Distinguishing Sensor and System Faults for Diagnostics and Monitoring

by

Morteza Taiebat

B.Sc., Mechanical & Automotive Engineering,

Iran University of Science and Technology, Tehran, Iran 2013

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

The Faculty of Graduate and Postdoctoral Studies
(Mechanical Engineering)

The University of British Columbia

(Vancouver)

August 2015

Abstract

Automated FDD systems depend entirely on reliability of sensor readings, since they are the monitoring interface of the system. With an unexpected variation in a sensor's reading from its anticipated values, the challenge is to determine if it is symptom of a fault in the sensor or the monitored system. The ability to identify the source of faults is crucial in the monitoring of a system, as different corrective actions are required in case of sensor or system faults.

To address this issue, first, it is clarified that by strict duplication of sensor elements, it is feasible to differentiate between sensor and system faults. However, duplication is not always practical. Hence, by aiming to identify the minimum degree of sensor redundancy, *a priori* knowledge of physical relationships (functional redundancy) between monitored variables is used to check the credibility of existing sensor observations via Analytical Computational Substitutions (ACS). In the proposed methodology for a certain class of systems, the system variables are modeled with serially connected causal network. Then the concept of Moving Monitoring Window (MMW) is introduced, which covers three nodes at the same time, as it traverses through the nodes of the system in the direction of causality. The Logic Set Unit consists of all system/sensor state possibilities called *System Behavioral Modes*, which allows decision-making on the health status of sensor or system or a combination. The generalization by deduction reveals that if the number of sensors is greater than 1.5 times of the number of monitored variables, the task of distinguishing between sensor and system faults can be done, as long as serial causality is valid between the monitored variables. Removing any more sensors from this configuration leads to inability to locate the faults, due to the lack of adequate behavioral modes for diagnosis decision. The effectiveness of the approach is verified on a system of interconnected multi reservoirs and control valves.

Preface

This thesis entitled **"Distinguishing sensor and system faults for diagnostics and monitoring"** presents the research conducted by Morteza Taiebat, based on the initial research question by the supervisor, Prof. Farrokh Sassani. The proposed methodology in this manuscript is original, unpublished, independent work by the author.

This research was carried out with funding from the Natural Sciences and Engineering Research Council of Canada (NSERC).

Table of Contents

List of Tables

List of Figures

Glossary

ACS	Analytical Computational Substitution
AI	Artificial Intelligence
ANN	Artificial Neural Networks
BBN	Bayesian Belief Networks
CBM	Condition Based Monitoring
DAG	Directed Acyclic Graph
DD	Detection Delay
EKF	Extended Kalman Filter
ES	Estimated Substitution
FAR	False Alarm Rate
FD	Fault Detection
FDD	Fault Detection and Diagnosis
FDI	Fault Detection and Isolation
FDII	Fault Detection, Isolation and Identification
IFAC	International Federation of Automatic Control
KF	Kalman Filter
LTI	Linear Time Invariant
LVDT	Linear Variable Differential Transformer
MAR	Missed Alarm Rate
MMW	Moving Monitoring Window
PCA	Principal Component Analysis
RTD	Resistance Temperature Detector
UIO	Unknown Input Observers
UKF	Unscented Kalman Filter

Acknowledgements

I would like to covey my deepest and most sincere appreciation to my supervisor, Professor Farrokh Sassani, for his inspiration, friendship, patience, continuous encouragement, and unconditional support throughout this endeavor. I was extremely privileged for having the opportunity to learn from him, and also the freedom to explore energy and environmental fields, which made me enthusiastic of research. I feel very lucky to have him not only as my academic mentor, but also as a role model in morality. He made my Master's an invaluable experience, which will benefit my whole life. I am indebted to him forever.

I am grateful to my brilliant colleagues in the Process Automation and Robotics Laboratory: Dr. Behnam Razavi, Dr. Atefeh Einafashar and Mr. Abbas Hosseini. Their friendship and support created a pleasant motivating environment during the period of my work.

I wish to express my genuine gratitude to my wonderful family for their kindness, support, and unwavering encouragement, especially to my kind mother for her infinite love. My sincere thanks are extended to my brothers, who have enlightened my path and motivated me for keeping forward: Mahdi who is a durable source of generosity and support in every step of my life, and Mojtaba, who is always a friend when I need someone to talk. Lastly, I am grateful to the one who is in my mind and heart, even from far away.

Chapter 1

Introduction to Fault Diagnostics

1.1 Introduction

A rising demand exists for dynamic systems to run autonomously in the occurrence of faults and failures in sensors, actuators, and plant components. Fault detection and diagnosis is a key element of operation and management of automated systems. Without some method of online fault diagnosis, the safety and autonomy cannot be fulfilled. If measurement data is to be used online, then this is particularly true for feedback control [1]. The faults may instantly result in instability of the controlled system. There is a high demand for the development of diagnostic systems that are capable of autonomous detection of presence of anomaly and localization of the faults that may occur in different components of a complex dynamic system while in operation. A monitoring system receives information about the system through sensors, and makes it available to the controller. The diagnostics system uses sensor readings to assess the state of the

system, detect abnormal states, and identify the root cause of the abnormal state in order to advise the operator about corrective actions to prevent substantial damage to the system.

1.2 Preliminary Remarks

1.2.1 Definition of Fault and Failure

According to the International Federation of Automatic Control (IFAC) SAFEPROCESS technical committee [2], a fault is defined as *"an unpermitted deviation of at least one characteristic property or parameter or response of a system from its normal or standard condition which may lead to the inability to fulfill the intended purpose"*. Practically, it is assumed that the system is healthy and no fault is present at the beginning but takes place sometime, with magnitude, type and time of occurrence being unknown [3]. It causes degradation in the system's performance but may not result in complete loss of system functionality. In this sense, faults are segregated from failures, where a failure is defined as a permanent interruption of the system's ability to perform the required functions under desired operating conditions. A symptom is an observed event or variable value, needed to detect and isolate faults [4].

1.2.2 Locations of Fault Occurrence

Representation of the system in block diagram and possible locations for occurrence of faults is shown in Figure 1.1.

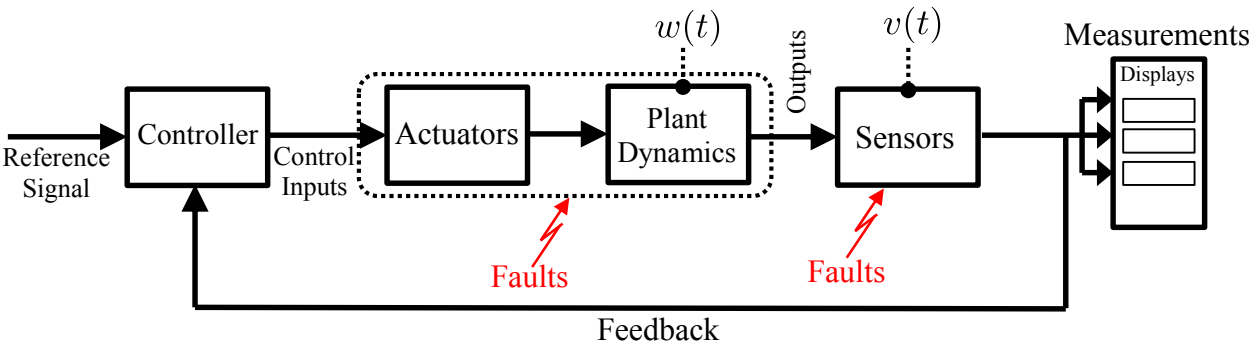


Figure 1.1: Classification of faults according to their location as sensor and system faults

The classification and mathematical representation of faults are explained in detail in chapter 2. Modern controllers are usually programs written in software such as MATLAB and LabVIEW, which are essentially reliable and robust in continuous operation with negligible probability of failure. Errors that may take place in the software are usually handled by completely different error detection and accommodation techniques which are mostly developed by computer science researchers [1] and are not within the scope of this research.

1.2.3 Fault Detection and Diagnosis (FDD)

The rapid developments of diagnostic techniques as well as the diversity of their applications have made them the focal point of interests in the field of reliability and design of fault tolerant control systems in recent years. From 1970's, significant research has been conducted in this field and for different applications, hence, fault diagnosis and fault tolerant control are well established. For instance, fault diagnosis is a standard component in automotive engines. In the aerospace industry on-board diagnostic algorithms reduce and accommodate many failures and malfunctions that can occur in the jet engines. In other industries, fault prognosis, diagnosis and fault accommodation are technologies that gradually enter into the new

design paradigm. This field is also active in the academic community, where highly mathematical methods are being developed to meet the challenging industrial needs.

In general, autonomous online health monitoring and fault diagnosis is essential for safety-critical systems. Hence, accurate assessment of faults allows system operators to either plan for a maintenance service for the faulty component, to switch to the redundant component if maintenance is not possible, or intelligently execute preventative actions in advance, in order to avoid catastrophic consequences.

The simplest diagnosis is to identify whether the system is able to operate or not. In higher levels, the first phase is the detection of presence of a fault. It is followed by recognizing the abnormal states of the system. In this case, the goal is to identify elements and sub-systems, which are degraded or non-operational and need to be reconfigured, repaired or replaced.

Three phases of state examination are defined by [1], [2], [5] These are the detection, isolation and identification of faults, as follows:

- Fault Detection: making a binary decision on the presence of a fault and determining of the instance of its detection/occurrence;
- Fault Isolation: determining the location of a detected fault, which consists of the type and place of the appearance of the fault; it follows the detection;
- Fault Identification: determination of the fault size (magnitude), severity and its changeability in time, once the fault is detected and isolated.

A fault diagnosis system, depending on its performance, is designated as FD (fault detection), FDI (fault detection and isolation) or FDII (fault detection, isolation and identification) [1]. The most commonly used terminology, however, is FDD (fault detection and diagnosis) [6].

Figure 1.2 illustrates the genral framework of a diagnostic structure for a feedback controlled system.

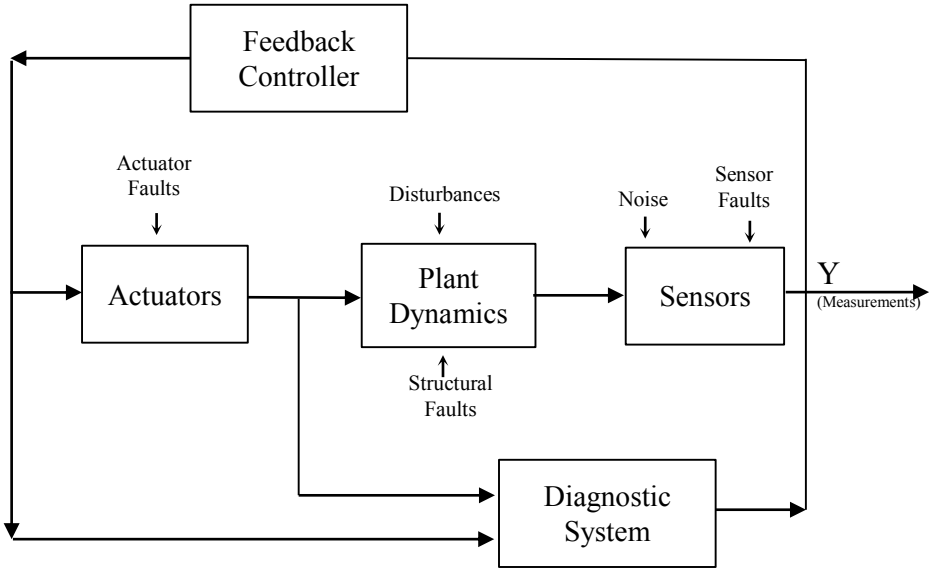


Figure 1.2: A general framework of diagnostic system, reproduced based on [7]

A group of concepts that are accepted by the SAFEPROCESS Committee [2] includes:

- Monitoring: is a task with the goal to collect and transfer data for the system variables; this task is performed in real time,
- Supervision: consists of monitoring and recognizing incorrect behavior, as well as making decisions, which help to ensure the correct operation of the system in the case of occurrence of a fault [4],
- Protection and reconfiguration: includes all operations and technical means that either eliminate a potentially disruptive event or prevent the consequences of it.

In general, supervision is defined as either a continuous or discrete observation of states, and sometimes diagnosis of a system [8].

Testing is the next action connected with diagnostics. This operation is understood to be a determined set of tests that are conducted in order to identify whether the values of useful properties of the system are within the range of determined parameters.

1.2.4 Aims of Diagnostics

An online diagnostic system is able to autonomously detect the presence, isolate the location, and identify the type and severity of faults present and occurring in different components of a complex dynamic system while the system is in operation [1], [9].

The consequences of faults can be extremely serious in terms of human fatalities, environmental impact, and economic loss. Furthermore, the increasing demand for safer, secure, and reliable operation of systems has essentially made FDD process extremely important. In other words, there is a growing need for the so-called autonomous fault-tolerant systems that are able to continue to operate reliably in presence of faults and failures in sensors, actuators, and components until they are attended to.

1.2.5 Alarm Management System

Once a fault is detected, an alarm signal should be triggered to notify the operator about the fault. “The processes and practices for determining, documenting, designing, operating, monitoring and maintaining alarm systems” are known as alarm management [10], [11].

The performance of fault diagnosis and alarm management systems is important in many applications and can be measured using three main principles: false alarm rate, missed alarm rate and detection delay [12]. The False Alarm Rate (FAR) refers to the probability of accepting the alarm in the normal condition while it is false. On the other hand, the Missed Alarm Rate (MAR)

refers to the probability of rejecting the alarm in the abnormal condition while it is true. Table 1.1 depicts the relationship between the FAR and MAR. The average time that the system requires to raise an alarm after an incident occurs is termed the detection delay (DD). In order to avoid performance degradation and have a robust diagnosis system, the above criteria should be considered in design of FDD system.

Table 1.1: Confusion matrix of the types of errors in decision-making

Decision		Fault Occurrence	
		Yes	No
Alarm	Yes	Correct Alarm	False Alarm
	No	Missed Alarm	Correct Alarm

1.3 An Overview of Different Approaches for FDD

There are numerous approaches to fault detection and diagnosis. Because each has its strengths and weaknesses [13], most practical applications combine multiple approaches (hybrid methods) [1]. In this section, we introduce a review of the relevant literature.

Figure 1.3 shows the general classification of fault detection, isolation and diagnosis techniques [5]–[7], [9], [14]–[17].

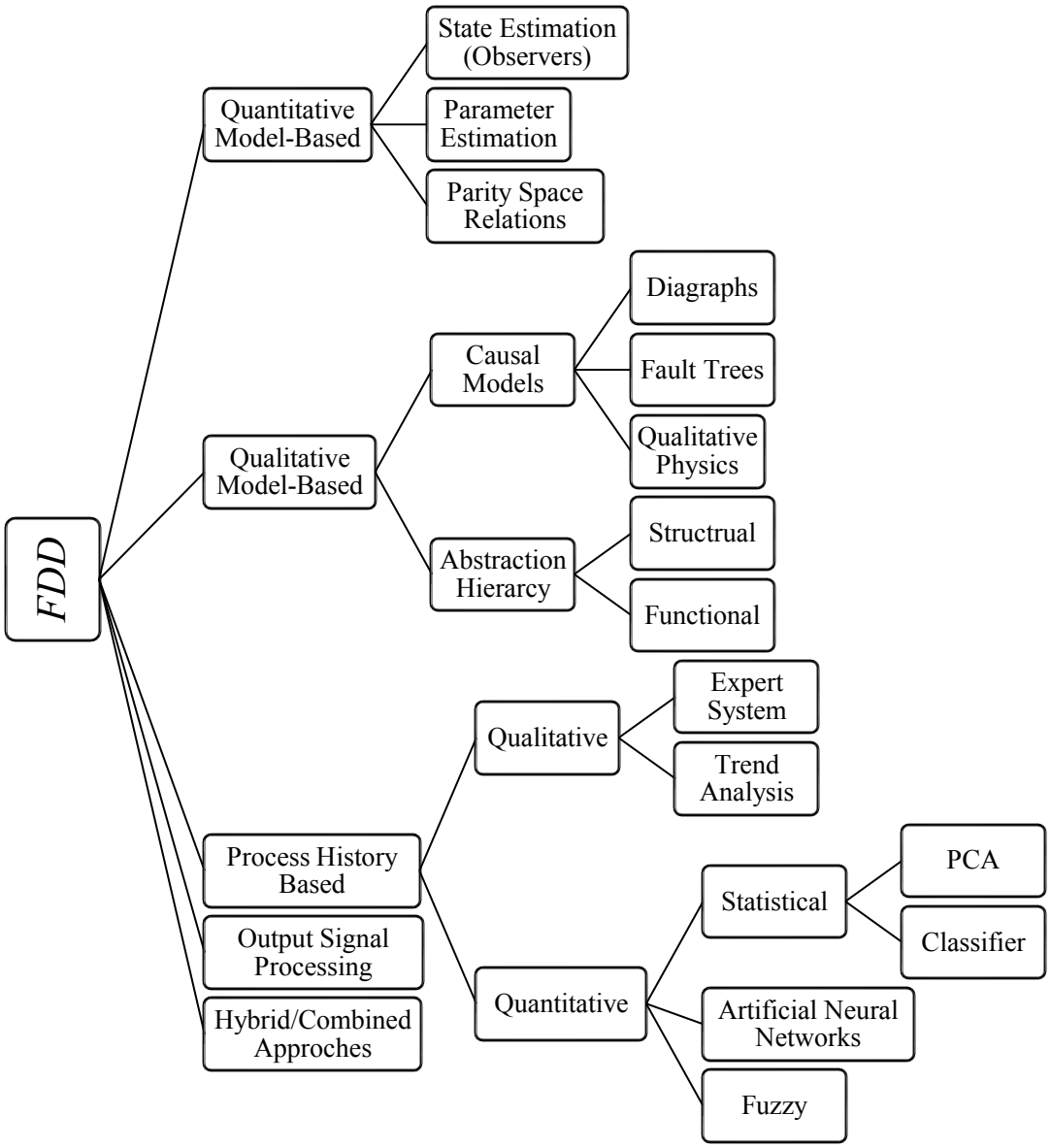


Figure 1.3: Classification fault detection and diagnosis systems

Methodologies for fault diagnosis in dynamic systems are well established. Many pioneering studies have been carried out in this field (Chen and Patton, 1999 [18]; Gertler, 1992 [19]; 1997 [20]; 1998 [21]; Isermann, 1984 [3]; Frank, 1990 [6]; Willsky, 1976 [22]; Massoumnia, 1986 [23]). For instance, Isermann [3] has provided one the best surveys on modeling and estimation

methods for FDI from his works in this field. PhD thesis of M. A. Massoumnia at MIT [23] is a founding research in the geometric FDI approaches in linear systems. Various reviews and contributions of P. M. Frank in analytical model based approaches are notable as well. The robustness issues in fault detection, optimized generation of residuals and residuals generation of non-linear systems have been addressed in a comprehensive manner, in these references [21]–[23]

Most of the existing FDD approaches can be divided into computational intelligence-based and model-based methods [1], [6]. The former approach employs quantitative historical data or qualitative information on the system. While in the latter, the mathematical model of the system is being utilized as an *a priori* source of information on the monitored system. Due to the accuracy and wide use of model-based techniques, this thesis is focused on these approaches.

1.3.1 Model-Based FDD and Residuals

In the area of automatic control, the most powerful, reliable and accurate diagnostic approaches are model-based techniques. All model-based FDD methods require two steps by relying on an explicit model of the monitored system. The first step generates inconsistencies between the actual and expected behavior. Such inconsistencies are referred to as *residuals* that reflect the existence of fault in the system and may contain information about the fault. Residual is basically a fault indicator, based on a deviation between a model-equation-based computations and measurements from sensors [1]. The second step is making a decision on the course of action for dealing with the detected fault.

A wide range of design methodologies can be used for creating model-based diagnostic systems. These methods are fundamentally developed branches of control theory, such as the

modeling and identification of dynamic systems, including state estimation [5]. For instance, techniques based on consistency modeling (describing direct relations of the consistency, or parity, of measurement data), *diagnostic state observers*, and *Kalman filters* are fundamental methods of designing *residual generators*.

Diagnostic decisions for the monitored dynamic process are derived from the results of the fault detection procedure, which evaluates residuals, which are suitably generated on the basis of the errors of the estimates of plant output signals [20]. The estimation is performed by "a rationally chosen mathematical model," which is called analytical redundancy [3], [16].

1.3.2 Analytical Redundancy and Residual Generation

The general concept of using analytical redundancy in the diagnostic systems is given in Figure 1.4.

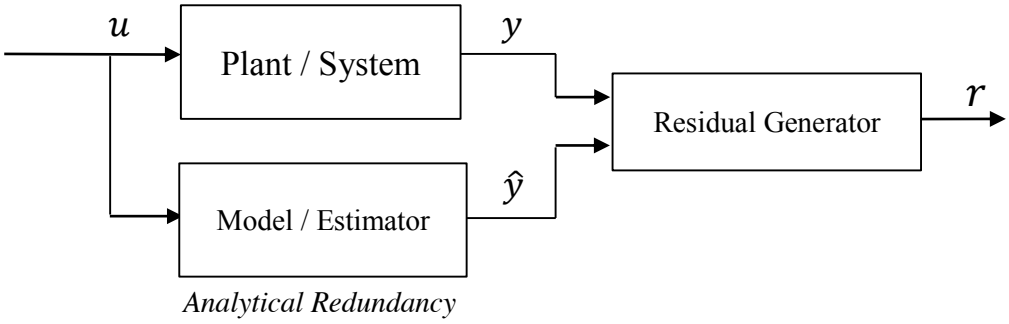


Figure 1.4: Schematic view of analytical redundancy and residual generation

$$r = y - \hat{y} \tag{1.1}$$

where r is residual vector, y is system outputs and $\hat{y}$ is fault free response or estimate from the analytical redundancy.

