2010

A Unified Method for Detecting and Isolating Process Faults and Sensor Faults in Nonlinear Systems

Niharika Sonti

Wright State University

Follow this and additional works at: http://corescholar.libraries.wright.edu/etd_all

Part of the Electrical and Computer Engineering Commons

Repository Citation


This Thesis is brought to you for free and open access by the Theses and Dissertations at CORE Scholar. It has been accepted for inclusion in Browse all Theses and Dissertations by an authorized administrator of CORE Scholar. For more information, please contact corescholar@www.libraries.wright.edu.
A Unified Method for Detecting and Isolating Process Faults and Sensor Faults in Nonlinear Systems

A thesis submitted in partial fulfillment of the requirements for the degree of
Master of Science in Engineering

By

Niharika Sonti
B.Tech., Jawaharlal Nehru Technological University, 2008

2010
Wright State University

________________________

Xiaodong Zhang, Ph.D.
Thesis Director

________________________

Kefu Xue, Ph.D.
Chair, Department of Electrical Engineering
College of Engineering and Computer Science

Committee on Final Examination

________________________

Xiaodong Zhang, Ph.D.

________________________

Kuldip S. Rattan, Ph.D.

________________________

Pradeep Misra, Ph.D.

________________________

Andrew T. Hsu, Ph.D.
Dean, School of Graduate Studies
Abstract

A Unified Method for Detecting and Isolating Process Faults and Sensor Faults in Nonlinear Systems

With the increase in the complexity of control systems design and the demand for more productivity, the possibility of the occurrence of faults in control systems has also significantly increased. In this thesis, a unified method for the fault diagnosis of sensor faults and process faults is developed for a class of Lipschitz nonlinear uncertain systems. The fault detection and isolation (FDI) architecture is comprised of a fault detection estimator and a bank of fault isolation estimators (FIEs), where each FIE is designed, based on the functional structure of a particular fault, in the fault class under consideration. The output residuals are generated, and adaptive thresholds are designed for the detection and isolation of the faults. The effectiveness of the fault detection and isolation algorithm is illustrated by a simulation example of single-link robotic arm. Extensive simulation studies have been conducted using Matlab/Simulink. Based on the nature of the residuals and their corresponding adaptive thresholds, the faults under consideration are successfully detected and isolated.

Advisor: Dr. Zhang
# Table of Contents

1. **Introduction** .................................................................................................................. 1  
   1.1 Fault Diagnosis ............................................................................................................. 1  
   1.2 Fault Diagnosis Methods .............................................................................................. 5  
      1.2.1 Model-Based Methods: .......................................................................................... 5  
      1.2.2 Model-Free Methods: .......................................................................................... 7  
   1.3 Research Motivation ..................................................................................................... 8  
   1.4 Thesis Organization ...................................................................................................... 10  

2. **Formulation of the problem** ............................................................................................. 12  

3. **Architecture of Fault Detection and Isolation** ............................................................... 18  
   3.1 Fault Detection Components.......................................................................................... 19  
   3.2 Fault Isolation Components .......................................................................................... 19  

4. **Design of Fault Detection Estimator** ........................................................................... 22  
   4.1 Fault Detection Estimator .............................................................................................. 22  
   4.2 Adaptive Threshold Design for Fault Detection .......................................................... 22  

5. **Design of Fault Isolation Estimators** .......................................................................... 26  
   5.1 Fault Isolation Estimators for Sensors Faults .............................................................. 26  
   5.2 Fault Isolation Estimators for Process Faults ............................................................. 27  
   5.3 Fault Isolation Decision Scheme for sensor and process faults ................................... 28
6.  Simulation Example and FDI Implementation ........................................... 30
   6.1 System Dynamics and Fault Model .......................................................... 30
   6.2 FDI Design ............................................................................................... 34
   6.3 Simulink Model Implementation ............................................................... 36

7.  Simulation Results .......................................................................................... 40
   7.1 Sensor fault in $y_2$ .................................................................................... 41
   7.2 Sensor fault in $y_3$ .................................................................................... 44
   7.3 Actuator fault .............................................................................................. 47
   7.4 Fault leading to extra abnormal friction in the motor ................................ 49

8.  Conclusions and Future work ........................................................................ 53

9.  References ....................................................................................................... 55
List of Figures

Figure 1: System performance affected by faults [30]........................................................................ 1
Figure 2: Fault diagnosis and control loop .......................................................................................... 2
Figure 3: Two Tank system [40]....................................................................................................... 3
Figure 4: The procedure of residual generation .................................................................................. 6
Figure 5: The Architecture of the Fault Detection and Isolation Scheme [39]................................. 18
Figure 6: Adaptive threshold vs. fixed threshold [30]........................................................................ 25
Figure 7: The Simulink model of the FDI algorithm for the single-link robotic arm.......................... 37
Figure 8: The block diagram of the Fault Isolation block .................................................................. 39
Figure 9: Sensor fault in \( y_2 \): The plots of the FDE and the FIEs for the remaining fault cases (sensor fault #2, process fault #1 and process fault #2) with the chosen residual component (solid line) and its corresponding threshold (dotted line) ................................................................. 41
Figure 10: Sensor fault in \( y_2 \): The plots of the FIE for the sensor fault #1, with the three residual components (solid line) and their corresponding thresholds (dotted line)................................................................. 42
Figure 11: The case of sensor fault in \( y_2 \) (sensor fault #1): the fault magnitude estimation for fault isolation estimator #1 ........................................................................................................................................ 43
Figure 12: Sensor fault in \( y_3 \): The plots of the FDE and the FIEs for the remaining fault cases (sensor fault #1, process fault #1 and process fault #2) with the chosen residual (solid line) and its corresponding threshold (dotted line) .............................................................................................................. 44
Figure 13: Sensor fault in \( y_3 \): The plots of the FIE for the sensor fault #2, with the three residual components (solid lines) and their corresponding thresholds (dotted lines)................................................................. 45
Figure 14: The case of sensor fault in \( y_3 \) (sensor fault #2): the fault magnitude estimation for fault isolation estimator #2........................................................................................................................................ 46
Figure 15: Actuator fault in \( y_3 \): The plots of the FDE and the FIEs for the remaining fault cases (sensor fault #1, sensor fault #2 and process fault #2) with the chosen residual (solid line) and its corresponding threshold (dotted line)................................................................. 47

Figure 16: Actuator fault in \( y_3 \): The plots of the FIE for the process fault #1, with the three residual components (solid lines) and their corresponding thresholds (dotted lines)......................... 48

Figure 17: The case actuator fault in \( y_3 \) (process fault #1): the fault magnitude estimation for fault isolation estimator #1........................................................................................................................................................................ 49

Figure 18: Process fault #2 in \( y_3 \): The plots of the FDE and the FIEs for the remaining fault cases (sensor fault #1, sensor fault #2 and process fault #1) with the chosen residual (solid line) and its corresponding threshold (dotted line)........................................................................................................... 50

Figure 19: Process fault #2 in \( y_3 \): The plots of the FIE for the process fault #2, with the three residual components (solid lines) and their corresponding thresholds (dotted lines)............... 51

Figure 20: The case of fault leading to extra abnormal friction in the motor (process fault #2): the fault magnitude estimation for fault isolation estimator #4. ................................................................. 52
1. Introduction

1.1 Fault Diagnosis

Any modern control system can be considered as a technical process. Each process is designed to produce a certain desired output or perform a certain desired function. However, ideal behavior may not always be exhibited by a system during its service life; some of the system components may degrade, which leads to variation from its normal behavior. These variations or anomalies, lead to undesirable effects in the system, often resulting in inaccurate outputs or even catastrophic effects. The causes for these anomalies in the functioning of a system, as a result of the degradation of system components, are termed as ‘faults’ as shown in Fig. 1 [30]. Fault diagnosis is very essential to ensure the reliable and smooth operation of modern control systems.

Figure 1: System performance affected by faults [30]
The block diagram of a typical control system, shown in Fig. 2, illustrates that a control system can be divided into three parts – the actuators, the process and the sensors. Therefore, depending on which system component is affected, the faults affecting the system can also be classified as [1, 2, 4, 14, 30] –

- **Actuator faults** – These kinds of faults are generally caused when the actuator in the system starts to function improperly.
- **Process faults** – These kinds of faults are caused when the existing relation between the inputs and outputs is altered due to changes in the system.
- **Sensors faults** – These kinds of faults are caused due to the errors in the measurement of the outputs of the system.

![Figure 2: Fault diagnosis and control loop](image)

The block diagram shown in Fig. 2 can be understood by considering a two tank system as shown in Fig. 3 [40].
In the two tank system shown in the figure above [40], water is pumped into the tank 1 through the pump P. The pump P is provided with an override to prevent the tank 1 from overflowing. The rate at which water flows into tank 1 is \( q_P \). There is a leakage in tank 1 that is represented by \( q_L \). The water level in tank 1 is measured by the sensor \( h_1 \). The tank 1 is connected to tank 2. In between the two tanks, there are two valves, \( V_a \) and \( V_{12} \). The level of water in the two tanks is controlled by the valve \( V_{12} \). Under normal conditions the auxiliary valve \( V_a \) is closed. In order to open the auxiliary valve \( V_a \), the valve \( V_{12} \) has to be closed. The rate at which water flows out of tank 2 is \( q_2 \). The level of water in tank 2 is measured by the sensor \( h_2 \).

