A METHODOLOGY FOR
INTEGRATION OF CONTROL AND FAULT DIAGNOSTICS

DISSEPTION

Presented in Partial Fulfillment of the Requirements for
the Degree Doctor of Philosophy in the Graduate
School of The Ohio State University

By
Yong Wha Kim, B.S., M.S.

*****

The Ohio State University
1999

Dissertation Committee:
Dr. Giorgio Rizzoni, Advisor
Dr. Vadim I. Utkin
Dr. Stephen Yurkovich

Approved by:

Dr. Giorgio Rizzoni
Advisor

Department of Mechanical Engineering
In an attempt to realize fault tolerant control, a methodology for integrated design of control and fault diagnostic is investigated in this dissertation. Reliable operation of a system with automatic control should depend, not only the robustness of the control system but also on the systems fault diagnostics capabilities. A comprehensive strategy therefore needs to be implemented for fault tolerant control. Such a control strategy would comprise of three modules: a control module, a fault diagnostics module and a controller reconfiguration module. Design of these modules is a nontrivial issue. There exists significant inter modular interaction. However, conventional design approaches hitherto did not consider such interactions during the design process. This dissertation focuses on a comprehensive integration of these three modules and develops a methodology for an integrated design of fault diagnostics and control.

Specifically, the integrated design of control and diagnostics is achieved by combining the Integral sliding mode control methodology, observers with hypothesis testing and controller reconfiguration. Information obtained from integral sliding mode control and from observers with hypothesis testing is utilized so that a fault can be detected, isolated and compensated for. In summary, the role of the control module and the diagnostics module can be stated as follows.

- Control Module
  
  Perform primary control task

  Detection of fault using additive control terms (equivalent value of integral control term)

  Input(actuator) fault compensation
• Diagnostics Module

Generation of analytic redundancy (estimation using observers with hypothesis testing)

Fault isolation

i) Input (actuator) fault is isolated by observers using both output and estimated fault.

ii) Output (sensor) fault is isolated by decoupling a faulty sensor or by direct estimation of the fault. In the latter case, information obtained from controller (time derivative of sensor fault is obtained using integral sliding mode control term)

As an application example, the air and fuel dynamics of IC engine is considered. A mean value engine model is developed and implemented in Simulink. The air and fuel dynamics of the engine are identified using experimental data. The proposed algorithm for integration of control and diagnostics is then validated using the identified engine model.

For both control and diagnostic application, two novel observer design methods are proposed. An observer using a binary sensor (HEGO sensor) is developed for the estimation of fuel dynamics and air to fuel ratio dynamics. A nonlinear input observer is also developed using a sliding mode methodology and applied to the indicated torque estimation problem for an IC engine.
Dedicated to my parents, my wife Youngsook
and my little son Jaejung
ACKNOWLEDGEMENTS

I would like to thank my advisor, Professor Giorgio Rizzoni, for his guidance, support and enthusiasm during my Ph.D. study at The Ohio State University. His mix of guidance, encouragement and ample academic independence helped me mature academically. Also his enthusiasm always motivated me to take that extra step, toward excellence. I really appreciate his lessons both in and out of the classroom. My gratitude also goes out to Professor Vadim I. Utkin for his insightful and often erudite guidance. I will miss the times that I spent with him having valuable discussions. I also thank Professor Stephen Yurkovitch for his valuable suggestions toward refining this dissertation.

I am grateful to Dr. Namhee Han who has been my mentor as an engineer and as a man. Without his encouragement and belief in me, I would never have started this long academic journey toward a Ph.D.

I also wish to thank friends in Powertrain Control and Diagnostics laboratory who stood by me during the hard times and shared the fun times. Especially, I enjoyed the discussions with Yueyun Wang, Vasanth Krishnaswami and Devesh Upadhyay. I also wish to thank my Korean friends in Columbus who enriched my life outside school.

I could not have finished my Ph.D. study without the love and support of my parents and family. I can never thank enough, my lovely wife, Youngsook, who has been there all the times when I needed her and has made this journey alongside of me with mature understanding. I also thank my son, Jaejung, who was born during the course of my Ph.D. study and continues to grow into a nice boy. A pleasant and loving distraction allowing me to stay in touch with life and helping me believe in my self.
VITA

January 15, 1965 .................................................. Born - Seoul, Korea
1987 ................................................................. B.S.M.E., Seoul National University
1987 - 1994 .............................................................. Research Engineer, Daewoo Motor Company, Korea
1995 ................................................................. M.S., The Ohio State University
1994 - present ..................................................... Graduate Research Associate, The Ohio State University

PUBLICATIONS

Research Publications:


FIELDS OF STUDY

Major Field: Mechanical Engineering

Studies in: System Dynamics and Control
# TABLE OF CONTENTS

ABSTRACTS ............................................................................................................. ii

DEDICATION .......................................................................................................... vi

ACKNOWLEDGEMENTS .......................................................................................... v

VITA ........................................................................................................................ vi

LIST OF TABLES ..................................................................................................... xi

LIST OF FIGURES .................................................................................................. xii

NOMENCLATURE .................................................................................................. xvi

