delta_{k,j}$ denotes the Kronecker delta

$$\delta_{k,j} = \begin{cases} 1, & k = j \\ 0, & k \neq j. \end{cases} \tag{2.18}$$

The initial conditions of the state variables x_0 are also assumed to be Gaussian random variables with mean and covariance

$$E\{x_0\} = \bar{x}_0, \qquad E\{(x_0 - \bar{x}_0)(x_0 - \bar{x}_0)^T\} = \mathbf{P}_0$$

$$E\{(x_0 - \bar{x}_0)(w_k - \bar{w}_k)^T\} = 0$$

$$E\{(x_0 - \bar{x}_0)(v_j - \bar{v}_j)^T\} = 0$$

$$\text{for } i = 0, 1, 2, ...$$

The Kalman filter determines the best estimate for the state vector x_k based on the

input variables u_k and the measured output variables y_k. For a least squares estimation the following must be true:

$$\min ||x_k - \hat{x}_{k|j}||^2 \tag{2.19}$$

where k is the present time and j is the time instant of the measurements. From [75] the estimator can be given different names depending on the time parameters:

$$\begin{aligned} k &> j, \text{ prediction problem} \\ k &= j, \text{ filtering problem} \\ k &< j, \text{ smoothing problem.} \end{aligned}$$

For fault detection and diagnosis the filtering and one-step ahead prediction functions are useful. The following definitions are useful for Kalman filter notation:

Output measurements used:

$$\mathbf{z}_j = \{z_0, z_1, ..., z_j\}. \tag{2.20}$$

Optimal estimates (unbiased minimum variance estimate of x_k based on observations up to time j):

$$\hat{x}_{k|j} = E\{x_k|\mathbf{z}_j\} \tag{2.21}$$

Estimation error:

$$\tilde{x}_{k|j} = x_k - \hat{x}_{k|j} \tag{2.22}$$

Innovation sequence:

$$\nu_k = z_k - \hat{z}_{k|k-1} \tag{2.23}$$

where $\hat{z}_{k|k-1}$ is the unbiased minimum-variance estimate of z_k on observations up to $k-1$ i.e. from the set $\{y_0, y_1, ..., y_{k-1}\}$.

Error covariance of $\hat{x}_{k|j}$:

$$\mathbf{P}_{k|j} = E\{(x_k - \hat{x}_{k|j})(x_k - \hat{x}_{x|j})^T\} \tag{2.24}$$

Finally, if it is assumed that all system parameters and statistics are known exactly, the innovation sequence of the Kalman filter can be generated in the following form [76]

$$\bar{x}_{k+1|k} = \mathbf{\Phi}[\bar{x}_{k|k-1} + \mathbf{K}_k\nu_k] + \mathbf{G}_k u_k + \mathbf{\Gamma}_k\bar{w}_k \tag{2.25}$$

$$\hat{x}_{0|-1} = x_0 \tag{2.26}$$

$$\nu_k = z_k - \mathbf{H}_k\hat{x}_{k|k-1} - \bar{v}_k \tag{2.27}$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_{k|-1}\mathbf{H}_k^T + \mathbf{R}_k)^{-1} \tag{2.28}$$

$$\mathbf{P}_{k+1|k} = \mathbf{\Phi}_k\mathbf{P}_k\mathbf{\Phi}_k^T + \mathbf{\Gamma}_k\mathbf{Q}_k\mathbf{\Gamma}_k^T \tag{2.29}$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1} \tag{2.30}$$

where $\mathbf{K}_k$ is the $n \times m$ Kalman gain matrix.

One of the reasons the Kalman filter is a useful tool for fault detection and diagnosis is that the statistics of the output are well known. The innovation sequence ν_k (which can be used as the residuals), is zero mean Gaussian white noise sequence with covariance equal to

$$\text{cov}(\nu_k) = (\mathbf{H}_k\mathbf{P}_{k|-1}\mathbf{H}_k^T + \mathbf{R}_k). \tag{2.31}$$

For fault isolation, it is more useful to consider the Standardized Innovation Sequence

$$\eta_k = (\mathbf{H}_k\mathbf{P}_{k|-1}\mathbf{H}_k^T + \mathbf{R}_k)^{-1/2}\nu_k \tag{2.32}$$

where $(\cdot)^{-1/2}$ represents the square root of the inverse of a matrix. Then it is clear that

$$E\{\eta_k\eta_j^T\} = \mathbf{I}\delta_{k,j} \tag{2.33}$$

where I denotes the identity matrix.

To illustrate this behavior, consider the 2-bus power system in Fig. 2.5. The bus on the left is regulated at 120V and 3A flows across the distribution line with resistance $R = 100\Omega$. With measurements of each bus voltage and current at each end of the line, a model for this system (which will be detailed in the next chapter) is constructed. Sampling 1,000 data

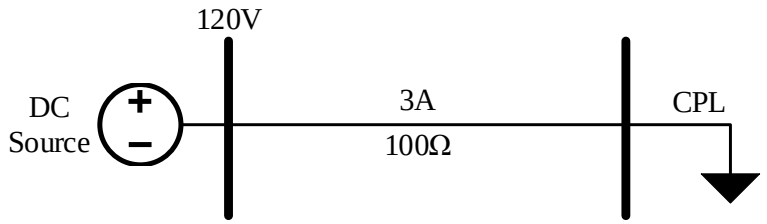


Figure 2.5: Two bus DC electric power system one-line diagram

point results in the black signals shown in Fig. 2.6. Passing this data through the Kalman filter produces estimates, y, for each measurement shown in red. Notice the reduction in noise provided by the Kalman filter. Next, the innovation sequence, ν is computed by Equation (2.27) and displayed in Fig. 2.7. This is a zero mean signal with non-unit variance. Lastly, the standardized innovation sequence, η is calculated by Equation (2.32) and shown in Fig. 2.8. The statistics of each sequence are captured in Table 2.1. Note that both the innovation and standard innovation sequences have mean approximately equal to zero, while only the standard innovation sequence has unit variance. This feature assists in verifying the normal behavior of the plant as well as diagnosing faults in the measurement equipment.

Table 2.1: Statistics of the Innovations

Measurement	**Mean ν**	**Variance ν**	**Mean η**	**Variance η**
V Bus 1	0.0333	0.0412	0.1663	1.0272
I Bus 1	-0.0029	0.0398	-0.0150	0.9904
V Bus 2	-0.0321	0.0396	-0.1606	0.9867
I Bus 2	0.0055	0.0429	0.0279	1.0689

2.4.2 Model Based Hypothesis Testing

The model-based framework is capable of detecting and diagnosing several types of faults in the system, including, biased sensors, noisy instruments, changes in levels of noise, changes in system parameters, changes in the system structure, and more through hypothesis testing. The convenience of the Kalman filter is that all of these faults cause the standardized

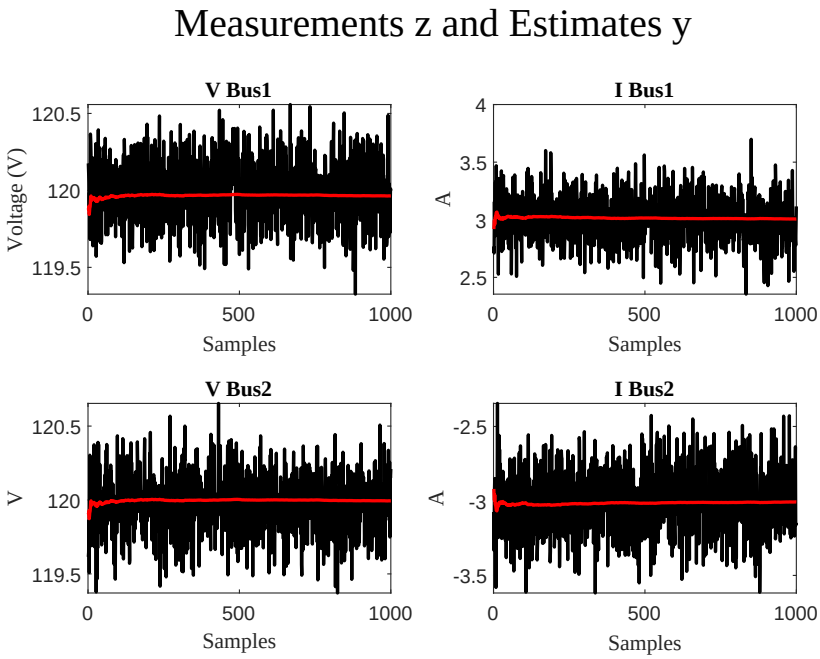


Figure 2.6: Measurements and Estimates of the 2-Bus System

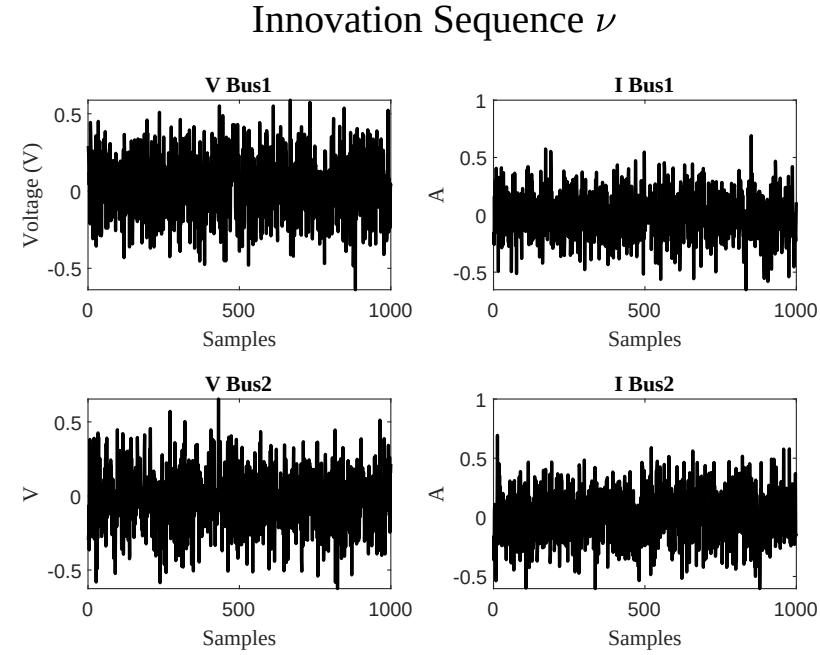


Figure 2.7: Innovation Sequence ν, of the 2-Bus System

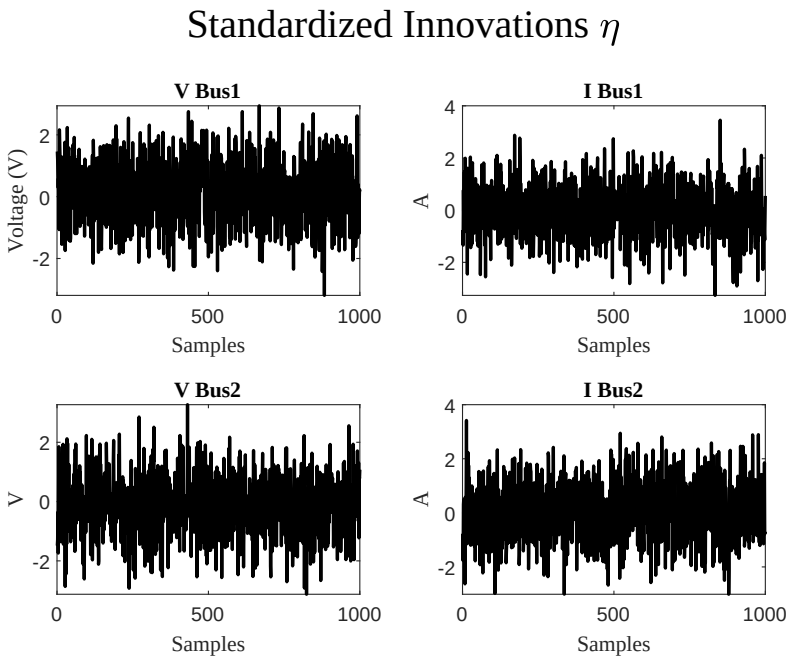


Figure 2.8: Standardized Innovation Sequence η of the 2-Bus System

innovation η_k to deviate from their zero mean, unit variance, and whiteness properties. The following three statistical tests can be conducted to check each characteristic of the standardized innovation.

(a) **Test of whiteness**

The first, and most important test on the innovation sequence, the test of whiteness, confirms independence of the signals at different time instants. This test can identify changes in instrument noise and communication failures. The test of whiteness is critical because the other tests make require the input signal to be white. Most tests for whiteness are based on the autocorrelation function c_k of a stationary process for lag $k = 1, 2, ...$ [77] as follows:

$$c_k = E(\eta_i - \bar{\eta})(\eta_{i-k} - \bar{\eta})^T \tag{2.34}$$

where $\bar{\eta}$ denotes the mean of η_i.

For practical purposes, c_k is often estimated as [51],

$$c_k = \frac{1}{N}\sum_{i=k}^{N}(\eta_i - \hat{\bar{\eta}})(\eta_{i-k} - \hat{\bar{\eta}})^T \tag{2.35}$$

where $\hat{\bar{\eta}}$ represents the sample mean

$$\hat{\bar{\eta}} = \frac{1}{N}\sum_{i=1}^{N}\eta_i. \tag{2.36}$$

It can be shown that $\hat{c}$ is an asymptotically unbiased and consistent estimate of c_k [78]. Furthermore, under the null hypothesis, $\hat{c_k}$ is asymptotically independent and normal with zero mean and covariance equal to I/N. Therefore the autocorrelation function must lie within the band $\pm 1.96/\sqrt{N}$ more than 95 percent of the sample periods for the null hypothesis. Randomness of the measured signals ensures that a univariate statistical process is acting on the system, a key assumption for the Kalman filter. When detecting for non-randomness, the autocorrelation function is typically only checked for the first lag ($k = 1$). In the event that randomness is not valid, a time series or non-linear model should be used.

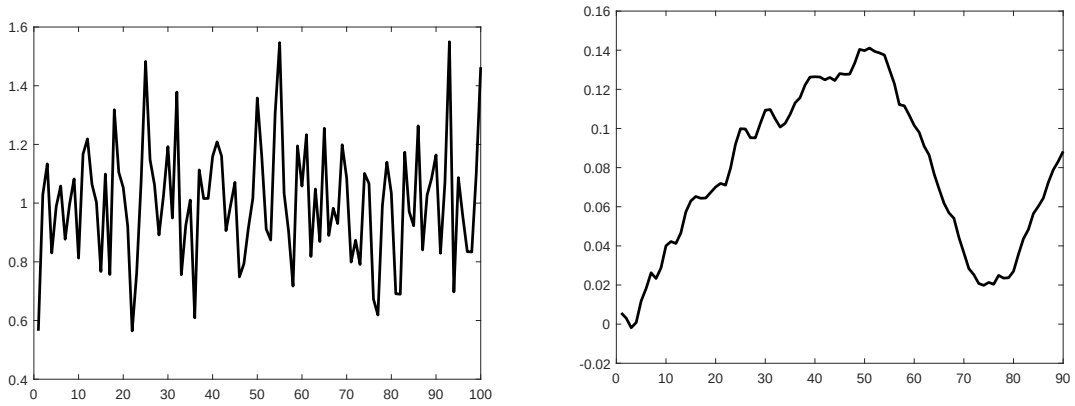
Fig. 2.9 shows a plot of the autocorrelation function for a Gaussian random signal for several time lags. The random signal has mean = 1 and variance = 0.2. Since the signal is random, the autocorrelation is near zero for all time lag separations and fits within the band $\pm 1.96/\sqrt{N}$ for more than 95 percent of the sample periods.

(b) **Tests of mean**

The tests of mean simply determine whether the innovation sequence is zero mean or not. The mean of the innovation sequence is estimated by

$$\hat{\bar{\eta}} = \frac{1}{N}\sum_{i=1}^{N}\eta_i. \tag{2.37}$$

where N is the sample size and $\bar{\eta}$ is the true mean. The null hypothesis states that $\hat{\bar{\eta}}$



(a) Gaussian Random Signal

(b) Autocorrelation Function of the Random Signal

Figure 2.9: Autocorrelation Function Example

has a Gaussian distribution with zero mean and covariance

$$E\{\hat{\bar{\eta}}\hat{\bar{\eta}}^T\} = I/N. \tag{2.38}$$

Thus, at a 5 percent significance level, the null hypothesis is rejected when

$$|\hat{\bar{\eta}}| \geq 1.96I/\sqrt{N}. \tag{2.39}$$

This test is limited by the fact that the covariance of η_i must be known. An alternative test which uses the T^2-statistic and sample covariance $\hat{c}_o$ can be found in [79].

(c) **Tests of covariance**

Under normal conditions the measurements of many processes, including DC electrical power systems, are uncorrelated. Therefore, a test is performed to check if the measurements are independent. Processes such as flight navigation systems may suffer from correlation within the measurements due to vibrations across the aircraft. This can lead to errors in fault detection and diagnosis by masking the true system data. Therefore a test of covariance on the residuals is useful. The covariance of the innovation sequence is estimated by

$$\hat{c}_0 = \frac{1}{N}\sum_{i=1}^{N}(\eta_i - \hat{\bar{\eta}})(\eta_i - \hat{\bar{\eta}})^T. \tag{2.40}$$

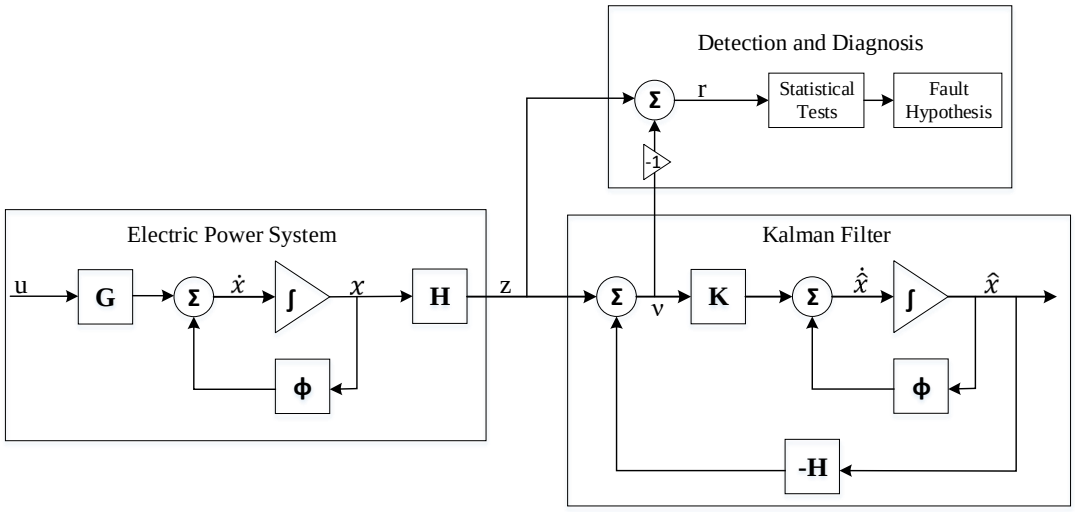


Figure 2.10: Fault detection and diagnosis using the Kalman filter

Under the null hypothesis, $\hat{c}_o$ has a Wishart Distribution and the trace of $\hat{c}_o$ has a Chi-Squared distribution with $(N-1)r$ degrees of freedom, where r is the number of observed measurements [79]. Therefore, $\hat{c}_o$ can be tested for its null hypothesis covariance equal to the identity matrix, I [51].

The estimates for the statistics of the innovation sequence described above are used for hypothesis testing. These tests are beneficial since they capture more detailed information for fault diagnosis, particularly in the measurement equipment. For example, the test of mean may fail on a single residual, thus indicating a bias on an instrument. These tests can often simplify the diagnostic procedure [80]. Fig. 2.10 shows an overview of the detection and diagnosis procedure using the Kalman filter.

2.5 Method Overview

Each approach to fault detection and diagnosis has various strengths and weaknesses. Depending on the number of known inputs and outputs, as well as environmental interactions, FDD can be a well-defined problem, regardless of the number of faults in the system. Sensor density adds to the effectiveness of any diagnostic procedure. A single sensor cannot be used to diagnose faults, because there is not enough information to distinguish between faults,

or even to validate the performance of the sensor itself. In most practical situations, sensor information lands somewhere in between a single sensed value, and complete knowledge of the system. Therefore it is up to the diagnostic method to account for how great or little information is available.

With enough information, qualitative model-based methods attempt to specify, if a fault can be detected, if one fault is distinguishable from other faults that may or may not have been considered in the original design, and if we can detect faults in the presence of measurement and process noise. Under these conditions, model-based methods can be made to reduce the effects of unknown disturbances in the system. The main cost in developing strong quantitative model-based methods is in the effort to construct the model. Depending on the placement of sensors, the number of unknown disturbances, and potential non-linear behaviors of the system, can make developing an accurate model of the system challenging. In practice, parameter drifts can cause severe modeling errors, which is another limitation of model-based methods.

Rules-based expert systems can be used in cases where fundamental principals are lacking. In particular, if there is an abundance of experience of a devices normal and faulty behavior, but not enough detail to develop accurate mathematical models, rules can be generated to classify faults. These methods are limited by the ability to diagnose faults not included in the original design, robustness to noise and disturbances, and the processing ability to distinguish between a large number of alarms. Causal-based rules can provide knowledge of the path of propagation of a fault. This can be very useful to a human operator who needs to make decisions about the actual faults present.

Pattern recognition approaches, as mentioned before, are based largely on historical process data. Neural networks have been shown to perform well in the face of noise and ability to accurately diagnose faults among several fault types. These methods are limited in the situations where the diagnostic tool has not been trained for a fault present in the system. Certain neural networking methods attempt to mitigate this problem by avoiding decisions in cases where there are no similar training patterns compared to the observed signal. In addition, these methods have difficulty in the diagnosis of multiple faults. This draws an important distinction between mode-based and historical data driven approaches.

Model-based observer methods can be explicitly designed to handle multiple fault identifications. The combinatorial complexity is mostly unavoidable, but can be reduced using efficient search techniques. In contrast, neural networks do not explicitly manage the idea of multiple fault diagnosis, Specifying each multiple fault class for neural networks and obtaining training data may not be feasible. However, there has been mixed success in generalizing multiple fault scenarios using neural networks, particularly if the input data structure is favorable.

Chapter 3

Methodology

This chapter introduces the basic structure of the DC electric power system components and discusses the development of the corresponding mathematical model. Then the hierarchical control paradigm is given, with a focus on the fault detection and diagnosis methods used at each level. Lastly the coordination of the three hierarchical supervisory methods is detailed.

3.1 DC Electric Power System Model

Electric power systems are made up of a variety of components that make the dynamics of the system very complex. In this section the fundamental components of a DC electric power systems for modern spacecraft are described. Next, a general strategy for the development of a mathematical model of the power system is defined. Then an overview of the communication layer, which consists of the location of sensor placement, and data acquisition techniques is provided. This work focuses on the Advanced Modular Power Systems (AMPS) hardware developed at NASA Glenn Research Center. The AMPS project aims to design and implement a DC power system comprised of the fewest number of base components, that are easily interchangeable in the event of a failure. These devices include, Main Bus Switching Units (MBSU), Secondary Bus Switching Units (SBSU), Power Distribution Units (PDU), DC-DC Converter Units (DDCU), Battery Charge-Discharge Units (BCDU), among a few others. The behavior of each component is described briefly below. More detailed information about the hardware can be found in [81]. Fig. 3.1 shows an example

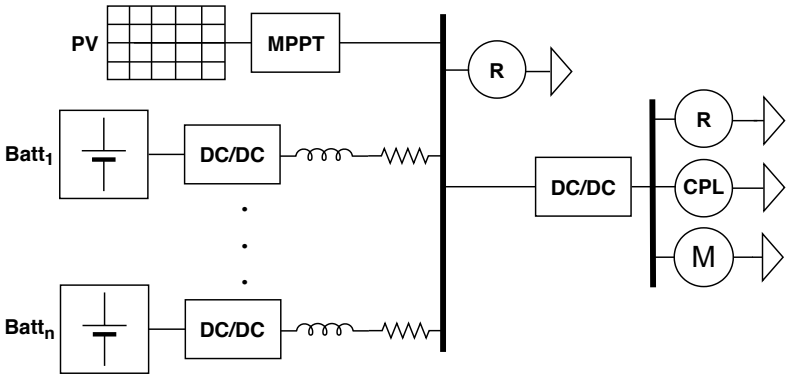


Figure 3.1: One-line diagram of a DC electric power system for spacecraft

of a DC electric power system for spacecraft.

3.1.1 DC Power System Components

(a) **Switchgear**

Switchgear modules primary function is to provide on/off control to sections of the power system. The switchgear also adds fault protection to the system. Depending on the level of current traveling through the switch, different circuit components (i.e. solid state, digital) may be applied, but the same overall functions are met in each case.

(b) **Converters**

The main function of power converters is to change the voltage across the power distribution feeders. Voltages may need to be increased to improve transmission losses, or decreased to meet the acceptable range for certain loads. Power converters on spacecraft can be DC/DC or DC/AC. Converters also typically provide galvanic isolation, which adds a level of fault protection.

(c) **Photovoltaic Arrays**

Photovoltaic (PV) sources are often connected to the electrical distribution network through a regulating device and the control of that device defines the dynamic response characteristics of the source [82]. The model of the PV defined in [83] emulates the dynamic behavior of the array as shown in Fig. 3.2, where the maximum power output of the array occurs at the peak of the curve. PV arrays are often controlled via maximum

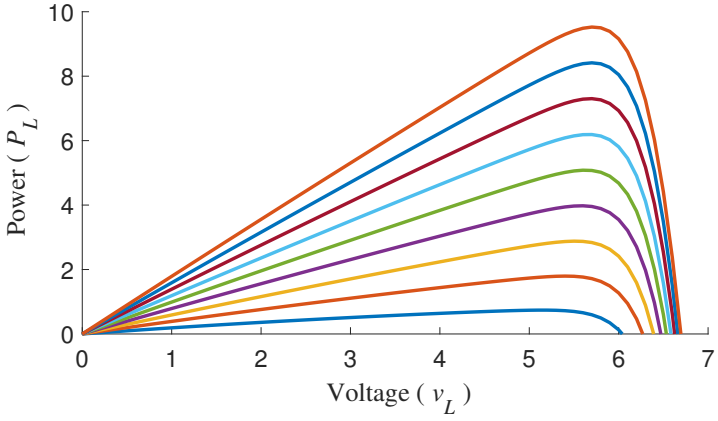


Figure 3.2: P-V characteristics of a solar array

power point tracking (MPPT) through a DC/DC converter, where the objective is to maximize the power output at a given bus voltage. Many algorithms have been developed for MPPT, including perturb and observe (P&O), hill-climb, and incremental conductance (IncCond) [84]. These algorithms successfully manage the PV to operate near the maximum power point, unaffected by the behavior of other subsystems. Therefore, the maximum power point (peak of the P-V curves) for each level of solar irradiation are used to model the PV array as a constant power source.

(d) **Batteries**

The distributed battery systems are responsible for maintaining bus voltage within a standard range. The effects of the other subsystems, such as the PV array and loads, creates challenges for the battery systems to maintain the proper bus voltage. In this study, a droop control method is used for bus voltage regulation as it is the most common method in microgrids. Other types of voltage control schemes can also be used and depending on the approach that is taken, there can be significant impacts on the dynamic behavior and stability of the system.

Droop control is a linear control whose main concept is to provide load sharing while accounting for the generating capacity of each load so that no source is overloaded. This paper uses the model for distributed battery systems operating under droop control from [85]

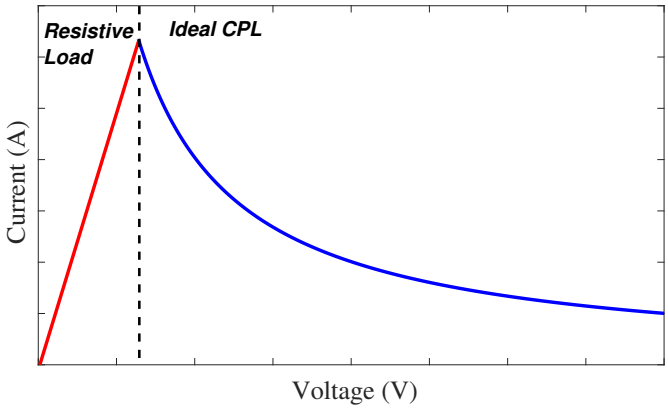


Figure 3.3: Quasi-steady state behavior of the DC/DC converter as a function of the input voltage

(e) **Loads**

A variety of load types may exist on a spacecraft, especially in the case of human missions. From an electrical perspective, the basic types are restive loads (R), AC loads such as motors (M) and pumps, and constant power loads. Constant power loads (CPLs) in the system considered are modeled as current sinks, where the constant current for each CPL is equal to the CPL power divided by the CPL voltage. In modern spacecraft, a DC/DC converter connects the loads to the main bus to regulate the source subsystem voltage to the CPL voltage. CPLs often have a minimum input voltage requirement, V_0. If the input voltage falls below V_0, the converter acts as a passive load. Fig. 3.3 shows a sample behavior of the DC/DC converter.

3.1.2 Linear Distribution System Model

A key challenge in the design of a model-based fault detection and diagnosis method is the construction of the model. A good model should accurately represent the behavior of the system. Due to the changing nature of space power systems, the model must be able to easily adapt to many power system network configurations. For example, a human rated space habitat such as the international space station (ISS) typically begins with a small number of buses and loads, but will grow as additional modules are added for new services and capabilities. Although certain hardware and communication elements may differ between

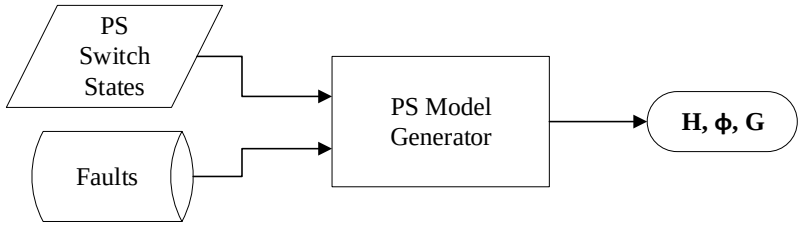


Figure 3.4: Block diagram of the model generation process

DC power systems manufacturers, the fundamental physics of each microgrid must generally abide by the same principals.

The FDD tool is able to automatically generate the power system model by defining the matrices **H**, **Φ**, and **G**. The model generation process uses the distribution switch states and faults to produce the model matrices as shown in Fig. 3.4. The system state vector, x, are the distribution system bus voltages. The behavior of the DC power system in the steady state (no changes in generation or load) does not produces changes in the system state over time. Therefore, the state update matrix **Φ** is equal to an $n \times n$ identity matrix. The model does not rely on a particular control strategy to regulate the bus voltage. Instead it uses the system measurements to converge around the operating point. Therefore the state space model does not take any inputs u, and the input distribution matrix, **G** is not needed. The most critical aspect of the model development is the output matrix **H**.

To generate the output matrix for the DC distribution system, the connectivity of the grid must be defined. In this application, a list of all lines and nodes are generated in a JavaScript Object Notation (JSON) file. Next, state measurements from the switchgear are used to determine the topology of the network. Once the connections in the system are set, the admittance of each line is determined. The branch admittance between each bus, Y, is defined as one over the impedance, Z, between each bus. Using this information the discrete -time linear model model is constructed by modifying the solution for the AC power flow as described in [86]. For DC spacecraft the line inductance is considered negligible, so that the impedance is purely real. The basis of the model consists of generating a connectivity matrix, **Y**, that relates all bus voltages to the line currents. This matrix can

be automatically generated using two rules,

$$\text{Diagonal elements } \mathbf{Y}_{kk} = \text{sum of admittances connected to bus } k$$

$$\text{Off-diagonal elements } \mathbf{Y}_{kn} = -(\text{sum of admittances connected between bus } k \text{ and } n),$$

$$\text{for } k \neq n.$$

After the connectivity matrix is defined we can develop the nodal equations for the distribution system,

$$I = \mathbf{Y}V \tag{3.1}$$

where I is the vector of currents injected into each bus, and V is the vector of bus voltages. For bus, k, the corresponding nodal equation is

$$I_k = \sum_{n=1}^{N} \mathbf{Y}_{kn} V_n. \tag{3.2}$$

The real power at any bus k is

$$P_k = V_k I_k. \tag{3.3}$$

These equations are used to generate the output matrix $\mathbf{H}$ that relates the measurements to the state variables. For each measurement time-step the switch states are analyzed and generates the $\mathbf{Y}$ matrix. To generalize the process of creating the model, a connection object is defined as shown in Fig. 3.5. The connection object contains information about the basic parameters nodes of each line within the electric power system. This includes the names of the orbital replacement unit (ORU) on each side of the connection, and name of the sub-component on either side of the connection. The sub-component may consist of a the switch tied to the distribution line, or the port name of the power electronic unit such as a BCDU. In addition, the resistance and inductance of the distribution line are included. The connection object definition is captured in Table 3.1.