There are three types of faults that can occur in this system. Then the possible faults that can occur in this two tank system can be grouped in the following manner –
• Actuator faults – As mentioned previously, Actuator faults are generally caused when the actuators in the system start to function improperly. In the above two tank example, an actuator fault may be caused by the improper functioning of the pump [40].

• Process faults – When certain changes occur in the system, the dynamic relation between the input and output is also altered. Then, the existing relation between the inputs and the outputs no longer hold, which leads to the occurrence of process faults. For instance, in the two tank system detailed above [40], when there is a leakage in tank 1 or when there is a blockage in the valve $V_{12}$, the equations describing the system dynamics have to be changed, to take into account the effect of the leakage and the blockage on the levels of water in the two tanks.

• Sensors faults – Sensor faults are caused due to the corrupted sensor measurement of the outputs of the system. In the two tank system example given above [17, 40], sensor faults are caused when there is bias in the measurement of the sensors $h_1$ and $h_2$.

As the design of modern control systems becomes more complex, the possibility of the occurrence of faults in the system also increases. In order to improve the efficiency and the performance of a system, a suitable system component must be designed, to monitor the system for the occurrence of faults. The process, of identifying the presence of a fault and its location and parameters, is termed as ‘fault diagnosis’. The procedure of fault
diagnosis involves – fault detection, fault isolation, and fault identification [11, 30, 39, 38, 37].

- Fault Detection
  The first step in fault diagnosis is to determine whether a fault has occurred in the system or not.

- Fault Isolation
  After it has been determined that a fault has occurred in the system, the next step is to find out the location of the fault.

- Fault Identification
  The next step after fault isolation is to estimate the fault magnitude of the fault that has occurred in the system.

1.2 Fault Diagnosis Methods

There are different methods of fault diagnosis, such as observer-based methods [11, 13, 30], parameter estimation [7, 13, 30] and neural networks approach [1, 2, 30] etc. These methods can be broadly classified into two categories – model-based and model-free methods.

1.2.1 Model-Based Methods:

The model-based methods of fault diagnosis involve the comparison of the system output obtained from measurements, with the output information obtained from the
mathematical model of the system under the same conditions. The difference between the actual and estimated outputs is termed as ‘residual’. The residual is then analyzed to determine the nature and the type of the fault leading to fault isolation.

![Figure 4: The procedure of residual generation](image)

More specifically, model-based fault diagnosis methods consist of the following components (as shown in the Fig. 4)

- **Generation of Residual** – In this stage, a residual is generated to determine whether or not a fault has occurred. Under ideal conditions, when there is no fault condition occurring in the system, the value of the residual is zero. But once a fault occurs, the value of the residual is significantly deviant from zero. There are many different methods of residual generation such as the observer-based approach (see, for instance, [11, 13, 30]), parity relation based approach (see, for instance, [1, 30]), fault detection filter (see, for instance, [1, 4]) etc.

- **Evaluation of the generated Residual** – Once a residual is generated; it is analyzed to determine whether or not a fault has occurred in the system. There are different
methods used to decide whether the residual indicates the occurrence of fault or not. Some of them include, applying a threshold to the residual values [1, 4], comparing the nature of the residuals to patterns of known faults etc. One residual is usually sufficient for the purpose of fault detection. For fault isolation, however, a set of residuals is needed [4, 33].

1.2.2 Model-Free Methods:

In order to use model-based methods for fault diagnosis, a mathematical model of the system is required. But, it may not always be possible to represent a system in terms of an accurate mathematical model, due to the inherent complexity of the given system. Also, it may not be possible to model all the faults, such that their characteristics are mathematically represented. In such cases, model-free methods are used. Some of the model-free methods used are – limit checking [14, 11], trend checking [14], Principal Component Analysis (see, for instance, [14, 11]).

Limit checking involves the monitoring of all the measured variables in a system and subsequently comparing their absolute values with some preset threshold values [14, 11]. It is one of the most simple and the most commonly used method. Generally, there are two sets of threshold values, a maximum value and a minimum value. When the absolute value of a measured variable lies between these two values, it means that the system is functioning normally.
Another model-free method that is used commonly is the PCA method. The acronym PCA stands for Principal Component Analysis. Principal Component Analysis involves the linear transformation of an original set of correlated variables, into a new set of uncorrelated variables [14, 33, 11]. These new set of uncorrelated variables are known as ‘principal components’. The main aspect of the PCA method is reducing the dimensionality of the given problem, while preserving the variation existing in the original set of variables. Initially a PCA model is generated, from the data collected when the system is in a steady state. This initial data is also used to calculate the limits for certain statistical measures – $T^2$ statistics and the $Q$ statistic (also known as the Square Prediction Error) [14, 33, 11]. Then, in order to monitor the system, it is sufficient to monitor the statistical measures of the current observations and there is no longer a necessity to keep track of several variables simultaneously. A fault is said to occur in the system, when the statistic measures of the current observation surpass the limiting values that have been calculated previously.

1.3 Research Motivation

Many conventional methods described above, which are used for fault diagnosis, are based on linear system models. But most of the systems encountered practically, are nonlinear systems. Therefore it is necessary to develop techniques for fault diagnosis in nonlinear systems. A lot of research has been conducted, in the area of fault diagnosis and fault accommodation of nonlinear systems [1, 2, 9, 11, 14, 8, 10, 12, 18, 19, 20, 29, 32]. One of the methods for fault diagnosis is based on adaptive and learning techniques [15, 16, 21, 25, 26, 27, 29].
One of the essential areas of fault diagnosis is sensor validation \cite{28, 34, 22, 3}. A fault in the output of a sensor may lead to certain undesirable situations, such as improper regulation and instability of the system. Another area of importance in fault diagnosis is the detection and accommodation of process faults, which are the faults that occur within the system. But, in many nonlinear fault diagnostic methods, process faults and sensor faults are dealt such that, at any given time, only type of fault case is assumed to occur in the system. To be more specific, while dealing with process faults \cite{35, 37}, one of the main assumptions is that there are no sensor faults occurring in the system, and while dealing with sensor faults \cite{38}, the assumption is that there are no process faults in the system. The drawback with this approach is that, when dealing with practical cases, a process fault may be misdiagnosed as a sensor fault and vice versa. This may lead to incorrect fault accommodation and also cause a false alarm to be raised. Therefore, in order to achieve accurate fault diagnostics, it is essential that both the sensors and the system components be monitored using a unified method.

In a previous paper \cite{39}, a unified fault diagnosis method for sensor faults and process faults was developed for a class of nonlinear uncertain systems, in which the known nonlinearity is represented as a function of known system signals (i.e. measurable input and output signals). However in many nonlinear systems, the nonlinearity is a function of partially measurable system states. Therefore it is necessary to extend the fault diagnostic method by considering nonlinear systems, whose nonlinearity is a function of the input and the partially measured states.
Based on the research motivation, the aims of this thesis are as follows

1. To develop a unified framework for the detection and isolation of sensor faults and process faults in nonlinear systems.

2. To apply the diagnostic algorithm to an application example and conduct the FDI design.

3. To develop a Simulink model of the fault diagnosis system and conduct the simulation studies to verify the effectiveness of the FDI algorithm.

1.4 Thesis Organization

This thesis is organized into the following chapters.

Chapter 1 is an introduction to fault diagnosis and some of the terms and concepts involved. Some of the previous research done in this area is also mentioned and an overview of the thesis is given.

Chapter 2 details the problem formulation for sensor and process fault diagnosis, for Lipschitz nonlinear systems. This chapter also contains the various assumptions made and the conditions that are required to be met, while designing the fault detection estimator and the fault isolation estimators.

Chapter 3 describes the procedure of fault modeling. In this paper, two types of faults – sensor faults and process faults are considered. The fault detection scheme using model-based methods is explained. Subsequently, the fault isolation estimator design for the sensor faults and process faults is described.
Chapter 4 contains the fault detection estimator design. The design of the threshold that is used to detect the fault condition is explained.

Chapter 5 explains the design of the fault isolation estimators that are used to isolate and estimate the faults introduced in the single-link robotic arm.

Chapter 6 presents the simulation example for the case of a single-link robotic arm with a revolute elastic joint. Two types of faults – sensor faults and process faults are introduced into the system. Then, the process of fault detection, fault estimation and fault isolation, for each fault is illustrated.

Chapter 7 consists of the results and graphs obtained from the simulation example given in chapter 6.

Chapter 8 details conclusions drawn from the simulation example that has been given chapter 6 and the scope of future research in this topic.
2. Formulation of the problem

We consider of a class of nonlinear multi-input-multi-output (MIMO) dynamic systems is described by

\[
\begin{align*}
\dot{x} &= Ax + D\zeta(x, u) + g(y, u) + \varphi(x, u, t) + \beta(t - T_x)f(y, u) \\
y &= Cx + \beta(t - T_y)F\theta
\end{align*}
\] (1)

where \(x \in \mathbb{R}^n\) is the state vector, \(u \in \mathbb{R}^m\) is the input vector, \(y \in \mathbb{R}^p\) is the output vector.

