ADVANCES IN FAULT DIAGNOSIS AND FAULT TOLERANT CONTROL
MOTIVATED BY LARGE FLEXIBLE SPACE STRUCTURE

by

Yao-Hong Kok

A thesis submitted in conformity with the requirements
for the degree of Master’s of Applied Science
Graduate Department of Electrical and Computer Engineering
University of Toronto

© Copyright 2013 by Yao-Hong Kok
Abstract

Advances in Fault Diagnosis and Fault Tolerant Control Motivated by Large Flexible Space Structure

Yao-Hong Kok
Master’s of Applied Science
Graduate Department of Electrical and Computer Engineering
University of Toronto
2013

In this thesis, two problems are studied. The first problem is to find a technique to generate a particular type of failure information in real time for large flexible space structures (LFSSs). This problem is solved by using structured residuals. The failure information is then incorporated into an existing fault tolerant control scheme. The second problem is a “spin-off” from the first. Although the $\mathcal{H}_\infty$ sliding mode observer (SMO) cannot be applied to the colocated LFSS, its ability to do robust state and fault estimation of the SMO makes it suitable to be used in an integrated fault tolerant control (IFTC) scheme. We propose to combine the $\mathcal{H}_\infty$ SMO with a linear fault accommodation controller. Our IFTC scheme is closed loop stable, suppresses the effects of faults and enjoys enhanced robustness to disturbances. The effectiveness of the IFTC is illustrated through the control of a permanent magnet synchronous motor under actuator fault.
To my respected supervisor, my parents, my five brothers and sisters, my friends and my sweetheart, Yun Luo.
Acknowledgements

I would like to express my deep gratitude to my research supervisor, Professor Raymond Kwong for his guidance, enthusiastic encouragement and useful critiques of this research work. His great patience and understanding has led me to good solutions for the problems we wanted to solve initially. Moreover, I cannot thank him enough for continuously spending time to help improve my writing, which is crucial in crafting this thesis.

I would also like to thank Dr. Zheng Wang for pointing out the possibility of using a linear controller to be combined with the sliding mode observer. My grateful thanks are also extended to Farzad Salehisadaghiani, Jimmy Tsai and Yupeng Yao for helping me to correct my writing. Furthermore, I am grateful to have made amazing friends at the System Control Group at University of Toronto, who provided much mental support and advice in doing research.

Also, I am grateful for being part of the Kok family. Thank you mom for calling me every week to check on me. Thank you dad for calling on different occasions to check on my thesis progress. As for my brothers and sisters, thank you for bringing so much joy and companionship to my life. As a son and a brother, I rarely say this: I love all of you.

Last but not least, I want to thank Yun, my sweetheart, who continuously provided me with mental support and encouragement to put my best effort into the writing of my thesis. On a side note, thank you for walking into my life unexpectedly. I love you, Yun.
Contents

1 Introduction .................................................. 1
   1.1 Problem Descriptions .................................. 2
   1.2 Contributions .......................................... 3
   1.3 Thesis Structure ........................................ 5

2 Background and Literature Review ......................... 6
   2.1 Mathematical Background and Notations ............... 6
   2.2 Large Flexible Space Structures (LFSS) ............... 7
      2.2.1 Colocated LFSS Model ............................. 9
      2.2.2 Failure Matrices ................................... 11
   2.3 Overview of Fault Tolerant Control Systems .......... 12
      2.3.1 Residual Generation ............................... 16
      2.3.2 Structured Residual ................................ 17
      2.3.3 Sliding Mode Control Theory ....................... 19
      2.3.4 Sliding Mode Observers ............................. 21
      2.3.5 Integrated Fault Tolerant Control .................. 22
   2.4 Fault Tolerant Control of LFSS ......................... 24
      2.4.1 Huang’s Fault Tolerant Controllers ................ 25
      2.4.2 Huang’s Fault Diagnosis Algorithm ............... 28

3 Fault Accommodation using Sliding Mode Observer (SMO) 30
   3.1 System Description ..................................... 31
   3.2 $H_\infty$ SMO ........................................... 31
   3.2.1 Assumptions and their Implications ................. 31
   3.2.2 Computation of $T$ and $S$ .......................... 32
   3.2.3 Overview of the $H_\infty$ SMO ....................... 34
   3.2.4 Existence of $H_\infty$ SMO and Its Noise Attenuation Property 36
   3.2.5 Construction of the $H_\infty$ SMO .................... 40
3.2.6 $\mathcal{H}_\infty$ Attenuation and the Fault Estimate .......................... 41
3.3 $\mathcal{H}_\infty$ SMO and the LFSS .................................................. 43
3.4 Fault Accommodation Controller .................................................. 44
  3.4.1 Assumptions ............................................................................. 44
  3.4.2 Controller Construction .......................................................... 45
3.5 Simulations .................................................................................... 49
3.6 Discussion ...................................................................................... 54

4 Fast Fault Diagnosis and Fault Tolerant Control for LFSS 57
  4.1 General Problem Description ....................................................... 58
  4.2 Observer-based Residual Generator (ORG) ................................. 60
  4.3 Structured Residual Generator (SRG) .......................................... 63
    4.3.1 Structures of the SRG for the colocated LFSS ....................... 64
    4.3.2 Desired Properties of SRG .................................................... 66
    4.3.3 Feasibility Study of the Two SRG ......................................... 67
      The number of zeros in rows of $\mathcal{M}(s)$ ................................ 67
      Selection of the structure for $\mathcal{M}(s)$ .................................... 69
    4.3.4 Fault Isolation using SRG ...................................................... 70
  4.4 Fault Isolation Filter (FIF) .......................................................... 75
    4.4.1 Stability and Practicality of FIF ............................................. 75
    4.4.2 Construction of FIF ............................................................... 77
  4.5 Integrating SRG with existing FTC technique ............................ 81
  4.6 System Reconfiguration for Subsequent Failure Isolation ............ 85
  4.7 Simulation Results ...................................................................... 87
  4.8 Matrices related to the Simulation ................................................. 91
  4.9 Discussion .................................................................................... 94

5 Conclusion and Future Work .......................................................... 98

Bibliography ..................................................................................... 101
List of Symbols

Fault-Tolerant Control using $\mathcal{H}_\infty$ Sliding Mode Observer

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>Positive scalar to minimize the region of convergence of $(x, e)$</td>
</tr>
<tr>
<td>$\beta$</td>
<td>Positive constant related to the stability of the closed loop system</td>
</tr>
<tr>
<td>$\delta$</td>
<td>Positive design constant used to reduce chattering in the sliding term</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>Part of the guaranteed $\mathcal{H}_\infty$ attenuation, $\sqrt{\gamma + \psi(t_f)}$</td>
</tr>
<tr>
<td>$\nu$</td>
<td>Sliding term for the $\mathcal{H}_\infty$ SMO</td>
</tr>
<tr>
<td>$\psi(t_f)$</td>
<td>A small value of the guaranteed $\mathcal{H}_\infty$ attenuation, $\sqrt{\gamma + \psi(t_f)}$</td>
</tr>
<tr>
<td>$\rho$</td>
<td>Upper bound on the norm of $f$</td>
</tr>
<tr>
<td>$\rho_0$</td>
<td>Positive design constant for the sliding term</td>
</tr>
<tr>
<td>$\xi$</td>
<td>Disturbance</td>
</tr>
<tr>
<td>$\xi_0$</td>
<td>Upper bound on the norm of $\xi$</td>
</tr>
<tr>
<td>$\Delta$</td>
<td>Input matrix for the disturbance</td>
</tr>
<tr>
<td>$\Delta_n$</td>
<td>Input matrix for the disturbance in new coordinates</td>
</tr>
<tr>
<td>$\Lambda$</td>
<td>The convergence region of the state estimation error after finite time, see (3.10)</td>
</tr>
<tr>
<td>$\Omega$</td>
<td>Matrix related to the stability of the $\mathcal{H}_\infty$ SMO, see (3.8) for further detail</td>
</tr>
<tr>
<td>$\Phi(x)$</td>
<td>Locally Lipschitz nonlinearity in the system</td>
</tr>
<tr>
<td>$\Psi$</td>
<td>Region of convergence for $(x, e)$</td>
</tr>
<tr>
<td>$e$</td>
<td>Error of state estimate</td>
</tr>
<tr>
<td>$f$</td>
<td>Fault</td>
</tr>
<tr>
<td>$\hat{f}$</td>
<td>Fault estimate from the $\mathcal{H}_\infty$ SMO</td>
</tr>
<tr>
<td>$p$</td>
<td>Number of outputs</td>
</tr>
<tr>
<td>$q$</td>
<td>Size of fault vector, $f$</td>
</tr>
</tbody>
</table>
m  
Number of control inputs

n  
Number of states

r  
Size of disturbance vector, \( \xi \)

t_{f}  
Expected duration the system runs

u  
Control signal

w  
Positive design constant for the fault accommodation controller

x  
State behavior

\( \hat{x} \)  
State estimate

\( x_n \)  
State behavior in new coordinates, \( x_n = [x_{n1}^T, x_{n2}^T]^T \)

\( \hat{x}_n \)  
Estimated state in new coordinates, \( \hat{x}_n = [\hat{x}_{n1}^T, \hat{x}_{n2}^T]^T \)

y  
Output signal

A  
System matrix

\( A_n \)  
System matrix in new coordinates

B  
Input matrix for the control

\( B \)  
Input matrix for the fault

\( B^* \)  
Matrix that satisfies \( (I - BB^*)E = 0 \)

\( \hat{B} \)  
\( \hat{B} = \frac{BB^T}{\|B\|^2} \)

C  
Output matrix

\( C_n \)  
Output matrix in new coordinates, \( C_n = \begin{bmatrix} C_{n1} & 0 \\ 0 & C_{n4} \end{bmatrix} \)

E  
Input matrix for the fault

\( E_n \)  
Input matrix for the fault in new coordinates, \( E_n = \begin{bmatrix} E_{n1} \\ 0 \end{bmatrix} \)

H  
Weighting matrix for the error

\( H_m \)  
The Hamilton matrix that is related to the existence of the controller

\( H_n \)  
Weighting matrix for the error in new coordinates

J  
Matrix related to the stability of closed loop system and the construction of the controller

\( K_n \)  
A submatrix in the observer gain matrix in new coordinates

L  
Observer gain matrix

\( L_n \)  
Observer gain matrix in new coordinates

\( L_\Phi \)  
Lipschitz constant for \( \Phi(x) \)

P  
Matrix arising from the chosen Lyapunov function

\( P_n \)  
\( P \) in new coordinates, \( P_n = diag(P_{n1}, P_{n2}) \)

S  
Transformation matrix

T  
Transformation matrix
# Fault Diagnosis of Large Flexible Space Structure

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>η</td>
<td>Integrator state for the PID controller</td>
</tr>
<tr>
<td>γ_i^j</td>
<td>An index of $M_i(s)$ in which $M_{γ_i^j}(s) = 0$</td>
</tr>
<tr>
<td>ξ</td>
<td>Mechanical displacement of the Large Flexible Space Structure (LFSS)</td>
</tr>
<tr>
<td>Δ</td>
<td>Modal damping matrix of the LFSS</td>
</tr>
<tr>
<td>Δ̄</td>
<td>Nonzero (vibration) modes’ damping matrix of the LFSS</td>
</tr>
<tr>
<td>Λ_0</td>
<td>Input/output distribution matrix of the colocated LFSS</td>
</tr>
<tr>
<td>Ω^2</td>
<td>Modal frequency matrix of the LFSS</td>
</tr>
<tr>
<td>ΔΩ̄^2</td>
<td>Nonzero (vibration) modes’ frequency matrix of the LFSS</td>
</tr>
<tr>
<td>d</td>
<td>Modal displacement of the LFSS</td>
</tr>
<tr>
<td>$e_y$</td>
<td>$e_y = y_m - y_{ref}$</td>
</tr>
<tr>
<td>$f_a$ or $F_a(s)$</td>
<td>Actuator failures, i.e. $(F_u - I)u$</td>
</tr>
<tr>
<td>$f_s$ or $F_s(s)$</td>
<td>Sensor failures, i.e. $(F_y - I)y$</td>
</tr>
<tr>
<td>$k_i$</td>
<td>Number of zeros in $M_i(s)$</td>
</tr>
<tr>
<td>m</td>
<td>Number of inputs/outputs of the colocated LFSS</td>
</tr>
<tr>
<td>n</td>
<td>Number of elements of $ξ$ or $d$ of the LFSS</td>
</tr>
<tr>
<td>r or $R(s)$</td>
<td>Residuals from observer-based residual generator</td>
</tr>
<tr>
<td>$\bar{r}$ or $\bar{R}(s)$</td>
<td>Structured residuals</td>
</tr>
<tr>
<td>u</td>
<td>Control input to the LFSS</td>
</tr>
<tr>
<td>y</td>
<td>Actual mechanical displacement of selected locations of the LFSS</td>
</tr>
<tr>
<td>$y_m$</td>
<td>Measured mechanical displacement from the sensor of selected locations of the LFSS</td>
</tr>
<tr>
<td>$y_{ref}$</td>
<td>Constant reference for the selected locations of the LFSS</td>
</tr>
<tr>
<td>$A$</td>
<td>System matrix for the LFSS, $A = \begin{bmatrix} 0 &amp; I \ -Ω^2 &amp; -Δ \end{bmatrix}$</td>
</tr>
<tr>
<td>$B$</td>
<td>Input matrix for the LFSS, $B = \begin{bmatrix} 0 \ L \end{bmatrix}$</td>
</tr>
<tr>
<td>$\mathfrak{B}_i$</td>
<td>The set of integers from 1 to $2m$, excluding $i$ and $i + m$, i.e. ${1, ..., 2m} \setminus {i,i+m}$</td>
</tr>
<tr>
<td>$C$</td>
<td>Output matrix for the LFSS, $C = \begin{bmatrix} L^T &amp; 0 \end{bmatrix}$</td>
</tr>
<tr>
<td>$D_0$</td>
<td>Mechanical damping matrix of the LFSS</td>
</tr>
<tr>
<td>$\mathfrak{D}$</td>
<td>The set of failure locations’ indices, i.e. ${i : F^{ii} = 0}$</td>
</tr>
<tr>
<td>F</td>
<td>Combined failure matrix</td>
</tr>
</tbody>
</table>
$F^{ii}$  
$i$-th location failure indicator of the LFSS

$F_u$  
Actuator failure matrix of the LFSS

$F_c(s)$  
Laplace transform of $[f_o^T f_s^T]^T$

$F^{ii}_u$  
i-th actuator failure indicator of the LFSS

$F_y$  
Sensor failure matrix of the LFSS

$F^{ii}_y$  
i-th sensor failure indicator of the LFSS

$G_f(s)$  
Fault-to-residual transfer matrix

$G^i_f(s)$  
i-th column of $G_f(s)$

$\hat{G}^i_f(s)$  
For the colocated LFSS, we select $\hat{G}^i_f(s) = \begin{bmatrix} G^i_f(s) & G^{i+m}_f(s) \end{bmatrix}$

$G^i$  
The set which represents the indices of the zeros in $M^i$, i.e. $G^i = \{\gamma_1^i, \cdots, \gamma_{k_i}^i\}$

$H$  
Observer gain for observer-based residual generator

$I^i$  
i-th standard basis vector

$K_0$  
Mechanical stiffness matrix of the LFSS

$K_D$  
Derivative gain for PD/PID controller

$K_I$  
Integral gain for PID controller

$K_P$  
Proportional gain for PD/PID controller

$L$  
Modal input/output distribution matrix of the colocated LFSS

$\hat{L}$  
Rigid modes’ input/output distribution matrix of the colocated LFSS

$L$  
Vibration modes’ input/output distribution matrix of the colocated LFSS

$\mathcal{L}^i$  
The image of the $i$-th standard basis vector, i.e. $\text{Im}(I^i)$

$M_0$  
Inertia-mass matrix of the LFSS

$\mathcal{M}(s)$  
Structured residual generator

$\mathcal{M}^{(i)}(s)$  
i-th row of $\mathcal{M}(s)$

$\mathcal{M}^{ik}(s)$  
$(i, k)$ entry of $\mathcal{M}(s)$

$\mathfrak{N}^i$  
The union of the all images of the standard basis vectors (from 1 to 2m), but excluding the $i$ and $i+m$ standard basis vectors, see (4.28) for further details

$T$  
$F$ with the rows of zeros removed

$\mathcal{W}(s)$  
Fault isolation filter

$\mathcal{W}^{(i)}(s)$  
i-th row of $\mathcal{W}(s)$
### List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFTC</td>
<td>Active Fault Tolerant Control</td>
</tr>
<tr>
<td>CR</td>
<td>Control Reconfiguration</td>
</tr>
<tr>
<td>FA</td>
<td>Fault Accommodation</td>
</tr>
<tr>
<td>FDI</td>
<td>Fault Detection and Isolation</td>
</tr>
<tr>
<td>FIF</td>
<td>Fault Isolation Filter</td>
</tr>
<tr>
<td>FTCS</td>
<td>Fault Tolerant Control System</td>
</tr>
<tr>
<td>FTD−PID</td>
<td>Fault Tolerant Decentralized PID</td>
</tr>
<tr>
<td>IFTC</td>
<td>Integrated Fault Tolerant Control</td>
</tr>
<tr>
<td>LFSS</td>
<td>Large Flexible Space Structure</td>
</tr>
<tr>
<td>MLSOS</td>
<td>Multivariable Linear Second-Order System</td>
</tr>
<tr>
<td>ORG</td>
<td>Observer-based Residual Generator</td>
</tr>
<tr>
<td>PFTC</td>
<td>Passive Fault Tolerant Control</td>
</tr>
<tr>
<td>ROM</td>
<td>Reduced Order Model</td>
</tr>
<tr>
<td>SMO</td>
<td>Sliding Mode Observer</td>
</tr>
<tr>
<td>SRG</td>
<td>Structured Residual Generator</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Modern industrial control systems often contain a large number of components to enhance functionality and performance. A fault or failure in one of the components can produce undesirable system behaviour, leading to economic loss or disastrous consequences [12]. Prime examples of such systems are airplanes, power systems, and nuclear power plants.

Owing to the large number and the cost of components, hardware redundancy to handle failures can only be deployed for the most mission-critical components. Hardware redundancy refers to the design methodology of installing more than one component (e.g. actuator or sensor) that performs the same function. For example, a factory may have two of the same motor drives that perform the same function. One of them is used for normal operations while the other serves as a backup.

To maintain safe and reliable operation of a system while reducing the reliance on hardware redundancy, substantial research effort has been devoted to active fault-tolerant control systems (FTCSs). FTCSs are designed to maintain satisfactory performance in the presence of faults. An active FTCS usually consists of two components: the fault detection and isolation (FDI) module and the fault tolerant control (FTC) module. The FDI module is used to detect the presence of faults and to isolate or identify which faults have occurred. Fault information from the FDI module can be used by the FTC module to mitigate the effects of faults. Reference [85] provides an extensive review of FDI and FTC methods.

Large flexible space structures (LFSSs) are common in today’s aerospace industry. For example, the wide arrays of solar panels on the satellites are essentially LFSSs. LFSSs are expensive to build and it is of vital importance to improve the longevity of such systems. When a fault occurs in a system, the system may become unstable. It may lead to damage or destruction of the system, resulting in huge economic losses. Despite the importance of LFSS in future space applications, little research has been done in fault tolerant control (FTC) of LFSSs. This motivates the development of FTCS for LFSSs. Some of the work that are related
to FTC of LFSSs are [2], [3], [38], [51] and [80].

In a recent work [38], Huang has proposed applying decentralized fault tolerant control that handles sensor and actuator failures in colocated LFSSs assuming failures are known. A ‘failure’ is defined as a complete breakdown of a component (actuator or sensor). The term ‘colocated’ refers to pairing up and placing each actuator and sensor at the same location. It should be emphasized that Huang’s main focus is on the FTC of LFSS, not the fault diagnosis component. Nonetheless, to relax the assumption that failures are known, Huang suggested an innovative approach of using reference inputs to determine the failed sensor-actuator pair.

The main drawback of Huang’s fault diagnosis algorithm is the long time required to identify (or isolate) failures. Therefore, one of the two main problems documented in this thesis is to develop a fault diagnosis technique for the LFSS that is faster than Huang’s algorithm. In our early work, a $\mathcal{H}_\infty$ sliding mode observer (SMO) proposed in [69] presented itself to be a possible candidate to replace Huang’s fault diagnosis algorithm. It is well-known that SMOs are capable of producing fault estimates [4], [25], [69] and [73]. Fault estimates can be used directly to determine which sensor-actuator pairs have failed. Unfortunately, the assumptions required by the $\mathcal{H}_\infty$ SMO and other SMOs are not satisfied by the colocated LFSS model. In order to provide fast fault diagnosis, an approach based on structured residual is proposed to replace Huang’s fault diagnosis algorithm.

Although it does not seem possible to use the $\mathcal{H}_\infty$ SMO to diagnose failures in LFSSs, its ability to provide state and fault estimates that are robust to system disturbances makes it an attractive FDI module to be used in a fault tolerant control setting. Therefore, we propose a simple linear controller that uses the state and fault estimates from the $\mathcal{H}_\infty$ SMO to perform fault tolerant control.

In Section 1.1, the problems studied in this thesis will be described in greater detail. Then, the contributions are summarized in Section 1.2. Finally, we highlight the structure of the thesis in Section 1.3.

1.1 Problem Descriptions

First, consider the colocated LFSS model being controlled using the fault tolerant control scheme proposed by Huang in [38]. The proposed controllers are able to handle actuators or sensors failure while keeping the system stable and maintaining satisfactory tracking performance. However, the fault diagnosis algorithm proposed by Huang uses output steady state information to determine the failures, and thus is not appropriate if fault information is needed in real-time.

Hence, if possible, we wish to find or develop a technique to generate real-time failure
information for the existing fault tolerant controllers. This is our first problem. If our proposed fault diagnosis algorithm cannot be combined with the existing fault tolerant controllers, then new fault tolerant controllers may have to be developed. Furthermore, it is important to study the properties of the new fault diagnosis algorithm. These properties include existence conditions, fault detectability, fault isolability, practicality and more. Finally, to show the effectiveness of the new scheme, its performance should be compared to the existing scheme developed in [38].

In addition to fault diagnosis work to complement the fault tolerant controller in [38], we would like to study FTC more generally for LFSS. Recently, a $H_\infty$ SMO is introduced in [69]. It can produce state and fault estimates that are robust to disturbances, which is an attractive feature for fault tolerant control purposes. Robust fault estimates can reduce the probability of false alarms when detecting faults. Unfortunately, conditions for the existence of the $H_\infty$ SMO are not satisfied in colocated LFSS model.

Although it does not seem possible to adapt the $H_\infty$ SMO to diagnose failure for the colocated LFSS model, its attractive properties suggest it can form an important component in a more general fault tolerant control design. This motivates us to develop a fault tolerant controller which uses the $H_\infty$ SMO to produce robust fault estimates. We refer to this as our second problem. Traditionally, the FDI module and control module have often been separately designed. Since the FDI module may not always provide timely fault information, an integrated diagnosis and control design such as the one we consider in our second problem is of great interest. For such integrated schemes, closed loop stability should be established. Furthermore, it is of interest to study whether robust state and fault estimates provided by the $H_\infty$ SMO can give rise to state response that is robust to faults and disturbances.

1.2 Contributions

In this thesis, two main contributions are made. In Chapter 3, the second problem is studied. Our proposed integrated fault tolerant controller uses the $H_\infty$ SMO to provide state and fault estimates to a linear controller proposed in [64] to perform actuator fault accommodation. Simulation results show that the robustness properties of the $H_\infty$ SMO can help to produce better state response, resulting in improved performance for our IFTC. In Chapter 4, the solution to the first problem is introduced. Our proposed solution uses structured residuals to provide the same type of failure information that are generated by Huang’s fault diagnosis algorithm. This failure information can be obtained in real time, which helps to improve the overall fault tolerant control performance.

We now describe our contributions in greater detail. The main contribution in Chapter 3
is the introduction of a new integrated control scheme which combines the $\mathcal{H}_\infty$ SMO with a modified version of the linear controller introduced in [64]. Under the presence of actuator faults, closed loop stability using our proposed scheme is proven using Lyapunov analysis.

It is worth emphasizing that this is one of the few mathematically rigorous studies that use a fault estimate from a SMO to perform fault accommodation. The fault accommodation controller uses the estimated fault obtained from the $\mathcal{H}_\infty$ SMO to cancel out the effect of the actual actuator fault. Similar schemes were used in [46] and [47]. The study conducted by [46] assumed the system has a special structure, which is exploited for FDI and FTC purposes. Reference [47] uses a nonlinear adaptive observer followed by a nonlinear controller, which requires high gain switching. Typically, switching signals are hard to implement on microcontrollers. High gain requires additional power amplification circuitry and may lead to control signal saturation. To overcome these limitations, we chose to use the linear controller proposed by [64].

Finally, our proposed integrated fault tolerant control scheme is compared to two other integrated fault tolerant control schemes. The first scheme uses a normal SMO to estimate faults, whereas the second scheme uses the nonlinear adaptive observer from [47]. Simulation results suggest that our proposed scheme enjoys enhanced robustness to disturbances, which can be attributed to the ability of the $\mathcal{H}_\infty$ SMO to attenuate disturbances.

In Chapter 4, the main contribution is the introduction of the structured residual generator as an alternative to the fault diagnosis algorithm proposed in [38]. An observer-based residual generator combined with the so-called “fault isolation filter” gives rise to the structured residual generator. To design the fault isolation filter, important properties of the structured residual generator have to be first studied. These important properties include fault isolability, stability of the structured residual generator and whether the implementation of the structured residual generator requires differentiators.

We introduce four equivalent conditions of fault isolability for the structured residual generator. They are useful for constructing the fault isolation filter and checking if the computed structured residual generator has the right form. By using one of the conditions for fault isolability, the design of the fault isolation filter can be all done automatically using symbolic computation software. Then, we study the connection between structured residuals with the so-called “combined failure matrix”, which is important for the applicability of the fault tolerant controllers proposed in [38]. Lastly, to make subsequent fault isolation possible after a fault has occurred, the concept of system reconfiguration is introduced.
1.3 Thesis Structure

This thesis is organized as follows. In Chapter 2, an introduction to some background and literature review related to our work is given. A fault accommodation controller using the fault estimate from the $\mathcal{H}_\infty$ SMO is developed in Chapter 3. Detailed stability analysis is provided. Simulation results are presented, which suggest our scheme makes the system more robust to noise and faults when compared to other existing fault estimators. Chapter 4 describes our proposed solution of generating the failure information for the purpose of fault tolerant control of LFSS. We develop the concept of structured residuals to provide real time fault/failure diagnosis information to the fault tolerant controllers proposed by Huang. Finally, Chapter 5 includes concluding remarks and future research directions.
Chapter 2

Background and Literature Review

In this chapter, the background of the necessary materials required to understand our results is provided and the literature closely related to our work is reviewed. First, some commonly used mathematical notation will be introduced. Then, a study on the large flexible space structure is performed. Followed by that, we review different schemes used in fault tolerant control systems. Lastly, two fault tolerant control laws proposed in [38] for the colocated large flexible space structure are introduced.

2.1 Mathematical Background and Notations

The mathematical notations used in this thesis are fairly standard in the control theory literature. Usually, capital Roman and Greek letters will be used to represent real matrices and sets in this document (with a few exceptions). Lower case letters typically represent real scalar and real vectors. Since this thesis covers two different problems, the variables defined in Chapter 3 and Chapter 4 do not carry the same meaning unless stated otherwise. Furthermore, the notation styles in Chapter 3 and 4 are slightly different.

First, we present notations that are commonly used in both chapters. If $A$ is a $m \times n$ real matrix, we write $A \in \mathbb{R}^{m \times n}$. Similarly, if $A$ is a $m \times n$ complex matrix, we write $A \in \mathbb{C}^{m \times n}$. Also, when $a$ is a real column vector with length of $n$, we use the notation $a \in \mathbb{R}^n$. Although seldom used, a row vector, $b$ is denoted $b \in \mathbb{R}^{1 \times n}$. The variable $t$ always represents time. System models usually contain time dependency. A signal $x(t)$ would typically be represented simply as $x$. The identity matrix is denoted as $I$.

In Chapter 3, the operation $\| \cdot \|$ represents the Euclidean norm of a vector or matrix. The symbol $\| \cdot \|_{L_2}$ represents the $L_2$-norm of a signal. The computation of $L_2$-norm of a signal is covered in Chapter 3. A positive definite matrix $X$ is written as $X > 0$. Similarly, $X < 0$ means $X$ is a negative definite matrix. The reader will also come across many variables
subscripted with an \( n \). These variables are those in a new coordinate system (i.e. a change of basis is performed). For example, \( L \) would represent the matrix in original coordinate, whereas \( L_n \) is the matrix in a new coordinate system. Also, we use the term \( \text{diag}(\cdot) \) to represent a diagonal matrix or block diagonal matrix.

In Chapter 4, we deal with linear systems that are represented in the frequency domain. The typical variable \( s \) will be used to represent the Laplace (or frequency) variable. If \( f \) is a signal in the time domain, then its corresponding Laplace transform variable is \( \mathcal{F}(s) \).

In Chapter 4, all vector and matrix element indexing is represented as a superscript. For example, if \( K \) is some matrix, its \((i, j)\) entry is represented as \( K^{ij} \). Also, the concept of standard basis is used quite often in the proofs of Chapter 4. We represent the \( k \)-th standard Euclidean basis vector as \( I^k \), i.e. the \( k \)-th column of an identity matrix. Subscripts are used to give a variable special meaning. For example, we call the actuator fault signal \( f_a \) and the sensor fault \( f_s \). Sets are typically denoted by capital letters with Fraktur script font style. Lastly, we want to emphasize that the same letter with different font styles does not represent the same variable.

### 2.2 Large Flexible Space Structures (LFSS)

Large flexible space structures (LFSSs) are often characterized by the following three important characteristics [7]:

1. LFSSs are distributed parameter systems. This means that the system has infinite state and they are best modeled by partial differential equation.

2. LFSSs have many low resonant vibration frequencies.

3. LFSSs have little natural damping. LFSSs typically take a long time to reach steady state.

In order to simplify controller and observer design, the distributed parameter system model is often discretized, via methods such as the finite element method, to obtain a finite dimensional model. In [7], the author refers to a finite dimensional model as a reduced order model (ROM). The ROM can be further linearized if it is a nonlinear model. Furthermore, the fact that LFSS having many low resonant vibration frequencies allows one to perform modal truncation. Modal truncation is the process of approximating the ROM by removing the higher frequencies mode from the original one. Other references related to LFSS include [49] and [67].
The obtained reduced order finite dimensional LFSS model is typically a multivariable linear second-order system (MLSOS). The dynamics of a MLSOS is typical given as follows:

\[ M_0 \ddot{\xi} + D_0 \dot{\xi} + K_0 \xi = \Lambda_0 u \]

with certain assumptions on \( M_0, D_0 \) and \( K_0 \). More details can be found in Section 2.2.1. It would be beneficial to study the stability, controllability and observability of a MLSOS in terms of \( M_0, D_0 \) and \( K_0 \). The stability of MLSOS has been studied in great detail by researchers. One of the recent works can be found in [24]. The authors have proposed necessary and sufficient conditions for the stability of MLSOSs in terms of \( M_0, D_0 \) and \( K_0 \). As for the observability and controllability of a MLSOS, the authors of [53] provide necessary and sufficient conditions for these properties to hold.