A list of the connection objects, and the most recent measured switch states, allows the software to automatically generate the $\mathbf{Y}$ matrix using the equations defined above. If a line is connected to each of its nodes, then it is considered active. Then the admittance

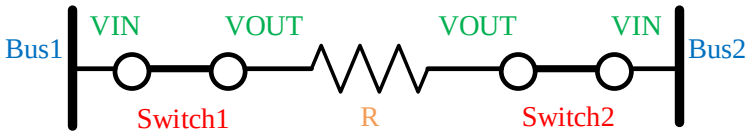


Figure 3.5: One-line diagram of a distribution system connection

Table 3.1: Connection Object

Value	Description	Type
oru1	name of the primary bus	string
oru2	name of the secondary bus	string
switch1	name of the primary switch	string
switch2	name of the secondary switch	string
node1	bus number of the primary side	int
node2	bus number of the secondary side	int
resistance	resistance of the line (Ω)	float
inductance	inductance of the line (H)	float
element	name of the spacecraft unit	string

of the line is used to populate the connectivity matrix as seen in Algorithm 1 is used to generate the connectivity matrix, $\mathbf{Y}$.

Algorithm 1 Generate Connectivity Matrix

for each connection **do**
  i = node1, j = node2
  $R =$ resistance of the line
  if the connection is active **then**
    $\mathbf{Y}_{ij} = -1/R$
    $\mathbf{Y}_{ji} = -1/R$
    $\mathbf{Y}_{ii} = 1/R + \mathbf{Y}_{ii}$
    $\mathbf{Y}_{jj} = 1/R + \mathbf{Y}_{jj}$
  end if
end for

Once the connectivity matrix has been defined, the output matrix, $\mathbf{H}$, can be written. The the rows of the output matrix are populated by iterating through the list of connections. Each connection contains a primary and secondary switch. First, the state of each connection must be determined via the power system telemetry. There are four possible states is depicted in Fig. 3.6, both switches closed, both switches opened, primary switch opened, and secondary switch opened. For each connection, the number of rows added to

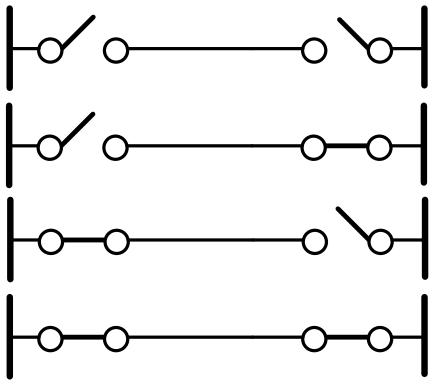


Figure 3.6: Four possible switch states of a distribution connection: *both open*, *primary open*, *secondary open*, *both closed*

the output matrix correspond to the number of sensors in the connection. For example, two bus switchgear modules add an additional six rows to the output matrix, to represent the two sets of input voltage, output voltage, and input current measurements found on the primary and secondary end of the connection. The four possible rows of the output matrix will be represented by $\mathbf{H}^{VIN}$, $\mathbf{H}^{VOUT}$, $\mathbf{H}^{IIN}$, and, $\mathbf{H}^{IOUT}$, which corresponding to the input voltage, output voltage, input current, and output current measurements respectively.

Algorithm. 2 outlines the process for assembling the output matrix. In the algorithm, the node numbers and switch states are assigned to the relative primary and secondary sides of the connection. Then for each measurement, a row is added to the output matrix using a simple formulation based on the switch states. For any combination of switch states, $\mathbf{H}^{VIN}$ always relates to a single state corresponding to the bus voltage. All other rows can be determined using Table. 3.2 based on the states of the primary and secondary switch. Where $\mathbf{I}_{i*} := col_i \left[\mathbf{I}^T\right]$ for $(i = 0, 1, 2, m)$. $\mathbf{I}$ represents an $n \times n$ identity matrix, where n is the number of power distribution buses. In addition, another state must be added for measurements that that are not be connected to the active part of the power system. An example of this would be a load that has been shed, and is currently disconnected for the network via an open actuator. In such a case, currents and voltages are equal to zero. To model these components, an additional state may be added to the DSE in the last column. Any measurements that are expected to equal zero should have a corresponding row in the output matrix that relate the the new state. Let $\Theta^T = [0, 0, 0, ..., 0, 1]$ denote the vector

Algorithm 2 Generate Output Matrix

```
for each end of the connection do
  if primary then
    i = node1
    j = node2
  end if
  if secondary then
    i = node2
    j = node1
  end if
  for each measurement do
    H^VIN = I_i*
    Use Look-Up Table. 3.2
  end for
end for
```

that relates measurements to the zero state.

Table 3.2: Output matrix look-up

	Primary Switch Open	Primary Switch Closed
Secondary Switch Open	$\mathbf{H}^{VOUT} = \Theta$	$\mathbf{H}^{VOUT} = \mathbf{I}_{i*}$
	$\mathbf{H}^{IOUT} = \Theta$	$\mathbf{H}^{IOUT} = \Theta$
	$\mathbf{H}^{IIN} = \Theta$	$\mathbf{H}^{IIN} = \Theta$
Secondary Switch Closed	$\mathbf{H}^{VOUT} = \mathbf{I}_{j*}$	$\mathbf{H}^{VOUT} = \mathbf{I}_{i*}$
	$\mathbf{H}^{IOUT} = \Theta$	$\mathbf{H}_i^{IOUT} = -\mathbf{Y}_{j,i}$ $\mathbf{H}_j^{IOUT} = \mathbf{Y}_{i,j}$
	$\mathbf{H}^{IIN} = \Theta$	$\mathbf{H}^{IIN} = \mathbf{H}^{IOUT}$

In addition to the power system switch states, faults are also used to set the mathematical model. To add faults in the system a mapping that relates sensor locations to indices in the matrix is created using the user input connections file. Sensor and communication fault models can be formed by setting rows in z and $\mathbf{H}$ equal to zero. For multiplicative faults that alter some physical parameter of the power system, such as line-to-ground and stuck switches, the connectivity of the model must be updated through $\mathbf{Y}_{bus}$, to generate the corresponding model. Using this technique, any combination of faults can be inserted into the power system model at one time.

Note that the power system model is derived entirely from the DC nodal equations

which is based on the fundamental physics of a DC power flow. The simulation used to test this software relies on a proprietary MatLab/Simulink library of electric power system components to model the EPS. The development mathematical model used in the fault detection and diagnosis software does not rely on the design of the power system simulation. The power system simulation is treated in a manner similar to that of real hardware, in that not all of the dynamics of the system are assumed to be known. Hence the simulation allows us to test the fault detection and diagnosis software for modeling errors in the state-space representation.

3.1.3 Faults in DC Power Systems

Line Segment Faults

A *line-to-ground* fault is said to occur when a path is created between either the positive or negative pole and the ground is created, and is the most common type of fault [87]. A *line-to-line* fault has occurred when a path between the positive and negative pole is created in a line segment. Both of these fault types are electric faults commonly referred to as short circuit faults. Depending on the location, short circuit faults can prevent loads from being served, impede the transmission and distribution paths, as well as loss of energy storage and generation capabilities. It is critical to detect and isolate short circuit faults in order to prevent further damage to the power system components such as generators, converters, loads, and energy storage. Failure to do so can lead to blackouts in the power system. Similarly, a short to ground fault can occur on a bus bar. In this case, the protection system needs to isolate the fault from all segments of the power system. This task is especially important for mission critical power systems such as spacecraft power systems. Load redistribution is needed to continue supporting all loads in the event of a short circuit fault.

Sensor Faults

A commonly overlooked aspect of fault detection and diagnosis in DC power systems are faults in the measurement data. This can occur in a variety of ways. A *sensor offset* occurs

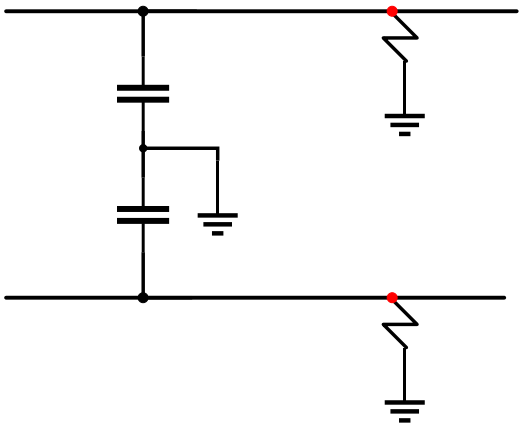


Figure 3.7: Line to ground fault

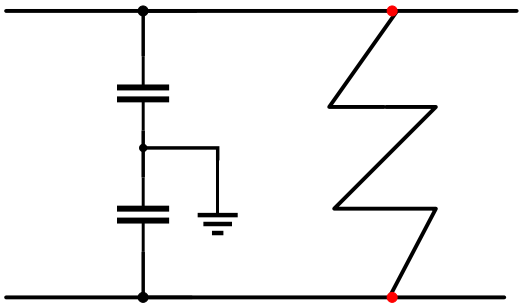


Figure 3.8: Line to line fault

when a sensor reading is biased by a constant value. For example a voltage sensor measuring a 120V bus, may produce a value near 130V in the event of a 10V sensor offset. *Excessive sensor noise* is said to occur when the noise level exceeds the maximum rating of the sensor set by the manufacturer. Lastly, a *stuck sensor* takes place in the event that a sensor reads a constant value regardless of the physical value of the measured parameter.

Switch Faults

Switches that connect transmission and distribution lines through a bus bar are also prone to failure. A *switch-stuck-open* fault is said to occur when a switch state reads closed, but the physical switch state is open. Conversely, a *switch-stuck-closed* fault occurs when a switch state reads open but is physically closed. This may be caused by a switch being fused shut after a thermal inrush or a failure in the state measurement. This type of fault is only made possible by mechanical switches. FET-based switches would not experience this type of behavior.

Communication Faults

The data and measurement acquisition of the electric power system is a critical component in reliable system operation. The communication system consists of the components and software which send and receive information from the reactive layer to the component layer and the global layer of the power system controller. It is vitally important for an autonomous system to be able to detect a failure in the communication system and respond in the appropriate manner. Research in electric power system fault monitoring often overlooks the importance of these failures.

Communication faults can occur in a number of varieties based upon the data acquisition implementation. Two types are discussed in this work, namely *stale data* and *missing data* communication faults. Stale data faults happen when the communication path from the reactive layer to the component layer is blocked, resulting in no measurement updates to a component. Because the components store the latest set of measurements in their registers, they emit the same data to the system level controller every time data is requested, resulting in a stale set of data. Missing data occurs when the communication path from the

component layer to the global level is blocked, resulting in a missing packet in the global data acquisition.

Source Faults

Source faults capture the faults and failures associated with power generation devices, in this case solar arrays and batteries. The design and construction of said devices vary greatly, thus making it difficult to capture the types of faults associates with these devices for a general study such as this. That being said, source faults will be modeled as line to ground shorts at the input of the source. This will provide the loss of generation and energy storage to the system, which creates challenges for other APC services such as automatic reconfiguration and load scheduling.

Load Faults

Similar to source faults, a large number of unique load types may exist on a spacecraft. In general, there are resistive loads such as heaters, constant power loads from certain electronic loads, as well as AC loads like the pumps needed for the life support system on manned missions. Based on the tendencies of any given load, failure types may vary, and may be difficult to capture at a quasi-steady state frequency. Signal models using high frequency data would offer one solution to this problem. However, the capability in this application offers only low to medium frequency measurements. Therefore, all load faults will be modeled as line to ground fault at the input of each load.

Power Electronic Faults

The power electronics (DC-DC converters and regulators), are a critical part of DC microgrids. The rapidly growing area of research has improved the cost and efficiency of these devices over the past few decades. Due to the high switching frequency, many of the fault types associated with DC power electronics occur at a very high frequency ($\leq 1ms$) and, thus, are beyond the scope of this research. However, additional information about these faults and a potential non-linear application for power electronic health monitoring is provided in the following chapter.

3.2 Hierarchical Control Framework

Modern electric power control systems utilize a hierarchical framework [61]. This design is useful in dealing with a number of dynamic phenomena occurring at different time scales. For example, DCDC converters responsible for battery charge and discharge regulate their voltage set points based on local measurements; however, these set points can also be changed by a command from the centralized controller based on the objectives of the entire system. The first level of control is the reactive layer where high frequency data is processed and physical EPS components are managed. The second level of control, the component layer, is responsible for handling a box-level device the electrical power system. In the case of aerospace power systems, component layer control operates over a particular device such as a bus switching unit, or a DC/DC converter. Lastly is the central level controller which has a global perspective of the power system, receiving data from every bus in the power system.

This research combines and embeds the strengths of multiple fault detection and diagnosis techniques, including automatic protection, rules-based fault detection, model-based fault detection and diagnosis, and active fault diagnosis. In the following sections, the implementation of these techniques is described for all three levels of the controller. Lastly, the interactions and coordination between the three levels are illustrated.

3.3 Reactive Layer

The reactive layer is made up of high frequency devices such as field programmable gate arrays (FPGAs), or programmable logic controllers (PLCs). These components obtain localized information from the power system, and perform automatic high speed actions on the system based on simple procedures [88]. These actions are designed to quickly protect the system from further damage in the event of a fault. The reactive layer is also responsible for taking measurements from sensors in the form of voltage, current, power, temperature at high speeds ($\sim$1MHz). Data is passed from the reactive layer devices upward to the component layer at a slower rate based on the capability of the communication system.

Arguably the most important role of the reactive layer is automatic protection. Due

to the harmful nature of certain faults, action must be taken as quickly as possible to avoid further damage. In the case of electric power systems, a ground fault may result in high current flow through the power system components. High current flow is often coupled with a large thermal inrush which may cause critical damage to transmission lines, switches, and power electronics. Therefore it is important for the reactive layer to respond rapidly to safe the system in the even of critical faults. For a DC electric power system, this is typically done in the form of current limiting circuit breakers [87]. Any action taken, or alarm detected by the reactive layer is sent to its corresponding component level controller through a status message.

3.3.1 Short Circuit Fault Detection

A short circuit on a DC bus causes devices connected to the bus such as generators, bus capacitors, and energy storage to experience excessive current, and voltages at both the loads and sources can drop. Although DC fault protection is less mature than its AC counterpart, various detection schemes have been proposed. They include large current change and oscillation pattern methods [89], dv/dt [90], di/dt and d^2i/dt^2 [91], traveling wave and wavelet-based methods [92] [93], and model based techniques [94]. The scope of this dissertation does not include design and implementation of reactive layer protection. Instead it is assumed that a bidirectional di/dt method will be implemented to quickly detect and the systems circuit breakers will isolate the fault.

Under normal conditions, line-to-ground faults, which are the most common, are straightforward to detect. Detection becomes more challenging for systems where the bus is ungrounded, or high-resistance grounded. For an ungrounded system, a single line-to-ground does not cause any fault current, and the system can continue operating. However, the impact of a second line-to-ground can be detrimental to the power system if the first fault is not cleared. Typically, spacecraft power systems are grounded to the spacecraft chassis. In addition, a large fault resistance can also make short circuits difficult to identify. This fault known as a *high-impedance short circuit* has a reduced fault current almost indistinguishable from a load current, and therefore can go undetected especially in systems with high levels of sensor noise. Current differential protection has been implemented to

compare and quickly detect high-impedance short circuits across buses. These methods are very successful at detecting high-impedance short circuits, but often require high speed data transmission to quickly identify the fault.

3.4 Component Layer

The component layer can be thought of as the "box-level" controller in which a local device is monitored and controlled. For space power systems this may include devices such as bus switch gear modules, power converters, batteries, and solar arrays. The component layer receives telemetry and messages from the reactive layer at a medium-frequency (∼5-100Hz). The telemetry can be processed for faults. This level of the FDD scheme is made up of simple physics-based rules, and rules made specifically for the hardware components based on the users expertise. Since the component layer FDD algorithms do not have a central perspective of the power system data, they can often produce false alarms especially in the event of faulty power system data. Therefore, fault messages are sent to the global level for fault validation. Moreover, in the event of a fault message coming from the reactive layer, the component layer is responsible for collecting the message and sending it to the global layer for validation. The following subsections detail the algorithms (designed in collaboration with Matthew Granger of NASA Glenn Research Center) that have been developed for each component layer controller.

3.4.1 Bus Switchgear Module

Reactive Layer Status

The bus switchgear module pends on messages from the reactive layer to receive fault status and alarm information. For the bus switchgear module, this includes a trip status, which indicates that a relay or circuit breaker has opened due a sensed over current. If a reactive layer status ($rStatus$) is detected, the component layer creates a fault object which includes information about the fault type and location of the fault. Algorithm 3 tracks the status registers for all relevant fault types, $faultType_n$, where $n = 1, 2, ..., m$ for m total number of fault types.

Algorithm 3 Reactive Layer Status

```
for n = 1 : m do
  if rStatus = faultType_n then
    FaultDetected(faultType_n)
  else
    FaultNotDetected(faultType_n)
  end if
end for
```

Whenever FaultDetected() is called in the central controller Algorithms 4 is used to generate a list of all active faults. Any fault status is checked to see if the fault has been observed k consecutive times before passing the fault message up to the central level controller. This prevents spurious messages from reaching the central level due to noise and disturbances.

Algorithm 4 Fault Detected

```
if faultMap → Contains(faultType_n) then
  repititions = faultMap → GetRepititions(faultType_n)
  if repititions < maxRepititions then
    repititions = repititions + 1
  end if
  if repititions = maxRepititions then
    PublishFaultToCentralLayer(faultType_n)
  end if
else
  faultMap → AddFault(faultType_n)
end if
```

Switchgear modules also use current, voltage, and state measurements to detect faults. For example, a switch or actuator that has failed stuck open or closed. Here the former indicates that a switch state reads closed but is actually open, and the later suggests that a switch state that reads open is actually closed. Measurements $VIN, VOUT, IIN, IOUT$ refer to input voltage, output voltage, input current, and output voltage respectively. Additionally, the *state* variable refers to the binary status of the switch (open or closed). The Algorithm 5 is used to detect the failed switch fault types. Where ΔV_{max} represents the maximum potential difference over a closed switch, and I_{max} represents the maximum current that can flow through an open switch. Both of these values are predefined by the system operator and must be set to tolerate the noise in the measurements.

Algorithm 5 Switch Stuck Open/Closed

for each switch **do**
  $\Delta V = |VIN - VOUT|$
  $I_{avg} = \frac{1}{2}(IIN + IOUT)$
  if $\Delta V > \Delta V_{max}$ **and** $state = closed$ **then**
    FaultDetected($switchStuckOpen$)
  end if
  if $I_{avg} > I_{max}$ **and** $state = open$ **then**
    FaultDetected($switchStuckClosed$)
  end if
end for

The component layer strategy defined here uses rules based on physical first principles. This helps to reduce complexity at the central layer, as well as quickly detect faults within the device. Waiting for multiple data points and allowing some tolerance in the alarms due to system noise will reduce the number of false detections sent to the central layer. Since the central layer is responsible for final fault verification, it is critical to minimize additional diagnostic computation caused to false alarms.

3.5 Central Layer

In the central (or global) layer of the of the FDD controller, measurements are received from the entire power system. At this level, measurements from the entire power system are collected and used for fault detection and diagnosis. The data, however, is sent at the lowest frequency of all three levels (∼1-10Hz). For the purposes of FDD there is a trade off between sampling resolution and measurement availability. Lower sampling frequency results in longer fault detection and diagnosis times. Therefore faults that are needed to be isolated quickly to avoid physical damage, such as the line to ground fault, should be detected at the lower levels. Faults that may exist for longer periods of time, such as sensor faults, are well suited for diagnosis at the central level. However, data may be insufficient for sensor fault diagnosis due to the lack of system wide information. Therefore an important design consideration is the balance of FDD across all three control levels based on the criticality and isolability of the fault.

In the application of DC power systems, data acquisition can be slow and latent. It is

important to synchronize the collection of data to the central level to ensure that the data measured data was sampled during the same operating state. If separate sensors gather data at slightly differing moments, there is risk for the system to change its operating state. For example, opening a distribution switch can result in false detections if the measurements a current measurement was collected slightly before the switch opened, but the voltages we collected after. Errors in measurement and communication equipment are unavoidable, thus perfect data synchronization can never be truly achieved. Intelligent data acquisition techniques such as synchronized polling and timestamp checking should be used to collect data as consistently as possible. These events can be viewed as disturbances in the system. Detection tools such as the fault filter help to minimize the impact of these disturbances.

In addition to the data collection service, many other power system control services exist in the central control layer. These services include, but are not limited to, energy storage balancing, energy forecasting, load scheduling, fault detection and diagnosis, and network reconfiguration. Each power system control service is given a particular set of objectives, and is able to communicate with the other services via synchronous or asynchronous messages to accomplish the objectives. In the case of fault detection and diagnosis the central layer is used to validate fault indications from system-wide measurements and vet fault alarms sent from the component and reactive layers. The central layer of the FDD tool is where most of the computational processing is done to diagnose faults. A combination of dynamic state estimation and physics-based rules are applied to diagnose faults quickly and accurately. Additional technical details about the central level FDD method are discussed in the subsections below.

3.5.1 Parameter Estimation

To improve the accuracy of the power system model, parameter estimation can be used to approximate uncertain variables. This will improve the accuracy of the DSE by providing a more accurate model of the system, and in turn, providing improved fault detection and diagnosis. This is an important step in minimizing the modeling errors of the power system simulation and hardware testbed. Poor estimation of the parameters can lead to large numerical errors in the state estimates. For the application of a DC power system,

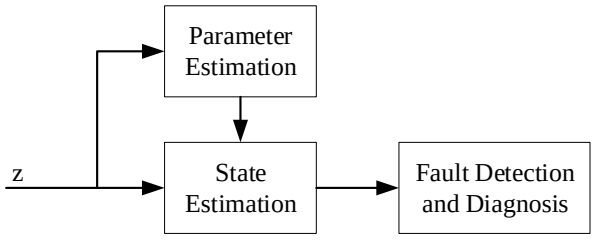


Figure 3.9: Block diagram of parameter estimation used for generating state estimation model

the parameter estimator uses a maximum likelihood estimation (MLE) to determine the distribution line impedance from power system measurements. Upon request from the controller, measurements (z) are sent to the parameter estimator, where the impedance between nodes is approximated and sent to the state estimator model as shown in Fig. 3.9. This process is performed on system start-up, but can also be called periodically as the system degrades. The parameter estimates are then used in the state estimator model, thus providing greater accuracy in the state estimates.

The parameter estimator analyzes power system measurements $x_1, x_2, ..., x_N$, to determine the best approximations for the line impedance. The probability of observing the given data as a function of some parameter θ is

$$lik(\theta) = f(x_1, x_2, ..., x_N|\theta). \tag{3.4}$$

The MLE of θ is the value of θ that maximizes $lik(\theta)$ based on the observed data. For independent identically distributed random variables (iid), the likelihood simplifies to

$$lik(\theta) = \prod_{i=1}^{N} f\left(x_i|\theta\right). \tag{3.5}$$

Using the fact that the logarithm is an increasing function, it is often simpler to maximize the log of the likelihood function, which produces the equivalent result.

$$l(\theta) = \sum_{i=1}^{N} log\left(f\left(x_i|\theta\right)\right. \tag{3.6}$$

Given that the measurements are normal random variables with mean μ and variance σ^2, the probability density can be written as,

$$f(x_i|\mu,\sigma^2) = \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(x_i-\mu)^2}{2\sigma^2}\right] \tag{3.7}$$

The log of the likelihood function for this distribution with respect to the parameters μ and σ is

$$l(\mu,\sigma) = -nlog(\sigma) - \frac{n}{2}log(2\pi) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(x_i-\mu)^2, \tag{3.8}$$

so that,

$$\frac{\partial l}{\partial \mu} = \frac{1}{\sigma^2}\sum_{i=1}^{N}(x_i-\mu) \tag{3.9}$$

$$\frac{\partial l}{\partial \sigma} = -\frac{n}{\sigma} + \sigma^{-3}\sum_{i=1}^{N}(x_i-\mu)^2 \tag{3.10}$$

Setting these equal to zero yields the MLE estimates for the mean and variance,

$$\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{3.11}$$

$$\hat{\sigma}^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i-\hat{x})^2 \tag{3.12}$$

These equations are used to acquire the MLE of the impedance based on voltage and current measurements in the power system. Using measurements V^{IN}, V^{OUT}, and I from each bus, the resistance R can be estimated as

$$R = \frac{V}{I} \approx \frac{|\left(V_1^{IN}+V_1^{OUT}\right) - \left(V_2^{IN}+V_2^{OUT}\right)|}{(I_1+I_2)}. \tag{3.13}$$

One caveat of the autonomous parameter estimation procedure is that each line must be actively supplying power to produce an estimate. Therefore a switching scheme may be used to iterate through different network configurations to ensure that each line is able to be estimated. Further research in this area should be done to minimize the impacts to the loads during this process.

3.5.2 Power System Disturbances

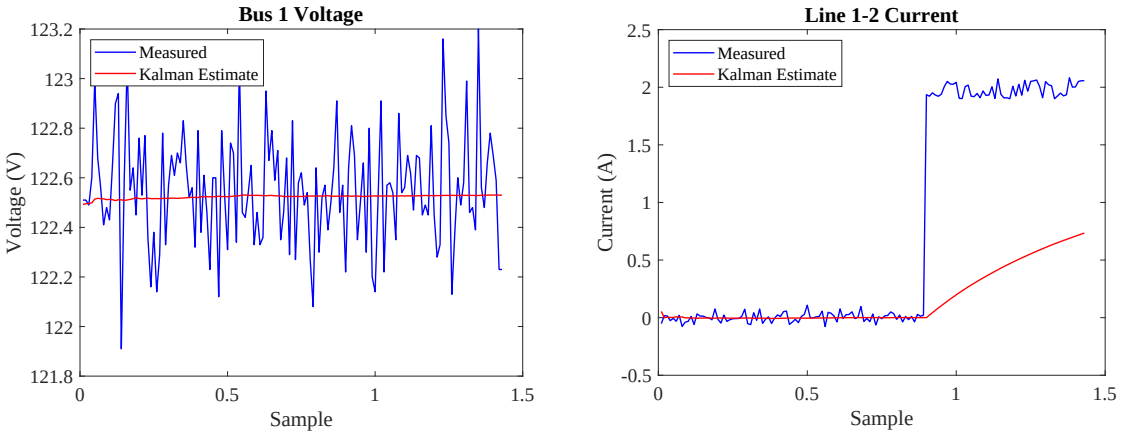
One challenge in performing dynamic state estimation is the ability to observe random jumps in the process state variables. The model for the DC electric power system can be viewed as a jump Markov linear system. For space power systems stochastic jumps in voltage, current, and power can be caused by events such as opening and closing of distribution switches or transitions in and out of eclipse. The challenge for the state estimator is to differentiate these disturbances from true faults.

To solve this issue the process noise covariance, **Q**, must be set properly. One of the frequently challenging aspects in Kalman filter design is the determination of the matrices **R**, **Q**, and **P**. The observation covariance, **R**, can be generated empirically, or by the manufacturers specifications of the sensors. An identity matrix often functions as a good choice for the a posteriori estimate covariance **P**. In particular, **Q** tends to cause confusion because information about the process noise is likely unknown. This parameter depends on the accuracy of the system model including any assumptions that were made about the plant that may lead to noise in the states.

For the case of a DC power system, the distribution of power should closely follow the laws of physics defined by Kirchoff's current and voltage laws. This straightforward linear model suggests that **Q** could be set equal to a matrix of all zeros otherwise known as a null matrix. This setting helps improve data smoothing as shown in Fig. 3.10(a). However, after a random jump in a measurement, such as an increase in current, the Kalman filter is unable to quickly track the new system state as seen in Fig. 3.10(b). This lack of convergence impedes the ability of the state estimator to diagnose faults.

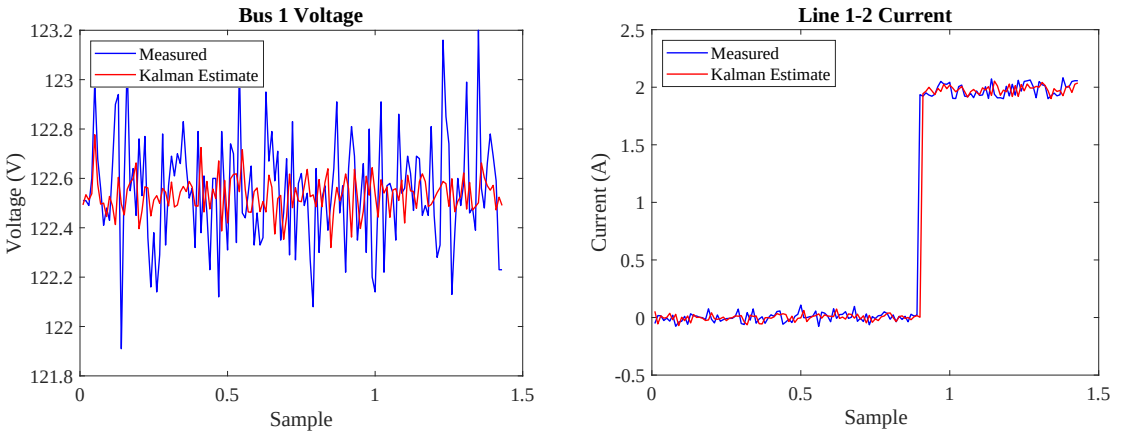
An alternative to the zero process covariance matrix, the identity matrix can be used to decouple each process state and therefore allow for better convergence after a random jump as shown in Fig. 3.11(b). Unfortunately, the decoupling of the process states also results in poorer estimates and data smoothing as seen in Fig. 3.11(a). This decreases the accuracy of the state estimates used by other functions in the system, such as generation control.

In order to incorporate the best features of each of these parameter settings, an adaptive strategy may be implemented. In this setup, the fault detection service will determine how



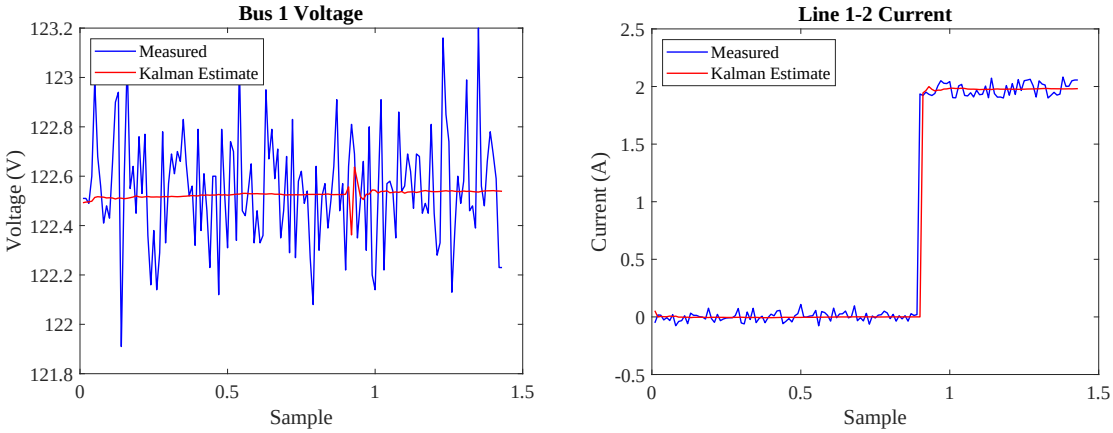
(a) Voltage Signal (b) Current Signal

Figure 3.10: Kalman filter with zero process covariance matrix



(a) Voltage Signal (b) Current Signal

Figure 3.11: Kalman filter with identity process covariance matrix



(a) Voltage Signal (b) Current Signal

Figure 3.12: Kalman filter with dynamic process covariance matrix

to set the process covariance matrix, **Q**. In order to provide quick convergence after a random jump in a state or measurement, the process covariance must be equal to the identity. When the measured data matches with the system model, the process covariance should be set to a matrix of zeros to provide optimal data smoothing. To achieve this, the residuals may be checked on every innovation to determine if a jump has occurred in the system. A jump in this case could be a random disturbance in the process due to a switching event or a change in load, or a fault in the system. Using a jump threshold (based on the noise parameters of the system) the process covariance can be determined. If one or more residuals exceeds the disturbance threshold, **Q** is equal to the identity matrix, and when all of the residuals remain under the threshold **Q** is set to a null matrix. The performance of this method is evident in Fig. 3.11(a) and Fig. 3.11(b). In this simulation, a disturbance due to a change in load occurs around 0.9s. Notice that both before and after the disturbance, the optimal data smoothing is achieved.. Then, the disturbance transition time is reduced to a few innovations. This is due to the change in process covariance during the transition, allowing for quick convergence. This adaptive solution captures the benefits from each of the two static process covariance matrices.

3.5.3 Handling Numerical Errors

A vector of residuals must be generated to evaluate the fault status of the electrical power system of interest. A convenient property of the Kalman filter is that the standardized innovation sequence, η has zero mean and unit variance. Recall that in order to compute the standardized innovation sequence, the matrix square root of $[\mathbf{H}(k)\mathbf{P}(k|-1)\mathbf{H}^T(k)+\mathbf{R}(k)]$ must be positive semidefinite (PSD). In theory, this quantity is PSD due to the nature of the parameters $\mathbf{H}, \mathbf{P}$, and $\mathbf{R}$. However, numerical round-off error can cause η to lose its PSD characteristic and therefore the computer is unable to compute the standardized innovation. An eigendecomposition of the matrix can be used to mitigate this issue.