The known nominal model is represented by

\[
\begin{align*}
\dot{x}_N &= Ax_N + g(y_N, u) + D\zeta(x_N, u) \\
y_N &= Cx_N
\end{align*}
\] (2)

where the matrices \(D \in \mathbb{R}^{n \times \nu}\) and \(C \in \mathbb{R}^{p \times n}\) are of full rank and \((A, C)\) is an observable pair.

The modeling uncertainty is represented by vector \(\varphi\) in (1), the term \(\beta(t - T_x)f(y, u)\) represents the change in the system dynamics due to a process fault, and the term \(\beta(t - T_y)F\theta\) represents the change in system dynamics due to a sensor fault. In particular, the function \(\beta(t - T_x)\) and the function \(\beta(t - T_y)\) are the fault time profiles of the process fault and the sensor fault respectively with \(T_x\) being the instant of time at which the process fault occurs and \(T_y\) being the instant of time at which the sensor fault occurs. The function \(f(y, u)\) specifies the effect of the process fault [37]. Similarly the function \(F\) is the fault distribution matrix, which specifies the location of the sensor fault, while the scalar \(\theta\) represents the magnitude of the constant sensor bias [38].
Assumption 1 [37, 38]. The matrix $D$ in (1) satisfies the conditions,

(i) The rank $(CD) = \nu$,

(ii) All the invariant zeros of $(A,D,C)$ (if they exist) lie in the left half plane (see [23]).

**Remark:** When the nonlinearity and the modeling uncertainty terms in (1) are not considered, then the system matrix of the system given in (1), can be written as

$$
\Sigma = \begin{bmatrix} sI - A & D \\ -C & 0 \end{bmatrix}
$$

(3)

When the system matrix in (3) is represented in the Smith form, then the roots of the invariant polynomials are defined as the invariant zeros of the system [23].

Then, as per [6] and Assumption 1, there exists a linear change of coordinates $z = Tx = \begin{bmatrix} z'_1 \\ z'_2 \end{bmatrix}$ with $z'_1 \in \mathbb{R}^{(n-p)}$ and $z'_2 \in \mathbb{R}^p$, so that the system described in (1) becomes a special case of the following model in the new coordinate system [38].

$$
\begin{align*}
\dot{z}_1 &= A_{11}z_1 + A_{12}z_2 + \psi_1(y,u) + \eta_1(z,u,t) \\
\dot{z}_2 &= A_{21}z_1 + A_{22}z_2 + \rho(z,u) + \psi_2(y,u) + \eta_2(z,u,t) + \beta(t - T_x)f(y,u) \\
y &= \bar{C}z_2 + \beta(t - T_y)F\theta
\end{align*}
$$

(4)

where, $TAT^{-1} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$ and the matrix $A_{11} \in \mathbb{R}^{(n-p)\times(n-p)}$ is stable, $TD = \begin{bmatrix} 0 \\ D_2 \end{bmatrix}$, and the matrix $D_2 \in \mathbb{R}^{p\times u}, CT^{-1} = \begin{bmatrix} 0 & I_p \end{bmatrix}$, where $I_p \in \mathbb{R}^{p\times p}$ is an identity matrix. The term $\rho$ is a smooth vector field representing the known system nonlinearity.

**Remark:** The transformation matrix $T$ can be obtained by following the procedure given below (see [5]). Let $(A,D,C)$ be a linear system with $p > \nu$ and rank $(CD) = \nu$. Then a change of coordinates exists so that the system triple with respect to the new coordinates has the following structure:
a) The system matrix can be written as

\[
\tilde{A} = \begin{bmatrix}
\tilde{A}_{11} & \tilde{A}_{12} \\
\tilde{A}_{21} & \tilde{A}_{22}
\end{bmatrix}
\]  

(5)

b) The input distribution matrix can be written as

\[
\tilde{B} = \begin{bmatrix} 0 \\ D_2 \end{bmatrix}
\]  

(6)

c) The output distribution matrix can be written as

\[
\tilde{C} = \begin{bmatrix} 0 & T_o \end{bmatrix}
\]

where \(T_o\) is orthogonal.

First, we apply an initial transformation matrix \(T_C\) such that the structure of the transformed output distribution matrix is of the form \([0 \ \ I_p]\).

The resulting matrices, after the application of the transformation matrix \(T_C\) are given as –

\[
A_C = T_C A T_C^{-1}
\]

\[
D_C = T_C \ast D
\]

\[
C_C = C T_C^{-1}
\]

Then, we apply another transformation matrix \(T_B\) in order to obtain the input distribution matrix in the desired structure. The resulting matrices after the application of the transformation matrix \(T_B\) can be written as –

\[
\tilde{A} = T_B \ast A_C \ast T_B^{-1}
\]

\[
\tilde{B} = T_B \ast D_C
\]

\[
\tilde{C} = C_C \ast T_B^{-1}
\]

The above matrices can also be expressed in the form of the original matrices, as given below –
\[
\tilde{A} = T_B \ast T_C \ast A \ast T_C^{-1} \ast T_B^{-1} = (T_B \ast T_C) \ast A \ast (T_B \ast T_C)^{-1}
\]

\[
\bar{D} = (T_B \ast T_C) \ast D
\]

\[
\bar{C} = C \ast T_C^{-1} \ast T_B^{-1} = C \ast (T_B \ast T_C)^{-1}
\]

In this coordinate system, the input distribution matrix is of the form \( \bar{D} = \begin{bmatrix} 0 \\ D_2 \end{bmatrix} \) and the output distribution matrix is of the form \( \bar{C} = [0 \quad T_o] \). Therefore it can be observed that the transformation matrix that is to be applied to the system matrices in the original system of coordinates is \( T = (T_B \ast T_C) \).

Then, the output distribution matrix \( \bar{C} = [0 \quad T_o] \) can be easily converted to the form \( C_1 = [0 \quad I_p] \) through another coordinate transformation.

In this thesis, it is assumed that at a given time, there is an occurrence of only one fault in the system and that the vector \( F \) specifies the location of the sensor fault. So, the general structure of the vector \( F \) is such that one of its components has the value of 1, while all the other elements are zeros (similar to [38]). The number of components in the \( F \) matrix is equal to the number of sensors whose measurements are to be monitored, such that each component in the vector \( F \) corresponds to a particular sensor in the system. Then, a sensor fault occurring in the system is modeled as in. We specify the location of the fault by assigning a value of one to a particular component in the vector \( F \) which corresponds to that sensor whose measurements are incorrect. This vector is then multiplied by the value of the constant bias in the sensor. The value of the constant sensor bias is represented with a scalar \( \theta \).
For the purpose of process fault isolation, it is assumed that there are \( N \) types of possible process faults. If all these possible types of process faults are represented as a fault set, then the unknown fault function in (1) belongs to the set of faults represented by –

\[
f \triangleq \{ f^1(y, u), \ldots, f^N(y, u) \}
\]

(8)

Here, each fault type \( f^s, s = 1, \ldots, N \), is of the form (see [37]) –

\[
f^s(y, u) \triangleq [(\theta_1^s)^\top g_1^s(y, u), \ldots, (\theta_N^s)^\top g_N^s(y, u)]^\top
\]

(9)

where, \( \theta_i^s, i = 1, \ldots, N \), is an unknown parameter vector and \( g_i^s \) represents the structure of that process fault.

**Assumption 2.** The unstructured modeling uncertainties in the system denoted by \( \eta_1 \) and \( \eta_2 \) in (3) are bounded by known functions [38], such that –

\[
|\eta_1(z, u, t)| \leq \bar{\eta}_1(y, u, t),
\]

\[
|\eta_2(z, u, t)| \leq \bar{\eta}_2(y, u, t)
\]

(10)

where the functions \( \bar{\eta}_1(y, u, t) \) and \( \bar{\eta}_2(y, u, t) \) are known and are bounded.

**Remark:** These bounds are necessary so that it can be determined whether the inconsistencies in the behavior of the system are to be attributed to the fault occurring in the system or to the modeling inconsistencies present in the system model.

**Assumption 3.** The system state vector \( z \) remains bounded prior to and subsequent to the occurrence of the fault.

**Remark:** This assumption means that the system is assumed to be stable and is capable of returning to its stable state of functioning even after the occurrence of a fault [38].
be more precise, this assumption assumes that the feedback control system has the ability to maintain the state variables within their respective bounds prior to and subsequent to the fault occurrence.

**Assumption 4.** The known nonlinear term $\rho(z, u)$ is uniformly Lipschitz, that is –

$$|\rho(z, u) - \rho(\hat{z}, \hat{u})| \leq \gamma|z - \hat{z}|$$  \hspace{1cm} (11)

where $\gamma$ is the known Lipschitz constant [38].

**Remark:** As per Assumption 4, it is necessary, that the known nonlinear term, in the equations representing the system dynamics, satisfy the Lipschitz condition. Examples of the nonlinearities that satisfy the Lipschitz condition include the sinusoidal signals used in robotics research [22].
3. Architecture of Fault Detection and Isolation

One of the common schemes used in fault diagnosis is the observer scheme. The observer scheme used in this thesis follows the general architecture proposed in [36, 39]. As shown in Fig. 5, the Fault Detection and Isolation architecture consists of \( p + N + 1 \) nonlinear adaptive estimators, where \( p \) is the number of sensor measurements, \( N \) is the number of types of process faults. Therefore, out of the \( p + N + 1 \) nonlinear adaptive estimators, \( p + N \) nonlinear adaptive estimators, known as fault isolation estimators (FIEs) [36, 39], are used to determine the type of the fault that has occurred in the system. They are activated once a fault has been detected. The remaining single nonlinear adaptive estimator is the fault detection estimator (FDE), which is used to detect the occurrence of faults in the given system model. Each FIE is designed based on the functional structure of the potential fault.