The control of LFSS has generated much interest in the control community. LFSSs usually have low natural damping. Therefore, major focus has been placed on using control techniques to improve damping and to suppress vibration. An early significant work that is related to control of LFSSs was performed by Mark Balas in [5]. Balas proposed to apply a simple state feedback control to a flexible system. The flexible system resembles a LFSS. The paper also investigates the phenomenon called spill-over which occurs due to unmodeled dynamics from the original distributed parameter model. It was observed that the spill-over phenomenon can lead to instabilities when state feedback control is used. Later on, in [6], Balas proposed to use a method called direct velocity feedback to increase the damping of a LFSS without triggering instabilities caused by spill-over. Reference [7] provides an excellent survey of work related to LFSS up to 1981.

Reference [59] compared two different control methods for LFSSs, namely independent modal control and coupled control. Independent modal control uses the modal representation of a LFSS to design controllers for each independent modal subsystem. Note that independent modal control is one type of decoupling control. Coupled control, on the other hand, refers to control that requires all the state or output information. It was found in [59] that the independent modal control presents more benefits in terms of computational efficiency and the energy required for control.

Another work that employed a scheme that is similar to decoupling control is [78]. It uses a simple decentralized PID control that is capable of eliminating the spill-over problem. A more detailed description can be found in Section 2.4. Positive position feedback was proposed in [26] which is also capable of stabilizing the system without being affected by spill-over. The proposed method is also simple to realize. An overview of active control technology suitable for applying to the LFSS is provided in [41]. The paper reviews concepts related to modeling of
LFSS, model simplification, system identification, tracking control and vibration suppression. More recently, reference [37] proposes a variable structure control that employs sliding mode control along with positive position feedback to perform tracking.

### 2.2.1 Colocated LFSS Model

In this thesis, we consider a colocated model for the LFSS. A colocated LFSS means that the actuators and sensors are placed in the same spatial location. It is shown in [11] and [67] that colocation enables a LFSS to be stabilized via the concept of passivity. Without colocation, a method called Q-parameterization can be used to stabilize a limited number of modes of the LFSS [11].

Let $n$ be the number of state resulting from the discretization process, $m$ is the number of sensor-actuator pairs and $\xi \in \mathbb{R}^n$ is the mechanical displacement (relative to some reference in space). The reduced-order, linear and colocated LFSS mechanical model taken from [7], [38] and [78], is given as:

$$
M_0 \ddot{\xi} + D_0 \dot{\xi} + K_0 \xi = \Lambda_0 u \tag{2.1}
$$

$$
y = \Lambda_0^T \xi \tag{2.2}
$$

where $M_0 \in \mathbb{R}^{n \times n}$ is a positive definite matrix representing the inertia-mass matrix, $D_0 \in \mathbb{R}^{n \times n}$ is a positive semidefinite matrix that denotes the damping matrix, $K_0 \in \mathbb{R}^{n \times n}$ is another positive semidefinite matrix acting as the stiffness matrix and $\Lambda_0 \in \mathbb{R}^{n \times m}$ can be viewed as an input/output distribution matrix. The vectors $u \in \mathbb{R}^m$ and $y \in \mathbb{R}^m$ are the input and output vectors, respectively. To understand more about the physical meaning of the model, refer to [49].

To gain more insight into the vibration dynamics of the LFSS, the authors in [7], [49] and [68] use a transformation on (2.1) and (2.2) to obtain the modal representation. The transformation is given as $d = T^{-1} \xi$, where $T \in \mathbb{R}^{n \times n}$ is some transformation matrix and $d$ is the state in new coordinates. The modal representation of the colocated LFSS is given as:

$$
\ddot{d} + \Delta \dot{d} + \Omega^2 d = Lu \tag{2.3}
$$

$$
y = L^T d \tag{2.4}
$$

where $\Delta$ and $\Omega$ are the modal damping and modal stiffness matrices. $L$ is the input distribution matrix in new coordinates. For the exact derivations of (2.3) and (2.4), see [38].

Note that (2.3) is a second-order vector differential equation. In order to study the LFSS, it is best to recast (2.3) as a first order ordinary differential equation. We will also introduce the
possibility of failure in the actuators and sensors of the LFSS. Define $y_m$ to be the measured output obtained from the sensors. Note that this may not be the same as the actual output from the system, $y$. Let $x = [d^T, d^T]^T$, the system (2.3) and (2.4) with failures can be described as follows:

$$
\dot{x} = \begin{bmatrix} 0 & I \\ -\Omega^2 - \Delta \end{bmatrix} x + \begin{bmatrix} 0 \\ L \end{bmatrix} F_u u
$$

(2.5)

$$
y_m = F_y [L^T, 0] x
$$

(2.6)

where $F_u, F_y \in \mathbb{R}^{m \times m}$ are diagonal matrices with their diagonal elements being 0 or 1. $F_u$ contains information on which actuators have failed, whereas $F_y$ contains information on which sensors have failed. When the $i$-th actuator to have a failure, the $i$-th diagonal element of $F_u$, $F_{ii}^u$ is then equal to 0. The same holds true for the case of sensor failure.

In our work, we assume for simplicity that either an actuator or a sensor in the LFSS is working normally or it fails completely. It is also common for partial failures to occur and there are many works that consider such failures. One example is reference [77]. Handling partial failures in fault tolerant control of LFSS is beyond the scope of this work. However, our work on fault accommodation using the $H_{\infty}$ SMO presented in Chapter 3 allows for general additive fault models.

The matrices $\Omega^2$ and $\Delta$ have additional properties and interpretations. The matrix $\Omega$ is always diagonal with all its diagonal elements being equal to or greater than 0. On the other hand, $\Delta$ is a positive semi-definite matrix. The diagonal elements of $\Omega^2$ can be interpreted as the vibration modes (or the squared of the frequency) of the system. The vibration modes that are 0 are called rigid body modes. It turns out that the number of rigid body modes of a system is particularly important for the set point tracking problem for the LFSS [19].

Part of this thesis is an extension of the work presented in [38], which gives an extensive treatment of the constant tracking problem for the LFSS under sensor-actuator failure. For this reason, we shall give a brief discussion on the rigid body modes of the LFSS. **Let the number of rigid body modes be $\hat{n}$**. Using the concept of rigid body mode, one can rewrite the following matrices via permutation of the states to be:

$$
\Omega^2 = \begin{bmatrix} 0_{\hat{n}} & 0 \\ 0 & \bar{\Omega}^2 \end{bmatrix}, \quad \Delta = \begin{bmatrix} 0_{\hat{n}} & 0 \\ 0 & \bar{\Delta} \end{bmatrix}, \quad L = \begin{bmatrix} \bar{L} \\ \bar{L} \end{bmatrix}
$$

(2.7)

where $0_{\hat{n}}$ represents an $\hat{n} \times \hat{n}$ zero matrix, $\bar{\Omega}^2 \in \mathbb{R}^{(n-\hat{n}) \times (n-\hat{n})}$ represents the vibration modes matrix which is diagonal and positive definite, $\bar{\Delta}$ is the vibration damping matrix that is strictly positive definite, $\bar{L} \in \mathbb{R}^{\hat{n} \times m}$ and $\bar{L} \in \mathbb{R}^{(n-\hat{n}) \times m}$. As suggested by (2.7), controlling the rigid
body mode may result in excitation of the vibration modes.

### 2.2.2 Failure Matrices

The actuator failure matrix, $F_u$ and sensor failure matrix, $F_y$ have some important properties that can be used for fault tolerant control purposes. In Section 2.4, the concept of the combined failure matrix is used to perform control reconfiguration on the controllers after failures have occurred. The combined failure matrix, $F \in \mathbb{R}^{m \times m}$ is defined as follows:

$$F = F_y F_u$$

Note that $F$ is a diagonal matrix with its diagonal entries being 0 or 1. It represents the location of the failures or, in other words, it shows which sensor-actuator pairs have failed. When the $i$-th row of $F$ is 0 (or the $i$-th diagonal element of $F$ is 0), that implies the $i$-th sensor-actuator pair has failed.

To illustrate the relationship between $F$ and the sensor-actuator pairs, consider the following simple example. Suppose there are 3 sensor-actuator pairs (or 3 locations). Assume sensor 2 has a failure and therefore $F_y = diag(1, 0, 1)$. On the other hand, all actuators are healthy, implying $F_u = I_3$. Hence, when location 2 experiences a failure, it implies $F = diag(1, 0, 1)$.

The combined failure matrix has a special property that allows one to use it for control reconfiguration. Assume that the up-to-date failure information is available. Then, $F$ satisfies the following equations:

$$F F_u = F_u F = F$$

$$F F_y = F_y F = F$$

This property is later used by Huang [38] to reconfigure the controller to maintain satisfactory operation of the LFSS.

Suppose that a fault has occurred, then $F$ can be used for performing system reconfiguration. System reconfiguration refers to the action of not using any of the failed sensor-actuator pairs in a faulty system. In Chapter 4, we will introduce the use of structured residual generator to isolate the failed locations. System reconfiguration allows the structured residual generator to isolate (or diagnose) subsequent faults. Let $m^* = \text{rank}(F)$ and $T \in \mathbb{R}^{m^* \times m}$ be:

$$T = F \text{ with all 0 rows removed}$$
Then, \( F \) can be decomposed to be:

\[
F = T^T T
\]  

(2.11)

The matrix \( T \) can be used to pick out the healthy sensors and actuators in the system. \( T \) also has a similar property to \( F \). They are given as follows:

\[
F_u T^T = T^T
\]  

(2.12)

\[
T F_y = T
\]  

(2.13)

Define:

\[
A = \begin{bmatrix} 0 & I \\ -\Omega^2 & -\Delta \end{bmatrix}, B = \begin{bmatrix} 0 \\ L \end{bmatrix}, C = \begin{bmatrix} L^T \\ 0 \end{bmatrix}
\]

For the case of actuators, one can choose \( u = F \bar{u} \), where \( \bar{u} \) represents the control signal applied to the system when no failures have occurred. This is a control reconfiguration strategy proposed in [38], which simply implies no control is performed for the failed locations. Using (2.12) and \( u = F \bar{u} \), the term \( BF_u u \) in (2.5) can then be simplified to be:

\[
BF_u u = (BT^T)(T \bar{u})
\]

Note that \( T \bar{u} \) can be viewed as the control signals from the healthy locations. As for the case of sensors, take \( \bar{y}_m = T y_m \), where \( \bar{y}_m \) represents all the measured output of the healthy locations.

To perform system reconfiguration, take \( u = F \bar{u} \) and \( \bar{y}_m = T y_m \). This would result in a system with only healthy sensor-actuator pairs, which results in the model presented below.

\[
\dot{x} = Ax + (BT^T)(T \bar{u})
\]  

(2.14)

\[
\bar{y}_m = TCx
\]  

(2.15)

The reconfigured system model will be considered again in Chapter 4.

### 2.3 Overview of Fault Tolerant Control Systems

All of our work in this thesis is related to fault tolerant control systems (FTCS). This section introduces various concepts in FTCS as well as recent developments in this area. Typically, FTCS can be categorized into passive FTCS and active FTCS. Whether a FTCS is active or passive depends on the controller design.

An active FTCS usually consists of two major components, namely: i) the fault detection and isolation (FDI) module, and ii) the control reconfiguration (CR) or fault accommodation
(FA) module. Figure 2.1 shows the general structure of a typical active FTCS. Typically, the FDI module would require the knowledge of the input and output to detect and isolate a fault. When an observer-based approach is used for FDI, one can use the estimated states from the observer for control purposes. Our scheme proposed in Chapter 3 is an example of this. The dotted lines in the figure represent optional information for the controller to perform basic control objectives. These are ultimately dependent on the design of the CR/FA module. One can either use estimated state or output to perform feedback control in the CR/FA module.

In Chapter 4, output feedback is used to stabilize the faulty system and to make the healthy locations track the constant reference. This is different from the scheme used in Chapter 3, which uses the estimated state to perform control.

The term ‘fault detection’ refers to the process of using the input and output to determine whether system behaviour is normal. Fault detection can be performed via different techniques like residual generation or fault estimation. A residual is a signal that is nominally zero when there is no fault present, and it is nonzero (for all time when the fault is present) in the event of a fault. This concept was developed as early as the 1970’s using different techniques, which can be found in [9], [18] and [48].

Reference [56] proposed to use a geometrical method to minimize the effect of noise on the residual generated by using the parity space approach proposed in [18]. Reference [65] proposed to augment the system with a stable filter to detect sensor faults using the schemes provided in [9] and [48]. In [21], Ding et al. showed that one can increase the order of the parity relation proposed in [18] to improve the robustness of the residual generated from the parity relation. Ding et al. also proposed the different algorithm to design the residual generator in [22] such that its robustness against disturbance and its sensitivity with respect to faults are
optimized. Reference [30] investigated methods to minimize the order of residual generator (as a system) for more efficient implementation. Nyberg studied the criteria for weak and strong fault detectability in [63]. More recently, [17] proposed to use higher order sliding mode differentiator to detect faults which requires fewer assumptions than conventional unknown input observers. A general overview of fault detection schemes is provided in [29].

Typically, just knowing that a fault has occurred is not sufficient for the CR/FA component to handle the fault. ‘Fault isolation’ is associated with the process of identifying what faults might have occurred. One way to achieve fault isolation is to design a residual such that the residuals react differently to different types of faults. Another way is to use fault estimators such as the ones proposed in [25], [46] and [47]. The process of fault detection and fault isolation is usually referred together as FDI.

A major breakthrough in fault isolation began with the work of Massoumnia. Massoumnia first started to restudy the techniques proposed by Beard [9] and Jones [48] using geometrical techniques. It was found to be similar to the restricted control decoupling problem. This work can be found in [57]. Then, Massoumnia et al. proposed an algorithm to design an observer capable of isolating fault using geometric concepts in [58]. The techniques used in [58] are still not widely adopted in the FTCS community. We believe that these techniques can be applied elsewhere for the purpose of isolating faults.

Reference [16] employs the concept of unknown input observers and disturbance decoupling to make the Beard and Jones fault detection filter more robust to disturbance. Fault isolation is achieved by designing the residual to react to each fault in a unique direction. The authors in [55] propose to use the concept of the fault detectability matrix and use its inverse to design an observer-based residual generator. Each of the resulting residuals has a one-to-one correspondence with each of the faults. A fault diagnosis technique capable of identifying multiplicative sensor and actuator faults was presented in [77]. This technique employs adaptive updating rules. In [71], eigenstructure assignment is also exploited in the design of unknown input observer for isolating faults. The book [15] covers a wide range of topics in fault detection and fault isolation using unknown input observers.

In [62], the authors used the concept of $\mathcal{H}_\infty$ filtering to design a fault estimator, which is capable of estimating a fault while reducing the effect of disturbance in the fault estimate. The design of the fault estimator is performed by solving a linear matrix inequality. Statistical methods such as hypothesis testing are used in [8] to detect and isolate faults. Reference [52] introduced an unknown input proportional input observer to perform robust state and fault estimations. More recently, authors of [54] proposed to use $\mathcal{H}_\infty$ filtering to attenuate the effect of disturbances on a residual generator. The residual generator is designed such that each of its residuals has a one-to-one correspondence to each fault. More information on other FDI
techniques can be found in the following survey papers [28], [40], [42], [44], [43] and [76].

Most of the references related to fault isolation mentioned up to this point assume that the number of faults that can occur is less than the number of outputs. The LFSS has $m$ outputs but there are $2m$ possible types of failures. Hence, many of these approaches cannot be applied to LFSS directly. One possible approach to solve the fast fault diagnosis problem for the LFSS is to use structured residuals, which are discussed later in Section 2.3.2.

Using the knowledge of the type of faults that have occurred (i.e. the isolated faults), the design of the CR/FA component becomes a problem of whether there exists a controller that could handle all of the possible faults in a particular fault category. A fault category refers to a group of faults that have some attributes in common. For example, one can group all actuator faults into one category and all sensor faults into another. Depending on how much information the fault category can provide to the CR/FA component, a passive fault tolerant control (FTC) or active FTC methodology may be used as the FTC method.

A FTC method, which does not use any fault information but uses its robustness property instead to handle the fault, is usually referred to as passive fault tolerant control (PFTC). Note that the FDI module may or may not be present in a passive FTCS because PFTC does not require the knowledge of the fault occurred. Reference [61] introduces a passive linear fault tolerant controller design using the Youla-Jabr-Bongiorno-Kucera parameterization. Another passive fault tolerant controller is designed specifically for a rigid body spacecraft in [14]. In both references, a FDI module is not present in their proposed scheme.

The disadvantage of PFTC is that it does not use fault information, and hence must work for all possible faults. This is clearly a strong requirement and limits its applicability. Thus, many researchers have assumed that fault information is available for performing FTC. With the knowledge of the type of fault that has occurred, a controller that incorporates fault information can be designed to guarantee the stability of the system while satisfying (partially) other control requirements. Such a FTC method, that actively seeks information about the faults so that it can be incorporated into the controller, is called active fault tolerant control (AFTC). Two methods are available to perform AFTC, namely control reconfiguration and fault accommodation. Control reconfiguration changes the controller so that the system under failure can still achieve a reduced set of objectives. Fault accommodation adjusts the controller so that the effects of faults on the system are mitigated.

We will introduce a few works that are related to AFTC. In [86], the authors proposed a new controller structure called generalized internal model control that can handle fault. Reference [84] used model following and command input management techniques to perform control reconfiguration. Reference [20] developed a FTCS for electric vehicle and hybrid electric vehicle. However, the paper does not put much focus on the FDI process. In [72], the authors
proposed a pseudo-inverse method for designing control gain for both healthy and faulty system via the minimization of the Frobenius norm of certain matrices. A more recent work [36] introduced the use of integral sliding mode and control allocation to perform fault tolerant control. Reference [85] provided an extensive review of FDI and FTC methods. In all of these references, the interaction between the FDI and CR/FA module is not well-studied. Therefore, it is unclear that the two components will provide satisfactory fault tolerant control. This motivates the study of integrated fault tolerant control schemes. Integrated fault tolerant control schemes consider the FDI module and CR/FA module together to ensure satisfactory performance when faults occur. The literature review on this topic can be found in Section 2.3.5.

In the following subsections, we will provide an overview of some of the literature related to the FTC design approach of this thesis. Topics covered include residual generation, structured residual, fault estimation using sliding mode observers and recent advances in integrated fault tolerant control schemes.

### 2.3.1 Residual Generation

We begin by discussing the classical fault detector: residual generators. A residual generator produces a signal that is nominally 0 when no fault is present, and it becomes nonzero when a fault is present. Some of the earlier significant works related to linear residual generator can be found in [9], [18] and [48]. The two popular residual generation techniques are: (i) parity space approach (developed in [18]), and (ii) observer-based approach (developed in [9] and [48]).

Consider a typical linear system with fault, \( f \in \mathbb{R}^k \) and disturbance, \( d \in \mathbb{R}^l \) as follows:

\[
\dot{x} = Ax + Bu + E_x d + F_x f \\
y = Cx + Du + E_y d + F_y f
\]  

(2.16)

where \( x \in \mathbb{R}^n, u \in \mathbb{R}^m \) and \( y \in \mathbb{R}^p \). All matrices are of appropriate dimensions. Disturbances are typically considered in the literature. Large disturbances may cause false alarm in the FDI unit. Thus, it is beneficial to consider disturbances in the model such that the residual generator can be designed to eliminate or minimize the chances of false alarm occurring.

We begin by discussing the parity space approach developed in [18]. The parity space approach utilizes two vectors, \( Y_q \) and \( U_q \) to produce the residual signal, \( r \in \mathbb{R}^h \). \( Y_q = [y^T, \dot{y}^T, \ldots, (y^{(q)})^T]^T \) is the vector of the measured output and all of its derivatives up to the \( q \)-th order derivatives. \( U_q = [u^T, \dot{u}^T, \ldots, (u^{(q)})^T]^T \) is the vector that concatenates the input and all of its derivatives up to the \( q \)-th order derivatives. The scalar \( q \) is a design parameter. Two special constant matrices, \( Q \in \mathbb{R}^{h \times (q+1)p} \) and \( T_{u,q} \in \mathbb{R}^{(q+1)p \times (q+1)m} \) are evaluated according to
A method presented in [21]. \( r(t) \) can be calculated as follows:

\[
    r(t) = QY_q(t) - QT_{u,q}U_q(t) \tag{2.17}
\]

Another approach commonly used is the observer-based residual generator. First, define the term input observable given by [58] (the term input observable was first coined by [70]) because it is easier to understand.

**Definition 2.1.** A triplet (or system) \((C, A, B)\) is input observable if the mapping \(B\) is one-to-one and the intersection of the image of \(B\) with the unobservable subspace of \((C, A)\) is \(\{0\}\).

For the design of the observer-based residual generator, assume that \(E_x\) and \(E_y\) are 0 in (2.16) and the triplet (or system) \((C, A, F_x)\) is input observable. The assumption that the pair \((C, A, F_x)\) is input observable informally means that a nonzero \(f\) (i.e. a fault has occurred) will affect the residual, \(r\). When \(F_x\) is one-to-one, this guarantees \(F_xf \neq 0\).

One possible observer-based residual generator to detect a fault is to use the well known Luenberger observer. It is given as follows:

\[
    \dot{\hat{x}} = Ax + Bu + L(C\hat{x} - y) \\
    r = C\hat{x} - y
\tag{2.18}
\]

where \(L \in \mathbb{R}^{n \times p}\) is the observer gain. Suppose that the observer is stable and steady state has occurred. When a fault occurs, the term \(\hat{x} - x \neq 0\). Hence, \(r\) is nonzero, which indicates a fault has occurred. A more general form for the observer-based residual generator can be found in [15].

Typically, residual generator can be easily designed to detect fault occurrence. However, the problem of designing a residual generator that can help to isolate for faults requires more design effort. The types of residual that are capable of isolating faults include structured residual and fixed direction residual [15], [32]. In the next subsection, an overview of structured residual will be provided since it will be used later on in Chapter 4.

### 2.3.2 Structured Residual

Structured residuals are residuals designed such that each of them only reacts to a certain subset of faults. Let \(\bar{r}^i(t)\) be the \(i\)-th structured residual. Define \(\Omega^i\) to be the span of some designer-selected standard basis vectors (which is of the same dimension as the fault vector) that \(\bar{r}^i(t)\) would react to. A more mathematical description of a structured residual is: \(\bar{r}^i(t) \neq 0\) if and only if \(f \in \Omega^i\).
Gertler and Singer focus on the problem of fault isolability of residual generators and proposed a framework in which one can obtain a residual generator capable of isolating faults in [33]. The framework involves performing a “model transformation” on the fault-to-residual transfer matrix. However, no systematic method has been given to choose the “transformation”. “Model transformation” refers to the action of multiplying another transfer matrix to the residuals. To avoid confusion with the process of transforming one state space model to another, we refer to this as filtering the residuals.

The overview of the framework is described as follows. First, obtain a set of residuals, \( r(t) \). Let \( R(s) \) be the Laplace transform of \( r(t) \), \( F(s) \) be the Laplace transform of the fault vector \( f(t) \), and \( G_f(s) \) be the fault-to-residual transfer matrix. Using the input-output model, the residuals are related to faults as shown:

\[
R(s) = G_f(s)F(s)
\]

The second step is to design which subset of faults will affect which element of the structured residual, \( \bar{R}(s) \). The last step is to find \( W(s) \) to filter \( R(s) \) to obtain \( \bar{R}(s) \). Note that one can write:

\[
\bar{R}(s) = W(s)R(s) = W(s)G_f(s)F(s) = M(s)F(s)
\]

where \( M(s) = W(s)G_f(s) \).

The second step is equivalent to selecting a structure for the transfer matrix \( M(s) \). In particular, one needs to choose the corresponding elements of \( M(s) \) to be zeros if one wants to stop an element of structured residual to react to certain faults. We illustrate this concept by an example. Suppose we have 1 structured residual and 4 faults, i.e. \( f(t) \in \mathbb{R}^4 \). This implies \( M(s) \in \mathbb{C}^{1 \times 4} \). To decouple the first and the third fault from the structured residual, choose \( M(s) = [0\,*\,0\,*] \), where the ’*’ represents strictly nonzero transfer function.

The final step of designing \( W(s) \) can be performed in various ways. In [33], the approach suggested requires solving linear equations differently depending on different cases. No systematic approach is given to solve the linear equations. Gertler proposed another method based on matrix inversion of \( G_f(s) \) in [31]. Therefore, \( G_f(s) \) is required to be square and invertible. These assumptions limit the applicability of this scheme. However, it is important to note that stability and causality of \( W(s) \) have been studied in this paper. In [34], Gertler presents an approach to construct structured residual using residuals generated by the parity-space approach.
directly in the frequency domain. The book [32] covers detailed information related to the
design of structured residuals.

In the colocated LFSS, the number of faults is greater the number of measured outputs.
Structured residual can handle cases where the number of faults is greater than the number of
measured outputs. Thus, structured residual is a good candidate to solve the failure diagnosis
problem for the LFSS. Other fault isolation techniques typically require assumptions that are
not satisfied by LFSSs. Detailed discussion on the application of structured residuals to isolate
failure in LFSSs is described in Chapter 4.

2.3.3 Sliding Mode Control Theory

In this thesis, we shall use a sliding mode observer to provide fault information to a fault
tolerant controller. Since most of the theory of the sliding mode observer originated from
sliding mode control, this subsection will cover the basics of sliding mode control. The basic
idea of sliding mode control is to use control to force some combinations of the state dynamics
to go to zero. In other words, the system dynamics are forced to stay on a manifold. Typically,
the control signal used is a discontinuous signal.

Sliding mode control is often associated with the term “variable structure control (VSC)”
or “variable structure system (VSS)”. Loosely speaking, these terms are associated the process
of switching one control law to another. References [39] and [74] provide excellent surveys
of the basics of VSC/VSS and sliding mode control. Reference [39] further provides recent
development of theories and application of VSC/VSS and sliding mode control.

In this section, we shall briefly review the concept of sliding mode control for linear sys-
tems. The materials presented here are largely taken from reference [39] and [66]. Consider a
typical linear system with all its state variables being accessible:

\[ \dot{x} = Ax + Bu \] (2.19)

where \( x \in \mathbb{R}^n, u \in \mathbb{R}^m \) and \( \text{rank}(B) = m \).

A sliding mode controller works in the following way. First, the sliding mode controller
would force the state to go to the so-called sliding surface and maintain the state trajectory
on the sliding surface. The state trajectory on the sliding surface would depend on the system
dynamics. Often, one can further design a lower order controller to redesign the dynamics on
the sliding surface.

Typically, the design of the sliding mode controller starts with the selection of a sliding
(or switching) function. Let \( s(x) = Cx \in \mathbb{R}^m \) be the sliding/swtiching function, where \( C \in \)
The sliding surface is defined as the set:

\[ \mathcal{S} = \{ x : s(x) = 0 \} \]

With the sliding function chosen, the next step is to design a sliding mode controller to stabilize the sliding surface. It should be noted that there are many choices available for the sliding mode controller. We will present the simplest (and perhaps most common) sliding mode control law. Define:

\[ \text{sgn}(z) = \begin{cases} 
1 & \text{when } z > 0 \\
-1 & \text{when } z < 0 
\end{cases} \]

where \( z \) is some scalar variable. Note that this definition of the sgn function is different from the conventional definition, where \( \text{sgn}(0) = 0 \). This is important later on in this section. Also note that one can represent \( \text{sgn}(z) = \frac{z}{\|z\|} \). The sliding mode controller is given as:

\[ u_i = -k_i \text{sgn}(s_i(x)), \forall i \in \{1, \ldots, m\} \quad (2.20) \]

where \( u_i \) is the \( i \)-th element of the control input \( u \), \( s_i \) is the \( i \)-th element of the sliding function \( s \) and \( k_i \) is a positive scalar.

The core design step is to choose the right values for \( k_i \) such that the reachability condition is met. The reachability condition is used to check whether the states will go to and stay on the sliding surface. When the states stay on the surface, this implies \( \dot{s}(x) = 0 \). The reachability condition can be obtained by using Lyapunov analysis and it is given as \( s^T(x)\dot{s}(x) < 0 \) when \( s \neq 0 \) [39]. A more general reachability condition that can guarantee finite time reachability can be found in [10].

After the proper values of \( k_i \) are selected, it is beneficial to study the behavior of the system when ideal sliding motion is achieved, i.e. \( s(x) = \dot{s}(x) = 0 \). Note that the sliding mode controller (2.20) has an undefined value when sliding motion is achieved. This undefined value of the control input during ideal sliding motion gives rise to the concept of equivalent control. Let the equivalent control input be \( u_{eq} \) and suppose that \( \text{rank}(C) = m \). The equivalent control input during sliding motion can be calculated as follows:

\[ u_{eq} = -(CB)^{-1}CAx \]

Using this expression in (2.19), one can simplify the system into a reduced-order system, where the states of the reduced-order system represent the dynamics on the sliding surface. More detailed information can be found in subsection 1.3.1 of [66]. The next subsection introduces sliding mode observers and how they are used to estimate faults.
2.3.4 Sliding Mode Observers

Sliding mode observers (SMOs) have been introduced in the FDI literature since the early 1990’s. SMOs have the capability to perform fault estimation as well as robust state estimation, making them an appealing FDI component in integrated fault tolerant control designs.

Reference [27] was one of the earlier works that proposed to use SMO to perform fault detection. Edwards et al. expanded this idea and provided rigorous mathematical analysis for using SMO to estimate faults [25]. The authors in [73] improved the fault estimate from the SMO by incorporating $H_\infty$ robustness in the fault estimate. Also, [4] presented a sliding mode controller which can accept fault estimates to perform FTC. Recently, a $H_\infty$ SMO is proposed in [69], which can estimate faults while minimizing the effects of noise in the state and fault estimation error.

In the mentioned literature, SMOs are typically designed for a linear system or a linear system with an extra Lipschitz nonlinear term. Consider the system shown in (2.16) with $F_y = 0$. Then, the corresponding SMO takes on the form of:

$$\dot{x} = Ax + Bu + L(y - Cx) + F_x \nu$$

where $L \in \mathbb{R}^{n \times p}$ is the linear observer gain and $\nu \in \mathbb{R}^k$ is known as the sliding term. Different approaches can be taken to choose $L$ and $\nu$. The sliding term often takes on the following form:

$$\nu = -\rho \frac{e}{\|e\|}$$

where $e \in \mathbb{R}^k$ is usually some linear function of $y - C\dot{x}$ and $\rho$ is some gain. The sliding term shown is very similar to $sgn(.)$ function. The gain $\rho$ can be chosen to be constant (for example, [25] and [69]) or it can be chosen to be an adaptive gain which can depend on $y$ or $u$ (for example, [73]).

For a SMO capable of driving the state estimation error to some sliding surface while estimating fault to exist, several assumptions are required. They are:

(i) The matrices, $C$ and $E$ satisfy: $\text{rank}(CF_x) = \text{rank}(F_x)$.

(ii) All of the invariant zeros of the triplet $(C, A, F_x)$ lie on the open left-half complex plane.