When the matrix square root function fails within the controller, an eigendecomposition is taken where for some matrix $\mathbf{A}$, with eigenvalues $\boldsymbol{\lambda}$ and eigenvectors v, such that $\mathbf{A}v = \boldsymbol{\lambda}v$. Then define the matrix $\mathbf{D}$ to be the diagonal matrix of eigenvalues, and $\mathbf{V}$ to be the matrix of eigenvectors as its columns, so that $\mathbf{AV} = \mathbf{VD}$. Thus the eigendecomposition can be taken as

$$\mathbf{A} = \mathbf{VDV}^{-1}. \tag{3.14}$$

In the event that the matrix $\mathbf{A} = \left[\mathbf{H}(k)\mathbf{P}(k|-1)\mathbf{H}^T(k)+\mathbf{R}(k)\right]$ is not PSD, the eigendecomposition is performed. Then the negative eigenvalues of $\mathbf{D}$, which have occurred due to round off error, are changed to zero. This will yield a close approximation to the original matrix and is now PSD, which will allow the computer to compute the matrix square root and estimate the standardized innovation sequence by,

$$\tilde{\eta} = \tilde{\mathbf{A}}^{-\frac{1}{2}}\nu, \tag{3.15}$$

where the approximation of $\mathbf{A}$ is

$$\tilde{\mathbf{A}} = \mathbf{V}\tilde{\mathbf{D}}\mathbf{V}^{-\frac{1}{2}}, \tag{3.16}$$

and $\tilde{\mathbf{D}}$ is the non-negative eigenvalue estimate of $\mathbf{D}$.

An optimal solution to this problem would be to determine the nearest PSD matrix to the original and compute the matrix square root. This problem consists of solving a computationally challenging optimization problem using the bisection method [95]. The

eigendecomposition method presented here works as a reasonable approximation that can be solved quickly. Thus leading to a suitable solution for the limited resources of a flight computer.

3.5.4 Sensor Monitoring via the Sequential Kalman Filter Algorithm

One setback of the multiple model-based approach for fault detection and diagnosis is the compounding computational complexity caused by complex modeling/filtering. The number of models and the number of sensors contained in each model can increase the number of computations necessary to diagnose a fault. In particular, many aerospace flight computers have strict limitations on the amount of processing power a an application may use. Therefore considerations in the design of the FDD approach for computational efficiency should be taken while maintaining diagnostic performance.

One factor that adds to the complexity of the FDD method is the ability to monitor multiple faults as they enter/leave the system. To diagnose multiple faults, the authors of [96] suggest generating additional filters to see if a fault has been added to or removed from the system. This may result in significant memory usage and processing power. Instead, low level rules can assist in guiding the fault detection and diagnosis engine toward the correct filter when changes occur in the system, without the use of additional models. Sensors faults, however are found using the mean test of the Kalman filter. After a sensor fault has been diagnosed, a new model is generated without the faulty sensor. This allows the system to be monitored for additional faults. The supervisory tool must continue to monitor the sensor, for the event that the sensor is repaired, or returns to normal functionality. Rather than implementing a full system model that contains the faulty sensor, the Sequential Kalman Filter algorithm can be used to calculate the residual of a single sensor [97]. This technique can be used after a sensor fault has been diagnosed. The Sequential Kalman Filter is able to determine the state of the sensor (normal or faulty) without generating an entirely new filter.

The Sequential Kalman Filter algorithm described in [98] states that the algorithm is particularly efficient if the measurement noise covariance matrix $\mathbf{R}$ is diagonal. If not, a transformation can be made using an orthogonal matrix $\mathbf{T}$ exists that can make the

transformation matrix $\mathbf{T}^T\mathbf{RT}$ diagonal.

Assuming $\mathbf{R}_k$ is diagonal, the Kalman gain matrix $\mathbf{K}_k$ and corresponding optimal state vector estimate $\hat{x}_{k|K}$ are computed. For clarity, the subscript k is removed, so that we may define the following parameters

$$z_i = \begin{bmatrix} z^1 \\ z^2 \\ \vdots \\ z^n \end{bmatrix}, \qquad \mathbf{H}_i = \begin{bmatrix} H^1, & H^2, & \ldots, & H^n \end{bmatrix},$$

and

$$\mathbf{R}_i = diag \begin{bmatrix} R^1, & R^2, & \ldots, & R^n \end{bmatrix}.$$

The sequential Kalman filter is performed as follows.

For a fixed k, set

$$P^0 = P_{k|k-1} \quad \text{and} \quad \hat{x}^0 = \hat{x}_{k|k-1}. \tag{3.17}$$

Then for $j = 1, 2, ..., n$ compute

$$\begin{cases} K^j &= \left(\frac{1}{(H^j)^T P^{j-1} H^j + R^j}\right) P^{j-1} H^j \\ \hat{x}^j &= \hat{x}^{j-1} + \left(z^j - \left(H^j\right)^T \hat{x}^{j-1}\right) K^j \\ P^j &= P^{j-1} - K^j \left(H^j\right)^T P^{j-1} \end{cases} \tag{3.18}$$

Then the state vector is estimated by

$$\hat{x}_{k+1|k} = \mathbf{\Phi}\hat{x}^n, \tag{3.19}$$

and observation of sensor j is estimated by

$$y^j = H_k^j \hat{x}^j \tag{3.20}$$

Finally, the residual for the estimate is computed by comparing the estimate to the raw

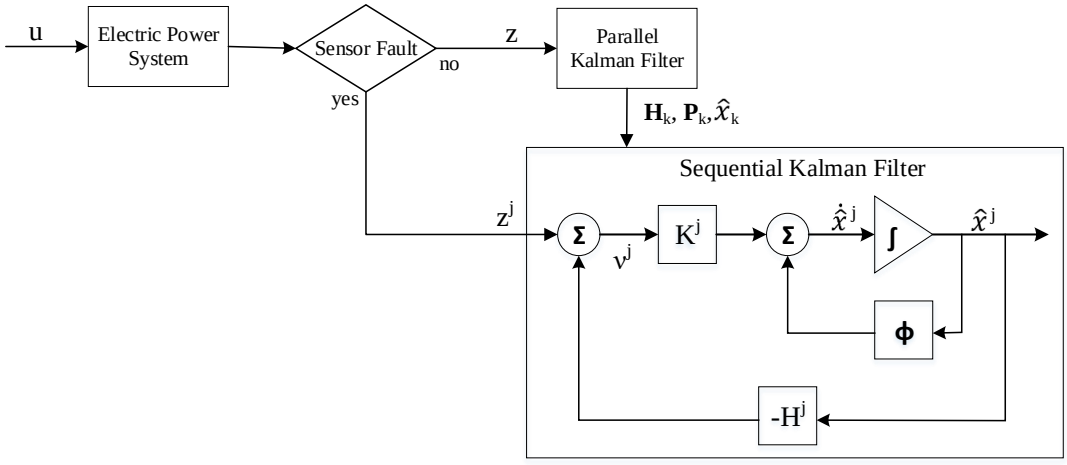


Figure 3.13: Block diagram of the sequential Kalman filter monitoring of erroneous data

value, $r^j = y^j - z^j$. This method can be applied for any number of observations in the system. For the application of this dissertation, only faulted sensors are monitored sequentially. For the case of a single observation, Equation 3.18 is only computed for faulty observation, j. Parameters $\mathbf{H}$, $\mathbf{P}$, and $\hat{x}$ come from the parallel system model as shown in Fig. 3.13. After the estimates have been generated for each observation, the statistical tests for whiteness, mean, and covariance are applied as defined in the previous chapter.

To illustrate the application of the sequential Kalman filter in the FDD process, consider the example of a sensor bias fault. After the fault is diagnosed, a sequential Kalman filter is applied to monitor the faulty sensor after the fault has been diagnosed. If the residuals of the sensor pass the statistical tests over a specified window of time, the sensor can be declared functional and added back into the parallel Kalman filter. This technique allows for the insertion and removal of sensor faults, without requiring redundant filtering of the entire system.

3.5.5 State Estimation - Flight Processor Application

Computers designed for aerospace flight control have limitations on the amount of processing power that can be allocated for a particular application. Computationally intensive algorithms such as the Kalman filter can require large amounts of memory and matrix operations depending on the number of states and observations in the system. To help relieve demands of larger systems, the state estimation model can reduced into multiple smaller systems based on the distribution connections. Each connected piece of the power system can be defined as its own microgrid or island. This application of the state estimator was designed for the purpose of low-powered spacecraft to use a bank of parallel micro-processors to compute the Kalman filter innovations of each island in parallel.

First, an algorithm is performed to determine which states are codependent. Let $\mathbf{\Psi}$ be the boolean matrix that represents the connectivity of power system states. Each row in $\mathbf{\Psi}$ represents one group or island of connected states, where the columns of the state connectivity matrix, $\mathbf{\Phi}$, directly map to the state vector x^T. To generate the connectivity, a search algorithm using the output matrix, $textbf{}H$ is performed. Algorithm 6 explains the method to find the rows of the output matrix with non-zero elements related to a particular state in column j.

The model reduction algorithm is designed for C/C++ implementation, and therefore, all rows and columns will be discussed with the first element of a vector or matrix with index zero. First, Algorithm 6 is used to determine which rows of the output matrix have a non-zero element in a particular column. Then Algorithm 7 checks each of the rows to see if there are any non-zero terms in the columns to be added to the state connectivity matrix.

Algorithm 6 Find Rows with State

```
j = state of interest
for i = 0 : rows in H do
  if |H_{i,j}| > 0 then
    add row i
  end if
end for
```

Algorithms 6 and 7 are visualized in the following example. First, in Fig. 3.14 state

Algorithm 7 Find New States from Rows

$k =$ row in $\mathbf{\Psi}$
$i =$ row of interest
for $j = 0$: columns in H **do**
  if $|H_{i,j}| > 0$ AND $i \notin \mathbf{\Psi}$ **then**
    $\psi_{k,j} = 1$
  end if
end for

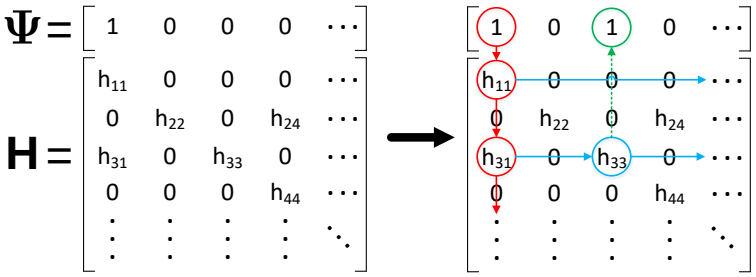


Figure 3.14: Example state connectivity matrix $\mathbf{\Psi}$ and output matrix $\mathbf{H}$ and the first three steps (red, blue, and green respectively) in determining the first set of connected states

connectivity is initialized to a $1 \times n$ matrix of zeros, where n represents the number of columns in the output matrix. Then, the first column of the is set to one, as it is assumed that the output matrix has been generated such that the system is fully observable. Next Algorithm 6 is implemented, shown in red, where the first column is searched for non-zero elements. In this case, rows 0 and 2 have non zero elements. Then Algorithm 7 is performed, shown in blue, where each of the rows is searched for non-zero columns. For this example, row 0 has no other non-zero elements, but row 2 has a non-zero element in column 2. The last step is to add the new state to state connectivity matrix, shown in green. This process continues until there are no new states returned from the Algorithm 7.

After one group of connected states has been collected, another row is added to the state connectivity matrix, where the first zero column of the previous row is set to one. Fig. 3.15 shows this by setting $\mathbf{\Psi}(1,1) = 1$. Then same process described above is repeated to find the second set of connected states. For this example, the next set of connections includes state 1 and 3. This process is continued until there every column of the state connectivity matrix is set to true in exactly one row.

Next, the Kalman filter can be broken into its smallest possible pieces for processing.

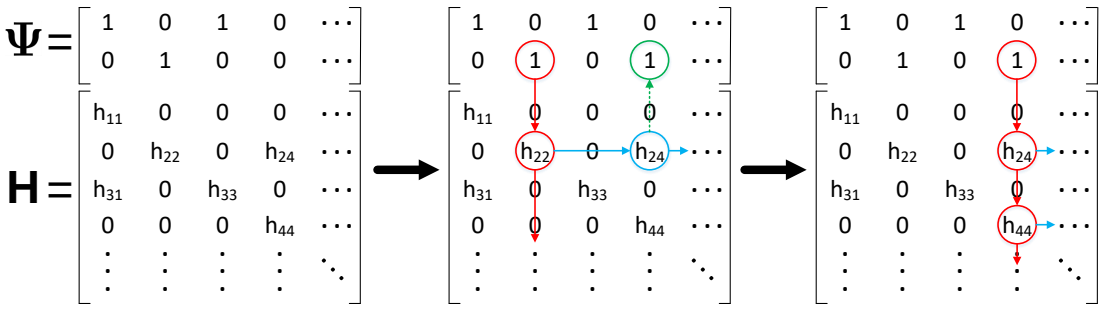


Figure 3.15: Following steps to create the second row of the state connectivity matrix $\mathbf{\Psi}$

To do this, Algorithms 8, 9, 10, and 11 process each row of the state connectivity matrix to determine the corresponding variables $\tilde{\hat{z}}^c$, $\tilde{\hat{x}}^c$, $\tilde{\mathbf{H}}^c$, $\tilde{\mathbf{P}}^c$, $\tilde{\mathbf{R}}^c$, and a vector γ^c, where the elements of γ are equal to the indices of the measurement vector y used in each island. Here $\tilde{\mathbf{H}}^c$ represents output matrix that only includes the connected states in row c of $\mathbf{\Psi}$. These algorithms provide the methods necessary for generating each piece of the partitioned Kalman filer. For this model of the DC distribution system, the state update matrix, $\mathbf{\Phi}$, is equal to the identity matrix, and therefore can be resized as needed for each piece of the algorithm. Once the algorithms for partitioning the Kalman filter have been developed,

Algorithm 8 Get z^c and γ^c

```
for i = 0 : rows in H do
  for j = 0 : columns in H do
    if Ψ_{c,j} > 0 AND |H_{i,j}| > 0 then
      i ∈ γ^c
      z_i ∈ z^c
    end if
  end for
end for
```

Algorithm 9 Get $\tilde{\hat{x}}^c$

```
for i = 0 : rows in H do
  if Ψ_{c,j} > 0 then
    x̂_i ∈ x̃̂^c
  end if
end for
```

each piece of the Kalman filter is run in parallel on the co-processor microcontrollers. The filters are distributed evenly across the microcontrollers so that the computational power

Algorithm 10 Get $\tilde{\mathbf{H}}^c$ and $\tilde{\mathbf{R}}^c$

for $i = 0$: rows in γ^c **do**
  $i2 = \gamma_i^c$
  $j2 = 0$
  for $j = 0$: columns in H **do**
    if $\mathbf{\Psi}_{c,j} > 0$ **then**
      $\tilde{\mathbf{H}}^c_{i,j2} = \mathbf{H}_{i2,j}$
      $\tilde{\mathbf{R}}^c_{i,j2} = \mathbf{R}_{i2,j}$
      $j2 = j2 + 1$
    end if
  end for
end for

Algorithm 11 Get $\tilde{\mathbf{P}}^c$

$k = 0$
$l = 0$
$m = 0$
for $i = 0$: columns in H **do**
  for $j = 0$: columns in H **do**
    if $i = j$ **then**
      $\tilde{\mathbf{P}}^c_{k,l} = \mathbf{P}_{i,j}$
      $l = l + 1$
      $m = 1$
    else
      if $\mathbf{\Psi}_{c,i} > 0$ AND $\mathbf{\Psi}_{c,j} > 0$ **then**
        $\tilde{\mathbf{P}}^c_{k,l} = \mathbf{P}_{i,j}$
        $l = l + 1$
        $m = 1$
      end if
    end if
  end for
  $l = 0$
  $k = k + m$
  $m = 0$
end for

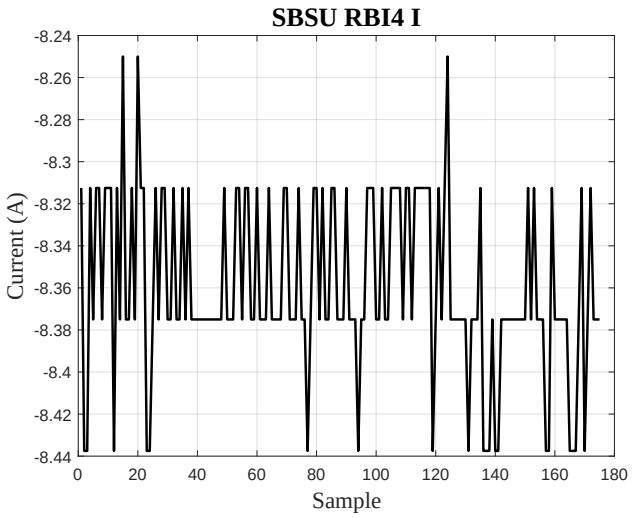


Figure 3.16: Measured current from the AMPS testbed secondary bus switching unit

is used to its full extent. Thus the computational burden of the central flight computer is relieved and state estimation is able to be performed quickly in a low-powered, low-memory environment.

Lastly, γ is used to reorganize each of the outputs y, $\mathbf{P}$, and $\hat{x}$ back into their original form, so that the rest of the FDD tool can be operated without additional changes. The Kalman filter can be processed as a whole or via islands based on the size and computational capacity of the spacecraft.

3.5.6 Dealing With Colored Noise - Hardware Testbed Application

The noise distribution of physical systems is often colored. In the case of some data acquisition methods, data is often manipulated in a manner that causes the probability distribution to become non-white. The hardware implementation used in this work suffers from low sampling resolution as seen in Fig. 3.16. This colored noise distribution conflicts with the assumptions make by the Kalman filter and may contribute to inaccurate estimates of the system states. A whitening filter may be applied to the measurements as a remedy to this problem [99] [100].

First, the whitening transform is introduced. Let Z be a multivariate Gaussian random vector with covariance matrix $\mathbf{\Sigma}$ and mean vector μ. The objective of the Whitening filter is to transform Z into another random vector whose covariance is equal to the identity

matrix, $\mathbf{I}$, so that the new signal is a white noise vector.

The first step is to decorrelate the components of Z, which produces a value drawn from a distribution with a diagonal covariance matrix. The second step consists of scaling the different components such that they have unit variance. This is done through an eigendecomposition of the covariance matrix $\mathbf{\Sigma}$. For the application of power system monitoring, this value can be estimated from the measurements. The procedure is outlined as follows.

$$\mu = E\left(Z\right) = 0 \tag{3.21}$$

and

$$\mathbf{\Sigma} = E(ZZ^T) \tag{3.22}$$

Using from the eigendecomposition the covariance can be expressed as

$$\mathbf{\Sigma} = \mathbf{\Phi}\mathbf{\Lambda}\mathbf{\Phi}^{-1} = \mathbf{\Phi}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{\Phi}^{-1} \tag{3.23}$$

where $\mathbf{\Lambda}$ is a diagonal matrix of the eigenvalues of $\mathbf{\Sigma}$, λ_k, and $\mathbf{\Phi}$ is the matrix of corresponding eigenvectors. Note also that the columns of $\mathbf{\Phi}$ are orthonormal such that

$$\mathbf{\Phi}^T = \mathbf{\Phi}^{-1}. \tag{3.24}$$

Next, U is a random vector,

$$U = \mathbf{\Phi}^T Z, \tag{3.25}$$

with decorrelated Gaussian distribution with variances equal to λ_k. Then a random vector W is defined by

$$W = \mathbf{\Lambda}^{-\frac{1}{2}}U = \mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{\Phi}^T Z. \tag{3.26}$$

Note that W has a standard multivariate distribution such that

$$E\left(WW^T\right) = \boldsymbol{I}. \tag{3.27}$$

Now that the process for transforming a colored sequence Z into a white sequence W is

defined, the process may be modified so that it can be run in an online application for the purpose of online fault monitoring.

The process starts by generating a set of n samples such that the matrix of measurements, $\mathbf{Z} = Z_1, Z_2, ..., Z_n$. After n samples have been observed, data whitening takes place on the signal starting at with Z_n. Then the mean and covariance of Z_n are estimated. Next, the signals must be made zero-mean. This can be done by subtracting off the estimated mean at time i, $\hat{\mu}_i$ of each measurement Z_i.

$$\hat{\mu}_i = \frac{1}{n}\sum_{i}^{i-n} Z_i \tag{3.28}$$

$$\tilde{Z}_i = Z_i - \hat{\mu}_i \tag{3.29}$$

Let $\tilde{Z}_i$ represent the zero-mean vector of measurements at sample time i. Then an estimate for the covariance $\hat{\mathbf{\Sigma}}$ is defined by

$$\hat{\mathbf{\Sigma}} = \frac{1}{n-1}\sum_{i}^{i-n}\left(\tilde{Z}_i \tilde{Z}_i^T\right) \tag{3.30}$$

Next the eigenvalues and eigenvectors of $\hat{\mathbf{\Sigma}}$ are estimated. Depending on the implementation, several programming libraries offer methods to estimate these terms. Then the zero-mean estimates $\hat{\tilde{U}}_i$ and $\hat{\tilde{W}}_i$ are calculated by

$$\hat{\tilde{U}}_i = \mathbf{\Phi}\tilde{Z}_i \tag{3.31}$$

$$\hat{\tilde{W}}_i = \mathbf{\Lambda}^{-\frac{1}{2}}\hat{U}. \tag{3.32}$$

Lastly the whitened signal must be re-conditioned for processing by the state estimator. This is done by adding the mean estimate back into the whitened signal so that

$$\hat{W}_i = \hat{\tilde{W}}_i + \hat{\mu}_i. \tag{3.33}$$

where $\hat{\mu}_i$ is a white noise estimate of Z_i.

The the online whitening filter is tested against data captured from an AMPS hardware

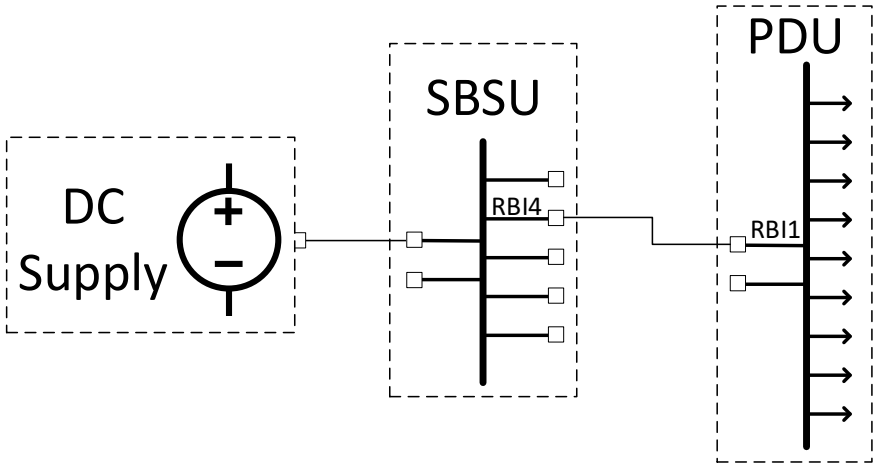


Figure 3.17: Block diagram of the AMPS hardware testbed

testbed at NASA Glenn Research Center. This testbed consists of a DC power supply feeding a secondary bus switching unit (SBSU), which is connected to a power distribution unit (PDU) that powers several constant power loads (CPLs) as shown in Fig. 3.17. The objective is to capture data from the testbed, pre-process it using the online whitening transform, and finally filter it through the dynamic state estimator. This 2-bus power system uses the measurements from SBSU RBI4 and PDU RBI1. Each end of the connection has a bus-side voltage sensor (VIN) a line-side voltage sensor (VOUT) and current sensor (I).

In the first experiment, the DC power supply supported a bank of constant power loads over a period of 175 seconds, with a data acquisition rate of 1Hz. The results of the whitening transform are shown in Fig. 3.18. The measured values from the testbed are shown in black, the whitened signal is shown in blue, and the Kalman filter estimates of the whitened signal are shown in red. The transform is successful in generating white noise signals that emulate the behavior of the measured data. Due to the quasi steady state nature of the data, the estimates generated by the Kalman filter reduce the noise drastically.

As described previously, the behavior of an islanded DC microgrid is not always steady state. Instead, stochastic step changes in the system states occur due to changes in load, generation, faults and other disturbances. To observe the performance of the whitening filter during these random jumps, an additional experiment is conducted where loads are

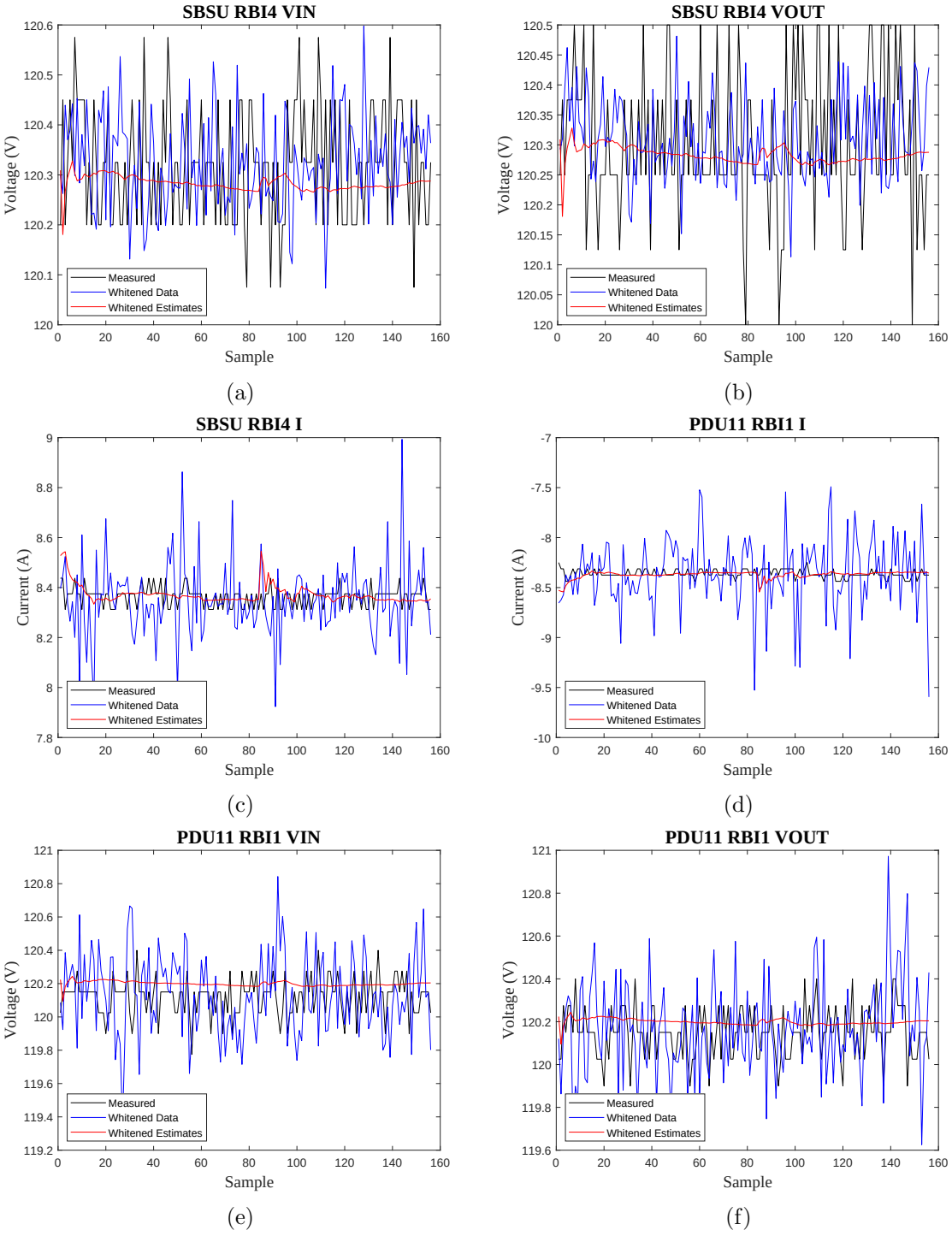


Figure 3.18: Hardware testbed data with whitened transform and Kalman filter estimates

switched on and then off, creating a disturbance in the system state.

Fig. 3.19 shows the behavior of the whitening filter and Kalman filter estimates during a change in load. The pre-processing filter successfully whitens the noise during the steady state periods. During the load transitions, the mean and covariance estimates of the measured data increase. This results in a transient of a few volts in magnitude in the artificially whitened data. The Kalman filter is able to reduce this signal and provide a more accurate estimate. One key factor to improving the performance is to use the dynamic method for determining the process covariance matrix $\mathbf{Q}$ as described previously in this chapter. This 2-bus example shows that the whitening filter can be used to estimate the states and diagnose fault for larger power systems.

3.5.7 Non-linear Application

An ongoing challenge in the field of fault detection and diagnosis, is the ability to identify faults in systems with non-linear dynamics. In the case of the DC electrical power system, power electronic devices such as the DC-DC buck-boost converters and certain loads exhibit this behavior. In order to apply the fault detection and diagnosis techniques in this dissertation to these systems, a non-linear state estimation method must be used.

The Unscented Kalman Filter (UKF) will be applied to show the ability to accurately track the non-linear state behavior of a DC-DC buck-boost converter as shown in Fig. 3.20. The UKF improves performance over its predecessor, the Extended Kalman Filter (EKF), by using a set of points called sigma points to approximate the non-linear function around the mean of the Gaussian signal [101]. If chosen carefully, the sigma points can accurately capture the true mean and covariance of the Gaussian random variable. A survey of DC-DC converter failures done in [102] examines a wide range of fault types thus motivating the need for advanced fault monitoring.

Many models for the high frequency dynamics of this class of power electronics have been studied in the literature. A common technique for modeling this behavior is to use the state-space averaging approach [103]. This method linearizes the dynamics of the device. In this case, the standard Kalman filter may be applied. These dynamics occur on the microsecond scale, which is unfit for the slower communication protocol of the spacecraft

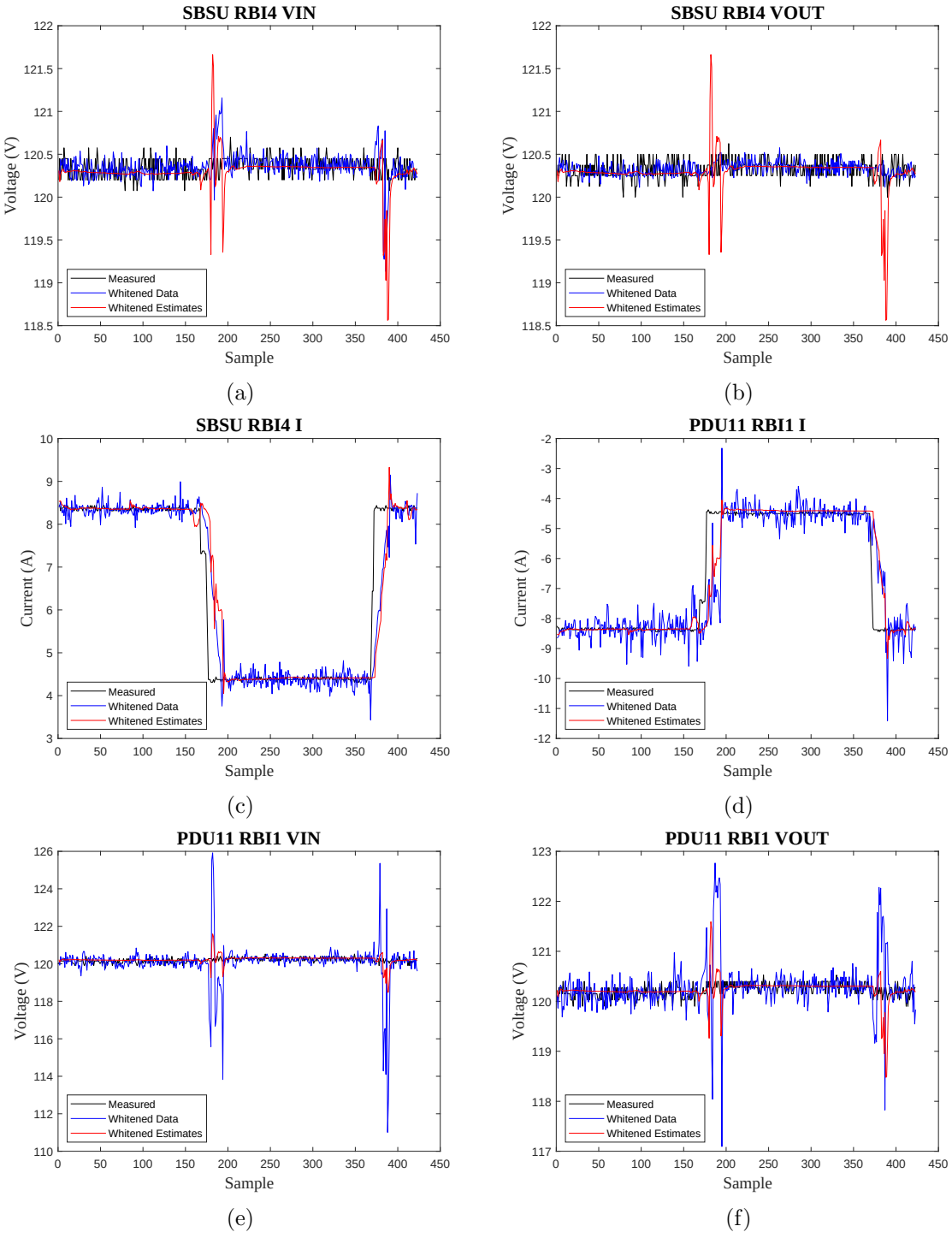


Figure 3.19: Hardware testbed data with whitened transform and Kalman filter estimates during a change in load

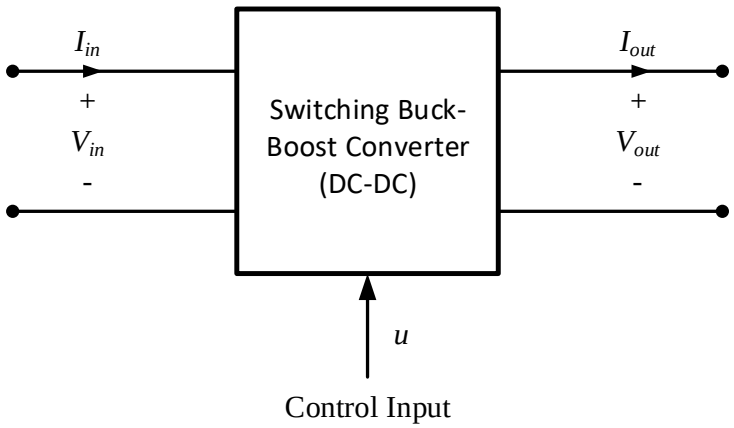


Figure 3.20: Block diagram of a DC-DC buck-boost converter with electrical and control interfaces

power system.