![Figure 5: The Architecture of the Fault Detection and Isolation Scheme [39]](image-url)
3.1 Fault Detection Components

The fault detection takes place in the Fault Detection Estimator (FDE) and the Fault Detection Decision Scheme (FDDS). The inputs given to the FDE are the input and the output signals of the system. The FDE then generates an estimated output which is then compared to the actual output. The difference between the two is termed as a ‘residual’. The generated residual is compared to a pre-designed threshold in the Fault Detection Decision block. Based on the relation of the threshold and the residual, the following conclusions are drawn

1. If the residual always remains below the threshold, it is concluded that no fault has occurred in the system.
2. If the residual exceeds the threshold at some finite time, it is concluded that a fault has occurred in the system at that instant of time.

The output of the Fault Detection Decision block serves as an alarm, in the case a fault has been detected and also as an activation signal to the fault isolation blocks, to determine the type of the fault that has occurred in the system.

3.2 Fault Isolation Components

Once a fault has been detected by the Fault Detection Decision block, a set of Fault Isolation Estimators (FIEs) are triggered in order to isolate the fault from the system. These four fault isolation estimators can be grouped into two types – process fault isolation estimators and sensor fault isolation estimators, as shown in Fig. 5.
The process fault isolation estimators block, represents a bank of fault isolation estimators, each of which is modeled on the structure of a process fault under consideration. Each of the fault isolation estimators in the process fault isolation estimators block, also generate residuals and corresponding thresholds, but they are used for fault isolation unlike those used for fault detection in the fault detection estimator.

Similarly, the sensor fault isolation estimators block, represents a bank of fault isolation estimators, each of which is modeled on the structure of a sensor fault. Again, each of the fault isolation estimators in the sensor fault isolation estimators block, also generate residuals and corresponding thresholds, for sensor fault isolation.

The residuals generated by these two FIE blocks are given as inputs to the fault isolation decision block. As mentioned previously, we consider that $p$ is the number of sensor measurements, hence $p$ is the number of sensor faults. Also, we consider that $N$ is the number of types of process faults so that a total of $p + N$ faults are considered. Then, within a unified framework [39], if $s$ represents the fault case under consideration, then for $s = 1, \ldots p$, the fault $s$ is a type of sensor fault and for $s = p + 1, \ldots p + N$, the fault $s$ is a type of process fault.
The fault isolation decision logic follows the general idea presented in [36]. The adaptive thresholds are designed in such a way, that for a fault in the \( s \)th component, where \( s = 1, \ldots \ N \), all the residuals generated by the \( s \)th fault isolation estimator always remain below their corresponding thresholds. Simultaneously, in the case of each of the remaining fault isolation estimators (assuming that all the fault cases are sufficiently different) at least one of the residual components would cross their corresponding threshold at a certain instant of time. Therefore, the type of the fault that has occurred can be determined.

Some of the observer schemes used are – dedicated observer scheme [1, 2] and the generalized observer scheme [1, 2]. Assuming that there are a total of \( 'N' \) of faults in the system, the FDI architecture of both the schemes consist of \( 'N' \) observers or state estimators. The design of dedicated observer scheme consists of \( 'N' \) state estimators such that \( s \)-th residual is sensitive to the \( s \)-th fault. The decision of fault detection is made if the \( s \)-th residual exceeds its threshold. On the other hand, the generalized observer FDI architecture compromises of \( 'N' \) state estimators such that the \( s \)-th residual is sensitive to all the faults excepting the \( s \)-th fault. The fault is detected when the \( s \)-th residual exceeds the thresholds generated by all the state estimators excepting the \( s \)-th state estimator. The FDI design that is developed in this thesis is based on the generalized observer scheme.
4. Design of Fault Detection Estimator

4.1 Fault Detection Estimator

In this chapter, the actual design of the Fault Detection Estimator is discussed. Based on the system model described by (2), the fault detection estimator (FDE) is chosen as described by the equations below (see [38]):

\[
\begin{align*}
\dot{z}_1 &= A_{11} z_1 + A_{12} \bar{C}^{-1} y + \psi_1(y, u) \\
\dot{z}_2 &= A_{21} z_1 + A_{22} \dot{z}_2 + \rho(\tilde{z}) + \psi_2(y, u) + L(y - \hat{y}) \\
\hat{y} &= \bar{C} \hat{z}_2 ,
\end{align*}
\]

(12)

In the above equation, \( \dot{z}_1 \) and \( \dot{z}_2 \) and \( \hat{y} \) are the estimated state and output variables respectively. In addition, \( L \) is the design gain matrix and \( \tilde{z} \equiv [(\dot{z}_1)^T (\bar{C}^{-1}y)^T]^T \). The initial conditions are taken as, \( \dot{z}_1(0) = 0 \) and \( \hat{z}_2(0) = 0 \). The residual for fault detection is then generated as \( (y - \hat{y}) \), where \( y \) is the actual output and \( \hat{y} \) is the estimated output.

4.2 Adaptive Threshold Design for Fault Detection

The generated residual is then compared with the threshold to make a diagnostic decision. Next, the details of the threshold design are shown (similar to the threshold design in [38]).

If the state estimation errors are denoted by the terms \( \tilde{z}_1 \equiv z_1 - \hat{z}_1 \) and \( \tilde{z}_2 \equiv z_2 - \hat{z}_2 \) and the output estimation error is denoted by the term, \( \bar{y} \equiv y - \hat{y} \), then prior to the occurrence of the fault, (for \( t < T_0 \)), the system dynamics are given by
\[
\dot{z}_1 = A_{11}z_1 + \eta_1(z,u,t) \\
\dot{z}_2 = A_{21}z_1 + \tilde{A}_{22}\dot{z}_2 + \rho(z) - \rho(\dot{z}) + \eta_2(z,u,t) \\
y = \tilde{C}\dot{z}_2 ,
\]
where \(\tilde{A}_{22} \triangleq A_{22} - L\tilde{C}\). Since \(\tilde{C}\) is nonsingular, \(L\) can always be chosen, such that \(\tilde{A}_{22}\) is stable.

By using (13) and (10) and then applying the triangle inequality, we get

\[
|\dot{z}_1(t)| \leq k_0 \omega_1 e^{-\lambda_0 t} + k_0 \int_0^t e^{-\lambda_0(t-\tau)} \bar{\eta}_1(z,u,\tau)d\tau,
\]
where \(k_0\) and \(\lambda_0\) are positive constants, chosen such that \(|e^{A_{11}t}| \leq k_0 e^{-\lambda_0 t}\) and \(\omega_1\) is a constant bound for \(|\dot{z}_1(0)|\). Since \(A_{11}\) is stable, the constants \(k_0\) and \(\lambda_0\) always exist.

Now, the output estimation error \(\tilde{y}(t) \triangleq y(t) - \hat{y}(t)\) is analyzed. By using (10), we have

\[
\tilde{y}_j(t) = \int_0^t \tilde{C}_j e^{A_{22}(t-\tau)} [A_{21}\tilde{z}_1(\tau) + \rho(z) - \rho(\dot{z})]d\tau + \int_0^t \tilde{C}_j e^{A_{22}(t-\tau)} \eta_2 d\tau ,
\]
(17)

Since

\[
z(\tau) - \dot{z}(\tau) = \begin{bmatrix} \phantom{0} z_1(\tau) - \dot{z}_1(\tau) \\ z_2(\tau) - \tilde{C}^{-1}y(\tau) \end{bmatrix} = \begin{bmatrix} \dot{z}_1(\tau) \\ 0 \end{bmatrix} .
\]

Therefore, we get –

\[
|z(\tau) - \dot{z}(\tau)| = |\dot{z}_1(\tau)|
\]
(18)

Making use of eq. (17), (10) and (11), we have –

\[
|\tilde{y}(t)| \leq k_j \int_0^t e^{-\lambda_j(t-\tau)} [\|A_{21}\| + \gamma]|\dot{z}_1(\tau)|| d\tau + k_j \int_0^t e^{-\lambda_j(t-\tau)} \bar{\eta}_2(z,u,\tau)d\tau + k_j \omega_2 e^{-\lambda_j t} ,
\]
where $k_j$ and $\lambda_j$ are positive constants chosen such that $|\tilde{C}_j e^{\tilde{A}_{22}t}| \leq k_j e^{-\lambda_j t}$ and $\omega_2$ is a constant bound for $|\tilde{z}_2(0)|$.

From (16) and (19), we have

$$v_j(t) \equiv k_j \int_0^t e^{-\lambda_j(t-\tau)}[(\|A_{21}\| + \gamma) \chi(\tau)] d\tau + k_j \int_0^t e^{-\lambda_j(t-\tau)} \tilde{\eta}_2(z,u,\tau) d\tau$$

$$+ k_j \omega_2 e^{-\lambda_j t},$$

where

$$\chi(\tau) \equiv k_0 \omega_0 e^{-\lambda_0 t} + k_0 \int_0^t e^{-\lambda_0(t-\tau)} \tilde{\eta}_1(z,u,\tau) d\tau.$$ \hspace{1cm} (21)

Based on the above discussions, the following fault detection decision scheme can be used.