The essence of analytical redundancy in fault diagnosis is to check the actual system behavior against the system model for consistency. Any inconsistency expressed as residuals, can be used for detection and also isolation purposes accompanied by one of two extremely important assumptions, which are described in the section 1.4. As mentioned earlier, an explicit mathematical model of the system is required for generation of the residuals.

Theoretically, under ideal steady state condition in dynamical systems, the residuals should be zero when no fault occurs. However, there is a fundamental practical limitation in that the system model and the analytical redundancy on which the system is never known precisely [3]. The consequence is that the generated residuals may be nonzero due to:

- Measurement, process and external interference noise,
- Model inaccuracies (un-modeled dynamics),
- Gross errors, precision and resolution issues in sensors and actuators [7],
- Disturbance in the plant, and
- Other sources of uncertainty [6], [26].

As a result certain threshold bounds around zero should be defined. If the residual breaches these thresholds, it is interpreted as a strong indication of the occurrence of fault(s). The weakness of this approach is that thresholds reduce the sensitivity of the diagnostic system to faults. They may also depend on the magnitude and nature of the disturbances, which should be differently handled from the faults. Choosing an excessively low threshold increases FAR. On the other hand, selecting a high value increases MAR as well as the time to fault detection [26]. It is an important and practical issue to have a safe balance between FAR and MAR. Investigators have studied many schemes to increase the robustness of the decision making process [18], [27], [28] This issue is further discussed in chapter 4.

1.3.3 Isolability in Diagnostic System

This is the capability of a diagnostic system to isolate different faults or fault modes from each other. Isolability of a particular fault from other faults depends on the way the faults affect the system output (i.e., fault observability). Presence of uncertainty sources adds challenge to achieve a high degree of isolability. It means that a diagnostic system with a high degree of isolability may be too sensitive to the uncertainties.

For the detection of a fault, a single residual set is enough. However, as set of residuals is required to achieve model-based fault isolation, based on one of two well-known frameworks [1], [18], [29]:

1. Structured residual,
2. Directional residual.

The main idea behind structured residual approach is to design a bank of residuals, in which some of the residuals are designed to be sensitive to a certain group of faults, while remain absolutely insensitive to others. The structured residual approach can be designed in two different ways: dedicated residual scheme and general residual scheme [6], [26], [29], [30]. On the other hand, for directional residuals, the basis is to design a fault detection filter, in which the residual vector receives specific directions, depending on the fault that is acting upon the system [23], [31]. Based on the nature of faults, the task of fault isolation can be carried out successfully.

1.4 Fault Localization

Identifying the root cause of a fault from other potential fault sources or to locate a faulty sensor or sub-system among various components of a system is absolutely necessary for Condition Based Monitoring (CBM). These components encompass sensors or sub-system. Localization capability is also fundamental of obtaining fault tolerance [32], [33], since proper analytical substitutions or counter-measures cannot be triggered without knowing the source of the malfunction in the system. However, fault localization is the most difficult feature to achieve in any FDD system. The focus this thesis is on this issue.

After successful detection of a fault and its remedy, it is essential to be able to localize it in order to precisely identify the source of the fault within the monitored system. All current studies and methodologies of locating the fault make either of following assumptions:

- Sensors are fully functional and healthy, consequently we can diagnose system faults,
- System is fully healthy and diagnosis is performed for sensor faults.

Although isolation of faults from each other may provide the ground for localization, there is a fundamental difference between these two definitions. In isolation process the fault of interest is known and attributable to a certain source. While in for general, the fault of a sensor or sub-system is unknown and stochastic.

1.5 Difficulty in Distinguishing between Sensor and System Faults

In some cases, the sensors may not be highly reliable compared to the system components that they are monitoring or vice versa. Consequently, the indication of a fault might be the result of a sensor malfunction or a system anomaly. This is a crucial question in localizing the detected

fault. The diagnostic system has to distinguish between the system faults and sensor faults, which might be the most sophisticated problem, since these faults might have same symptom and mask each other. The ability to differentiate system faults and sensor malfunctions is essential in the monitoring of a system, as different compensatory responses are required.

Having this introduction in mind, to the best of author's knowledge and literature surveyed, presently there appears to be no way of knowing which of these faults causes the triggering of the alarm, using the current fault diagnosis schemes. In this study we have investigated this problem and propose a methodology to address it for a certain class of systems.

Chapter 2

Difficulty with Distinction of Sensor and System Faults

2.1 Introduction

In chapter 1, an overview of fault detection and diagnosis is presented. In this chapter, the focus is on problem statement, explanation of details and challenges as well as literature review.

Sensors are considered as the monitoring interface for a dynamic system, since the measurement data is the only source of information about the system and sensor. A sensor is aimed to generate credible measurement data, which provides an estimate of a variable or parameter. In practice, the sensor measurement is not a perfect representation of the parameter: the effects of the sensor, including faults and noise are also present in the measurement signal. Many investigations have been carried out in the fault detection domain, which attempt to extract sensor, actuator and plant fault from measurement data; however, there is no framework to

differentiate between them. The inherent difficulties of distinguishing these faults from a single measurement are obvious, as the sensor is the only way of communicating with system.

2.1.1 An Example of Wood Drying Kiln

Consider an industrial kiln for the drying of batches of wood. The monitoring parameter is temperature, and the control room is away from the kiln. Indeed, it is the actual temperature that makes the wood dry or over-dry, while the observed temperature is measurement data from the sensors. Therefore, if the sensed temperature is high, it cannot be necessarily concluded that the woods will over-dry. The real temperature might be high, or the temperature sensor might be reading high (biased), or even stuck at a particular value.

On the other hand, if we end up with over-dry woods, and see that the temperature appears in normal range, it cannot be conclusively ruled out that the actual temperature was high, because the temperature might be reading low or frozen at low. Possible root causes of the over-dry woods include various sensor failure modes, sudden jumps (in contrast to average temperature), incorrect set-point temperature by operator, temperature controller malfunction, and various uncertainties associated with physical properties of wood such moisture and texture.

2.2 Sensor Health and Fault

Sensors are basically the output interface of a system to the external world, and convey information about a system's behavior and its internal states. The primary goal of using sensors in any system is to provide real-time feedback on the variables for control purposes as well as enabling the system to successfully perform its tasks. As an instance, in order to achieve the desired positioning accuracy and high precision manufacturing tasks, a joint position sensor is

essential for a robotic arm. They also facilitate monitoring of variations in parameters by providing reliable and accurate data to update, estimate and monitor system health status. To fulfill these objectives, there is an implicit assumption that all sensors operate consistently at their design specifications.

When a sensor produces an output measurement signal proportional to the input stimulus, within an acceptable amount of deviation as dictated by the sensor physics, resolution, accuracy, application requirements, etc., it is considered 'healthy'. This deviation is called 'noise'. However, the effects of faults are manifested as *undesirable deviations* in the sensor output such as drift, bias, loss of effectiveness, full failure, etc. Such phenomena may occur intermittently or steadily over a period indicating the development of gradual sensor degradation. In the extreme case, there may be a complete loss of information from a sensor due to an abrupt failure of the sensing element or power/signal transmission lines, connectors, and faults in the onboard signal processing circuits [34]. In the cases where the readings are used for control purposes, it may lead to undesirable system behavior, if the sensor measurement readings become faulty, unavailable or invalid. Hence we can conclude that the presence of faults in sensors may deteriorate state estimates and consequently result in inefficient and/or inaccurate control.

Furthermore, using data from faulty sensors may lead to false alarms and missed alarms of actual system faults. Therefore, sensor faults may cause substantial performance degradation of decision-making systems or processes that relies on the integrity of measurements or observation from system. Having this, the sensors health status should be taken into account, in deciding whether to use the data from a particular sensor or not.

In the next chapter, a methodology is proposed, which is capable to decide on this issue and utilize either sensor reading or 'Analytical Computational Substitution (ACS)'. Consequently, it has an ability to tolerate some sensor faults.

2.2.1 Sensor Faults and Malfunctions Classification

Sensor outputs are designed based on specific changes in the physical/ electrical properties of their sensing elements. As stated already, in normal conditions, it has an inherent proportionality to a physical input. For instance, a change in the resistance of strain gages in response to load, gives an output voltage. However, any change in the characteristics of the sensing element (due to wear, tear, aging, etc.) may cause the sensor readings to deviate from their nominal values. In addition, despite many advances in technology of design and manufacturing of sensors, many of them still remain vulnerable to drastic changes in the operating environment, such as excessive temperatures, exposure to magnetic fields, radiation, etc. Hence in literature, they are referred as weak link in system components [34].

The widely used sensors in automotive, manufacturing, aerospace and other large industries includes but not limited to: Thermocouples, Resistance Temperature Detectors (RTDs), Piezoresistive Sensors, Resistive Strain Gage and Linear Variable Differential Transformer (LVDT). A review of some common faults in these sensors is surveyed in [34]–[36]. We can classify different behavioral categories of sensor faults as the following:

Bias: A constant offset from the nominal sensor signal. Bias can occur due to incorrect calibration or physical changes in the sensor system. The governing equation is $Y_{i_f} = X_i + b_i$ where b_i is the constant offset value. A time varying b_i leads to drift.

Drift: A time varying offset from the nominal values of the sensor signal. Generally, only linear drifts have been modeled in the literature [1]. However, a nonlinear drift may be possible. Drift failures may be represented $Y_{i_f} = X_i + \delta(t)$, where is the time-varying offset factor. A form of $Y_{i_f} = X_i + b_i(t)$ represents linear drift.

Loss of Effectiveness: In this failure, which is also known as floating signal, the signal magnitudes are scaled by a factor of $a(t)$. It can be represented by $Y_{i_f} = a(t) * X_i$, where $0 < a(t) < \infty$ is a scaling constant that may be time-varying. The other form of showing loss of effectiveness is $Y_{i_f} = X_i + b_i(t)$, where $b_i(t)$ is accuracy coefficient such that $b_i \in [-\bar{b}_i\,, \bar{b}_i]$, where $\bar{b}_i > 0$ is an arbitrary number. It is highly cumbersome to deal with this type of fault.

Hard Failure: The sensor output is stuck at a particular level expressed by $Y_{i_f} = X_i(t_{F_i})$, where t_{F_i} is probable time of failure. In general there are two subcategories for hard failures.

- **Loss of Signal:** represents the complete loss of sensor data where the output from the sensor is zero.
- **Sensor Freezing:** represents the situation where sensor output is stuck at a constant value.

Summarized we can all categorize faults as follows:

$$Y_i(\mathrm{t}) = \begin{cases} X_i(t) & & \forall\ t \geq t_0 & \text{No faults} \\ X_i(t) + b_i & b_i = 0,\quad b_i(t_{F_i}) \neq 0 & & \text{Bias} \\ X_i(t) + b_i(t) & |b_i(t)| = ct,\ \ 0 < c \ll 1 & \forall\ t \geq t_{F_i} & \text{Drift} \\ X_i(t) + b_i(t) & b_i \in [-\bar{b}_i\,, \bar{b}_i],\quad \bar{b}_i > 0 & \forall\ t \geq t_{F_i} & \text{Loss of Effectiveness} \\ X_i(t_{Fi}) & & \forall\ t \geq t_{F_i} & \text{Failure or Freezing} \end{cases} \tag{2.1}$$

Finally, we can represent the above cases, with the following generic mathematical model:

$$\mathrm{Y} = A_m\mathrm{X} + \mathrm{B} \tag{2.2}$$

where A_m is a positive definite diagonal matrix and its elements are within $[0, \infty)$, or slowly varying in this range, and elements of vector B fall or slowly vary in range of $[-\bar{b}_i, \bar{b}_i]$. Figure 2.1 schematically illustrates the effect of various sensor faults on measurement signal.

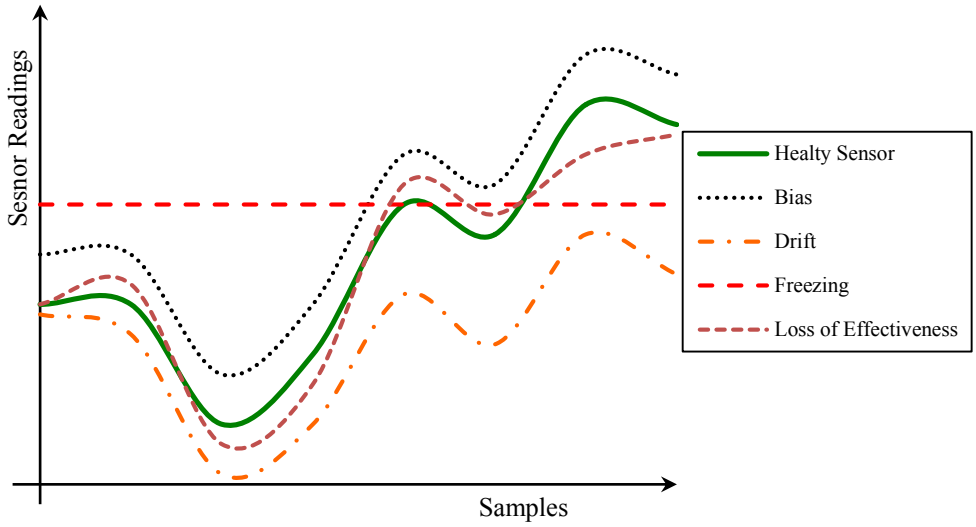


Figure 2.1: A schematic view of the effect of various sensor faults on measurement signal

2.3 System Health and Faults

An unanticipated deviation in a sensor's readings from its expected values under specific operating conditions may not necessarily be due to a fault in the sensor itself, but may be indicative of a more serious potential fault in the monitored system. In this research, system faults consist of actuators and plant dynamic components' faults. The aim of most of diagnostic systems is to detect and prevent the development and propagation of these faults. The system faults are usually represented as cases where some condition changes in the system, which make the nominal dynamic equation of the system invalid. System faults are dependent on the sub-systems and actuators being monitored. Some examples include but not limited to power source failures in satellites [8], [37]; lock-in-place or freezing, float, hard-over-failure and loss of effectiveness in electromechanical or electromagnetic actuators such as valves and robotic arms

[38]; leakage in a tank in chemical systems [10], [39]; accumulation of debris and clogging in hydraulic cylinders [40], [41]; propulsion systems or control surface damage faults in aerial vehicles [1]; bearing faults in rotational equipment such as engines; friction faults due to lubricant deterioration; and tooth breakage and crack in gears of a gearbox system . In a rotary equipment such as en engine, common faults consist of bearing faults [42]–[44]; friction faults due to lubricant deterioration; and tooth breakage and crack in gears. A damaged bearing may lead to substantial ripple in the output torque as well as excessive vibrations and acoustic noise. The effect of the flawed bearing might be observed as large spikes in the readings from the accelerometers and the torque sensor mounted on the crankshaft. Although such a fault would be eventually detected by diagnostic system, but a diagnosis would be obtained only after an extended period of monitoring, by that time the fault may have seriously affected the system capability.

System faults may have minor to extremely severe consequences. For example, an unexpected failure of the aircraft engine components may cause significant economic as well as fatal loss [45], [46]. Thus, it is extremely crucial to diagnose these faults at early stages of component degradation in order to avoid catastrophic consequences.

Mathematical representation or modeling of these faults is sometimes very difficult and extensive experimentation may be needed before constructing a model. In general, system faults can be represented by a change in the system's state equation. It is either a parametric change or a structural/functional change.

Without any knowledge of the system status, in some cases the unusual sensor readings may erroneously be interpreted as potential faults in the monitoring sensors. For instance, if limit

checking is used to validate the sensor measurement without extra knowledge of system, the diagnosis might be a sensor fault. One the other hand, sometimes a simple sensor fault might be diagnosed as a system fault and trigger unnecessary actions. Hence it is necessary to make a correct decision for assessment of the system and sensor's health status, relying on the available knowledge of system as well as obtainable measurements from all the sensors in monitoring system. This issue explicitly brings us to the problem statement.

2.4 Problem Statement: Distinguishing Sensor and System Faults

Automated fault detection and diagnosis depends entirely on sensor readings. The usage of the term "sensors" includes process, plant dynamic or actuator monitoring instrumentation for flow, level, pressure, temperature, power, etc. As stated in Section 1.4, the fault detection and diagnosis schemes only consider either the malfunction of system assuming that the sensing system is functioning normally or sensor failure while the system is fault free. In practice, this is not always the case. System fault detection based on sensor observations and measurements is valid only when it is guaranteed that all sensors are working properly and faultless. A system fault should be detected and isolated immediately. On the other hand, sensor faults, which lead to faulty measurements, can be detected and diagnosed, assuming that anomaly and inconsistency is not from the system. If a sensor fault is unobserved and confused with a system fault, resources will be wasted in trying to identify and eliminate system faults which may not really exist. Therefore, in diagnostic and monitoring systems, it is essential to distinguish sensor faults and system faults, in order to make diagnostic and corrective decision for the operation of system.

2.4.1 Motivation and Importance

Basically, distinguishing between sensor malfunction and system fault is a problem of isolation and decision-making upon the exact location of fault occurrence without any of aforementioned assumptions. The essence of this distinction is due to the fact that either has a different corrective action or compensatory response.

- In case of a sensor faults, the sensors can be replaced physically, redundant sensors can be deployed or the measurement can be mathematically compensated temporarily. As an instance, a faulty reading of air speed in an aircraft can be counterweighted by flight crew until the end of flight.

- System faults on the other hand, often requires immediate attention, which might range from a simple diagnostic alarm to notify the operator, to the severe cases, where full shut down of the operation is inevitable, as soon as it becomes safely possible. For example, a leakage in an aircraft engine, need immediate action, such as emergency landing before it leads to a fatal consequence.

2.4.2 Fault Masking

Both the system faults and the sensor faults may have similar symptoms in sensor observations or mask each other in the worst case. Consequence of similar symptom or masking in these two faults leads to inability to differentiate between them.

2.4.3 Issues with Feedback Control

Feedback control adds to the complexity of diagnosis by masking measurement deviations that might indicate a fault, and by making it difficult to distinguish sensor and system faults. For

example, Figure 2.2 shows serially connected multi reservoirs (tanks) with valves and sensor to monitor the height of liquid. Feedback controllers regulate the heights of liquid.

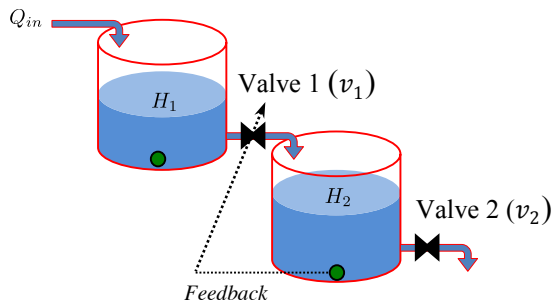


Figure 2.2: A view of connected reservoirs with feedback controller to regulate the height of liquid

Feedback control might compensate deviations that can indicate a fault either in the system or the sensor, which adds difficulty to distinguish between sensors, tank leakage and valve failure. Without the benefit of knowing the nature of the fault and the actual flow rate, it is only possible to construct a number of hypotheses to explain a momentary variation in expected sensor readings:

- Plant fault or disturbance: A variation in the dynamics of system (such as a rise in liquid density) or external flow (such as rain) changes the flow rate, and the controller responds by modifying the flow rate back towards the set point.
- Valve fault: A fault in the valve (such as leakage) results in a change in the input flow rate of second tank without a corresponding adjustment in the control signal. The sensor feedback urges the controller to correct the flow.
- Sensor fault: A fault in the sensor introduces a deviation into the measurement, the controller responds to this deviation by modifying the flow rate. For example a bias in sensor reading causes the true mass flow to be maintained at a level below the set point.

Furthermore, a fault may occur elsewhere in the plant upstream of the valve and sensor, such as a leakage in the tank body. Similarly, after the initial variation measured by sensor, the reported average liquid height returns to the set point, making the fault diagnosis cumbersome. Several conclusions can be drawn from this example:

- Feedback control makes it difficult to distinguish between sensor, actuator and plant faults.
- The effects of sensor faults are particularly serious, as feedback acts to compensate any measurement deviation. Thus after any initial fault, the measurement may appear normal.
- Since there may be a limited time window for observing the effects of a sensor or system faults, the diagnosis time and corrective action should be triggered in the shortest possible time.

2.5 Literature Review

As discussed in Section 1.3, there is an abundance of literature on fault detection and diagnosis for both sensor and system individually. Similarly, many researchers have investigated the concept of distinguishing between disturbances and faults using hardware redundancy in chemical process [47] or analytical methods in stochastic environments. The idea of using dedicated observers is frequently used for detecting individual predefined faults. In these studies a set of residuals is designed for **isolating** faults [31], [32], [48], [49]. The adaptive nonlinear design of observers is studied by Wang H. et al. in [50]. Because the design of residuals does not consider the two types of faults in a unified framework, a differentiation between system and sensor faults cannot be achieved. Hayes et al in 2008 [51], Hajiyev et al. in 2000 [52] and Xue et al. in 2007 [53] and used fault specific threshold selection to achieve isolation for some known sensor/actuator faults by Robust Kalman filtering and statistical analysis of innovation sequence.