Instead, a UKF can be applied to track the quasi steady state efficiency of the converter based on the converter equation,

$$V_{out}I_{out} = \frac{1}{\epsilon}V_{in}I_{in}, \tag{3.34}$$

where V_{in}, and I_{in} represent the input voltage and current of the converter respectively and V_{out}, and I_{out} is the input voltage and current. In addition, ϵ represents the efficiency of the converter, where $0 \leq \epsilon \leq 1$. In an ideal converter there are no losses and $\epsilon = 1$. However, switching losses and other physical limitations contribute to a small decrease in efficiency. Moreover, a fault in the device can result a change in the efficiency. An unscented Kalman filter can be used to track the unmeasured state, ϵ, to monitor for faults and component degradation.

The following example details the use of the UKF for state estimation. A 120V buck-boost converter nominally regulates the output to 90V, with 16A input and expected efficiency of 0.95. The measurement vector consists of observations $z = [I_{in}, I_{out}, V_{in}, V_{out}]'$.

Set initial parameter state vector $\hat{x}$

$$\hat{x}_0 = \begin{bmatrix} V_{in0} \\ I_{in0} \\ V_{out0} \\ \epsilon_0 \end{bmatrix} = \begin{bmatrix} 120 \\ 16 \\ 90 \\ 0.99 \end{bmatrix}.$$

Next define the nonlinear measurement matrix $\mathbf{H}$ and noise covariance matrix $\mathbf{R}$ as

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \frac{x_1 x_2}{x_3} \end{bmatrix}, \qquad \mathbf{R} = \begin{bmatrix} r_{vin} & 0 & 0 & 0 \\ 0 & r_{iin} & 0 & 0 \\ 0 & 0 & r_{vout} & 0 \\ 0 & 0 & 0 & r_{iout} \end{bmatrix},$$

where $r_{vin} = r_{iin} = r_{vout} = r_{iout} = 0.02$.

Then define process covariance matrix $\mathbf{Q}$, initial error covariance matrix $\mathbf{P}_0$, and state update matrix $\mathbf{\Phi}$ equal to the identity matrix $\mathbf{I}$. With the input parameters have been set, the Unscented Transformation (UT) is implemented.

First, the scaling parameter, λ is set by,

$$\lambda = \alpha^2(L + \kappa) - L, \tag{3.35}$$

where α determines the spread of the sigma points around the mean of x, $\bar{x}$, typically set to a small value around 1e$-$3, κ is another scaling parameter typically set to 0. In this example we set α = 1e$-$3, and $\kappa = 0$. The UT is used to calculate the statistics of the random variables which undergo a nonlinear transformation [104]. The random variable x of dimension L is passed through a non-linear function $y = g(x)$, where x has mean $\bar{x}$ and covariance P_x. The UT calculates the statistics of y by forming a matrix $\boldsymbol{\chi}$ of $2L + 1$ sigma

vectors χ_i with weights W_i.

$$\boldsymbol{\chi}_0 = \bar{x} \tag{3.36}$$

$$\boldsymbol{\chi}_i = \bar{x} + \left(\sqrt{(L+\lambda)P_x}\right)_{i-L}, \qquad i = 1, \ldots, L \tag{3.37}$$

$$\boldsymbol{\chi}_i = \bar{x} + \left(\sqrt{(L+\lambda)P_x}\right)_{i-L}, \qquad i = L, \ldots, 2L \tag{3.38}$$

$$W_0^{(m)} = \lambda/(L+\lambda) \tag{3.39}$$

$$W_0^{(c)} = \lambda/(L+\lambda) + \left(1 - \alpha^2 + \beta\right) \tag{3.40}$$

$$W_i^{(m)} = W_i^{(c)} = \lambda/\left\{2(L=\lambda)\right\}, \qquad i = i, \ldots, 2L \tag{3.41}$$

where β is used to scale prior knowledge of the distribution of the states, x. In our example, we apply a Gaussian distribution so $\beta = 2$ is optimal [101]. Here, $\left(\sqrt{(L+\lambda)P_x}\right)_i$ represents the ith row of the matrix square root. W_i represents the weights of each sigma point.

The UKF is an extension of the UT on the recursive estimation equation, $\hat{x}_k = \hat{x}_{k-1} + \mathbf{K}_k \cdot [y_k - z_k]$. In the recursion equation, the state variable is redefined as the concatenation of the the original state and noise variables $x_k^a = \left[x_k^T v_k^T n_k^T\right]$. The UT is applied to $\hat{x_k^a}$ to calculate the new sigma points $\boldsymbol{\chi}_k^a$. The UKF equations are summarized as follows [101].

$$\text{For } k \in 1, \ldots, \text{inf}, \tag{3.42}$$

Calculate the sigma points:

$$\boldsymbol{\chi}_{k|k-1}^x = \left[\hat{x}_{k-1}^a \hat{x}_{k-1}^a \pm \sqrt{(L+\lambda)|textbf P_{k-1}^a}\right] \tag{3.43}$$

Time update:

$$\boldsymbol{\chi}^x_{k|k-1} = \boldsymbol{\Phi}\left[\boldsymbol{\chi}^x_{k-1}, \boldsymbol{\chi}^v_{k-1}\right] \tag{3.44}$$

$$\hat{x}^-_k = \sum_{i=0}^{2L} W_i^{(m)} \boldsymbol{\chi}^x_{i,k|k-1} \tag{3.45}$$

$$\mathbf{P}^-_k = \sum_{i=0}^{2L} W_i^{(c)} \left[\chi^x_{i,k|k-1} \hat{x}^-_k\right] \left[\chi^x_{i,k|k-1} \hat{x}^-_k\right]^T \tag{3.46}$$

$$\boldsymbol{\Upsilon}_{k|k-1} = \mathbf{H}\left[\chi^x_{k|k-1}, \chi^n_{k-1}\right] \tag{3.47}$$

$$\hat{y}^-_k = \sum_{i=0}^{2L} W_i^{(m)} \Upsilon_{i,k|k-1} \tag{3.48}$$

Measurement update equation:

$$\mathbf{P}_{\tilde{y}_k \tilde{y}_k} = \sum_{i=0}^{2L} W_i^{(c)} \left[\Upsilon_{i,k|k-1} - \hat{y}^-_k\right] \left[\Upsilon_{i,k|k-1} - \hat{y}^-_k\right]^T \tag{3.49}$$

$$\mathbf{P}_{x_k y_k} = \sum_{i=0}^{2L} W_i^{(c)} \left[\chi_{i,k|k-1} \hat{x}^-_k\right] \left[\Upsilon_{i,k|k-1} \hat{y}^-_k\right]^T \tag{3.50}$$

$$\mathbf{K} = \mathbf{P}_{x_k y_k} \mathbf{P}^{-1}_{\tilde{y}_k \tilde{y}_k} \tag{3.51}$$

$$\hat{x}_k = \hat{x}^-_k + \mathbf{K}(y_k - \hat{y}_k) \tag{3.52}$$

$$\mathbf{P}_k = \mathbf{P}^-_k - \mathbf{K}\mathbf{P}_{\tilde{y}_k \tilde{y}_k} \mathbf{K}^T \tag{3.53}$$

where $\boldsymbol{\chi}^a = \left[(\boldsymbol{\chi}^x)^T (\boldsymbol{\chi}^v)^T (\boldsymbol{\chi}^n)^T\right]$.

Processing simulated data through the UKF with the parameters defined above produce the signals in Fig. 3.21. Measured signals are shown in blue, and the state estimates are shown in red. Note that the UKF is able to accurately estimate the non-linear states in the presence of Gaussian noise. In particular, the unmeasured state of efficiency is nominally rated at 95 percent shown by the blue dashed line.

At sample 50, a fault internal to the power converter occurs, which causes a 24 percent decrease in efficiency. This change in state is tracked using the voltage and current measurements. The estimate for the efficiency is about 71 percent after the fault. Then after the fault is removed/repaired, the estimator determines that the converter has returned to its rated operating efficiency.

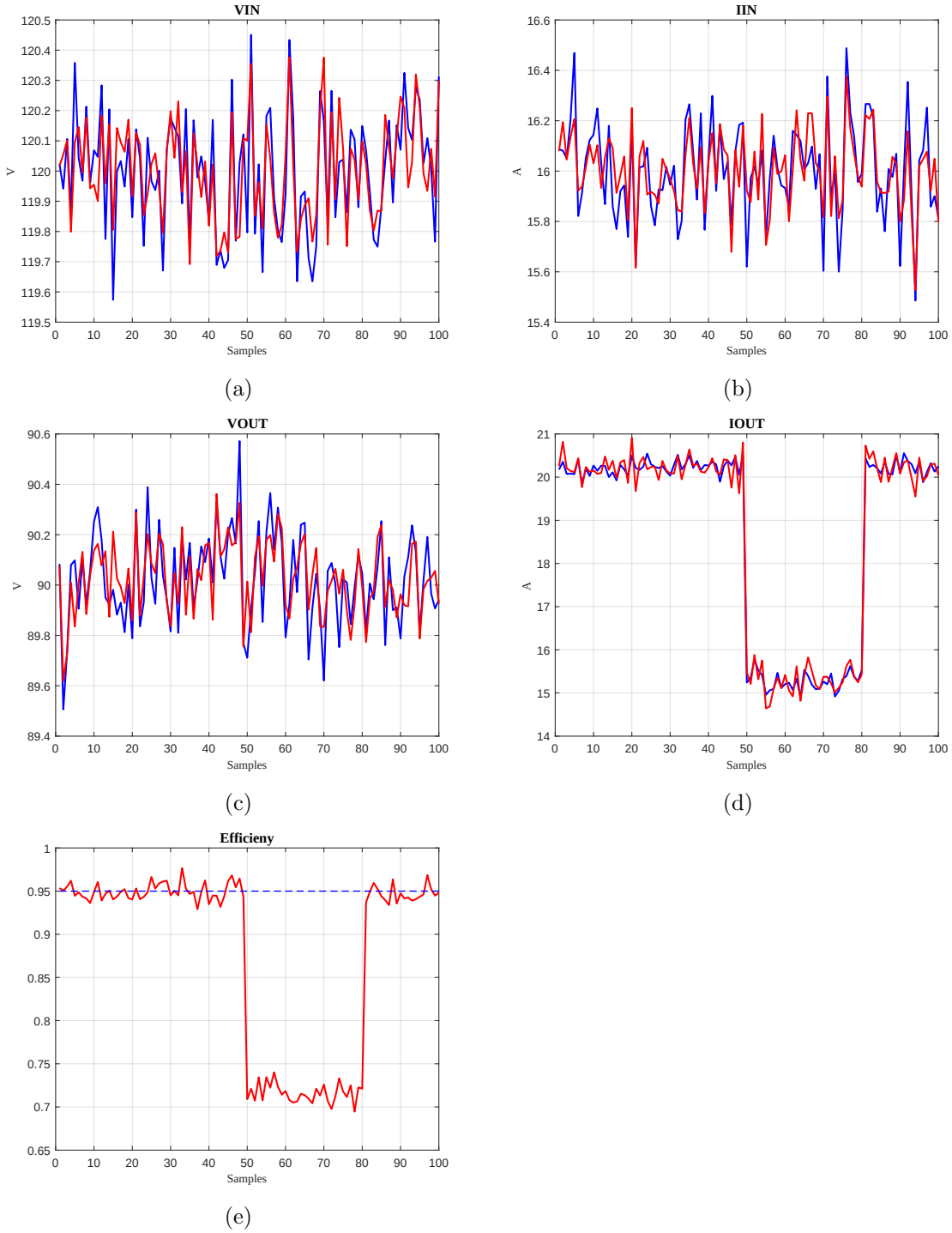


Figure 3.21: Measured signals (red) and UKF estimates (blue) of a simulated buck-boost converter

This tool, monitors the steady state fault status of the power converter by estimating the efficiency. Having an accurate understanding of the converter efficiency is useful in maximizing generator capacity. Because the observations have a non-linear relationship with the states, the unscented Kalman filter was used. This result indicates that similar diagnostic capabilities achieved with the linear Kalman filter could be done with non-linear systems using the UKF. Although the focus of the remainder of this work explores linear systems, a prototype for non-linear dynamic system state estimation is a valuable tool. More research should be conducted in this area of non-linear systems.

3.6 Active Fault Diagnosis

Due to the increased complexity of systems over the past several decades, research in fault detection and diagnosis techniques has grown substantially [105]. Another way of categorizing fault detection and diagnosis methods is based on the interaction between the FDD tool and the physical system. The first group of methods, covered only in this work so far, consists of passive fault diagnosis (PFD). PFD has been well established in the literature, as reviewed in Chapter 2. Fig. 3.22 displays the flow diagram for the general PFD method. In PFD, inputs to the monitored system, such as power electronic set points, are controlled via a feedback controller. The PFD makes its decision based on both the inputs and measured outputs of the system. This consists of alarm processing (AP), a residual generator (RG), and a decision making engine (DM). In this configuration, the PFD has no ability to excite or probe the input controller. Only natural changes in the system or disturbances will effect the controller.

One challenge in designing PFD methods is that modeling uncertainties are unavoidable, especially in practice. Even the most accurately designed models often miss particular aspects of the physical system. In particular, insufficient measurement information from the system and poor signal to noise ratio can make PFD quite challenging. In such situations, faults may go diagnosed for large periods of time. Another problem that may arise is overlapping residual sequences caused by multiple fault types. This case may result in no fault isolation, due to multiple models passing the statistical tests.

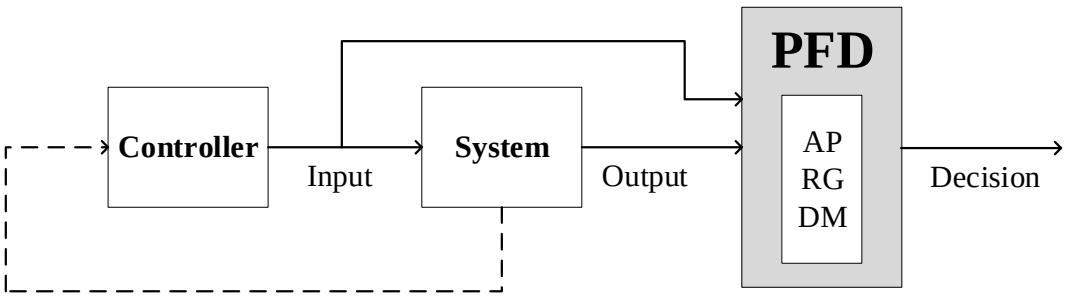


Figure 3.22: Flow diagram of Passive Fault Diagnosis (PFD)

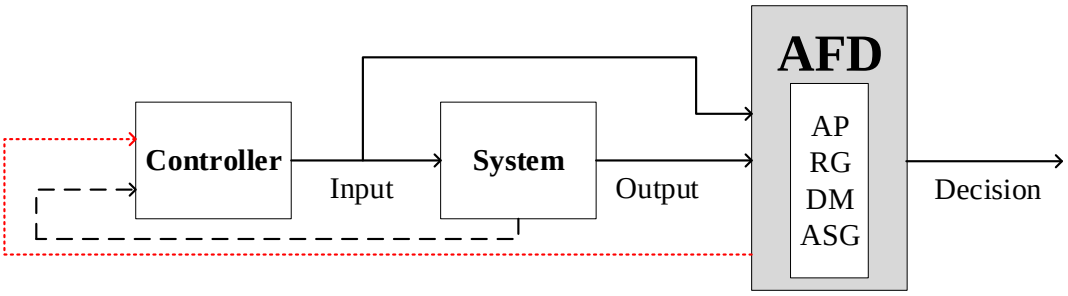


Figure 3.23: Flow diagram of Active Fault Diagnosis (AFD)

One way to address the aforementioned problems is to utilize active fault diagnosis (AFD). AFD uses an auxiliary signal to probe or excite the monitored system, in order to generate more useful information for the diagnosis tool. This idea active signal probing has been used in other fields of engineering such as, optimal experimental design [106], system identification [107], and dual control [108]. More recently, active control for fault diagnosis applications have been researched, such as [109] and [110].

Fig. 3.23 displays a block diagram of the AFD approach. The output of the monitored system (S) is processed by the active fault detector, that also has a residual generator (RG) and decision making engine (DM). The AFD function also has an auxiliary signal generator (ASG) that creates an additional excitation signal into the system. This signal must be carefully chosen, so that it does not negatively impact the performance objectives of the system.

A few important design features must be addressed when developing an AFD method.

First, the length of time the auxiliary input signal is injected can be a fixed finite interval, or variable finite interval, or infinite time interval. For fixed finite interval, the auxiliary input signal is injected for an interval of a predetermined, fixed length. decisions about fault can be provided at any time step of the interval. Variable finite time interval is similar to fixed finite interval, but the length of the time interval is able to be dynamically set. This may be useful to when running a series of statistical tests that may take a varying amount of time to complete. Thirdly, infinite time interval allows the auxiliary signal to be fed continuously into the model until a decision is made. This method is convenient in that it allows the AFD tool time to make a decision, but may lower the performance of the physical control system.

A second consideration is the ability for additional faults to occur within the auxiliary input signal time interval. In this case, the fault status of the monitored system may change during the AFD process, and corresponding model updates must be made. This makes the AFD problem more challenging. If it can be assumed that the fault status of the system remains unchanged during the testing interval, the AFD method only needs to discriminate between the fault-free and individual fault models.

3.6.1 DC Power System Application

The need for active fault diagnosis is made evident in the case of the high impedance line to ground fault. In this situation, a distribution line becomes shorted to ground with a moderate to large resistance. This resistance limits the current from the line to ground, preventing any circuit breaker tripping. For a DC power system, this fault can often be indistinguishable from a sensor offset failure on one of the current sensors on the cable. AFD can be used to differentiate these faults.

First, a directional residual scheme is used to detect the possibility of a high impedance fault. This technique utilizes the direction of the residual vector to indicate a particular fault. Here, the fault detection filter can be considered an observer with a set of projections that map each fault to a specific residual direction [50]. The fault detection an diagnosis tool in this work will use directional residual information to generate fault hypothesis for fault diagnosis.

Let measurements z_a and z_b represent the current observed at each end of a connection. For the case of the high impedance fault, the measurements read different values due to the current leak on the cable. This will cause residuals r_a and r_b to fail the zero mean test. Based on the direction of the residual vector, a fault on the distribution line (l_{ab}) is detected. Next, a dedicated fault model for the high impedance fault is generated. To do this, a fault gain matrix $\mathbf{D}$ and residual vector, $r = z - \hat{z}$, are used to create the fault filter such that

$$x_{k+1} = \mathbf{\Phi} x_k + \mathbf{G} u_k + \mathbf{D}(z_k - \hat{z}_k) \tag{3.54}$$

$$z_k = \mathbf{H} x_k \tag{3.55}$$

Using the residual vector in the development of the fault model allows faults of different magnitudes to be diagnosed. Using r, the fault filter can determine the relative size of the fault. This is useful in determining the criticality of the fault and may influence reconfiguration decision making.

After the high impedance fault is injected, both the high impedance fault model, and current sensor offset model pass the residual tests. An auxiliary signal to open RBI2 and turn off discharge from BCDU1-1 is used to isolate the possible fault, and generate additional data for fault diagnosis. An experiment using active fault diagnosis for the case of the high impedance fault is discussed in the next chapter.

3.7 FDD Methodology Overview

Now that each of the levels of the fault detection and diagnosis hierarchy have been reviewed, the framework for the overall strategy for fault detection and diagnosis is discussed. Faults in the system may be detected at any of the three hierarchical levels. The level of detection depends on the amount of information needed to identify the fault, as well as the safety criticality of the fault. For example line-to-ground faults are detected in the reactive layer because they require quick intervention to prevent further damage and can typically be identified without system-wide information. Failed distribution switches are detected at the component level because they rely on local data, however they generally do not have

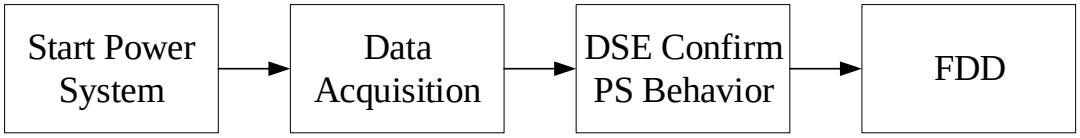


Figure 3.24: Startup procedure for fault monitoring

the same level of time-sensitivity as a ground fault. Sensor faults are detected by the central level using the residuals of the dynamic state estimator because they require system-wide information. Sensor faults can be detected at a much slower rate since they unless the sensor reading poses a direct impact the hardware. In such a case, redundant sensors should be used.

The reactive and component layers have measurements only in their immediate area. This can lead them to make false decisions about the fault state of the system, especially in cases where missing or erroneous data has occurred. For this reason, all faults are sent to the central level for final diagnosis. This prevents misdiagnosis caused by the limited knowledge of the rules based method used at the reactive and component levels. It is worth noting that automatic protection occurs at the time of fault detection in order to protect the system as soon as possible.

Upon system startup, the central layer of the APC begins polling for data from the distributed component layer controllers. Next, the FDD tool begins by verifying the non-faulted behavior of the power system based on the first window of data. This is done to ensure that the model and hardware are in sync. If the power system data is valid and the DSE produces residuals that pass the three tests (whiteness, zero mean, uncorrelated), then the FDD tool moves on to monitor for faults as shown in Fig. 3.24. In this stage, the reactive layer and component layers use the pre-programmed rules to detect faults. At the central level, the DSE monitors the power system to ensure the physical system matches the expected values of the mathematical model.

Fig. 3.25 shows the fault monitoring process when no faults are currently detected. Using the switch state telemetry and any previously identified faults, the power system model is generated each time step. After every iteration, the statistical tests are performed

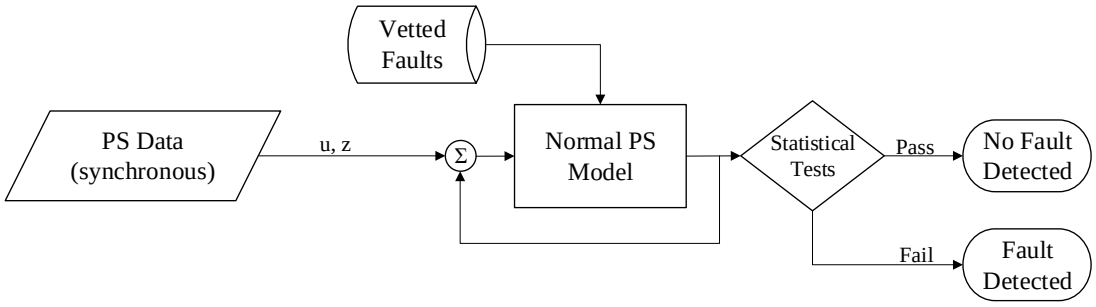


Figure 3.25: Fault monitoring during normal condition

on the residuals and checked for any fault indications by the limits described in the previous chapter.

In the event that a disturbance occurs such that the residuals fail one or more of the statistical tests, a fault hypothesis is generated. Based on the way the tests have failed, different fault indications may be present. For example, if the test of mean fails, a sensor bias fault hypothesis is generated for the sensor with the greatest absolute mean residual. Or if the autocorrelation function becomes zero, a stuck sensor fault hypothesis is generated. After the fault detection has taken place, a fault filter for each fault hypothesis is created automatically and produces a set of residuals to test as seen in Fig. 3.26. If one of the models is able to pass the statistical tests, the fault(s) assigned to that model are considered diagnosed and are added to the list of vetted faults.

Due to the large number of power system components and failure types, it becomes cumbersome to deploy a filter for every possible fault every time a disturbance is detected. In addition, the residuals may not provide enough information to determine which fault or faults to test. Therefore, a set of rules-based fault detection methods are used at the reactive and component layers to aid the DSE in selecting the correct fault models.

During the monitoring process, faults are detected in the reactive and component layers using expert system based rules. Then each fault is sent via an asynchronous message to the central layer, where it is used for hypothesis testing. During each synchronous call to receive power system data, the central layer checks to see if it has received any fault indications from the lower level controllers. In this scenario, the FDD tool checks to see if the fault has been previously detected. If not, then the fault message is added to the list

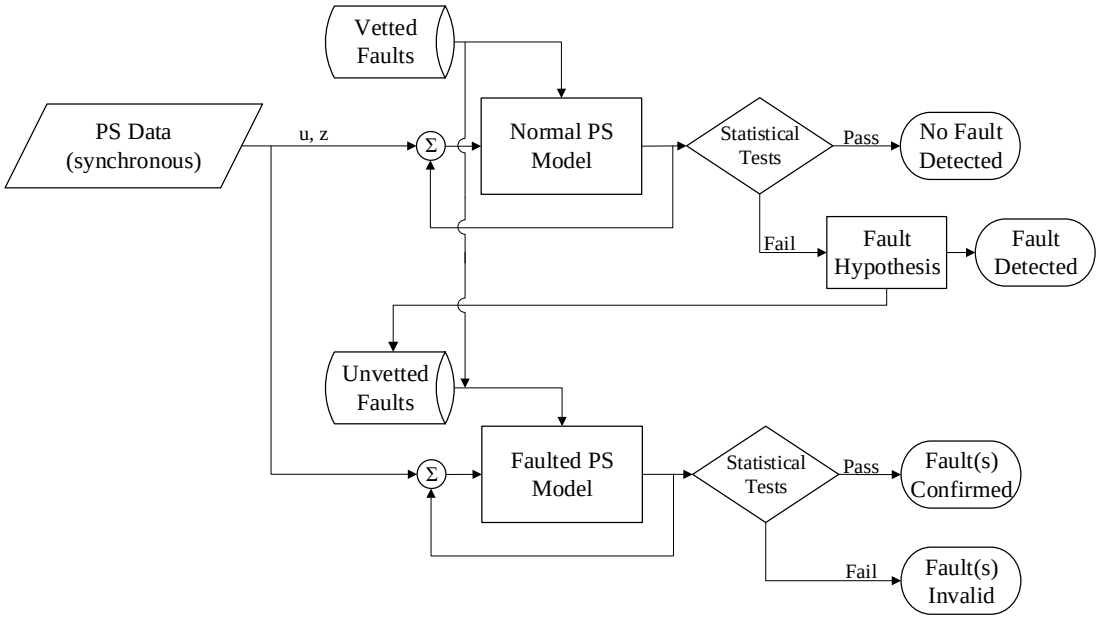


Figure 3.26: Fault diagnosis following DSE disturbance detection

of unvetted faults. Conversely, if the data indicates a component has returned to normal behavior (or repaired by the operator), a fault removed message is added to the unvetted fault list in a similar manner. This method of fault diagnosis is captured in Fig. 3.27.

For the application of spacecraft, the ability to repair a fault or malfunction in the power system may not be possible or may take a long time depending on the availability of the crew. Thus an important feature of the FDD tool is the ability to continue monitoring for faults in the system after one or more faults have already been identified. In order to do this, the DSE must be able to model any combination of faults that can exist in the power system. This is done using the mapping described previously in this chapter.

A key challenge in the diagnosis of multiple sequential faults (including the removal of faults) is to determine which fault models to analyze. Examples in the literature such as [51], suggest maintaining a bank of filters for each fault to handle diagnosis. However, in order to provide quick detection and diagnosis for large systems, the number of fault filters necessary would be too great. One simple example, a sensor fault, the zero mean test fails on a particular sensor. In such event, only one additional model needs to be generated to confirm the failed sensor. Parametric faults such as an actuator failure becomes slightly

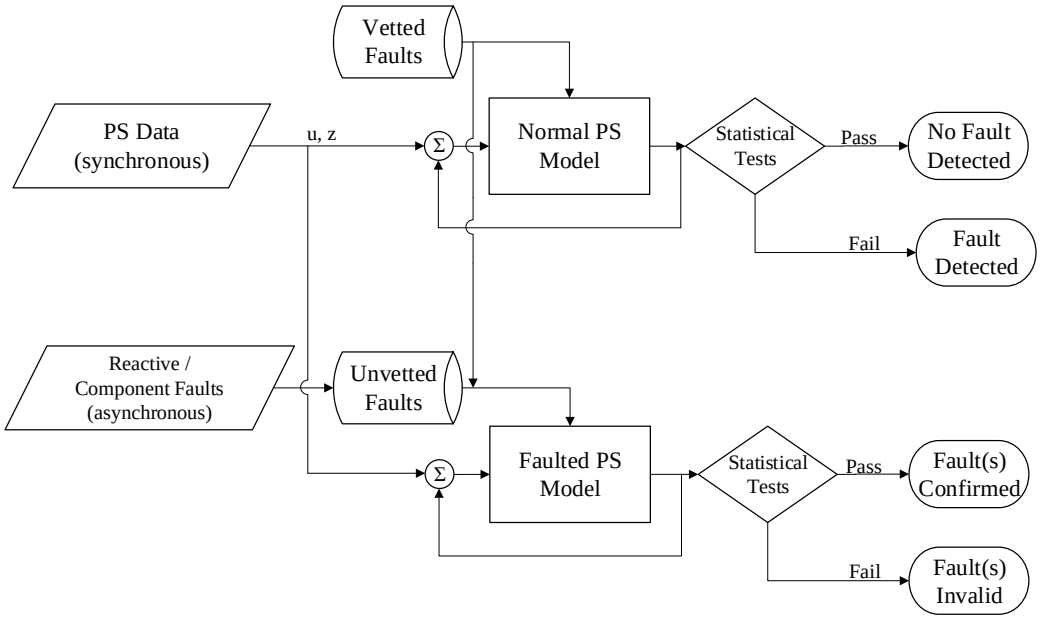


Figure 3.27: Fault diagnosis following reactive and component layer fault detection

more difficult to identify, since the rules based method can easily be triggered by a sensor failure. Therefore two models are generated, one for the actuator fault that was detected by the rule, and another for the sensor with the largest mean residual. The situation can become further complicated when the rules-based methods cause faulty alarms based on erroneous data.

In a system with limited repair capability, multiple faults may enter and leave the system at random times. Thus, the need for more fault models becomes apparent. Depending on the computational capabilities of the system, the number of fault models may vary. This problem is solved by developing one model for each fault in the list of unvetted faults, and another which contains all of the unvetted faults. This is done to check most of the likely fault possibilities, without testing an exponentially large number of fault combinations. These models are deployed in parallel to ease the computational burden of the filters. Fault detection and diagnosis methods with greater processing power, such as terrestrial microgrids, may consider testing a greater number of fault models.