The decision that a fault has occurred, is made when the absolute value of at least one component of the output estimation error $\tilde{y}_j(t)$, exceeds its corresponding threshold $v_j(t)$ (similar to [38]), given by –

$$v_j(t) \equiv k_j \int_0^t e^{-\lambda_j(t-\tau)}[(\|A_{21}\| + \gamma) \chi(\tau)] d\tau + k_j \int_0^t e^{-\lambda_j(t-\tau)} \tilde{\eta}_2(z,u,\tau) d\tau + k_j \omega_2 e^{-\lambda_j t}.$$ \hspace{1cm} (20)

The instant of time at which at least one of the residual component (or the output estimation error component) exceeds its corresponding threshold is termed as the ‘fault detection time’.

There are basically two kinds of thresholds – fixed thresholds and adaptive thresholds [1, 2, 30]. A fixed threshold is a constant value and an adaptive threshold is a time varying
function. Note that while designing, it is usually better to opt for designing an adaptive threshold rather than a fixed threshold, in order to avoid the problem of false alarms and missed detections. When a normal situation is determined as a faulty situation, it is termed as a false alarm. When a faulty situation is determined as a normal situation, it is termed as missed detections. This is illustrated in Fig. 6 given below.

![Figure 6: Adaptive threshold vs. fixed threshold](image-url)

*Figure 6: Adaptive threshold vs. fixed threshold [30]*
5. Design of Fault Isolation Estimators

In this section, we describe the fault isolation method, including fault isolation estimator design and the isolation decision scheme.

5.1 Fault Isolation Estimators for Sensors Faults

Assuming a fault has been detected at time $t = T_d$, the FIEs are activated. The design of each FIE is similar to the design of the FIEs in [38], where the design of each FIE is based on the functional structure of one of the possible cases of sensor faults. In this case we have $p$ nonlinear adaptive estimators to estimate each sensor fault; for each $k = 1, \ldots, p$, the design equations are

$$\dot{z}_1^k = A_{11} z_1^k + A_{12} \tilde{c}^{-1} (y - P^k \hat{\theta}^k) + \psi_1 (y, u) + \Omega_1^k \dot{\theta}^k$$

$$\dot{z}_2^k = A_{21} z_1^k + A_{22} z_2^k + \rho (\tilde{z}^k, u) + \psi_2 (y, u) + L^k (y - \hat{y}^k) + \Omega_2^k \dot{\theta}^k$$

$$\dot{\Omega}_1^k = A_{11} \Omega_1^k - A_{12} \tilde{c}^{-1} P^k$$

$$\dot{\Omega}_2^k = A_{22} \Omega_2^k - L^k P^k$$

$$\hat{y}^k = \tilde{c} z_2^k + P^k \hat{\theta}^k$$

where $z_1^k$, $z_2^k$ and $\hat{y}^k$ represent the estimated state and output variables respectively. In addition, $L^k$ is the design gain matrix, $\hat{\theta}^k$ is the estimate of the sensor bias provided by the $k$-th isolation estimator. The initial conditions are $z_1^k(T_d) = 0$, $z_2^k(T_d) = 0$, $\Omega_1^k(T_d) = 0$ and $\Omega_2^k(T_d) = 0$.

The adaption law for adjusting $\hat{\theta}^k$ is given (similar to [38]) as –

$$\dot{\theta}^k = P_{\theta^k} \left[ \Gamma_y \left( \tilde{c} \Omega_2^k + P^k \right)^T \tilde{y}^k \right]$$

(27)
where $\Gamma_y$ is the learning rate, $P_{\theta^k}$ is a projection operator, that restricts the value of $\hat{\theta}^k$ to the corresponding known set $\Theta^k$ to ensure the stability of the learning algorithm in the occurrence of modeling uncertainty [7, 13] and $\tilde{y}^k$ is the output estimation error. The parameter estimate provides fault information. But it cannot be guaranteed that in the case of matched fault isolation estimator, the parameter estimate will converge to the actual value of the fault magnitude, except if we assume persistent excitation [7, 13].

### 5.2 Fault Isolation Estimators for Process Faults

Assuming a fault has been detected at time $t = T_d$, the FIEs are activated. Each FIE is designed based on the functional structure of one potential fault [37]. In this case we have $N$ nonlinear adaptive estimators to estimate each process fault; for each $q = 1, ..., N$, we have

\begin{align*}
\hat{z}_1^q &= A_{11} \hat{z}_1^q + A_{12} \tilde{C}^{-1} y + \psi_1(y, u) \\
\hat{z}_2^q &= A_{21} \hat{z}_1^q + A_{22} \hat{z}_2^q + \rho(\hat{z}^q, u) + \psi_2(y, u) + L^q(y - \hat{y}^q) + \Omega^q_x \hat{\theta}^q \\
&\quad + \hat{f}^q(y, u, \hat{\theta}^q) \\
\hat{\Omega}_x^q &= \tilde{A}_{22} \hat{\Omega}_x^q - G^q(y, u) \\
\hat{\gamma}^q &= \tilde{C} \hat{z}_2^q
\end{align*}

\(28\)
\(29\)
\(30\)
\(31\)

In the above equations $\hat{z}_1^q$, $\hat{z}_2^q$ and $\hat{\gamma}^q$ represent the estimated state and output variables respectively. In addition, $L^q$ denotes the design gain matrix, $\hat{f}^q(y, u, \hat{\theta}^q)$ is the estimate of the fault function provided by the $q$-th isolation estimator. As shown in (8), the fault approximation model $\hat{f}^q$ is made linear in the adjustable weights $\hat{\theta}^q$. Therefore gradient matrix $G^q$ does not depend on $\hat{\theta}^q$.

The adaption law for adjusting $\hat{\theta}^q$ is given below (similar to [37])
\[
\hat{\theta}_q = P_{\theta^q} \{ \Gamma_x \Omega^q x^T \bar{C}^T \bar{y}_q \}
\]  
(32)

where \( \Gamma_x \) is the learning rate, \( P_{\theta^q} \) is a projection operator, that restricts the value of \( \hat{\theta}_q \) to the corresponding known set \( \Theta^q \) to ensure the stability of the learning algorithm in the occurrence of modeling uncertainty and \( \bar{y}_q \) is the output estimation error.

### 5.3 Fault Isolation Decision Scheme for sensor and process faults

If this thesis, we consider a total of \( p \) sensor faults and \( N \) process faults. A fault case \( s \) is considered a sensor fault when \( s = 1, \ldots, p \). Similarly, when \( s = p + 1, \ldots, p + N \), the fault case is considered a process fault. The decision of fault isolation is based on the general idea in [36]. If a fault occurs in the \( s \)-th component at time \( T_0 \), then for the set of residual components generated by the \( s \)-th isolation estimator, a corresponding set of adaptive thresholds \( \mu^s_j(t) \) can be designed. These thresholds are designed so that the condition \( |\bar{y}^s_j(t)| \leq \mu^s_j(t) \) (for all \( t > T_d \)) is satisfied by the \( j \)-th component of its output estimation error. Here, \( T_d \) is the time at which the fault is detected.

The design of adaptive threshold for fault isolation has been investigated in [37, 38]. Below, we directly give the result.

If a sensor fault occurs, the \( j \)-th component of the output estimation error of the \( s \)-th isolation estimator satisfies the inequality [38] –

\[
|\bar{y}^s_j(t)| \leq k_j \int_{T_d}^t e^{-\lambda_j(t-\tau)} \left[ (\|A_{21}\| + \gamma)(\chi^s(\tau) + \|\Omega^s \hat{\theta}_s\|) + \hat{\eta}_2 \right. \\
+ \gamma |\bar{C}^{-1} F_s| \| \hat{\theta}_s \| d\tau + k_j \omega_2 e^{-\lambda_j(t-T_d)} + |\bar{C} j \Omega^s_2 + F_s^s| \| \hat{\theta}_s \| \tag{33}
\]
where

\[
\chi^s(\tau) \triangleq k_0 \int_{T_d}^{t} e^{-\lambda_0(t-\tau)} \left[ \tilde{\eta}_1(x, u, \tau) \right] d\tau + k_0 \omega_1 e^{-\lambda_0(t-T_d)}
\]  

(34)

If a process fault occurs, at a time \( T_d \), then for \( t > T_d \), the \( j \)-th component of the output estimation error of the \( s \)-th isolation estimator satisfies the following inequality [37]

\[
\left| \tilde{y}_j^s(t) \right| \leq k_j \int_{T_d}^{t} e^{-\lambda_j(t-\tau)} \left[ (\|A_21\| + \gamma) \chi^s(\tau) + \tilde{\eta}_2(y, u, \tau) \right] d\tau \\
+ \left| (C_j \Omega_2)^T \tilde{\tilde{z}}^s \right| + k_j \omega_2 e^{-\lambda_j(t-T_d)}
\]

(35)

where –

\[
\chi^s(\tau) \triangleq k_0 \int_{T_d}^{t} e^{-\lambda_0(t-\tau)} \left[ \tilde{\eta}_1(x, u, \tau) \right] d\tau + k_0 \omega_1 e^{-\lambda_0(t-T_d)}
\]