The invariant zeros (also known as the transmission zeros) of the triplet $(C, A, F_x)$ refer to set of complex number, $s$ which satisfies the following inequality [19]:

$$\text{rank} \begin{bmatrix} sI - A & F_x \\ C & 0 \end{bmatrix} < n + \min(k, p)$$
To estimate the faults using the SMO, one needs to show that the SMO will reach its ideal chosen sliding surface and stay on it even if faults occur. When a SMO reaches its ideal sliding surface, one can use the concept of equivalent control and take the estimated fault, $\hat{f} \in \mathbb{R}^k$ to be:

$$\hat{f} = \nu_{eq}$$

where $\nu_{eq}$ is the equivalent output error injection of the SMO. Consider the case when sliding motion is achieved. Roughly speaking, the equivalent output error injection is the continuous output error signal, $y - C\dot{x}$ required to keep the state trajectory on the sliding surface assuming $\nu = 0$. Note that $\nu$ is a discontinuous signal that is used for actual state estimation, whereas $\nu_{eq}$ is a continuous signal used for analyzing the system after sliding motion is achieved. The paper [25] proposes to take the fault estimate, $\hat{f}$ to be:

$$\hat{f} = -\rho \frac{e}{\|e\| + \delta} \approx \nu_{eq}$$

where $\delta$ is a small positive constant.

### 2.3.5 Integrated Fault Tolerant Control

One of the contributions in this thesis is the development of two new integrated fault tolerant control schemes. In this subsection, a literature review of some current IFTC schemes is presented.

Traditionally, the FDI and CR/FA modules have been designed separately. The approach ignores the interaction between the FDI and CR/FA modules. Often fault diagnosis is performed under uncertainty and in real-time. Control action may need to be taken before fault information has been accurately determined. An incorrect diagnosis can thus lead to a poor performance of the CR/FA module. Integrated fault tolerant control (IFTC) schemes, where the control design takes both FDI and CR/FA into consideration, address this issue. However, their design and analysis is more difficult as a result of the interaction between diagnosis and control.

An IFTC scheme is proposed in [47] where the authors’ proposed control law uses the fault estimate from a nonlinear adaptive observer to cancel the fault. The system that is considered by the authors is similar to (2.16), with an additional Lipschitz nonlinear term, $g(t, x)$. The matrices $E_x$, $E_y$, $F_y$, and $D$ are taken to be 0. The fault estimation method proposed uses an observer-based approach. The fault, $f$ is assumed to be differentiable with respect to time and
its norm is bounded. The observer for both the states and the fault is given as:

$$
\begin{bmatrix}
\dot{\hat{x}} \\
\dot{\hat{f}}
\end{bmatrix} =
\begin{bmatrix}
A - KC & E \\
-GRTC & -\sigma \Gamma
\end{bmatrix}
\begin{bmatrix}
\hat{x} \\
\hat{f}
\end{bmatrix} +
\begin{bmatrix}
B & K \\
0 & \Gamma R
\end{bmatrix}
\begin{bmatrix}
u \\
y
\end{bmatrix} +
\begin{bmatrix}
g(t, \hat{x})
\end{bmatrix}
$$

(2.21)

where $\hat{x}$ is the state estimate, $\hat{f}$ is the fault estimate, $\Gamma = \Gamma^T > 0$ is a weighting matrix, $\sigma > 0$ is a constant and $R$ is a matrix that is required to be computed. Both $\Gamma$ and $\sigma$ are design parameters.

With knowledge of the fault estimate, the authors of [47] incorporate it into a FTC scheme which guarantees the stability of the faulty system. The fault tolerant control law requires two technical assumptions to be satisfied, which will not be covered in this description. The control law is given as:

$$
u = -B^T H \hat{x} - B^* E \hat{f} - \frac{\eta^2(\hat{x}, t)}{||\eta(\hat{x}, t)B^T H \hat{x}|| + \epsilon/2} B^T H \hat{x}
$$

(2.22)

where $B^*$ and $\eta(\hat{x}, t)$ arise from the two technical assumptions.

Another similar strategy is used in [46]. The paper considered a linear discrete-time stochastic system with actuator faults. The authors presented a simple IFTC scheme. Let $x(k) \in \mathbb{R}^n$ be the system states, $u(k) \in \mathbb{R}^m$ be the control input, $f(k) \in \mathbb{R}^q$ be the fault, $\omega(k) \in \mathbb{R}^n$ be the system noise, $y(k) \in \mathbb{R}^r$ be the output and $v(k)$ be the output noise. The system is assumed to have a special structure as shown:

$$
\begin{bmatrix}
x_1(k+1) \\
x_2(k+1) \\
x_3(k+1)
\end{bmatrix} =
\begin{bmatrix}
A_1 \\
A_2 \\
A_3
\end{bmatrix}
\begin{bmatrix}
x(k) \\
u(k) \\
f(k)
\end{bmatrix} +
\begin{bmatrix}
E_1 \\
E_2 \\
E_3
\end{bmatrix}
\begin{bmatrix}
\omega_1(k) \\
\omega_2(k) \\
\omega_3(k)
\end{bmatrix}
$$

$$
y(k) =
\begin{bmatrix}
0 & 0 & I_r \\
0 & I_q & 0
\end{bmatrix}
\begin{bmatrix}
x(k) \\
u_1(k) \\
u_2(k)
\end{bmatrix}
$$

where $x_1, \omega_1$ are of dimension $n - r$, $x_2, \omega_2, y_1$ are of dimension $r - q$, and $x_3, \omega_3, y_2$ are of dimension $q$. The system noise and output noise are assumed to be Gaussian, and hence a Kalman Filter is used to estimate the states. If $E_3 \in \mathbb{R}^{q \times q}$ is invertible, then the fault can be easily estimated as follows:

$$
\hat{f}(k - 1) = (E_3)^{-1}[y_2(k) - A_3 \hat{x}(k - 1) - B_3 u(k - 1)]
$$

(2.23)

where $\hat{x}(k - 1)$ is the state estimate from the Kalman filter.

For the fault tolerant controller, [46] only considered the case of actuators losing their
effectiveness. The fault in this case can be modeled as \( E = -B \) and \( f(k) = R(k)u(k) \), where \( R(k) \) is a diagonal matrix with appropriate dimension. The fault tolerant controller for this fault is given as:

\[
u(k) = (I - \hat{R}(k))^{-1}u_n(k)
\]

where \( u_n(k) \) is the control law for the nominal/healthy case (meaning no faults are present), and \( \hat{R}(k) \) is determined from the fault estimator (2.23).

Reference [82] uses artificial neural networks to estimate the faults and employs an adaptive control law to perform fault accommodation. It requires the designer to model all the possible faults and find a set of basis functions for each fault. The process of modeling every possible fault will be complex.

In [83], the authors consider a linear discrete time system. The authors proposed to use the interacting multiple model algorithm, an extension of the multiple model technique, to diagnose the fault that had occurred in a probabilistic fashion. The basic idea is to model the system with all possible faults and design a Kalman filter for each faulty system and the nominal system. By comparing the output from the system and the estimated output from the Kalman filter, fault occurrence can be diagnosed. The diagnosis procedure will not be described here, see [83] for the details.

The FTC scheme proposed by [83] is based on the so-called eigenstructure assignment. Eigenstructure refers to both the eigenvalues and eigenvectors. Eigenstructure assignment refers to the process of assigning the poles of the system as well as eigenvectors of the resulting closed loop system. It is achieved by first placing the new poles of the faulty system to be those of the healthy system. Then, the eigenvectors of the healthy system are projected onto the subspace spanned by the eigenvectors of the uncontrolled faulty system. This gives rise to the eigenvectors of the controlled faulty system. The eigenvalues and eigenvectors of the faulty system are assigned using a feedback matrix. Lastly, a feedforward gain is designed to ensure a satisfactory steady-state performance. Other IFTC-related work includes [23], [60], and [79].

### 2.4 Fault Tolerant Control of LFSS

We begin this section by presenting a short literature review of FTC of a LFSS. In [80], the author integrates various learning techniques to provide fault diagnosis and fault tolerant control for LFSS. The overall control system includes radial basis function networks, a bidirectional associative memory and cerebellar model articulation controller network. Ahmad et al. [3] used interval transfer functions to model LFSSs with predicted faults and employed \( \mu \)-synthesis to develop robust controllers for each of the models with faults. A more detailed discussion on
this technique can be found in [2]. The conditions for stability of a colocated lump-modeled LFSS under failure were studied in [51].

References [2], [3] and [80] do not provide rigorous mathematical study of the stability and performance of their proposed FTC scheme. Huang [38] employs two modified decentralized control schemes which originated in [78] to perform simple yet powerful FTC. Closed loop stability under these fault tolerant controllers is proven. Due to the simplicity and provable properties of these controllers, we prefer to use these controllers to perform FTC on the LFSS if possible. Huang has also proposed a fault diagnosis algorithm to be used in conjunction with his proposed fault tolerant controllers. However, it generally needs a long time to produce a diagnosis. To understand more about Huang’s fault tolerant controllers and fault diagnosis scheme, a general overview of the controllers and fault diagnosis is presented in the rest of this section.

2.4.1 Huang’s Fault Tolerant Controllers

First, we present the original control objective without the consideration of failures. Consider the model given in (2.5) and (2.6). Assume that no failures will occur at the actuators and sensors. The goal is to stabilize the LFSS and that each measured output of the LFSS asymptotically tracks a constant reference signal. This problem has been fully studied in [78]. The authors proposed a decentralized proportional integral derivative (PID) control law which is capable of achieving the following three objectives:

1. The closed loop system is asymptotically stable, i.e. all eigenvalues of the closed loop system are in the open left half plane.

2. For all initial conditions, the measured outputs, \( y_m(t) \) asymptotically track a given constant reference signal, \( y_{ref}(t) \) while rejecting all constant disturbances.

3. Objectives 1 and 2 hold for all perturbations in the triplet \( (\Omega^2, \Delta, L) \) that do not cause instability in the resulting controlled system.

The term “decentralized” means that the control law governing the \( i \)-th actuator depends solely on the reading from the \( i \)-th sensor. Let \( K_P, K_D \) and \( K_I \) be real, positive definite, block diagonal matrices (with dimensions of \( m \times m \)) that represent the proportional, derivative and integral gain for the controller. The block diagonal structures of \( K_P, K_D \) and \( K_I \) are the result of enforcing the control law to be decentralized. Define \( \eta \in \mathbb{R}^m \) to be the integrator state. The decentralized PID controller is given as:

\[
\dot{\eta} = e_y
\]
\[ u = -K_P e_y - K_D \dot{e}_y - \epsilon K_I \eta \] (2.25)

where \( e_y = y_m - y_{ref} \), \( y_m \) is the output from the sensor of the LFSS and \( y_{ref} \) is the reference signal that we would like the output to track. The term \( \epsilon > 0 \) is a number that ultimately limits the amount of influence that the integrator can have in the control signal.

The tuning parameter \( \epsilon \) should be chosen small to guarantee the aforementioned three objectives can be achieved by the decentralized PID controller. The original system considered by the authors of [78] is slightly more complicated because they allow certain locations to have the freedom of not having to track any reference signal. We will assume that every location that can be measured will have to track a reference signal.

The decentralized PID controller cannot always be designed to achieve objectives 1 to 3. The necessary and sufficient condition for the existence of the PID controller is \( \text{rank}(\hat{\mathcal{L}}) = \hat{n} \), where \( \hat{\mathcal{L}} \) is defined in (2.7). For more information, refer to Section 2.3 of [38] or the paper [78].

The decentralized PID controller shown in (2.24) and (2.25) serves as the skeleton for the two fault tolerant controllers later described in this section. To construct the fault tolerant controllers, one will need to obtain the combined failure matrix, \( F \) (more information regarding \( F \) can be found in Section 2.2). \( F \) can be obtained through the FDI module. Control reconfiguration is used in this scheme to ensure normal operation of the faulty system. The type of control reconfiguration used is a simple strategy of not employing any faulty sensor-actuator pairs to control the LFSS via the decentralized PID controller shown in (2.24) and (2.25). Therefore, the first fault tolerant controller is a straightforward extension of the decentralized PID controller, which is given as follows:

\[ \dot{\eta} = F e_y \] (2.26)

\[ u = -F(K_P F e_y - K_D F \dot{e}_y - \epsilon K_I F \eta) \] (2.27)

where \( F \) is given in (2.8). We call this controller the fault tolerant decentralized PID (FTD-PID) controller.

It turns out that the FTD-PID controller can achieve similar objectives as the nominal decentralized PID controller under certain conditions. The three objectives that can be achieved by the FTD-PID controller are:

(A1) The reduced-order closed loop system is stable, i.e. all eigenvalues of the closed loop system are in the closed left half plane. The term “reduced-order” refers to the system which one discards all the faulty sensor-actuator pairs. Thus, the system is only left with healthy sensor-actuators. Refer to Section 3.3.3 of [38].
(A2) For all initial conditions, the measured outputs of the healthy sensor-actuator pairs \( F_{y_m}(t) \) asymptotically track a given constant reference signal \( F_{y_{\text{ref}}}(t) \) while rejecting all constant disturbances.

(A3) Objectives (A1) and (A2) hold for all perturbations in the triplet \((\Omega^2, \Delta, L)\) that do not cause instability in the resulting controlled system.

The condition at which the FTD-PID controller can achieve objectives (A1)-(A3) is given by Theorem 2.2 of [38], stated below.

Theorem 2.1. The FTD-PID controller, given by (2.26) and (2.27), can achieve objectives (A1)-(A3) if and only if \( \text{rank}(\hat{L}F) = \hat{n} \), where \( \hat{n} \) is the number of rigid body modes.

This theorem shows that the FTD-PID controller can only control the LFSS successfully when the failures do not reduce the number of rigid body modes of the system that is available for control. Further information related to the FTD-PID controller can be found in Chapter 3 of [38].

The FTD-PID controller is a good control law to handle failures when the failures are known. However, if the failures are not known, then there is no way to apply the FTD-PID controller. This motivated Huang to propose a passive fault tolerant controller, namely a feedforward decentralized proportional derivative (PD) controller. The PD controller is a PFTC technique. It uses its inherent robustness against changes in the system to stabilize the system while maintaining some degree of tracking. A feedforward term is added to the control law to improve the tracking performance of the output.

Before introducing the feedforward decentralized PD controller, it is required to first introduce a special matrix \( \Gamma \) that contains information related to the steady state value of \( y_m \). The matrix \( \Gamma \) is given by (2.28).

\[
\Gamma = (L^T(\Omega^2 + LL^T)^{-1}L)^{-1} - I_m \quad (2.28)
\]

With the knowledge of \( \Gamma \), one can compute the feedforward decentralized PD control law as follows:

\[
u = -\alpha(K_pe_y + K_D\dot{e}_y) + \Gamma y_{\text{ref}} \quad (2.29)
\]

where \( \alpha > 0 \) is a design parameter. Without the integrator state in the controller, it is possible for the controller to stabilize the LFSS even when a failure has occurred. However, the downside is one cannot achieve perfect tracking. The feedforward term, \( \Gamma y_{\text{ref}} \) is introduced to improve tracking performance of the decentralized PD controller.

Under certain conditions, the feedforward decentralized PD controller can control the LFSS such that the following goals are satisfied:
(B1) For all $\alpha > 0$, the poles of the closed loop system are in the open left half plane.

(B2) For all initial conditions, as $\alpha \to \infty$, the measured outputs of the healthy sensor-actuator pairs $Fy_m(t)$ asymptotically track a given constant reference signal $Fy_{ref}(t)$ while rejecting all constant disturbances.

(B3) Objectives (B1) and (B2) hold for all perturbations in the triplet $(\Omega^2, \Delta, L)$ that do not cause instability in the resulting controlled system.

The condition for which the feedforward decentralized PD controller is guaranteed to achieve goals (B1) to (B3) is similar to the FTD-PID controller. It is shown in the following theorem, which summarizes the Theorem 4.15 and 4.16 of [38].

**Theorem 2.2.** Assume $F = F_u$. For all $\alpha > 0$, the feedforward decentralized PD controller can achieve property (B1)-(B3) if and only if $\text{rank}(\hat{L}F) = \hat{n}$.

For the stability of the closed loop system using the PD controller, a necessary and sufficient condition for the closed loop system to be stabilizable is simply $\text{rank}(\hat{L}F) = \hat{n}$. The assumption $F = F_u$ is only used to guarantee satisfactory tracking.

### 2.4.2 Huang’s Fault Diagnosis Algorithm

Not only can the decentralized PD controller stabilize a faulty system, it can also be used to perform fault diagnosis (fault isolation) via perturbation of reference signals. We will summarize some of the results related to Huang’s fault diagnosis algorithm. Let $\bar{y}$ be the steady state output value of $y(t)$ and $S_{ref} \in \mathbb{R}^{m \times m}$ be the steady-state output reference matrix. The matrix $S_{ref}$ can be calculated as follows:

$$ S_{ref} = (\Gamma + F_u K_P F_y)^{-1} F_u K_P $$

(2.30)

The main result that is used for fault diagnosis is stated in Corollary 5.4 of [38]. We will state it as a theorem.

Define $\bar{S}_{ref} = F_y S_{ref}$ and its corresponding $i$-th diagonal element be $\bar{S}_{ref}^{ii}$. Also, let $S_{ref(nom)}$ be $S_{ref}$ with $F_u = F_y = I_m$, i.e. the nominal steady-state output reference matrix with no failed actuators or sensors. We define the $i$-th element of $S_{ref(nom)}$ to be $S_{ref(nom)}^{ii}$.

**Theorem 2.3.** Assume that the faulty closed loop system using the decentralized PD controller is stable. Let $i \in \{1, ..., m\}$. Then, if the $i$-th sensor-actuator pair is healthy (i.e. $F^i = 1$), $S_{ref}^{ii} \geq S_{ref(nom)}^{ii}$. 
When a failure occurs, the decentralized PD controller with no feedforward compensation is used to control the LFSS. Using the Theorem 2.3, Huang proposed the following algorithm to diagnose failed locations in a LFSS.

1. Let the initial constant reference signal be \( \sigma \). Acquire the steady-state output measurement \( y_{m0} = F_y \bar{y}_0 \), where \( \bar{y}_0 \) is the actual steady state output using the initial constant reference signal.

2. Set \( i = 1 \).

3. Choose a small \( \rho_i > 0 \) reasonably small to \( \sigma \). Also choose a small perturbation reference basis vector, \( \tilde{\sigma}_i \), which has to be linearly independent of other chosen \( \tilde{\sigma}_j \) (\( j \neq i \)). Typically, one would just choose \( \tilde{\sigma}_i = I_i \), where \( I_i \) is the \( i \)-th standard basis vector.

4. Set \( y_{ref}(t) = \sigma + \rho_i \tilde{\sigma}_i \). Denote the resulting steady state output \( y_{mi} = F_y \bar{y}_i \), where \( \bar{y}_i \) is the actual steady state output using the new constant reference signal.

5. Wait for steady state to occur and record \( y_{mi} \). Alternatively, one can use some extrapolation filter to obtain \( y_{mi} \).

6. Calculate the vector \( v_i = \tilde{S}_{ref} \tilde{\sigma}_i = 1/\rho_i (y_{mi} - y_{m0}) \).

7. Increment \( i \). Stop if \( i > m \), else repeat step 3 to step 6.

8. Construct the matrices \( V = \begin{bmatrix} v_1 & \ldots & v_m \end{bmatrix} \) and \( \Sigma = \begin{bmatrix} \tilde{\sigma}_1 & \ldots & \tilde{\sigma}_m \end{bmatrix} \). Compute \( \tilde{S}_{ref} = V \Sigma^{-1} \).

9. Calculate \( S_{ref(nom)} \) using (2.30) and setting \( F_u = F_y = I_m \).

10. Apply Theorem 2.3 to check which locations have failed.

The main drawback of this fault diagnosis algorithm is the requirement to change the output and wait for steady state information to determine the failed locations. Since LFSS are inherently very slow system, this will result in a long time to produce a diagnosis.
Chapter 3

Fault Accommodation using Sliding Mode Observer (SMO)

The main contribution in this chapter is the introduction of a new integrated fault tolerant control (IFTC) scheme using the $\mathcal{H}_\infty$ sliding mode observer (SMO) proposed in [69] as a fault estimator to perform fault accommodation (FA) similar to the approach in [47]. SMOs have the capability to perform fault estimation as well as state estimation, making them an appealing fault detection and isolation (FDI) component for possible application to large flexible space structures (LFSSs).

The $\mathcal{H}_\infty$ SMO proposed in [69] is capable of producing robust state and fault estimates. First, the $\mathcal{H}_\infty$ SMO is introduced and studied. Then, we investigate the applicability of the $\mathcal{H}_\infty$ SMO to the colocated LFSS. However, it was found that the $\mathcal{H}_\infty$ SMO cannot be used to estimate both actuator and sensor faults on the LFSS. Nonetheless, the attractive robust state and fault estimation capabilities of the SMO motivates us to develop a new IFTC scheme using the $\mathcal{H}_\infty$ SMO for a class of nonlinear system. Therefore, we will not apply the $\mathcal{H}_\infty$ SMO to the LFSS. Instead, we will merge the $\mathcal{H}_\infty$ SMO with a fault accommodation controller to perform fault tolerant control for a class of nonlinear systems.

Our IFTC scheme enjoys enhanced robustness to disturbances due to the ability of the $\mathcal{H}_\infty$ SMO to attenuate their effects. The enhanced robustness against disturbance is not present in the scheme presented in [47]. While [47] uses a nonlinear adaptive observer followed by a nonlinear controller, we combine the $\mathcal{H}_\infty$ SMO with a modified version of the linear controller introduced in [64]. The controller proposed in [64] does not consider the occurrence of faults. Furthermore, our proposed controller does not require high gain switching control to stabilize the closed loop system, which makes it more practical for implementation. With bounded system noise, it is shown that the states and the state estimation error are bounded under our IFTC scheme. We illustrate the effectiveness of the proposed IFTC scheme to handle system
noise in the control of a permanent magnet synchronous motor under actuator failure.

### 3.1 System Description

Consider the following nonlinear continuous-time system:

\[
\dot{x} = Ax + Bu + \Phi(x) + Ef + \Delta \xi \\
y = Cx
\]  

(3.1)  

(3.2)

where \( x \in \mathbb{R}^n \) is the state, \( u \in \mathbb{R}^m \) is the control input, \( y \in \mathbb{R}^p \) is the output, and \( \xi \in \mathbb{R}^r \) is the system disturbance. \( A, B, C, E, \Delta \) are real constant matrices with the appropriate dimensions. The nonlinearity \( \Phi(x) \) is assumed to be a continuous, globally Lipschitz nonlinear function, with a Lipschitz constant \( L \Phi \), i.e., \( \| \Phi(x) - \Phi(x_0) \| \leq L \Phi \| x - x_0 \| \). The operator \( \| \cdot \| \) represents the Euclidean norm. The states are not assumed to be measured so that only output feedback controllers are considered.

The vector \( f \in \mathbb{R}^q \) represents the faults that occur in the system. The faults are assumed to lie in the actuation space. Both system and measurement noise are considered in the development of the \( H_\infty \) SMO in [69]. To simplify the analysis of our IFTC scheme, only system disturbances are considered in this work.

### 3.2 \( H_\infty \) SMO

We begin to introduce our IFTC scheme by first revisiting the \( H_\infty \) SMO. Two crucial results of the original paper [69] are represented in this section and the proofs are shown as well. This results are important in developing the proof in Section 3.4 to show that the overall faulty closed loop system can be stabilized using our proposed approach.

#### 3.2.1 Assumptions and their Implications

For the design of the \( H_\infty \) SMO, the following set of assumptions is made on the system (3.1) and (3.2).

Assumption 3.1.

(a) The fault and system disturbance, \( f \) and \( \xi \) satisfy \( \| f(t) \| \leq \rho \) and \( \| \xi(t) \| \leq \xi_0 \), for all \( t \), for some \( \rho \geq 0, \xi_0 \geq 0 \).
(b) The so-called matching condition holds, i.e. there exist a matrix $F \in \mathbb{R}^{q \times p}$ and a symmetric positive definite matrix $P \in \mathbb{R}^{n \times n}$ such that:

$$E^T P = FC$$

This is shown in [69] to be equivalent to:

$$\text{rank}(CE) = \text{rank}(E)$$

Further, assume $\text{rank}(E) = q$.

(c) The minimum phase condition holds: For every complex number $s \in \{s : \text{Re}(s) \geq 0\}$,

$$\text{rank} \begin{bmatrix} sI_n - A & E \\ C & 0 \end{bmatrix} = n + \text{rank}(E)$$

The fact that $\xi$ is bounded implies that $\xi$ is square integrable from time 0 to $t_f$ (i.e. $\xi \in \mathcal{L}_2[0, t_f]$), for each $t_f > 0$. Assumption 3.1(c) guarantees that stable sliding motion can be achieved. Assumption 3.1(b) is used to ensure that when sliding motion is achieved, the occurrence of fault will not destroy it. Also, in [69], it is shown that Assumption 3.1(b) is equivalent to the existence of nonsingular transformation matrices $T \in \mathbb{R}^{n \times n}$ and $S \in \mathbb{R}^{p \times p}$ such that:

$$E_n = TE = \begin{bmatrix} E_{n1} \\ 0 \end{bmatrix}$$

$$C_n = SCT^{-1} = \begin{bmatrix} C_{n1} & 0 \\ 0 & C_{n4} \end{bmatrix}$$

where $E_{n1} \in \mathbb{R}^{q \times q}$, $C_{n1} \in \mathbb{R}^{q \times q}$ and $\text{rank}(E_{n1}) = q$. The transformations $T$ and $S$ are used to transform the state and output to a new coordinate system. For clarity of presentation, we use the subscript ‘n’ to denote variables and matrices in this new coordinate system. $T$ and $S$ are important matrices used in constructing the SMO. However, the computation is not straightforward. We present the algorithm to compute them in the next subsection.

### 3.2.2 Computation of $T$ and $S$

The algorithm provided in this section is taken directly from the original paper [69]. Assume that Assumption 3.1(b) holds. The steps for computing $T$ and $S$ are given as follows:

1. Find a permutation matrix, $Z_1 \in \mathbb{R}^{n \times n}$ to transform the matrix $E$ into $\hat{E} = Z_1 E$ such
that:

\[ \tilde{E} = \begin{bmatrix} \tilde{E}_1 \\ \tilde{E}_2 \end{bmatrix} \]

where \( \tilde{E}_1 \in \mathbb{R}^{q \times q} \) and \( \text{rank}(\tilde{E}_1) = q \). This is always possible when \( \text{rank}(E) = q \).

2. Construct \( \tilde{T}_1 \) as follows:

\[ \tilde{T}_1 = \begin{bmatrix} I_q & 0 \\ -\tilde{E}_2 \tilde{E}_1^{-1} & I_{n-q} \end{bmatrix} \]

where \( I_q \) is the identity matrix of dimension \( q \times q \).

3. Calculate \( \tilde{C} = CZ_1^{-1}T_1^{-1} = [\tilde{C}_1 \ \tilde{C}_4] \), where \( \tilde{C}_1 \in \mathbb{R}^{p \times q} \).

4. Find another permutation matrix, \( Z_2 \in \mathbb{R}^{p \times p} \) to transform the matrix \( \tilde{C}_1 \) into \( \tilde{C}_1 = Z_2\tilde{C}_1 \) such that:

\[ \tilde{C}_1 = \begin{bmatrix} \tilde{C}_{11} \\ \tilde{C}_{12} \end{bmatrix} \]

where \( \tilde{C}_{11} \in \mathbb{R}^{q \times q} \) and \( \text{rank}(\tilde{C}_{11}) = q \). Again, this is always possible due to \( \text{rank}(CE) = \text{rank}(E) = q \). Then, use \( Z_2 \) to transform \( \tilde{C} \) into \( \tilde{C} = Z_2\tilde{C} \), which results in:

\[ \tilde{C} = \begin{bmatrix} \tilde{C}_{11} & \tilde{C}_{41} \\ \tilde{C}_{12} & \tilde{C}_{42} \end{bmatrix} \]

where \( \tilde{C}_{42} \in \mathbb{R}^{(p-q) \times (p-q)} \).

5. Construct \( \tilde{S} \) as given:

\[ \tilde{S} = \begin{bmatrix} I_q & 0 \\ -\tilde{C}_{12} \tilde{C}_{11}^{-1} & I_{p-q} \end{bmatrix} \]

6. Construct \( \tilde{T}_2 \):

\[ \tilde{T}_2 = \begin{bmatrix} I_q & \tilde{C}_{11}^{-1} \tilde{C}_{41} \\ 0 & I_{n-q} \end{bmatrix} \]

Note that:

\[ \tilde{T}_2^{-1} = \begin{bmatrix} I_q & -\tilde{C}_{11}^{-1} \tilde{C}_{41} \\ 0 & I_{n-q} \end{bmatrix} \]

7. Finally, construct \( T = \tilde{T}_2\tilde{T}_1Z_1 \) and \( S = \tilde{SZ}_2 \).

The permutation matrices \( Z_1 \) and \( Z_2 \) can be easily created using numerical software such as MATLAB. The main idea is to rearrange the rows of matrices \( E \) and \( \tilde{C}_1 \) to satisfy the rank condition. The permutation matrices come as a by-product of the elementary row operations.
Using the resulting $T$ and $S$, we would like to verify that $E_n = TE$ and $C_n = SCT^{-1}$:

$$E_n = TE = \tilde{T}_2 \tilde{T}_1 Z_1 E = \tilde{T}_2 \tilde{T}_1 \tilde{E}$$

$$= \tilde{T}_2 \begin{bmatrix} I_q & 0 \\ -\tilde{E}_2 \tilde{E}_1^{-1} & I_{n_q} \end{bmatrix} \begin{bmatrix} \tilde{E}_1 \\ \tilde{E}_2 \end{bmatrix}$$

$$= \begin{bmatrix} I_q & \tilde{C}_{11}^{-1} \tilde{C}_{41} \\ 0 & I_{n_q} \end{bmatrix} \begin{bmatrix} \tilde{E}_1 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} \tilde{E}_1 \\ 0 \end{bmatrix}$$

$$C_n = SCT^{-1} = \tilde{S} Z_2 C Z_1^{-1} \tilde{T}_1^{-1} \tilde{T}_2^{-1} = \tilde{S} Z_2 \tilde{C} T_2^{-1} = \tilde{S} \tilde{C} T_2^{-1}$$

$$= \begin{bmatrix} I_q & 0 \\ -\tilde{C}_{12} \tilde{C}_{11}^{-1} & I_{p_q} \end{bmatrix} \begin{bmatrix} \tilde{C}_{11} & \tilde{C}_{41} \\ \tilde{C}_{12} & \tilde{C}_{42} \end{bmatrix} \begin{bmatrix} I_q & -\tilde{C}_{11}^{-1} \tilde{C}_{41} \\ 0 & I_{n_q} \end{bmatrix}$$

$$= \begin{bmatrix} \tilde{C}_{11} & 0 \\ 0 & -\tilde{C}_{12} \tilde{C}_{11}^{-1} \tilde{C}_{41} + \tilde{C}_{42} \end{bmatrix}$$

Note that $E_{n1} = E_1$, $C_{n1} = \tilde{C}_{11}$ and $C_{n4} = -\tilde{C}_{12} \tilde{C}_{11}^{-1} \tilde{C}_{41} + \tilde{C}_{42}$. Hence, the constructed $T$ and $S$ can be used to obtain the structures shown in (3.3) and (3.4).

### 3.2.3 Overview of the $\mathcal{H}_\infty$ SMO

The $\mathcal{H}_\infty$ sliding mode observer is capable of providing state and fault estimates that are robust to disturbances, which makes it attractive as part of an IFTC scheme. Hence, we propose to use the $\mathcal{H}_\infty$ SMO as the FDI module in our IFTC scheme. For brevity, the $\mathcal{H}_\infty$ SMO is presented using a mix of original and new coordinates. We first recall the construction of the $\mathcal{H}_\infty$ SMO from [69].