Fig. 3.28 shows the architecture of the fault detection and diagnosis from all three levels of the control hierarchy. First, data is acquired from the sensors by the field programmable

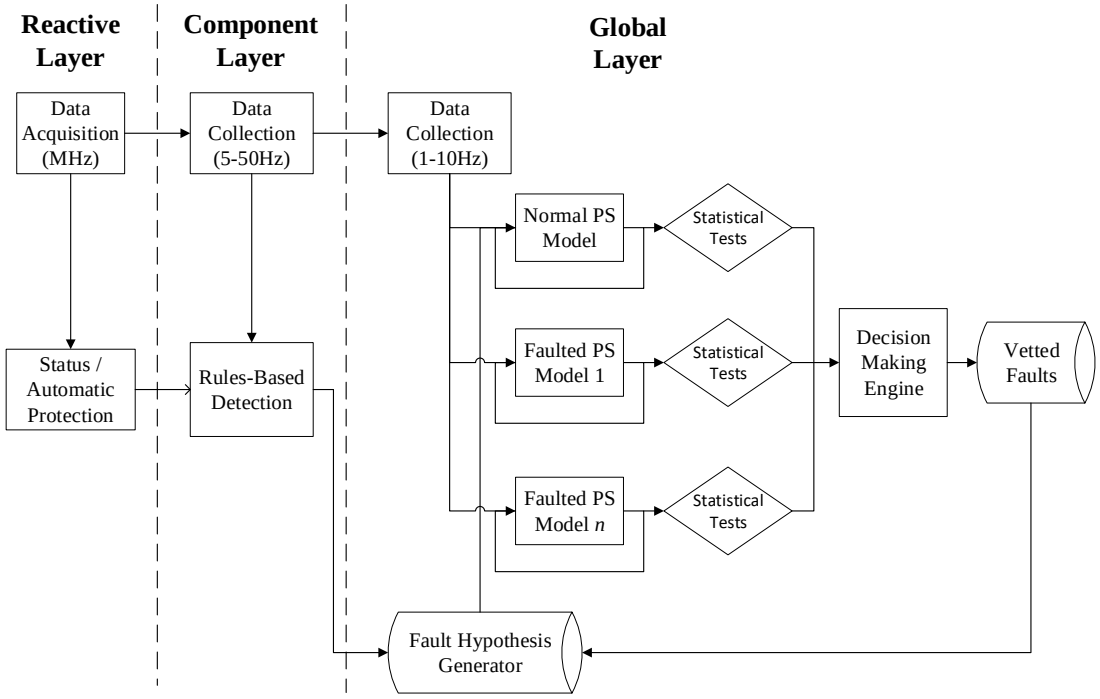


Figure 3.28: Flow diagram of the hierarchical Fault Detection and Diagnosis Method

gate arrays (FPGAs) at a MHz frequency. Using this high frequency data, automatic protection and alarm monitoring are performed. Next, data is passed to the component level controllers at a medium frequency (5-50Hz). Here, rules based methods are developed using simple physics of the power system. These rules help detect failed switches and communication faults at within a particular power system unit. Lastly, the global layer received the system-wide data at the slowest rate (1-10Hz). This data is used to build and run the dynamic state estimators (DSEs) for the normal and fault models. To track the faults detected by each of the three levels, a map is held at the global layer, labeled Fault Hypothesis. This map collects any fault messages from the component and reactive layer, as well as faults detected by the DSE. As mentioned earlier, the DSE residuals are passed through a set of statistical tests, which may indicate the presence of a fault.

Whenever a fault enters the fault hypothesis map, additional DSE models are generated and ran in parallel with the normal model. Once a sufficient number of residuals are generated, the statistical tests are performed. When one or more models pass the statistical tests, they are sent to an expert system who publishes the fault to a queue of vetted

faults. Once published, any of the other APC services can access the fault information. If designed properly, only one fault model should pass he statistical tests at any time. However, modeling and telemetry limitations sometimes result in one or more fault models passing the tests due to indistinguishable data. The decision making engine can then be used to determine based on the expertise of the system designer to publish the most likely fault. This may be based on a rule as simple as accepting the fault with minimum sum of squared errors (Equation 3.56) on the mean of the residuals. A more robust method in the decision making engine is to utilize the concept of active fault diagnosis. This technique would probe the system for additional data by enabling some physical action allowed by the system controller. If the proper auxiliary signal is probed, the indistinguishable fault models may be separated and accurately diagnosed.

$$e = \frac{1}{N} \sum_{i=1}^{N} \eta_i^2 \tag{3.56}$$

The fault detection and diagnoses method provided here offer a combination of model-based and rule-based techniques to create a system that is capable of diagnosing faults both quickly and accurately. The computational complexity is attempted to be reduced for the application of space flight software by using physics based rules of the power system to reduce the number of Kalman filters needed to diagnose the fault. The system is capable of diagnosing a wide range of fault types including sensors, actuators, parametric faults, and communications. The structure allows for additional fault types to be added in as needed.

In the following chapter, this structure for fault detection and diagnosis will be used for the Markov-Chain Monte-Carlo (MCMC) performance testing. There, several single and multiple fault scenarios are tested and analyzed. The success and failure criteria for the fault detection and diagnosis tool are discussed, as well as a performance analysis of the system.

Chapter 4

Results and Discussion

To characterize the strengths and weaknesses of the fault detection and diagnosis tool several experiments are performed. A simulation environment containing over 380 unique faults is developed. A test suit capable of injecting and removing the faults is used to capture the results of the FDD tool. The first section of results examines the response of the FDD software to each fault type. In these experiments the controller begins in an unfaulted state, than transitions into a fault, and lastly removes the fault. This provides a detailed look into the interaction between the rules-based and model-based methods and shows impact to the dynamic state estimator for each fault type. All of the single fault examples are tested using abrupt insertion. The following set of experiments examine the case of incipient faults and intermittent faults. Then a series of tests are performed to study the case of sequential faults. This test characterizes the ability of the FDD tool to diagnose faults when there are already one or more existing faults in the system. Lastly a high impedance ground fault will be used to test the active fault diagnosis portion of the tool.

4.1 Power System Simulation

A MatLab/Simulink simulation of a conceptual spacecraft developed by PC Krause and Associates is used to test the fault detection and diagnostic system. The electrical power system architecture is shown in Fig. 4.1. The power system is made up of two main power distribution networks. These networks remain isolated under normal operating conditions,

however, cross ties may be used to connect the loads to opposite channels.

The simulation is generated based on a proprietary library of power system component models. Each of the models captures the detailed behavior of the interconnected components. The design in this case differs was not used in the development of the mathematical model that was generated for the fault detection and diagnosis system. Treating these models as separate entities helps improve the usefulness of the simulation testing because it treats the power system behavior with some unknowns, similar to that of a hardware testbed.

A C++ interface connects the simulation to the power system controller to serve three functions. First, telemetry from the simulation is converted into a structure to represent the power system and sent to the component level controllers at a rate of 5Hz. Second, the interface receives commands from the APC to set operating points and switches in the power system. Third, a fault generator service can also send commands to the interface to inject faults into the simulation. The component level controllers send data to the central controller at a 1Hz rate.

4.2 Single Faults

For the case of single faults, it is necessary that the FDD tool provide total or near total coverage of the fault types identified by the system designers. For this set of experiments the following fault types in Table 4.1 are considered. In the following subsections, the fault detection and diagnosis for each fault type is examined.

4.2.1 Sensor Bias Voltage Output

When a sensor becomes biased, the output of the sensor reading is offset by a constant value. To test the diagnostic capability of this fault, a sensor bias of +6V is applied to BCDU1-1 VOUT. The FDD tool will have a period of time to diagnose the fault before the bias is removed. Then the FDD tool must validate that the fault has left the system, and normal operation has resumed. Fig. 4.2(a) displays the measurement of the BCDU1-1 output voltage in blue and its estimate in red. The fault is inserted at sample 10, when

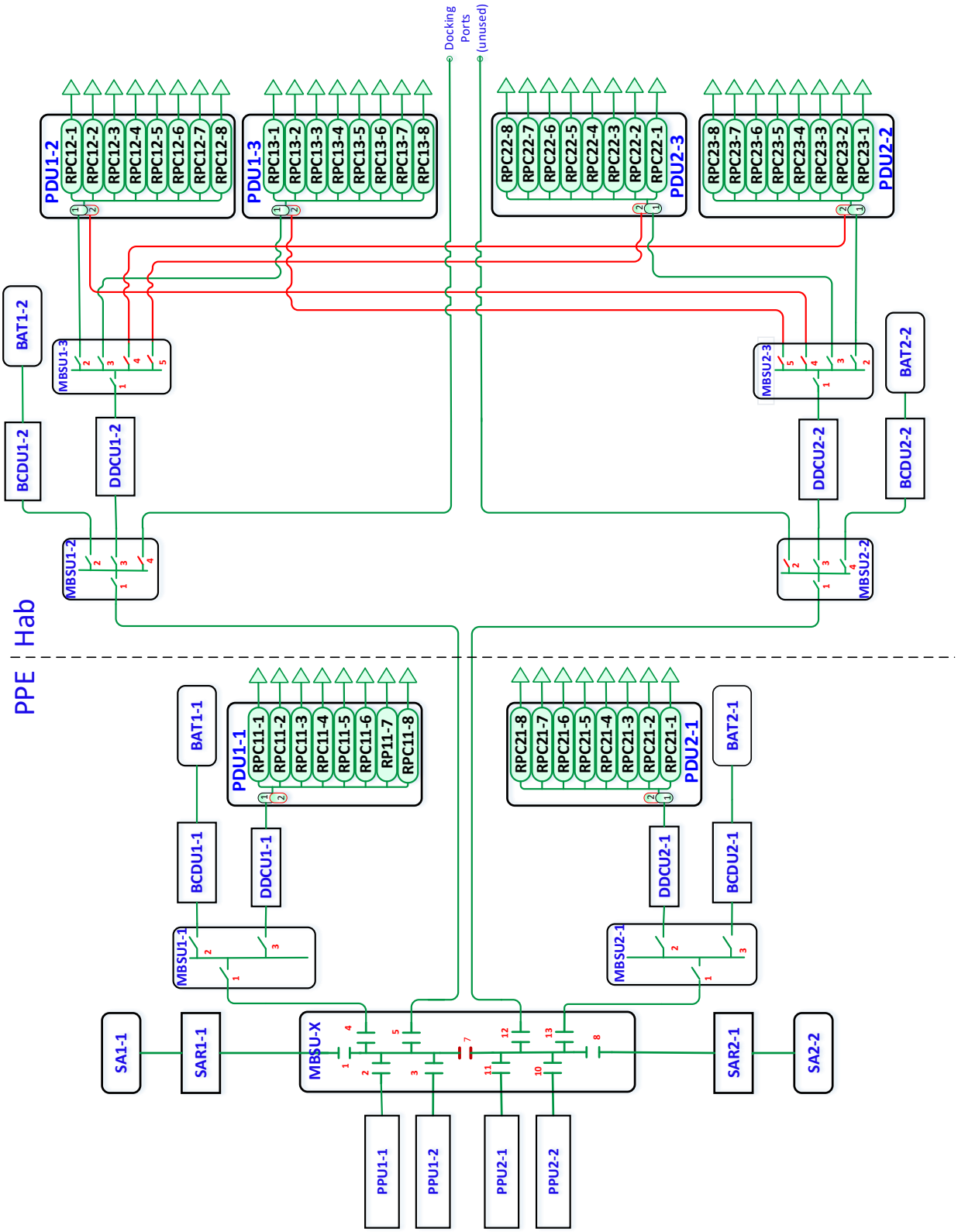


Figure 4.1: One-line diagram of a conceptual Deep Space Vehicle EPS

Table 4.1: Fault Types

Id	Fault Type	Description
1.	Sensor Bias	Measurement output is offset from its true value by a constant
2.	Switch Stuck Open	Actuator on a distribution switch is set open but reads closed
3.	Switch Stuck Closed	Actuator on a distribution switch is set closed but reads open
4.	Short Circuit	Over current detected on circuit breaker, tripping has occurred
5.	Stale Data	Measurements on a component have stopped updating
6.	Excessive Sensor Noise	Measurement noise has exceeded the acceptable range
7.	Stuck Sensor	Measurement is fixed at a constant value
8.	High Impedance Short	A line has shorted to ground with a resistance limiting the flow of current to the fault

the measured voltage on the BCDU output is offset from its normal value of about 121V to 127V. The DSE detects the difference in the measured output and its estimate. Note that the estimate remains near the nominal value of 124V during the fault. This implies that the model is accurately representing the system even in the face of erroneous data.

The sensor fault induces a non-zero mean residual from the state estimator. Fig. 4.2(b) shows the residuals of the voltage signal, and Fig. 4.2(c) shows the mean average of the residuals for the most recent five samples. After the test for mean has failed, a fault model for a sensor fault with the greatest residual mean is generated, in this case BCDU1-1 VOUT. Next, the FDD method analyzes the residuals from both the normal model and the sensor fault model. After the specified number of sample points has been reached, the FDD method determines that only the sensor fault model passes all of the statistical tests, the fault is considered valid and is published to the rest of the APC.

Once the fault has been published, the sensor fault model is now redefined as the normal model. In the fault model, the measurement for the BCDU output voltage measurement is overwritten to zero, and its corresponding row in the output model matrix is set to zero. This is shown in Fig. 4.2(a) from sample 20 to 44, where both the measurement and

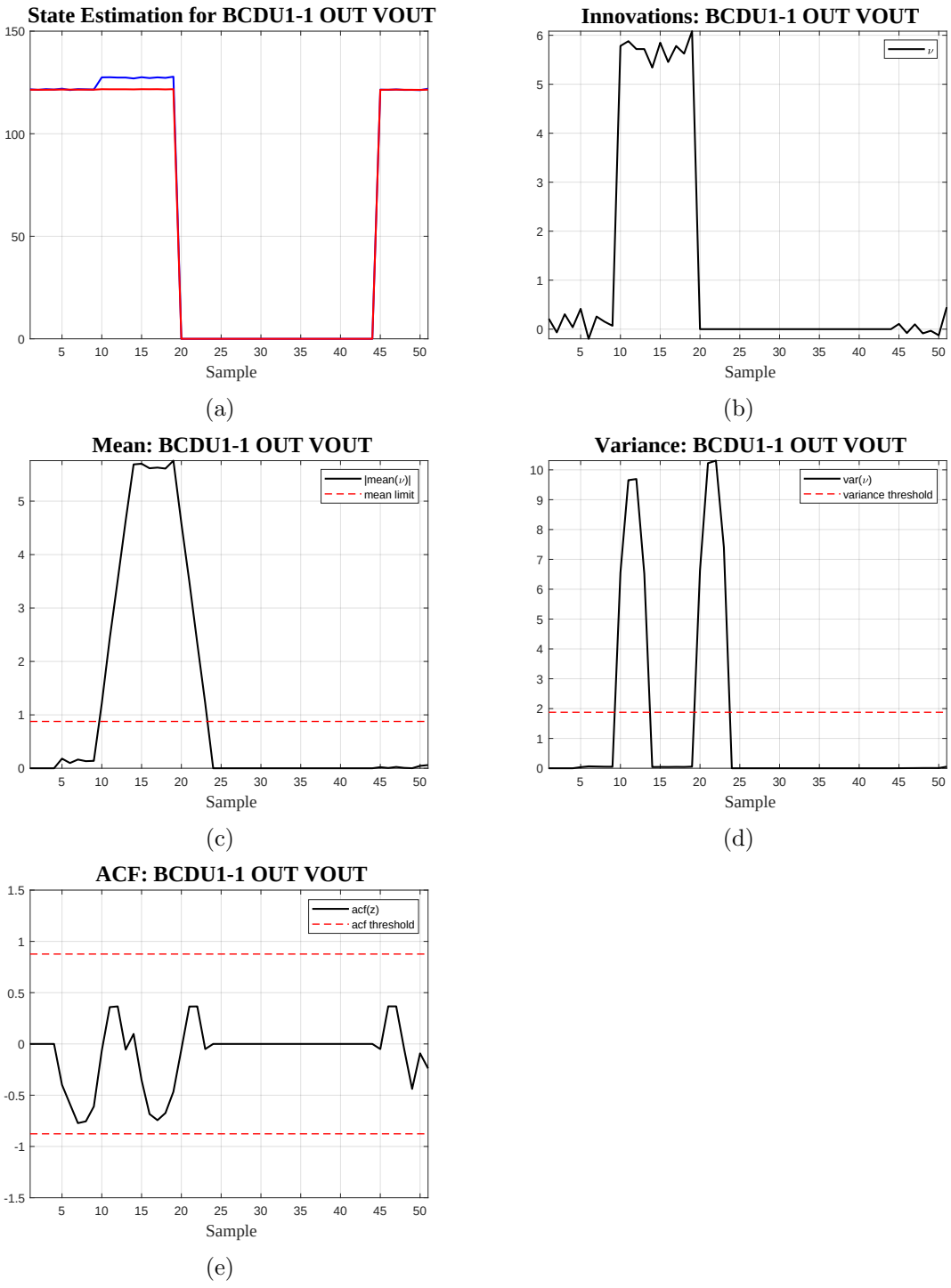


Figure 4.2: DSE response to stuck sensor noise on MBSU1-1 RBI2 and statistical tests on innovations

estimate equal zero. The purpose of this is to remove the erroneous data from the filter, while maintaining a constant model size. The constant model size is useful in mapping the rows and columns of the model to the sensors within the power system. In the event of additional faults, the FDD tool is able to locate the proper elements of the state space model to manipulate the output matrix as needed.

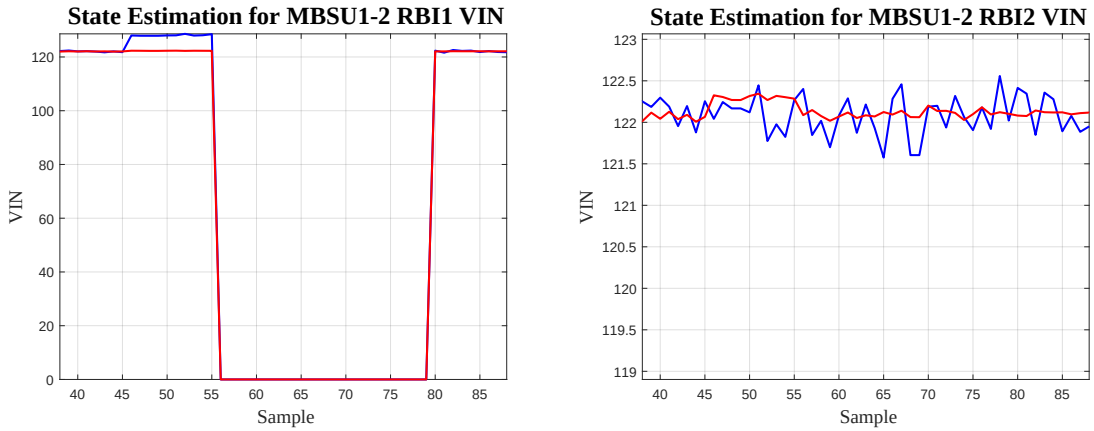
As the new normal model is processed, the FDD tool tracks that a sensor fault has been diagnosed in the system. To detect whether or not the sensor has returned to normal status a sequential model is implemented to run in series with the full model as discussed in the previous chapter. If the sequential filter passes the statistical tests for the failed sensor, it can be brought back into the full model, and the fault can be cleared from the system. The validated fault clearance occurs at sample 45, and the DSE continues monitoring using the full set of sensors.

Fig. 4.2(d) displays the variance of the residual signal. During the changes in operation at sample 10 and 20, the variance spikes above the allowable threshold until until the transition period has passed through the residual window. This does not result in a sensor variance diagnosis because the criteria for a noise fault must exceed the limit later in this chapter.

Lastly, Fig. 4.2(e) shows the results of the autocorrelation function during the sensor fault. Because the noise characteristics remain white during the fault, the ACF stays within the acceptable range. When the sensor is removed from the model, the ACF flattens to zero. from sample 25 to 44. This accounts for the zero value which has replaces the actual sensor value.

4.2.2 Sensor Bias Voltage Input

Next a test is run on an input voltage sensor on MBSU1-2 RBI1, where the input voltage is offset by +6V. Fig. 4.3(a) displays the measurements and estimates for the input voltage signal. Notice that the filter responds in a similar manner to the signals in Fig. **??**. The FDD tool uses the same techniques described above, regardless of the sensor type or location. Therefore similar diagnostic results are produced . The measurement and estimate of the input voltage on switch MBSU1-2 RBI2 is displayed in Fig. 4.3(b). Here we see that the



(a) (b)

Figure 4.3: DSE response to sensor bias voltage input

measurements and estimates remain nearly equal, however, the biased data does effect estimate of the voltage input before in the period when the sensor has been diagnosed and removed from the model. Samples 46 through 56 show the estimate increase slightly due to the occurrence of bad data on the state variable for the bus voltage. After the fault has been removed from the system, the residual of the unaffected measurement returns to its zero mean value.

4.2.3 Sensor Bias Current Output

Next, a test for an output current sensor bias on MBSU2-1 RBI2 is performed. This process is very similar for input current sensors as well. In this test the current measurement is offset by +6A. Then the FDD tool attempts to diagnose the fault. Lastly the test platform removes the fault so the normal behavior of the sensor can be validated. Fig. 4.4(a) shows the current measurement and estimate on MBSU2-1 RBI2. After the bias is applied at sample 192, the measurement increases by 6A, while the estimate increases by almost 3A. This is due to the nature of the state space model. Because the state vector consists of only bus voltages, there is less weighting put on the current measurements in the system. This decision was made to reduce the total number of states in the model, while still accurately capturing the power system behavior. Still, the zero mean test fails for this residual, and the fault model is generated, and validated. Nuances in the design of the model is a critical

factor to consider when analyzing the residuals

Fig. 4.4(b) shows the measurements and estimates for the input voltage of MBSU2-1 RBI2. Note that the estimate for the bus voltage is not biased to the same magnitude as in Fig. 4.3(b). This is because the current measurement has a lesser effect on the state than the voltage measurement. In addition the number of redundant voltage sensors on MBSU2-1 greatly outweighs the number of current measurements, thus weighting the estimate of the bus voltage closer to its true value.

Fig. 4.4(c) displays the current measurement and estimate of the input current of DDCU2-1. As seen in the spacecraft power system schematic, this is the current through the switch connected to MBSU2-1 RBI2. Note that the measurement does not deviate from its normal behavior, but the estimate is effected by the biased current on MBSU2-1 RBI2. Due to the nodal equations applied in the model, The residual mean on the current measurement of MBSU2-1 RBI2 is greater than DDCU2-1, so the sensor bias is correctly identified. Lastly, Fig. 4.4(d) shows the input voltage measurement. Note that the estimate droops slightly while the fault is being diagnosed. This is due to the lack of additional valid measurements on this state variable. In this case there are three measurements that relate to the state variable, DDCU2-1 I_{IN}, DDCU2-1 V_{IN}, and MBSU2-1 RBI2 I_{OUT}, the last of which contains bad data. Since the number of sensors relating to the state is fewer, the weighting from the bad data on MBSU2-1 RBI2 I is more pronounced.

4.2.4 Incipient Sensor Bias Voltage Input

The previous three sensor bias cases have been examples of abrupt faults. Next we examine the case of an incipient sensor bias. In the situation, the sensor bias will increase slowly over time. These faults are challenging to detect, as the impact to the measurement will initially be small. It is important to detect such faults quickly, so that impacts to other processes such as voltage regulation control will be minimized. Fig. 4.5(a) shows the results of input voltage sensor on MBSU1-2 RBI2 during an incipient sensor fault. Measurements are displayed in blue and the estimate is in red. The fault is inserted at sample 50, the fault is diagnosed and removed from the system at sample 86. The fault is removed at 96, where the FDD service recognizes that the sensor has returned to normal functionality and adds

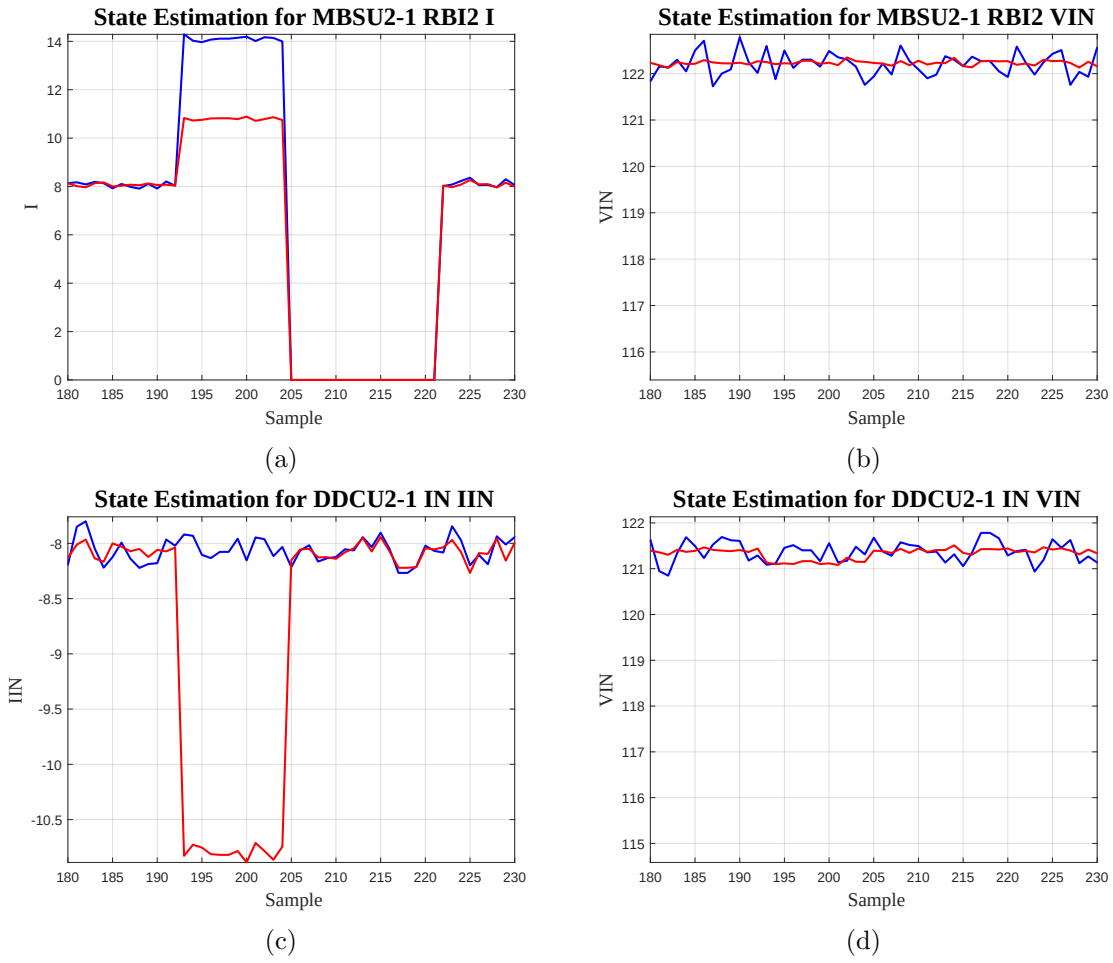
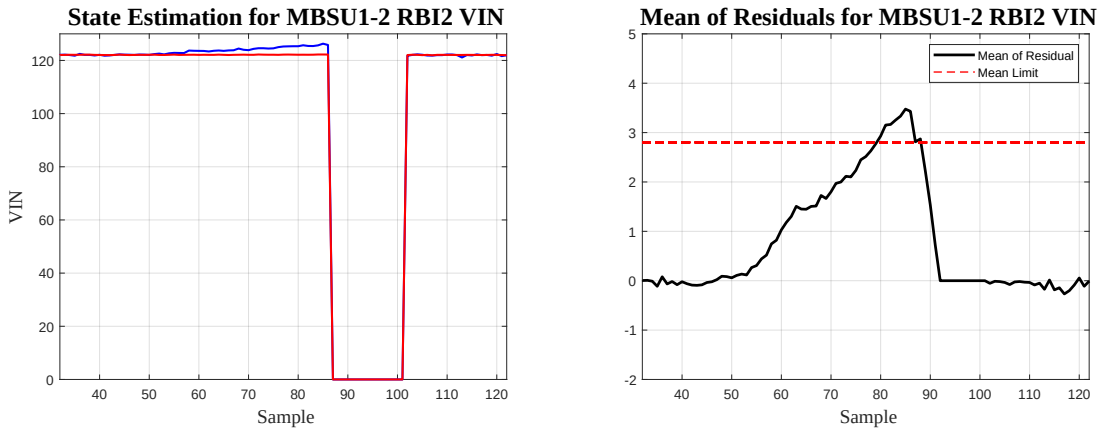


Figure 4.4: DSE response to sensor bias current



(a) (b)

Figure 4.5: DSE response to incipient sensor bias voltage input

the measurement back into the model at sample 101.

For this example, the rate of increase of the sensor bias is set to 0.1 Volts per second. The Kalman filter is able to diagnose the sensor fault after 36 seconds (measurements samples at 1Hz). The threshold for the test of mean for each residual is defined as $1.96/\sqrt{N}$. In order to limit the false detections due to disturbances and the low number of measurements in the residual window (5), the mean threshold was increased to 2.8. Fig. 4.5(b) displays the residual of the signal in black against the threshold for failure of the test of zero mean in red. The fault is detected at sample 80 when mean of the residual crosses the threshold. A fault model for the sensor fault is generated and processes measurements until it succe all of the residual tests. At sample 86 the fault model residuals pass the tests and the fault is diagnosed.

The two dominating factors that determine the time to detection for incipient faults, including are the rate of increase of the fault, the threshold for the test of zero mean, the signal to noise ratio. The time to detection for incipient faults with a constant rate of increase can be estimated as

$$\text{time to detection} \approx \frac{\text{mean limit}}{\text{fault slope}} \tag{4.1}$$

for an incipient fault with zero initial bias. In this example the estimated time to detection is 28 seconds, which is confirmed by Fig. 4.5(b).

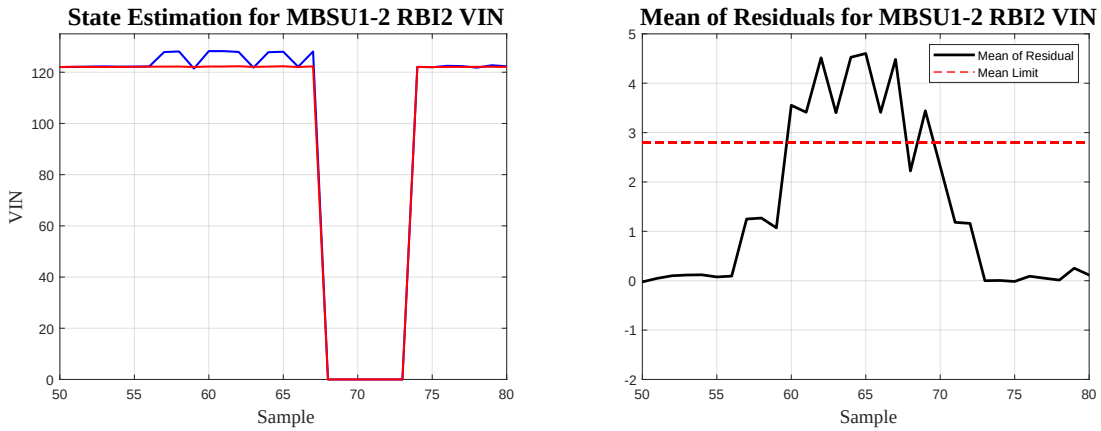
Operator judgments should be used to determine the optimal setting for the zero mean limit. The factors that play a role in setting this parameter include the signal to noise ratio, the number of measurements in the residual window, the frequency of disturbances in the system. Most importantly, the operator must determine the optimal ratio of false detections to slow/missed detections for small and incipient faults. This ratio will vary depending on the application and the parameters listed above.

4.2.5 Intermittent Sensor Bias

The other type of time dependency a fault may occur takes place in the form of an intermittent fault. In this situation, parameter in which the fault has occurred deviates from its normal condition in stochastic time intervals. These faults are more challenging to diagnose because the fault does not follow a steady state behavior. For this FDD scheme there are two time-based classes of intermittent faults. Fist are slow intermittent faults. In this case, the rate at which the signal is switching from faulted to normal is greater than the number of residual samples N. In this case the FDD tool would diagnose the faulty and normal condition as they appear. Additional logic could be added to the central layer to track how often a fault has been entered and thus determine the fault state based on based on maximum likelihood. The second case consists of intermittent faults occurring smaller than the size of the residual window.

This example will analyze the effect of an intermittent sensor bias on MBSU1-2 RBI2 VIN that occurs shorter than the residual window. In this case, a bias of 6V is applied then returns to an unbiased reading in a cycle of 2 seconds on and 1 second off. The measurements and estimates of the signals are shown in Fig. 4.6(a).

Fig. 4.6(b) shows the mean value of the residuals for the faulted signal over a window of 5 samples. At sample 67, the mean of the intermittent fault still falls above the residual threshold and the fault is diagnosed. After the fault is removed, the state estimator recognizes the removal of the fault and returns it to normal status at sample 74. For very small intermittent additive faults, such as the sensor bias, the threshold for detection depends on the ratio of time spent between faulty and normal behavior. This ratio and the magnitude of the fault will determine if a fault is severe enough to be detected. A balance must be



(a) (b)

Figure 4.6: DSE response to intermittent sensor bias voltage input

held between the number of false detections cause by disturbances and missed detections. Other methods for modeling and identifying intermittent faults are discussed in [1].

4.2.6 Switch Stuck Open

The next fault type examined is a switch stuck open. In this fault, a distribution actuator, or switch, fails open while the sensor measurement reads closed. In this example RBI3 on MBSU1-3 fails open at sample 317. This causes the current between MBSU1-3 and PDU1-3 to drop to zero, as seen in Fig. 4.7(c) and Fig. 4.7(d). Note that the state estimator attempts to track the disturbance, however, the model discrepancies in the node voltages (Fig. 4.7(a), Fig. 4.7(b), Fig. 4.7(d) and Fig. 4.7(e)) fail the test of zero mean while the fault is diagnosed in samples 317 through 324.

During the diagnostic process the FDD tool is made aware of a potential switch stuck open fault at MBSU1-3 RBI3 by the rules described in Algorithm 5. The FDD tool then generates a fault model based on the current measurements and sets the corresponding stuck switch to open due to the fault message. This model is then processed over a window of time, and if successful in passing all of the statistical tests. Once the tests have passed the fault is published and the fault is inserted into the FDD's new normal model. The new normal model is shown in Fig. 4.7 during samples 325-327.