Krysander M. et al. in 2005 [54] and 2008 [55] and Rosich A. in 2012 [56] proposed the sensor placement algorithm for detectability and isolability of different known faults based on structural models. Bhushan and Rengaswamy addressed the problem of sensor location assignment for optimal fault observability based on graph theoretic approaches [57].

On the other hand, the issue of distinguishing between sensor and system faults does not appear to have received a significant amount of prior attention in the monitoring and diagnosis literature. Therefore, it has remained a principal challenge for further successful practical application of monitoring and FDD systems, corrective decision-making and fault tolerance.

The existence of this issue has been acknowledged in a very large number of publications, however only a few has tried to tackle this problem. Amongst those, there are some conceptual studies for self-validating components as well as reports on unsuccessful application of diagnosis methods on further processing of detection results for differentiation between sensor and system malfunctions.

The ideas of self-validating sensor and then self-validating actuator were introduced by Henry and Clarke in 1993 [58], and Yang and Clark in 1999 [59], respectively. They describe the concept of online diagnostics and uncertainty handling in sensor and actuator, based on a reference in a Coriolis flow meter and a generic actuator. Although this matter is mathematically proven, it has never come to practice.

Indeed, the issue of reliability for aerospace vehicle is of the highest importance for NASA. Researchers at NASA Ames Research Center have two studies on detection, classification and mitigation of sensor faults using Artificial Neural Network [34] and Expert System [60] on an experimental test rig. While the focus of the study is on detection of sensor faults under noise in

a particular test rig, they have concluded that concurrent deviation of several residuals is a strong indication of system fault. However, the limitation of this method is that it is not general and also the controller can compensate the effects of sensor faults. Besides, if a subsystem is monitored only with one sensor, the deviation of its residual is not conclusive. In another study by Alag et al., [61] a methodology for validating senor readings is presented and verified with empirical data from a gas turbine. The method consist of the estimation of all the parameters of interest, statistical analysis and tracking the values of interest over different time windows and using estimation algorithms i.e. Kalman filtering (for sensors whose outputs can be modeled accurately) to create a validation gate or a region within which the sensor reading is expected to fall. Furthermore, multiple sensor readings are used and correlated in order to verify the performance of sensor. Similar to the pervious case, upon detection of multiple sensor faults, it can be concluded that a system component might be malfunctioning, but distinguishing between sensor and system faults in not addressed in these studies.

Tao and Feng [62] explicitly looked into this problem via data driven and statistical methods. They investigated a hydraulic tank and pressure line with Principal Component Analysis (PCA), and concluded that these faults are indistinguishable without employing hardware redundancy. PhD thesis of Krishnamoorthy [38] at University of Texas has also introduced a framework based on Bayesian Belief Networks to detect and isolate multiple faults. The foundation of its method is probabilistic inference, which allows for incorporating and propagating uncertainties. By updating belief from sensor readings, this framework decides on origin of the fault. However, the shortcoming of this method is that belief updating takes place upon knowing where the fault (discrepancy) is injected. It has also been reported that the method has no efficacy on all edge nodes. Additionally, without redundant knowledge, some information

is produced, which is not clear in the report. The reproduction of this method revealed that the limitations of this framework make it insufficient for distinguishing sensor and system faults.

2.6 Dynamic System Modeling

In Fault diagnosis literature, it is common to show linear time invariant (LTI) dynamic system in form of a continuous or discrete state space model, represented in equations (2.3) and (2.4), respectively.

$$\begin{cases} \dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) + v(t) \\ \boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{D}\boldsymbol{u}(t) + e(t) \end{cases} \tag{2.3}$$

$$\begin{cases} \boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k) + v(k) \\ \boldsymbol{y}(k) = \boldsymbol{C}\boldsymbol{x}(k) + \boldsymbol{D}\boldsymbol{u}(k) + e(k) \end{cases} \tag{2.4}$$

For a system with *n* states, *m* sensors and *r* inputs, $\boldsymbol{x}(t) \in \boldsymbol{R^n}$ is a $n \times 1$ vector of state variables, $\boldsymbol{u}(t)$ is the $r \times 1$ vector of control inputs, $v(t)$ represents additive disturbance, $\boldsymbol{y}(t) \in \boldsymbol{R^m}$ is the $m \times 1$ is output variable vector corresponding to sensor measurement vector, $e(t)$ indicates measurement noise, assumed to be a random vector with zero mean under normal conditions, $\boldsymbol{A}$ is the $n \times n$ state transition matrix, $\boldsymbol{B}$ is the $n \times r$ input matrix, $\boldsymbol{C}$ is the $m \times n$ measurement matrix and $\boldsymbol{D}$ is the feed-forward matrix, which is assumed to be zero. We should notice that for each output variable, there should be a corresponding appropriate sensor for the purpose of measurement. Since dealing with noise and disturbance is studied by many researchers in control engineering area, we omit them in our discussions.

2.6.1 Fault Modeling

System faults appear in the state equation of the system. Assume that an actuator has failed in the system. The fault can be modeled by an additional term in the equation:

$$\begin{cases} \dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{f}_i n \\ \boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t) \end{cases} \tag{2.5}$$

where $\boldsymbol{f}_i \in \boldsymbol{R}^{\boldsymbol{m}}$ is defined as a system fault event vector (which is the column of B associated with the i^{th} actuator), and n is a scalar function which represents the time evolution of the fault.

$$n = \begin{cases} = 0\,, & t < T_f \\ \neq 0\,, & t \geq T_f \end{cases} \tag{2.6}$$

where T_f is time of fault occurrence. Similarly, sensor faults appear in the output equation:

$$\begin{cases} \dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) \\ \boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{f}_s \zeta \end{cases} \tag{2.7}$$

where $\boldsymbol{f}_s \in \boldsymbol{R}^{\boldsymbol{m}}$ is a sensor fault event vector and ζ a scalar function which represents the evolution of the fault. Usually, $\boldsymbol{f}_s$ is a standard unit vector $[0 \ldots 1 \ldots 0]^T$, which represents the i^{th} sensor fault.

2.6.2 Time Characteristics of Faults

Figure 2.3 schematically depicts the time evolution of faults. The variety of fault modes that can occur may be classified as follows:

1. Abrupt (sudden) faults,
2. Incipient (subtle) faults i.e. slowly developing,
3. Intermittent faults i.e. step-like changes

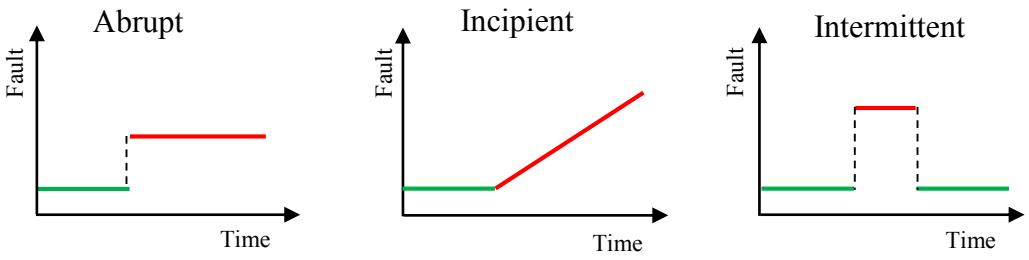


Figure 2.3: Classification of faults according to their time characteristics

An abrupt fault occurs instantaneously, in which the fault term changes sharply from the nominal value to the faulty value. Usually these faults are highly severe, as they affect the performance and stability of the controlled system. Typically, abrupt faults play a role in safety-relevant systems where they have to be detected early enough to prevent catastrophic consequences [1] by reconfiguration or triggering back up components. Abrupt sensor faults can be caused by a failure of the sensing mechanism itself, a power failure, loose or corroded contacts, or the electrical and processing system interpreting the data. These sensor faults might be overcome by mathematical compensation or triggering back up redundant sensors.

An incipient fault represents slow parametric changes, often as a result of aging, where the fault term changes gradually from the nominal value to the faulty value. They are more difficult to detect due to their slowness [61]. Incipient faults are connected with maintenance problems where early detection of worn components is required. An incipient sensor fault, such as a drift or bias in the sensor readings, is caused by deterioration or degradation in the sensing element. In this case the magnitude of faults is small and progressive, which makes the detection time relatively large makes. Probabilistic prognostics methods are established to predict these faults.

A fault, which randomly appears and disappears, is called intermittent fault. The fault term changes from the nominal value to the faulty value and returns back to the normal value in a

short period of time. Intermittent sensor faults may be seen as deviations from normal readings appear and disappear repeatedly from the sensor signal. Due to their random nature, they are the most difficult to track, identify, and account for in diagnostics algorithms. Because of the transient nature and temporary effect of this type of fault, in literature, they are disregarded in some cases, since they might not affect the system [1] unless in very critical cases, such as precise feedback controllers, computer processors, drives and their magnetic sensing elements.

2.6.3 Controllability of System

Controllability itself does not concern with fault detection and distinguishing, and has no bearing on procedure of detection, while is a point of importance for control procedure. However, in a partially uncontrollable system, if the fault occurs in the uncontrollable part, the fault is not detectable [33]. Accordingly, there is a duality between controllability and fault detectability in this sense.

For the purpose of control, the uncontrollable part or subsystem is usually eliminated in model reduction. Therefore, the uncontrollable modes of the original system are no longer present in the reduced model. If a fault occurs in the eliminated (uncontrollable) states, the lost information is needed for detection. Generally, the fault diagnosis studies only consider the faults in controllable part of the system.

2.6.4 Observability of System

Observability has a fundamental role in fault diagnosis. Since the interface of system is sensors, before the design of diagnostic system, the observability should be guaranteed. Essentially, for determining the minimum number of sensors for monitoring the system, the

definition of observability should be considered [54], [55]. In equation (2.5) if the pair $(\boldsymbol{A}, \boldsymbol{C})$ is completely observable, then the fault is asymptotically detectable. The proof and further discussion is given in [9]. The pair $(\boldsymbol{A}, \boldsymbol{C})$ is said to be completely observable if for any $0 < t_1 < \infty$, the intial state $x(0) \in \boldsymbol{R^n}$ can be determined from the time history of output $\boldsymbol{y} : [0, t_1]$ [9]. On a similar basis, let $\{v_1, v_2, \dots, v_n\}$ be eigenvectors of $\boldsymbol{A}$. Then the pair $(\boldsymbol{A}, \boldsymbol{C})$ is observable if and only if:

$$\boldsymbol{C}v_i \neq 0 \quad \forall\, i = 1, 2, \dots, n \tag{2.8}$$

Alternatively, the observability matrix in equation (2.9) has a full column rank that $M_o = n$.

$$M_o = [\boldsymbol{C} \quad \boldsymbol{CA} \quad \boldsymbol{CA}^2 \quad \dots \quad \boldsymbol{CA}^{n-1}]^T \tag{2.9}$$

2.7 Use of Diagnostic Observers or Filters

Many researchers have approached the fault isolation problem by the concept of state estimation, directly starting with single or banks of Luenberger observers, Kalman filters or fault detection filters [6]. While these methods are successful for isolation of a known fault (i.e. the fault type and characteristics are known), they have shortcomings in localization of random faults and distinguishing between sensor and system faults. In this research, we made an effort to solve the aforementioned problem with the diagnostic observer and filter design approach; hence, a summary of it with the challenges are reported in the following.

The basic idea of the observer approach is to reconstruct the outputs of the system by observers {Kalman filters} using estimation error {innovation} from the sensor measurements or subsets of the measurements, which further is used as the residual for detection and then isolation of faults. The observers are used in deterministic case, while Kalman filters are used in stochastic

case, where uncertainty such as noise is taken into account. Basically Kalman filters are observer with optimal estimation gain.

The fundamental of designing state estimator (Luenberger observer) for LTI systems is reviewed here. Consider the LTI system given in equation (2.3), omitting the argument (t). The state estimation $\hat{x}$ and output estimation $\hat{y}$ of a Luenberger observer are governed by equation (2.10):

$$\begin{cases} \dot{\hat{x}} = A\hat{x} + Bu + L(y - \hat{y}) \\ \hat{y} = C\hat{x} \end{cases} \tag{2.10}$$

where *L* is an output injection (feedback) gain matrix and *A-LC* is a Hurwitz matrix [6]. We can define observer estimation errors as $e = x - \hat{x}$ and the observer residual as $r = Ce$. Then:

$$\begin{cases} \dot{e} = (A - LC)e + f_i n \\ r = Ce \end{cases} \tag{2.11}$$

The residual *r* goes to zero or $\hat{y}$ converges to y when there is no fault. Since the pair *(A,C)* is observable, we can select a gain matrix for which $A - LC = \lambda I$, where a scalar $\lambda < 0$ and I is an identity matrix. In equation (2.11), if a fault is present, then:

$$r(t) = Ce^{\lambda t}e(0) + \alpha(t)f_i \tag{2.12}$$

where $\alpha(t) = \int_0^t Ce^{\lambda(t-\tau)}n(\tau)d\tau$ is a scalar value which depends on time t. We conclude that in presence of a system fault, the residual *r* aligns with fault f_i since the first term in right hand side of this equation goes to zero as $t \to \infty$. If we have more than one fault in the system, we can identify and separate faults using two or more residual generators [23]. To avoid false alarms the threshold must be chosen larger than zero. However, this reduces the sensitivity to faults [6]. The procedure of designing optimal estimation gain for Kalman filters for consideration of noise is addressed in [3], [4].

For detection and isolation of sensor faults, Park J. et al. [49] have extended the state and output equations by an additional state,. This auxiliary state describes the behavior of the sensor fault event, which is to be diagnosed. The auxiliary state treats the sensor fault as a system fault. By a similar procedure, we can design an observer sensitive to each fault. Since sensor and system faults are treated the same, they can be considered as multiple faults. To diagnose any number of faults a *bank* of observers can be designed. These multiple faults can be theoretically detected and isolated using the scheme of *Unknown Input Observers (UIO),* by a set of estimation errors (residuals) [6], [22]. Each UIO is assigned to be sensitive a fault and invariant to other predefined faults. This brings maximum isolation for the faults. Figure 2.4 illustrates this scheme.

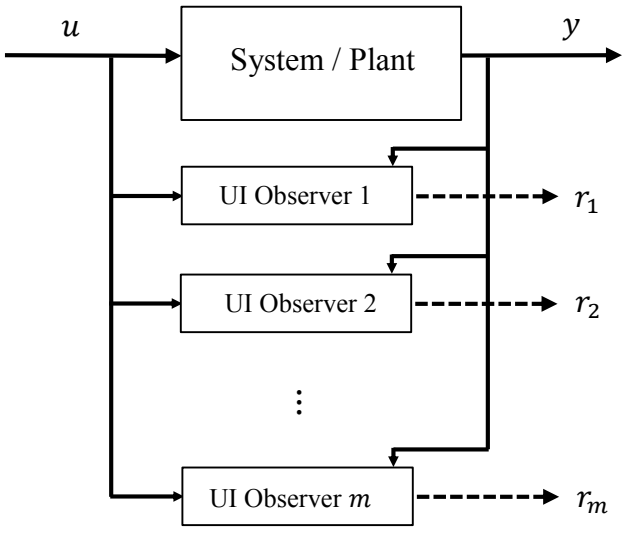


Figure 2.4: Scheme of UIO for isolating multiple faults

Up to this point, we have designed the residuals and fault sensitive observers for isolating known faults. Theoretically, it can isolate any predefined fault, which has been taken in to account at the design phase, however, the residuals are still computed using information of

measurements from sensors. Implementing this scheme reveals that in either situations of system or sensor fault, there are two cases that the residual can breach the threshold. In the first case, a known fault takes place (and this residual or observer is designed to be sensitive to this fault). The other case is that a sensor fault occurs, and the computation of this residual uses the faulty measurement. Practically, a differentiation between system and sensor faults cannot be reached, when the measurements are affected by sensor faults even if they are already taken into account. Hence, using observer (or Kalman filter) scheme is not conclusive on localizing and distinguishing between system and sensor faults.

2.8 Need for Redundant Information

Through the course this research, a number of approaches such as observers and filters (see section 2.7), Bayesian Belief Networks and Neural Networks on different application such as wood drying kiln and hydraulic systems have been investigated, implemented and examined. None of these methods showed a promising solution for distinction of sensor and system faults, since this issue is one step ahead of conventional fault detection and isolation and requires redundant and *a priori* knowledge of the system and its components. Basically, making a decision regarding this issue has an analogy with solving an equation with two unknowns.

Some qualitative database of knowledge rules can narrow down this problem; however, it can be only realized in the extreme cases. For instance, in the example of wood drying kiln that has been exemplified in the section 2.1.1, when the controller set point is on 70 ºC, if a sensor indicates a reading of << 70 ºC or very low temperatures (less than the outside temperature), it would be a strong indication of sensor fault rather than a fault in actuators (heaters) of the system. However, these rules are highly application dependent and require expert's knowledge.

They are also only valid in the extreme cases and are not conclusive during the most of operation cycle of the kiln.

Redundancy is generally defined as *"the repetition or duplication of elements within a system"*. Sensor redundancy refers to multiple sensors measuring the same variable or state within a dynamic system. The physical sensor redundancies provide a high degree of certainty and are also relatively easy to implement. Utilizing sensor redundancy together with some knowledge of system can lead to a decision regarding the distinction of faults. In the next chapter, a framework has been proposed to address this.

Chapter 3

Methodology: Utilizing Sensor Redundancy

3.1 Introduction

In chapter 2, the problem of distinguishing sensor faults from system faults is discussed and explained in details. In this chapter, a methodology is developed to address this issue by utilizing redundant sensors and model-based knowledge of system.

3.2 Causal Networks

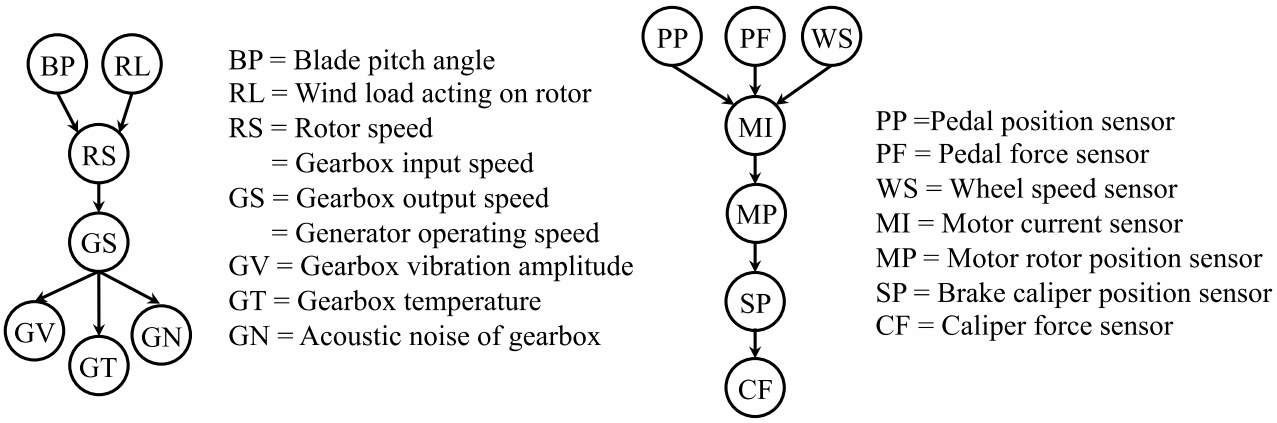
Causal network is a graphical and intuitive model presentation based on physical principles, which can assist users to realize the model. The use of causal networks is common in modeling of real system as well as fault diagnosis and fault propagation [39], [63].

In this study, the causal networks are used to illustrate the monitored variables and their physical relations. In essence, a causal network represents the underlying *first principle* relationships between the different variables that represent the monitored parameters in the system. The graphical framework of causal networks provides an intuitive understanding of the system variables being modeled. A causal network is shown as a graphical structure that consists of a set of nodes that represent the variables related to the physical domain of interest. These nodes are connected by a set of directed links, which explicitly represent the dependencies between the variables. The lack of a link between two nodes clearly represents their independence. The structure is referred to as a Directed Acyclic Graph (DAG) [39]. Based on the definition of a DAG, a directed path from a node to itself should not be formed.

If a node A and a node B are connected by a directed link as $A \rightarrow B$, then B is said to be dependent on A. A is said to be the parent node of B and B is its child. A node, which has no parents but only children, is referred as a root node. Conversely, a node, which has no children, is referred as a leaf node. Any non-root or non-leaf node is referred to as an intermediate node, while root and leaf nodes are referred as edge nodes. The structure of a causal network may have different forms:

- Multiply connected,
- Serially connected,
- Tree.

The further details on causal networks can be found in [63], [64]. Figure 3.1 shows the examples of causal network for modeling of the real world problems. As shown, causal networks are used to model wind turbine gearbox [65] and automotive brake-by-wire system [38] for the purpose of fault diagnosis.



(a) (b)

Figure 3.1: Examples of representing real world problems with causal networks: (a) wind turbine gearbox, (b) automotive brake-by-wire system [38].