Let

$$[x_{n1}^T, x_{n2}^T]^T = Tx = T[x_1^T, x_2^T]^T$$

$$P_n = \text{diag}(P_{n1}, P_{n2}) = (T^T)^{-1} PT^{-1}$$

where $x_{n1}, x_1 \in \mathbb{R}^q$ and $P_{n1} \in \mathbb{R}^{q \times q}$. Let $\hat{x} \in \mathbb{R}^n$ be the state estimate, $L \in \mathbb{R}^{n \times p}$ be the observer gain matrix, and $\rho_0 > 0 \in \mathbb{R}$, $\delta \geq 0 \in \mathbb{R}$ be design parameters to be introduced below. The proposed $\mathcal{H}_\infty$ SMO in the original coordinates is given by:

$$\dot{\hat{x}} = A\hat{x} + Bu + \Phi(\hat{x}) + L(y - C\hat{x}) + E\nu$$  (3.5)
where
\[
\nu = (\rho + \rho_0) \frac{E_{n1}^TP_{n1}(x_{n1} - \hat{x}_{n1})}{\|E_{n1}^TP_{n1}(x_{n1} - \hat{x}_{n1})\| + \delta/2}
\] (3.6)

The signal \( \nu \) will be a more complicated looking function when expressed in the original coordinates. The term \( x_{n1} \) from (3.6) is a linear function of the output, \( y \). Partition \( S \) such that:
\[
S = \begin{bmatrix} S_1 \\ S_2 \end{bmatrix}
\]
where \( S_1 \in \mathbb{R}^{q \times p} \) and \( S_2 \in \mathbb{R}^{(p-q) \times p} \). The term \( x_{n1} \) can be calculated as:
\[
x_{n1} = C_{n1}^{-1}S_1y
\]

The observer gain in the new coordinates, \( L_n \), takes on a special form:
\[
L_n = TLS^{-1} = \begin{bmatrix} (A_{n1} - A_{n1}^s)C_{n1}^{-1} & 0 \\ A_{n3}C_{n1}^{-1} & P_{n2}^{-1}K_n \end{bmatrix}
\] (3.7)

where \( K_n \) is a gain matrix to be determined, \( A_{n1}^s \) is a stable design matrix (i.e. all of its eigenvalues are in the open left half plane) and \( A_{n1}, A_{n3} \) are block matrices determined from the matrix \( A_n \) in the new coordinate system:
\[
A_n = TAT^{-1} = \begin{bmatrix} A_{n1} & A_{n2} \\ A_{n3} & A_{n4} \end{bmatrix}
\]

A matrix, \( H_n = diag(H_{n1}, H_{n2}) \), \( H_{n1} \in \mathbb{R}^{q \times q} \) is to be selected to give weighting to the error vector, \( e_n = x_n - \hat{x}_n \) in the transformed coordinates. Denote this matrix in the original coordinates to be \( H = H_nT \).

Denote the \( L_2 \) norm of a signal, \( \omega(t) \) to be \( \|\omega\|_{L_2} = \sqrt{\int_0^t \omega^T(t)\omega(t)dt} \). Fix \( t_f > 0 \) (which can be viewed as the expected duration the system runs), the \( \mathcal{H}_\infty \) gain from the disturbance \( \xi \) to the weighted error \( He \) is defined similar to [69] as:
\[
\Upsilon^2 = \sup_{\|\xi\|_{L_2} \neq 0} \frac{\|He\|_{L_2}^2}{\|\xi\|_{L_2}^2}
\]
where \( e = x - \hat{x} \) is the state estimation error. Note that the \( \mathcal{H}_\infty \) gain is always calculated assuming \( e(0) = 0 \). One can choose the weighting matrix \( H \) or \( H_n \) in similar fashion to the state weighting matrix in the linear-quadratic regulator (LQR) design. If \( \Upsilon < 1 \), the effect of \( \xi \) on the state estimation error is attenuated [45]. In such a case, one can refer to the \( \mathcal{H}_\infty \) gain as
\( \mathcal{H}_\infty \) attenuation. One of the goals in designing the \( \mathcal{H}_\infty \) SMO is to minimize the value of \( \Upsilon \).

### 3.2.4 Existence of \( \mathcal{H}_\infty \) SMO and Its Noise Attenuation Property

Once the matrix \( H_n \) is selected, one can proceed with the design of the SMO given by (3.5)-(3.6). Let \( Q_m = -[(A - LC)^T P + P(A - LC) + L_\delta^2 I + H^T H] \), where \( P^T = P > 0 \) and \( L \) are matrices to be determined by solving a linear matrix inequality (LMI). For more information on LMI, we refer the reader to the book [13], which provides an extensive coverage on LMIs.

Also, for \( \gamma > 0 \), define:

\[
\Omega = \begin{bmatrix}
Q_m - P^2 & -P \Delta \\
-\Delta^T P & \gamma I
\end{bmatrix}
\]

\( a = \lambda_{\min}(\Omega)/\lambda_{\max}(P) \), \( b = \rho\delta + \gamma \xi_0^2 \)

and the set:

\[
\Lambda = \{ e : \|e\|^2 \leq b/(a\lambda_{\min}(P)) \}
\]

where \( \lambda_{\min}(X) \) (or \( \lambda_{\max}(X) \)) denotes the minimum (or maximum) eigenvalue of a matrix \( X \). Let:

\[
\psi(t_f) = \frac{\rho\delta t_f}{\int_0^{t_f} \xi^T(t)\xi(t)dt}.
\]

Note that since \( \xi \) is square integrable for a given \( t_f \), \( \psi(t_f) \) must be a finite constant provided that \( \xi(t) \) is not zero for all time. We can now state Theorem 3.1, which establishes the existence of a stable SMO.

**Theorem 3.1.** Consider the system (3.1) and (3.2). Let Assumption 3.1 hold. Suppose there exist matrices \( K_n \) (recall that \( K_n \) is hidden within \( L \)) and \( P \) such that the following LMI optimization problem has a solution: Minimize \( \gamma \) subject to \( 0 \leq \gamma \leq 1 \), \( P > 0 \) and

\[
\begin{bmatrix}
-Q_m & P \Delta & P \\
\Delta^T P & -\gamma I & 0 \\
P & 0 & -I
\end{bmatrix} < 0.
\]

Then the \( \mathcal{H}_\infty \) SMO (3.5)-(3.6) has a guaranteed \( \mathcal{H}_\infty \) attenuation \( \sqrt{\gamma + \psi(t_f)} \) (i.e. \( \|\mathcal{H}_\infty\| \leq \sqrt{\gamma + \psi(t_f)} \)) and the state estimation error \( e \) converges to the set \( \Lambda \).

For convenience, we say that the \( \mathcal{H}_\infty \) SMO exists if the above optimization problem is solvable.
Remark 3.1. The parameter $\delta$ plays an important role in the overall IFTC scheme. For $\delta$ sufficiently small, $\gamma + \psi(t_f)$ will be $< 1$. From (3.9) and (3.10), a smaller $\delta$ implies a smaller bound on $e$, and $\xi_0$ ultimately determines the bound on the errors. Furthermore, $\psi(t_f)$ decreases with $\delta$, which indicates the $\mathcal{H}_\infty$ attenuation would approach $\sqrt{\gamma}$, which is the ideal case.

**Proof.** Let $e = x - \hat{x}$. Using (3.1) and (3.5), we obtain the error system:

$$
\dot{e} = (A - LC)e + [\Phi(x) - \Phi(\hat{x})] + \Delta \xi + E(f - \nu)
$$

(3.13)

Take the Lyapunov function candidate to be $V_o = e^T Pe$. By using (3.13) and defining $\Phi_d = \Phi(x) - \Phi(\hat{x})$, we arrive at:

$$
\dot{V_o} = e^T [(A - LC)^T P + P(A - LC)]e + e^T P\Phi_d + \\
\Phi_d^T Pe + 2e^T P E(f - \nu) + e^T P \Delta \xi + \xi^T \Delta^T Pe
$$

(3.14)

It is easy to establish that:

$$
e^T P\Phi_d + \Phi_d^T Pe \leq e^T P^2 e + \gamma e^T e
$$

(3.15)

Let $e_n = [e_{n1}^T e_{n2}^T]^T$ and $e_{n1}^T = e_{n1}^T P_n E_{n1}$. By using (3.6), we find that

$$
e^T P E(f - \nu) = e_{n1}^T P_n E_{n1}(f - \nu)
$$

$$
\leq ||e_{n1}||\rho - (\rho + \rho_0)(||\bar{e}_{n1}|| - \frac{(\delta/2)||\bar{e}_{n1}||}{||\bar{e}_{n1}|| + \delta/2})
$$

$$
= \rho \left(\frac{(\delta/2)||\bar{e}_{n1}||}{||\bar{e}_{n1}|| + \delta/2}\right) - \rho_0 ||\bar{e}_{n1}|| + \delta/2
$$

$$
\leq \rho(\delta/2)
$$

(3.16)

Thus, using (3.15) and (3.16), (3.14) satisfies:

$$
\dot{V_o} \leq e^T [(A - LC)^T P + P(A - LC) + P^2 + L_\Phi^2 I]e + e^T P \Delta \xi + \xi^T \Delta^T Pe + \rho \delta
$$

(3.17)

To establish robustness against disturbances in the $\mathcal{L}_2$ sense, we follow [75] to show that the following inequality

$$
\dot{V_o} + e^T H^T He - \gamma \xi^T \xi \leq 0
$$

(3.18)

holds in some set of $e$.

We first prove that the state estimation error converges to a bounded set. By adding
\( e^T H^T H e - \gamma \xi^T \xi \) on both sides of (3.17), we obtain:

\[
\dot{V}_o + e^T H^T H e - \gamma \xi^T \xi \leq -[e^T \xi^T] \Omega \begin{bmatrix} e \\ \xi \end{bmatrix} + \rho \delta \leq - \lambda_{\min}(\Omega)(\|e\|^2 + \|\xi\|^2) + \rho \delta 
\]

(3.19)

where \( \Omega \) is given in (3.8). If the \( \Omega \) is positive definite matrix, then (3.18) holds true for the set 
\[ D = \{ (e, \xi) : \|e\|^2 + \|\xi\|^2 > \frac{\rho \delta}{\lambda_{\min}(\Omega)} \} \]

Therefore, it is required to solve for \( P, K \) and \( \gamma \) such that 
\[ \dot{V}_o - \gamma \xi^T \xi \leq \dot{V}_o + e^T H^T H e - \gamma \xi^T \xi \]

Suppose that the LMI problem is solvable. Then (3.19) holds, and since,
\[ \lambda_{\min}(P)\|e\|^2 \leq V_o \leq \lambda_{\max}(P)\|e\|^2 \]

we obtain, on using (3.19),
\[
\dot{V}_o \leq - \lambda_{\min}(\Omega)(\|e\|^2 + \|\xi\|^2) + \rho \delta + \gamma \xi^T \xi \\
\leq - \lambda_{\min}(\Omega)\|e\|^2 + \rho \delta + \gamma \xi_0^2 \\
\leq -aV_o + b
\]

(3.20)

(3.21)

where \( a, b \) is given in (3.9). Also, note that if \( e \) belongs to the set complement of \( \Lambda \) (i.e. \( \{ e : \|e\|^2 > b/(a\lambda_{\min}(P)) \} \) ), then \( \dot{V}_o < 0 \) which implies the estimation error will converge to \( \Lambda \).

Next, we prove that a certain \( \mathcal{H}_\infty \) attenuation can be achieved. Given that the LMI problem is solvable, it is immediate from (3.19) that:
\[ \dot{V}_o + e^T H^T H e - \gamma \xi^T \xi \leq \rho \delta \]

By integrating both sides from time 0 to \( t_f \):
\[ V_o(t_f) - V_o(0) + \|He\|^2_{L_2} - \gamma \|\xi\|^2_{L_2} \leq \rho \delta t_f \]

where \( \|\omega\|_{L_2} = \sqrt{\int_0^{t_f} \omega^T(t)\omega(t)dt} \). Since \( V_o(0) = 0 \) (from the definition of \( \mathcal{H}_\infty \) gain) and
\[ V_0(t_f) \geq 0, \] it follows that:
\[
\frac{\int_{0}^{t_f} e(t)^T H^T H e(t) dt}{\int_{0}^{t_f} \xi^T \xi dt} \leq \gamma + \psi(t_f)
\]
where \( \psi(t_f) \) is defined in (3.11). This shows that an \( H_\infty \) attenuation of \( \sqrt{\gamma + \psi(t_f)} \) is guaranteed for the SMO.

The \( H_\infty \) SMO can also be used to perform fault estimation by using the concept of equivalent output error injection. One can take the estimated fault as \( \hat{f} = \nu \) defined in (3.6) [73]. This will be important in the design of the fault-tolerant controller. It should be emphasized that the fault estimate is not a perfect reconstruction of the fault signal due to the effect of disturbances and nonlinearities. Let \( \Delta_n = [\Delta_{n1} \Delta_{n2}]^T = T \Delta \) and \( \sigma_{\max}(X) \) to be the largest singular value for the matrix \( X \). The following theorem gives a bound on \( f - \hat{f} \) and shows that some \( H_\infty \) attenuation is obtained for \( f - \hat{f} \): 

**Theorem 3.2.** Assume that the \( H_\infty \) SMO exists and \( \delta = 0 \). Then, an ideal sliding motion takes place in finite time on the hyperplane \( S_e = \{ e : e_{n1} = 0 \} \). When sliding motion takes place and \( H \) is nonsingular, the bound on the fault estimation error is given by:

\[
\|f - \hat{f}\|_{L_2} \leq (\sqrt{\chi_1 + \chi_2}) \|\xi\|_{L_2} \tag{3.22}
\]

where \( \chi_2 = \sigma_{\max}(E_{n1}^{-1} \Delta_{n1}) \) and

\[
\chi_1 = \sigma_{\max}(H^{-1}) \left[ \sigma_{\max}(E_{n1}^{-1} A_{n2}) + L_\phi \|T\| \|T^{-1}\| \sigma_{\max}(E_{n1}^{-1}) \right].
\]

**Remark 3.2.** Ideally, \( \delta \) should be chosen as 0 to make \( e \) go to 0 and to attain sliding motion. However, as \( \delta \) approaches 0, \( \nu \) starts to chatter (due to non-ideal conditions) and it no longer estimates the fault. Thus, to take \( \hat{f} = \nu \), a nonzero small \( \delta \) (of magnitude \( 10^{-2} \sim 10^{-7} \)) is chosen. The paper [81] presents various methods such as boundary layer control, disturbance compensation and more to overcome the problem of chattering. However, these methods are beyond the scope of our work.

The term \( \sqrt{\gamma} \) in (3.22) shows that the fault estimation error obtains certain degree of \( H_\infty \) attenuation. The remaining disturbances are solely due to \( \chi_2 \), which can be viewed as the consequence of the sliding term trying to estimate any disturbance that lies in the subspace of the faults, \( Im(E) \).

**Proof.** The proof is very similar to Theorem 2 in [69] and its Corollary, and is omitted. \( \square \)
3.2.5 Construction of the $\mathcal{H}_\infty$ SMO

The $\mathcal{H}_\infty$ SMO consists of several design parameters, namely $A_{n1}^s$, $H_n$, $\rho_0$ and $\delta$. Furthermore, the design process is not shown explicitly from the previous discussion in Section 3.2. This subsection will show the design process explicitly and how one should choose the design parameter.

Previously, in subsection 3.2.4, we presented an LMI that guarantees the existence of a stable $\mathcal{H}_\infty$ SMO in (3.12). The LMI uses matrices in the original coordinates. However, to simplify the design process, it is beneficial to construct and solve the LMI in the new coordinates. The reason is the computation of the sliding term (3.6) and the observer gain matrix (3.7) requires the matrices $K_n$, $P_{n1}$ and $P_{n2}$ (which are all in new coordinates). Define the following variables:

\[
\begin{aligned}
\Pi_1 &= P_{n1}A_{n1}^s + (A_{n1}^s)^T P_{n1} + \tilde{L}_\Phi I_q + H_n^T H_n \\
\Pi_2 &= P_{n2}A_{n4} + A_{n4}^T P_{n2} - (K_nC_{n4} + C_{n4}^TK_n^T) + \tilde{L}_\Phi I_{n-q} + H_n^T H_n
\end{aligned}
\]

where $\tilde{L}_\Phi = \|T\|\|T^{-1}\|L_\Phi$. In new coordinates, the LMI (3.12) can be rewritten as (taken from [69]):

\[
\begin{bmatrix}
\Pi_1 & P_{n1}A_{n2} & P_{n1}\Delta_{n1} & P_{n1} & 0 \\
A_{n2}^T P_{n1} & \Pi_2 & P_{n2}\Delta_{n2} & 0 & P_{n2} \\
\Delta_{n1}^T P_{n1} & \Delta_{n2}^T P_{n2} & -\gamma I_r & 0 & 0 \\
P_{n1} & 0 & 0 & -I_q & 0 \\
0 & P_{n2} & 0 & 0 & -I_{n-q}
\end{bmatrix} < 0 \tag{3.23}
\]

where $\Delta_{n1} \in \mathbb{R}^{q \times r}$ and $\Delta_{n2} \in \mathbb{R}^{(n-q) \times r}$ are block matrices resulting from the disturbance matrix $\Delta_n$ in new coordinates:

\[
\Delta_n = T\Delta = \begin{bmatrix}
\Delta_{n1} \\
\Delta_{n2}
\end{bmatrix}
\]

One of the major works in designing the $\mathcal{H}_\infty$ SMO is solving the LMI. It is worth noting that the LMI may not always be solvable. Discussion on this matter will be made when the design process is introduced. The design process is summarized below:

1. Check if Assumption 3.1 holds. Typically, Assumption 3.1(a) holds in many cases because faults and disturbances are typically bounded. Assumption 3.2(b) can be verified easily using the rank condition. Finally, the rank condition in Assumption 3.2(c) can be checked using symbolic computation tools. Otherwise, a different condition that is related to the detectability of some matrix pairs described in Lemma 2 of [69] can be used to verify if Assumption 3.2(c) holds.
2. Compute \( T \) and \( S \) using algorithm presented in Section 3.2.2.

3. Transform \( A, C, \Delta \) into new coordinates, i.e. compute \( A_n = TAT^{-1}, C_n = SCT^{-1} \) and \( \Delta_n = T\Delta \). Also, determine \( \hat{L}_\Phi \).

4. Select the design matrix, \( A_{n1}^s \). The eigenvalues should be chosen in the open left half plane. \( A_{n1}^s \) will affect the solvability of the LMI equation. Eigenvalues that are further away from the imaginary axis (meaning the real part is ‘more negative’) will improve but not guarantee the solvability of the LMI equation.

5. Select the error weighting matrix, \( H_n \). The error weighting matrix can be selected in a similar fashion to how one select the state weighting matrix in the LQR design. To make the \( i \)-th state estimation more robust to noise, select a larger value for the \( i \)-th diagonal element of \( H_n \). It was found that \( H_n \) affects the solvability of the LMI. Usually, smaller values in the elements of \( H_n \) would improve solvability of the LMI.

6. Construct and solve the LMI listed in (3.23). MATLAB offers a toolbox called the ‘LMI toolbox’ that helps to solve LMI minimization problem. If the LMI minimization problem is solved successfully, this will provide the designer with matrices \( P_{n1}, P_{n2}, K_n \) and the scalar value \( \gamma \). If MATLAB fails to solve the LMI, then one will need to change \( A_{n1}^s \) or \( H_n \). It is worth emphasizing that \( \hat{L}_\Phi \) plays a huge role in determining the solvability of the LMI equation. If permitted, \( \hat{L}_\Phi \) can be changed too.

7. Construct the observer gain, \( L_n \) stated in (3.7).

8. Select \( \rho_0 \) and \( \delta \) for the sliding term (3.6). As a rule of thumb, \( \rho_0 \) should be selected to be roughly 5 times larger than \( \rho \). Its primary function is to help sliding motion to be achieved faster. As mentioned previously in Remark 3.1 and Remark 3.2, \( \delta \) plays an important role of making \( \nu \) or \( \hat{f} \) to estimate the faults. Typically, a value of magnitude in between \( 10^{-7} \) to \( 10^{-2} \) can result in a good fault estimation. A value that is too small will make \( \hat{f} \) (or equivalently \( \nu \)) to be a high frequency switching signal.

9. Construct the observer and run simulations to verify that the \( \mathcal{H}_\infty \) SMO is able to estimate faults. If \( \hat{f} \) is a high frequency signal, increment \( \delta \). Repeat until satisfactory performance is obtained.

### 3.2.6 \( \mathcal{H}_\infty \) Attenuation and the Fault Estimate

In Theorem 3.2, we presented a bound on the fault estimation error (3.22). The bound was shown to be related to the disturbance attenuation scalar, \( \gamma \). However, it is not immediately
obvious how the fault estimate is able to attenuate disturbance. This subsection serves to clarify this question.

In order to study this phenomenon, consider the error system in new coordinates (taken from (69) of [69]):

\[
\dot{e}_{n1} = A_{n1}^* e_{n1} + A_{n2} e_{n2} + \Phi_{n1} + E_{n1}(f - \nu) + \Delta_{n1} \xi \\
\dot{e}_{n2} = (A_n4 - L_{n4} C_{n4})e_{n2} + \Phi_{n2} + \Delta_{n2} \xi
\]

where \(L_{n4} = P_n^{-1} K_n\), \(\Phi_{n1}\) and \(\Phi_{n2}\) comes from:

\[
\Phi_{n}(x, \dot{x}) = T(\Phi(x) - \Phi(\dot{x})) = \begin{bmatrix} \tilde{\Phi}_{n1} \\ \tilde{\Phi}_{n2} \end{bmatrix}
\]

Note that the sliding surface is chosen to be the set \(\{ e_n : e_{n1} = [I_q\ 0] e_n = 0 \}\). Also, observe that \(e_{n2}\) is not affected by faults. This is due to careful selection of the transformation matrix \(T\). However, \(e_{n2}\) is affected by disturbances. One can write \(e_{n2}\) as:

\[
e_{n2}(t) = e^{(A_{n4} - L_{n4} C_{n4})t} e_{n2}(0) + \int_0^t e^{(A_{n4} - L_{n4} C_{n4})(t - \tau)} \tilde{\Phi}_{n1}(x(\tau), \dot{x}(\tau)) d\tau + \\
\int_0^t e^{(A_{n4} - L_{n4} C_{n4})(t - \tau)} \Delta_{n2} \xi(\tau) d\tau
\]

Now, suppose that \(\delta = 0\) and sliding motion takes place. Strictly speaking, when \(\delta = 0\), one cannot obtain \(\nu_{eq}\). The reason we take \(\delta = 0\) is to perform a rough analysis of how the disturbance in \(e_{n2}\) will affect \(\nu_{eq}\) or the fault estimate. Intuitively, when \(\delta\) is sufficiently small but not zero, then the same argument can be applied.

From Section 2.3.3 this implies \(e_{n1} = \dot{e}_{n1} = 0\). Then, using (3.24), the equivalent output error injection (also the fault estimate) is:

\[
\nu_{eq} = f + E_{n1}^{-1} (A_{n2} e_{n2} + \Phi_{n1} + \Delta_{n1} \xi)
\]

From the above equation, one realizes that the equivalent output error injection does not only depend on the fault. It also depends on the disturbances, the nonlinearity and \(e_{n2}\). One can view \(\nu_{eq}\) as a compensation signal to make \(e_{n1} = 0\). The term \(\nu_{eq}\) makes \(e_{n1}\) more robust against noise, nonlinearity and faults by estimating the sum of these effects. Hence, one could say that \(e_{n1}\) does not require any \(\mathcal{H}_\infty\) disturbance attenuation because the sliding term dominates the role as robust compensation signal for \(e_{n1}\).

However, this also means that the fault estimate is contaminated by other signals. The effect
from the terms $\tilde{\Phi}_{n1}$ and $\Delta_{n1} \xi$ can be hard to minimize. However, it is possible to minimize the magnitude of $e_{n2}$. Consider the expression in (3.25). When $\mathcal{H}_\infty$ technique is used to design the SMO, $L_{n4}$ is chosen by the LMI solver to minimize the effect of $\xi$ on $e_{n2}$. Thus, $v_{eq}$ or $\hat{f}$ will be less affected by the disturbance influencing $e_{n2}$.

In other words, the main purpose of the $\mathcal{H}_\infty$ attenuation is to minimize the effect of $\xi$ on the subsystem of $e_{n2}$. The subsystem of $e_{n1}$ does not really require $\mathcal{H}_\infty$ attenuation due to the sliding term. Therefore, when no disturbance is present in the subsystem of $e_{n2}$, the fault estimate resulting from a $\mathcal{H}_\infty$ SMO and a normal SMO (a SMO designed without applying $\mathcal{H}_\infty$ design technique) are almost identical.

### 3.3 $\mathcal{H}_\infty$ SMO and the LFSS

The main benefit of using the $\mathcal{H}_\infty$ SMO to perform fault estimation is that the estimated fault can be used for fault detection and fault isolation. When faults are estimated successfully, if the $i$-th element of $\hat{f}$ is nonzero, this indicates the $i$-th fault has occurred. Therefore, fault isolation is obtained for free whenever faults are estimated.

Furthermore, the estimated fault signal can be used to determine the seriousness of a fault occurrence and the type of faults that have occurred in a particular actuator or sensor. However, fault isolation typically only identifies which part of a system is faulty. For example, suppose the $i$-th actuator of a system has lost its effectiveness. Fault isolation can only pinpoint to the $i$-th actuator has a fault, but it cannot identify if a loss of actuator effectiveness has occurred. Fault estimation can estimate the fault signal, which can be potentially used to identify if an actuator is faulty along with how much effectiveness has it lost.

The ability to isolate and extract further information of a fault makes the $\mathcal{H}_\infty$ SMO an attractive FDI module to solve the failure diagnosis problem of the LFSS. In this section, we will investigate whether the $\mathcal{H}_\infty$ SMO can be applied to the LFSS.

Consider the colocated LFSS model in (2.5) and (2.6). These equations are rewritten to be:

$$
\dot{x} = \begin{bmatrix} 0 & I \\ -\Omega^2 & -\Delta \end{bmatrix} x + \begin{bmatrix} 0 \\ L \end{bmatrix} u + \begin{bmatrix} 0 \\ L \end{bmatrix} f_a
$$

$$
y_m = [L^T \ 0]x + f_s
$$

where:

$$
f_a = (F_u - I)u
$$

$$
f_s = (F_y - I)[L^T \ 0]x
$$
The matching condition from Assumption 3.1(b) will not be satisfied because $CE = 0$. This issue maybe remedied if we take $\dot{y}_m$ to be additional outputs (or the outputs to be used to perform fault estimation).

Also, referring to the system (3.1) and (3.2), the $\mathcal{H}_\infty$ SMO is not designed to handle sensor faults. Reference [73] provides a technique by introducing a filter at the output to estimate sensor faults. Using the same technique, it may be possible to estimate both sensor and actuator faults. However, additional technical conditions need to be satisfied, making it very challenging to apply this technique to the LFSS to diagnose both sensor and actuator failures. In addition, differentiation of the output is necessary for estimating the actuator faults using the $\mathcal{H}_\infty$ SMO. However, doing this does not always satisfy the matching condition. Hence, we chose not to use the $\mathcal{H}_\infty$ SMO to solve the fast fault diagnosis problem for the LFSS.

### 3.4 Fault Accommodation Controller

Although the $\mathcal{H}_\infty$ SMO cannot be applied to the LFSS, its ability to attenuate disturbance in the state and fault estimation is still an attractive feature. Furthermore, to the author’s best knowledge, there does not appear to be any work that utilizes fault estimation from a SMO to perform fault tolerant control with detailed treatment of closed loop stability. Therefore, we propose a simple fault accommodation strategy that uses the fault estimates from $\mathcal{H}_\infty$ SMO combined with a linear controller introduced in [64] to perform fault accommodation on the system shown in (3.1) and (3.2). Note that we do not consider the occurrence of any sensor faults for performing fault accommodation on the system (3.1) and (3.2).

In our proposed scheme, we use the state estimate from the $\mathcal{H}_\infty$ SMO to stabilize the system and the fault estimate to perform actuator faults accommodation. A similar approach is proposed in [47] which uses a nonlinear controller to perform fault accommodation via cancellation using fault estimate. The nonlinear controller requires high gain switching control, which is a drawback in practical implementation. Our proposed controller does not require high gain switching. The fault estimate from the SMO is then used to cancel out the effect of the fault and certain disturbances. The robustness of state and fault estimation against disturbance provided by the $\mathcal{H}_\infty$ SMO, as suggested by the simulation results from [69], may give rise to states that are less affected by disturbances. This will be verified later through simulation.

#### 3.4.1 Assumptions

The following are the assumptions required for the controller to perform fault accommodation:

Assumption 3.2.
(a) There exist a matrix $K_c \in \mathbb{R}^{m \times n}$ and a scalar $w > 0$ such that the Hamiltonian matrix, $H_m$, is hyperbolic (i.e. $H_m$ has no eigenvalues on the imaginary axis), where $H_m$ is given as:

$$H_m = \begin{bmatrix}
(A - BK_c) & I - \tilde{B} \\
-(2L_\Phi^2 + w)I & -(A - BK_c)^T
\end{bmatrix}$$

and $\tilde{B} = \frac{BB^T}{\|B\|^2}$.

(b) $(A, B)$ is controllable

(c) $\text{rank} \left( \begin{bmatrix} B & E \end{bmatrix} \right) = \text{rank}(B)$

Consider the following continuous time algebraic Riccati equation:

$$A_c^T J + JA_c + J \left( I - \tilde{B} \right) J + 2L_\Phi^2 I = -wI \quad (3.26)$$

where $A_c = A - BK_c$ and $J$ is a positive definite matrix. Assumption 3.2(a) guarantees that there exists a unique positive definite solution $J$ to (3.26). Further information regarding the solvability of (3.26) can be found in [1]. Assumption 3.2(b) ensures that pole placement can be performed on $A$. Hence, one can obtain $K_c$ via pole placement. Assumption 3.2(c) implies that only faults that lie in the actuation space $\text{Im}(B)$ is considered. This implies that there exists some $B^*$ such that:

$$(I - BB^*)E = 0 \quad (3.27)$$

For the rest of the section, let Assumption 3.2 hold. We discuss the construction of the fault tolerant controller in the next subsection.

### 3.4.2 Controller Construction

We propose the following control law to perform FA:

$$u = -\frac{B^TJ\hat{x}}{2\|B\|^2} - K_c\hat{x} - B^*E \hat{f} \quad (3.28)$$

where $\hat{f} = \nu$. We now show that stability of the closed loop system under (3.28) is guaranteed and the states and estimation errors would be ultimately bounded in some set. Let $\alpha$ be an arbitrary positive scalar. Define:

$$\beta = \frac{w\lambda_{\text{min}}(\Omega)}{\left(\|J^2\| + 2\|JBK_c\|\right)^2} \quad (3.29)$$

$$c_1 = \alpha w \beta / 2 \quad (3.30)$$
\[ c_2 = (1 - \alpha/2)\lambda_{\text{min}}(\Omega) - 2\alpha\beta[\|T^{-1}\|\|T\|\|A\| + L_\Phi]^2 \]  
(3.31)

\[ d = [\gamma + 2\alpha\beta\|T^{-1}\|\|A\| + L_\Phi]^2] \xi_0^2 \]  
(3.32)

Finally, let:

\[ k_{\text{max}} = \max[\lambda_{\text{max}}(P), \alpha\beta\lambda_{\text{max}}(J)] \]  
(3.33)

\[ k_{\text{min}} = \min[\lambda_{\text{min}}(P), \alpha\beta\lambda_{\text{min}}(J)] \]  
(3.34)

\[ (r^*)^2 = \inf_{\alpha} \left( \frac{k_{\text{max}}}{k_{\text{min}}} \right) \frac{d}{\min(c_1, c_2)} \]  
subject to \( c_2 > 0 \)  
(3.35)

\[ \Psi = \{(x, e) : \|x\|^2 + \|e\|^2 \leq (r^*)^2 \} \]  
(3.36)

**Theorem 3.3.** Let Assumptions 3.1 and 3.2 hold. Suppose that the \( \mathcal{H}_\infty \) SMO defined by (3.5)-(3.6) with \( \delta = 0 \) exists. Then the FA control law (3.28) stabilizes the system (3.1) and (3.2) where \( \hat{x} \) and \( \hat{f} = \nu \) are provided by the \( \mathcal{H}_\infty \) SMO. Furthermore, the pair \((x, e)\) will converge to the set \( \Psi \), given in (3.36), in finite time after sliding motion is achieved.