(36)

The terms on the right hand side of (33) and (35) are used to design the adaptive thresholds for the purpose of fault isolation. The effect of the bounds \( \omega_1 \) and \( \omega_2 \) decreases exponentially, therefore the performance of the fault isolation algorithm is not affected significantly.
6. Simulation Example and FDI Implementation

6.1 System Dynamics and Fault Model

In this chapter, we use a single-link robotic arm example considered in [38] to illustrate the effectiveness of the unified FDI method. The motion equations are given by –

\[
J_l \ddot{q}_1 + F_l \dot{q}_1 + k(q_1 - q_2) + mgh \sin q_1 = 0
\]

\[
J_m \ddot{q}_2 + F_m \dot{q}_2 - k(q_1 - q_2) = k_r u
\]

where \( q_1 \) and \( q_2 \) are the angular positions of the link and the motor respectively. The other parameters in the equation are link inertia \( J_l \); the motor rotor inertia \( J_m \); the elastic constant \( k \), the link mass \( m \), the gravity constant \( g \), the center of mass \( h \) and the viscous friction coefficients \( F_l \) and \( F_m \). In this simulation study, the above parameters take the following values; \( k = 2 \) Nm/rad, \( F_m = 1 \) Nm/V, \( F_l = 0.5 \) Nm/V, \( J_m = 1 \) kg m², \( J_l = 4.5 \) kg m², \( m = 4 \) kg, \( g = 9.8 \) m/s², \( h = 0.5 \) m, \( k_r = 1 \) Nm/V. The initial conditions of the plant are, \( q_1(0) = q_2(0) = 0 \). The input to the system is \( u = 2 \sin(\frac{t}{2}) \). The states chosen are, \( x_1 = q_1 \), \( x_2 = \dot{q}_1 \), \( x_3 = q_2 \), \( x_4 = \dot{q}_2 \). It is also assumed that the motor position, the link velocity and the link position are measured.

Then, the above model can be rewritten in state space form as follows

\[
\begin{bmatrix}
\dot{x}_1 \\
\dot{x}_2 \\
\dot{x}_3 \\
\dot{x}_4
\end{bmatrix} = \begin{bmatrix}
0 & 1 & 0 & 0 \\
-k/J_l & -F_l/J_l & k/J_l & 0 \\
k/J_m & 0 & 0 & 1 \\
k/J_m & 0 & -k/J_m & -F_m/J_m
\end{bmatrix} \begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
x_4
\end{bmatrix} + \begin{bmatrix}
0 \\
-mgh \sin x_1/l_1 \\
0 \\
k_r u/j_m
\end{bmatrix}
\]

\[
y = \begin{bmatrix}
1 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
x_4
\end{bmatrix}
\]

(38)
By using a linear change of coordinates with \( T = \begin{bmatrix} -50 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \)

The state space model in the new coordinate system is
\[
\begin{bmatrix}
\dot{z}_1 \\
\dot{z}_2
\end{bmatrix} = \begin{bmatrix}
-1 & 50 & 0 & 0 \\
-0.027 & 0.89 & -0.44 & 0 \\
0 & 0 & 1 & 0 \\
-0.04 & 0 & -2 & -1
\end{bmatrix} \begin{bmatrix}
z_1 \\
z_2
\end{bmatrix} + \begin{bmatrix}
0 \\
-4.36\sin\left(\frac{z_1}{50}\right) \\
0 \\
u
\end{bmatrix} + \eta(x, u, t) + \beta(t - T_x)f(y, u)
\]
\[
y = \begin{bmatrix}
-1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix} z_2 + \beta(t - T_y)F\theta
\]

The terms \( \eta(x, u, t), \beta(t - T_x)f(y, u) \) and \( \beta(t - T_y)F\theta \) which represent the effects of the modeling uncertainty, the process faults and the sensor faults, are added to the state space equations of the system in the new coordinate system.

When we compare the state space model given above, to the general equation representation of nonlinear systems given by (4), it can be observed that, \( \psi_1 = 0 \) and \( \psi_2 = [0 \ 0 \ u]^T \) and \( \rho = [-4.36\sin x_4 \ 0 \ 0]^T \).

For the given plant model, the nonlinearity in the system is a nonlinear function of the state variables, satisfying a Lipschitz condition. In this case, the nonlinear term \( -4.36\sin\left(\frac{z_1}{50}\right) \) has a Lipschitz constant \( \gamma = 0.087 \). According to the Assumption 2 in Chapter 2, the modeling uncertainties in the system are unstructured and unknown.
functions of $z, u$ and $t$, but are bound by given functionals. The modeling uncertainty in the given plant model is assumed to be an inaccuracy of up to 5\% in the amplifier gain $k_\tau$ \cite{38}. Therefore the bounding functions of the modeling uncertainty are $\eta_1 = 0$ and $\eta_2 = \frac{0.05k_1}{J_m}|u(t)|$.

For the given plant model, four different types of faults are considered. The aim is to detect, isolate and estimate these faults as accurately as possible. The types of faults considered in this simulation are – sensor faults and process faults. Upon the occurrence of any one of these faults in the system, not only should the fault be detected, but the type of fault, the nature of the fault and the magnitude of the fault should be determined.

The following fault scenarios are considered –

- **Fault case scenario 1: Sensor fault due to bias in $y_2$**

In this case, the fault is considered to have occurred due to a sensor bias in the sensor measuring the $y_2$ component of the output (denoted by $y$). In the simulation example we assume that there are three sensors, each sensor measuring each of the three components of the output signal. The bias is represented by the equation:

$$F = F_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

and $\theta_1$, where $\theta_1$ is the magnitude of the sensor bias, that appears in the sensor measuring $y_2$, in radians. It is assumed that the range of the magnitude of the sensor bias is $\theta_1 \in [0, 0.2]$. 

32
• Fault case scenario 2: Sensor fault due to bias in $y_3$

The fault is considered to have occurred due to a sensor bias in the sensor measuring the $y_3$ component of the output (denoted by $y$). The bias is represented by the equation: $F = F_3 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ and $\theta_2$, where $\theta_2$ is the magnitude of the sensor bias, that appears in the sensor measuring $y_3$, in radians. As in the case of fault scenario 1, the range of the magnitude of the sensor bias is assumed to be $\theta_2 \in [0, 0.2]$.

• Fault case scenario 3: Actuator fault (see [37])

By definition, an actuator fault occurs due to the improper functioning of the components actuating the system. In this system, the effect of an actuator fault can be observed in the magnitude of the nominal control input signal being given to the system. Therefore, when the effect of the actuator fault is taken into the consideration, the nominal control input can be modeled as $u = \bar{u} + \vartheta_1 \bar{\bar{u}}$, where $\bar{u}$ is the nominal control input in the non-fault case. Here, the range of the parameter $\vartheta_1$ is $\vartheta_1 \in [-1, 0]$. The range of the parameter $\vartheta_1$ represents the extent of the improper functioning of the actuator. The no fault condition is indicated by the value $\vartheta_1 = 0$, while the complete failure of the actuator is indicated by the value $\vartheta_1 = -1$. If the structure of the actuator fault is represented by $g_1(u)$, then the fault function can be represented by the equation –

$$f(y, u) \triangleq [0 \ 0 \ 0 \ \vartheta_1 g_1(u)]^T$$
with the fault function structure being, \( g_1(u) = \frac{k_u u}{J_m} \)

- Fault case scenario 4: Fault leading to increase in friction in the motor (see [37])

The fault is modeled based on its effect on the friction constant, the viscous friction constant \( F_m \) increases from 1 Nm/V to 3.3 Nm/V. The significance of the extra friction is given by the parameter \( \vartheta_2 \). The range of the parameter is \( \vartheta_2 \in [-10] \). The structure of the fault can be represented by \( g_2(u) \). The fault function can be represented by the equation

\[
f(y, u) = [0 \ 0 \ \vartheta_2 g_2(u)]^T
\]

with the fault function structure being, \( g_2(u) = \frac{2.3 y_2}{J_m} \) and \( \vartheta_2 = -1 \)

### 6.2 FDI Design

As explained in the chapters 3, 4 and 5, since there are four types of fault cases that are being considered, one fault detection estimator (FDE) and four different fault isolation estimators need to be designed. The equations of the FDE are

\[
\dot{\hat{z}}_1 = -\hat{z}_1 + [-50 \ 0 \ 0]y
\]

\[
\dot{\hat{z}}_2 = \begin{bmatrix} -0.027 \\ 0 \\ -0.04 \end{bmatrix} \hat{z}_1 + \begin{bmatrix} 0.89 & -0.44 & 0 \\ 0 & 0 & 1 \\ 0 & -2 & -1 \end{bmatrix} \hat{z}_2 + \begin{bmatrix} 4.36 \sin(\hat{z}_1 / 50) \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + L(y - \hat{y})
\]

\[
\hat{y} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \hat{z}_2
\]