3.2.1 Representation of Sensor and System on Causal Network

Figure 3.2 shows the simplest causal network with two nodes. A and B correspond to some physical variables measurable, using appropriate sensors. The link $A \rightarrow B$ between the two nodes denotes that they are causally related i.e. $B = f(A)$ and thus represents the '*system AB'*. To make this point clear, we may refer to the liquid tanks process, which is described in chapter 2 (section 2.4.3). The dynamic model of system is derived in Appendix A. The analogy of components of this process to a two-node causal network is given in Table 3.1.

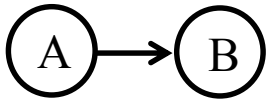


Figure 3.2: A causal network representing a two-variable system

Table 3.1: Modeling a two-tank process with causal network and its analogy

Causal Network Model	Equivalent in Multi Reservoirs System
Node A	Liquid level in tank 1 (H_1)
Node B	Liquid level in tank 2 (H_2)
Link $A \rightarrow B$	Valve (v_1)

3.2.2 Serially Connected Causal Network

In case of a serially connected causal network, there is only one path between any two nodes in the network. Only this architecture illustrated in Figure 3.3 is useful for modeling of system in the proposed methodology. Alternatively, multiply connected and tree network can be truncated to several serially connected causal network. This point is further discussed in section 3.8.

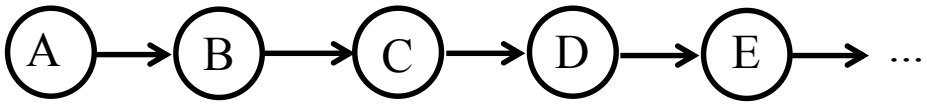


Figure 3.3: A serially connected causal network, for modeling a multivariable system

Among all variables of a system, a certain subset is required to be monitored to make the system detectable. This subset is the minimum number of variables, which makes it possible to *isolate* the system faults from each other. The necessary (but not sufficient) condition is that the system should be observable. The criteria for detectability of the system and isolability of faults from each other are addressed in [54], [55].

Once the minimum subset of variables is defined, which clearly must be observable, each of the variables should be equipped with a proper sensor in the design phase for the purpose of monitoring. In Figure 3.4, the sensors are shown as boxes corresponding to each variable.

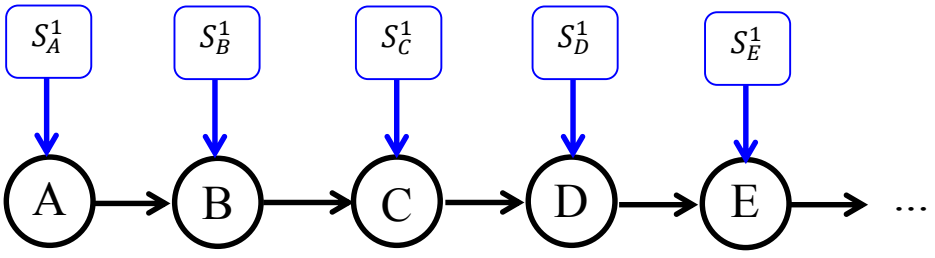


Figure 3.4: A set of variables in the system equipped with single sensor for monitoring

Many papers discuss the issue of sensor placement for fault detection and isolation [54]–[56]. This is the necessary configuration (but not sufficient) to detect and isolate faults. However, relying only on the detected fault after the detection procedure, no decision can be made at this point regarding the origin of the fault, i.e. system or sensor itself. Since monitoring this subset is the necessary condition for detection of faults, the lesser number of sensors will result in inability to detect and isolate faults. The magnitudes of faults and handling the uncertainties, e.g. noise and unmodeled parameters have a substantial rule in correct detection of faults.

Since the purpose of this research is crisp distinguishing of sensor and system faults, the issue of uncertainty is not considered for explaining the methodology. The physics of sensor as well as handling noise and uncertainty are discussed in chapter 4 as a complementary problem.

3.3 Duplication of Sensors

Based on the definition of fault (a deviation in measurement from the model output) the aforementioned configuration of sensors will provide the grounds for detecting faults with model based detection techniques (either sensor or system). However, for the purpose of corrective action or compensatory response, no crisp decision is conclusive on the origin of the detected fault with single sensor on each variable. Some studies suggested following statements regarding the isolability of sensor faults [55], [56]

- A fault of a sensor placed for solving the detectability problem is only detectable
- A fault of two sensors placed for solving the detectability problem is always isolable between them.

These statements implicitly describe the criteria for distinguishability of sensor faults from system faults. The following statement is the assumption that this methodology is built upon:

The probability of simultaneous faults in the essential and the redundant sensors is statistically close to zero.

While the probability of individual faults in each sensor is not zero, this point is statistically true, since the essential and redundant sensors are installed in parallel, which makes the probability of concurrent faults next to zero. Assuming this, in presence of a redundant sensor S_i^2 for the essential sensor S_i^1, a sensor fault in either S_i^1 or S_i^2 is isolable between them. Notation S_i^j corresponds to the sensor of i^{th} variable and superscript j denotes the number of sensor installed on the variable.

While both sensors S_i^1 and S_i^2 monitor one variable, if the readings at a particular instant have discrepancy (i.e. larger than the defined threshold for consideration of measurement noise), it is a strong indication of a fault in either sensor. The *duplex sensor system*, which takes advantage of two identical sets of instruments have been also discussed in [24], [66], [67]. Two-like sensors detect the occurrence of a fault by hardware redundancy. Since handling uncertainty is discussed later, for convenience, we consider that in fault-free condition, the readings from both sensors are equal.

Having this, in model based fault diagnosis environment, while the sensors are duplicated, any fault can be isolated and the origin can be identified (localized), since one sensor confirms the model output, and the other is deviated. On the other hand, system fault will result in deviation of both sensors (according to sensor placement problem for fault diagnosis theory), and the deviation from the model, will result in successful detection of the fault. The configuration of duplication is shown schematically in Figure 3.5. The significance of reading for both sensors S_i^1

and S_i^2 is exactly the same; therefore, there is no privilege between them. Following, the distinguishability criteria are given:

- A mathematical model of system's physics is available,
- The system is observable,
- The minimum subset of variables for solving detectability and isolability of fault modes is satisfied,
- Each variable in the aforementioned subset to be monitored, is equipped properly with two sensors (essential and redundant),
- The probability of simultaneous faults in the essential and the redundant sensors is zero.

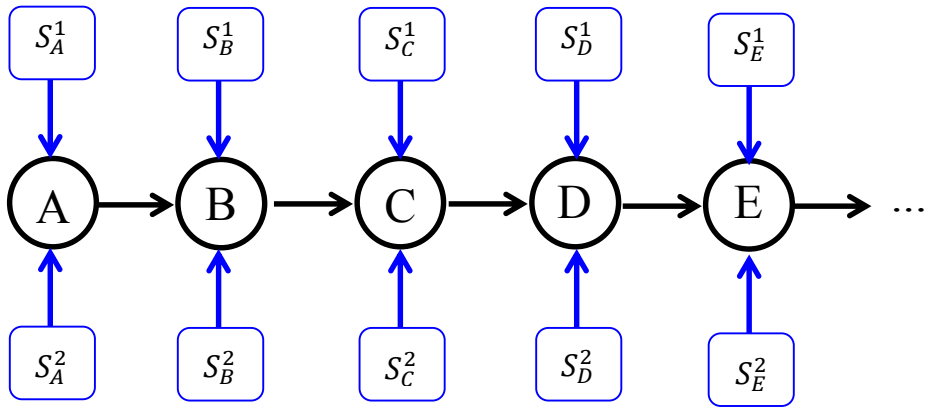


Figure 3.5: Schematic view of a system with duplication of all sensors

Hence, by satisfying the aforementioned criteria, the problem is solved. Therefore, *the sensor fault and system fault are distinguishable only and only if all sensors are duplicated*.

3.3.1 Difficulties Associated with Sensor Redundancy

It is common to employ multiply redundant sensor, for measuring the variables, which are critical for monitoring the system. It is important in particular for those systems that require very

high overall reliability [34], in order to enhance safety and performance, and reduce uncertainty associated with measurement and estimation [61]. It was shown in the previous section that duplication of sensors is effective in distinguishing the case where a sensor fault or a system fault is responsible for the indication of an abnormal state.

However, there are downsides associated with sensor redundancy (which is referred as providing *"direct redundant measurement"*). These factors include but not limited to additional costs; weight, space, electrical/power and installation constraints; increased complexity; and finally the sophistication associated with redundant data from measurements.

For example, space shuttles or satellites have a very limited capacity for carrying extra weight and each redundant sensor will be an additional undesired load. Similarly, implementing redundant sensors are not always feasible. As an instance, for measuring strain at a particular point of a beam, only one sensor can be placed.

Given these constraints with redundant sensors, there is a reluctance to add sensors in order to fully satisfy the duplicated configuration subset. Therefore, in order to successfully perform the task of sensor/system fault differentiation, we should minimize the sensor set, while producing enough redundant analytical substitutions using functional relationships to either confirm or reject the measurement data from existing sensors.

Henceforward, the problem of distinguishing sensor and system faults will be addressed as 'what *degree of redundancy* is sufficient to perform a crisp decision on differentiation of the aforementioned faults?'

3.4 Minimum Sensor Redundancy

Strict duplication of sensors will result in crisp distinguishability of sensor and system faults. The diagnostic methodology described in this section is based on minimizing the sensor redundancy and knowledge utilization, without comprising the distinguishability and diagnosability of the system. Hence, by using physical relationships between the monitored variables, we can reduce the degree of redundancy from strict duplications. Figure 3.6 depicts the structure of the method.

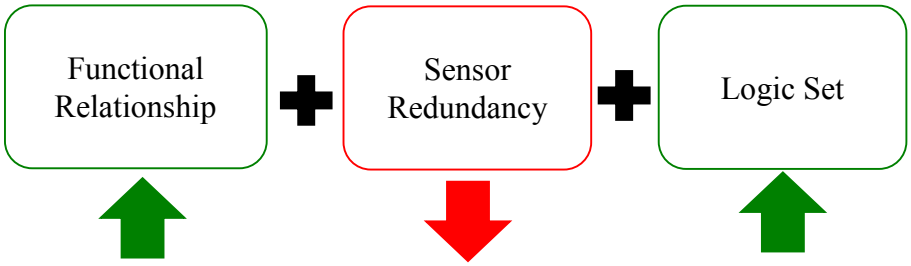


Figure 3.6: Overview of the structure of the method

Functional relationships are physical relations between the variables, which are described in a mathematical form. It basically models the first principle relationships between variables, e.g. mass balance. The form of mathematical equation can vary from linear to highly nonlinear, temporal etc., as long as there is a closed form solution between the two variables. Given the form of causal model in the section 3.2.2, these functional relationships can generate analytical values to check the credibility of sensor readings in neighboring nodes.

Definition 1: The analytical values generated by functional relationships, using readings from sensors are called *Analytical Computational Substitutions (ACS)* in this manuscript. Generation of ACS could be computationally expensive, depending on the form of functional

relationships; however, they should be calculated online at each instant of time while monitoring. Figure 3.7 shows relationship between ACS and credibility checking for sensor readings.

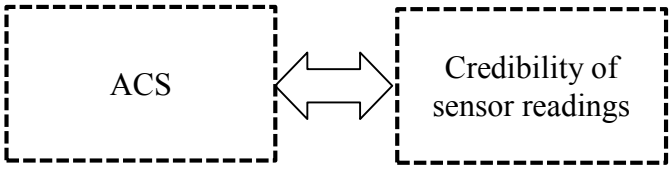


Figure 3.7: The relationship between ACS and credibility checking

Definition 2: Logic set consists of all system/sensor state possibilities, which are called *System Behavioral Modes*. It is designed offline with a set of knowledge-based rules (e.g., IF symptom AND symptom THEN conclusion). The parametric design of the logic set allows on-line decision-making by comparing the generated ACS and sensor readings at the sampling point.

3.4.1 Mitigation of Redundant Sensors

Now we use the definition of ACS and logic set to eliminate a number of redundant sensors from duplicated configuration. Removing a duplicated sensor from a particular variable is effective, only if neighboring nodes can generate the values of it, i.e. ACS. Hence, a sufficient subset must at least contain three nodes.

Definition 3: We define a window, which covers three nodes at a time, as it traverses through the nodes of the system. This window is referred to as *'Moving Monitoring Window' (MMW)*.

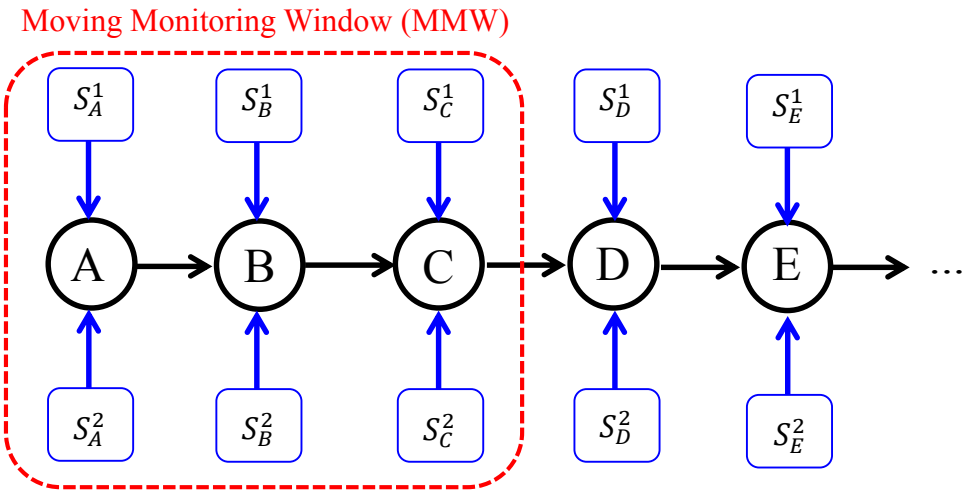
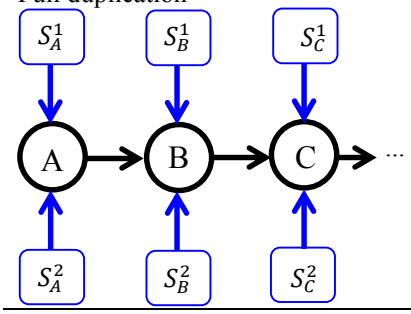
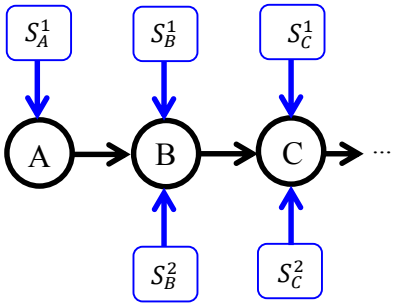
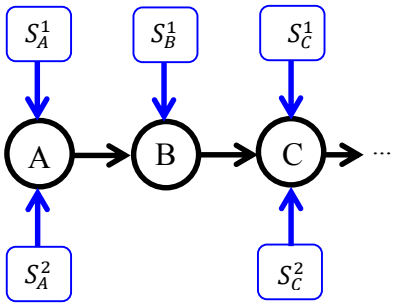
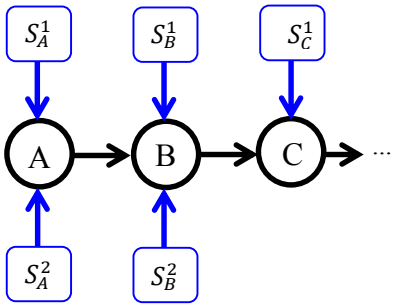


Figure 3.8: MMW considers three variables in a step of monitoring

Given three variables (nodes) in one MMV, the permutations of sensor culling and their effects are given in Table 3.2. It should be noted that the significance of S_i^1 and S_i^2 are the same; therefore removing either of them will have the same consequence.

Table 3.2: Permutations of mitigating redundant sensors in MMW

Configuration	Capability
Full duplication 	Faults detectable Sensor/system faults distinguishable
	Faults detectable Edge node with single sensor (sensor/system faults on *A* not distinguishable) [not enough relationships] sensor/system faults on *B* and *C* distinguishable
	Faults detectable Middle node with single sensor Using ACS to check the credibility of sensor on *B* †sensor/system faults distinguishable
	Faults detectable Edge node with single sensor (sensor/ system faults on *C* not distinguishable) [not enough relationships] sensor/system faults on *A* and *B* distinguishable

By using the described configuration noted with †, in each MMW we will have three variables and five sensors, which are placed in a way that middle variable (node) is bordered by variables (nodes) with duplicated sensors. Indeed, by removing this sensor from full duplication configuration, we lose some state possibilities of system, where the lack of these states is detrimental to distinguishing procedure in a few fault modes. Sensors will provide five measurements from three variables, at each sampling time. On the other hand, functional relationships between variables will generate six ACS corresponding to variable, as shown by arrows in Figure 3.9.

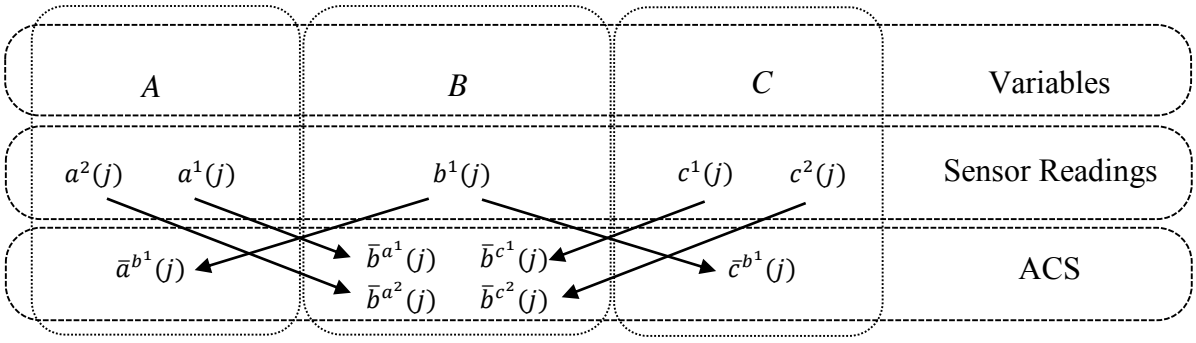


Figure 3.9: Variables, sensor readings, and ACS at time *(j)*

At any sampling point, j, two measurements from sensors S_A^1 and S_A^2, ($a^1(j)$ and $a^2(j)$) can provide two ACS for variable B ($\bar{b}^{a^1}(j)$ and $\bar{b}^{a^2}(j)$). Similarly, S_C^1 and S_C^2 provide two ACS for variable B. Measurement of S_B^1 provides one ACS for A and one for C using functional relationships ($\bar{a}^{b^1}(j)$ and $\bar{c}^{b^1}(j)$, respectively). These six values and five measurements enter the logic set unit for further processing.

3.4.2 Structure of the Logic Set Unit

This unit contains all possible states and combinations of measurements and values, and decides based on a bank of knowledge-based rules (behavioral modes). In addition to detection of faults and anomalies, this unit can act as the distinguisher and differentiate between sensor faults and system faults, and generate signals for corrective actions or compensatory responses. The parametric structure of the logic set allows comparing and further processing of sensor measurements and corresponding ACS. Table 3.3 lists all thirty state possibilities of behavioral modes and corresponding diagnosis.