**Remark 3.3.** The expression for \( d \) in (3.32) shows that if there is no disturbance (i.e. \( \xi_0 = 0 \)), then \( r^* = 0 \). Hence, \((x, e)\) would go to 0. In the presence of system disturbance, it is not possible to regulate \((x, e)\) to 0. In that case, the theorem gives error bounds for \((x, e)\) as a function of the design parameters and the magnitude of the disturbance.

The design parameter \( \delta \) is taken to be 0 for the analysis of the closed loop system. In practice, it is chosen to be a small nonzero value, whose selection should be done carefully because if \( \delta \) is too small, \( \hat{f} \) will become a high frequency switching signal, which makes the control law hard to implement.

**Proof.** The closed loop system can be obtained by substituting (3.28), (3.27) and \( e = x - \hat{x} \) into (3.1) which results in the following:

\[ \dot{x} = (A - BK_c - \tilde{B}J/2)x + (BK_c + \tilde{B}J/2)e + E(f - \hat{f}) + \Phi(x) + \Delta \xi \]  
(3.37)

Take the Lyapunov function candidate to be \( V = V_o + \alpha\beta x^T Jx \), where \( V_o \) is the Lyapunov function for the SMO defined previously. Using (3.20) and the fact that \( \delta = 0 \):

\[ \dot{V} = \dot{V}_o + 2\alpha\beta x^T J\dot{x} \leq -\lambda_{\text{min}}(\Omega)\|x\|^2 + \gamma\xi_0^2 + 2\alpha\beta x^T J\dot{x} \]  
(3.38)
CHAPTER 3. FAULT ACCOMMODATION USING SLIDING MODE OBSERVER (SMO) 47

Using (3.37), we can expand:

\[ 2x^T J \dot{x} = x^T [A_c^T J + J A_c - J \ddot{B} J] x + 2x^T J (BK_c + \ddot{B} J/2) e + 2x^T J \Phi(x) + \] (3.39)

\[ 2x^T J \left[ E(f - \dot{f}) + \Delta \xi \right] \] (3.40)

Note also that:

\[ 2x^T J \left[ E(f - \dot{f}) + \Delta \xi \right] \leq x^T \left( J^2 / 2 \right) x + 2L_{\Phi}^2 x^T x \] (3.41)

Consider the term \( 2x^T J \left[ E(f - \dot{f}) + \Delta \xi \right] \) in (3.39). We first show that the state is bounded before sliding motion is achieved. Let \( g = \frac{EE^T P_{en1}}{E^T P_{en1}} \). Clearly, \( \|g\| \leq \|E\| \). Then, it can be shown that:

\[ \|E(f - \dot{f})\|^2 \leq \|E\|^2 (2\rho + \rho_0)^2 \]

Hence,

\[ 2x^T J \left[ E(f - \dot{f}) + \Delta \xi \right] \leq x^T \left( J^2 / 2 \right) x + 2(\rho\|E\| + \xi_0\|\Delta\|)^2 \] (3.42)

where \( \bar{\rho} = (2\rho + \rho_0) \).

Substituting (3.26), (3.41) and (3.42) into (3.39), it follows that:

\[ 2x^T J \dot{x} \leq -w\|x\|^2 + \ell_3\|x\|\|e\| + 2\bar{\ell} \] (3.43)

where \( \ell_3 = \|J^2\| + 2\|JBK_c\| \) and \( \bar{\ell} = (\bar{\rho}\|E\| + \xi_0\|\Delta\|)^2 \).

By using (3.29) on left hand side of the following expression, one can show that:

\[ -\alpha \beta w\|x\|^2 + \alpha \beta \ell_3\|x\|\|e\| - \alpha \lambda_{min}(\Omega)\|e\|^2 \leq -\left[ \frac{\alpha w \beta\|x\|^2}{2} + \frac{\alpha \lambda_{min}(\Omega)\|e\|^2}{2} \right] \] (3.44)

Letting \( z = [x^T e^T]^T \), and by using (3.43), (3.44) in (3.38), we finally obtain:

\[ \dot{V} \leq -c_1\|x\|^2 - \bar{c}_2\|e\|^2 + \bar{d} \]

\[ \leq -\min(c_1, \bar{c}_2)(\|z\|^2) + \bar{d} \] (3.45)

where \( c_1 \) is given in (3.30), \( \bar{c}_2 = (1 - \alpha/2)\lambda_{min}(\Omega) \) and \( \bar{d} = \gamma \xi_0^2 + 2\alpha \beta \bar{\ell} \). Note that one can choose \( \alpha \) (usually making it small) such that \( \bar{c}_2 > 0 \). Therefore, for \( \|z\| \) sufficiently large, \( \dot{V} < 0 \), which implies \((x, e)\) will be bounded.

Now consider the case when sliding motion is achieved in the \( \mathcal{H}_\infty \) SMO. This is guaranteed
by Theorem 2. We then have:
\[ 2x^T J[E(f - \hat{f}) + \Delta \xi] \leq x^T (J^2/2)x + 2\|T^{-1}\|^2 \cdot \|E_n(f - \hat{f}) + \Delta_n \xi\|^2 + 2\|T^{-1}\|^2 \|\Delta_n\|^2 \xi_0^2 \]

Using the fact that \( \dot{e}_n = 0 \) (i.e. sliding motion is achieved), one can arrive at:
\[ \|f + E_n^{-1}\Delta_n \xi - \hat{f}\| \leq \|E_n^{-1}\||T||\|A\| + L_\Phi\|e\| \]

and using it on the previous inequality, which results in:
\[ 2x^T J[E(f - \hat{f}) + \Delta \xi] \leq x^T (J^2/2)x + 2\ell_1 \|e\|^2 + 2\ell_2 \quad (3.46) \]

where \( \ell_1 = \|T^{-1}\|^2 \|T\|^2 (\|A\| + L_\Phi)^2 \) and \( \ell_2 = \|T^{-1}\|^2 \|\Delta_n\|^2 \xi_0^2 \).

By replacing (3.46) instead of (3.42) in the steps to obtain (3.43) and (3.45), we arrive at the following equations:
\[ 2x^T J\dot{x} \leq -w \|x\|^2 + \ell_3 \|x\| \|e\| + 2\ell_1 \|e\|^2 + 2\ell_2 \quad (3.47) \]
\[ \dot{V} \leq -\min (c_1, c_2) \|z\|^2 + d \quad (3.48) \]

where \( c_2 \) and \( d \) are given in (3.31) and (3.32).

We will now use (3.48) to show that \( (x, e) \) will converge to the set \( \Psi \) defined in (3.36) once sliding motion is achieved. The bound from \( \Psi \) make the bound to be 0 when \( \xi_0 = 0 \). This is in contrast to (3.45), which contains a constant that does not go to zero when \( \xi_0 = 0 \).

To proceed, first define 2 class \( K_\infty \) functions (see [50] for definition) \( \phi_1, \phi_2 : [0, \infty) \rightarrow [0, \infty) \) to be:
\[ \phi_1(s) = \min \left[ \lambda_{\min}(P), \alpha \beta \lambda_{\min}(J) \right] s^2 \]
\[ \phi_2(s) = \max \left[ \lambda_{\max}(P), \alpha \beta \lambda_{\max}(J) \right] s^2 \]

It is simple to prove that:
\[ \phi_1(\|z\|) \leq V \leq \phi_2(\|z\|) \quad (3.49) \]

Now, define a positive definite function:
\[ M(z) = \kappa \min(c_1, c_2)(\|z\|^2) \]
and \( \mathcal{S} = \{ z : \| z \|^2 \geq h^2 \} \), where \( 0 < \kappa < 1 \) is a scalar and \( h \) is defined to be
\[
h^2 = \frac{d}{(1 - \kappa) \min(c_1, c_2)}
\]
(3.50)

Using (3.48), the following holds:
\[
\dot{V} \leq -\kappa \min(c_1, c_2)(\| z \|^2) = -M(z), \forall z \in \mathcal{S}
\]
(3.51)

Let \( r \) to be:
\[
r^2 = \left( \frac{k_{\text{max}}}{k_{\text{min}}} \right) h^2
\]
where \( k_{\text{min}}, k_{\text{max}} \) are defined by (3.34) and (3.33) respectively. Using the facts that: (i) (3.49) and (3.51) holds, (ii) \( V \geq 0 \) holds for all \( (x, e) \), and (iii) \( \phi_1 \) belongs to class \( \mathcal{K}_\infty \), one can obtain the following expression by applying Theorem 4.18 from [50]:
\[
\| x \|^2 + \| e \|^2 \leq \left[ \phi_1^{-1}(\phi_2(h)) \right]^2 = r^2, \quad \forall t > t_b
\]
(3.52)

where \( t_b \) is some positive scalar.

The bound \( r \) is a function of various design and system parameters. Its dependence can be expressed, by abuse of notation, as:
\[
r = r(\alpha, \kappa; H, A_{q_1}, w; \gamma, \xi_0)
\]

To get a smaller \( r \), we see that (3.50) implies one can take \( \kappa \to 0 \) to minimize \( r \). If we solve the optimization problem (3.35) over \( \alpha \), the bound (3.52) becomes:
\[
\| x \|^2 + \| e \|^2 \leq (r^*)^2, \quad \forall t > t_b
\]
which implies \( (x, e) \) will eventually converge into \( \Psi \) defined in (3.36).

\[
\square
\]

### 3.5 Simulations

In this section, the design of our proposed IFTC scheme is illustrated on a permanent magnet synchronous motor (PMSM) that was studied in [35]. The nonlinearity in this system is not globally Lipschitz but locally Lipschitz with \( L_\phi \) when \( x \) is in some set. We will assume that the states never goes out of the set when the system is under control. Thus, we can still apply
our IFTC design to this system.

Let $\theta$, $\omega$ be the angular position and velocity of the motor respectively, and $i_q$, $i_d$ be the quadrature and direct electrical current respectively. Note that $x^T = [\theta, \omega, i_q, i_d]^T$. The state space model matrices are:

$$A = \begin{bmatrix}
0 & 1 & 0 & 0 \\
0 & 0 & \frac{3p\phi_m}{2J_r} & 0 \\
0 & -\frac{p\phi_m}{L_s} & -\frac{R_s}{L_s} & 0 \\
0 & 0 & 0 & -\frac{R_s}{L_s}
\end{bmatrix}$$

$$B = \begin{bmatrix}
0 & 0 \\
0 & 0 \\
1/L_s & 0 \\
0 & 1/L_s
\end{bmatrix}, \quad C = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}$$

$$E = \begin{bmatrix}
0 \\
0 \\
1/L_s \\
1/L_s
\end{bmatrix}, \Phi(x) = \begin{bmatrix}
0 \\
-\tau/J_r \\
-p\omega_i_d \\
p\omega_i_q
\end{bmatrix}, \Delta = \begin{bmatrix}
0 & 0 \\
1 & 0 \\
0 & 1 \\
0 & 0
\end{bmatrix}$$

where $J_r$ is the moment of inertia of the rotor, $R_s$ and $L_s$ are the stator resistance and inductance respectively, $p$ is the number pole pairs in the PMSM, $\phi_m$ is the amplitude of permanent magnet flux linkages and $\tau$ is the load torque. The system parameters are taken to be $J_r = 9 \times 10^{-3}$ kgm$^2$, $R_s = 0.34$ $\Omega$, $L_s = 0.02$ $H$, $p = 4$, $\phi_m = 9.57 \times 10^{-2}$ and $\tau = 0$ Nm, which are similar to those in [35]. Also, the Lipschitz constant was chosen to be $L = 0.6$. The initial condition of the system is taken to be $x(0) = [0, 0, 0, 0.1]^T$.

The system noises, $\xi_1$ and $\xi_2$, are taken to be uniform random variables $U[-2, 2]$ and $U[-0.7, 0.7]$ respectively, where $\xi = [\xi_1, \xi_2]^T$. The fault was taken to be similar to the fault considered in the example of [47]:

$$f(t) = \begin{cases}
0, & t < 10 \\
0.4 + 0.5 \cos(2(\pi/20)t), & t \geq 10
\end{cases}$$

First, the $H_\infty$ SMO is designed. The design parameters were taken to be: $\rho = 1$, $\rho_0 = 10$, $\delta = 5 \times 10^{-3}$, $A_1^s = -30$ and $H_n = diag(0.01, 0.1, 0.1, 0.1)$. Using the algorithm provided in
In [69], $T$ and $S$ were calculated to be:

$$T = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad S = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The minimization problem for the LMI (3.12) was solved using MATLAB in the transformed coordinates (refer to [69] for exact expressions). One finds $\gamma = 5.46 \times 10^{-4}$,

$$P_n = \begin{bmatrix} 0.0120 & 0 & 0 & 0 \\ 0 & 131.7646 & -0.8611 & 0 \\ 0 & -0.8611 & 0.0057 & 0 \\ 0 & 0 & 0 & 47.1666 \end{bmatrix}, \quad K_n = \begin{bmatrix} 9590.9 \\ 9.4328 \\ 1734.1 \end{bmatrix}.$$

Next, the fault tolerant controller is designed. The design parameters were chosen to be: $w = 0.1$ and

$$K_c = \begin{bmatrix} 0.1881 & -0.2919 & 0.3200 & 0 \\ 0 & 0 & 0 & 1.6600 \end{bmatrix}$$

Solving (3.26) gives:

$$J = \begin{bmatrix} 0.9222 & 0.0283 & -0.0017 & 0 \\ 0.0283 & 0.0638 & 0.0959 & 0 \\ -0.0017 & 0.0959 & 0.1977 & 0 \\ 0 & 0 & 0 & 0.0041 \end{bmatrix}$$

To show the effectiveness of our IFTC scheme in dealing with noise, comparisons against several IFTC schemes were made. We used the same fault accommodation controller proposed in (3.28) but with different observers, which include the adaptive observer proposed by [47] and a normal SMO design mentioned in [69]. While $H_\infty$ SMO design involves the minimization of disturbance attenuation, the normal SMO does not. This is the key difference between the two SMOs.

The normal SMO is designed using the same parameters as the $H_\infty$ SMO. In this case, the matrix $H_n$ is not used since the normal SMO does not require one to minimize the $H_\infty$ gain.
By solving an LMI feasibility problem using MATLAB, one obtains:

\[
P_{n(\text{normal})} = \begin{bmatrix}
0.0547 & 0 & 0 & 0 \\
0 & 4.1225 & -1.0621 & 0 \\
0 & -1.0621 & 0.4722 & 0 \\
0 & 0 & 0 & 2.5492
\end{bmatrix}
\]

\[
K_{n(\text{normal})} = \begin{bmatrix}
16.9749 & 0 \\
-0.7314 & 0 \\
0 & -31.9779
\end{bmatrix}
\]

We now design the adaptive observer based on [47]. To avoid confusion, all parameters from [47] retain the same symbol as the original paper but are subscripted with the symbol 'J'. We chose \( \Gamma_J = 1 \), \( \sigma_J = 0.5 \), \( Q_J = 0.1I_4 \), \( \epsilon_J = 0.01 \), and

\[
K_J = \begin{bmatrix}
27.0978 & -1.4061 & 0 \\
2.6461 & 25.9026 & 0 \\
-1.8728 & 36.9022 & 0 \\
0 & 0 & 8
\end{bmatrix}
\]

Finally, by solving the LMI proposed in [47], \( P_J \) can be obtained to give \( R_J = [-1.1743, 43.9679, 0]^T \).

For more information on the adaptive observer, we refer the reader to the original paper [47].

For comparison purposes, the plots in this section are labelled with three different symbols, namely \([H_\infty], [N], [J]\) to represent all the results related to the \( H_\infty \) SMO, the normal SMO, and the adaptive observer, respectively. Figures 3.1 and 3.2 show the norm of the state estimation errors and the fault estimates from the three different observers respectively. Both figures show that the \( H_\infty \) SMO produces state and fault estimates that are more robust against disturbances when compared to the estimates from the 2 other observers. This is due to the attenuation properties of the \( H_\infty \) SMO.

Next, Figure 3.3 shows the effectiveness of our proposed fault accommodation strategy by considering 2 cases. In the first case (denoted by [No FA]), no fault cancellation strategy is performed, i.e. \( \hat{f} = 0 \), whereas the full fault accommodation scheme is applied in the second case (denoted by [FA]). The \( H_\infty \) SMO is used to generate both state and fault estimates for the controller (3.28). Without fault cancellation, both angular position and velocity of the rotor exhibits oscillating behaviour once the fault occurs at \( t = 10s \). However, no significant change in state behaviour is observed when our fault accommodation strategy is used.

Finally, Figure 3.4 shows the actual state behaviour using the three different observers to provide state and fault estimates to our proposed controller. It can be inferred from the graphs
Figure 3.1: Norm of the state estimation error for the three different observers.

Figure 3.2: Estimated fault obtained from the three different observers.
that almost all the states have smaller amplitudes when the $\mathcal{H}_\infty$ SMO is used in our controller instead of the two alternatives. The $\mathcal{H}_\infty$ SMO provides state and fault estimates that are more robust to noise as illustrated in Figure 3.1 and 3.2. Hence, the controller has more accurate information to control the system. This illustrates the benefits of having $\mathcal{H}_\infty$ attenuation in the state and fault estimates when they are used for control and FA purposes.

3.6 Discussion

In this chapter, the $\mathcal{H}_\infty$ SMO was first studied. Then, we studied its applicability to the LFSS and found that $\mathcal{H}_\infty$ SMO is not a good solution for solving the failure isolation problem of the LFSS. However, the $\mathcal{H}_\infty$ SMO provides state and fault estimates that are robust against disturbance. Hence, it becomes an attractive component to be used for fault tolerant control purposes. Using the linear controller proposed in (3.28), we were able to show that the fault estimate can be used to mitigate the effect of faults in the system. Furthermore, simulation results show that the robust state and fault estimates obtained from the $\mathcal{H}_\infty$ SMO help to reduce the effect of disturbances on the state behavior.

Our proposed integrated fault tolerant control scheme enjoys the following advantages:
1. The $\mathcal{H}_\infty$ SMO provides fault estimates which can be used for fault isolation purposes. It can also provide further details about the type of faults that has occurred and the magnitude of the fault. For detailed discussion, refer to Section 3.3.

2. The states, state estimates and fault estimates of the overall scheme are robust to disturbances.

3. The proposed fault accommodation controller is simple to design and it requires no high frequency switching. This makes the overall scheme more practical to be implemented in real life.

However, this scheme suffers from various limitations as well. They are listed as follows:

1. The design process of the $\mathcal{H}_\infty$ SMO is complicated and requires trial and error tuning. Detailed design steps have been outlined in subsection 3.2.5 to help clarify the design process.

2. The assumptions, in particular Assumption 3.1(b), required by the $\mathcal{H}_\infty$ SMO may be restrictive. For example, the LFSS in its original form will not satisfy the matching condition.
3. The LMI proposed in (3.12), or equivalently (3.23), may not always be solvable. The design matrices must be chosen carefully to ensure solvability of the $\mathcal{H}_\infty$ SMO.
Chapter 4

Fast Fault Diagnosis and Fault Tolerant Control for LFSS

Previously, in Chapter 3, it was observed that it is difficult to adapt the $H_\infty$ sliding mode observer (SMO) to estimate both sensor and actuator faults for the purpose of fault isolation. To solve the problem of isolating for faults in real time, we propose to use structured residuals.

In this chapter, we consider the first-order, finite dimensional, modal large flexible space structure (LFSS) model given in (2.5) and (2.6). Let $n$ be the number of states and $m$ be the number of sensor-actuator pairs. For ease of the reader’s reference, the model is restated below:

\[ \dot{x} = \begin{bmatrix} 0 & I \\ -\Omega^2 & -\Delta \end{bmatrix} x + \begin{bmatrix} 0 \\ L \end{bmatrix} F_u u \]

\[ y_m = F_y [L^T \ 0] x \]

where $u$, $y_m \in \mathbb{R}^m$ are the input and the measured output of the LFSS respectively, $\Omega$, $\Delta \in \mathbb{R}^{n \times n}$ are the modal stiffness and damping matrices respectively, $L \in \mathbb{R}^{n \times m}$ is the input (output) distribution matrix, and $F_u$, $F_y \in \mathbb{R}^{m \times m}$ are input and output failure matrices (which are diagonal matrices with elements of 0 or 1 on the diagonal). The states, $x$ is a concatenation of the modal position, $\dot{d} \in \mathbb{R}^n$, and the modal velocity, $\dot{d}$. For more information on the model, refer to Section 2.2. To simplify the development of our work, we do not consider constant disturbances in our work, which Huang takes into account in [38].

Huang presents two different fault tolerant control (FTC) laws and a fault diagnosis algorithm in [38]. The main drawback of the proposed fault diagnosis algorithm by Huang [38] is that it requires output steady state information to diagnose the pairs of sensor-actuator that have failed. Due to the low damping of LFSSs, it takes a long time for LFSSs to reach steady state. Hence, Huang’s fault diagnosis algorithm is slow.
We propose to use structured residuals to provide fast fault diagnosis for Huang’s fault tolerant controllers. Fault estimation techniques proposed in [46], [47] and unknown input observers scheme proposed in [15] require the system to satisfy the same conditions required by the $\mathcal{H}_\infty$ SMO. Therefore, techniques as such are not applicable to colocated LFSSs. On the other hand, schemes proposed in [82] and [83] are complicated and time consuming to design because they require the designer to model all faults. More details can be found in Section 2.3.5. Structured residual generator is simple to design and it does not require restrictive assumptions. By assuming the faults are additive, structured residuals can be used to detect and isolate a wide range of fault types. It is also not required to know or model the faults a priori. Structured residuals can be used to isolate faults when the number of faults outnumbers the number of residuals. This is an attractive feature that can be used to solve the fast fault diagnosis problem for colocated LFSSs. For information on structured residuals, refer to Section 2.3.2.

Our proposed scheme uses an observer-based residual generator to generate residuals. A fault isolation filter is constructed to transform the residuals to a set of structured residuals that gives direct information on which sensor-actuator pairs have failed. The observer-based residual generator combined with the fault isolation filter is called the structured residual generator. The information provided by the structured residuals is then passed to the fault tolerant controllers proposed by Huang.

Unlike Huang’s fault diagnosis algorithm, structured residuals do not require steady state information to diagnose/isolate faults in LFSSs and can thus diagnose failures much faster. For the same system that took Huang’s algorithm roughly 21200s to diagnose for failure, our method only requires approximately 10s. This is a significant improvement in the time taken for diagnosis. When used as a FDI module, the ability to produce real-time diagnosis can also lead to improvements in the closed-loop response of an active FTC scheme.

### 4.1 General Problem Description

In order to apply structured residuals to the LFSS, the model stated in (4.1) and (4.2) needs to be rewritten into a more general form. First, let:

\[
A = \begin{bmatrix} 0 & I \\ -\Omega^2 & -\Delta \end{bmatrix} \in \mathbb{R}^{2n \times 2n} \tag{4.3}
\]

\[
B = \begin{bmatrix} 0 \\ L \end{bmatrix} \in \mathbb{R}^{2n \times m} \tag{4.4}
\]
\[ C = \begin{bmatrix} L^T & 0 \end{bmatrix} \in \mathbb{R}^{m \times 2n} \]

Furthermore, let:

\[ f_a = (F_u - I)u \] \hspace{1cm} (4.6)
\[ f_s = (F_y - I)Cx \] \hspace{1cm} (4.7)

Then, (4.1) and (4.2) can be rewritten as:

\[ \dot{x} = Ax + Bu + Bf_a \] \hspace{1cm} (4.8)
\[ y_m = Cx + f_s \] \hspace{1cm} (4.9)

We assume that the LFSS model is perfectly known and there are no disturbances or uncertainties in the system. The model considered by Huang in [38] takes constant disturbances into account. Our assumptions are made to simplify the development of the structured residual generator.

From here on, the new model (4.8) and (4.9) will be used to design the structured residual generator. Note that the multiplicative failures, \( F_u \) and \( F_y \) have been changed into additive faults, \( f_a \) and \( f_s \) respectively. We refer to a failure as the complete breakdown of a component (either an actuator or a sensor) and a fault as a phenomenon that alters the behavior of a component. When there is no fault/failure present, \( f_a = 0 \) and \( f_s = 0 \). There are many works in the literature that study diagnosis of additive faults.

Recall that every actuator is paired with a sensor and placed in a same spatial location on the LFSS. This motivates us to define the location of a failure or a fault. Informally speaking, if a location is said to have a failure (or fault), then it means either the sensor or its paired actuator at the particular location has failed (or has a fault). Denote the \( i \)-th element of \( f_a \) and \( f_s \) by \( f_a^i \) and \( f_s^i \) respectively. The formal definition is provided as follows:

**Definition 4.1.** Let \( i \in \{1, 2, ..., m\} \). The \( i \)-th location is said to have a fault (or failure) if the \( i \)-th actuator or \( i \)-th sensor has a fault (or failure), i.e. either \( f_a^i \neq 0 \) or \( f_s^i \neq 0 \).

The main goal of this chapter is to devise a fast fault diagnosis scheme to determine which location has a fault/failure if a fault occurs. More formally, we would like to detect when \( f_a \neq 0 \) or \( f_s \neq 0 \), and determine the index \( i \) of \( f_a \) or \( f_s \) that is nonzero. The process of determining when \( f_a \neq 0 \) or \( f_s \neq 0 \) is often known as fault detection. This can be easily done by using observer-based generators. The operation of determining \( i \) belongs to a process called fault isolation. For more information on fault detection and fault isolation methods, refer to Section 2.3.
4.2 Observer-based Residual Generator (ORG)

To construct a structured residual generator, one begins with constructing a normal residual generator. In our work, we propose to use an observer-based residual generator to generate residuals, which can be easily used to detect if failures have occurred. In Section 4.3, the residuals are processed to isolate failure. The observer-based residual generator is given below:

\[
\dot{\hat{x}} = A\hat{x} + Bu + H(y_m - C\hat{x}) \tag{4.10}
\]
\[
r = y_m - C\hat{x} \tag{4.11}
\]

where \( H \in \mathbb{R}^{2n \times m} \) is the observer gain (chosen by the designer) and \( r \in \mathbb{R}^m \) is the residual signal. Under the assumption that no uncertainty is present and transient effects have completely died off, the residual generator is designed such that \( r = 0 \) implies there is no fault present and \( r \neq 0 \) would signify the occurrence of faults.

It turns out that with the assumptions previously made on the LFSS model in [78], a LFSS is not guaranteed to be fully controllable and observable. Since our proposed methodology requires an observer, it is beneficial to study the observability of the LFSS. The authors of [53] give the condition for which LFSSs are observable. The condition is adapted to our notation and is shown in the following proposition.

**Proposition 4.1.** The system (2.5) and (2.6), without failure (i.e. \( F_u = I \) and \( F_y = I \)), is observable if and only if, for all eigenvalues \( \lambda \) of \( A \):

\[
\text{rank} \left[ \begin{array}{c} C \\ \lambda^2 I + \lambda\Delta + \Omega^2 \end{array} \right] = n \tag{4.12}
\]

It can be shown that \( \text{det}(A - \lambda I) = \text{det}(\lambda^2 I + \lambda\Delta + \Omega^2) \). This implies that if \( \lambda \) is an eigenvalue of \( A \), \( \text{rank}(\lambda^2 I + \lambda\Delta + \Omega^2) \neq n \). Hence, \( C \) must always be considered when investigating the observability of the LFSS. It also tells us that the locations of the sensor-actuator pairs are crucial in making the LFSS observable.

The assumptions made for the decentralized servomechanism problem for the LFSS were found to be insufficient to guarantee that a LFSS is observable. In other words, a stable observer for the LFSS might not exist. To avoid the case that an observer may not exist, we will assume that the LFSS is always observable. This assumption is not necessarily restrictive. One can carefully select the locations of the sensor-actuator pairs to make the LFSS observable. In the case when the system is only detectable, then an ORG can still be designed but the structured residuals will take a longer time to isolate faults.

**Assumption 4.1.** Assume that (4.12) holds. This assumption is equivalent to the fact that a
LFSS is observable and, thus, a stable observer always exists.

After examining the observability of the LFSS, the next step would be to study the so-called fault detectability. Informally, fault detectability refers to the ability of a residual generator to generate a nonzero residual signal when a single fault occurs. A single fault refers to the case of only a single element in the fault vector is nonzero. The following formal definition of (weak) fault detectability is taken from [12], which can be commonly found in other fault detection literature.

**Definition 4.2.** Let \( i \in \{1, 2, ..., q\} \), where \( q \) is some integer. Also, define \( f \in \mathbb{R}^q \) to be the fault vector that contains the faults of interest. The \( i \)-th fault, \( f^i \) (the \( i \)-th element of \( f \)) is said to be detectable if there exists a stable residual generator such that \( r(t) \neq 0 \) when \( f^i \neq 0 \).

A necessary and sufficient condition for the detectability of a single fault can be given in terms of the structure of the residual generator in the frequency domain. Let \( R(s) \) be the Laplace transform of \( r(t) \) and \( F(s) \) be the Laplace transform of \( f(t) \). Then, design a residual generator and represent it in the frequency domain as \( G_f(s) \). They are often related as follows:

\[
R(s) = G_f(s)F(s)
\]

More explicitly, we can represent the above equation as:

\[
R(s) = \begin{bmatrix}
F^1(s) \\
\vdots \\
F^i(s) \\
\vdots \\
F^q(s)
\end{bmatrix} = \begin{bmatrix}
G^1_f(s) & ... & G^i_f(s) & ... & G^q_f(s)
\end{bmatrix}
\]

where \( G^i_f(s) \) is the \( i \)-th column of \( G_f(s) \) and \( F^i(s) \) is the \( i \)-th element of \( F(s) \). The condition for checking the detectability of a single fault (taken from [12] and [13]) is given below:

**Proposition 4.2.** The \( i \)-th fault is detectable if and only if \( G^i_f(s) \neq 0 \).

The investigation of fault detectability and fault isolability can be performed through the study of the input-output behavior of the faults and the residuals. First, consider the error system obtained from the observer and the LFSS. Let \( e = x - \hat{x} \), and using (4.8), (4.9), (4.10) and (4.11), one obtains the following error system:

\[
\dot{e} = (A - HC)e + \begin{bmatrix} B & -H \end{bmatrix} \begin{bmatrix} f_a \\ f_s \end{bmatrix}
\]

(4.13)
The input-output behaviour can be studied in the frequency domain. Therefore, (4.13) and (4.14) are transformed into the frequency domain:

\[
\mathcal{R}(s) = \mathcal{G}_f(s) \begin{bmatrix} \mathcal{F}_a(s) \\ \mathcal{F}_s(s) \end{bmatrix}
\]  

(4.15)

where

\[
\mathcal{G}_f(s) = C(sI - A + HC)^{-1} \begin{bmatrix} B - H \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & I \end{bmatrix}
\]  

(4.16)

Using Proposition 4.2, for every fault (i.e. \( f^i_a \) and \( f^i_s \)) to be detectable, none of the columns of \( \mathcal{G}_f(s) \) stated in (4.16) can be 0. This condition can be checked by evaluating \( \mathcal{G}_f(s) \) using MATLAB or some symbolic mathematics tools and checking for columns of zeros.