The design equations for the FIEs for the sensor faults are
\[
\dot{z}_i = [-1] \begin{bmatrix} z_i \\ \end{bmatrix} + [50 \ 0 \ 0] \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} y - F^s \hat{\theta}^s \end{bmatrix} + \Omega_i^s \dot{\hat{\theta}}^s
\] (43)

\[
\dot{z}_2 = \begin{bmatrix} -0.027 \\ 0 \\ -0.04 \end{bmatrix} \begin{bmatrix} z_i \\ \end{bmatrix} + \begin{bmatrix} -1.9111 & -0.44 & 0 \\ 0 & -3 & -0.1 \end{bmatrix} \begin{bmatrix} z_2^s \\ \end{bmatrix} + \begin{bmatrix} 4.36 \sin \left( \frac{z_2^s}{50} \right) \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \end{bmatrix} + \Omega_2^s \dot{\hat{\theta}}^s + L^s (y - \hat{y}^s)
\] (44)

\[
\dot{\Omega}_i^s = [-1] \begin{bmatrix} \Omega_i^s \end{bmatrix} - [50 \ 0 \ 0] \bar{C}^{-1} F^s
\] (45)

\[
\dot{\Omega}_2^s = \begin{bmatrix} -1.5 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} \Omega_2^s \\ \end{bmatrix} - \begin{bmatrix} \Omega_2^s \end{bmatrix} - L^s F^s
\] (46)

\[
\hat{y}^s = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} \Omega_2^s \\ \end{bmatrix} + \begin{bmatrix} \Omega_2^s \end{bmatrix} + F^s \dot{\hat{\theta}}^s
\] (47)

The fault parameter estimation is given by the adaptive algorithm for [38]

\[
\dot{\hat{\theta}}^s = \gamma \begin{bmatrix} \Omega_2^s \end{bmatrix} + \begin{bmatrix} F^s \end{bmatrix}^T \hat{y}^s
\] (48)

To prevent parameter drift as a result of the fault function approximation error, the projection operator is used [13], such that

\[
\dot{\hat{\theta}}^s = \begin{cases} 0 & \text{if } \hat{\theta}^s > M_\theta \text{ and } \dot{\hat{\theta}}^s > 0 \text{ or } \hat{\theta}^s < -M_\theta \text{ and } \dot{\hat{\theta}}^s < 0 \\ \{ \Gamma_y (\bar{C} \Omega_2^s + F^s)^T \hat{y}^s \} & \text{if } \hat{\theta}^s > M_\theta \end{cases}
\]

where \( M_\theta \) is the boundary of the compact set for \( \hat{\theta}^s \).

The design equations for the FIEs for the process faults are –

\[
\dot{z}_i = [-1] \begin{bmatrix} z_i \\ \end{bmatrix} + [50 \ 0 \ 0] \bar{C}^{-1} y
\] (49)
\[ + L^\xi (y - \hat{y}^\xi) + \hat{f}^\xi (y, u, \hat{\theta}^\xi) \]  

\[ \hat{\dot{\xi}}_x = \begin{bmatrix} -1.5 & 0 & 0 \\ 0 & -1.7 & 0 \\ 0 & 0 & -1.9 \end{bmatrix} \Omega^\xi_x + G^\xi (y, u) \]  

\[ \hat{y} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \hat{z}_2^\xi \]  

The fault parameter estimation is given by the adaptive algorithm for [37]

\[ \hat{\hat{\theta}}^\xi = P_\theta \{ \Gamma^\xi_x \Omega^\xi_x^T \bar{C}^T \hat{y}^q \} \]  

To prevent parameter drift as a result of the fault function approximation error, the projection operator is used [13], such that

\[ \hat{\dot{\hat{\theta}}}^\xi = 0 \quad \text{if} \quad \hat{\dot{\hat{\theta}}}^\xi > M_\theta \quad \text{and} \quad \hat{\dot{\hat{\theta}}}^\xi > 0 \quad \text{or} \quad \hat{\dot{\hat{\theta}}}^\xi < -M_\theta \quad \text{and} \quad \hat{\dot{\hat{\theta}}}^\xi < 0 \]

\[ \hat{\dot{\hat{\theta}}}^\xi = \{ \Gamma^\xi_x \Omega^\xi_x^T \bar{C} \hat{y}^q \} \quad \text{if} \quad \hat{\dot{\hat{\theta}}}^\xi > M_\theta \]

The observer gain matrix \( L \) is set to be \( L = \begin{bmatrix} -2.3889 & -0.4444 & 0 \\ 0 & 1.7 & 1 \\ 0 & -2 & -0.9 \end{bmatrix} \) so that the poles of the matrix \( \bar{A}_{22} \) are situated at \(-1.5, -1.7\) and \(-1.9\). The other constants are chosen so that their values are \( k_0 = k_1 = k_2 = k_3 = 1, \ \lambda_0 = 1, \ \lambda_1 = 1.2, \ \lambda_2 = 1.4, \ \lambda_3 = 1.9 \).  

6.3 Simulink Model Implementation

For the diagnostic system described above, a simulink model has to be developed. The Simulink model for the given system is shown in Fig. 7 below.
Figure 7: The Simulink model of the FDI algorithm for the single-link robotic arm

The block diagram shown above can be divided into three blocks –

- The single-link robotic arm system
- The fault detection block
- The Fault isolation and estimation block

It is assumed that all the fault cases in this thesis are abrupt faults. Therefore the fault time profile can expressed in the form of a step function, where the fault occurs at the instant of time $T_0$. When a fault occurs in the system, it is represented by adding a step
function that has been multiplied with the fault function, whose structure has been modeled on the fault that has been introduced into the system.

The fault detection block consists of the fault detection estimator for the system. As per the derivations in Chapter 4, if the output can be divided into three components, then a threshold is to be designed for each component. If any one of the output residual components crosses its corresponding threshold, then the conclusion is that a fault has occurred in the system. The output of this block is an activation signal to the fault isolation and estimation subsystem. When no fault is detected in the system, the activation signal is zero. On the other hand, when a fault is detected, then an activation signal is sent to the FIE block.

The third block is the fault isolation block. An in depth view of this block is shown in Fig. 8. In the case of a no-fault condition, this subsystem is not activated at all. However, if a fault has been detected by the fault detection estimation subsystem, the FIE block is triggered. Since four different faults are being considered, this block consists of four further subsystems. Each subsystem represents the FIE designed for a particular fault structure.

Since the output residual that is generated, is represented as three different components, each FIE (designed for a particular fault structure) is required to generate three thresholds, corresponding to each of the output components. It is assumed that the four fault cases being considered are sufficiently different from each other in terms of the fault
structure. Then, for a particular fault case, only one of the four FIEs will be able to generate fault isolation residuals and their corresponding fault isolation thresholds, such that, all the fault isolation residuals generated by the FIE will remain below their corresponding fault isolation thresholds. Then, it can be concluded that the fault structure of that particular FIE matches the fault structure of the actual fault that has occurred and hence the fault can be identified.

Figure 8: The block diagram of the Fault Isolation block
7. **Simulation Results**

Once a fault has been introduced into the system, the FDE block has to detect the fault. It is assumed that only one type of fault occurs in the system at any given time. The FDE block generates an estimated output, which is compared with the actual output. The difference between the actual output and the estimated output is termed as a residual.

As the output signal, the residual signal also has three components. Based on the design of the FDE, the FDE block also generates a threshold for each of these residual components. The fault detection happens when any one of the output residual components generated by an FDE exceeds its corresponding threshold generated by the FDE.

Once a fault has been detected in the fault detection block, the fault type is to be identified and isolated in the fault isolation block. Each FIE estimates the output which is compared to the actual output to generate a residual that has three components. Similarly, each FIE also generates three thresholds, each for one residual component. A fault is said to be isolated, if the following conditions are satisfied

- If all of the output residual components generated by an FIE, whose fault structure matches the fault structure of the actual fault, do not exceed the thresholds generated.

- When the at least one of output residual components exceed their corresponding thresholds generated by each of the mismatched FIEs.

Below, we give the simulation results for the simulink model of the FDI algorithm for the single-link robotic arm that has been taken as the example.


7.1 Sensor fault in $y_2$

The Fig. 9 shows the simulation results when there is a bias of $\theta_1 = 0.18$ radians in the sensor measuring $y_2$ at $T_0 = 5$sec. The top left plot, shows the simulation result of the FDE, where the output residual component $y_3$ (the solid line) and its corresponding threshold (the dashed line) generated by the FDE are shown. The fault is detected at approximately $T_d = 5.084$sec. This detection can be seen when the output residual component $y_3$ (solid line) crosses the threshold (dashed line).

![Figure 9: Sensor fault in $y_2$: The plots of the FDE and the FIEs for the remaining fault cases (sensor fault #2, process fault #1 and process fault #2) with the chosen residual component (solid line) and its corresponding threshold (dotted line)](image)

Figure 9: Sensor fault in $y_2$: The plots of the FDE and the FIEs for the remaining fault cases (sensor fault #2, process fault #1 and process fault #2) with the chosen residual component (solid line) and its corresponding threshold (dotted line)
At the moment of detection, the four fault isolation estimators are activated. The other three plots show the output residual component $y_3$ and the corresponding threshold generated by the FIE #2, FIE #3 and FIE #4. It can be observed from Fig. 10 above, that in FIE #2, FIE #3 and FIE #4, the residual component $y_3$ (solid line) exceeds its threshold (dashed line). Therefore, it can be concluded, that the fault structures, on which each of these FIEs are modeled, do not match the fault structure of the fault that has occurred in the system. So, the fault that has occurred is not the fault case 2, fault case 3 or fault case 4.