Table 3.3: Logic set

State of Behavioral Modes	System Fault	Sensor Fault	Diagnosis
$a^2 = a^1, \bar{b}^{a^1} = b^1, \bar{c}^{b^1} = c^1 = c^2$	NO	NO	$S_A^1, S_A^2, S_B^1, S_C^1, S_C^2, AB, BC$ OK
$a^2 = a^1, \bar{b}^{a^1} = b^1, c^1 \neq c^2, \bar{c}^{b^1} = c^1, \bar{c}^{b^1} \neq c^2$	NO	YES	$S_A^1, S_A^2, S_B^1, S_C^1, AB, BC$ OK S_C^2 Faulty
$a^2 = a^1, \bar{b}^{a^1} = b^1, c^1 \neq c^2, \bar{c}^{b^1} \neq c^1, \bar{c}^{b^1} = c^2$	NO	YES	$S_A^1, S_A^2, S_B^1, S_C^2, AB, BC$ OK S_C^1 Faulty
$a^2 = a^1, \bar{b}^{a^1} = b^1, c^1 = c^2, \bar{c}^{b^1} \neq c^1, \bar{c}^{b^1} \neq c^2$	YES	NO	$S_A^1, S_A^2, S_B^1, S_C^1, S_C^2, AB$ OK BC Faulty
$a^2 = a^1, \bar{b}^{a^1} \neq b^1, c^1 = c^2, \bar{b}^{c^1} = b^1, \bar{b}^{c^1} \neq \bar{b}^{a^1}$	YES	NO	$S_A^1, S_A^2, S_B^1, S_C^1, S_C^2, BC$ OK AB Faulty
$a^2 = a^1, \bar{b}^{a^1} \neq b^1, c^1 = c^2, \bar{b}^{c^1} \neq b^1, \bar{b}^{c^1} = \bar{b}^{a^1}$	NO	YES	$S_A^1, S_A^2, S_C^1, S_C^2, AB, BC$ OK S_B^1 Faulty
$a^2 = a^1, \bar{b}^{a^1} \neq b^1, c^1 = c^2, \bar{b}^{c^1} \neq b^1, \bar{b}^{c^1} \neq \bar{b}^{a^1}$	YES	YES	* SHUT DOWN
$a^2 = a^1, \bar{b}^{a^1} \neq b^1, c^1 \neq c^2, \bar{b}^{c^2} = b^1, \bar{b}^{c^2} \neq \bar{b}^{a^1}$	YES	YES	$S_A^1, S_A^2, S_B^1, S_C^2, BC$ OK S_C^1, AB Faulty
$a^2 = a^1, \bar{b}^{a^1} \neq b^1, c^1 \neq c^2, \bar{b}^{c^2} \neq b^1, \bar{b}^{c^2} = \bar{b}^{a^1}$	YES	YES	S_A^1, S_A^2, S_C^2, BC OK S_B^1, S_C^1, AB Faulty
$a^2 = a^1, \bar{b}^{a^1} \neq b^1, c^1 \neq c^2, \bar{b}^{c^2} \neq b^1, \bar{b}^{c^2} \neq \bar{b}^{a^1}$	YES	YES	* SHUT DOWN
$a^2 = a^1, \bar{b}^{a^1} \neq b^1, c^1 \neq c^2, \bar{b}^{c^1} = b^1, \bar{b}^{c^1} \neq \bar{b}^{a^1}$	YES	YES	$S_A^1, S_A^2, S_B^1, S_C^1, BC$ OK S_C^2, AB Faulty
$a^2 = a^1, \bar{b}^{a^1} \neq b^1, c^1 \neq c^2, \bar{b}^{c^1} \neq b^1, \bar{b}^{c^1} = \bar{b}^{a^1}$	YES	YES	S_A^1, S_A^2, S_C^1, BC OK S_B^1, S_C^2, AB Faulty
$a^2 = a^1, \bar{b}^{a^1} \neq b^1, c^1 \neq c^2, \bar{b}^{c^1} \neq b^1, \bar{b}^{c^1} \neq \bar{b}^{a^1}$	YES	YES	* SHUT DOWN

$a^2 = a^1, c^1 = c^2, \bar{b}^{a^1} \neq b^1, \bar{b}^{c^1} \neq b^1, \bar{c}^{b^1} \neq \bar{b}^{a^1}$	YES	NO	* SHUT DOWN AB, BC Faulty
$a^2 \neq a^1, \bar{b}^{a^2} = b^1, \bar{c}^{b^1} = c^1 = c^2, \bar{c}^{b^1} = \bar{b}^{a^2}$	NO	YES	$S_A^2, S_B^1, S_C^1, S_C^2, AB, BC$ OK S_A^1 Faulty
$a^2 \neq a^1, \bar{b}^{a^2} = b^1, c^1 \neq c^2, \bar{c}^{b^1} = c^1, \bar{c}^{b^1} = \bar{b}^{a^2}$	NO	YES	$S_A^2, S_B^1, S_C^1, AB, BC$ OK S_A^1, S_C^2 Faulty
$a^2 \neq a^1, \bar{b}^{a^2} = b^1, c^1 \neq c^2, \bar{c}^{b^1} = c^2, \bar{c}^{b^1} = \bar{b}^{a^2}$	NO	YES	$S_A^2, S_B^1, S_C^2, AB, BC$ OK S_A^1, S_C^1 Faulty
$a^2 \neq a^1, \bar{b}^{a^2} = b^1, c^1 = c^2, \bar{c}^{b^1} \neq c^1, \bar{c}^{b^1} \neq c^2$	YES	YES	$S_A^2, S_B^1, S_C^1, S_C^2, AB, BC$ OK S_A^1, BC Faulty
$a^2 \neq a^1, \bar{b}^{a^2} \neq b^1, c^1 = c^2, \bar{b}^{c^1} = b^1, \bar{b}^{c^1} = \bar{b}^{a^1}$ (then $\bar{b}^{a^1} = b^1$)	NO	YES	$S_A^1, S_B^1, S_C^1, S_C^2, AB, BC$ OK S_A^2 Faulty
$a^2 \neq a^1, \bar{b}^{a^2} \neq b^1, c^1 = c^2, \bar{b}^{c^1} = b^1, \bar{b}^{c^1} \neq \bar{b}^{a^1}$	YES	YES	* SHUT DOWN
$a^2 \neq a^1, \bar{b}^{a^2} \neq b^1, c^1 = c^2, \bar{b}^{c^1} \neq b^1, \bar{b}^{c^1} = \bar{b}^{a^1}$ (then $\bar{b}^{a^1} = b^1, \bar{c}^{b^1} \neq c^1$)	NO	YES	$S_A^1, S_C^1, S_C^2, AB, BC$ OK S_A^2, S_B^1 Faulty
$a^2 \neq a^1, \bar{b}^{a^2} \neq b^1, c^1 = c^2, \bar{b}^{c^1} \neq b^1, \bar{b}^{c^1} \neq \bar{b}^{a^1}$	YES	YES	* SHUT DOWN
$a^2 \neq a^1, \bar{b}^{a^2} \neq b^1, c^1 \neq c^2, \bar{b}^{c^1} \neq b^1, \bar{b}^{c^1} \neq \bar{b}^{a^1}$	YES	YES	* SHUT DOWN
$a^2 \neq a^1, \bar{b}^{a^1} = b^1, c^1 = c^2, \bar{c}^{b^1} = c^1, \bar{b}^{c^1} = \bar{b}^{a^1}$	NO	YES	$S_A^1, S_B^1, S_C^1, S_C^2, AB, BC$ OK S_A^2 Faulty
$a^2 \neq a^1, \bar{b}^{a^1} = b^1, c^1 \neq c^2, \bar{c}^{b^1} = c^1, \bar{b}^{c^1} = \bar{b}^{a^1}$	NO	YES	$S_A^1, S_B^1, S_C^1, AB, BC$ OK S_A^2, S_C^2 Faulty
$a^2 \neq a^1, \bar{b}^{a^1} = b^1, c^1 \neq c^2, \bar{c}^{b^1} = c^2, \bar{b}^{c^2} = \bar{b}^{a^1}$	NO	YES	$S_A^1, S_B^1, S_C^2, AB, BC$ OK S_A^2, S_C^1 Faulty
$a^2 \neq a^1, \bar{b}^{a^1} \neq b^1, c^1 = c^2, \bar{b}^{c^1} = b^1, \bar{b}^{c^1} \neq \bar{b}^{a^1}$	YES	YES	* SHUT DOWN
$a^2 \neq a^1, \bar{b}^{a^1} = b^1, c^1 = c^2, \bar{b}^{c^1} \neq b^1, \bar{b}^{c^1} = \bar{b}^{a^2}$	NO	YES	$S_A^2, S_C^1, S_C^2, AB, BC$ OK S_A^1, S_B^1 Faulty
$a^2 \neq a^1, \bar{b}^{a^1} = b^1, c^1 = c^2, \bar{b}^{c^1} \neq b^1, \bar{b}^{c^1} \neq \bar{b}^{a^2}$	YES	YES	* SHUT DOWN
$a^2 \neq a^1, \bar{b}^{a^1} = b^1, c^1 \neq c^2, \bar{b}^{c^1} \neq b^1, \bar{b}^{c^1} \neq \bar{b}^{a^2}$	YES	YES	* SHUT DOWN

3.4.3 Multiple Faults

When the number of concurrent faults in a MMW exceeds than two or three, practically, the operation of system should be halted. As shown in the Table 3.3, some diagnoses are "Shut Down", denoted by *. This means that the number of faults in either sensor or system is more

than the case that we can distinguish the origin of it, due to the missing information of corresponding state possibilities, which is the consequence of removing a sensor from full duplication configuration. This is also practically not feasible to have a high number of faults at the same time, unless another underlying issue caused it. In these situations, the system should be shut down immediately to stop the catastrophic consequences. However, in all other cases the diagnostic system is able to successfully handle multiple faults, while maintains the ability to distinguish and locate sensor and system malfunctions.

3.4.4 Sensor Fault Tolerance Strategy

This diagnostic methodology is capable to decide on this issue and utilize either sensor reading or 'analytical computational substitution (ACS)'. Subsequently it has a capacity to tolerate some sensor faults temporarily; however, the malfunctioning sensor(s) should be fixed or replaced as soon as possible. For example in the case that the diagnosis is that S_B^1 is faulty and all other component are healthy, the ACS from node A or C can be substitute for monitoring node B. Hence, $\bar{b}^{a^1}$, $\bar{b}^{a^2}$, $\bar{b}^{c^1}$or $\bar{b}^{c^2}$can cover the reading of S_B^1. However, this substitution is not sustainable and may not be valid after some sampling points; hence the fault tolerance cannot be guaranteed for large time spans. In this situation, the system works in *'degraded mode'*.

As is also clear, a fault in either sensors of A or C can be tolerated by using the validated measurement from its corresponding sensor. For example, if S_A^1 is faulty, the validated reading from S_A^2 (a^2) is used as the correct measurement of node A. Since it is for a single fault, we call that sensor fault tolerant degree one. The system is also capable of sensor fault tolerance degree two, which means that two sensor faults can be tolerated. For instance, a fault in S_A^2 and a fault in S_B^1 can be tolerated at the same time with the aforementioned logic. It should be noted that the

probability of concurrent faults in duplicated sensors is zero, hence, S_A^1 and S_A^2 cannot be faulty at the same time.

3.4.5 Structure of Proposed Diagnostic System

Figure 3.10 depicts the structure of the proposed diagnostic system with minimum number of redundant sensors. To re-iterate, measurements are used to generate ACS. Then measurements and ACS enters the logic set unit to make a decision on the status of the system, based on predefined behavioral modes. If the system is unable to continue operation, it will be shut down. Moreover, if the system is able to operate in presence of a fault, the diagnostic system runs the fault isolation procedure, commands the system to operate in degraded mode and temporarily initiates the sensor fault tolerant strategy.

It can be concluded that:

When a variable (node) is bordered with two nodes with duplicated sensors, one sensor is sufficient for the task of distinguishing sensor fault from system fault, since the ACS generated by neighboring measurements can provide redundant values in order to check the credibility of reading of the single senor installed on the middle node. Removing any more sensors leads to inability to locate and differentiate between sensor and system faults, due to the lack of adequate behavioral modes for diagnosis decision-making.

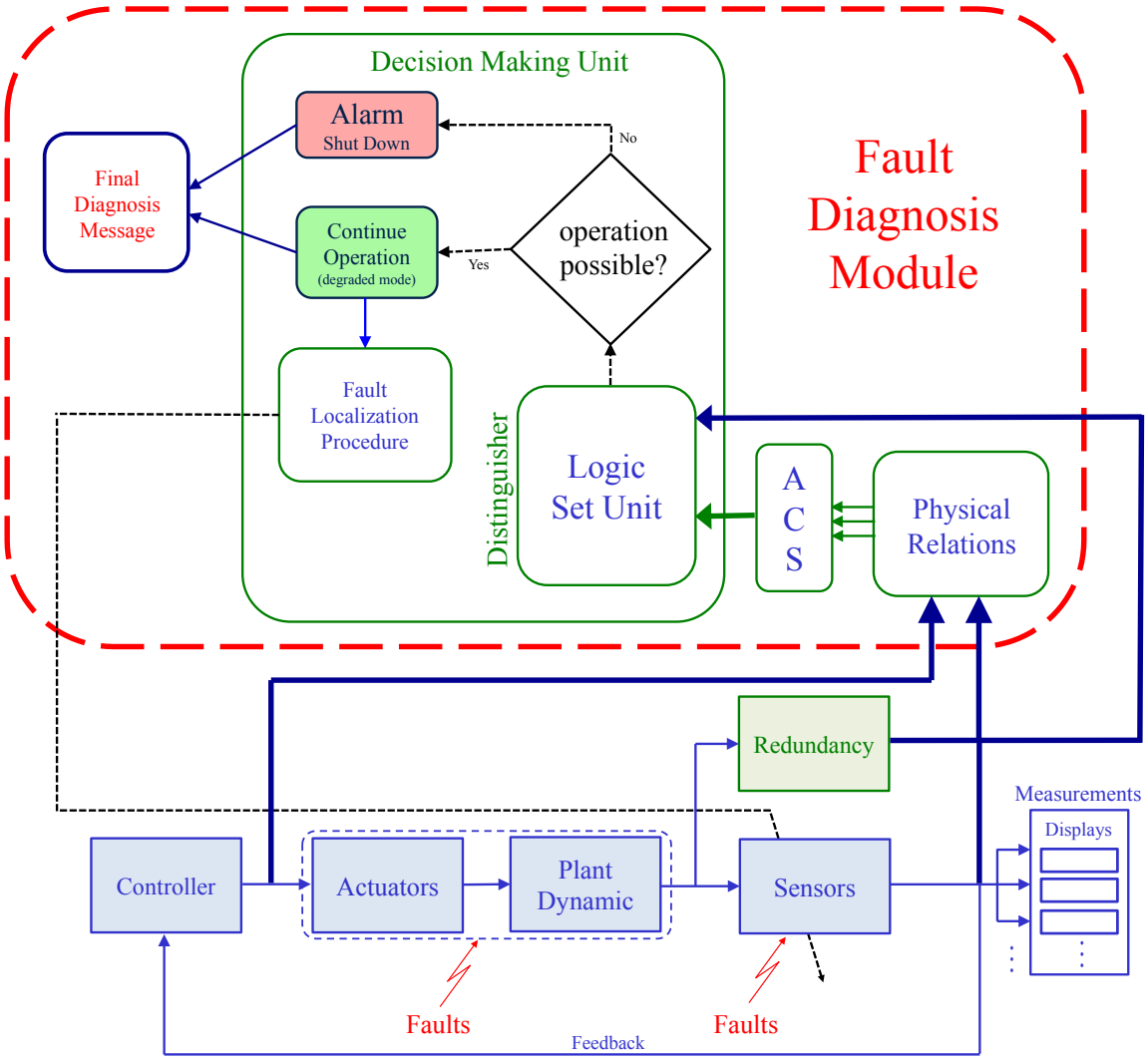


Figure 3.10: The structure of the proposed diagnostic system

3.5 Generalization by Deduction

We discussed the concept of MMW, which monitors three variables and then traverses in direction of causality to monitor the next proper set. If the system is modeled with only three variables, there is no need to move the window. However, for the systems with larger number of variables, the redundancy configuration methodology has two strategies. It is noted that edge nodes must always have duplicated sensors, and MMW moves in the direction of causality.

As shown in Figure 3.11, for a system with four variables, since the first and fourth nodes are edge nodes, they must have full sensor duplication. There are two subsets with three nodes, $\{A, B, C\}$ and $\{B, C, D\}$ that MMW can cover. Considering Figure 3.11 (a), MMW first monitors subset $\{A, B, C\}$, where B has only one sensor. Since D has sensor duplication, there is no need to move MMW and monitor subset $\{B, C, D\}$. Alternatively, as shown in Figure 3.11 (b), MMW can first monitor subset $\{B, C, D\}$, where C has only one sensor. There is no fundamental difference between these two solutions and they show that by having four variables, instead of eight sensors, we can perform fault localization with seven sensors.

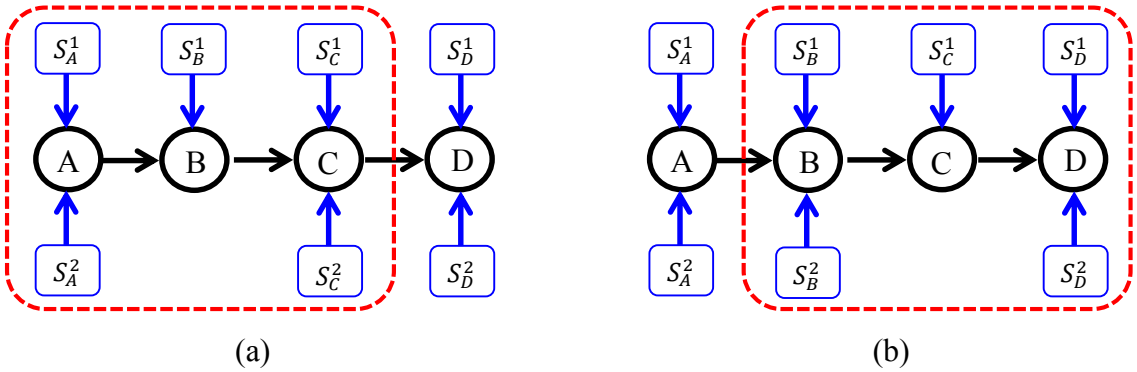


Figure 3.11: A system with four variables (nodes), (a) node *B* is single-sensor; or alternatively in (b) node *C* is single-sensor

For a system with five variables, we have three subsets of $\{A, B, C\}$, $\{B, C, D\}$ and $\{C, D, E\}$, when MMW moves in the direction of causality. Since A and E are edge nodes, they must have duplicated sensors. As schematically shown in Figure 3.12 (a), when MMW covers subset of $\{A, B, C\}$, the node B is single-sensor. Moving MMW in the direction of causality to monitor $\{B, C, D\}$, since edge node B is not duplicated, monitoring of this subset is not effective. Then by covering subset $\{C, D, E\}$, the node D can be single-sensor, since it is bordered by two sensor-duplicated nodes. Hence, the MMW is at position 1 at first to check $\{A, B, C\}$, and then moves to position 2 and performs exact similar action on $\{C, D, E\}$, as clearly shown in Figure 3.12 (b).

The nodes B and D are dominated with two nodes, which have duplicated sensors. Therefore, one sensor is enough for each of them, and then ACS, provides redundant values. For a five-variable system, eight sensors are enough to perform the fault localization task.

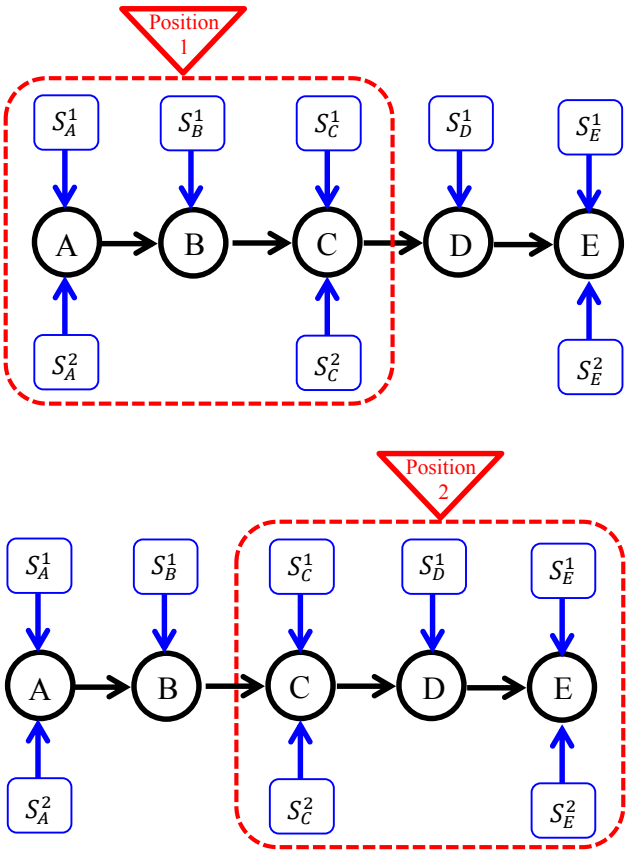


Figure 3.12: A system with five variables (nodes), MMW moves from position 1 in (a) to position 2 in (b)

For a system with six variables, the condition is similar to the system with four variables. There are four subsets of $\{A, B, C\}$, $\{B, C, D\}$, $\{C, D, E\}$, and $\{D, E, F\}$. Given the sensor duplication of edge nodes, covering of subsets of $\{B, C, D\}$ and $\{D, E, F\}$ by MMW is not effective. Hence, as shown in Figure 3.13 (a), MMW starts with subset $\{A, B, C\}$ and then moves to position 2, to monitor subset $\{C, D, E\}$. Since F has duplicated sensors, there is no need to move MMW toward that. In this configuration, by having six variables, ten sensors are enough to distinguish sensor and system faults.