At sample 328 the fault is removed at the switch is set to closed once again. Here the

normal model with the embedded switch stuck open fault fails the residual test, since no current should be able to flow between the two nodes. This results in the discrepancy in signals Fig. 4.7(c) and Fig. 4.7(d). The component level controllers detect that the fault has been removed via Algorithm 5, and send a message to the global level FDD tool that the fault has left the system. Upon receiving the message, the FDD tool generates a model with the switch stuck open removed and performs the statistical tests. Once the tests pass (sample 335) the normal model is reset, now having no faults in the map.

4.2.7 Switch Stuck Closed

Another fault type is the event of an actuator failing closed while the telemetry senses that it is open. This can cause power to flow in a distribution branch that is not intended by the EPS controller. For this test, normally opened switch MBSU2-3 RBI5 is set closed at sample 421. This causes current to flow between MBSU2-3 and PDU2-3. Similar to the switch stuck open fault, the component level controllers detect the fault through a first principal based rule defined in Algorithm 5. A fault model capturing the switch failure is generated and run in parallel with the normal model during samples 421 through 429. At sample 430 the fault is published and the normal model is redefined with the switch stuck closed fault. Lastly, at sample 433 the fault is removed and the FDD tool recognizes the change in telemetry and updates its model after validating the fault removal at sample 439, where the residuals return to their zero mean status.

4.2.8 Short Circuit

As discussed earlier, the short circuit fault is caused by a distribution line shorting to ground, resulting in one or more circuit breakers tripping within the power system. The tripping algorithms that reside on the reactive layer controllers can take several forms, and must be carefully set to handle the bidirectional power flow across the power system. This is due to the distributed nature of generation, energy storage, and loads. In order to simplify the complexity caused by the reactive layer automatic protection scheme, short circuit faults are set in the quasi steady state simulation by tripping a single switch.

This test will emulate a short circuit fault by tripping the circuit breaker on MBSU2-2

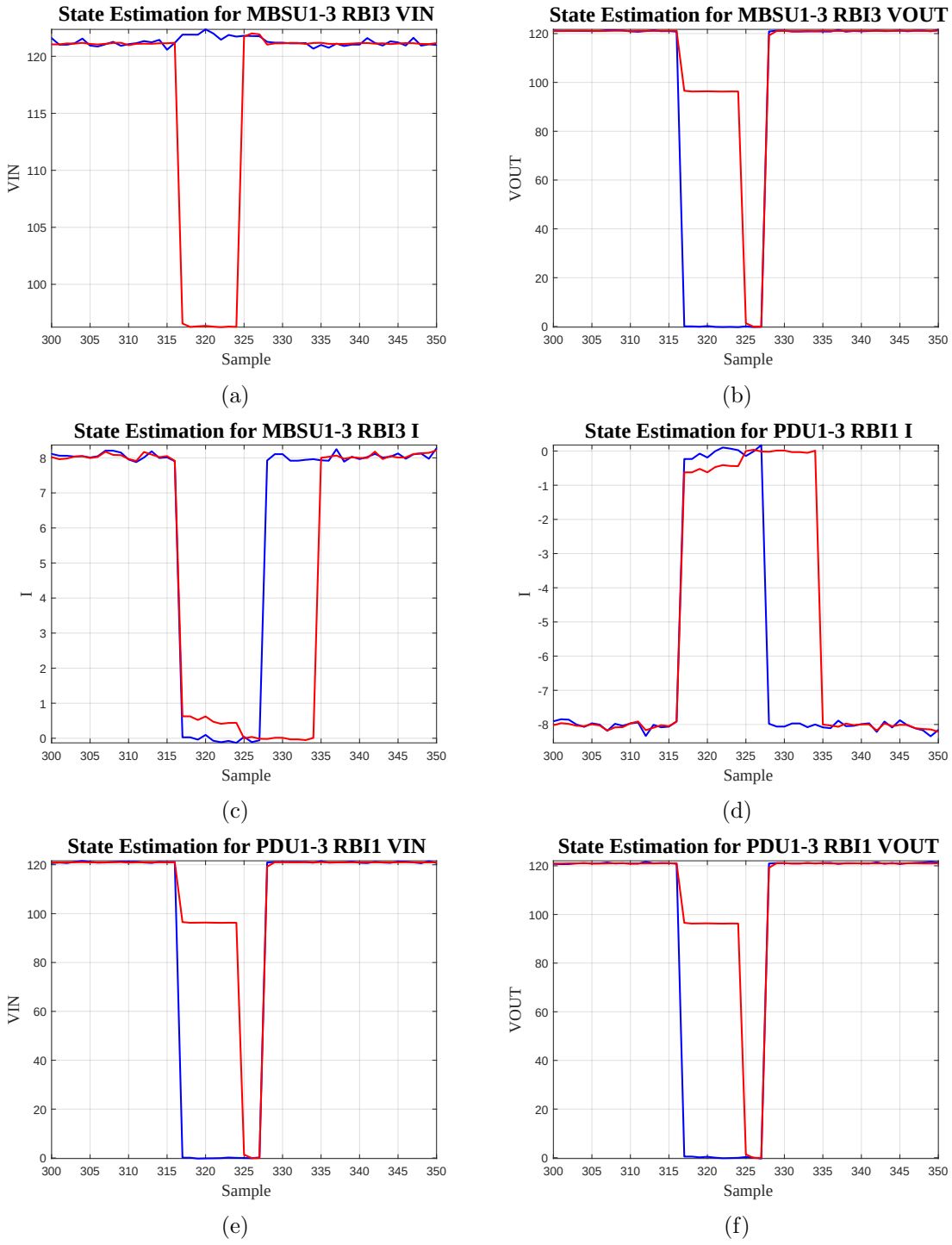


Figure 4.7: DSE response to MBSU1-3 RBI3 switch stuck open

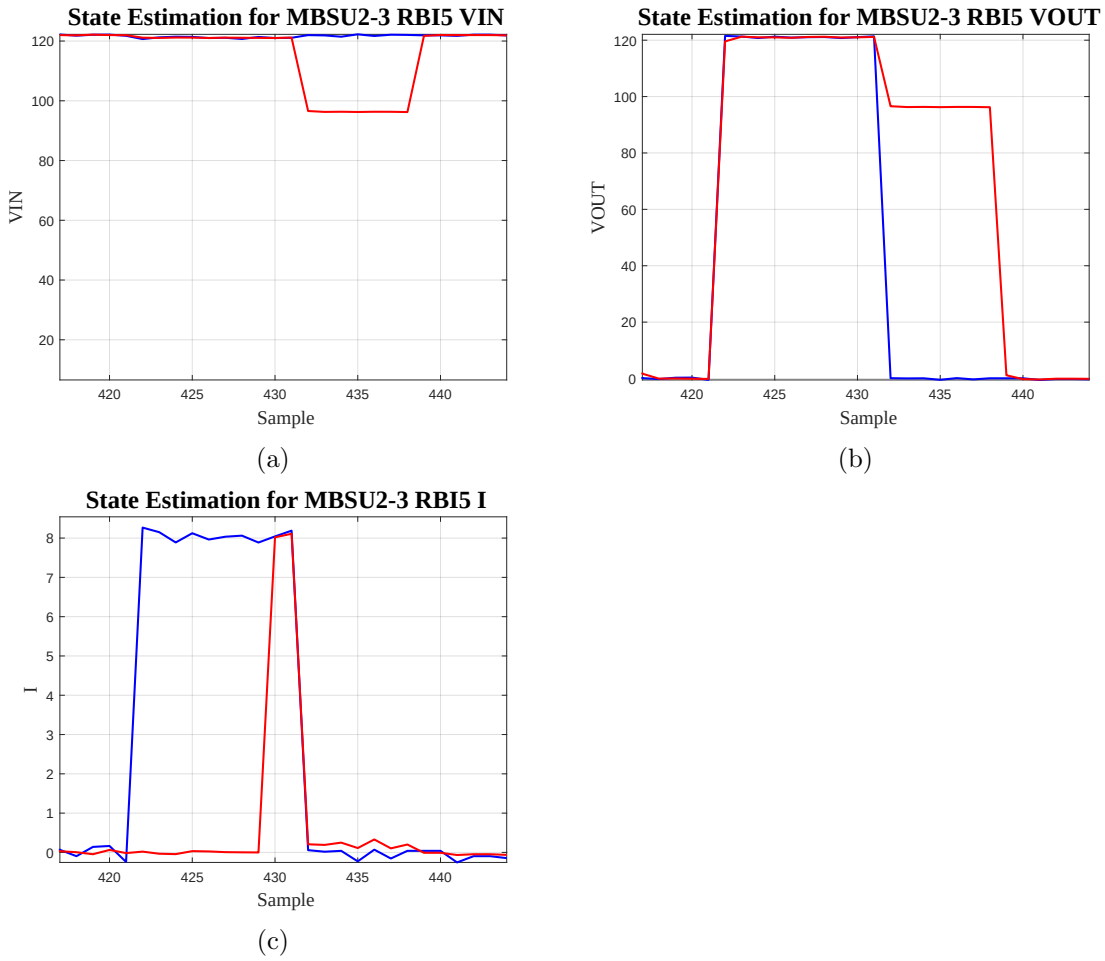


Figure 4.8: DSE response to MBSU2-3 RBI5 switch stuck closed

RBI3. In addition to opening the circuit breaker, the simulation also reports a tripping status sent by the reactive layer. This mimics the behavior of the hardware protection detecting the over current. The component level controller on MBSU2-2 receives the trip signal and generates a short circuit fault to be sent the the central controller. Then the central controller then generates a model with the short circuit fault and process the measurement data until the statistical tests pass. A similar structure is performed during the removal of the fault.

Fig. 4.9(a) and Fig. 4.9(b) show the input and output voltages of the tripped switch respectively. Fig. 4.9(c) shows the current through the switch before and after the fault. Note that the sample time (1Hz) of the global controller is too small to observe any current transient or voltage droop before the hardware protection activates the breaker. Another feature of this simulated fault is that the capacitance on the loads causes a slow drop in voltage as displayed in Fig. 4.9(b) samples 357 through 377. While this slow decay in voltage may not accurately represent the behavior of the actual hardware, the FDD tool is able to track the states because the abide by the physics defined in the the DSE model.

4.2.9 Stale Data

A communication failure is one category of fault that may occur in any process that uses feedback control. These faults are dependent on the physical structure as well as the software implementation of the communication system. Based on the nature of the communication fault types, several detection and diagnosis methods may be used.

For this experiment, a stale data fault is defined as the loss of communication from the reactive layer to the component layer. Based on the communication protocol of the APC, a component level controller sends its most recent data packet to the central layer upon request of the central controller. If the component controller has not received updated data from the reactive layer, it will send its last set of data cached in memory. On each iteration of data aquisition, the central layer controller checks the range of timestamps for stale data. If aone or more data packets exceeds the maximum timestamp lag, a stale data fault is generated.

Several other communication fault types are possible in systems such as a DC power

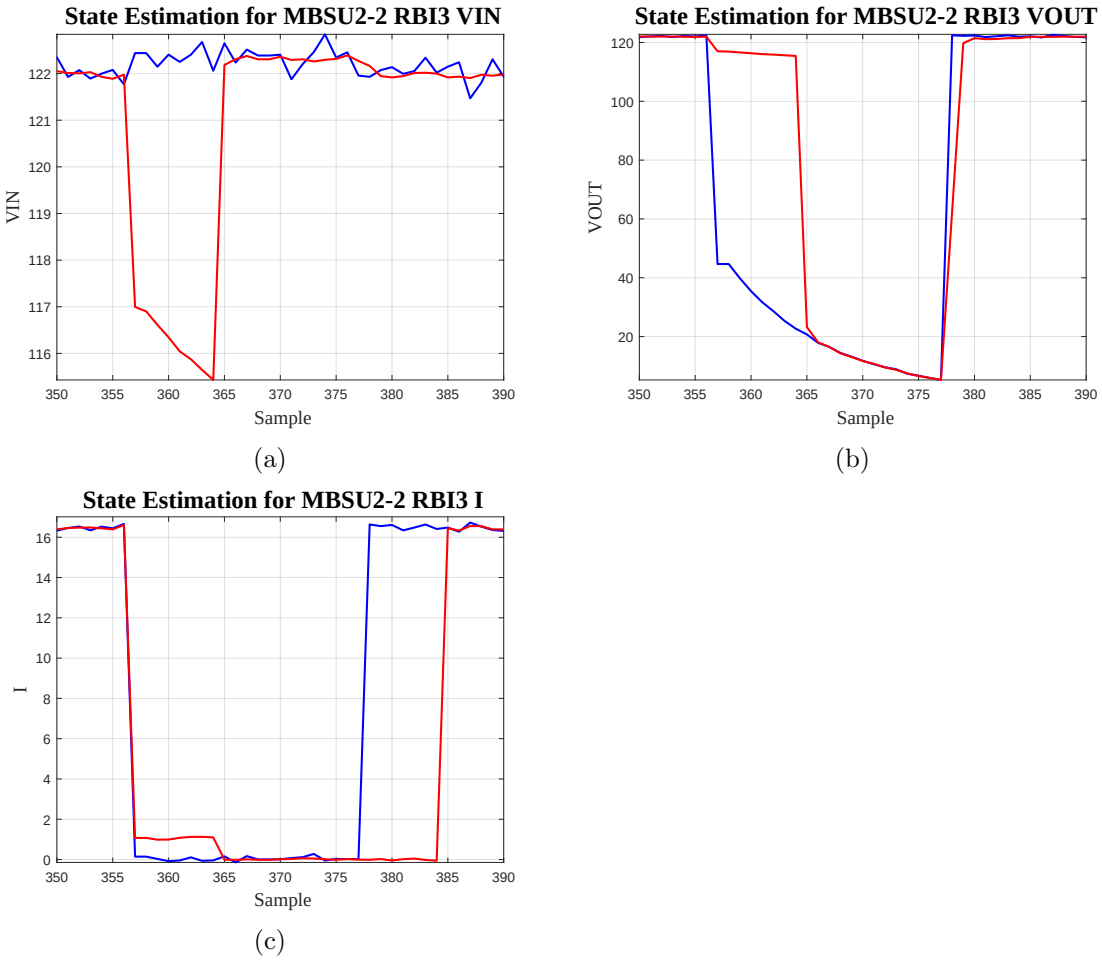


Figure 4.9: DSE response to a short circuit at MBSU2-2 RBI3

system, and the identification methods for each type should be addressed by the FDD tool. The diagnostic process proposed here, is similar for all communication failures.

Fig. 4.10 shows measurements and data before during and after a stale data communication fault. Fig. 4.10(a), Fig. 4.10(b), and Fig. 4.10(c) display data from the component with the communication failure, PDU1-2. The fault begins at sample 258, where the central controller does not receive updated measurements for each time step. Once the data packet tolerance is passed, the central layer generates a message for the PDU1-1 component level controller, signaling the potential stale data fault. The FDD tool develops a model for this fault by setting all measurements and corresponding rows in the output matrix to zero. Note that the test of mean is not violated in the normal model during the communication fault. To validate the faulty behavior, the test for whiteness must fail for the PDU1-2 measurements. At sample 278 the faulted model is redefined as the new normal and the fault is published. Once the fault is removed and data begins to publish as normal. Then the fault manager is notified of the change from the component level controller, and the non-faulted model passes the statistical tests, thus validating the removal of the communication fault as seen from samples 290 through 310.

A test for observability is critical when dealing with losses in communication. Without sufficient data, the system could become unobservable resulting in the Kalman filter being unable to estimate system states. To do this, the rank of the output matrix is tested for each new fault model that is generated. This ensures the ability to perform dynamic state estimation to monitor for faults.

Fig. 4.10(d) displays a measurement on MBSU1-3 RBI2, which is depended on the voltage state from the comm-faulted PDU1-2. Although there is no voltage directly measured from the PDU bus, estimates are still able to be generated for the sensor because the state is still observable. In addition the missing data points from PDU1-2 are able to be estimated as well. This can be useful for other APC applications that may depend on full system telemetry for monitoring and control.

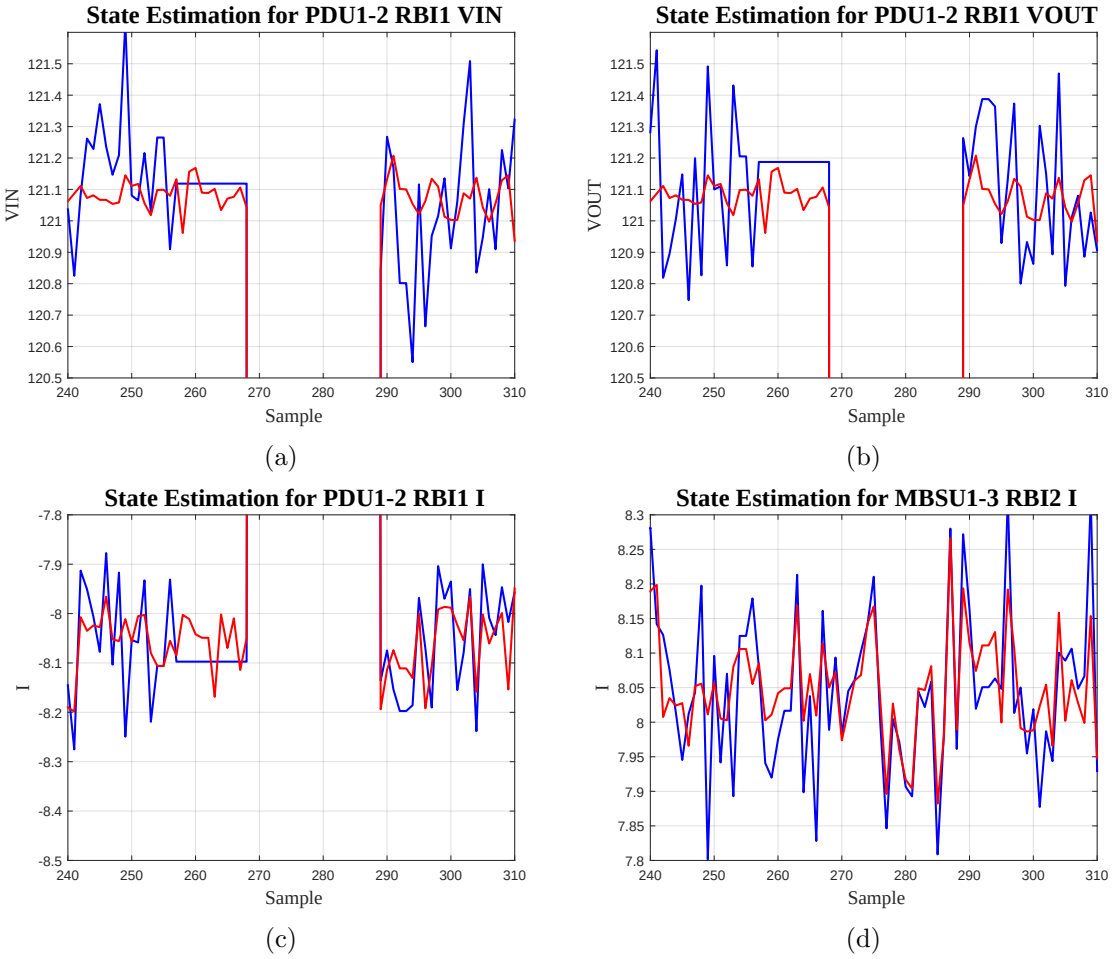


Figure 4.10: DSE response to stale data at PDU1-2

4.2.10 Stuck Sensor

The next fault experiment is a stuck sensor. In this case a stuck sensor refers to the event that a sensor has malfunctioned such that it produces the same measurement reading each time it polls for data. This same behavior can also be caused by a failure in the communication link between the sensor and the local controller. Stuck sensor faults can be particularly challenging to detect via the standard rules based methods because the faulted value may exist within the normal operating range, not triggering an alarm. This can lead to more dangerous faults being undetected due to the masking of data created by the stuck sensor. Thus it is important to quickly identify any faulty or stuck data in the power system.

The example in Fig. 4.11(a) shows the observations and state estimates for MBSU1-1 RBI2 VIN before, during, and after a stuck sensor failure. At sample time 316 the measured value of the voltage sensor becomes stuck at 122.2V. Fig. 4.11(c) shows the results of the test of mean. In this case, the mean remains well below the fault threshold. Fig. 4.11(b) and Fig. 4.11(d) display the decreases the variance of the residuals during the time of the failure. This indicates that there may be a sensor related fault. Lastly, in Fig. 4.11(e) the autocorrelation function returns zero during the stuck senor failure. This occurs when both the variance and covariance of the measurements, z, become zero. Because ACF = Σ (covariances) / (unscaled variance) the function is undefined and returns zero. This test indicates that a stuck sensor has been detected. The fault can then be diagnosed by removing the sensor from the model, and performing the statistical tests. Once the sensor returns to its normal behavior at sample 336, the sequential filter detects the change, the fault is removed, and the measurement is returned to the normal mode model. This methodology also works for a sensor whose noise parameters have become colored due to a malfunction. Further tests of this nature should be performed.

4.2.11 Excessive Sensor Noise

Another fault type that creates challenges in diagnosis is excessive fault noise. This fault occurs when a sensor noise, while still white, has a variance beyond its rated value. The example in Fig. 4.12 shows the behavior of the DSE during an excessive noise failure.

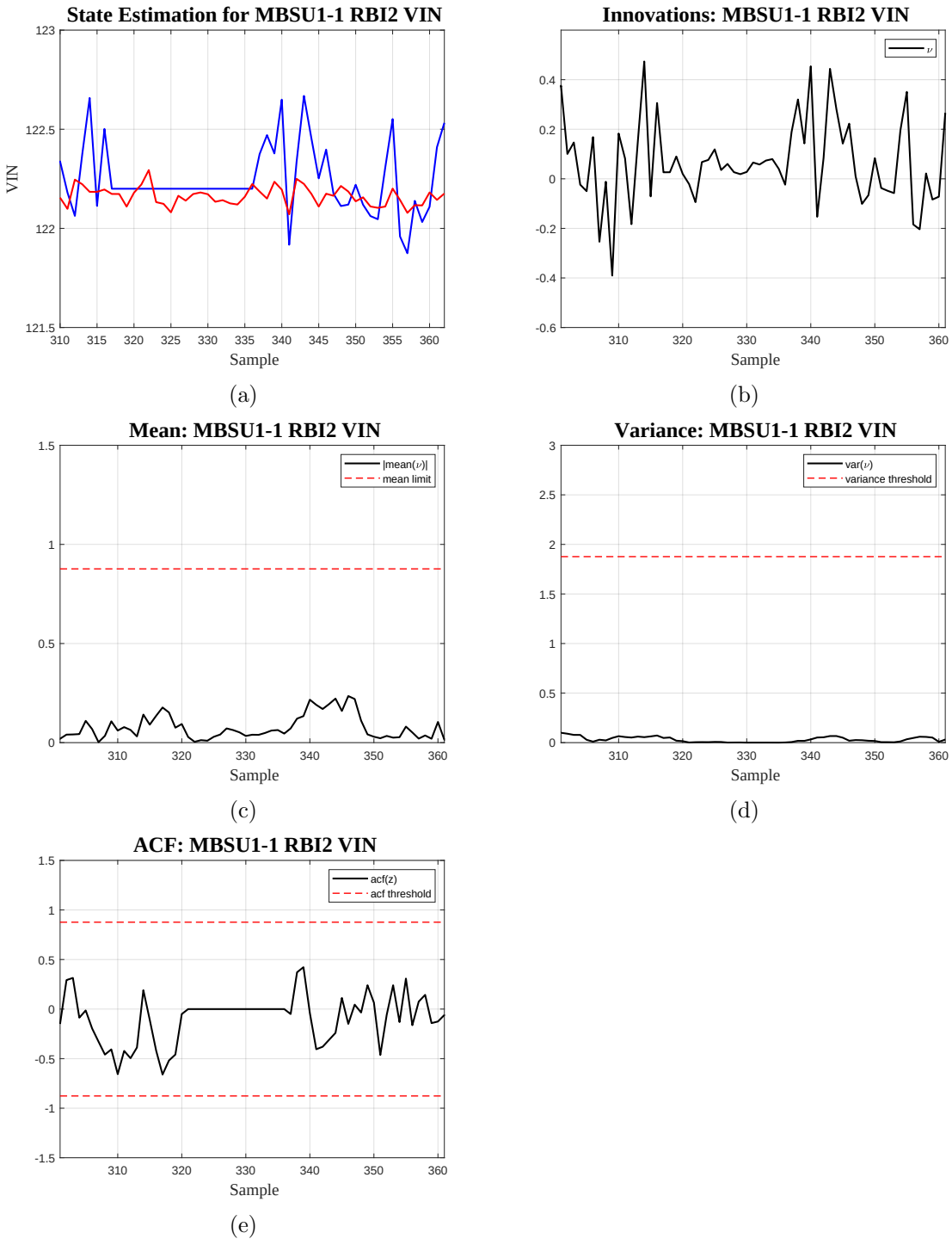


Figure 4.11: DSE response to stuck sensor noise on MBSU1-1 RBI2 and statistical tests on innovations

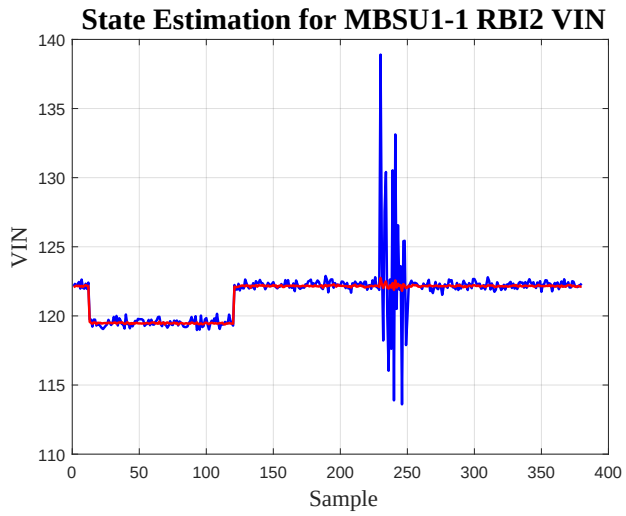


Figure 4.12: DSE response to excessive sensor noise on MBSU1-1 RBI2 VIN without fault diagnosis

Fig. 4.13(a) shows the same fault with the response of fault diagnosis. At sample 47 the sensor becomes excessively noisy. Then, sample 59 the fault is diagnosed based on information from the statistical tests. Then the sensor is removed from the model. As with the other sensor faults, a sequential filter is generated for this sensor, and continues to be monitored.

Fig. 4.13(b) shows the response of the innovations before and after the sensor fault. Fig. 4.13(c) displays the mean average of the innovations for $N = 5$ samples. Excessive sensor noise has no effect on the mean of the signal, however in the case of the small innovation sample size, the mean can exceed the threshold, triggering a fault hypothesis model for a sensor bias.

Fig. 4.13(d) shows the variance of the innovations over time. The variance is calculate as a part of the autocorrelation function and can be used as another statistical test for measurement noise. In this case, the test triggers a sensor noise fault after the variance exceeds the threshold. The variance threshold can be set based on the expected variance of the sensor, or if standardized innovations are used, based on the unit variance behavior. Because the variance test exceeds the threshold by a greater margin than the mean test, the fault is classified as a sensor noise fault. The difference between these failures however is insignificant in the recovery operation of the power system or the fault diagnostic scheme.

The sensor is taken offline and continues to be monitored by the sequential Kalman filter until it is repaired or normal behavior is detected.

Lastly, Fig. 4.13(e) shows the one-lag autocorrelation of the measurements. Note that although the variance in noise changes, the signal remains white. This is reflected by the ACF staying within its normal bounds, indicating that the faulty signal still passes the whiteness test.

4.2.12 High Impedance Line to Ground

High impedance faults occur when a line is shorted to ground through a current limiting resistance. This resistance can be caused due to degraded insulation, or a piece of debris between the line and ground. In this experiment, a high impedance line to ground with fault resistance of 20Ω is set between MBSU1-1 RBI2 and BCDU1-1, as shown in Fig. 4.14. Here, MBSU1-1 feeds current to both the BCDU and the fault. Although this fault does not generate a high enough current to enable the circuit breakers for detection, high impedance shorts often precede a hard fault which can damage equipment due to over current and thermal inrush. In addition, a high impedance fault can consume a significant amount of power from the sources and storage devices, leaving less power available for critical spacecraft loads. For these reasons, it is important to quickly diagnose and isolate the fault.

Fig. 4.15 shows the measured data and estimates from before, during, and after the fault is injected into the simulation. Before the fault has occurred, the solar arrays provide 30A to MBSU1-1 RBI2 to charge the batteries during insolation. The current convention used in the model indicates that the current leaving a bus is positive, and the current entering a bus is negative. When the fault is injected at sample 50, the short to ground adds additional load to the distribution line. In the power system simulation, a fault resistance of 20Ω consumes 6A with a remaining 26.5A charging the battery.

In Fig 4.15(a) and Fig. 4.15(b) the residuals from the current measurements at each bus fail the test of zero mean, indicating a fault. The fault hypothesis generator creates a model for a sensor bias fault at BCDU1-1 I. Using the directional residual analysis, a fault model is also generated for a high impedance line to ground between MBSU1-1 RBI2

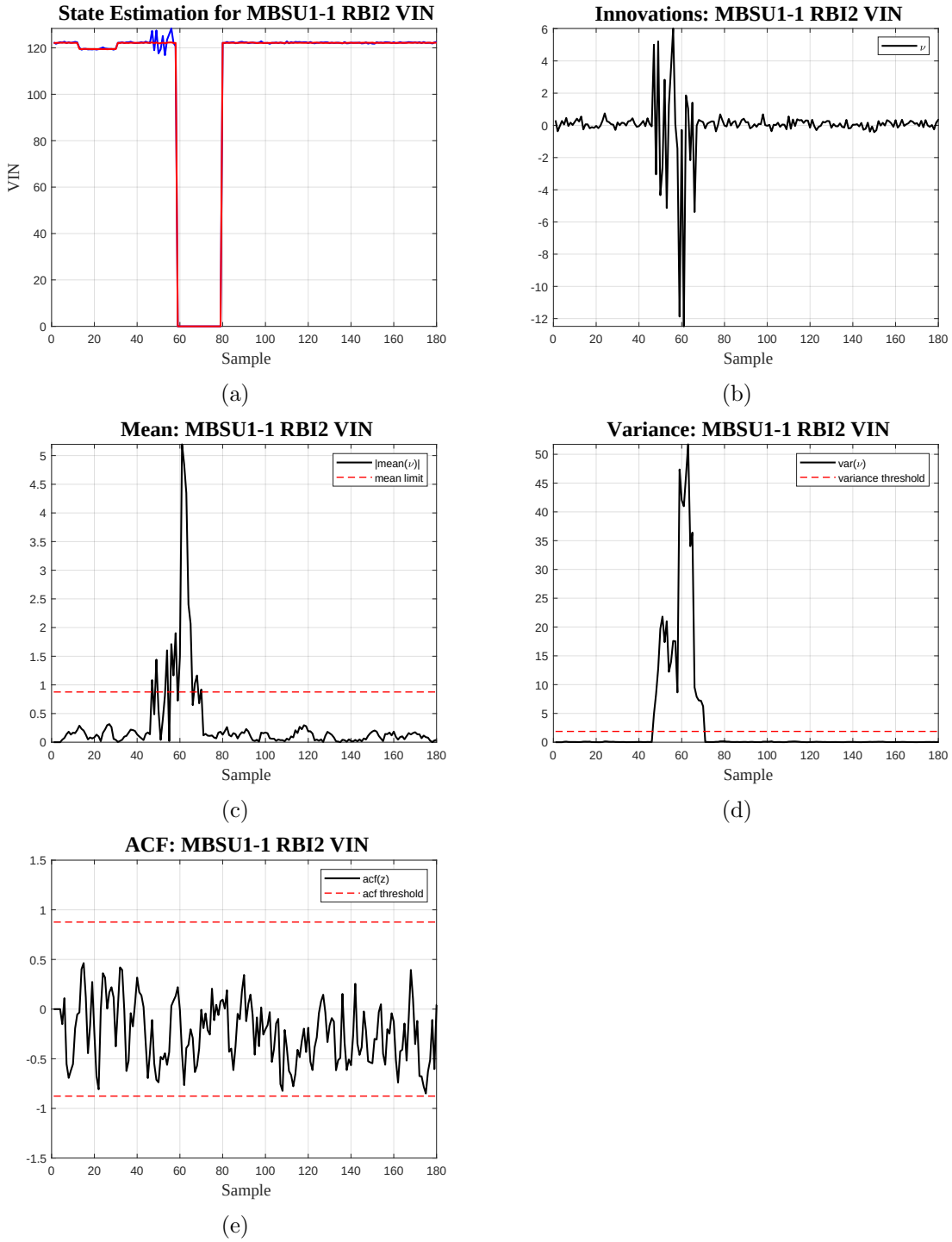


Figure 4.13: DSE response with FDD during excessive sensor noise on MBSU1-1 RBI2 and statistical tests on innovations

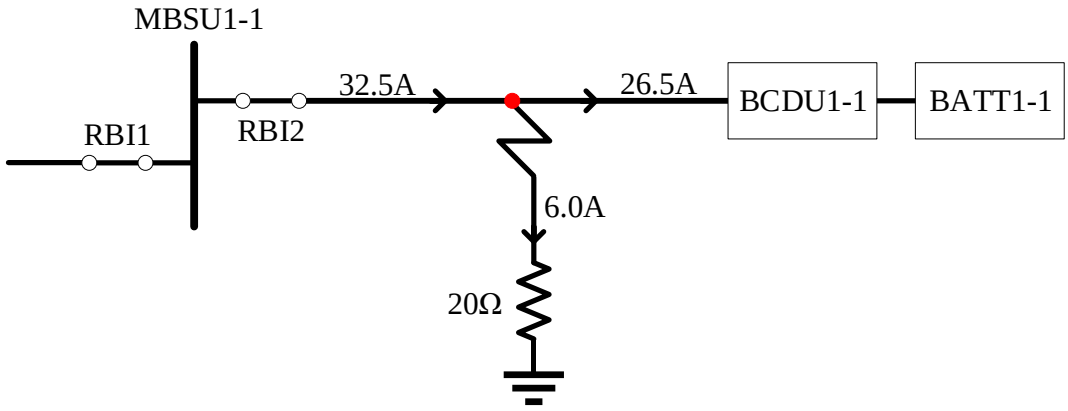


Figure 4.14: One line diagram of a high impedance line to ground fault between MBSU1-1 RBI1 and BCDU1-1

and BCDU1-1. In this fault model, the residual signal is used to generate the fault output matrix **D**, as discussed in the previous chapter.