Figure 10: Sensor fault in $y_2$: The plots of the FIE for the sensor fault #1, with the three residual components (solid line) and their corresponding thresholds (dotted line)
The Fig. 10 shows the fault isolation estimator for sensor fault #1. It can be observed, that all of the residual components generated by the FIE #1 do not exceed their corresponding thresholds. Therefore, from the simulation results shown in Fig. 9 and Fig.10, it can be concluded that the fault that has occurred in the system is fault case 1, which is a sensor fault in the output component $y_2$.

Figure 11: The case of sensor fault in $y_2$ (sensor fault #1): the fault magnitude estimation for fault isolation estimator #1.

The Fig. 11 shows the fault magnitude estimation of the fault isolation estimator (FIE) #1. It can be seen that the value of the estimated fault magnitude is very close to the actual sensor bias magnitude of 0.18 radians. This information can be used to conduct fault-tolerant control, hence improving the control performance in the presence of the fault.
7.2 Sensor fault in $y_3$

The Fig. 12 shows the simulation results when there is a bias of $\theta_2 = 0.15$ radians in the sensor measuring $y_2$ at $T_0 = 5$ sec. The top left plot shows the simulation result of the FDE, where the output residual component $y_2$ (solid line) and its corresponding threshold (dashed line) generated by the FDE are shown. The fault is detected at approximately $T_d = 5.36$ sec. This detection can be seen when the output residual component $y_2$ (solid line) exceeds its threshold (dashed line) generated by the FDE.

Figure 12: Sensor fault in $y_3$: The plots of the FDE and the FIEs for the remaining fault cases (sensor fault #1, process fault #1 and process fault #2) with the chosen residual (solid line) and its corresponding threshold (dotted line)
The other three plots show the output residual component $y_2$ and the corresponding threshold generated by the FIE #1, FIE #3 and FIE #4. It can be observed from Fig. 13 above, that in FIE #2, FIE #3 and FIE #4, the residual component $y_2$ (solid line) exceeds its threshold (dashed line). Therefore, it can be concluded, that the fault structures, on which each of these FIEs are modeled, do not match the fault structure of the fault that has occurred in the system. So, the fault that has occurred is not the fault case 1, fault case 3 or fault case 4.

![FIE for sensor fault #2](image)

**Figure 13:** Sensor fault in $y_3$: The plots of the FIE for the sensor fault #2, with the three residual components (solid lines) and their corresponding thresholds (dotted lines)

Fig. 13 shows the fault isolation estimator for sensor fault #2. It can be observed, that all of the residual components (solid lines) generated by the FIE #2 do not exceed their
corresponding thresholds (dashed lines). Therefore, from the simulation results shown in Fig. 12 and Fig. 13, it can be concluded that the fault that has occurred in the system is fault case 2, which is a sensor fault in the output component $y_3$.

![Graph showing fault magnitude estimation for fault isolation estimator #2.](image)

**Figure 14:** The case of sensor fault in $y_3$ (sensor fault #2): the fault magnitude estimation for fault isolation estimator #2.

Fig. 14 shows the fault magnitude estimation of the fault isolation estimator (FIE) #2. It can be seen that the value of the estimated fault magnitude is very close to the actual sensor bias magnitude of 0.15 radians. This information can be used to conduct fault-tolerant control, to enhance the performance of the controller in the presence of the fault.
7.3 Actuator fault

Fig. 15 shows the simulation results when there is an actuator fault with a fault magnitude of \( \theta_1 = -0.6 \) in \( y_2 \) at \( T_0 = 5 \) sec. The top left plot shows the simulation result of the FDE, where the output residual component \( y_3 \) (solid line) and its corresponding threshold (dashed line) generated by the FDE are shown. The fault is detected at approximately \( T_d = 5.015 \) sec.

![Figure 15](image)

**Figure 15**: Actuator fault in \( y_3 \): The plots of the FDE and the FIEs for the remaining fault cases (sensor fault #1, sensor fault #2 and process fault #2) with the chosen residual (solid line) and its corresponding threshold (dotted line)

The other three plots also show the output residual component \( y_3 \) (solid line) and the corresponding threshold (dashed line) generated by the FIE #1, FIE #2 and FIE #4.
Similar to the first two cases, the residual component $y_3$ exceeds its threshold in FIE #1, FIE #2 and FIE #4. So, the fault that has occurred is not the fault case 1, 2 or 4.

Figure 16: Actuator fault in $y_3$: The plots of the FIE for the process fault #1, with the three residual components (solid lines) and their corresponding thresholds (dotted lines)

The Fig. 16 shows the fault isolation estimator for process fault #1. It can be observed, that all of the residual components (solid lines) generated by the FIE #3 do not exceed their corresponding thresholds (dashed lines). Therefore, from Fig. 15 and Fig. 16, it can be concluded that the fault structure on which the FIE is modeled, matches the fault structure of the fault, that has occurred in the system and the fault that has occurred is the process fault case 1, which is the actuator fault.
Figure 17: The case actuator fault in $y_3$ (process fault #1): the fault magnitude estimation for fault isolation estimator #1.

The Fig. 17 shows the fault magnitude estimation of the fault isolation estimator (FIE) #3. It can be seen that the value of the estimated fault magnitude is very close to the actual fault magnitude of -0.6. This information can be used to conduct fault-tolerant control, hence improving the control performance in the presence of the fault.

7.4 Fault leading to extra abnormal friction in the motor

The Fig. 18 shows the simulation results the fault case occurs and the viscous friction constant $F_m$ increases from 1 Nm/V to 3.3 Nm/V. The significance of the extra friction is given by the parameter $\varphi_2$. The value of the parameter is $\varphi_2 = -0.7$ at $T_0 = 5$ sec. The top
left plot shows the simulation result of the FDE, where the output residual component \( y_3 \) (solid line) and its corresponding threshold (dashed line) generated by the FDE are shown. The fault is detected at approximately \( T_d = 5.34 \text{sec} \).

![FDE](image1)

![FIE for sensor fault #1](image2)

![FIE for sensor fault #2](image3)

![FIE for process fault #1](image4)

**Figure 18: Process fault #2 in \( y_3 \):** The plots of the FDE and the FIEs for the remaining fault cases (sensor fault #1, sensor fault #2 and process fault #1) with the chosen residual (solid line) and its corresponding threshold (dotted line)

Once again, it can be observed, that in three of these fault isolation estimators, at least one of the residual components (solid line) generated by them exceeds its corresponding threshold (dashed line). Hence, it can be concluded, that the fault structure, on which each of these FIEs are modeled, do not match the fault structure of the fault that has occurred in the system.
Figure 19: Process fault #2 in $y_3$: The plots of the FIE for the process fault #2, with the three residual components (solid lines) and their corresponding thresholds (dotted lines).

The Fig. 19 shows the fault isolation estimator for process fault #2. It can be observed, that the residual components generated by the FIE do not exceed their corresponding thresholds.

Therefore, it can be concluded that the fault structure on which the FIE is modeled, matches the fault structure of the fault, that has occurred in the system. The fault structure of FIE #4 corresponds to a fault leading to extra abnormal friction in the motor.
Figure 20: The case of fault leading to extra abnormal friction in the motor (process fault #2): the fault magnitude estimation for fault isolation estimator #4.

The Fig. 20 shows the fault magnitude estimation of the fault isolation estimator (FIE) #4. It can be seen that the value of the estimated fault magnitude is very close to the actual fault magnitude of -0.7. This information can be used to conduct fault-tolerant control, hence improving the control performance in the presence of the fault.
8. Conclusions and Future work

In this thesis, we developed a unified method for detecting and isolating process faults and sensor faults in a class of nonlinear uncertain systems under consideration. More specifically, the following contributions are made

1. A unified framework for the diagnosis of sensor faults and process faults for the class of nonlinear systems under consideration is developed.

2. The FDI method so developed is applied to the example of the single-link robotic arm and the fault detection and isolation design is conducted.

3. The Simulink model for the FDI algorithm of the single-link robotic arm is implemented and satisfactory results are obtained for the various cases of fault conditions.

Some of the topics which can be researched in the future include the following –

The algorithm presented in this thesis only deals with the fault detection and isolation of the various fault cases, but not the fault tolerant control problem. However, fault tolerant control is equally important as the process of fault diagnosis itself. The fault information obtained from the FDI algorithm can be used to ensure that the corresponding fault tolerant control algorithm can be improved. Therefore, the integration of the FDI algorithm along with fault tolerant control is an interesting topic of research.
In this thesis, the only uncertainty considered in the nonlinear system, is the modeling uncertainty. It is assumed that there is no sensor noise affecting the output of the system. However, in practical cases, there is a possibility of the output of the sensors being affected by a small magnitude of sensor noise. Therefore, sensor noise is another factor that is to be considered along with modeling uncertainty, while designing the FDI algorithm.

One of the main areas where fault diagnosis is very important is in the area of large scale distributed networks, such as, in large scale power systems, wireless communication systems, intelligent vehicle systems, etc. Most of the FDI algorithms that have been developed so far have been developed for centralized systems. However, these centralized methods cannot be applied in the case of distributed networks, due to the constraints of computational limitations, communication bandwidth and the inherent complexity of the system. Therefore, the development of an FDI algorithm, for such large scale systems is an interesting topic for future research.
9. References


of Hull, UK, 163-168