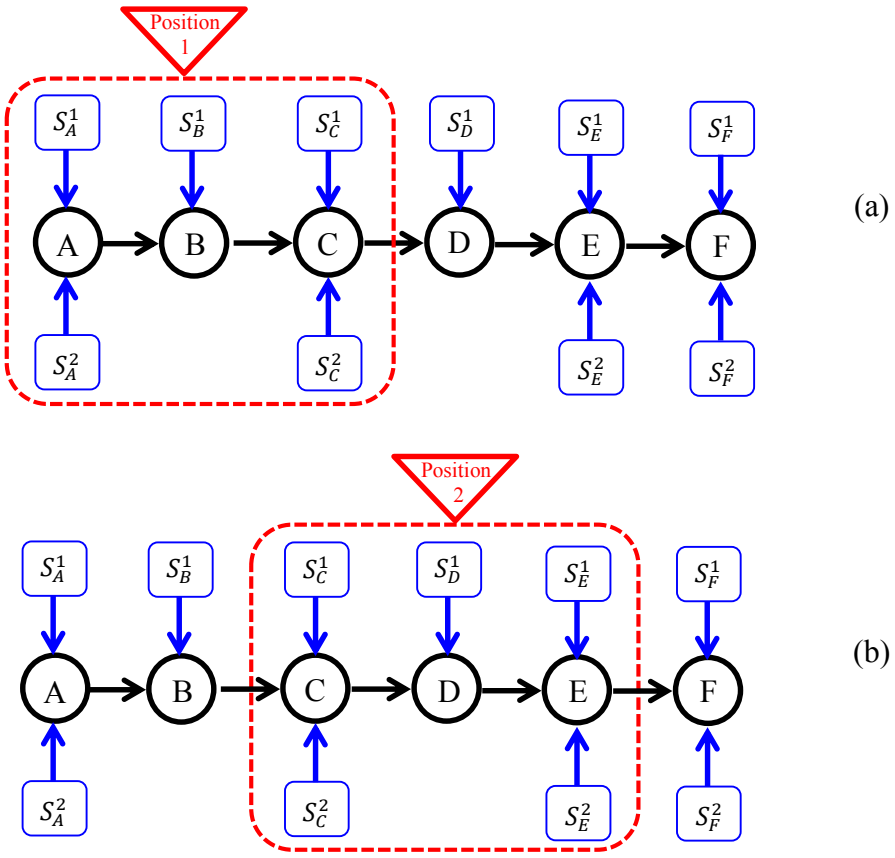


Figure 3.13: A system with six variables (nodes), MMW moves from position 1 in (a) to position 2 in (b)

For a larger number of variables, the process is analogous; they are either similar to a system with 5 variables or 6 variables. When the number of variables in a causal model is odd, the situation is similar to the system with 3 or 5 variables. On the other hand, when the number of variables is even, is it similar to the system with 4 or 6 variables. By *deduction*, the methodology remains analogous. Therefore, we can define the number of required sensors, which can perform the defined fault localization task. By extrapolating of the pattern of sensor placement, it is evident that the function, which shows the number of required sensors, depends on whether the number of variables in the causal model is even or odd.

Let's define m as the number of variable to be monitored, and n_d as the number of sensor required for the crisp distinguishing of sensor fault and system fault based on the aforementioned method, by deduction:

$$\left.\begin{array}{l} \text{if } m \in odd \;\Rightarrow\; n_d = 1.5 \times m + 0.5 \\ \text{if } m \in even \;\Rightarrow\; n_d = 1.5 \times m + 1 \end{array}\right\} \Rightarrow \; n_d > 1.5 \times m \tag{3.1}$$

It should be noted that for the purpose of control or model-based fault detection, having at least m sensors is sufficient. Figure 3.14 depicts a schematic graph, where the number of variables - to be monitored - and the number of required sensors are plotted against each other. We have found that by using the described configuration, the sensor and system faults are distinguishable. In this configuration, if the number of sensor is more than 1.5 times of variable, the task is feasible; hence the green-hatched area represents distinguishability region for the number of sensors in this arrangement.

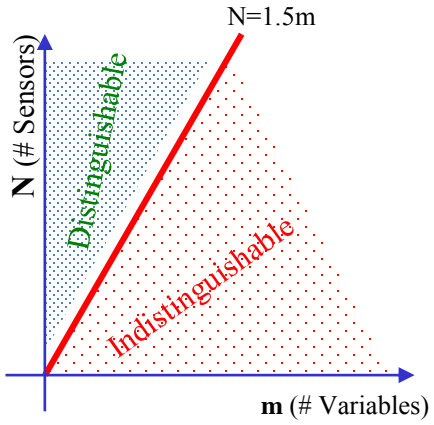


Figure 3.14: Distinguishability region based on the number of sensor and variables in a system

The generalization is established on deduction, since the methodology works for any number of variables. It means that if the variables can be modeled as serially a connected causal network, the number of required sensor to distinguish sensor faults and system faults should be at least greater than one and half times of the number of variables. Without this number of sensors in the aforementioned configuration, full localization of sensor and system faults is inconceivable. Table 3.4 numerically shows that strict duplication of sensors is not necessary in this case.

Table 3.4: The number of variables, full duplication configuration versus the proposed minimum sensor redundancy

Number of variables	Number of sensors for full duplication configuration	Minimum number of sensors for the proposed method
1	2	2
2	4	4
3	6	5
4	8	6
5	10	8
6	12	10
7	14	11
8	16	13
9	18	14
10	20	16
⋮	⋮	⋮
99	198	149
100	200	151
⋮	⋮	⋮
m	$2m$	$\begin{cases} 1.5m + 0.5 & \text{if } m \epsilon \text{ odd} \\ 1.5m + 1 & \text{if } m \epsilon \text{ even} \end{cases}$

3.6 Sensor Culling Degrees of Freedom

There are some cases that minimization of sensor duplication is not the objective. On the other hand, there might be a restriction for a particular variable, in a way that it cannot be sensor-duplicated. Some of these restrictions are discussed in section 3.3.1, such as infeasibility or design requirements. As an instance, for measuring strain at a particular point of a beam, only

one sensor can be placed. Alternately, there are some cases that removing sensor duplication on a non-optimal set is more cost functional (e.g. the sensor is expensive, while bordering nodes are monitored with less expensive sensors).

In these cases, instead of minimizing the number of duplicated sensors, the objective is to maintain the distinguishability of the sensor and the system, while only one sensor is used to monitor the variable. For these cases, the idea of MMW can be used, regardless of minimizing the sensor subset. The variables of interest should be positioned as the middle node of MMW, while bordered by two neighboring variables, which have the capability to be sensor-duplicated. This brings a design Degree of Freedom (DOF), while maintains the distinguishability criteria.

To clarify this point, let's consider a system with nine nodes $\{A, B, C, D, E, F, G, H, I\}$. Given that the edge nodes of MMW should be kept sensor-duplicated, putting aside these nodes, the remaining set $\{B, D, F, H\}$ is the maximal DOF (set 1). However, if there is a design restriction for nodes in this set, the set $\{C, E, G\}$ can be single-sensor, as the alternative maximal DOF (set 2). While this set does not minimize the duplications, it brings a degree of freedom for sensor culling. The on-going discussion can be generalized to a larger number of variables. The permutations of removing sensor duplications with respect to the number of variables (nodes) are given in Table 3.5. Set 1 and 2 represent the upper bound of the number of sensors which can be removed from strict duplication configuration, while maintaining the distinguishability criteria. The total number of sensor culling configurations is many orders of magnitude larger.

Table 3.5: Permutations of sensor culling in case of design restrictions

Number of variables	Upper bound on restricted DOF	
	Set 1	Set 2
1	0	0
2	0	0
3	1	0
4	1	1
5	2	1
6	2	2
7	3	2
8	4	3
9	4	3
10	4	4
⋮	⋮	⋮
99	49	48
100	49	49
⋮	⋮	⋮
m	$\begin{cases} \frac{m-1}{2} & \text{if } m \in \text{odd} \\ \frac{m-2}{2} & \text{if } m \in \text{even} \end{cases}$	$\begin{cases} \frac{m-3}{2} & \text{if } m \in \text{odd} \\ \frac{m-2}{2} & \text{if } m \in \text{even} \end{cases}$

3.7 Features and Applications of the Method

Basically any dynamical system with serial causality in its model can be monitored with this method, and consequently, sensor and system faults will be distinguished. The multi reservoirs for liquids, transmission pipelines with several output valves, interconnected gas containers, etc. are examples of systems, which have several variables to be monitored, while their variables have physical (first principal) relationships between each other. The variables in these systems can be modeled with serially connected causal network, and then this diagnostic framework can be applied to them.

This method is capable of detecting faults, as well as distinguishing between sensor and system faults. Therefore, it can localize faults by determining the source. The underlying nature of the methodology can provide the grounds for tolerance of sensor faults, by deciding whether

to use the reading from a sensor or not. Since ACS is generated corresponding to each variable, its value can be used instead of sensor measurement for a short period of time for control and monitoring purposes, while the system can operate. But it is unsustainable, since in long run the faulty sensors or system components must be corrected or replaced. As discussed in section 3.4.4, the diagnostic system has embedded capability of sensor fault tolerance of degree one and degree two (see section 3.4.4).

Moreover, by tracking the faulty sensor signal based on the approach described in [60], it possible to identify the type of occurred sensor fault. For instance statistical analysis of senor signal (mean and variance) may reveal bias, drift or loss of effectiveness of sensor.

3.8 Extension of the Method to Non-Serially Connected Systems

Up to this point, the foundation of the methodology for mitigation of duplicated sensors is for a certain class of systems, which can be modeled with serially connected causal networks. While many real world systems can be described as serially connected causal models [64], it is noted that where the model cannot be defined as a serially connected causal network, the method is not effective. At any branching/joining point of a combined serial-parallel network, the sensors of the corresponding nodes should be duplicated in order to comply with the task of distinguishing. As shown in Figure 3.15 (a), the sensors for the nodes B, C and $\acute{C}$ are duplicated.

However, if we can find a branch in the causal network that has more than three nodes itself, the concept of MMW can be used to eliminate the duplication of sensor for the middle node of this branch. For instance, in Figure 3.15 (b), the branch with the subset of $\{\acute{C}, \acute{D}, \acute{E}\}$ can be

considered as a separate serially connected causal network, hence MMW can cover it. Given this, the node $\acute{D}$ can be single sensor, while the fault distinguishing capability is maintained.

By considering this argument, the non-serially connected causal networks (tree and multiply connected) can also be broken down to serially connected causal networks. Then the methodology can be applied to each branch individually. Although the number of sensors will not be minimized as before, it can be reduced from strict duplication configuration, but the minimum number of sensors required is configuration-dependent.

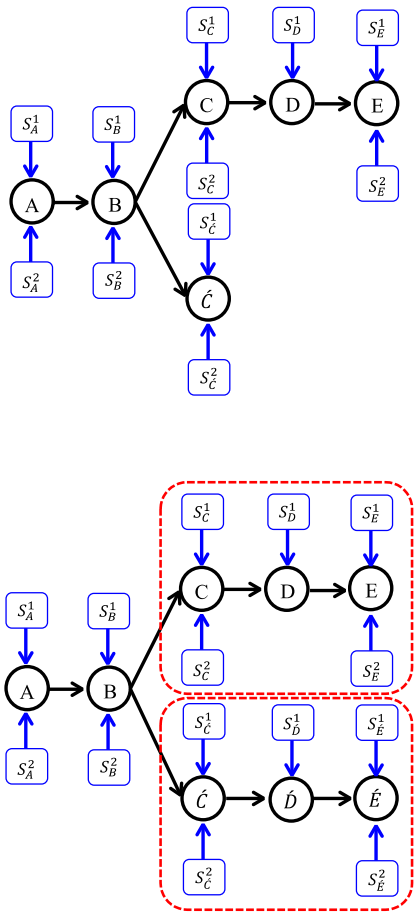


Figure 3.15: Extension for non-serially connected causal networks

Chapter 4

Example, Verification and Remarks

4.1 Introduction

In pervious chapter, a framework was introduced to distinguish sensor and system faults. By modeling the system a serially connected causal network and then using MMW and the logic set, the faults can be successfully localized and attributed to the sensor, system or a combination. In this chapter, we verify the methodology on a practical industrial system, in order to clarify the procedure.

4.2 Example of Interconnected Multi Reservoirs

In order to show the effectiveness of the aforementioned diagnostic method and sensor placement algorithm for distinction of sensor and system faults, we studied multi reservoirs process (commonly known as Continuous Stirred Tank Reactors (CSTR) in chemical industries) as an example. The process is described in chapter two and its dynamics is derived in Appendix A. It consists of a series of reservoir (tanks), control valves and liquid level sensors. In it noted that no other variable is monitored in this system. Reservoirs can have different architectures in their connections. However, the type of connection makes no difference in causality modeling. For simplicity, we follow the architecture shown in Figure 4.1 (a).

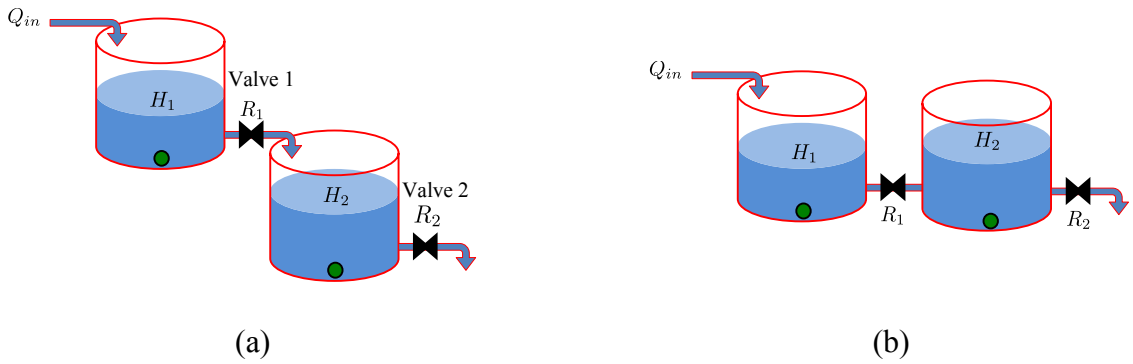


Figure 4.1: Different architectures of serially connected reservoirs: (a) flow rate in valve 1(v_1) is independent of liquid height in tank two; (b) flow rate in (v_1) is dependent to liquid heights in both tanks

4.2.1 Potential Faults in the Operation

The system fault and sensor fault, which may occur during the operation of process include:

- *Valve fault:* A fault in the valve (such as leakage) results in a change in the input flow rate of descendent tank without a corresponding adjustment in the control signal.
- *Sensor fault:* A fault in the liquid level sensor introduces a deviation into the measurement. For example, a bias in sensor reading causes the true flow rate to be maintained at a level below the set point.

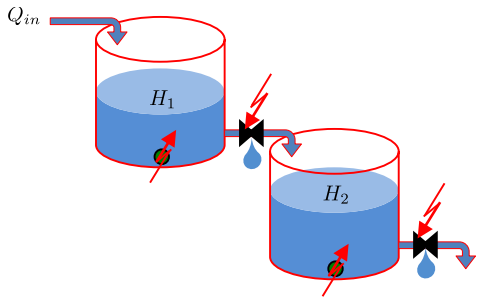


Figure 4.2: Schematic view of potential sensor/system faults in multi reservoir process

As schematically shown in Figure 4.3, these faults originate from system and sensor, respectively, but the size, type and time of occurrence are unknown. As discussed in section 2.4.3, in case of presence of feedback controller, the former may urge the controller to eventually correct the flow. Similarly in the latter, the controller responds to the deviation by modifying the flow rate. Now, the task is to determine whether a detected fault or anomaly is originated from system or sensor.

4.2.2 Modeling of Interconnected Multi Reservoirs with Causal Networks

Multi reservoirs can be modeled by causal networks, since the level of liquid in each tank is proportional to the flow rate from the valves. The flow rate in valves is also a function of height in the tank before the valve. The serially connected tanks architecture satisfies the serially connected casual network modeling requirement, which is described in the methodology. The liquid levels in the tanks are represented by nodes in causal network and the valves are the links between nodes. Now we consider a system with different number of interconnected tanks, and apply the methodology. For a system with one or two tanks, the number of permutations for behavioral modes is not sufficient to reduce the degree of redundancy from duplication.

We consider a system with three interconnected tanks, and duplicate all height-level measurement sensors. This configuration will provide a ground to fully distinguish sensor and system faults.

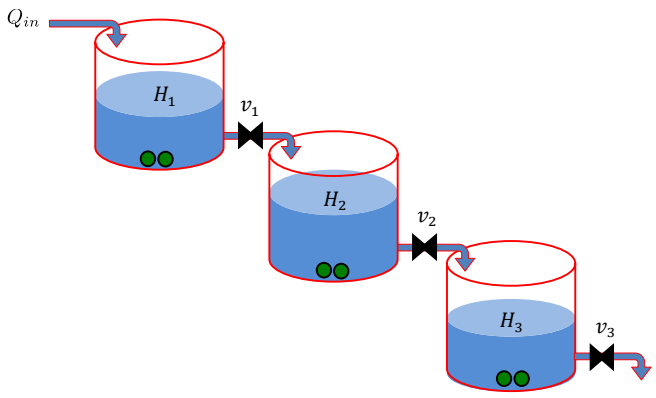


Figure 4.3: A system of three interconnected liquid reservoirs with duplicated height level sensors

Based on Figure 4.3, there are six different permutations of removing a sensor. According to the methodology, the only optimal redundancy mitigation solution is to remove one of the sensors in tank two. Therefore, tank number one and number three are monitored by duplicated sensors and tank two has only one sensor. This configuration has sufficient number of sensors to comply with the task. From the derivation in Appendix A, it is clear that the known control inputs are: $Q_{in}(k)$, $R_1(k), R_2(k)$, $R_3(k)$, which are input flow to the tank 1, and resistance of v_1, v_2 and v_3, respectively. The variables to be monitored and measured are $H_1(k)$, $H_2(k)$ and $H_3(k)$ at sampling time (k). Based on causality, we can define the relationships between the heights of the tanks as $H_2 = f(H_1)$ and $H_3 = g(H_2)$. By having the control inputs and model of system, the output can be generated (or estimated), as shown in Figure 4.4.

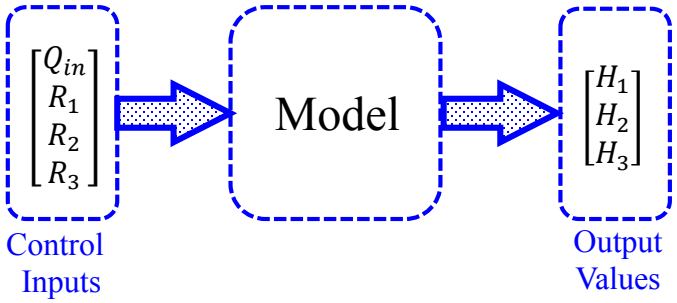


Figure 4.4: Relationships between inputs and outputs

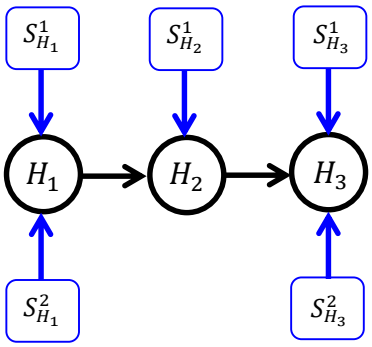


Figure 4.5: The causal network and sensor configuration for liquid tank process

The process of diagnosis and distinguishing for this system based on the methodology described in section 3.4 is as follows. At each sampling time, we have five measurements from sensors, corresponding to three variables, as shown in Figure 4.5. The sensor readings are:

- h_1^1 and h_1^2 corresponding to H_1,
- h_2^1 corresponding to H_2,
- h_3^1 and h_3^2 corresponding to H_3,

where superscript indicates the sensor that has been used for measurement and subscript corresponds to the variable. Additionally, we have three equations relating to the physics of the problem. Each equation computes the variation of height in a tank:

$$\dot{H}_1(k) = \frac{Q_{in}}{A_1} - \frac{H_1(k)}{A_1 R_1(k)} \quad \text{(for tank 1)} \tag{4.1}$$

$$\dot{H}_2(k) = \frac{H_1(k)}{A_2 R_1(k)} - \frac{H_2(k)}{A_1 R_2(k)} \quad \text{(for tank 2)} \tag{4.2}$$

$$\dot{H}_3(k) = \frac{H_2(k)}{A_3 R_2(k)} - \frac{H_3(k)}{A_3 R_3(k)} \quad \text{(for tank 3)} \tag{4.3}$$

Having the initial conditions, with these relationships, the measurements can be used to obtain ACS. By substituting each measurement, the ACS from one tank to another tank will be obtained. Using five measurements and three equations, six values in total can be derived.

Table 4.1: Variables, sensor readings, and ACS for three-reservoir system

Variable	H_1	H_2		H_3
Sensor measurements	h_1^1 , h_1^2	h_2^1		h_3^1 , h_3^2
ACS	$\bar{h}_1^{h_2^1}$	$\bar{h}_2^{h_1^1}$ $\bar{h}_2^{h_3^1}$	$\bar{h}_2^{h_1^2}$ $\bar{h}_2^{h_3^2}$	$\bar{h}_3^{h_2^1}$

In Table 4.1 $\bar{h}_p^q$ is the analytical value (ACS) derived from physical relationships. Here, index p denotes the variable number corresponding to ACS, and q denotes the sensor reading value that has been used to generate ACS. As shown in Table 4.1:

- Variable H_1: two measurements, one ACS
- Variable H_2: one measurements, four ACS
- Variable H_3: two measurements, one ACS

This configuration will result in 30 distinct behavioral modes in the system including all state possibilities, derived from system model, which are designed off-line similar to Table 3.3. The logic set contains the knowledge base parametric rules (e.g., IF symptom AND symptom THEN conclusion). These parametric values in this configuration are enough to construct a logic set. As mentioned before, it consists of 30 statements, which represent all distinctive behavioral modes of the process.

4.2.3 Fault Emulation

The procedure of constructing logic set is described in section 3.4.2. Hence, by having measurements from sensors and online generated ACS, a table similar to Table 3.3 can be composed. After constructing the logic set, any faults in the valves or liquid level sensors can be detected, and then localized. It is clear that in fault free case, $h_1^1 = h_1^2 = \bar{h}_1^{h_2^1}$, $h_2^1 = \bar{h}_2^{h_1^1} = \bar{h}_2^{h_1^2} = \bar{h}_2^{h_3^1} = \bar{h}_2^{h_3^2}$ and $h_3^1 = h_3^2 = \bar{h}_3^{h_2^1}$. This means that the level measurements and corresponding ACS for each monitored height are equal. Any discrepancy between these values is indicative of a fault. To avoid replication, only five scenarios of single fault in various sensors/components of liquid reservoirs are given in Table 4.2. Then in each scenario, the corresponding behavioral mode, which leads to distinction of sensor and system faults, is presented. The first two scenarios represent the cases where the system is faulty (i.e. control valves are leaking). The next three scenarios characterize the cases where sensor faults are responsible for discrepancy between measurement and ACS values. Only three sample sensor faults amongst all permutations of five liquid level sensors in the tanks are given here. The rest of the faults including multiple faults cases can be similarly diagnosed and localized.