After enough residuals have been collected to perform the statistical tests, both fault models pass the tests. Since the faults are indistinguishable by the current set of data, active fault diagnosis (AFD) is implemented to send an auxiliary signal to the controller. At sample 60, the auxiliary signal is sent to open RBI2 and turn off the battery discharge from BCDU1-1 to probe for additional data. Fig. 4.16 shows a one line diagram of the connection after the AFD signal has been sent. Next, the FDD tool analyzes the fault model residuals to determine that sensor BCDU1-1 I is functioning normally. Thus the high impedance line to ground is determined to be the true cause of the fault.

Because the AFD signal isolates the fault, no further reconfiguration steps are needed to safe the power system. The controller remains in this state until the fault is repaired and the power system is reset at sample 124. The high impedance fault case is a good example of how small failures can be difficult to diagnose, especially in the face of potentially erroneous data. Model based techniques combined with AFD are a powerful method of differentiating similar fault types of both large and small magnitude.

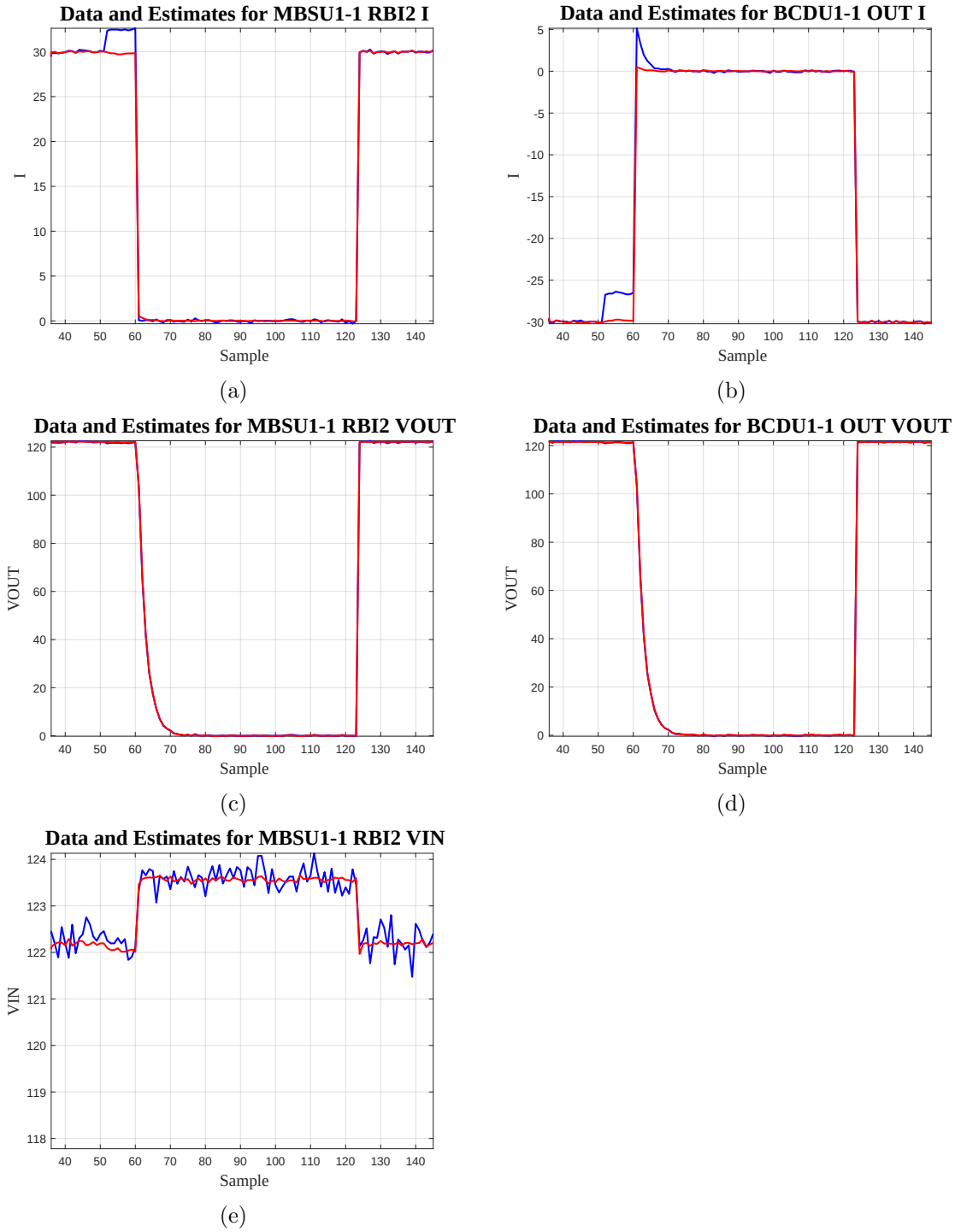


Figure 4.15: DSE response to a high impedance fault between MBSU1-1 RBI2 and BCDU1-1

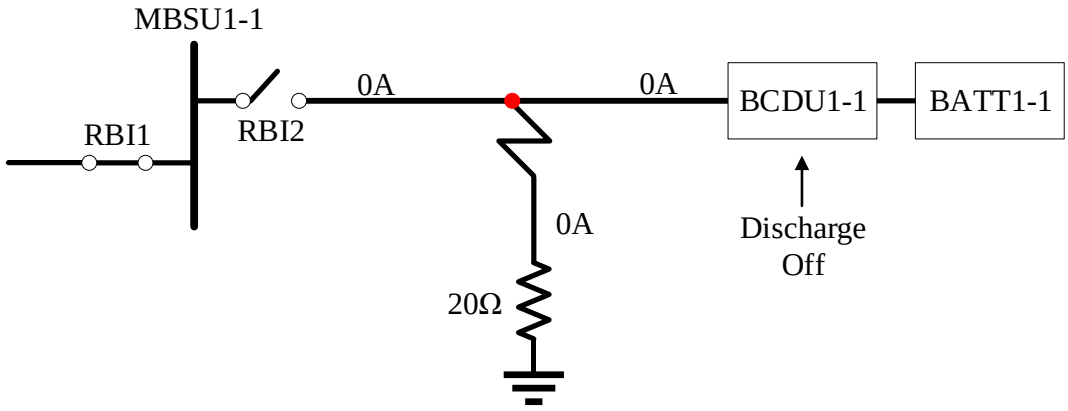


Figure 4.16: One line diagram of a high impedance line to ground fault between after auxiliary AFD signal

4.3 Multiple Faults

The ability to troubleshoot failures, repair equipment, or perform component maintenance for exploratory spacecraft is often very limited due to the challenges presented in deep space. Unmanned vehicles likely have little or no ability to perform these operations. Therefore it is critical for the FDD tool onboard the spacecraft to have the ability to identify multiple sequential faults as they enter/leave the system. To analyze this capability of the FDD software, several experiments will be conducted. In these experiments, faults are injected into the power system simulation in sequence. Because the number of possible fault combinations is exponentially large, Markov Chain (MC) is used to randomly generate the fault state of the power system.

4.3.1 Markov Chain Fault Generator

In the next experiment, Markov chains are used to randomly generate sequences of faults. Markov chains are stochastic models that describe a sequence of possible events. Markov chains are integer time processes where each random variable x_t, $t \geq 0$ is integer valued and depends only on the state of the previous event x_{t-1} (the markovian property) [111]. At every time instant t, the system makes a transition in the state space $S = 1, 2, ..., M$. The Markov process is parameterized by a transition probability matrix $\mathbf{P}$, where $p_{i,j}$ represents

the probability of a transitioning from state i to state j.

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1M} \\ p_{21} & p_{22} & \cdots & p_{2M} \\ \cdots & & & \\ p_{M1} & p_{M2} & \cdots & p_{MM} \end{bmatrix}.$$

The sum of each row in $\mathbf{P}$ must equal one, thus ensuring a state transition at time t. The probability of generating a sequence X in a Markov chain is

$$Pr(X) = \pi(x_0) \prod_{t=1}^{T} pr(x_t|x_{t-1}) \tag{4.2}$$

where $pr(x_t|x_{t-1})$ is the probability of observing the state x_t at time instant t given that the previous state is x_{t-1}. The probability of the initial state is given by $\pi(x_0)$.

The random fault generator will determine which fault state the power system is in at a given time t. The state space $S = x_0, x_1, ..., x_M$ is the set of all possible faults in the system with x_0 being the state with no faults. Our experiment will begin in the normal state, thus the initial probability $\pi(x_0) = [1, 0, 0, ..., 0]$. Then the state transition matrix $\mathbf{P}$ is set based on the likelihood of each fault occurring in the system. Due to a lack of available information on the probability of faults occurring in the electric power systems of spacecraft, each fault type (e.g. Short Circuit, Sensor Bias VIN, etc.) will be assigned equal probability. Then each possible location of the fault type (e.g. MBSU1-2 RBI3) is assigned an equal portion of the probability of the fault type. A uniform random number generator is used to update the state of each time instant.

A Markov sequence of 10,000 faults was generated to observe the distribution of faults. Fig. 4.17 shows the number of times each fault was generated in the sequence. Note that each fault type is produced with approximately the same frequency as shown in Fig. 4.18, and that each fault of a given type appears roughly the same amount of times. Based on this result, we know that the sequential tests will be performed with a known fault probability distribution.

To improve the validity of the random fault generator, measurement data from the

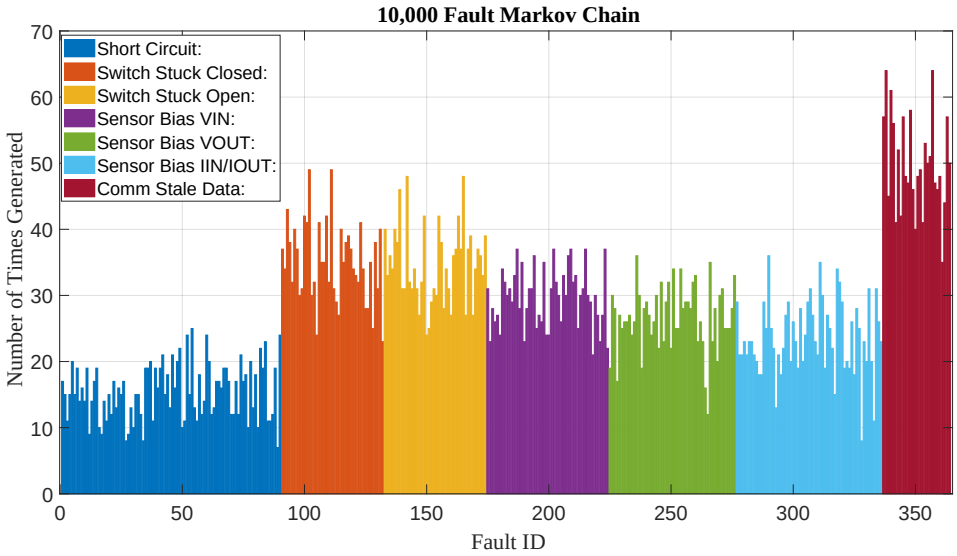


Figure 4.17: Distribution of faults generated from a Markov chain with length 10,000

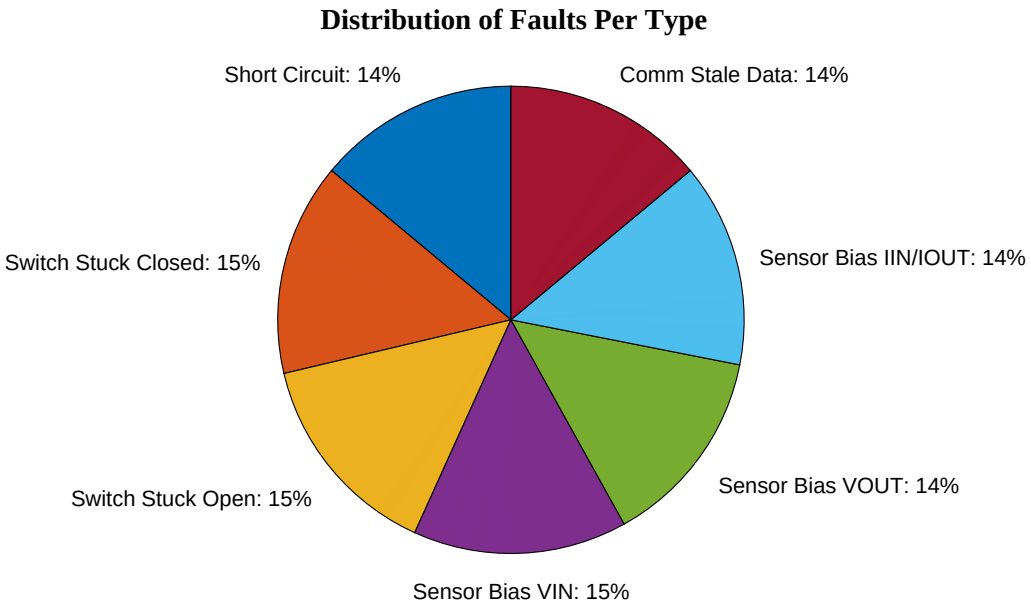


Figure 4.18: Distribution of fault types generated from a Markov chain with length 10,000

power system is used to test whether or not the next fault injected into the power system is physically viable. For example, measurement from the state of a switch can be used to determine that a closed switch could not fault closed. In this event a new fault is generated, and if found to be physically viable, is injected into the power system. Alternatively, if the fault fails the validity check a new fault is randomly generated until a valid fault is found.

4.3.2 Markov Chain Experiment

The number of possible fault sequences in the spacecraft power system is too large to test each one individually. Therefore, a large number of randomly generated fault sequences will be used to estimate the success probability. For each sequence a fault is randomly generated from the Markov chain and injected via the simulation for a predetermined length of time to see if the fault was diagnosed. If the fault was correctly identified, a new fault is injected into the system. This process continues until the fault is failed to be diagnosed. Upon failure, all of the faults are cleared from the simulation and the FDD method is reset to the normal condition.

In this experiment, 100 Markov chain fault sequences were generated. The results of the fault detection and diagnosis tool are shown in Table 4.2. The performance of the software can be characterized by its mean number of faults successfully diagnosed. In this study, an average of 11.1 faults were diagnosed by the system in each test. Insufficient data from sensor and communication faults is the main cause of the FDD method failure. In addition, degradation caused by multiple faults can lead to unsuccessful diagnoses as failures are masked by the blackouts within the power system.

Table 4.2: Markov Chain Test Results

Parameter	**Value**
Number of tests	100
Total number of faults inserted	1200
Mean faults diagnosed	11.1
Standard Deviation	6.75
Minimum faults diagnosed	2
Maximum faults diagnosed	34

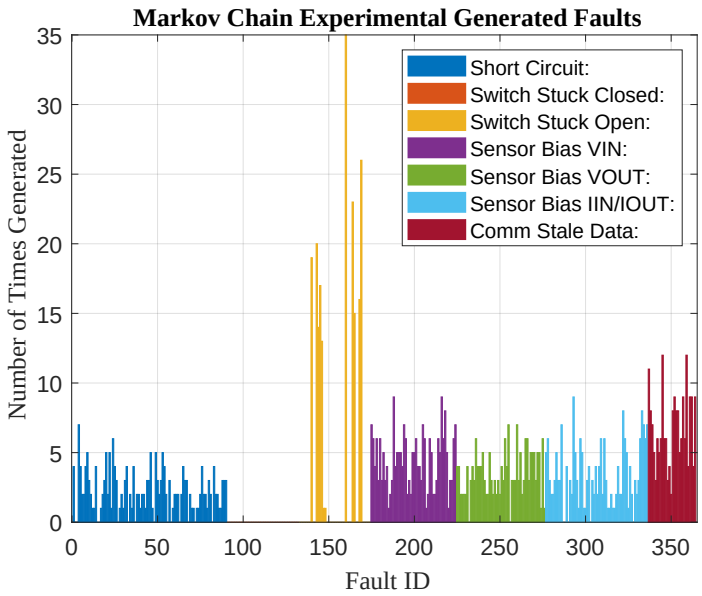


Figure 4.19: Fault Distribution of Markov Chain experiment

Every fault injected during one of the experiments is recorded in Fig. 4.19. The first notable difference in this experiment's distribution is that no switch stuck closed faults were inserted. This is because the Markov chain generator uses power system telemetry to determine which faults are physically viable. These tests set all of the main distribution switches closed in the nominal configuration of the spacecraft. Therefore, a failed switch stuck closed would not be viable. Using the same distribution topology for every test helps increase consistency of the tests to expose weaknesses in the FDD method. The other noticeable difference is that only a subset of the switch stuck open faults are used in the experiment, increasing the overall frequency of the remaining switch stuck open faults. This is because the rule used to detect these faults is based on a voltage differential between the switch. In many cases, the sources and energy storage devices produce a voltage on each end of several of the switches when they are open. Therefore, these faults were removed from this experiment altogether.

Depending on the sequence of faults, the amount of information available for further diagnostics may differ. This is evident in the large range between the minimum and maximum number of faults diagnosed. Fig. 4.20 shows the number of faults successfully diagnosed in each Markov chain test. For example, communication and sensor faults limit the number

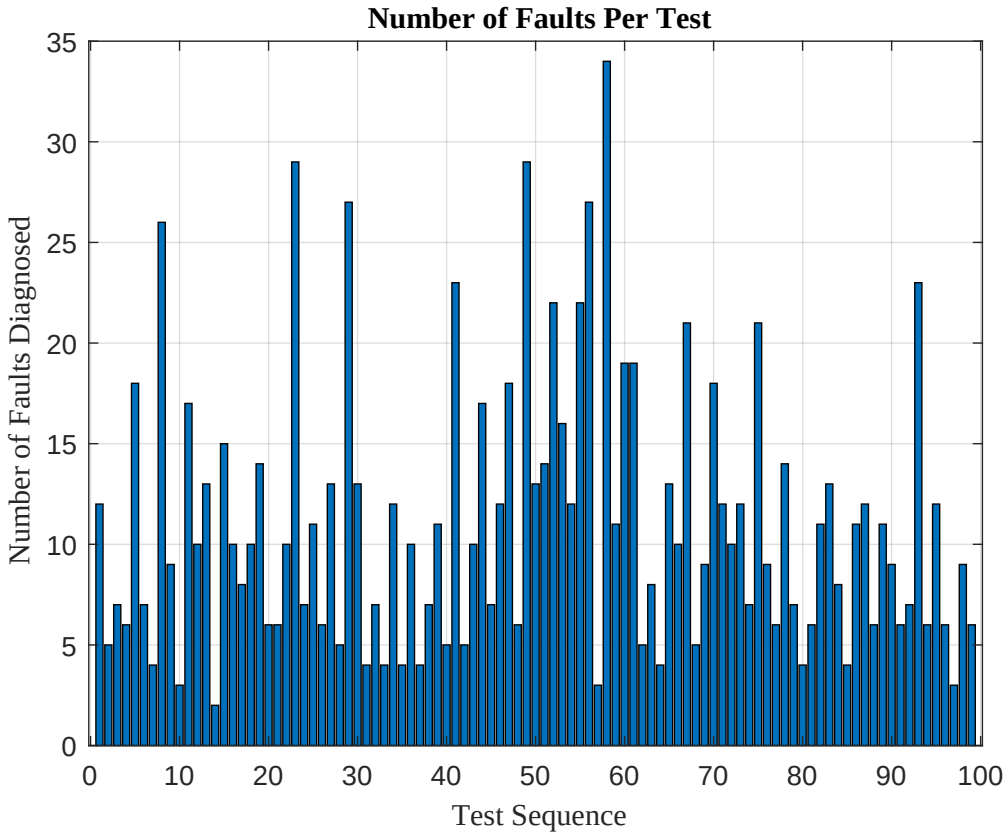


Figure 4.20: Number of faults successfully diagnosed for each Markov chain test

of available sensors used for diagnosis. Hardware failures resulting in loss of power to the distribution system may also decrease the amount of useful data in certain parts of the network.

In order to determine any patterns relating fault types to the ability to diagnose Fig. 4.21 displays a heat map of each fault type in the order that they appeared in all 100 tests. From this, it is evident that sensor bias current faults were less successful further in the fault sequence. This can be expected due to the fact that the current measurements rely on two system states (i.e. bus voltages), which may not be available when several faults exist in the system. Overall the distribution shows no distinct pattern of fault sequences that were successful or failed prematurely.

Another metric that gives insight to the performance of the FDD tool, is the number of times each fault type was failed to be diagnosed. This result is shown in Fig. 4.22. Sensor bias on current measurements were the highest fault that was failed to be diagnosed,

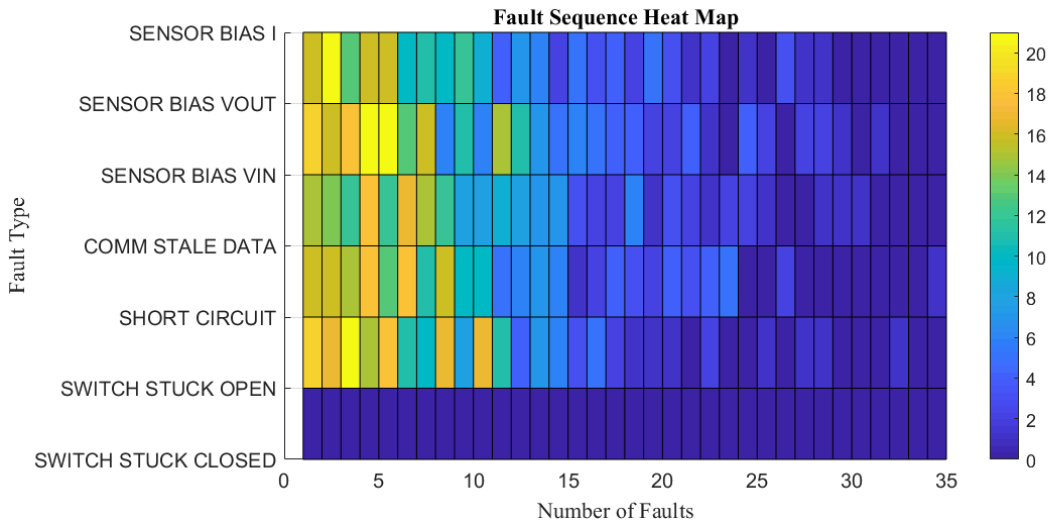


Figure 4.21: Heat map showing the number of each fault type inserted during the 100 Markov chain experiments

followed by switch stuck open. However, the distribution of failures among all fault types was relatively even. Communication stale data faults and sensor bias input voltage faults had the fewest failed diagnoses. Table 4.3 shows the fault types and locations that resulted in multiple failed diagnoses. Overall the distribution is rather even. Short circuit on MBSU1-3 RBI1 was the greatest with 5 failures. (Note: need to check the code to see if there is a problem here!).

Overall, the results of this hierarchical FDD method were successful in diagnosing multiple faults in the system. Fault detection averages at one second, and diagnosis averages under six seconds. This rate is largely dependent on the polling rate of the communication system. With higher frequency data, diagnosis can be cut down significantly. This study made the assumption that all faults would remain in the system until an astronaut is able to clear them manually. Thus only the insertion of faults was considered. In certain cases, faults may return to their normal functionality. This feature must be identified by the FDD software. Therefore more studies should be conducted in which faults are both inserted and removed at random times.

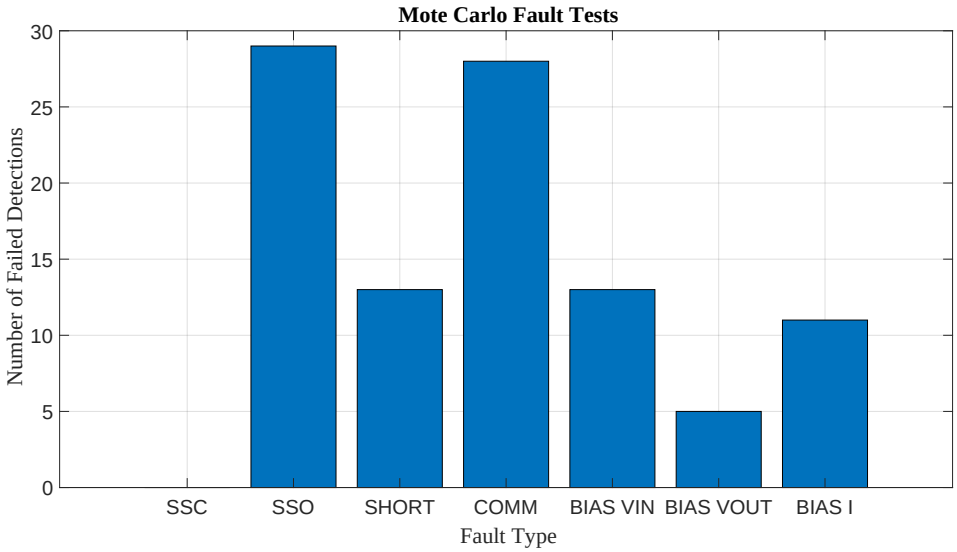


Figure 4.22: Number of times each fault type was failed to be diagnosed

Table 4.3: Repeated Failed Fault Diagnoses

Fault Type	**ORU ID**	**Switch ID**	**Number of Failed Diagnoses**
SWITCH STUCK OPEN	MBSU2-3	RBI1	9
COMM STALE DATA	MBSU1-3		5
SWITCH STUCK OPEN	MBSU2-1	RBI2	5
COMM STALE DATA	MBSU2-3		5
SWITCH STUCK OPEN	MBSU1-2	RBI3	4
SWITCH STUCK OPEN	MBSU1-3	RBI1	3
SWITCH STUCK OPEN	MBSU2-3	RBI5	2
COMM STALE DATA	MBSU2-1		2
SWITCH STUCK OPEN	MBSU2-2	RBI3	2
SENSOR FAULT BIAS IOUT	SAR2-1		2
COMM STALE DATA	MBSU2-X		2
SHORT CIRCUIT	MBSU2-3	RBI1	2
COMM STALE DATA	PDU2-2		2
COMM STALE DATA	SAR1-1		2
SENSOR FAULT BIAS VIN	DDCU2-1		2

Chapter 5

Conclusions and Future Work

Significant improvements can be made to the current state of fault supervision for spacecraft electric power systems. This technology has the potential to provide greater reliability and robustness in the event of power failures in comparison to the current standard.

5.1 Conclusions

The goal of this dissertation was to provide a software tool capable of detecting and diagnosing a wide range of DC power system related faults for spacecraft. It was critical that the faults were diagnosed quickly and accurately, so that the system would not experience further degradation. It also allows the controller to reconfigure the power system so that the maximum number of critical loads are powered with minimal outages.

A three-level hierarchical framework was developed to detect and diagnose a variety of power system component faults. The level at which the fault is detected depends on the amount of data needed to reliably recognize the fault condition. All faults are diagnosed at the central level, where the steady state behavior of the faulted power system is verified via a dynamic fault filter. In a similar fashion, the removal of faults can also be detected and diagnosed.

To accommodate the existing automatic protection scheme, a combination of rules-based and model-based techniques were applied. The rules-based approach allows for quick fault detection that is easy to design and understand. Unfortunately, these techniques can trigger

false alarms or missed faults due to disturbances, noise, and erroneous or missing data. The model-based technique, while computationally complex, provides a more robust picture of the full system behavior and is capable of overcoming the three aforementioned challenges. The relationship between the two methods provided accurate diagnostic capability, while reducing the overall computational power needed at the central control level. This framework can easily be expanded to larger systems with additional failure modes.

Due to the lack of available hardware a spacecraft power system simulation was modified to generate power system telemetry. This included the ability to set and remove over 380 unique fault types.

A number of special topic applications for the fault supervision tool were discussed.

1. A parameter estimation scheme is was used to automatically generate the impedance characteristic for the Kalman filter model.

2. A low-power microcontroller implementation of the state estimator was developed as a co-processor for the application of low powered spacecraft.

3. An unscented Kalman filter was developed for nonlinear fault diagnosis applications such as a DC-DC power converter.

4. A whitening filter was applied to hardware testbed data so that the colored noise could be used with the Kalman filter for fault diagnosis.

5. Active fault diagnosis was used to separate indistinguishable fault types through the use of an auxiliary control signal. The case of a high impedance line to ground fault was analyzed.

To measure the success of the fault supervision tool, each of the faults were tested by injecting the fault and checking for a diagnosis. Then the fault was removed and processed to see if the tool diagnosed the return to the normal state. In this condition, all of the faults were able to be diagnosed, as well as determine that the fault had returned to normal functionality, with the exception of a few switch stuck open faults, where the rules-based technique failed to detect the fault. To provide a deeper understanding as to how the FFD method performs the diagnosis, experimental data was analyzed and presented.

In the event that the spacecraft is unmanned or if there are limited astronaut resources, a fault may not be able to be repaired quickly. In such a situation, it is critical that the fault diagnostic tool be able to continue monitoring for faults, even after one or more faults have already been identified. To test the success of the tool, a Markov chain was used to randomly generate faults in sequence. The software was successful in diagnosing an average of 11 faults per sequence. The limiting factor in the Markov test was fault observability due to losses in data caused by faulty sensors and communication failures. Overall, the method was successful in detecting and diagnosing sequential faults.

This research provides a platform to be used for electrical power system fault detection and diagnosis for the application of spacecraft. The fault test suite and simulation were used to determine the overall performance of the tool in a variety of conditions. Further work is needed to mature the code base and verify the capability against hardware.

5.2 Future Work

There are several areas of work needed to mature in order to expand the capability of the fault detection and diagnosis tool presented here.

First, the FDD software should be operated with a real hardware testbed in the loop. Due to a lack of availability to power system hardware, these experiments were conducted only against simulation data. Hardware testing is the most critical step in verifying the overall performance and capability of the software.

Using the hardware test bed, several enhancements can be made to the FDD method. Modifications to the dynamic system model can be made to address any differences between the model and the hardware. In addition, the hardware telemetry will provide additional valuable information about the noise and disturbances characteristics of the testbed. This information is critical for setting the fault detection thresholds. Furthermore, there may be hardware specific failures that the sofware must be capable of handling. Unfortunately the power system simulation cannot expose certain failures for a particular set of hardware. All of these issues must be addressed in order to improve upon the current FDD method.

Next, converting the software into a flight suitable language, such as NASA's core flight

software (cFS) is another step that must taken to mature the project. After the FDD tool have been extensively tested and vetted the code should be implemented in a full flight environment. This includes running the code on a flight computer, following the computational requirements laid out by NASA.

Another area of future work includes expanding the number of known fault types that the tool is able to detect. Although this dissertation covered a wide range of fault types from different devices, there are a number of other fault modes that were not able to be simulated in the context of this work. Power supply failures, line-to-return failures, and communication cable failures are a few of the types that were identified, but not yet included in this effort. As mentioned before, other hardware related failures may exist that cannot easily be predicted or simulated.

A known limitation of most fault diagnostic methods is that they can only identify failures explicitly defined by the software. Even with thorough research and testing, it is impractical to determine every possible fault type that may occur. Therefore there may be instances where the FDD method detects a failure, but is unable to diagnose the cause of the fault. More research should be conducted to determine the best method to resolve the unknown fault type. This may include using the residual vector to develop a hypothesis as to the type of corrective action that should be taken. This technique may also be useful in diagnosing multiple simultaneous faults.

Lastly, the link between protection and controls is critical in completing system objectives. Fault accommodation should be addressed within the controller to ensure that any faults that are unable to be repaired or isolated have minimal impact to the operation. The coordination between fault detection and diagnosis and generation, distribution and load control is a critical aspect in optimizing system performance. An integrated control and fault supervisory scheme could be used to achieve optimal service to the spacecraft critical loads.