However, calculation of the matrix inverse in \( \mathcal{G}_f(s) \) typically takes a long time. Furthermore, fault detectability should be inherently a system property and one should be able to study it using just the knowledge of the system. The author of [63] proposed a fault detectability condition that only uses the knowledge of the system. The condition is adapted to our notation and system as follows:

**Theorem 4.1.** Let \( i \in \{1, ..., m\} \). The \( i \)-th actuator fault, \( f^i_a \) is detectable if and only if, for all \( s \in \mathbb{C} \):

\[
\text{Im} \left( \begin{bmatrix} 0 \\ B^i \end{bmatrix} \right) \cap \text{Im} \left( \begin{bmatrix} C \\ A - sI \end{bmatrix} \right) = \{0\}
\]

where \( B^i \) is the \( i \)-th column of \( B \). Similarly, the \( i \)-th sensor fault, \( f^i_s \) is detectable if and only if, for all \( s \in \mathbb{C} \):

\[
\text{Im} \left( \begin{bmatrix} I^i \\ 0 \end{bmatrix} \right) \cap \text{Im} \left( \begin{bmatrix} C \\ A - sI \end{bmatrix} \right) = \{0\}
\]

where \( I^i \) is the \( i \)-th standard basis vector (or the \( i \)-th column in the identity matrix).

To check if a fault is detectable using computer, then one can apply the following corollary that follows directly from the above theorem (Theorem 4.1).

**Corollary 4.1.** The \( i \)-th actuator fault, \( f^i_a \) is detectable if and only if, for all \( s \in \mathbb{C} \):

\[
\text{rank} \left( \begin{bmatrix} 0 & C \\ B^i & A - sI \end{bmatrix} \right) = \text{rank} \left( \begin{bmatrix} C \\ A - sI \end{bmatrix} \right) + 1
\]
Similarly, the $i$-th sensor fault, $f_i^s$, is detectable if and only if, for all $s \in C$:

$$\text{rank} \begin{bmatrix} I^i & C \\ 0 & A - sI \end{bmatrix} = \text{rank} \begin{bmatrix} C \\ A - sI \end{bmatrix} + 1$$

**Proof.** Suppose $V_1$ and $V_2$ are two matrices with the same number of rows. For the actuator fault, let $V_1 = [0^T (B^iT)^T]^T$ and $V_2 = [C^T (A - sI)^T]^T$. As for the sensor fault, let $V_1 = [(I^i)^T 0^T]^T$ and $V_2 = [C^T (A - sI)^T]^T$.

A fault is detectable if and only if $\text{Im}(V_1) \cap \text{Im}(V_2) = 0$. It is well known that $\text{Im}(V_1) \cap \text{Im}(V_2) = 0$ is equivalent to $\text{rank}([V_1 \ V_2]) = \text{rank}(V_1) + \text{rank}(V_2)$. By applying this fact, the final result follows directly.

So far, we have only considered the detectability of a single fault occurrence. A reasonable question to ask is: when there are multiple faults occurring at the same time, can the residual generator always generate $r \neq 0$ (or can one detect the multiple faults occurrence)? In general, the answer is no because it depends on the fault signal.

To study this problem, Massoumnia, in [57], proposed to study if the faults (i.e. $[f_a^T f_s^T]^T$) in system (4.13), (4.14), or alternatively (4.15), is input observable. The definition for input observability is given in Definition 2.1. As suggested in [57], one way to study if the faults are input observable is to check if there exists a left inverse for $G_f(s)$ expressed in (4.16).

Since $G_f(s)$ has more columns than rows, one can never find a left inverse for it. Therefore, the faults are not input observable, which means that the occurrence of multiple faults can generate $r = 0$ for all time after the faults have occurred. An intuitive explanation to this problem is the occurrence of multiple faults may have the chance to cancel out their effect at the output. Of course, the chances of having this to occur are rather slim because this phenomenon requires specific fault signals to occur together at the same time. To guarantee fault detectability, we will assume only single fault will occur.

### 4.3 Structured Residual Generator (SRG)

In the previous section, we showed that residuals from the ORG can be used to detect single fault occurrence if all single faults satisfy Theorem 4.1. However, it is also possible to isolate faults using residuals. One way to perform fault isolation using residuals is through structured residuals. The book [32] provides extensive material on structured residual. In this book, the method taken to design structured residuals is not systematic and it is not studied rigorously (see Example 7.9 and Example 7.10 of [32]). In fact, to our best knowledge, there is little mathematically rigorous study of this method in the literature.
In our work, we applied the structured residual generator to the colocated LFSS, which greatly improves the time taken to perform fault diagnosis when compared to [38]. Also, we systematized the design methodology presented by Gertler [32] for isolating fault in a colocated LFSS, which allows the design to be performed automatically by a computer. Then, we showed formal mathematical proofs to some known results related to structured residuals which are not available in the literature. Many of the results shown in this section originated from Gertler’s work in [33] and [34]. Finally, we presented the fault isolability condition of a faulty location based on our proposed structured residual generator for the colocated LFSS.

Our fault isolation problem is as follows: Given the residual \( r \) or \( \mathcal{R}(s) \) from the ORG, find a transformation matrix (or system) \( \mathcal{W}(s) \in \mathbb{C}^{m \times m} \) such that the resulting output from \( \mathcal{W}(s) \), \( \bar{r} \in \mathbb{R}^m \) or \( \bar{\mathcal{R}}(s) \) can determine which locations of a LFSS are faulty or have failed. The term \( \bar{r} \) represents the collection of structured residuals. The relationship between \( \mathcal{R}(s) \) and \( \bar{\mathcal{R}}(s) \) is described below:

\[
\bar{\mathcal{R}}(s) = \mathcal{W}(s)\mathcal{R}(s)
\]

Define the fault-to-structured-residual transfer matrix, \( \mathcal{M}(s) \in \mathbb{C}^{m \times 2m} \) to be:

\[
\mathcal{M}(s) = \mathcal{W}(s)\mathcal{G}_f(s)
\]

(4.17)

Then, by using (4.15), it follows directly that:

\[
\bar{\mathcal{R}}(s) = \mathcal{M}(s) \begin{bmatrix} \mathcal{F}_a(s) \\ \mathcal{F}_s(s) \end{bmatrix}
\]

(4.18)

The term \( \bar{r} \) is known as structured residual because the elements of \( \bar{r} \) are designed to respond to certain faults only. When a fault occurs, certain elements of \( \bar{r} \) would show up as zeros while others are strictly nonzero in an uncertainty-free situation. By investigating which elements are zeros in \( \bar{r} \), one can deduce which fault has occurred. Using (4.18), this means that one would have to assign 0 to specific elements in \( \mathcal{M}(s) \) by appropriately selecting \( \mathcal{W}(s) \). \( \mathcal{M}(s) \) is called the Structured Residual Generator (SRG). Note that which elements are selected to be 0 is dependent on the nature of the problem and the choice of the designer. Later on, we will show that there is a limit on the number of 0’s to be assigned on a row of \( \mathcal{M}(s) \).

### 4.3.1 Structures of the SRG for the colocated LFSS

In this subsection, we will investigate the possible structure of \( \mathcal{M}(s) \) that can be used for isolating the pairs of sensor-actuator that are faulty (or the locations that are faulty). We propose 2 structure for \( \mathcal{M}(s) \) to isolate faulty location. Note that the faults considered are general
additive actuator and sensor faults as shown in (4.8) and (4.9). In Section 4.3.4 we will apply our results for additive faults specifically to failures.

Since the LFSS has a colocated structure, it would be useful if one can isolate faults based on the locations. For example, if actuator 1 or sensor 1 has a fault, then the residuals will indicate that location 1 has a fault. Having the SRG to behave as such allows us to easily generate the combined failure matrix, $F$, which is used in the PID fault tolerant controller introduced in Section 2.4. The best structure for $M(s)$ to have in order to achieve this fault isolation scheme is as follows:

$$
M(s) = \begin{bmatrix}
* & 0 & \cdots & 0 & * & 0 & \cdots & 0 \\
0 & \ddots & \ddots & \vdots & 0 & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots & 0 \\
0 & \cdots & 0 & * & 0 & \cdots & 0 & *
\end{bmatrix}
$$

(4.19)

where $*$ denotes a nonzero rational function of $s$.

The structure presented in (4.19) is very attractive for $M(s)$. This structure implies that if the $i$-th residual is nonzero, then there is a fault in either the $i$-th actuator or $i$-th sensor. With such behavior, the structured residuals can be used to isolate failures that occur simultaneously. For example, if location 1 and location 2 have faults occurred, then residual 1 and residual 2 would be nonzero while other residuals are 0. Unfortunately, we will show (later on) that one cannot obtain this structure in general.

Now, let ‘*’ denote some rational function of $s$ not excluding the possibility of it being 0. Another structure is to change the ‘*’ to 0 and the 0 to ‘*’ in (4.19), meaning:

$$
M(s) = \begin{bmatrix}
0 & * & \cdots & * & 0 & * & \cdots & * \\
* & \ddots & \ddots & \ddots & \vdots & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \vdots & \ddots & \ddots & 0 \\
* & \cdots & 0 & * & 0 & \cdots & * & 0
\end{bmatrix}
$$

(4.20)

The idea behind this structure is that if the $i$-th location has a fault, then the $i$-th residual will have a value of 0. The main disadvantage of this structure is one cannot identity simultaneous faults from 2 different locations. Hence, if one were to use the structure in (4.20), then one will have to assume that only one location can have a single fault. If two faults from different locations occur simultaneously, two possible outcomes might happen:

- All structured residuals are nonzeros. This is a good indication that multiple failures have occurred simultaneously.
- One or more structured residuals are zeros. This might be possible because the faults may lie within the nullspaces of the rows $M(s)$. This will result in one or more of the structured residuals to be zeros. When only one structured residual is 0, this can lead to an incorrect fault diagnosis. To illustrate this, consider the following simple example.

**Example 4.1.** Consider a static system. Suppose that $M(s)$ is given as follows:

$$M(s) = \begin{bmatrix}
0 & -5 & 2 & 0 & 5 & -5 \\
5 & 0 & 3 & 2 & 0 & -4 \\
2 & 5 & 0 & -4 & -2 & 0
\end{bmatrix}$$

Now, assume that actuator 2 has a fault of $F^2_a(s) = 1$ and sensor 3 has a fault of $F^3_s(s) = -1$. This implies:

$$\bar{R}(s) = M(s) \begin{bmatrix}
0 \\
1 \\
0 \\
0 \\
0 \\
-1
\end{bmatrix} = \begin{bmatrix}
0 \\
4 \\
5
\end{bmatrix}$$

Using the aforementioned rule to diagnose the fault location, this would imply location 1 has a fault (because structured residual 1 is zero). However, in reality, both actuator 2 and sensor 3 are faulty. This shows that using the SRG in the presence of multiple faults can lead to an incorrect diagnosis.

In subsection 4.3.3, the feasibility of the two proposed structures, (4.19) and (4.20), will be investigated. Before investigating their feasibility, the desired properties of the SRG (i.e. $M(s)$) are described in the next subsection.

### 4.3.2 Desired Properties of SRG

Before designing the SRG, $M(s)$, one needs to understand what are the important properties of $M(s)$. There are only 2 essential properties that the SRG should have:

1. The SRG must be able to identify the location of each single fault occurrence, i.e. able to identify either $f^i_a \neq 0$ or $f^i_s \neq 0$. 
2. The SRG must be stable and requires no differentiator to implement it. This is desirable because, in practice, we do not want $\bar{r}$ to become large due to the presence of disturbances and uncertainties and not the fault.

### 4.3.3 Feasibility Study of the Two SRG

In this subsection, we will investigate the feasibility of the 2 structures stated in (4.19) and (4.20).

#### The number of zeros in rows of $\mathcal{M}(s)$

Recall that the SRG is governed by the relationship given in (4.17). To proceed with our study, one ask the following question when selecting zeros in $\mathcal{M}(s)$ - are there some conditions that would guarantee the existence of $\mathcal{W}(s)$ to give rise to the selected structure of $\mathcal{M}(s)$? In order to answer this question, we will divert our focus from designing a SRG for the LFSS to a general linear system. By abusing previous notation, redefine $\mathcal{G}_f(s) \in \mathbb{C}^{p \times q}$ to be a general fault-to-residual transfer matrix [not the one defined in (4.16)] and $\mathcal{W}(s) \in \mathbb{C}^{p \times p}$ to be its corresponding fault isolation filter (since we want to keep the problem similar to ours where $\mathcal{W}(s)$ is square).

First, we introduce the algorithm to construct zeros in a row of $\mathcal{M}(s)$ via appropriate construction of the corresponding row in $\mathcal{W}(s)$. Let $\mathcal{W}(s) = [(\mathcal{W}^{(1)}(s))^T \cdots (\mathcal{W}^{(p)}(s))^T]^T$ and $\mathcal{G}_f(s) = [\mathcal{G}_f^{(1)}(s) \cdots \mathcal{G}_f^{(q)}(s)]$, where $\mathcal{W}^{(i)}(s) \in \mathbb{C}^{1 \times p}$ is the $i$-th row vector of $\mathcal{W}(s)$, $\mathcal{G}_f^\ell(s) \in \mathbb{C}^{p \times 1}$ is the $\ell$-th column vector of $\mathcal{G}_f(s)$. Thus, from (4.17),

$$\mathcal{M}(s) = \begin{bmatrix} \mathcal{W}^{(1)}(s)\mathcal{G}_f^{(1)}(s) & \cdots & \mathcal{W}^{(1)}(s)\mathcal{G}_f^{(q)}(s) \\ \vdots & \ddots & \vdots \\ \mathcal{W}^{(p)}(s)\mathcal{G}_f^{(1)}(s) & \cdots & \mathcal{W}^{(p)}(s)\mathcal{G}_f^{(q)}(s) \end{bmatrix}$$

(4.21)

Hence, we denote the $i$-th row of $\mathcal{M}(s)$ to be:

$$\mathcal{M}^{(i)}(s) = \begin{bmatrix} \mathcal{W}^{(i)}(s)\mathcal{G}_f^{(1)}(s) & \cdots & \mathcal{W}^{(i)}(s)\mathcal{G}_f^{(q)}(s) \end{bmatrix}$$

(4.22)

To construct $\mathcal{W}(s)$, the algorithm starts with focusing on creating the zeros at the selected indices in the 1-st row of $\mathcal{M}(s)$ and repeat itself by incrementing the row index, $i$ until one reaches $i = p$. The indices of the zeros of every row in $\mathcal{M}(s)$ are first selected by the designer. Then, based on those indices, the corresponding row of $\mathcal{W}(s)$ is computed.

We start with defining the design variables. Let $k_i$ be the number of zeros in $\mathcal{M}^{(i)}(s)$. Recall that $q$ is the number of columns of $\mathcal{G}_f(s)$. Then, denote $\gamma^i_j \in \{1, \cdots, q\}$, where
\{1, \cdots, k_i\}, to be one of the indices of \(M^{(i)}(s)\), i.e. \(M^{(i)}\gamma^j_i(s) = 0\). The term \(M^{(i)}\gamma^j_i(s)\) refers to the \((i, \gamma^j_i)\) entry of \(M(s)\). Thus, the set \(\mathcal{G}^i = \{\gamma^1_i, \cdots, \gamma^j_i\}\) represents the indices (or locations) of the zeros in \(M^{(i)}(s)\). It will be shown that one cannot arbitrarily choose the number of zeros in a row of \(M(s)\).

To create the zeros in the indices specified by \(\mathcal{G}^i\), construct \(W^{(i)}(s)\) as follows. First, create the following matrix:

\[
\hat{G}^i_j(s) = \begin{bmatrix} G^i_1(s) & \cdots & G^i_{k_i}(s) \end{bmatrix}
\]  

(4.23)

where \(\hat{G}^i_j(s) \in \mathbb{C}^{p \times k_i}\). If \((\hat{G}^i_j(s))^T\) has a non-trivial null space, then one can simply choose \((W^{(i)}(s))^T \in \text{Ker} \left((\hat{G}^i_j(s))^T\right)\) to create zeros in the indexes specified by \(\mathcal{G}^i\). By referring to (4.22), an intuitive explanation is \(W^{(i)}(s)\) is chosen to “kill off” the columns in \(\hat{G}^i_j\) specified by \(\mathcal{G}^i\) to result in the appropriate \(M^{(i)}(s)\). Since this concept will be used often in later development of our proposed methodology, we state this result as a lemma.

**Lemma 4.1.** Suppose \(\mathcal{G}^i\) is given. If \(\hat{G}^i_j(s)\) has a non-trivial null space, then there exist \(W^{(i)}(s)\) to create zeros in the selected entries of \(M^{(i)}(s)\) specified by \(\mathcal{G}^i\). This can be achieved by selecting \((W^{(i)}(s))^T \in \text{Ker} \left((\hat{G}^i_j(s))^T\right)\).

**Proof.** First note that the indices of the zeros in \(M^{(i)}(s)\) and its transpose, \((M^{(i)}(s))^T\) are the same because they are just vectors. Note that:

\[
(M^{(i)}(s))^T = \begin{bmatrix} (G^i_1(s))^T \\ \vdots \\ (G^i_{k_i}(s))^T \end{bmatrix} (W^{(i)}(s))^T
\]

Now construct \(\hat{G}^i_j(s)\) shown in (4.23). It is obvious that \(\text{Im}(\hat{G}^i_j(s)) \subset \text{Im}(G^i(s))\). Now, assume that \((\hat{G}^i_j(s))^T\) has a non-trivial null space and we choose \((W^{(i)}(s))^T \in \text{Ker} \left((\hat{G}^i_j(s))^T\right)\). It follows directly that the elements of \(M^{(i)}(s)\) specified by the indices in \(\mathcal{G}^i\) would be 0. \(\square\)

Using the algorithm provided in Lemma 4.1 we can answer our original question: are there some conditions on the numbers of zeros of \(M(s)\) that would guarantee the existence of a solution to \(W(s)\)? This question is answered by the following theorem.

**Theorem 4.2.** If the number of zeros in each row of \(M(s)\) is less than or equal to \(p - 1\), then there exist non-trivial solutions for \(W(s)\) (i.e. \(W(s) \neq 0\)) using the algorithm presented in Lemma 4.1. Also, all the rows of \(W(s)\) are nonzero row vectors.
Proof. Suppose that $\mathcal{M}^{(i)}(s)$ has $k_i$ zeros, where $k_i \leq p - 1$. Also, assume that this holds true for every row of $\mathcal{M}(s)$. By the rank-nullity theorem,

$$\dim \left( \text{Ker} \left( (\hat{G}_i^i(s))^T \right) \right) = p - \text{rank} \left( (\hat{G}_i^i(s))^T \right)$$

Since $k_i \leq p - 1$, it is obvious that $\text{rank} \left( (\hat{G}_i^i(s))^T \right) \leq p - 1$. This results in:

$$\dim \left( \text{Ker} \left( (\hat{G}_i^i(s))^T \right) \right) \geq 1$$

In other words, $(\hat{G}_i^i(s))^T$ always has a nontrivial nullspace. Thus, if $k_i \leq p - 1$, then Lemma 4.1 always holds and one could use the algorithm to construct $\mathcal{W}(s)$.

Use the algorithm to construct $\mathcal{W}^{(1)}(s)$ to $\mathcal{W}^{(p)}(s)$. Since $(\hat{G}_i^i(s))^T$ has a non-trivial nullspace, one would have constructed $\mathcal{W}(s)$ with none of its rows to be 0. Hence, if every row in $\mathcal{M}(s)$ has less than or equal to $p - 1$ zeros, then $\mathcal{W}(s)$ is a non-trivial solution that will result in the desired $\mathcal{M}(s)$.

A condition similar to Theorem 4.2 was provided in [33]. However, it is unclear whether the condition was a sufficient or necessary condition. Also, to our best knowledge, no explicit proof can be found in the literature. It should be emphasized that the work in [33] does not use the concept of nullspace to construct $\mathcal{W}(s)$. In our work, by constructing $\mathcal{W}(s)$ using the notion of nullspace, it was found that this condition is just a sufficient condition and the proof can be proven using simple linear algebra.

Selection of the structure for $\mathcal{M}(s)$

We will now return to finding a feasible structure for $\mathcal{M}(s)$ to identify the fault location for the colocated LFSS. In our case, Theorem 4.2 will be used to evaluate the feasibility of the structure presented by (4.19) and (4.20).

The number of zeros in every row of $\mathcal{M}(s)$ described by (4.19) is $2(m - 1)$. For Theorem 4.2 to be applicable (take $p = m$ and $q = 2m$), every row must have no more than $m - 1$ zeros. Clearly, the number of zeros required in (4.19) is more than the limit presented by Theorem 4.2. Thus, there is no guarantee that there exists a $\mathcal{W}(s)$ to create this particular structure.

On the other hand, the number of zeros in every row of $\mathcal{M}(s)$ described by (4.20) is always 2 regardless of the number of sensors/actuators. It is intuitive that there are certain cases that one can use Theorem 4.2 to guarantee the existence of $\mathcal{W}(s)$. The following result (a straightforward corollary from Theorem 4.2) gives a sufficient condition for the existence of $\mathcal{W}(s)$.
Corollary 4.2. If the number of inputs (or outputs), \( m \geq 3, m \in \mathbb{Z}^+ \), then there exist non-trivial solutions for \( \mathcal{W}(s) \) with nonzero rows such that \( \mathcal{M}(s) \) defined by (4.17) has the structure defined by (4.20) (however the ‘*’ maybe 0).

Corollary 4.2 says that there always exist solutions to \( \mathcal{W}(s) \) if a LFSS has sufficient number of actuator/sensor pairs. In practice, it would be highly unlikely that a LFSS would have less than 3 pairs of actuator and sensor. Thus, assume that the LFSS under consideration has at least 3 sensor-actuator pairs such that a solution for \( \mathcal{W}(s) \) always exist. Under this assumption, the diagonals of zeros can always be created and \( \mathcal{M}(s) \) is chosen to taken on the structure given by (4.20). Note that no claim has yet to be made on whether the ‘*’ elements are nonzeros. This will be investigated in Section 4.3.4.

### 4.3.4 Fault Isolation using SRG

In this subsection, we will investigate issues related to fault isolation using structured residuals. First, we summarize the assumptions required by the structured residuals, \( \tilde{r} \). Next, the fault isolability conditions of \( \mathcal{M}(s) \) are studied. Finally, these conditions are re-examined using \( \mathcal{W}(s) \) and \( \mathcal{G}_f(s) \).

We begin by introducing the assumptions required by our scheme.

**Assumption 4.2.** In order to perform fault isolation, the following assumptions have been made:

(a) Only single fault occurrences are allowed within any given time frame.

(b) Every single fault occurrence can be detected. Hence, we know there is a need to isolate faults.

(c) There are at least 3 sensor-actuator pairs. By Corollary 4.2, one can always find \( \mathcal{W}(s) \) to obtain \( \mathcal{M}(s) \) with a structure shown in (4.20). However, some of the ‘*’ maybe 0.

Denote \( \mathcal{L}^k = \text{Im}(I^k) \) where \( I^k \in \mathbb{R}^{2m} \) is the \( k \)-th column vector of \( 2m \times 2m \) identity matrix. Also, define \( \mathcal{F}_c(s) = [(\mathcal{F}_a(s))^T (\mathcal{F}_s(s))^T]^T \). Mathematically, Assumption 4.2(a) implies that, for \( s \in \mathbb{C} \):

\[
\mathcal{F}_c(s) \in \bigcup_{k=1}^{2m} \mathcal{L}^k
\]  

(4.24)

When a single fault occurs in the \( i \)-th location, one can represent it as \( \mathcal{F}_c(s) \in \mathcal{L}^i \cup \mathcal{L}^{i+m} \). Assumption 4.2(a) is required because it is possible for the occurrence of multiple faults to produce an unwanted zero signal in one of the elements of \( \tilde{r} \). The reasoning is as follows.
Every row of $\mathcal{M}(s)$ has a nullspace. When $\mathcal{F}_c(s)$ belongs to nullspaces of 2 or more rows of $\mathcal{M}(s)$, this may lead to unwanted zero signals in the elements of $\bar{r}$. Therefore, for simplicity of developing our theory, we assume only single fault occurrences are allowed within some given time window. According to Assumption 4.2(b), any $i$-th location fault, where $i \in \{1, ..., 2m\}$, would result in $\mathcal{R}(s) \neq 0$. If $\mathcal{R}(s) = 0$, then it is obvious that a fault will make $\bar{\mathcal{R}}(s) = 0$ using the relationship of $\bar{\mathcal{R}}(s) = \mathcal{W}(s)\mathcal{R}(s)$. Hence, when a fault is not detected, the fault cannot be isolated. Conditions for fault detectability are given in Section 4.2.

Assume that Assumption 4.2(c) holds. In subsection 4.3.3, a naive algorithm to generate 0 in a row $\mathcal{M}(s)$ was presented in Lemma 4.1. In order to generate the zeros as presented in (4.20), one can simply choose $\hat{G}_f^i(s)$ to be:

$$\hat{G}_f^i(s) = \left[ G_f^i(s) \ G_f^{i+m}(s) \right]$$

(4.25)

When Assumption 4.2(c) holds, one can only guarantee zeros to be created on the specified locations (i.e. the left and right diagonals). There is no guarantee that the ‘*’ are nonzeros. If the ‘*’ were zeros, that would mean a fault in the $k$-th location, $f_k^i \neq 0$ or $f_k^i \neq 0$ would result in $\bar{r}^i = 0$, where $i \neq k$. In other words, false identification of faults may happen.

The major focus of this subsection is the study of fault isolability of a single fault occurrence. Since we only consider the use of SRG to isolate faults, our definition of fault isolability only relates to whether we can use the SRG to isolate faults. Let $i \in \{1, 2, ..., m\}$.

**Definition 4.3.** A single fault has occurred in the $i$-th location if $\mathcal{F}_c(s) \in \mathcal{L}^i \cup \mathcal{L}^{i+m}$.

The definition of single fault isolability of a location using the SRG is given below.

**Definition 4.4.** A single fault in the $i$-th location is said to be isolable if there exists $\mathcal{M}^{(i)}(s)$ such that $\mathcal{F}_c(s) \notin \mathcal{L}^i \cup \mathcal{L}^{i+m}$ is equivalent to $\bar{\mathcal{R}}^i(s) \neq 0$.

Note that the statement of “$\mathcal{F}_c(s) \notin \mathcal{L}^i \cup \mathcal{L}^{i+m}$ is equivalent to $\bar{\mathcal{R}}^i(s) \neq 0$” is the same as “$\mathcal{F}_c(s) \in \mathcal{L}^i \cup \mathcal{L}^{i+m}$ is equivalent to $\bar{\mathcal{R}}^i(s) = 0$”.

The isolability of a single fault is closely related to the structure of $\mathcal{M}(s)$. First, a lemma that relates single fault isolability to $\mathcal{M}(s)$ is presented.

**Lemma 4.2.** Assume that Assumption 4.2 holds and $\mathcal{M}(s)$ takes on the form of (4.20) with no assumptions made on ‘*’. Then, a fault in the $i$-th location is isolable if and only if

$$\text{Ker}(\mathcal{M}^{(i)}(s)) \cap \left[ \bigcup_{k \in \mathfrak{B}_i} \mathcal{L}^k \right] = \{0\}$$

(4.26)

where $\mathfrak{B}_i = \{1, ..., 2m\} - \{i, i + m\}$. 


Proof. Assumption 4.2(b) implies that faults can always be detected and hence fault isolation can be performed. Assumption 4.2(c) guarantees the existence of $\mathcal{W}(s)$ for our choice of $\mathcal{M}(s)$. From (4.18), observe that:

$$\bar{R}_i(s) = \mathcal{M}^{(i)}(s)F_c(s)$$

(4.27)

Also let:

$$\mathcal{P}^i = \bigcup_{k \in \mathcal{B}_i} \mathcal{L}^k$$

(4.28)

First, we prove sufficiency. Assume that (4.26) holds. Take $F_c(s) \notin \mathcal{L}^i \cup \mathcal{L}^{i+m}$. Using Assumption 4.2(a) or equivalently using (4.24), the statement of $F_c(s) \notin \mathcal{L}^i \cup \mathcal{L}^{i+m}$ is equivalent to $F_c(s) \in \mathcal{P}^i$. Since (4.26) holds, then $F_c(s) \notin \text{Ker}(\mathcal{M}^{(i)}(s))$. Using (4.27), it follows that $\bar{R}_i(s) \neq 0$ is equivalent to $F_c(s) \notin \mathcal{L}^i \cup \mathcal{L}^{i+m}$. Note that this final statement is the definition of a fault in the $i$-th location being isolable.

Next, necessity will be proven. Assume that a fault in the $i$-th location is isolable, i.e., $\bar{R}_i(s) \neq 0$ is equivalent to $F_c(s) \notin \mathcal{L}^i \cup \mathcal{L}^{i+m}$. Once again, by Assumption 4.2(a), $F_c(s) \in \mathcal{P}^i$. We will use proof by contradiction to arrive at the final result.

Assume the following is true: $\bar{R}_i(s) \neq 0 \Leftrightarrow F_c(s) \in \mathcal{P}^i$ but $\text{Ker}(\mathcal{M}^{(i)}(s)) \cap \mathcal{P}^i \neq \{0\}$. Since $\text{Ker}(\mathcal{M}^{(i)}(s)) \cap \mathcal{P}^i$ is non-empty, one can take $F_c(s)$ to be strictly nonzero vector that lies within that subspace. This implies $F_c(s) \in \text{Ker}(\mathcal{M}^{(i)}(s))$. Subsequently, using (4.27), this results in $\bar{R}_i(s) = 0$ when $F_c(s) \in \mathcal{P}^i$, which contradicts with $\bar{R}_i(s) \neq 0$. Hence, necessary condition of Lemma 4.2 must be true. 

Lemma 4.2 presents a geometrical condition to determine if a fault in a certain location is isolable by the designed SRG. Geometrical condition is often difficult to check. Since $\mathcal{L}^k$ are subspaces spanned by standard basis vectors, it presents additional structure to $\mathcal{M}(s)$ if one wants the SRG to isolate single fault occurrence in every location. Lemma 4.3 shows that the "*' shown in (4.20) has to be strictly nonzero in order to isolate single fault occurrence.

Lemma 4.3. Assume that Assumption 4.2 holds and $\mathcal{M}(s)$ takes on the form of (4.20). Then, a fault in the $i$-th location is isolable if and only if $\mathcal{M}^{ik}(s) \neq 0$ for all $k \in \mathcal{B}_i$, where $\mathcal{M}^{ik}(s) \in \mathcal{C}$ is the $(i, k)$ entry of $\mathcal{M}(s)$.