Table 4.2: Fault scenarios in different components of reservoirs system and corresponding behavioral modes

Fault type	Fault location	Corresponding Behavioral Mode
System fault	Control Valve 1 v_1	$h_1^1 = h_1^2, \bar{h}_2^{h_1^1} \neq h_2^1, h_3^1 = h_3^2,$ $\bar{h}_2^{h_3^1} = h_2^1, \bar{h}_2^{h_3^1} \neq \bar{h}_2^{h_1^1}$
System fault	Control Valve 2 v_2	$h_1^1 = h_1^2, \bar{h}_2^{h_1^1} = h_2^1, h_3^1 = h_3^2,$ $\bar{h}_3^{h_2^1} \neq h_3^1, \bar{h}_3^{h_2^1} \neq h_3^2$
Sensor fault	Sensor 1 in tank 2 $S_{H_2}^1$	$h_1^1 = h_1^2, \bar{h}_2^{h_1^1} \neq h_2^1, h_3^1 = h_3^2,$ $\bar{h}_2^{h_3^1} \neq h_2^1, \bar{h}_2^{h_3^1} = \bar{h}_2^{h_1^1}$
Sensor fault	Sensor 1 in tank 1 $S_{H_1}^1$	$h_1^1 \neq h_1^2, \bar{h}_2^{h_1^2} = h_2^1,$ $\bar{h}_3^{h_2^1} = h_3^1 = h_3^2, \bar{h}_3^{h_2^1} = \bar{h}_2^{h_1^2}$
Sensor fault	Sensor 2 in tank 3 $S_{H_3}^2$	$h_1^1 = h_1^2, \bar{h}_2^{h_1^1} = h_2^1, h_3^1 \neq h_3^2,$ $\bar{h}_3^{h_2^1} = h_3^1, \bar{h}_3^{h_2^1} \neq h_3^2$

After distinguishing between the sensor and the system fault (localization), the diagnostic system commands for further actions. This action can vary from a simple alarm as a notification to full shut down and scheduling a maintenance plan, depending on the situation and severity of malfunction. Additionally, the sensor fault tolerant algorithms can be executed in some cases. For example if the diagnostic system conclude that the sensor in tank two is faulty (third scenario in Table 4.2), the corresponding ACS can be utilized instead.

4.2.4 Extension for Larger Number of Reservoirs

The procedure is extendable for systems with more tanks (and consequently more variables), by traversing MMW in the direction of causality between liquid levels in the tanks. Similar to the procedure given in section 3.5, the methodology can be applied to a process with any number of serially connected tanks. For example, in a system with six tanks, there are four subsets of $\{H_1, H_2, H_3\}$, $\{H_2, H_3, H_4\}$, $\{H_3, H_4, H_5\}$, and $\{H_4, H_5, H_6\}$. For consideration of minimum sensor set, given the sensor duplication of first and last tanks, covering of subsets of $\{H_2, H_3, H_4\}$ and $\{H_4, H_5, H_6\}$ by MMW is not effective. Hence, a MMW starts with subset $\{H_1, H_2, H_3\}$, and then moves to the next position, to monitor subset $\{H_3, H_4, H_5\}$. In this configuration, by having six variables of height, a minimum of ten sensors is sufficient to distinguish sensor and system faults. Alternatively, for the purpose of design freedom, MMW can covers subset $\{H_2, H_3, H_4\}$, which tank 3 has only one sensor and the rest have duplicated sensors.

In a similar manner, the methodology can be applied to any system with more number of tanks (variables). It can be generalized that by maintaining the configuration of sensors in each position of MMW (three variables, five sensors), while first and last tanks have necessarily duplicated sensors, the procedure remains analogous. Accordingly, the configuration of sensors

confirms that the number of sensors should be greater than 1.5 times of variables. This degree of redundancy makes is feasible to distinguish sensor and system faults.

This configuration can be applied to a parallel but independent variable in the system as well. For example, if the temperature is also monitored in the tanks, while the number of temperature sensors is greater than 1.5 times of times of the number of tanks (considering the placement configuration), the temperature sensor faults and system faults (e.g. heaters in the tanks) are distinguishable.

4.3 Remarks on Presence of Uncertainty

The first principle models (physical relationships) used in this methodology are deterministic. Once we determine the degree of sensor redundancy for distinguishing sensor and system faults in deterministic case, we can incorporate uncertainty models as add-ons to the procedure. The important issue of robustness to uncertainty is not the subject of current research, however, this part can be considered as an introduction for future works (section 5.3). Usually the uncertainty representations for estimation and detection are extensions of the deterministic model. The sources of uncertainty are discussed in section 1.3.2, where measurement interference noise, model inaccuracies and disturbance in system are the main components. It is common in literature to model the uncertainty by additional noise terms to the measurements and state equations as shown in equations (2.3) and (2.4). The components of uncertainty are usually derived from empirical models such as regression model.

Indeed, introducing noise as well as using detection techniques leads to a number of missed and/or false alarms in the diagnostic procedure. This part is concerned with ‘detection’ of faults,

whereas establishing the minimum number of sensors, described in this methodology, results in crisp distinguishing of all detected faults. In other words, if we create the minimum number of sensors in the configuration, the origin (sensor or system) of any detected fault by detection techniques can be distinguished via modified logic set rules. The following sections describe the procedure for accommodating uncertainty in the methodology.

4.3.1 Dealing with Uncertainty

The uncertainty makes the fault detection procedure challenging and indefinite. Many researchers have addressed this issue in control engineering community, and novel studies have investigated the robustness of detection. There are always tradeoffs in objectives of fault detection versus robustness, and optimal solution should be considered with respect to the physics and sensitivity of the problem.

Dealing with uncertainty, inconsistency and noise, often requires some type of weighted evidence, in addition to some numerical calculations. This might be represented in the form of probabilities, Bayesian analysis and defining noise parameters such as variance, covariance and distribution. Generally, combination of estimation techniques and optimal threshold selection leads to finest tuning of fault detection [18], [27], [28], [61]. The following sections briefly review these complementary topics.

4.3.2 Estimation

In the deterministic case, we use first principle functional relationships to generate ACS, while in case of presence of noise, these values are inherently uncertain. Therefore, instead of a definite value, ACS will be an estimate of corresponding variable.

Definition 4: The estimated values by functional relationships, using readings from sensors are called *Estimated Substitutions (ES)*. These values are uncertain analogous of ACS. For generation of ES, a form of estimation techniques should be used.

In other words, at sampling time *(j)*, where MMW covers three nodes, instead of generating six definite ACS ($\bar{a}^{b^1}(j)$, $\bar{b}^{a^1}(j)$, $\bar{b}^{a^2}(j)$, $\bar{b}^{c^1}(j)$, $\bar{b}^{c^2}(j)$, $\bar{c}^{b^1}(j)$) term, we will have ES in the form of $\hat{p}^{q^i}(j)$, where $\hat{p}$ corresponds to node p, and is estimated using the measurement q^i. Figure 4.6 illustrate the relationships for generating ES corresponding to each node.

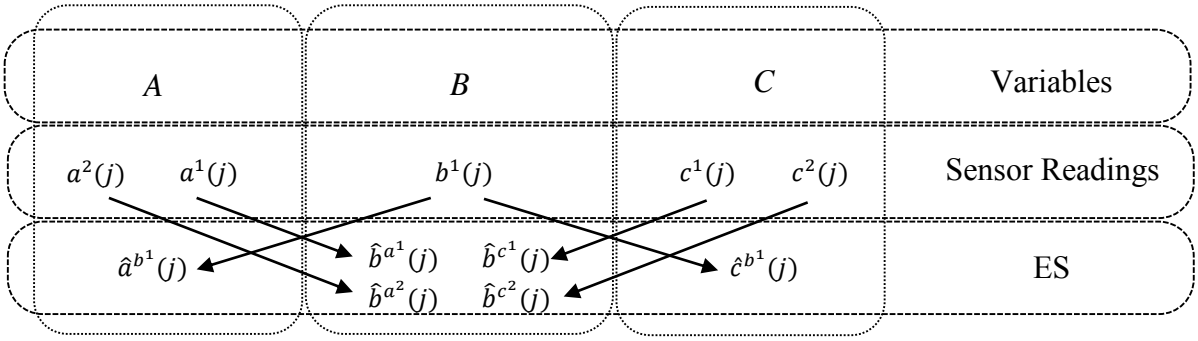


Figure 4.6: Variables, sensor readings, and ES at time *(j)*, in presence of uncertainty

A number of methods can be employed for generating ES, such as likelihood, least square error estimation, Kalman filtering and Bayesian estimate [68]. Bayesian model is stated in probabilistic terms. The probabilistic foundations of this method allow for incorporating and propagating uncertainties. The Kalman filter (KF) is an optimal recursive estimator, developed by R. Kalman. We can use KF or one if its extensions (e.g. Extended/ Unscented Kalman filter for nonlinear systems), in order to estimate the state of system and then generate ES. The details about this filter can be found in [69], [70]. The Kalman filter balances the model uncertainty with measurement uncertainty to come up with an optimal estimate. The a priori information for the

KF is the discretized system dynamic (equation (2.4)) and noise property of the system and sensor which can be empirically derived from historical data. Figure 4.7 illustrates a short review of the algorithm.

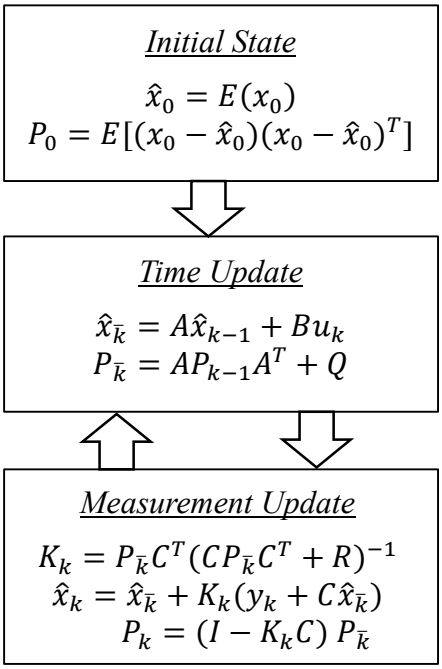


Figure 4.7: Kalman filter algorithm, reproduced based on [71]

A combination of known deterministic physical relationships with estimation techniques provides redundant values (ES), which are used for checking the credibility of measurements, instead of ACS in deterministic case.

4.3.3 Threshold Selection

Upon generation of ES for a variable, the sensor reading and its corresponding ES are not necessarily equal in fault free condition, because of associated uncertainty in ES. Additionally, even two similar sensors for measuring a variable do not have exactly equal readings, due to the effects of measurements noise in sensors. Hence, for accommodation of uncertainty, *Thresholds*

should be employed in the logic set. These thresholds are representatives of valid regions. If differences of ES with sensor measurement or readings from essential and redundant sensors fall within these regions, they are considered equal, otherwise, it is indicative a discrepancy. Tuning the thresholds is an iterative process based on the experience of the designer. Setting an excessively low threshold increases FAR and selecting a high one increases MAR.

For this purpose, all equality and inequalities in Table 3.3, where we defined the logic set, should be changed into valid and invalid regions by using the definition of thresholds. For instance, within a MMW if $|a^2 - a^1| < r_1$, $\left|\hat{b}^{a^1} - b^1\right| < r_2$, $\left|\hat{c}^{b^1} - c^1\right| < r_3$, $\left|\hat{c}^{b^1} - c^2\right| < r_4$, and $|c^1 - c^2| < r_5$, then all sensors and system components are healthy and there is not fault. All other rows of Table 3.3 (logic set) should be modified via ES and threshold accordingly, in order to deal with uncertainty. The diagnosis and distinguishing sensor and system faults, will be fully maintained as long as the minimum redundancy is satisfied.

Chapter 5

Conclusions

5.1 Summary

Fault detection and diagnosis (FDD) is a key element of operation and management of automated systems to increase reliability and safety. There is a high demand for the development of diagnostic systems that are capable of autonomous detection of anomaly presence as well as localization of the faults that may occur in different components of a complex dynamic system while in operation. Automated FDD systems depend entirely on sensor readings, since they are the monitoring interface of the system. An unanticipated deviation in a sensor's readings from its expected values under specific operating conditions may not necessarily be a fault in the sensor mechanism itself, but may be a symptom of a more serious potential fault in the monitored system. Hence, system and sensor faults might be manifested with the same symptoms. Present FDD schemes only consider either the malfunction of system, assuming that the sensing system

is functioning normally, or sensor failure while the system is fault free. The ability to identify the source of faults is crucial in the monitoring of a system, as different corrective actions or compensatory responses are required in case of sensor or system faults. Despite the importance for practical application of diagnostic schemes, distinguishing between sensor and system faults does not appear to have received a substantial amount of prior attention in monitoring and diagnosis literature, while it is a challenging fundamental issue.

In this study we develop a framework to address this problem for a certain class of systems, which have serial causality between their variables. At first, it is clarified that by strict duplication of sensor elements, it is feasible to differentiate between sensor and system faults, assuming that the probability of concurrent faults in the essential and the redundant sensors is zero. If the readings at a particular instant have discrepancy, it is a strong indication of a fault in either sensor. Duplication of sensors for the monitored variables can determine the status of sensor/system, however, redundant sensors are not always practical, due to considerations of weight, cost, space constraints, electrical/power constraints, and increased complexity.

Hence, by aiming to employ minimum redundant sensors, *a priori* knowledge of physical relationships (functional redundancy) between monitored variables is used to check the credibility of existing sensor observations. In this methodology, the system variables are modeled with serially connected causal network. Then the concept of Moving Monitoring Window (MMW) is introduced, which covers three nodes at the same time, as it traverses through the nodes of the system in the direction of causality. Two edge nodes must have duplicated sensors, while the middle one is monitored with only one sensor. By using the physical relationships, the five sensor measurements at each sampling point in MMW can generate total six Analytical Computational Substitutions (ACS). Measurements and ACS are

processed in Logic Set Unit. This unit consists of all system/sensor state possibilities, which are called System Behavioral Modes. It is designed offline in form of a bank of knowledge-based rules. The parametric design of the logic set allows decision-making on the health status of sensor or system by comparing the on-line generated ACS and sensor readings at each sampling point.

In addition to detection of faults and anomalies, this unit can act as the distinguisher and differentiates between sensor faults and system faults. Moreover, the proposed diagnostic procedure can command for safe shutting down or continuing the operation of system in degraded mode, upon the occurrence of different faults. The generalization by deduction reveals the degree of redundancy for larger number of variables, as long as serial causality is valid between monitored variables. If the number of sensors is greater than 1.5 times of the number of monitored variables, the task of distinguishing between sensor and system faults can be done effectively and successfully. Removing any more sensors from this configuration leads to inability to locate and differentiate between sensor and system faults, due to the lack of adequate behavioral modes for diagnosis decision.

The underlying nature of the methodology provides the ground for tolerance of some sensor faults. Although, some modes of multiple faults can be successfully managed, a shortcoming of methodology is that in some cases of several concurrent faults, the localization is not possible and diagnostic system commands for shutting down. This is due to the missing information of corresponding state possibilities, which is the consequence of removing sensors from full duplication configuration. The permutations of sensor culling degree of freedom (DOF), when the design restrictions urge some variables to be single-sensor, are also analyzed

The effectiveness of approach is exemplified and verified on a system of interconnected multi reservoirs and control valves. The extension of work for accommodating noise and uncertainty by estimation and threshold selection is also explained at the end, which can be further investigated in future works.

5.2 Contributions

The major contribution of this thesis in the field of fault diagnosis is alluded to rather frequently throughout the manuscript. This study opens the way to better understanding of the criteria needed for distinguishing between sensor faults and system faults in a controlled system, in order to make attributable diagnostic decision with respect to the type of the fault. To undertake this task, the following efforts have been done:

- Establishing the minimum degree of sensor redundancy,
- Defining the concept of MMW and the logic set structure.

This research is the first step in the perspective of the effective distinguishing of sensor fault from system faults.

5.3 Suggestions for Future Works

This research has opened a new area for exploration in the field of fault diagnosis, which has the capacity for more contributions and novel investigations. There are many ways to improve the proposed framework. In future works, the methodology can be extended in the following directions:

- Investigation of the method for non-serial network:

A thorough study of further general criteria that could be applied to parallel and combined serial-parallel system needs to be pursued in the future. Additionally, analysis of the method for diverse application domains is suggested.

- Theoretical evaluation of distinguishability criteria:

Although the approach leads to crisp distinguishing of sensor and system faults, a study of theoretical criteria would reinforce the method. The current generalization is based on deduction, since it is true for any number of variables, however, a mathematical derivation such as structural models may lead to a closed form distinguishability criteria, which makes a substantial contribution in this field.

References

[1] E. Sobhani-Tehrani and K. Khorasani, "Fault diagnosis of nonlinear systems using a hybrid approach," in *Lecture Notes in Control and Information Sciences*, vol. 383, M. Thoma, F. Allgöwer, and M. Morari, Eds. Springer Dordrecht Heidelberg, 2009, pp. 1–266.

[2] R. Isermann and P. Ballé, "Trends in the application of model-based fault detection and diagnosis of technical processes," *Control Eng. Pract.*, vol. 5, no. 5, pp. 709–719, May 1997.

[3] R. Isermann, "Process fault detection based on modeling and estimation methods: A survey," *Automatica*, vol. 20, no. 4, pp. 387–404, Jul. 1984.

[4] R. Isermann, "Model-based fault-detection and diagnosis: Status and applications," *Annu. Rev. Control*, vol. 29, no. 1, pp. 71–85, Jan. 2005.

[5] R. Isermann, "Supervision, fault-detection and fault-diagnosis methods: An introduction," *Control Eng. Pract.*, vol. 5, no. 5, pp. 639–652, May 1997.

[6] P. M. Frank, "Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy," *Automatica*, vol. 26, no. 3, pp. 459–474, May 1990.

[7] 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, no. 3, pp. 293–311, Mar. 2003.

[8] A. Einafshar, "Fault tolerant reconfiguration of multi-satellite interactions using high-level petri nets." University of British Columbia, 2015.

[9] J. Korbicz, J. Koscielny, Z. Kowalczuk, and W. Cholewa, *Fault Diagnosis: Models, Artificial Intelligence, Applications*. Springer Science & Business Media, 2012.

[10] F. A. Alrowie, "Fault Isolation and Alarm Design in Non-linear Stochastic Systems," University of British Columbia, 2015.

[11] "ANSI/ISA-18.2-2009 Management of Alarm Systems for the Process Industries."

[12] I. Izadi, S. Shah, D. Shook, and T. Chen, "An Introduction to Alarm Analysis and Design," in *Fault Detection, Supervision and Safety of Technical Processes*, 2009, pp. 645–650.

[13] Y. Seng Ng and R. Srinivasan, "Multi-agent based collaborative fault detection and identification in chemical processes," *Eng. Appl. Artif. Intell.*, vol. 23, no. 6, pp. 934–949, Sep. 2010.

[14] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri, "A review of process fault detection and diagnosis: Part II: Qualitative models and search strategies," *Comput. Chem. Eng.*, vol. 27, pp. 313–326, 2003.

[15] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri, "A review of process fault detection and diagnosis: Part III: Process history based methods," *Comput. Chem. Eng.*, vol. 27, pp. 293–311, 2003.

[16] R. Isermann, *Fault-Diagnosis Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.

[17] S. Simani, C. Fantuzzi, and R. J. Patton, "Model-based Fault Diagnosis in Dynamic Systems Using Identification Techniques," in *Advances in Industrial Control*, Springer Science & Business Media, 2010.

[18] J. Chen and R. J. Patton, "Robust model-based fault diagnosis for dynamic systems," in *Asian Studies in Computer and Information Science*, vol. 11, no. 14, K.-Y. Cai and C. Beijing University Of Aeronautics Beijing, Eds. Kluwer Academic Publishers, 1999, p. 356.

[19] J. Gertler, "Analytical redundancy methods in fault detection and isolation," in *IFAC/IMACS Symposium on Fault Detection, Supervision and Safety for Technical Processes*, 1992.

[20] J. Gertler, "Fault detection and isolation using parity relations," in *Control Engineering Practice*, 1997, vol. 5, no. 5, pp. 653–661.

[21] J. Gertler, *Fault Detection and Diagnosis in Engineering Systems*. Marcel Dekker, 1998.

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

[23] M. A. Massoumnia, "A geometric approach to failure detection and identification in linear systems." Massachusetts Institute of Technology, 1986.

[24] R. J. Patton, P. M. Frank, and R. N. Clark, *Fault Diagnosis for Dynamic Systems*. London: Springer-Verlag Ltd., 2000.

[25] P. M. Frank, S. X. Ding, and T. Marcu, "Model-based fault diagnosis in technical processes," *Trans. Inst. Meas. Control*, vol. 22, no. 1, pp. 57–101, Mar. 2000.