Appendices

Appendix A

Supported Fault Types

Table A.1: Short Circuit Faults

Id	ORU	Switch	Type
8.1	MBSU1-1	RBI1	Short Circuit
8.2	MBSU1-1	RBI2	Short Circuit
8.3	MBSU1-1	RBI3	Short Circuit
8.4	MBSU1-X	RBI1	Short Circuit
8.5	MBSU1-X	RBI2	Short Circuit
8.6	MBSU1-X	RBI3	Short Circuit
8.7	MBSU1-X	RBI4	Short Circuit
8.8	MBSU1-X	RBI5	Short Circuit
8.9	MBSU1-2	RBI1	Short Circuit
8.10	MBSU1-2	RBI3	Short Circuit
8.11	MBSU1-2	RBI4	Short Circuit
8.12	MBSU1-3	RBI1	Short Circuit
8.13	MBSU1-3	RBI2	Short Circuit
8.14	MBSU1-3	RBI3	Short Circuit
8.15	MBSU1-3	RBI4	Short Circuit
8.16	MBSU1-3	RBI5	Short Circuit
8.17	PDU1-1	RBI1	Short Circuit
8.18	PDU1-1	RBI2	Short Circuit
1.19	PDU1-2	RBI1	Short Circuit
8.20	PDU1-2	RBI2	Short Circuit
8.21	PDU1-3	RBI1	Short Circuit
8.22	PDU1-3	RBI2	Short Circuit
8.23	PDU1-1	RPC1	Short Circuit
8.24	PDU1-1	RPC2	Short Circuit
8.25	PDU1-1	RPC3	Short Circuit
8.26	PDU1-1	RPC4	Short Circuit
8.27	PDU1-1	RPC5	Short Circuit
8.28	PDU1-1	RPC6	Short Circuit
8.29	PDU1-1	RPC7	Short Circuit
8.30	PDU1-1	RPC8	Short Circuit
8.31	PDU1-2	RPC1	Short Circuit
8.32	PDU1-2	RPC2	Short Circuit
8.33	PDU1-2	RPC3	Short Circuit
8.34	PDU1-2	RPC4	Short Circuit
8.35	PDU1-2	RPC5	Short Circuit
8.36	PDU1-2	RPC6	Short Circuit
8.37	PDU1-2	RPC7	Short Circuit
8.38	PDU1-2	RPC8	Short Circuit
8.39	PDU1-3	RPC1	Short Circuit
8.40	PDU1-3	RPC2	Short Circuit
8.41	PDU1-3	RPC3	Short Circuit
8.42	PDU1-3	RPC4	Short Circuit
8.43	PDU1-3	RPC5	Short Circuit
8.44	PDU1-3	RPC6	Short Circuit
8.45	PDU1-3	RPC7	Short Circuit
8.46	PDU1-3	RPC8	Short Circuit

Table A.2: Switch Stuck Closed Faults

Id	ORU	Switch	Type
7.1	MBSU1-1	RBI1	Switch Stuck Closed
7.2	MBSU1-1	RBI2	Switch Stuck Closed
7.3	MBSU1-1	RBI3	Switch Stuck Closed
7.4	MBSU1-X	RBI1	Switch Stuck Closed
7.5	MBSU1-X	RBI2	Switch Stuck Closed
7.6	MBSU1-X	RBI3	Switch Stuck Closed
7.7	MBSU1-X	RBI4	Switch Stuck Closed
7.8	MBSU1-X	RBI5	Switch Stuck Closed
7.9	MBSU1-2	RBI1	Switch Stuck Closed
7.10	MBSU1-2	RBI3	Switch Stuck Closed
7.11	MBSU1-2	RBI4	Switch Stuck Closed
7.12	MBSU1-3	RBI1	Switch Stuck Closed
7.13	MBSU1-3	RBI2	Switch Stuck Closed
7.14	MBSU1-3	RBI3	Switch Stuck Closed
7.15	MBSU1-3	RBI4	Switch Stuck Closed
7.16	MBSU1-3	RBI5	Switch Stuck Closed
7.17	PDU1-1	RBI1	Switch Stuck Closed
7.18	PDU1-1	RBI2	Switch Stuck Closed
7.19	PDU1-2	RBI1	Switch Stuck Closed
7.20	PDU1-2	RBI2	Switch Stuck Closed
7.21	PDU1-3	RBI1	Switch Stuck Closed
7.22	PDU1-3	RBI2	Switch Stuck Closed

Table A.3: Switch Stuck Open Faults

Id	ORU	Switch	Type
6.1	MBSU1-1	RBI1	Switch Stuck Open
6.2	MBSU1-1	RBI2	Switch Stuck Open
6.3	MBSU1-1	RBI3	Switch Stuck Open
6.4	MBSU1-X	RBI1	Switch Stuck Open
6.5	MBSU1-X	RBI2	Switch Stuck Open
6.6	MBSU1-X	RBI3	Switch Stuck Open
6.7	MBSU1-X	RBI4	Switch Stuck Open
6.8	MBSU1-X	RBI5	Switch Stuck Open
6.9	MBSU1-2	RBI1	Switch Stuck Open
6.10	MBSU1-2	RBI3	Switch Stuck Open
6.11	MBSU1-2	RBI4	Switch Stuck Open
6.12	MBSU1-3	RBI1	Switch Stuck Open
6.13	MBSU1-3	RBI2	Switch Stuck Open
6.14	MBSU1-3	RBI3	Switch Stuck Open
6.15	MBSU1-3	RBI4	Switch Stuck Open
6.16	MBSU1-3	RBI5	Switch Stuck Open
6.17	PDU1-1	RBI1	Switch Stuck Open
6.18	PDU1-1	RBI2	Switch Stuck Open
6.19	PDU1-2	RBI1	Switch Stuck Open
6.20	PDU1-2	RBI2	Switch Stuck Open
6.21	PDU1-3	RBI1	Switch Stuck Open
6.22	PDU1-3	RBI2	Switch Stuck Open

Table A.4: Sensor Bias VIN Faults

Id	ORU	Switch	Type
1.1	MBSU1-1	RBI1	Sensor Bias Vin
1.2	MBSU1-1	RBI2	Sensor Bias Vin
1.3	MBSU1-1	RBI3	Sensor Bias Vin
1.4	MBSU1-X	RBI1	Sensor Bias Vin
1.5	MBSU1-X	RBI2	Sensor Bias Vin
1.6	MBSU1-X	RBI3	Sensor Bias Vin
1.7	MBSU1-X	RBI4	Sensor Bias Vin
1.8	MBSU1-X	RBI5	Sensor Bias Vin
1.9	MBSU1-2	RBI1	Sensor Bias Vin
1.10	MBSU1-2	RBI3	Sensor Bias Vin
1.11	MBSU1-2	RBI4	Sensor Bias Vin
1.12	MBSU1-3	RBI1	Sensor Bias Vin
1.13	MBSU1-3	RBI2	Sensor Bias Vin
1.14	MBSU1-3	RBI3	Sensor Bias Vin
1.15	MBSU1-3	RBI4	Sensor Bias Vin
1.16	MBSU1-3	RBI5	Sensor Bias Vin
1.17	PDU1-1	RBI1	Sensor Bias Vin
1.18	PDU1-1	RBI2	Sensor Bias Vin
1.19	PDU1-2	RBI1	Sensor Bias Vin
1.20	PDU1-2	RBI2	Sensor Bias Vin
1.21	PDU1-3	RBI1	Sensor Bias Vin
1.22	PDU1-3	RBI2	Sensor Bias Vin
1.23	DDCU-1	IN	Sensor Bias Vin
1.24	DDCU-2	IN	Sensor Bias Vin
1.25	PPU1-1	IN	Sensor Bias Vin
1.26	PPU1-2	IN	Sensor Bias Vin

Table A.5: Sensor Bias VOUT Faults

Id	ORU	Switch	Type
2.1	MBSU1-1	RBI1	Sensor Bias Vout
2.2	MBSU1-1	RBI2	Sensor Bias Vout
2.3	MBSU1-1	RBI3	Sensor Bias Vout
2.4	MBSU1-X	RBI1	Sensor Bias Vout
2.5	MBSU1-X	RBI2	Sensor Bias Vout
2.6	MBSU1-X	RBI3	Sensor Bias Vout
2.7	MBSU1-X	RBI4	Sensor Bias Vout
2.8	MBSU1-X	RBI5	Sensor Bias Vout
2.9	MBSU1-2	RBI1	Sensor Bias Vout
2.10	MBSU1-2	RBI3	Sensor Bias Vout
2.11	MBSU1-2	RBI4	Sensor Bias Vout
2.12	MBSU1-3	RBI1	Sensor Bias Vout
2.13	MBSU1-3	RBI2	Sensor Bias Vout
2.14	MBSU1-3	RBI3	Sensor Bias Vout
2.15	MBSU1-3	RBI4	Sensor Bias Vout
2.16	MBSU1-3	RBI5	Sensor Bias Vout
2.17	PDU1-1	RBI1	Sensor Bias Vout
2.18	PDU1-1	RBI2	Sensor Bias Vout
2.19	PDU1-2	RBI1	Sensor Bias Vout
2.20	PDU1-2	RBI2	Sensor Bias Vout
2.21	PDU1-3	RBI1	Sensor Bias Vout
2.22	PDU1-3	RBI2	Sensor Bias Vout
2.23	SAR1-1	OUT	Sensor Bias Vout
2.24	BCDU1-1	OUT	Sensor Bias Vout
2.25	BCDU1-2	OUT	Sensor Bias Vout
2.26	DDCU1-1	OUT	Sensor Bias Vout
2.27	DDCU1-2	OUT	Sensor Bias Vout

Table A.6: Sensor Bias IIN Faults

Id	ORU	Switch	Type
4.1	DDCU1-1	IN	Sensor Bias Iin
4.2	DDCU1-2	IN	Sensor Bias Iin
4.3	PPU1-1	IN	Sensor Bias Iin
4.4	PPU1-2	IN	Sensor Bias Iin

Table A.7: Sensor Bias IOUT Faults

Id	ORU	Switch	Type
3.1	MBSU1-1	RBI1	Sensor Bias Iout
3.2	MBSU1-1	RBI2	Sensor Bias Iout
3.3	MBSU1-1	RBI3	Sensor Bias Iout
3.4	MBSU1-X	RBI1	Sensor Bias Iout
3.5	MBSU1-X	RBI2	Sensor Bias Iout
3.6	MBSU1-X	RBI3	Sensor Bias Iout
3.7	MBSU1-X	RBI4	Sensor Bias Iout
3.8	MBSU1-X	RBI5	Sensor Bias Iout
3.9	MBSU1-2	RBI1	Sensor Bias Iout
3.10	MBSU1-2	RBI3	Sensor Bias Iout
3.11	MBSU1-2	RBI4	Sensor Bias Iout
3.12	MBSU1-3	RBI1	Sensor Bias Iout
3.13	MBSU1-3	RBI2	Sensor Bias Iout
3.14	MBSU1-3	RBI3	Sensor Bias Iout
3.15	MBSU1-3	RBI4	Sensor Bias Iout
3.16	MBSU1-3	RBI5	Sensor Bias Iout
3.17	PDU1-1	RBI1	Sensor Bias Iout
3.18	PDU1-1	RBI2	Sensor Bias Iout
3.19	PDU1-2	RBI1	Sensor Bias Iout
3.20	PDU1-2	RBI2	Sensor Bias Iout
3.21	PDU1-3	RBI1	Sensor Bias Iout
3.22	PDU1-3	RBI2	Sensor Bias Iout
3.23	SAR1-1	OUT	Sensor Bias Iout
3.24	BCDU1-1	OUT	Sensor Bias Iout
3.25	BCDU1-2	OUT	Sensor Bias Iout
3.26	DDCU1-1	OUT	Sensor Bias Iout
3.27	DDCU1-2	OUT	Sensor Bias Iout

Table A.8: Stale Data Communication Faults

Id	ORU	Switch	Type
5.1	MBSU1-X	N/A	Stale Data
5.2	MBSU1-1	N/A	Stale Data
5.3	MBSU1-2	N/A	Stale Data
5.4	MBSU1-3	N/A	Stale Data
5.5	PDU1-1	N/A	Stale Data
5.6	PDU1-2	N/A	Stale Data
5.7	PDU1-3	N/A	Stale Data
5.8	SAR1-1	N/A	Stale Data
5.9	BCDU1-1	N/A	Stale Data
5.10	BCDU1-2	N/A	Stale Data
5.11	DDCU1-1	N/A	Stale Data
5.12	DDCU1-2	N/A	Stale Data
5.13	PPU1-1	N/A	Stale Data
5.14	PPU1-2	N/A	Stale Data

Bibliography

[1] S. Huang, K. K. Tan, P. V. Er, and T. H. Lee, *Intelligent Fault Diagnosis and Accommodation Control*. CRC Press, 2020.

[2] I. Hwang, S. Kim, Y. Kim, and C. E. Seah, "A Survey of Fault Detection, Isolation, and Reconfiguration Methods," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 3, pp. 636–653, 2010.

[3] P. M. Frank, "Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy: A survey and some new results," *automatica*, vol. 26, no. 3, pp. 459–474, 1990.

[4] R. Isermann, "Supervision, fault-detection and fault-diagnosis methods—an introduction," *Control engineering practice*, vol. 5, no. 5, pp. 639–652, 1997.

[5] J. Minkel, "The 2003 northeast blackout–five years later," *Scientific American*, vol. 13, 2008.

[6] T. E. D. Liacco, "The adaptive reliability control system," *IEEE Transactions on Power Apparatus and Systems*, no. 5, pp. 517–531, 1967.

[7] T. E. Dy Liacco, "Real-Time Computer Control of Power Systems," *Proceedings of the IEEE*, vol. 62, no. 7, pp. 884–891, 1974.

[8] L. H. Fink and K. Carlsen, "Operating under stress and strain [electrical power systems control under emergency conditions]," *IEEE spectrum*, vol. 15, no. 3, pp. 48–53, 1978.

[9] A. Sajadi, "ANALYTICAL TOOLS FOR TRANSMISSION PLANNING STUDIES FOR OFFSHORE WIND FARM INTEGRATION," pp. 1–158, 2016.

[10] G. Aaseng, J. Frank, M. Iatauro, C. Knight, R. Levinson, J. Ossenfort, M. Scott, A. Sweet, J. Csank, J. Soeder *et al.*, "Development and testing of a vehicle management system for autonomous spacecraft habitat operations," 2018.

[11] J. Frank, L. Spirkovska, R. McCann, L. Wang, K. Pohlkamp, and L. Morin, "Autonomous mission operations," in *2013 IEEE Aerospace Conference*. IEEE, 2013, pp. 1–20.

[12] K. Loparo and F. Abdel-Malek, "A Probabilistic Approach To Dynamic Power System Security," *IEEE Transactions on Circuits and Systems*, vol. 37, no. 6, pp. 787–798, 1990.

[13] R. Isermann, *Fault-diagnosis systems: an introduction from fault detection to fault tolerance*. Springer Science & Business Media, 2006.

[14] A. Zolghadri, D. Henry, J. Cieslak, D. Efimov, and P. Goupil, *Fault Diagnosis and Fault-Tolerant Control and Guidance for Aerospace Vehicles*, 2014, vol. 1.

[15] L. Fesq, "Current Fault Management Trends in NASA's Planetary Spacecraft," *Aerospace Conference, 2009 IEEE*, pp. 1–9, 2009.

[16] H. A. Cikanek, "Space Shuttle Main Engine Failure Detection," *IEEE Control Systems*, vol. 6, pp. 13–18, 1986.

[17] S. Poll, A. Patterson-Hine, J. Camisa, D. Garcia, D. Hall, C. Lee, O. J. Mengshoel, C. Neukom, D. Nishikawa, J. Ossenfort *et al.*, "Advanced diagnostics and prognostics testbed," in *Proceedings of the 18th International Workshop on Principles of Diagnosis (DX-07)*, 2007, pp. 178–185.

[18] B. Ricks and O. J. Mengshoel, "Methods for Probabilistic Fault Diagnosis: An Electrical Power System Case Study," Oct. 2009.

[19] O. J. Mengshoel, "Diagnosing Faults in Electrical Power Systems of Spacecraft and Aircraft," NASA Ames Research Center, Tech. Rep.

[20] L. Zhang and X. R. Li, "Detection and Diagnosis of Sensor and Actuator Failures Using IMM Estimator," *IEEE Transactions on Aerospace and Electrical Systems*, vol. 34, no. 4, pp. 1293–1313, 1998.

[21] K. Keller, K. Swearingen, and J. Sheahan, "Aircraft Electrical Power System Prognostics and Health Management." IEEE, Mar. 2006, pp. 1–12.

[22] D. Gorinevsky, S. Boyd, and S. Poll, "Estimation of Faults in DC Electrical Power System." IEEE, Jun. 2009, pp. 4334–4339.

[23] L. Fesq and K. Ksenia, "Model-based off-nominal state isolation and detection system for autonomous fault management," *Aerospace Conference, 2016 IEEE*, p. 1, 2016.

[24] P. S. Nag, G. kumar Silla, V. H. V. Gummadi, C. Harishankar, V. K. Ray, and C. S. Kumar, "Model based fault diagnosis of low earth orbiting (leo) satellite using spherical unscented kalman filter," *IFAC-PapersOnLine*, vol. 49, no. 1, pp. 635–638, 2016.

[25] M. Hong, K. Loparo, and W. Culver, "Towards a More Reliable and Efficient Electric Distribution System," *Energy Tech*, vol. 1, pp. 1–5, 2013.

[26] M. Amin and B. F. Wollenberg, "Toward a smart grid-power delivery for the 21st century," *IEEE Power and Energy Magazine*, vol. 1, p. 114, 2008.

[27] X. Zhiying and Y. M. Fei Qiao, "Design and implementation of inference engine for fault prognosis in Power System." IEEE, Mar. 2010.

[28] V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri, and K. Yin, "A review of process fault detection and diagnosis: Part iii: Process history based methods," *Computers & chemical engineering*, vol. 27, no. 3, pp. 327–346, 2003.

[29] E. Cardozo and S. N. Talukdar, "A distributed expert system for fault diagnosis," *IEEE Transactions on Power Systems*, vol. 3, no. 2, pp. 641–646, 1988.

[30] Z. Yongli, H. Limin, and L. Jinling, "Bayesian Networks-Based Approach for Power Systems Fault Diagnosis," *IEEE Transactions on Power Delivery*, vol. 21, no. 2, pp. 634–639, 2006.

[31] F. N. Chowdhury, J. P. Christensen, and J. L. Aravena, "Power System Fault Detection and State Estimation Using Kalman Filter with Hypothesis Testing," *IEEE Transactions on Power Delivery*, vol. 6, no. No. 3, pp. 1025–1030, 1991.

[32] A. A. Grigis and G. R. Brown, "Application of Kalman Filtering in Computer Relaying," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-100, no. NO. 7, pp. 3387–3397, 1981.

[33] D. Gorinevsky, S. Boyd, and S. Poll, "Estimation of faults in dc electrical power system," 2009.

[34] G. Gross, A. Bose, C. DeMarco, M. Pai, J. Thorp, and P. Varaiya, "Consortium for Electric Reliability Technology Solutions Grid of the Future: Real Time Security Monitoring and Control Power Systems," *PSERC Transmission Reliability Program*, pp. 1–35, 1999.

[35] T. P. Dever, R. D. May, and P. H. Morris, "Autonomous spacecraft communication interface for load planning," 2014.

[36] R. May and K. Loparo, "The Use of Software Agents for Autonomous Control of a DC Space Power System," *Propulsion and Energy Forum, 12th International Energy Conversion Engineering Conference*, pp. 1–5, 2014.

[37] J. F. Soeder, A. Mcnelis, R. Beach, N. McNelis, T. Dever, L. Trase, and R. May, "Overview of intelligent power controller development for human deep space exploration," in *12th International Energy Conversion Engineering Conference*, 2014, p. 3833.

[38] R. May, J. F. Soeder, R. Beach, P. George, J. D. Frank, M. Schwabacher, S. P. Colombano, L. Wang, and D. Lawler, "An architecture to enable autonomous control

of a spacecraft," in *12th International Energy Conversion Engineering Conference*, 2014, p. 3834.

[39] J. Csank, J. Soeder, J. Follo, M. Muscatello, Y. H. Hau, and M. Carbone, "An intelligent autonomous power controller for the nasa human deep space gateway," in *2018 International Energy Conversion Engineering Conference*, 2018, p. 4634.

[40] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri, "A review of process fault detection and diagnosis part I: Quantitative model-based methods," *Comput. Chem. Eng.*, vol. 27, pp. 297–311, 2003.

[41] J. Gertler, *Fault detection and diagnosis*. Springer, 2013.

[42] ——, *Fault Detection and Diagnosis in Engineering Systems*. New York, NY: Marcel Dekker, Inc, 1998, vol. 1.

[43] H.-J. Cho and J.-K. Park, "An expert system for fault section diagnosis of power systems using fuzzy relations," *IEEE transactions on power systems*, vol. 12, no. 1, pp. 342–348, 1997.

[44] R. Isermann and P. Balle, "Trends in the Application of Model-Based Fault Detection and Diagnosis of Technical Processes," *Control Eng. Practice*, vol. 5, no. 5, pp. 709–719, 1997.

[45] A. S. Willsky, "A survey of design methods for failure detection in dynamic systems," *Automatica*, vol. 12, no. 6, pp. 601–611, 1976.

[46] D. M. Himmelblau, *Fault detection and diagnosis in chemical and petrochemical processes*. Elsevier Science Ltd, 1978, vol. 8.

[47] R. Isermann, "Process fault detection based on modeling and estimation methods—a survey," *automatica*, vol. 20, no. 4, pp. 387–404, 1984.

[48] ——, "Fault diagnosis of machines via parameter estimation and knowledge processing—tutorial paper," *Automatica*, vol. 29, no. 4, pp. 815–835, 1993.

[49] J. J. Gertler, "Survey of model-based failure detection and isolation in complex plants," *IEEE Control systems magazine*, vol. 8, no. 6, pp. 3–11, 1988.

[50] R. V. Beard, "Failure accomodation in linear systems through self-reorganization." Ph.D. dissertation, Massachusetts Institute of Technology, 1971.

[51] R. K. Mehra and J. Peschon, "An Innovations Approach to Fault Detection and Diagnosis in Dynamic Systems," *Automatica*, vol. 4, pp. 637–640, 1971.

[52] A. Willsky and H. Jones, "A generalized likelihood ratio approach to the detection and estimation of jumps in linear systems," *IEEE Transactions on Automatic control*, vol. 21, no. 1, pp. 108–112, 1976.

[53] R. N. Clark, D. C. Fosth, and V. M. Walton, "Detecting instrument malfunctions in control systems," *IEEE Transactions on Aerospace and Electronic Systems*, no. 4, pp. 465–473, 1975.

[54] R. Montgomery and A. CAGLAYAN, "A self-reorganizing digital flight control system for aircraft," in *12th Aerospace Sciences Meeting*, 1974, p. 21.

[55] J. Deckert, M. Desai, J. Deyst, and A. Willsky, "F-8 dfbw sensor failure identification using analytic redundancy," *IEEE Transactions on Automatic Control*, vol. 22, no. 5, pp. 795–803, 1977.

[56] M. Kitamura, "Detection of sensor failures in nuclear plants using analytic redundancy," Oak Ridge National Lab., Tech. Rep., 1980.

[57] H. E. Rauch, "Intelligent fault diagnosis and control reconfiguration," *IEEE Control systems*, vol. 14, no. 3, pp. 6–12, 1994.

[58] E. Chow and A. Willsky, "Analytical redundancy and the design of robust failure detection systems," *IEEE Transactions on automatic control*, vol. 29, no. 7, pp. 603–614, 1984.

[59] J. Gertler, "Fault detection and diagnosis," *Encyclopedia of Systems and Control*, pp. 417–422, 2015.

[60] ——, "Analytical redundancy methods in fault detection and isolation-survey and synthesis," *IFAC Proceedings Volumes*, vol. 24, no. 6, pp. 9–21, 1991.

[61] M. Ilic and S. Liu, *Hierarchical power systems control: its value in a changing industry*. Springer Science & Business Media, 2012.

[62] T. D. Liacco, "An overview of practices and trends in power system control centers," *IFAC Proceedings Volumes*, vol. 11, no. 1, pp. 1499–1510, 1978.

[63] F. Schweppe and J. Wildes, "Power System Static-State Estimation, Part I: Exact Model," *Transactions on Power Apparatus and Systems*, vol. PAS-89, no. 1, pp. 120–125, 1970.

[64] F. Schweppe and D. Rom, "Power System Static-State Estimation Part II: Approximate Model," *Transactions on Power Apparatus and Systems*, vol. PAS-89, no. 1, pp. 125–130, 1970.

[65] F. Schweppe, "Power System Static-State Estimation, Part III: Implementation," *Transactions on Power Apparatus and Systems*, vol. PAS98, no. 1, p. 130, 1970.

[66] J. Zhao, A. Gómez-Expósito, M. Netto, L. Mili, A. Abur, V. Terzija, I. Kamwa, B. Pal, A. K. Singh, J. Qi *et al.*, "Power system dynamic state estimation: Motivations, definitions, methodologies, and future work," *IEEE Transactions on Power Systems*, vol. 34, no. 4, pp. 3188–3198, 2019.

[67] A. S. Meliopoulos, G. J. Cokkinides, P. Myrda, Y. Liu, R. Fan, L. Sun, R. Huang, and Z. Tan, "Dynamic state estimation-based protection: Status and promise," *IEEE Transactions on Power Delivery*, vol. 32, no. 1, pp. 320–330, 2016.

[68] H. Modir and R. A. Schlueter, "A dynamic state estimator for dynamic security assessment," *IEEE Transactions on Power Apparatus and Systems*, no. 11, pp. 4644–4652, 1981.

[69] J. J. Sanchez-Gasca and D. Trudnowski, "Identification of electromechanical modes in power systems," *IEEE Task Force Report, Special Publication TP462*, 2012.

[70] A. K. Singh and B. C. Pal, "Decentralized nonlinear control for power systems using normal forms and detailed models," *IEEE Transactions on Power Systems*, vol. 33, no. 2, pp. 1160–1172, 2017.

[71] ——, "Decentralized control of oscillatory dynamics in power systems using an extended lqr," *IEEE Transactions on Power Systems*, vol. 31, no. 3, pp. 1715–1728, 2015.

[72] M. Ariff and B. Pal, "Adaptive protection and control in the power system for wide-area blackout prevention," *IEEE Transactions on Power Delivery*, vol. 31, no. 4, pp. 1815–1825, 2016.

[73] H. F. Albinali and A. Meliopoulos, "A centralized substation protection scheme that detects hidden failures," in *2016 IEEE Power and Energy Society General Meeting (PESGM)*. IEEE, 2016, pp. 1–5.

[74] V. Vittal, F. Ma, G. Hou, J. McCalley, and L. Tang, "Next generation on-line dynamic security assessment," *Power Systems Engineering Research Center., Arizona, Rep. PSERC Publication*, pp. 11–09, 2011.

[75] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME-Journal of Basic Engineering*, vol. 82, no. D, pp. 35–45, 1960.

[76] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," *Journal of basic engineering*, vol. 83, no. 1, pp. 95–108, 1961.

[77] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[78] D. G. Watts and G. Jenkins, "Spectral analysis and its applications," *San Francisco*, 1968.

[79] T. W. Anderson, "An introduction to multivariate statistical analysis," Wiley New York, Tech. Rep., 1962.

[80] W. D, "New developments in system state estimation and anomaly detection. final report," 1969.

[81] B. G. Gardner, "Modular standards for space power systems," in *70th International Astronautical Congress (IAC)*, 2019.

[82] D. Chen, L. Xu, and L. Yao, "Dc voltage variation based autonomous control of dc microgrids," *IEEE transactions on power delivery*, vol. 28, no. 2, pp. 637–648, 2013.

[83] A. Abusorrah, M. M. Al-Hindawi, Y. Al-Turki, K. Mandal, D. Giaouris, S. Banerjee, S. Voutetakis, and S. Papadopoulou, "Stability of a boost converter fed from photovoltaic source," *Solar energy*, vol. 98, pp. 458–471, 2013.

[84] T. Esram and P. L. Chapman, "Comparison of photovoltaic array maximum power point tracking techniques," *IEEE Transactions on energy conversion*, vol. 22, no. 2, pp. 439–449, 2007.

[85] A. P. N. Tahim, D. J. Pagano, E. Lenz, and V. Stramosk, "Modeling and stability analysis of islanded dc microgrids under droop control," *IEEE Transactions on Power Electronics*, vol. 30, no. 8, pp. 4597–4607, 2015.

[86] J. Glover, M. Sarma, and T. J. Overbye, "Power sytem analysis and design, 2012," *Cengage Learning, Stamford, USA, ISBN-13*, pp. 978–1.

[87] T. Dragicevic, P. Wheeler, and F. Blaabjerg, *DC distribution systems and microgrids*. Institution of Engineering and Technology, 2018.

[88] R. Nadira, T. D. Liacco, and K. Loparo, "A hierarchical interactive approach to electric power system restoration," *IEEE Transactions on Power Systems*, vol. 7, no. 3, pp. 1123–1131, 1992.

[89] L. Tang and B.-T. Ooi, "Protection of vsc-multi-terminal hvdc against dc faults," in *2002 IEEE 33rd Annual IEEE Power Electronics Specialists Conference. Proceedings (Cat. No. 02CH37289)*, vol. 2. IEEE, 2002, pp. 719–724.

[90] S. Foster, L. Xu, and B. Fox, "Control of an lcc hvdc system for connecting large offshore wind farms with special consideration of grid fault," in *2008 IEEE Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century*. IEEE, 2008, pp. 1–8.

[91] A. Meghwani, S. Srivastava, and S. Chakrabarti, "A non-unit protection scheme for dc microgrid based on local measurements," *IEEE Transactions on Power Delivery*, vol. 32, no. 1, pp. 172–181, 2016.

[92] S. Azizi, M. Sanaye-Pasand, M. Abedini, and A. Hasani, "A traveling-wave-based methodology for wide-area fault location in multiterminal dc systems," *IEEE Transactions on Power Delivery*, vol. 29, no. 6, pp. 2552–2560, 2014.

[93] W. Li, A. Monti, and F. Ponci, "Fault detection and classification in medium voltage dc shipboard power systems with wavelets and artificial neural networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 11, pp. 2651–2665, 2014.

[94] Y. Ji and J. Bals, "Multi-model based fault detection for the power system of more electric aircraft," in *2009 7th Asian Control Conference*. IEEE, 2009, pp. 93–98.

[95] N. J. Higham, "Computing a nearest symmetric positive semidefinite matrix," *Linear algebra and its applications*, vol. 103, pp. 103–118, 1988.

[96] T. E. Menke and P. S. Maybeck, "Sensor/actuator failure detection in the vista f-16 by multiple model adaptive estimation," *IEEE Transactions on aerospace and electronic systems*, vol. 31, no. 4, pp. 1218–1229, 1995.

[97] D. Willner, C.-B. Chang, and K.-P. Dunn, "Kalman filter configurations for multiple radar systems," MASSACHUSETTS INST OF TECH LEXINGTON LINCOLN LAB, Tech. Rep., 1976.

[98] C. Chui and G. Chen, "Kalman filtering with real time applications," *Applied Optics*, vol. 28, p. 1841, 1989.

[99] M. Hossain, "Whitening and coloring transforms for multivariate gaussian random variables," *Project Rhea*, 2016.

[100] A. Koivunen and A. Kostinski, "The feasibility of data whitening to improve performance of weather radar," *Journal of Applied Meteorology*, vol. 38, no. 6, pp. 741–749, 1999.

[101] E. A. Wan and R. Van Der Merwe, "The unscented kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*. Ieee, 2000, pp. 153–158.

[102] S. Yang, A. Bryant, P. Mawby, D. Xiang, L. Ran, and P. Tavner, "An industry-based survey of reliability in power electronic converters," *IEEE transactions on Industry Applications*, vol. 47, no. 3, pp. 1441–1451, 2011.

[103] R. W. Erickson and D. Maksimovic, *Fundamentals of power electronics*. Springer Science & Business Media, 2007.

[104] S. J. Julier and J. K. Uhlmann, "New extension of the kalman filter to nonlinear systems," in *Signal processing, sensor fusion, and target recognition VI*, vol. 3068. International Society for Optics and Photonics, 1997, pp. 182–193.

[105] I. Punčochář and J. Škach, "A survey of active fault diagnosis methods," *IFAC-PapersOnLine*, vol. 51, no. 24, pp. 1091–1098, 2018.

[106] A. Atkinson, A. Donev, R. Tobias *et al.*, *Optimum experimental designs, with SAS*. Oxford University Press, 2007, vol. 34.

[107] M. B. Zarrop and Z. MB, "Optimal experiment design for dynamic system identification," 1979.

[108] N. M. Filatov and H. Unbehauen, *Adaptive dual control: Theory and applications*. Springer Science & Business Media, 2004, vol. 302.

[109] X. J. Zhang, *Auxiliary signal design in fault detection and diagnosis.* Springer, 1989, vol. 134.

[110] S. L. Campbell and R. Nikoukhah, *Auxiliary signal design for failure detection.* Princeton University Press, 2015, vol. 54.

[111] P. A. Gagniuc, *Markov chains: from theory to implementation and experimentation.* John Wiley & Sons, 2017.