Proof. Using Assumption 4.2(a), we have $F_c(s) \in \mathcal{L}^k$ or $F_c(s) = I^k F_c^k(s)$, where $F_c^k(s) \in \mathcal{C}$ is the $k$ element of $F_c(s)$. By substituting this relationship into (4.27), one obtains:

$$\bar{R}_i(s) = \mathcal{M}^{ik}(s)F_c^k(s)$$

where $\mathcal{M}^{ik}(s) = \mathcal{M}^{(i)}(s)I^k$. 


Using Lemma 4.2, the condition of a fault in the \( i \)-th location is isolable is given by (4.26), which can be further simplified as follows:

\[
\bigcup_{k \in \mathcal{B}_i} \left[ \text{Ker}(\mathcal{M}^{(i)}(s)) \cap \mathcal{L}^k \right] = \{0\}
\]

The above condition is simply equivalent to \( \text{Ker}(\mathcal{M}^{(i)}(s)) \cap \mathcal{L}^k = \{0\} \) for all \( k \in \mathcal{B}_i \). Therefore, to show that a single fault in the \( i \)-th location is isolable, we just have to show that, for all \( k \in \mathcal{B}_i \), \( \text{Ker}(\mathcal{M}^{(i)}(s)) \cap \mathcal{L}^k = \{0\} \) if and only if \( M^{ik}(s) \neq 0 \).

The necessary condition of this lemma is simple to prove. Assume \( \text{Ker}(\mathcal{M}^{(i)}(s)) \cap \mathcal{L}^k = \{0\} \) for all \( k \in \mathcal{B}_i \). Then, it follows that \( M^{ik}(s) = \mathcal{M}^{(i)}(s) I^k \neq 0 \) for all \( k \in \mathcal{B}_i \). On the other hand, the sufficient condition of this corollary can be shown using proof by contradiction. The argument is similar to the proof by contradiction in Lemma 4.2 and it is omitted.

Lemma 4.3 provides a good way to check whether the constructed \( \mathcal{M}(s) \) can isolate single fault. A naive approach is to use (4.25) and the algorithm presented in Lemma 4.1 to construct \( \mathcal{M}(s) \) iteratively until the condition proposed by Lemma 4.3 is satisfied. Such a scheme can be inefficient because symbolic computation is very slow and there might not exist a solution for \( \mathcal{W}(s) \). This motivates us to construct \( \mathcal{M}(s) \) that has every row satisfying Lemma 4.3 using just one iteration. Furthermore, it would be wise to look for a condition to check for the existence of \( \mathcal{W}(s) \) based on \( \mathcal{G}_f(s) \) before even designing the SRG.

We present the following theorem that summarizes the relationship of single fault isolability and the structure of \( \mathcal{M}(s), \mathcal{W}(s) \) and \( \mathcal{G}_f(s) \).

**Theorem 4.3.** Assume that Assumption 4.2 holds and \( \mathcal{M}(s) \) takes on the form of (4.20). Recall that \( \mathcal{B}_i = \{1, ..., 2m\} - \{i, i + m\} \). Each of the following statements is equivalent:

(a) A fault in the \( i \)-th location is isolable on the colocated LFSS.

(b) The nullspace of \( \mathcal{M}^{(i)}(s) \) satisfies:

\[
\text{Ker}(\mathcal{M}^{(i)}(s)) \cap \left[ \bigcup_{k \in \mathcal{B}_i} \mathcal{L}^k \right] = \{0\}
\]

(c) The individual entries, \( M^{ik}(s) \neq 0 \) for all \( k \in \mathcal{B}_i \).

(d) The \( i \)-th row of \( \mathcal{W}(s) \), \( \mathcal{W}^{(i)}(s) \) satisfies:

\[
(\mathcal{W}^{(i)}(s))^T \in \text{Ker} \left( (\mathcal{G}_f^{(i)}(s))^T \right)
\]

(4.29)
and, for all \( k \in \mathcal{B}_i \),

\[
(W^{(i)}(s))^T \not\in \text{Ker} \left( (G^k_f(s))^T \right) \tag{4.30}
\]

where \( \hat{G}^i_f(s) \) is defined by [4.25].

(e) For all \( k \in \mathcal{B}_i \), every columns of \( G_f(s) \) in that set satisfies:

\[
\text{Im}(\hat{G}^i_f(s)) \cap \text{Im}(G^k_f(s)) = \{0\} \tag{4.31}
\]

Or, equivalently, for all \( k \in \mathcal{B}_i \):

\[
\text{rank}([\hat{G}^i_f(s) \ G^k_f(s)]) = \text{rank}(\hat{G}^i_f(s)) + 1 \tag{4.32}
\]

Proof. Part (b) and (c) of this theorem is taken directly from Lemma [4.2] and Lemma [4.3]. To prove part (d), consider the transpose of (4.17):

\[
(M^{(i)}(s))^T = (G_f(s))^T (W^{(i)}(s))^T
\]

To design the \( i \)-th element of \( \tilde{r} \) to not respond to either the \( i \)-th actuator or sensor fault, we have chosen \( M(s) \) to have the structure shown in (4.20). This implies \( M^{ii}(s) = M^{i(i+m)}(s) = 0 \), which is equivalent to the condition stated in (4.29). Using the statement from part (c), one can see that it is equivalent to \( (W^{(i)}(s))^T \not\in \text{Ker} \left( (G^k_f(s))^T \right) \) for all \( k \in \mathcal{B}_i \). Hence, part (d) proven.

To prove part (e), consider the statement in part (d). Using the definition of a subset, the statement is equivalent to, for all \( k \in \mathcal{B}_i \), \( \text{Ker} \left( (\hat{G}^i_f(s))^T \right) \not\subset \text{Ker} \left( (G^k_f(s))^T \right) \). We would first like to show that \( \text{Ker} \left( (\hat{G}^i_f(s))^T \right) \not\subset \text{Ker} \left( (G^k_f(s))^T \right) \) if and only if \( \text{Ker} \left( (\hat{G}^i_f(s))^T \right) + \text{Ker} \left( (G^k_f(s))^T \right) = \mathbb{C}^{2m} \).

The necessity part of the above statement can be proven as follows. Since every single fault is detectable, there are no columns of \( G_f(s) \) that are 0. None of the columns of \( G_f(s) \) being 0 implies \( \text{dim} \left( \text{Ker} \left( (G^k_f(s))^T \right) \right) = 2m - 1 \). The fact that \( \text{Ker} \left( (\hat{G}^i_f(s))^T \right) \not\subset \text{Ker} \left( (G^k_f(s))^T \right) \) means that there exists at least one basis vector from \( \text{Ker} \left( (\hat{G}^i_f(s))^T \right) \) that does not lie in \( \text{Ker} \left( (G^k_f(s))^T \right) \). Thus, \( \text{Ker} \left( (\hat{G}^i_f(s))^T \right) + \text{Ker} \left( (G^k_f(s))^T \right) = \mathbb{C}^{2m} \). The sufficiency part can be shown using proof by contradiction and is straightforward.

Now, we introduce the orthogonal complement to complete the proof. Let \( \perp \) denote the orthogonal complement of a linear subspace. Let \( a \) be some column vector. It is well known that \( (\text{Ker}(a^T))^\perp = \text{Im}(a) \). Also, let \( U \) and \( V \) be some linear subspace. It can be shown that \( (U + V)^\perp = U^\perp \cap V^\perp \). Using these two identities, \( \text{Ker} \left( (\hat{G}^i_f(s))^T \right) + \text{Ker} \left( (G^k_f(s))^T \right) = \mathbb{C}^{2m} \) is equivalent to (4.31). Finally, following a similar proof to Corollary [4.1], one can conclude
from (4.31) and obtain the expression in (4.32). Thus, part (e) proven.

As mentioned previously, Part (c) of Theorem 4.3 can be used to check if the SRG, $M(s)$ has been designed properly. Theorem 4.3(e) allows one to check whether it is possible to construct the FIF, $W(s)$ that is capable of isolating single fault at every location. Part (d) of the above theorem provides one with the algorithm to construct the FIF which guarantees fault isolability in every location. $W(s)$ can be constructed in a row-by-row fashion using Theorem 4.3(d), which can be performed by a symbolic computation software.

### 4.4 Fault Isolation Filter (FIF)

In Section 4.3.2, we mentioned that it is important to have a stable fault isolation filter (FIF). Also, the FIF has to be implemented without the use of differentiator. These two issues will be addressed in this section. Finally, the construction of the FIF is summarized in this section.

#### 4.4.1 Stability and Practicality of FIF

Recall that in subsection 4.3.2, the SRG has to not only isolate fault, but to be also bounded-input bounded-output (BIBO) stable. From here onwards, we will simply refer BIBO stable as stable in future discussion of this chapter. Furthermore, we would also like to have a $W(s)$ that is implementable without the use of any differentiator.

The core ideas that are presented in this subsection are the same as the work performed in [31]. The author, Gertler presented 2 algorithms to construct a stable $W(s)$ and one method to make the entries of $W(s)$ to be proper rational. However, no formal proofs are provided in that paper. To fill this gap, we present the proofs to our proposed algorithm, which creates a stable $W(s)$ with its entries to be proper rational.

The SRG, $M(s)$ is constructed by concatenating the ORG, $G_f(s)$ and the FIF, $W(s)$. This relationship is represented by (4.17), which implies if all the elements of $G_f(s)$ and $W(s)$ have stable poles, then $M(s)$ would be stable. By (4.16), the poles of all the elements in $G_f(s)$ is solely dependent on the term $A - HC$, which is designed to have stable poles. Hence, all elements of $G_f(s)$ is always stable. Thus, $M(s)$ is unstable if and only if $W(s)$ is unstable.

Since the FIF, $W(s)$ is constructed in a row-by-row fashion, it is important to study the stability of each row of $W(s)$ while isolating single fault occurrence in a location. It turns out that there is no loss of generality of assuming there always exists a stable $W^{(i)}(s)$ if the $i$-th location’s fault is isolable.

**Lemma 4.4.** If single fault occurrence in $i$-th location can be isolated, then there always exist $W^{(i)}(s)$ in which all of its elements are stable.
Chapter 4. Fast Fault Diagnosis and Fault Tolerant Control for LFSS

Proof. Assume single fault occurrence in \( i \)-th location can be isolated. Thus, one can always construct \( \tilde{W}^{(i)}(s) \) that satisfies (4.29) and (4.30). In our case, the symbolic computation software can provide a \( \tilde{W}^{(i)}(s) \) that satisfies both condition. Symbolic computation does not necessarily produce a \( \tilde{W}^{(i)}(s) \) in which all of its elements are stable.

Thus, assume that the computation returns a \( \tilde{W}^{(i)}(s) \) that is unstable. With no loss of generality, one can rewrite:

\[
\tilde{W}^{(i)}(s) = \begin{bmatrix} \tilde{W}^{i1}(s) & \ldots & \tilde{W}^{im}(s) \end{bmatrix} = \begin{bmatrix} \tilde{W}^{i1}(s)/\tilde{w}^{i1}(s) & \ldots & \tilde{W}^{im}(s)/\tilde{w}^{im}(s) \end{bmatrix}
\]

where \( \tilde{W}^{ij}(s) \) is the stable transfer function of the \( j \)-th entry of \( \tilde{W}^{(i)}(s) \), \( \tilde{w}^{ij}(s) \) is the polynomial with roots that are the unstable poles of \( \tilde{W}^{ij}(s) \) and \( j \in \{1,\ldots,m\} \). When there are no unstable poles in \( \tilde{W}^{ij}(s) \), then \( \tilde{w}^{ij}(s) = 1 \).

Take \( \tilde{W}^{(i)}(s) = \tilde{w}(s)\tilde{W}^{(i)}(s) \), where:

\[
\tilde{w}(s) = \prod_{j=1}^{m} \tilde{w}^{ij}(s)
\]

This implies that all the unstable poles in each element will be cancelled out. Thus, \( \tilde{W}^{(i)}(s) \) is stable. Furthermore, note that \( \tilde{w}(s) \) is just a scaling factor of \( \tilde{W}^{(i)}(s) \) and \( \tilde{W}^{(i)}(s) \) satisfies both (4.29) and (4.30). Hence, \( \tilde{W}^{(i)}(s) \) must also satisfies (4.29) and (4.30).

Lemma 4.4 provided a simple method to generate a stable \( W(s) \). However, there might be elements of \( W(s) \) that are not rational functions. This presents an implementation problem, namely one will require the use of differentiator to implement the irrational functions. One can never implement a perfect differentiator in real life and differentiators are known to amplify high frequency noises. Thus, it is not practical to use a differentiator for implementation. We will show that one can also always construct a \( W(s) \) that is stable and does not require the use of differentiator in the following theorem.

Theorem 4.4. If single fault occurrence in \( i \)-th location can be isolated, then there always exist \( W^{(i)}(s) \) in which all of its entries are stable and they are proper rational functions.

Proof. Suppose that single fault occurrence in \( i \)-th location can be isolated. By Lemma 4.4 it is obvious that one can always construct \( W^{(i)}(s) \) with all of is elements being stable. Hence, let assume we can always construct a \( \tilde{W}^{(i)}(s) \) which has all its elements (i.e. the transfer functions) to be stable but not proper rational functions. Also, \( \tilde{W}^{(i)}(s) \) is chosen to satisfy (4.29) and (4.30).
We will show that one can construct $\mathcal{W}^{(i)}(s)$ using $\bar{\mathcal{W}}^{(i)}(s)$ such that all its elements are stable and they are proper rational functions. Define $\mathcal{W}^{ij}(s)$ to be the $j$-th entry of $\mathcal{W}^{(i)}(s)$, $k_n^{ij}$ to be the number of zeros of $\mathcal{W}^{ij}(s)$ and $k_d^{ij}$ to be the number of poles of $\mathcal{W}^{ij}(s)$. For $\mathcal{W}^{ij}(s)$ to be a proper rational, one will require that $k_n^{ij} \leq k_d^{ij}$. This can be achieved by constructing a polynomial function, $b(s)$ that satisfy the following two properties:

1. Define $\bar{k}_n^{ij}$ and $\bar{k}_d^{ij}$ to be the number of zeros and poles of $\bar{\mathcal{W}}^{(i)}(s)$ respectively. $b(s)$ has $k_b$ roots, where:
   \[
   k_b = \max_{j \in \{1, ..., m\}} (\bar{k}_n^{ij} - \bar{k}_d^{ij}) \text{ subject to } k_b \geq 0 \quad (4.33)
   \]
   One can view $k_b$ as the minimum number of extra poles required to make all the elements of $\bar{\mathcal{W}}^{(i)}(s)$ to be proper rational functions. Note that this also implies $k_b \geq \bar{k}_n^{ij} - \bar{k}_d^{ij}$ for all $j \in \{1, ..., m\}$.

2. All the roots of $b(s)$ must be on the open left half plane of $\mathbb{C}$.

To construct a $\mathcal{W}^{(i)}(s)$ in which all its elements are stable and proper rational, construct $b(s)$ and take:

\[
\mathcal{W}^{(i)}(s) = \frac{1}{b(s)} \bar{\mathcal{W}}^{(i)}(s)
\]

Note that all the transfer functions in $\mathcal{W}^{(i)}(s)$ are stable because all the elements of $\bar{\mathcal{W}}^{(i)}(s)$ are stable and the roots of $b(s)$ are in the open left half plane of $\mathbb{C}$. Next, note that $\mathcal{W}^{ij}(s)$, has $k_n^{ij} = \bar{k}_n^{ij}$ zeros and $k_d^{ij} = \bar{k}_d^{ij} + k_b$ poles. Using (4.33), it is clear that $\bar{k}_n^{ij} + k_b \geq \bar{k}_n^{ij}$, or equivalently, $k_d^{ij} \geq k_n^{ij}$. Hence, every element of $\mathcal{W}^{(i)}(s)$ is a proper rational function. Finally, since $\frac{1}{b(s)}$ is just a scaling factor and $\bar{\mathcal{W}}^{(i)}(s)$ is chosen to satisfy (4.29) and (4.30), then $\mathcal{W}^{(i)}(s)$ would also satisfy (4.29) and (4.30). Therefore, it can be used to isolate a single fault occurrence in the $i$-th location.

With the introduction of Theorem 4.3 and 4.4, one can finally construct a stable, implementable FIF that can isolate single fault occurrence. In the next section, we will show the algorithm to design $\mathcal{W}(s)$ explicitly and how it relates to the theorems that have been shown previously.

### 4.4.2 Construction of FIF

The algorithm to construct $\mathcal{W}(s)$ is given in this subsection. It serves to summarize the results shown in Section 4.3.4 and 4.4.1. Prior to showing the algorithm, recall that the following assumptions have been made:

1. The LFSS is observable.
2. Only a single fault can occur.

3. Every single fault occurrence in the LFSS is detectable.

4. The LFSS has at least 3 sensor-actuator pairs.

Since LFSSs contain a large amount of states, computation of $W(s)$ by hand is practically impossible. The methods proposed in [33]-[34] require one to choose certain entries of $M(s)$ to be certain transfer function manually. Therefore, such methods are very inefficient when one applies them to LFSSs. Our algorithm relies heavily on (4.29), (4.30) and Theorem 4.4 to enable symbolic computation software to compute $W(s)$ without the need of user to choose certain elements of $M(s)$ manually. The algorithm to construct $W(s)$ is given as follows:

1. Construct the ORG. Select all the poles of $A - HC$ to be in the open left hand plane.

2. Compute the transfer matrix, $G_f(s)$.

3. Check the fault isolability condition proposed in (4.32) for all $i \in \{1, \ldots, m\}$, where $\hat{G}_f(s)$ is given in (4.25). If (4.32) holds for all $i \in \{1, \ldots, m\}$, then proceed to the next step. Else, stop.

4. Compute $V_i(s) = \text{Ker} \left( (\hat{G}_f(s))^T \right)$ for all $i \in \{1, \ldots, m\}$.

5. Set $i = 1$.

6. Take $(\hat{W}^{(i)}(s))^T$ to be some linear combination of the basis vectors which span $V_i(s)$. (Here we assume a basis can be constructed.) Check that $\hat{W}^{(i)}(s)G^k_f(s) \neq 0$ for all $k \in B_i$, where $B_i = \{1, \ldots, 2m\} - \{i, i + m\}$. If $\hat{W}^{(i)}(s)G^k_f(s) = 0$, then repeat this step again. There should exist a $\bar{W}^{(i)}(s)$ such that $\bar{W}^{(i)}(s)G^k_f(s) \neq 0$ since part (e) of Theorem 4.3 holds.

7. Use the obtained $\bar{W}^{(i)}(s)$ to construct a row vector $\bar{W}^{(i)}(s)$ that has all its elements to be stable using the method introduced in the proof of Lemma 4.4.

8. Use $\bar{W}^{(i)}(s)$ to build the row vector $W^{(i)}(s)$ using the proposed method in the proof of Theorem 4.4. This will guarantee all the entries of $W^{(i)}(s)$ are stable and they are all proper rational function of $s$.

9. Repeat step 6 to 8 by incrementing $i$. Stop when $i > m$ because we have finished constructing $W(s)$. 
From the above algorithm, it is clear that the construction of \( \mathcal{W}(s) \) is relatively straightforward. However, transfer matrices cannot be implemented directly into a computer. Therefore, one would like to represent \( \mathcal{W}(s) \) using state space representation. We will briefly discuss some issues related to the implementation of the FIF construction algorithm in MATLAB. In order to preserve maximal numerical accuracy during the conversion from transfer matrix (symbolic data type) to state space representation (numerical data type), we propose to represent \( \mathcal{W}(s) \) using the pole-zero-gain representation. In MATLAB, the pole-zero-gain representation can be constructed using the command \texttt{zpk()}. The conversion of the pole-zero-gain representation to the state space representation using MATLAB can be done simply using the command \texttt{ss()}

We illustrate the construction of the SRG and the FIF with the following example.

**Example 4.2.** Consider the following system. Take the system matrices to be the following:

\[
\Omega^2 = \Delta = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, L = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}
\]

This results in:

\[
A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & -1 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}
\]

This system is found to be observable, thus an ORG can be constructed. We choose the poles of the error system to be \( \{-1, -1, -1, -1, -1, -1\} \), which results in the following observer gain:

\[
H = \begin{bmatrix} 1 & -1 & 1 \\ 1 & 1 & -1 \\ -0.5 & 0.5 & 0.5 \\ 0.5 & -0.5 & 0.5 \\ 0.5 & 0.5 & -0.5 \\ 0.5 & -0.5 & -0.5 \end{bmatrix}
\]
This results in the following $G_f(s)$:

$$G_f(s) = \begin{bmatrix}
\frac{2}{(s+1)^2} & \frac{1}{(s+1)^2} & \frac{1}{(s+1)^2} & \frac{s^2}{2(s+1)} & 0 & 0 \\
\frac{1}{(s+1)^2} & \frac{2}{(s+1)^2} & \frac{1}{(s+1)^2} & \frac{2s^2+s+1}{2(s+1)^2} & 0 & \frac{1}{2(s+1)} \\
\frac{1}{(s+1)^2} & \frac{2}{(s+1)^2} & \frac{1}{(s+1)^2} & \frac{1}{2(s+1)} & \frac{2s^2+s+1}{2(s+1)^2} & 0 \\
\end{bmatrix}$$

Next, construct $\hat{G}_f^1$, $\hat{G}_f^2$ and $\hat{G}_f^3$. If one were to check the fault isolability condition given in (4.32), then one would see that $\hat{G}_f^1$, $\hat{G}_f^2$ and $\hat{G}_f^3$ all satisfies it. Hence, we know that a FIF can be constructed.

In this example, we show the construction of one row of $W(s)$, namely the third row, i.e. $W^{(3)}(s)$. If one were to compute $W^{(1)}(s)$ and $W^{(2)}(s)$, one would find that the resulting transfer functions of the basis vector are stable and rational. $W^{(3)}(s)$ is slightly more interesting.

To construct $W^{(3)}(s)$, we first construct $\hat{G}_f^3$, which is shown below.

$$\hat{G}_f^3(s) = \begin{bmatrix}
\frac{1}{(s+1)^2} \\
\frac{1}{(s+1)^2} \\
\frac{2}{(s+1)^2} \\
\end{bmatrix}$$

Using MAPLE, one can compute $\text{Ker}((\hat{G}_f^3(s))^T)$ and the result is given as follows:

$$\text{Ker}((\hat{G}_f^3(s))^T) = \text{span} \begin{bmatrix}
-\frac{s-1+2s^2}{(s+1)^2} \\
-\frac{2s^2+s+1}{(s+1)^2} \\
-1 \\
\end{bmatrix}$$

Note that the transfer functions are all stable. However, they are not proper rational functions. To make them proper rational, we choose to multiply the basis vector with a scaling factor of $1/(s+1)$. Hence, we take:

$$W^{(3)}(s) = \begin{bmatrix}
\frac{-s-1+2s^2}{(s+1)^2} & -\frac{2s^2+s+1}{(s+1)^2} & \frac{1}{(s+1)^2} \\
\end{bmatrix}$$

By using the same procedure to compute $W^{(1)}(s)$ and $W^{(2)}(s)$, $W(s)$ was found to be:

$$W(s) = \begin{bmatrix}
0 & -1 & 1 \\
-\frac{s-1+2s^2}{2s^2+s+1} & -\frac{s+1}{2s^2+s+1} & 1 \\
\frac{-s-1+2s^2}{(s+1)^2} & -\frac{2s^2+s+1}{(s+1)^2} & \frac{1}{(s+1)^2} \\
\end{bmatrix}$$
This results in $\mathcal{M}(s)$ taking on the structure:

$$
\mathcal{M}(s) = \begin{bmatrix}
0 & * & 0 & * \\
* & 0 & * & 0 \\
* & * & 0 & * \\
* & * & 0 & 0
\end{bmatrix}
$$

where ‘*’ represents some nonzero transfer function. The actual values in $\mathcal{M}(s)$ is omitted for the brevity of this example.

### 4.5 Integrating SRG with existing FTC technique

In this section, we would like to integrate the SRG with existing fault tolerant control technique presented in [38]. For more information on the fault tolerant control technique, refer to Section 2.4. By Definition 4.4, when a fault is isolable, we know that a single fault at the $i$-th location will produce a zero signal at the $i$-th residual. The Fault Tolerant Decentralized PID (FTD-PID) controller, given in (2.26) and (2.27), requires the knowledge of the combined failure matrix, $F = F_y F_u$ introduced in (2.8). Furthermore, the FTD-PID is designed to handle failure specifically.

Hence, to integrate the SRG with the FTD-PID controller and the decentralized PD controller, we will now assume that all faults that occur are failures, i.e. the complete malfunction of actuator or sensor. Currently, the SRG can only produce structured residuals. How they relate to $F$ remains unclear. We will assume we have prior knowledge of any existing failed locations. The goal is to study how a new unknown failed location can be isolated using the structured residual. Define $F_{kk}$ to be the $(k, k)$ entry of $F$, i.e. the $k$-th diagonal element of $F$. Let $\mathcal{D} = \{k : F_{kk} = 0\}$ and $\mathcal{J} = \{1, ..., m\} \setminus \{i\} \setminus \mathcal{D}$. The set $\mathcal{D}$ can be seen as the known failed locations prior to a new failure occurrence, whereas $\mathcal{J}$ is the remaining healthy location after a new failure is detected and the failed location has been isolated.

**Theorem 4.5.** Let Assumption 4.2 holds. Suppose that $\mathcal{D}$ is given and $i \in \{1, ..., m\} \setminus \mathcal{D}$. Also, enforce that $u^k$ and $y^k$ are not zero signal for all $k \in \{1, ..., m\} \setminus \mathcal{D}$, where $y = C x$ is the actual output of the system.

Assume that a single failure occurrence at the $i$-th location is isolable. Then, the $i$-th structured residual, $\bar{R}^i(s) = 0$ if and only if all of the following hold: (i) $F_{ii} = 0$, (ii) $F_{\ell \ell} = 0$, for all $\ell \in \mathcal{D}$, and (iii) $F_{jj} = 1$ for all $j \in \mathcal{J}$. 
Proof. By the assumption that only failure can occur, using (4.6) and (4.7), we have:

\[
\begin{align*}
    f_a &= (F_u - I)u \\
    f_s &= (F_y - I)Cx = (F_y - I)y
\end{align*}
\]

Since \( F_u \) and \( F_y \) are diagonal matrices, elements of \( f_a \) and \( f_s \) can be rewritten as:

\[
\begin{align*}
    f_a^c &= (F^c - 1)u^c \\
    f_s^c &= (F^c - 1)y^c
\end{align*}
\]

where \( c \in \{1, ..., m\} \), \( F^c_{cc} \) is the \((c,c)\) entry of \( F_u \) and \( F^c_{cc} \) is the \((c,c)\) entry of \( F_y \). Due to the fact that \( u^k \) and \( y^k \) are not zero signals for all \( k \in \{1, ..., m\} \setminus \mathcal{D} \) (i.e. by abuse of notation, \( u^k \neq 0 \) and \( y^k \neq 0 \)), one can conclude the following:

\[
\begin{align*}
    f_a^k \neq 0 &\iff F^k_{kk} = 0 \quad (4.34) \\
    f_s^k \neq 0 &\iff F^k_{kk} = 0 \quad (4.35)
\end{align*}
\]

Note that we do not need to consider the known failed locations since the failed locations will never recover.

By Definition 4.4, we know that if a single fault occurrence in the \( i \)-th location is isolable then:

\[
    \mathcal{F}_c(s) \in \mathcal{L}^i \cup \mathcal{L}^{i+m} \iff \mathcal{R}^i(s) = 0
\]

Hence, we would like to show:

\[
    \mathcal{F}_c(s) \in \mathcal{L}^i \cup \mathcal{L}^{i+m} \iff \left( F^{ii} = 0 \text{ AND } F^{\ell\ell} = 0 \text{ AND } F^{jj} = 1 \right) \quad (4.36)
\]

for all \( \ell \in \mathcal{D} \) and \( j \in \mathcal{J} \). We start with the left hand side of the expression. We have:

\[
    \mathcal{F}_c(s) \in \mathcal{L}^i \cup \mathcal{L}^{i+m} \iff \left( \mathcal{F}_c(s) \in \mathcal{L}^i \text{ OR } \mathcal{F}_c(s) \in \mathcal{L}^{i+m} \right) \quad (4.37)
\]

However, recall that \( \mathcal{F}_c(s) = [\mathcal{F}_a^T(s) \ \mathcal{F}_s^T(s)]^T \), therefore:

\[
\begin{align*}
    \mathcal{F}_c(s) \in \mathcal{L}^i \iff \left( \mathcal{F}_a(s) \in \text{Im}(I^i) \text{ AND } \mathcal{F}_s(s) = 0 \right) \\
    \mathcal{F}_c(s) \in \mathcal{L}^{i+m} \iff \left( \mathcal{F}_a(s) = 0 \text{ AND } \mathcal{F}_s(s) \in \text{Im}(I^i) \right)
\end{align*}
\]

where \( I^i \in \mathbb{R}^m \).

Using the fact that \( \left( \mathcal{F}_a(s) \in \text{Im}(I^i) \right) \text{ AND } \left( \mathcal{F}_s(s) = 0 \right) \), we know that \( f_a \in \text{Im}(I^i) \) and
\( f_s = 0 \). The expression \( f_a \in \text{Im}(I^i) \) is equivalent to the following:

\[
\begin{align*}
  f_a^i &= (F_{u}^{ii} - 1)u^i \neq 0 \\
  f_a^j &= (F_{u}^{jj} - 1)u^j = 0, \quad \forall j \in \mathcal{J}
\end{align*}
\]

Using (4.34), this implies \( F_{u}^{ii} = 0 \) and \( F_{u}^{jj} = 1 \), \( \forall j \in \mathcal{J} \). Also, using (4.35), one can conclude that \( f_s = 0 \) is equivalent to \( F_{y}^{kk} = 1 \) for all \( k \in \{1, ..., m\} \) \( - \mathcal{D} \).

Since \( F = F_y F_u \) and both \( F_y, F_u \) are diagonal matrices, it is clear that \( F^{kk} = F_y^{kk} F_u^{kk} \). By the definition of \( \mathcal{D} \), we know that \( F^{\ell\ell} = 0 \) for all \( \ell \in \mathcal{D} \). Using all these information, one can reach to the conclusion that:

\[
\left( \mathcal{F}_a(s) \in \text{Im}(I^i) \text{ AND } \mathcal{F}_s(s) = 0 \right) \Leftrightarrow \left( F^{ii} = 0 \text{ AND } F^{\ell\ell} = 0 \text{ AND } F^{jj} = 1 \right) \quad (4.38)
\]

for all \( \ell \in \mathcal{D} \) and \( j \in \mathcal{J} \).

Using a similar argument for \( \left( \mathcal{F}_a(s) = 0 \text{ AND } \mathcal{F}_s(s) \in \text{Im}(I^i) \right) \), one can arrive at \( F^{ii} = 0, F_y^{jj} = 1, \forall j \in \mathcal{J} \) and \( F^{uu} = I \). Thus, we claim that:

\[
\left( \mathcal{F}_a(s) = 0 \text{ AND } \mathcal{F}_s(s) \in \text{Im}(I^i) \right) \Leftrightarrow \left( F^{ii} = 0 \text{ AND } F^{\ell\ell} = 0 \text{ AND } F^{jj} = 1 \right) \quad (4.39)
\]

for all \( \ell \in \mathcal{D} \) and \( j \in \mathcal{J} \).