[26] P. M. Frank, "On-line fault detection in uncertain nonlinear systems using diagnostic observers: a survey," *Int. J. Syst. Sci.*, vol. 25, no. 12, pp. 2129–2154, Dec. 1994.

[27] R. N. Clark, “State estimation schemes for instrument fault detection,” in *Fault Diagnosis in Dynamic System: Theory and Application*, R. J. Patton and et al., Eds. Prentice Hall, 1989, pp. 21–45.

[28] R. N. Clark, “Instrument Fault Detection,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-14, no. 3, pp. 456–465, May 1978.

[29] M. Saif and W. Chen, “Observer-based strategies for actuator fault detection, isolation and estimation for certain class of uncertain nonlinear systems,” *IET Control Theory Appl.*, vol. 1, no. 6, pp. 1672–1680, Nov. 2007.

[30] Pau-Lo Hsu, Ken-Li Lin, and Li-Cheng Shen, “Diagnosis of multiple sensor and actuator failures in automotive engines,” *IEEE Trans. Veh. Technol.*, vol. 44, no. 4, pp. 779–789, 1995.

[31] L. Tang, X. Zhang, and J. DeCastro, “Diagnosis of engine sensor, actuator and component faults using a bank of adaptive nonlinear estimators,” *IEEE Aerosp. Conf. Proc.*, 2011.

[32] H. Noura, D. Theilliol, J.-C. Ponsart, and A. Chamseddine, *Fault-tolerant Control Systems: Design and Practical Applications*. Springer Berlin Heidelberg, 2009.

[33] P. M. Frank, E. Alcorta García, and B. Köppen-Seliger, “Modelling for fault detection and isolation versus modelling for control,” *Math. Comput. Simul.*, vol. 53, no. 4–6, pp. 259–271, 2000.

[34] E. Balaban, A. Saxena, P. Bansal, K. F. Goebel, and S. Curran, “Modeling, detection, and disambiguation of sensor faults for aerospace applications,” *IEEE Sens. J.*, vol. 9, no. 12, pp. 1907–1917, 2009.

[35] J. G. Webster, *The Measurement, Instrumentation, and Sensors: Handbook*. Springer Science & Business Media, 1999.

[36] J. S. Wilson, *Sensor Technology Handbook, Volume 1*. Newnes, 2005.

[37] A. Einafshar, B. Razavi, and F. Sassani, “Integrated Reconfiguration of Multi-Satellite Network Communication Using Colored Petri Nets,” in *Integrated Systems: Innovations and Applications*, Springer Berlin Heidelberg, 2015, pp. 3–28.

[38] G. Krishnamoorthy, “A Framework for Utilizing Data from Multiple Sensors in Intelligent Mechanical Systems,” University of Texas at Austin, 2010.

[39] M. A. Atoui, S. Verron, and A. Kobi, “A Bayesian network dealing with measurements and residuals for system monitoring,” *Trans. Inst. Meas. Control*, May 2015.

[40] M. Sepasi and F. Sassani, “On-line fault diagnosis of hydraulic systems using Unscented Kalman Filter,” *Int. J. Control. Autom. Syst.*, vol. 8, no. 1, pp. 149–156, Feb. 2010.

[41] S. Gayaka and B. Yao, “Fault detection, identification and accommodation for an electro-hydraulic system: An adaptive robusut appro,” *IFAC Proc. Vol.*, vol. 17, no. 1 PART 1, pp. 13815–13820, 2008.

[42] R. Tafreshi, F. Sassani, H. Ahmadi, and G. Dumont, “Local discriminant bases in machine fault diagnosis using vibration signals,” *Integr. Comput. Aided. Eng.*, vol. 12, no. 2, pp. 147–158, Apr. 2005.

[43] H. Ahmadi, G. Dumont, F. Sassani, and R. Tafreshi, “Performance of Informative Wavelets for Classification and Diagnosis of Machine Faults,” *Int. J. Wavelets, Multiresolution Inf. Process.*, vol. 01, no. 03, pp. 275–289, Sep. 2003.

[44] P. K. Kankar, S. C. Sharma, and S. P. Harsha, “Fault diagnosis of ball bearings using machine learning methods,” *Expert Syst. Appl.*, vol. 38, no. 3, pp. 1876–1886, Mar. 2011.

[45] B. Razavi, “Decision analysis models for aircraft engine maintenance planning using discrete event simulation.” University of British Columbia, 2015.

[46] B. Razavi, A. Einafshar, and F. Sassani, “Decision Analysis Model for Optimal Aircraft Engine Maintenance Policies Using Discrete Event Simulation,” in *Integrated Systems: Innovations and Applications*, Springer Berlin Heidelberg, 2015, pp. 69–87.

[47] C. L. Stork and B. R. Kowalski, “Distinguishing between process upsets and sensor malfunctions using sensor redundancy,” *Chemom. Intell. Lab. Syst.*, vol. 46, no. 2, pp. 117–131, 1999.

[48] M. Du, J. Scott, and P. Mhaskar, “Actuator and sensor fault isolation of nonlinear process systems,” *Chem. Eng. Sci.*, vol. 104, pp. 2940–303, 2013.

[49] J. Park, G. Rizzoni, and W. B. Ribbens, “On the representation of sensor faults in fault detection filters,” *Automatica*, vol. 30, no. 11, pp. 1793–1795, 1994.

[50] H. Wang, Z. J. Huang, and S. Daley, “On the use of adaptive updating rules for actuator and sensor fault diagnosis,” *Automatica*, vol. 33, no. 2, pp. 217–225, Feb. 1997.

[51] M. J. Hayes, R. Izadi-Zamanabadi, and S. M. Mahdi Alavi, “Robust fault detection and isolation technique for single-input/single-output closed-loop control systems that exhibit actuator and sensor faults,” *IET Control Theory Appl.*, vol. 2, no. 11, pp. 951–965, Nov. 2008.

[52] C. Hajiyev and F. Caliskan, “Sensor/actuator fault diagnosis based on statistical analysis of innovation sequence and Robust Kalman Filtering,” *Aerosp. Sci. Technol.*, vol. 4, no. 6, pp. 415–422, 2000.

[53] W. Xue, Y. Guo, and X. Zhang, “A Bank of Kalman Filters and a Robust Kalman Filter Applied in Fault Diagnosis of Aircraft Engine Sensor/Actuator,” in *Second International*

Conference on Innovative Computing, Informatio and Control (ICICIC 2007), 2007, pp. 10–10.

[54] M. Krysander and M. Nyberg, "Fault Isolability Prediction of Diagnostic Models," in *Proceedings of 16th International Workshop on Principles of Diagnosis DX-05, Pacific Grove, CA, USA.*, 2005.

[55] M. Krysander and E. Frisk, "Sensor Placement for Fault Diagnosis," *Syst. Man Cybern. Part A Syst. Humans, IEEE Trans.*, vol. 38, no. 6, pp. 1398–1410, 2008.

[56] A. Rosich, "Sensor Placement for Fault Detection and Isolation Based on Structural Models," in *Fault Detection, Supervision and Safety of Technical Processes*, 2012, vol. 8, no. 1, pp. 391–396.

[57] M. Bhushan and R. Rengaswamy, "Design of sensor location based on various fault diagnostic observability and reliability criteria," *Comput. Chem. Eng.*, vol. 24, no. 2–7, pp. 735–741, Jul. 2000.

[58] M. P. Henry and D. W. Clarke, "The self-validating sensor: rationale, definitions and examples," *Control Eng. Pract.*, vol. 1, no. 4, pp. 585–610, Aug. 1993.

[59] J. C. Yang and D. W. Clarke, "The self-validating actuator," *Control Eng. Pract.*, vol. 7, no. 2, pp. 249–260, Feb. 1999.

[60] J. C. Da Silva, A. Saxena, E. Balaban, and K. Goebel, "A knowledge-based system approach for sensor fault modeling, detection and mitigation," *Expert Syst. Appl.*, vol. 39, no. 12, pp. 10977–10989, 2012.

[61] S. Alag, A. M. Agogino, and M. Morjaria, "A methodology for intelligent sensor measurement, validation, fusion, and fault detection for equipment monitoring and diagnostics," *Artif. Intell. Eng. Des. Anal. Manuf.*, vol. 15, no. 4, pp. 307–320, 2001.

[62] T. Xu and Z. Feng, "Research on method for distinguishing sensor fault and system fault with PCA," *2010 Int. Conf. Meas. Technol. Mechatronics Autom. ICMTMA 2010*, vol. 3, pp. 112–114, 2010.

[63] N. Mehranbod, M. Soroush, M. Piovoso, and B. A. Ogunnaike, "Probabilistic Model for Sensor Fault Detection and Identification," *AIChE J.*, vol. 49, no. 7, pp. 1787–1802, Jul. 2003.

[64] J. Pearl, "Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference," in *Series in Representation and Reasoning*, R. Brachman, Ed. Morgan Kayfmann Publishers Inc, 1988.

[65] Z. Hameed, Y. S. Hong, Y. M. Cho, S. H. Ahn, and C. K. Song, "Condition monitoring and fault detection of wind turbines and related algorithms: A review," *Renew. Sustain. Energy Rev.*, vol. 13, no. 1, pp. 1–39, Jan. 2009.

[66] M. Labarrere, "Aircraft Sensor Failure Detection by Analytic Redundancy," in *Systems and Control Encyclopedia*, M. G. Singh, Ed. Oxford: Pergamon Press, 1987, pp. 246–251.

[67] R. Onken and N. Stuckenberg, "Failure detection in signal processing and sensing in flight control systems," in *1978 IEEE Conference on Decision and Control including the 17th Symposium on Adaptive Processes*, 1978, pp. 449–454.

[68] C. Brown, H. Durrant-Whyte, J. Leonard, B. Rao, and B. Steer, "Distributed data fusion using Kalman filtering: A robotics application," in *Data Fusion in Robotics and Machine Intelligence*, M. A. Abidi and R. C. Gonzalez, Eds. Boston: Academic Press, 1992, pp. 267–309.

[69] M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice*. Prentice-Hall, Inc., 1993.

[70] F. Pirmoradi, F. Sassani, and C. W. de Silva, "Health Monitoring of Mechatronic System," in *Mechatronic Systems: Devices, Design, Control, Operation and Monitoring*, C. W. de Silva, Ed. CRC Press, 2007.

[71] M. Sepasi, "Fault Monitoring in Hydraulic Systems using Unscented Kalman Filter," University of British Columbia, 2007.

[72] J. H. Richter and L. Jan, "H∞-based virtual actuator synthesis for optimal trajectory recovery," in *in Preprints of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SafeProcess'09)*, 2009, pp. 1587–1592.

[73] H. E. Merritt, *Hydraulic Control Systems*. John Wiley & Sons Inc., 1967.

[74] K. Ogata, *Modern Control Engineering*. Pearson, 2006.

Appendix A

Here the dynamics of interconnected multi reservoirs system is modeled and derived. The similar idea is presented in [72] for the purpose of H_∞ control. The liquid heights in each reservoir are described as controllable outputs, which are regulated by control valves, equipped with electrical actuators. It is assumed that the valves resistances are set in a way that the height of liquid in each reservoir is proportional to the required flow rate [73]. We derive the governing equation of each reservoir (tank) individually by writing the physical relationships, and then integrate them to a general form.

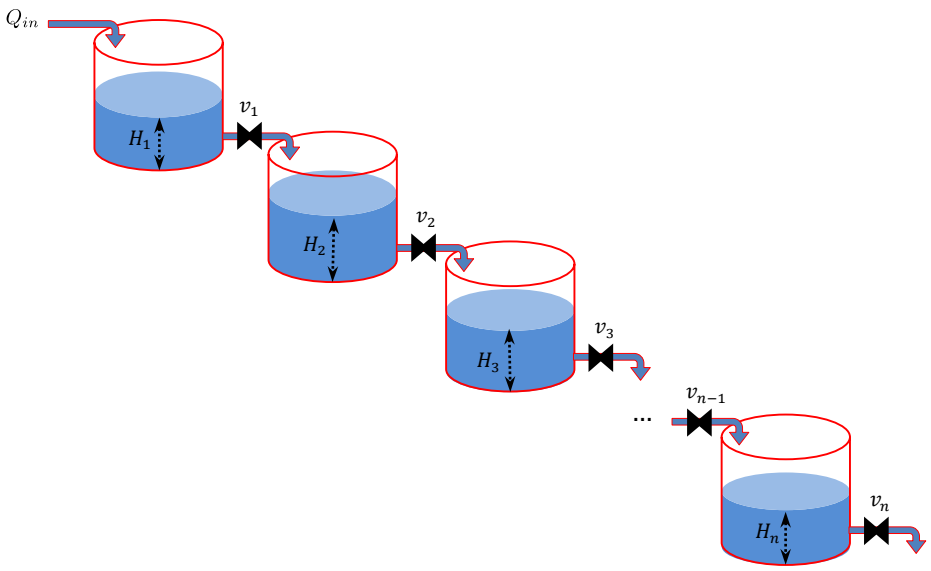


Figure A.1: A schematic view of interconnected multi reservoirs with overhead input flow

For the first tank $\dot{m}_1 = {d(\rho V_1)}/{dt}$, and $V_1 = A_1 H_1$, where $\dot{m}_1$ is the mass flow rate of the tank, V_1 is the volume of the tank, ρ is the liquid density, A_1 is the area, and H_1 is the height of the liquid. If we consider Q_{in} as the input flow rate to the tank one, and $q_1$as the output flow rate, the following equation indicates the governing equation for tank one:

$$\rho Q_{in} - \rho q_1 = \dot{m}_1 = \frac{d(\rho V_1)}{dt} = \rho A_1 \frac{d(H_1)}{dt} \tag{A.1}$$

Since we assume that the liquid is incompressible (water, oil, etc.), ρ can be eliminated from both side of equation:

$$Q_{in} - q_1 = A_1 \frac{d(H_1)}{dt} \tag{A.2}$$

In practice, the resistance of a valve is not constant and varies with the height of the upstream liquid. However, for modeling purposes, it is common to assume it constant and invariant with respect to the pressure behind it, unlike orifices [73]. Assuming that the resistance in the valve one (v_1) is the control input, the output flow rate of valve one can be written as $q_1 = {H_1}/{R_1}$, which means that the flow rate of the valve is proportional to the height of the liquid in the reservoir. Therefore, the governing equation for the first reservoir can be written as:

$$Q_{in} - \frac{H_1}{R_1} = A_1 \dot{H}_1 \tag{A.3}$$

Hence, the nonlinear physical equation for tank one at each sampling instant can be described as:

$$\dot{H}_1(k) = \frac{Q_{in}(k)}{A_1} - \frac{H_1(k)}{A_1 R_1(k)} \tag{A.4}$$

This equation describes the variation of height of the liquid with respect to input flow rate ($Q_{in}(k)$) and control input ($R_1(k)$). By having these known values and initial condition, the variation of height can be calculated at each sampling time.

Similar analysis can be done for the other consecutive reservoirs. For example, for tank two, assuming the mass conservation in the system, the input flow rate is q_1 (or output flow rate of tank one) and q_2 is the output flow rate. Therefore:

$$q_1 - q_2 = A_2 \dot{H}_2 \tag{A.5}$$

Since the valves are similar, the characteristics of their resistance are similar too, hence $q_2 = {H_2}/{R_2}$, so the governing equation can be written as:

$$\dot{H}_2 = \frac{H_1}{A_2 R_1} - \frac{H_2}{A_2 R_2} \tag{A.6}$$

It is clear that the form of state equation is constant for all tanks. Hence, for tank number n with the input flow rate q_{n-1} (or output flow rate of tank $n-1$) and q_n the output flow rate, $q_{n-1} - q_n = A_n \dot{H}_n$. Assuming that $q_n = {H_n}/{R_n}$:

$$\dot{H}_n = \frac{H_{n-1}}{A_n R_{n-1}} - \frac{H_n}{A_n R_n} \tag{A.7}$$

This form is the general equation of each in the series. Therefore, the nonlinear state equation of the system can be written as:

$$\begin{cases} \dot{H}_1 = \dfrac{Q_{in}}{A_1} - \dfrac{H_1}{A_1 R_1} \\ \dot{H}_2 = \dfrac{H_1}{A_2 R_1} - \dfrac{H_2}{A_2 R_2} \\ \vdots \\ \dot{H}_n = \dfrac{H_{n-1}}{A_n R_{n-1}} - \dfrac{H_n}{A_n R_n} \end{cases} \tag{A.8}$$

Therefore, the state space equation of the system can be written as follow:

$$\begin{cases} \dot{\mathbf{X}} = \dot{\mathbf{H}} = \begin{bmatrix} \dot{H}_1 \\ \dot{H}_2 \\ \vdots \\ \dot{H}_n \end{bmatrix} = \mathrm{G}(\mathbf{H}, \mathbf{R}) \\ \mathbf{Y} = \boldsymbol{I}\mathbf{H} \end{cases} \tag{A.9}$$

where **H** is the state vector, $\dot{\mathbf{H}}$ represents the variation in heights of the tanks, **R** is the control input representing resistances of control valves, G is a nonlinear function, **Y** is the system output vector, and ***I*** is an appropriate order identity matrix.

It should be noted that the architecture of connection makes no essential difference in the dynamic modeling and causality. As shown in Figure A.2, the bottom valve between two tanks causes the flow rate to be dependent to liquid heights in both tanks, whereas in overhead valve architecture, it is independent of the liquid height in the consecutive tank.

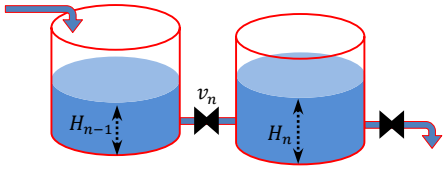


Figure A.2: The multi reservoirs system with a bottom valve

Hence, the output flow rate in v_n can be written as:

$$q_n = \frac{H_{n-1} - H_n}{R_n} \tag{A.10}$$

It is apparent that the model is nonlinear. The nonlinearity of these equations can be linearized around the operating zone based on the Taylor series expansion. An operating point is usually defined as an equilibrium point. The obtained linearized model corresponds to the relationship between the variation of the system output and the variation of the system input around this operating point. The principals of linearizing with Taylor series are given in [74].

Since the dynamics of tanks are similar, for simplicity we study a system with three reservoirs and derive the linearized state space equation. Let us consider that exactly at the steady state operating point for tank one: $Q_{in} = Q_{in_0}$, $H_1 = H_{1_0}$, $R_1 = R_{1_0}$, where superscript

'0' stands for 'Operating Point'. Around the operating zone, the system variables can be considered as: $Q_{in} = Q_{in_0} + q_{in}(t)$, $H_1 = H_{1_0} + h_1(t)$, $R_1 = R_{1_0} + r_1(t)$, and $\dot{H}_1 = \frac{Q_{in}}{A_1} - \frac{H_1}{A_1 R_1}$, then based on the Taylor series expansion using the first term of time derivate, the state equation can be written as: (omitting time function (t) for simplicity)

$$\frac{dh_1}{dt} = \frac{-1}{A_1 R_{1_0}} h_1 + \frac{1}{A_1} q_{in} + \frac{H_{1_0}}{A_1 R_{1_0}^2} r_1 \tag{A.11}$$

This can be considered as a state equation for tank one. Similar derivation can be done for tank two and three:

$$\frac{dh_2}{dt} = \frac{1}{A_2 R_{1_0}} h_1 + \frac{-1}{A_2 R_{2_0}} h_2 + \frac{-H_{1_0}}{A_2 R_{1_0}^2} r_1 + \frac{H_{2_0}}{A_2 R_{2_0}^2} r_2 \tag{A.12}$$

$$\frac{dh_3}{dt} = \frac{1}{A_3 R_{2_0}} h_2 + \frac{-1}{A_3 R_{3_0}} h_3 + \frac{-H_{2_0}}{A_3 R_{2_0}^2} r_2 + \frac{H_{3_0}}{A_3 R_{3_0}^2} r_3 \tag{A.13}$$

Since the general form of a state space equation is $\begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{Ax} + \boldsymbol{Bu} \\ \boldsymbol{y} = \boldsymbol{Cx} + \boldsymbol{Du} \end{cases}$, we can write a linearized state space equation of the system based on these:

$$\begin{cases} \dot{x} = \begin{bmatrix} \dot{h}_1 \\ \dot{h}_2 \\ \dot{h}_3 \end{bmatrix} = \begin{bmatrix} \frac{-1}{A_1 R_{1_0}} & 0 & 0 \\ \frac{1}{A_2 R_{1_0}} & \frac{-1}{A_2 R_{2_0}} & 0 \\ 0 & \frac{1}{A_3 R_{2_0}} & \frac{-1}{A_3 R_{3_0}} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} + \begin{bmatrix} \frac{1}{A_1} & \frac{H_{1_0}}{A_1 R_{1_0}^2} & 0 & 0 \\ 0 & \frac{H_{1_0}}{A_1 R_{1_0}^2} & \frac{H_{2_0}}{A_2 R_{2_0}^2} & 0 \\ 0 & 0 & \frac{-H_{2_0}}{A_3 R_{2_0}^2} & \frac{H_{3_0}}{A_3 R_{3_0}^2} \end{bmatrix} \begin{bmatrix} q_{in} \\ r_1 \\ r_2 \\ r_3 \end{bmatrix} \\ y = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} \end{cases} \tag{A.14}$$