Using (4.38) and (4.39), we obtain:

\[
\begin{align*}
  \mathcal{F}_c(s) \in \mathcal{L}^i &\Leftrightarrow \left( F^{ii} = 0 \text{ AND } F^{\ell\ell} = 0 \text{ AND } F^{jj} = 1 \right) \\
  \mathcal{F}_c(s) \in \mathcal{L}^{i+m} &\Leftrightarrow \left( F^{ii} = 0 \text{ AND } F^{\ell\ell} = 0 \text{ AND } F^{jj} = 1 \right)
\end{align*}
\]

for all \( \ell \in \mathcal{D} \) and \( j \in \mathcal{J} \). Applying this result in (4.37), then (4.36) is shown to hold true. Finally, assume that single failure occurrence in the \( i \)-th location is isolable, then using (4.36), one arrive at the conclusion that:

\[
\mathcal{R}^i(s) = 0 \Leftrightarrow \left( F^{ii} = 0 \text{ AND } F^{\ell\ell} = 0 \text{ AND } F^{jj} = 1 \right)
\]

for all \( \ell \in \mathcal{D} \) and \( j \in \mathcal{J} \).

The assumption that \( u^k \) and \( y^k \) are not zero signals for all \( k \in \{1, ..., m\} \) may seem to be restrictive. This assumption is needed to guarantee that a failure is “visible”, i.e. a failure will affect system behaviours. This assumption is often a hidden assumption made in the fault
diagnosis literature. Referring to (4.6) and (4.7), if $u^k(t) = 0$ or $y^k(t) = 0$, then $f^k_a(t) = 0$ or $f^k_s(t) = 0$ respectively, regardless of whether a failure has occurred. Thus, in order to make $f^k_a(t) \neq 0$ or $f^k_s(t) \neq 0$ when $F^k_u = 0$ or $F^k_y = 0$, the assumption that $u^k(t) \neq 0$ or $y^k(t) \neq 0$ is required. Else, a failed actuator or sensor will not change the system behavior. Thus, this assumption is necessary for the failure to be detectable. In real life, it is unusual for an actuator not to perform any control action throughout the duration of operation, i.e. it is uncommon that $u^k(t) = 0$. Also, when $u(t) \neq 0$, the individual elements of $y(t)$ will typically be a nonzero signal. One possible method is to change the reference signal by a bit to cause change in the output before performing fault isolation using the SRG. Hence, this assumption is not restrictive.

Theorem 4.5 is important because it states that SRG can be used to solve the failure isolation problem for the LFSS. Furthermore, due to the structure of $\mathcal{M}(s)$ given in (4.20), the $i$-th structured residual generated by the SRG correspond directly to $F^{ii}$. Using this result, we will use the SRG to provide the combined failure matrix, $F$ to the FTD-PID controller given by (2.26) and (2.27). This allows the overall fault tolerant control scheme to respond to failure faster. The decentralized PD controller shown in (2.29), but without the feedforward term $\Gamma y_{ref}$, is used when a fault is detected but not isolated. This control configuration is taken because the FTD-PID controller may trigger instabilities and cause the states of the LFSS to grow large. This may cause unwanted strain to the LFSS. Figure 4.1 summarizes our proposed integrated fault tolerant control scheme.

![Figure 4.1: Our proposed integrated fault tolerant control scheme](image)

Furthermore, with introduction of the set $\mathcal{D}$, Theorem 4.5 is applicable to more general case. Suppose that certain locations are known to have failed. Also, assume that the failed locations are handled appropriately such that when a new failure occurs, the expression $F_c(s) \in \mathcal{L}^i \cup \mathcal{L}^{i+m}$ still holds. Theorem 4.5 states that one can continue to use the structured residuals
to determine the failed location. The set $\mathcal{D}$ can be constructed by first assuming it to be an empty set (i.e. there is no failed location) and subsequently update it by reassigning $\mathcal{D}$ to be $\mathcal{D} \cup \{i\}$ when the $i$-th location has a failure.

4.6 System Reconfiguration for Subsequent Failure Isolation

In the previous section, it was shown that it was possible to continue to isolate failed location provided that one can ensure $F_c(s) \in \mathcal{L}^i \cup \mathcal{L}^{i+m}$. In other words, at any time instances, only one of the elements of $f_a$ or $f_s$ can be nonzero. In real life, it is possible that one failure can occur after another. When this happens, $F_c(s)$ may have two nonzero elements. This would violate the assumption required by the SRG and failure isolation can no longer be performed.

Thus, in this section, we will propose a simple method to reconfigure a system with failure such that a new SRG can be designed and it can isolate the next failure. The basic idea is to make the system stop using actuators of the faulty locations for control purposes and to ignore the sensors of the faulty sensors for residual generation and fault isolation. To reconfigure the system, the combined failure matrix, $F$ is used, which is introduced in Section 2.2.2. First, reassign the control law to be $u = F\bar{u}$, where $\bar{u}$ represents the control signal applied to the system when no failures have occurred. Also, take the new measured sensors output to be $\bar{y}_m = T y_m$, where $\bar{y}_m$ represents all the measured output of the healthy locations. This results in a system with only healthy sensor-actuator pairs, which was previously given in (2.14) and (2.15). From Section 2.2.2 recall that $F = T^T T$. For convenience, we shall restate the reconfigured system model:

\[
\dot{x} = Ax + (BT^T)(T\bar{u}) \\
\bar{y}_m = TCx = Ty_m
\]

Note that one can no longer use the same ORG to perform accurate state estimation for the LFSS using both healthy and failed sensors. In order to perform state estimation successfully, one will have to use only the healthy sensors. According to the above reconfigured system model, the ORG for reconfigured system is given as shown:

\[
\dot{x} = A\hat{x} + (BT^T)(T\bar{u}) + HT(y_m - C\hat{x}) \\
r = T(y_m - C\hat{x}) = Tr_o
\]

where $r_o = y_m - C\hat{x}$. Note that the computation of $r_o$ is the same as the original ORG. The main difference here is the observer gain term $HT(y_m - C\hat{x})$. The term $T(y_m - C\hat{x})$ implies only
readings from the healthy locations are used for state estimation, assuming the reconfigured system is observable. This allows the state estimation to be successfully performed again. Hence, the residuals would behave in the same manner as the original residuals, i.e. when no failure occur, \( r = 0 \), else \( r \neq 0 \).

Suppose that the pair \((A, TC)\) is observable. Then, an ORG can be designed for the reconfigured system. The error system associated with the reconfigured system and the new ORG is given as follows:

\[
\dot{e} = (A - HTC)e + \begin{bmatrix} BT^T \\ 0 \end{bmatrix} \begin{bmatrix} T_{fa} \\ Tfs \end{bmatrix}
\]

\[
r = TCe + Tfs
\]

or equivalently,

\[
\mathcal{R}(s) = \left( TC(sI - (A - HTC))^{-1} \begin{bmatrix} BT^T \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & I \end{bmatrix} \right) \begin{bmatrix} T_{Fa}(s) \\ T_{Fs}(s) \end{bmatrix}
\]  

(4.40)

First, the matrix \( T \) hides all the known faults from the new ORG due to terms \( T_{fa} \) and \( Tfs \). This ensures the ORG can detect the next faulty location. Also, note that the fault-to-residual transfer matrix takes on the same form as (4.16). Therefore, the previous theoretical results can be all applied to (4.40) to redesign a new FIF, which then result in a new SRG.

Assume that \( \kappa \) < \( p \) faults have occurred and a SRG can still be designed after system reconfiguration. Suppose that \( \mathcal{W}(s) \in \mathbb{C}^{(p-\kappa) \times (p-\kappa)} \) have been successfully designed for the reconfigured ORG shown in (4.40). Then, the output from \( \mathcal{W}(s) \) is simply:

\[
\tilde{\mathcal{R}}(s) = \mathcal{W}(s) \mathcal{R}(s)
\]

where \( \mathcal{R}(s) \in \mathbb{C}^{p-\kappa} \) represent the residual from the new ORG, \( \tilde{\mathcal{R}}(s) \in \mathbb{C}^{p-\kappa} \) is the new structured residual. The original structured residuals (for a healthy system) were designed such that the \( i \)-th structured residual will not respond to a failure in the \( i \)-th location. This is no longer the case when \( \tilde{\mathcal{R}}(s) \) is used as the new structured residual. When \( \tilde{\mathcal{R}}(s) = 0 \) while other structured residual are nonzero, this may not mean a failure has occurred in the \( i \)-th location. The failure could be in another location due to the “order reduction” resulting from system reconfiguration using \( T \). For example, suppose that the 1st location had a fault previously and a new SRG has been designed to isolate the next failure. In this case, the first element of the new structured residual \( \tilde{\mathcal{R}}(s) \) does not correspond to location 1 but location 2.

Hence, to make the \( i \)-th structured residual (from the new SRG) not to respond to faults or
failures in the $i$-th location, we take the structured residuals to be:

$$\mathcal{R}(s) = T^T \mathcal{W}(s) \mathcal{R}(s)$$  \hspace{1cm} (4.41)

Equation (4.41) implies that all structured residuals that correspond to the known failed locations will always be 0. Thus, when isolating for a new fault or failure, structured residuals corresponding to the known failed locations should not be considered.

### 4.7 Simulation Results

In this section, we will demonstrate the performance of our proposed integrated FDI-FTC scheme, namely merging the SRG with the two fault tolerant controllers proposed in [38]. Our integrated scheme is applied on a finite-dimensional model of the Purdue space platform, which is a benchmark example for LFSS. The details of deriving the model can be found in Appendix B.3 of [38]. Our proposed scheme will be compared against Huang’s scheme introduced in [38]. It will be shown that that our proposed scheme can detect and isolate single fault occurrence in a particular location much faster than its predecessor. Also, we will show that one can reconfigure the system to handle subsequent failures.

The system matrices of the LFSS $(\Omega, \Delta, L)$ are given by (4.42)-(4.44). All matrices with large dimension can be found in the next section. Using these matrices, one can easily form the system matrices $(A, B, C)$ that are given in (4.3)-(4.5). Our goal is to make the output to track the reference signal, $y_{ref} = [1 \ 2 \ 0.5 \ -2 \ -3]^T$. At time $t = 3000s$, sensor 5 will suffer a complete failure. Later on, actuator 1 will also suffer from a complete failure at $t = 4500s$.

Recall that the LFSS has to satisfy several assumptions before one can design a SRG for it. The Purdue space platform example is observable, every single fault occurrence is detectable and it has 5 sensor-actuator pairs. These conditions were found to hold true even after the system is reconfigured to ignore the 5th sensor-actuator pair. Hence, it satisfies all the assumptions and a SRG can be designed.

First, consider the system with no known failures. In order to design the SRG, we follow the algorithm proposed in Section 4.4.2. First, we construct the ORG described by (4.13) and (4.14). We selected 24 poles evenly spaced from $-0.05$ to $-0.01$, which results in $H_1$ shown in (4.45).

Then, $G_{f1}(s)$ is computed using the system matrices of the ORG. The fault isolability condition proposed in part (e) of Theorem 4.3 was found to hold true for all $i$ from 1 to 5. Thus, we start to design $\mathcal{W}_1(s)$. Steps 5 to 9 of the algorithm presented in Section 4.4.2 are done automatically using a program developed by us in MATLAB. It involves picking a random row
for \( W_1(s) \) basing on the nullspace of \( (\hat{G}_f^1(s))^T \) and making it to have all of its entries to be stable and rational. Finally, to implement \( W_1(s) \), we converted the zeros-poles-gain representation of \( W_1(s) \) into a state space form. The state space representation of \( W_1(s) \) which we are using consist of 344 states. Clearly, the computation of \( W_1(s) \) is hard to do manually and the program developed to solve for \( W_1(s) \) is an important contribution in this research work.

Next, consider the system reconfigured knowing that the 5th sensor-actuator pair has a failure. The same poles were chosen for the reconfigured ORG and it results in the observer gain, \( H_2 \) shown in (4.46). Note that \( H_2 \) has one less column because the 5-th sensor-actuator pair has been turned off and it is not used for control and FDI purposes. Once \( H_2 \) is obtained, the same procedure for designing \( W_2(s) \) is done through our MATLAB program again. For this case, we obtain a state space model with 248 states for \( W_2(s) \).

Once the two SRGs have been designed, we construct the fault tolerant controllers described by (2.24), (2.25) and (2.29). The fault tolerant controller is a simple decentralized PID/PD controller that requires the knowledge of \( F = F_yF_u \). We choose 
\[
K_p = \text{diag}([500 500 2000 500 500]) \\
K_d = \text{diag}([4 \cdot 10^4 4 \cdot 10^4 16 \cdot 10^4 4 \cdot 10^4 4 \cdot 10^4]) \\
K_i = \text{diag}([1 1 4 1 1]) \quad \text{and} \quad \epsilon = 0.5.
\]
To implement the PD controller, we need to calculate the feedforward term, \( \Gamma = (L^T(\Omega^2 + LL^T)^{-1}L)^{-1} - I \). It is found to be:

\[
\Gamma = \begin{bmatrix}
61.7 & -32.4 & -58.6 & 61.7 & -32.4 \\
-32.4 & 61.7 & -58.6 & -32.4 & 61.7 \\
-58.6 & -58.6 & 235.0 & -58.6 & -58.6 \\
61.7 & -32.4 & -58.6 & 61.7 & -32.4 \\
-32.4 & 61.7 & -58.6 & -32.4 & 61.7
\end{bmatrix}
\]

Figure 4.2 shows the measured output, \( y_m(t) \) of the LFSS before and after the failures and Figure 4.3 shows the behavior of the structured residual, \( \bar{r}(t) \) in response to failures. From Figure 4.2 one can observe that all the healthy locations of the system try to track their respective reference signal after a failure has occurred.

After time 3000s, \( y_5^m \) (solid green line) becomes 0. However, the actual position (dotted green line) at sensor 5 is shown to be oscillating. This is expected since actuator 5 is turned off due to the sensor failure and it can no longer be used to move node 5. Also, at time 3000s, all of the structured residuals are nonzero signal except structured residual 5. Thus, one can conclude it is node 5 that has a failure occurred.

Similarly, after time 4500s, the position \( y_1^m \) starts to oscillate due to the actuator failure. Once again, all the nodes at the healthy location do not lose tracking ability due to the fast control reconfiguration. From Figure 4.3 it can be seen that only \( \bar{r}_2 \) to \( \bar{r}_4 \) is nonzero at time 4500s. To determine the failed location, we discard the structured residual that is associated
Figure 4.2: Output of LFSS using our proposed SRG integrated with existing FTC scheme (Sensor 5 fails at 3000s and Actuator 1 fails at 4500s)

with the known failed location, i.e. $\tilde{r}_5$. This means that at time 4500s, only $\tilde{r}_1 = 0$ which implies that location 1 has a failure.

Next, a comparison of using structured residuals and Huang’s fault diagnosis algorithm proposed in [38] for fault tolerant control purposes is studied. All the parameters associated to the fault tolerant controllers remain the same when using Huang’s scheme. Figure 4.4 shows the output response of the LFSS when Huang’s fault diagnosis algorithm is used to diagnose failures. When using Huang’s algorithm, no control reconfiguration has yet been performed. Thus, if one were to look further in time, one will find that there is steady state error in the output. After time 3000s, the PD control is switched on to make the system output to go to steady state when Huang’s fault diagnosis algorithm is used. However, when the structured residual approach is used, the healthy output of the system tries to re-track the reference signal shortly after the failures occur. This is because the failure is determined almost in real-time and the PID control is reactivated right after the failure location has been determined.

This example illustrates that the SRG can be used as a substitute for Huang’s fault diagnosis algorithm. Furthermore, when a single failure occurs at one location, it is clear that SRG can isolate the location that has failed much faster than Huang’s fault diagnosis algorithm. In this example, the SRG takes roughly 10s to detect and isolate failures, whereas Huang’s fault
Figure 4.3: Response of Structured Residual (Sensor 5 fails at 3000s and Actuator 1 fails at 4500s)

Figure 4.4: Output of LFSS using Huang’s FD algorithm and FTC scheme (Sensor 5 fails at 3000s and Actuator 1 fails at 4500s)
diagnosis algorithm takes 21,200s. This is a significant improvement in the time taken for fault diagnosis.

4.8 Matrices related to the Simulation

All the matrices related to the simulation (i.e. the previous section) with larger dimensions are provided in this subsection. They are shown as follows.

\[
\Omega^2 = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1.28 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 3.4 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 15.5 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix} \times 10^{-4} \quad (4.42)
\]

\[
\Delta = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1.28 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 3.4 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix} \times 10^{-8} \quad (4.43)
\]
$L = \begin{bmatrix}
2.36 & 2.36 & 2.36 & 2.36 & 2.36 \\
4.26 & -4.26 & 0 & -4.26 & 4.26 \\
4.26 & 4.26 & 0 & -4.26 & -4.26 \\
4.82 & 4.82 & -2.49 & 4.82 & 4.82 \\
6.38 & -6.38 & 0 & 6.38 & -6.38 \\
4.46 & 4.46 & 0 & -4.46 & -4.46 \\
-7.03 & 7.03 & 0 & 7.03 & -7.03 \\
2.55 & 2.55 & 3.49 & 2.55 & 2.55 \\
-7.65 & 7.65 & 0 & -7.65 & 7.65 \\
0.475 & 0.475 & 0 & -0.475 & -0.475 \\
5.95 & 5.95 & -1.18 & 5.95 & 5.95 \\
8.45 & 8.45 & 0 & -8.45 & -8.45 \\
\end{bmatrix} \times 10^{-4}$ (4.44)
\[
H_1 = \begin{bmatrix}
6.83 & 6.59 & 86.5 & 7.57 & 5.84 \\
26.2 & -25.1 & -12.5 & -24.3 & 25.3 \\
306.0 & 39.2 & 1444.0 & 67.2 & 273.0 \\
11.1 & -1.24 & -39.0 & 0.942 & 8.94 \\
7.64 & -5.41 & 6.44 & 4.99 & -2.76 \\
468.0 & -107.0 & 1899.0 & 2.75 & 376.0 \\
-23.0 & 23.2 & -3.25 & 23.4 & -23.2 \\
10.6 & 11.5 & 233.0 & 11.4 & 10.7 \\
-36.8 & 35.7 & -3.49 & -35.5 & 34.4 \\
1.5 \times 10^5 & 3.73 \times 10^4 & 7.42 \times 10^5 & 4.2 \times 10^4 & 1.46 \times 10^5 \\
46.2 & 31.7 & -133.0 & 34.2 & 43.6 \\
-6744.0 & -1144.0 & -3.26 \times 10^4 & -1655.0 & -6555.0 \\
0.042 & 0.0412 & 0.579 & 0.0514 & 0.0318 \\
0.15 & -0.14 & -0.11 & -0.132 & 0.142 \\
1.01 & 0.111 & 4.73 & 0.22 & 0.882 \\
0.0344 & -0.0314 & -0.093 & -0.0196 & 0.0226 \\
-0.107 & 0.104 & -0.0114 & -0.102 & 0.0982 \\
-42.8 & -10.2 & -209.0 & -11.0 & -41.1 \\
-0.869 & 0.84 & 0.335 & 0.816 & -0.846 \\
-0.148 & 0.351 & 4.76 & 0.325 & -0.122 \\
-0.569 & 0.283 & -0.835 & -0.226 & -0.0601 \\
-4699.0 & -3211.0 & -2.68 \times 10^4 & -2499.0 & -4888.0 \\
0.93 & -1.48 & -2.2 & -1.04 & 0.495 \\
733.0 & 341.0 & 3900.0 & 298.0 & 733.0
\end{bmatrix}
\] (4.45)
### 4.9 Discussion

The goal of developing the SRG for the LFSS was to provide the failure-related information to the FTD-PID controller in a short amount of time. The SRG is designed by first constructing an ORG, followed by a FIF. The ORG has to be constructed such that fault at every location is detectable, whereas the FIF is designed such that fault at every location is isolable. Due to careful selection of the structure of $\mathcal{M}(s)$, the SRG can be integrated directly with the FTD-PID controller proposed by [38].

The use of the SRG as a FDI module for the LFSS presents several advantages. These benefits are listed as follows:

\[
H_2 = \begin{bmatrix}
39.7 & -0.503 & -72.7 & 39.0 \\
63.3 & -19.1 & -32.0 & 43.9 \\
58.4 & -1.38 & -1366.0 & 52.2 \\
18.1 & -2.92 & -159.0 & 15.1 \\
28.2 & -5.0 & 98.2 & 23.2 \\
165.0 & -1.16 & -1800.0 & 180.0 \\
13.5 & 24.0 & -367.0 & 37.3 \\
-5.32 & 22.6 & 369.0 & 17.4 \\
-59.5 & 74.5 & 257.0 & 15.1 \\
5.69 \cdot 10^4 & -1.13 \cdot 10^4 & -7.53 \cdot 10^5 & 4.63 \cdot 10^4 \\
4.06 & 104.0 & 147.0 & 108.0 \\
-2399.0 & 552.0 & 3.27 \cdot 10^4 & -2177.0 \\
0.149 & 0.00873 & -0.433 & 0.157 \\
0.245 & -0.0752 & -0.321 & 0.169 \\
0.173 & 0.00208 & -4.31 & 0.161 \\
-0.277 & 0.0238 & 0.174 & -0.252 \\
0.682 & -0.0425 & -2.11 & 0.635 \\
-16.4 & 2.32 & 201.0 & -13.2 \\
-0.382 & -1.49 & -4.66 & -1.88 \\
0.856 & 1.57 & -0.909 & 2.43 \\
-2.77 & 5.22 & -21.1 & 2.45 \\
-2200.0 & 144.0 & 2.89 \cdot 10^4 & -1544.0 \\
0.601 & 6.1 & -36.9 & 6.69 \\
310.0 & -38.4 & -4055.0 & 231.0 
\end{bmatrix}
\]
1. The SRG is capable of isolating for failed location in a short amount of time. In the space platform example, we are able to shorten the time took for failure isolation by roughly a factor of 2120. This is the main advantage using the SRG as opposed to the fault diagnosis algorithm proposed in [38].

2. The SRG can provide the same failure information to the fault tolerant controller presented by [38].

3. The SRG is capable of not only identifying which location has failed, but also additive fault which is more general than just failure.

4. The SRG does not require changes in output reference signal, $y_{ref}$ as opposed to the fault diagnosis algorithm presented by [38]. Changing the output reference often will degrade the tracking performance, which may not be desirable in some application.

Nonetheless, there are a few disadvantages associated with using the SRG to isolate failed locations. These disadvantages are discussed below:

1. Within a certain time frame, the SRG can only isolate one failed location that is either an actuator failure or a sensor failure, but not both. If both actuator and sensor failed at a location, the SRG may still be able to isolate it but no guarantees can be made.

2. The computation for the FIF, $W(s)$ takes a long time to run on a computer. For the space platform, computation of $W(s)$ can take up to 4 to 5 hours. This is because all computations are done symbolically and not numerically.

3. The resulting state space representation for $W(s)$ typically has many state. A LFSS model like the space platform with 24 states can result in a state space representation of 344 states for $W(s)$.

4. Several assumptions are required to hold for the SRG to operate normally, e.g. fault detectability and fault isolability. These assumptions were not previously required for the fault diagnosis algorithm presented by [38].

The fact that several assumptions are required for the SRG and not for the fault diagnosis algorithm presented by [38] is interesting. By using the SRG, one obtains fault isolation speed improvement and a fault isolation method that is not “intrusive”. With such gains, it is common to have more assumptions enforced upon the system.

The main disadvantages of using the SRG are: (i) the time it takes to compute $W(s)$ and (ii) the resulting state space representation for $W(s)$ is often large. The first thing to note is that
a FIF is not unique because each row of $\mathcal{W}(s)$ is taken from a nullspace of columns of $\mathcal{G}_f(s)$. This implies that they may exist some methods to select a vector from the nullspace that result in a $\mathcal{W}(s)$ that gives rise to a smallest (hence unique) state space representation.

We suggest to reformulate and solve the problem in state space to improve the computation speed of $\mathcal{W}(s)$. This allows the model to be represented solely by numerical data type, which we expect it to run much faster. It could potentially reduce the computation time from 4 to 5 hours to several seconds.

We will briefly give a naive formulation for the SRG in state space. Consider the ORG presented by (4.10) and (4.11). Now, we want to find a system that accepts $r$ from the ORG and outputs the structured residual $\bar{r}$. Let $A_w \in \mathbb{R}^{\tilde{n} \times \tilde{n}}$, $B_w \in \mathbb{R}^{\tilde{n} \times \tilde{m}}$, $C_w \in \mathbb{R}^{\tilde{p} \times \tilde{n}}$ and $D_w \in \mathbb{R}^{\tilde{p} \times \tilde{m}}$ be the respective system matrices. Hence, we have the state space model of the FIF as follows:

$$\dot{z} = A_w z + B_w r \quad (4.47)$$

$$\bar{r} = C_w z + D_w r \quad (4.48)$$

Since the ORG and the FIF are concatenated system, we can merge (4.10) and (4.11) with (4.47) and (4.48) to obtain the following expression:

$$\begin{bmatrix} \dot{e} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} A - HC & 0 \\ B_w C & A_w \end{bmatrix} \begin{bmatrix} e \\ z \end{bmatrix} + \begin{bmatrix} B & -H \\ 0 & B_w \end{bmatrix} \begin{bmatrix} f_a \\ f_s \end{bmatrix} \quad (4.49)$$

$$\bar{r} = \begin{bmatrix} D_w C & C_w \end{bmatrix} \begin{bmatrix} e \\ z \end{bmatrix} + \begin{bmatrix} 0 & D_w \end{bmatrix} \begin{bmatrix} f_a \\ f_s \end{bmatrix} \quad (4.50)$$

Assume single fault occurrence at every location is isolable. The naive problem formulation can then be stated as such: Given (4.49) and (4.50), find $A_w$, $B_w$, $C_w$, $D_w$ and their appropriate dimensions $\tilde{n}$, $\tilde{m}$, $\tilde{p}$ such that $\mathcal{M}(s)$ takes on the structure (4.20) and every row of $\mathcal{M}(s)$ satisfies Theorem 4.3(c).

The described problem is not a conventional control problem for the following two reasons:

1. How does the structure of $\mathcal{M}(s)$ relate to (4.49) and (4.50)?

2. The number of states, $\tilde{n}$ is not fixed. This means there could potentially be infinite number of solutions to the problem. Typical control/observer problem involves solving for matrices with known dimension. Solving for a matrix with unknown dimension will definitely make the problem more complicated because there should exist a minimum for $\tilde{n}$ such that a solution exists.

It is worth mentioning that results from [57] can be potentially used to solve this problem. However, more investigation in this area is required.
A final issue to be discussed is the effect of initial conditions on the LFSS has on the SRG. Since the SRG consists of an ORG, it is expected that when the initial conditions of the actual system differ from the observer, then the error signal, $e$, will be a nonzero. If the observer is asymptotically stable, then $e$ will ultimately go to 0. By referring to (4.11), when the initial conditions of the system are different from the observer, then $r$ will be nonzero too. This implies that there will be a false alarm in the SRG when the observer is in the transient phase. Therefore, the SRG should only be used after the transient phase of the system to avoid false alarms.
Chapter 5

Conclusion and Future Work

In this thesis, two contributions are made in the field of fault tolerant control system. In reference [38], an integrated fault tolerant control scheme was proposed to diagnose and handle sensors-actuators failures for large flexible space structures. However, the long time taken by the fault diagnosis algorithm proposed in [38] is a major drawback. Thus, this limitation motivated our research to find a fast fault detection and isolation scheme to diagnose failure for large flexible space structures.

The $\mathcal{H}_\infty$ sliding mode observer was considered to be a possible candidate to diagnose failures. Unfortunately, the colocated large flexible space structure model does not satisfy the conditions required by the observer. Due to the attractive robust property of the $\mathcal{H}_\infty$ sliding mode observer, we diverted our attention to develop an integrated fault tolerant control scheme using the $\mathcal{H}_\infty$ sliding mode observer. To solve the original failure diagnosis problem for the large flexible space structure, we proposed to use structured residual to detect and isolate for single failure. The structured residual generator was designed to act as a ‘drop-in replacement’ for the fault diagnosis algorithm proposed by Huang in [38]. This allowed us to use the same fault tolerant controllers proposed by Huang in our integrated fault tolerant control scheme for the large flexible space structure.

The new integrated fault-tolerant control scheme using the $\mathcal{H}_\infty$ sliding mode observer is presented in Chapter 3. It uses fault and state estimates generated by the $\mathcal{H}_\infty$ sliding mode observer to perform actuator fault accommodation and output feedback control. Lyapunov stability analysis shows that the closed loop system states and state estimation errors are bounded in the presence of faults and system noise. The $\mathcal{H}_\infty$ sliding mode observer provides enhanced robustness to disturbances in the fault estimates and state responses. The performance improvement in using the $\mathcal{H}_\infty$ SMO is illustrated in the control of a permanent magnet synchronous motor by comparing its closed loop behaviour against those produced using the same controller but with a different observer. Simulation results showed that the effects of noise disturbances
on fault estimation and controlled state response are much more suppressed when the $H_\infty$ sliding mode observer is used.

In Chapter 4, a new fault diagnosis scheme for isolating failed location of large flexible space structures is presented. This fault diagnosis scheme uses a well-known technique called structured residuals to detect and isolate fault locations. The structured residual generator is constructed by first designing an observer-based residual generator, followed by designing a fault isolation filter. The observer-based residual generator produces residuals that can be used for basic fault detection purposes. The residuals are then passed through the fault isolation filter to produce structured residuals, which are capable of identifying failed location(s). Conditions for faults and failures to be detectable and isolable are also studied. The conditions for faults to be isolable provide an approach for constructing an algorithm (which can be implemented in software) to design the fault isolation filter. The structured residuals are then related to the combined failure matrix of the colocated large flexible space structure model. This shows that the structured residual generator can be used as ‘drop-in replacement’ for the fault diagnosis algorithm of the overall fault tolerant control scheme proposed in [38]. To handle subsequent failure(s), a system reconfiguration method is introduced to enable the design of a new structured residual generator that is not responsive to the known failures. Simulation results verified the validity of our method, as well as its superior performance over the fault diagnosis algorithm proposed in [38].

From our research, many insights have been obtained from these two problems. They can lead us to new possible extensions for these problems. We will first discuss the possible future work for the integrated fault tolerant scheme proposed in Chapter 3. They are listed as follows:

- Study the effect that $\delta$ has on the fault estimate produced by the sliding mode observer. Recall that $\delta$ is introduced to smoothen the high frequency switching of the sliding term.
- Design the $H_\infty$ sliding mode observer in discrete time. This will eliminate the need for tuning $\delta$, which can be time consuming. The sampling time of the discrete time sliding mode observer can be selected to match the speed of the computer/microprocessor and thus eliminate the need for high frequency switching.
- Study the interaction between the disturbance attenuation scalar, $\gamma$ and the two matrices resulting from solving the linear matrix inequality, namely $P$ and $K$.
- Develop a controller capable of output tracking with fault accommodation capability and to handle more general faults like system faults.
- Create a sliding mode observer to estimate sensor, system and actuator faults collectively. There are works that can estimate sensor faults and system/actuator faults individually.
However, to the author’s best knowledge, there isn’t a sliding mode observer that can cover all three types of faults.

Chapter 4 focused on applying structured residuals to detect and isolate for failures in large flexible space structures. One possible future work is to study the construction of the fault isolation filters in state space because it can improve the design computational time significantly. A discussion of this topic is given in Section 4.9. Another interesting extension would be to further extend this method or to search for other methods to isolate multiple faults simultaneously. To improve the practicality of this scheme, future work should also take system and output noise into consideration. One can reduce the chance of incorrect diagnosis by designing the fault isolation filter while minimizing the effect of noise on the structured residuals.
Bibliography