CHAPTEERS

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. INTRODUCTION .......................................................... 1</td>
<td></td>
</tr>
<tr>
<td>1.1. Motivation and Objectives .................................. 1</td>
<td></td>
</tr>
<tr>
<td>1.2. Contributions of the Study .................................. 3</td>
<td></td>
</tr>
<tr>
<td>1.3. FDI Preliminaries ................................................ 4</td>
<td></td>
</tr>
<tr>
<td>1.4. Problems in Automotive Engine Control and Fault Diagnostics ....... 15</td>
<td></td>
</tr>
<tr>
<td>1.5. Conclusion .............................................................. 19</td>
<td></td>
</tr>
<tr>
<td>2. REVIEW .......................................................... 20</td>
<td></td>
</tr>
<tr>
<td>2.1. Model Based FDI Methods .................................... 20</td>
<td></td>
</tr>
<tr>
<td>2.1.1. Parity Equation Based Residual Generation ........ 23</td>
<td></td>
</tr>
<tr>
<td>2.1.2. Observer Based Residual Generation ................. 25</td>
<td></td>
</tr>
<tr>
<td>2.2. Fault Isolation using Multiple Observers .................. 27</td>
<td></td>
</tr>
<tr>
<td>2.2.1. Dedicated Observer Scheme ............................... 28</td>
<td></td>
</tr>
<tr>
<td>2.2.2. Generalized Observer Scheme ......................... 31</td>
<td></td>
</tr>
</tbody>
</table>
2.2.3. Nonlinear Parity Equation Residual Generation Scheme ............................................. 33

2.3. Diagnostic Observers ........................................................................................................... 37

2.3.1. Diagnostic Observer for Linear System ............................................................................. 38

2.3.2. Diagnostic Observer for Nonlinear System ...................................................................... 47

2.4. Fault Tolerant Control ......................................................................................................... 57

2.4.1. Fault Tolerant Control in general ....................................................................................... 57

2.4.2. Integrated Design of Diagnostics and Control (4 parameter controller) ....................... 59

2.5. Application of Modern Control and Estimation Theory to Automotive Engine AFR Control and Diagnosis ....................................................................................................................... 63

2.5.1. Control Application ............................................................................................................. 63

2.5.2. Diagnostics Application ..................................................................................................... 67

2.6. Conclusion ......................................................................................................................... 69

3. INTEGRATION OF CONTROL AND FAULT DIAGNOSTICS ............................................. 70

3.1. Problems Involved in Fault Tolerant Control and Fault Diagnostics ................................. 71

3.1.1. Closed-loop Control System and Fault ............................................................................. 72

3.1.2. Estimation and Identification Problems in Fault tolerant Control and Diagnostics ....... 75

3.2. Estimation and Identification Problems in Sliding Mode .................................................. 76

3.2.1. Introduction to Sliding Mode Methodology ................................................................. 77

3.2.2. Sliding Mode Control Design by Equivalent Control Method ..................................... 78

3.2.3. Estimation and Identification Problems in Sliding Mode ............................................. 80

3.3. Estimation Methods for Special Cases ............................................................................... 91

3.3.1. Input Estimation using Sliding Mode .............................................................................. 91

3.3.2. Observer using Binary Sensor ....................................................................................... 94

3.4. Integration of Control and Fault Diagnostics - Methodology ........................................... 100

3.4.1. Integral Sliding Mode Control ...................................................................................... 100

3.4.2. Use of Integral Sliding Mode Control for Fault Detection and Compensation .......... 103

3.4.3. Fault Isolation using Observers with Hypothesis ........................................................ 106

3.5. Conclusion ......................................................................................................................... 112

4. APPLICATION TO IC ENGINE AIR AND FUEL MANAGEMENT SYSTEM ...................... 113

4.1. Internal Combustion Engine Model – Mean Value Engine Model .................................... 114

4.1.1. Model Description ........................................................................................................... 116

4.2. Model Identification .......................................................................................................... 126

4.2.1. Experimental Setup ........................................................................................................ 126

4.2.2. Identification of Intake Air Dynamics ............................................................................ 127
4.2.3. Identification of Fuel Dynamics (Wall Wetting Model) ........................................ 133

4.3. Experimental Study on Estimation and Identification ............................................. 141
   4.3.1. Estimation of Cylinder Air and Efficiency ...................................................... 141
   4.3.2. Estimation of Throttle Angle ........................................................................... 146
   4.3.3. Estimation of Fuel Film Dynamics and AFR using UEGO Sensor ...................... 148
   4.3.4. Estimation of Fuel Film Dynamics and AFR using HEGO Sensor ...................... 151
   4.3.5. Estimation of Indicated Torque for an IC Engine ........................................... 153
   4.3.6. Diagnostics Using NPERG Scheme with Observers ......................................... 158

4.4. Simulation Study on Integration of Control and Diagnostics ................................... 161
   4.4.1. Problem Formulation ...................................................................................... 161
   4.4.2. AFR Control via Estimation and Identification ............................................... 164
   4.4.3. Integration of Control and Diagnostics .......................................................... 170

4.5. Conclusion .............................................................................................................. 183

5. CONCLUSIONS ......................................................................................................... 184

Bibliography .................................................................................................................. 188

Appendix A. Specification of FORD V12 4.6liter Engine ........................................... 197

Appendix B. Powertrain Simulator (PTSIM) ................................................................. 203


LIST OF TABLES

<table>
<thead>
<tr>
<th>TABLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1.</td>
<td>General system variables and faults</td>
</tr>
<tr>
<td>1.2.</td>
<td>Example of structured residual</td>
</tr>
<tr>
<td>1.3.</td>
<td>Inputs and outputs of automotive engine</td>
</tr>
<tr>
<td>1.4.</td>
<td>HC and NOx emission limits for passenger cars</td>
</tr>
<tr>
<td>2.1.</td>
<td>Mathematical Description of System</td>
</tr>
<tr>
<td>2.2.</td>
<td>Characteristics of the primary residual for DOS example</td>
</tr>
<tr>
<td>2.3.</td>
<td>Signature for the secondary residual for DOS example</td>
</tr>
<tr>
<td>2.4.</td>
<td>Characteristics of the primary residual for GOS example</td>
</tr>
<tr>
<td>2.5.</td>
<td>Signature for the secondary residual for GOS example</td>
</tr>
<tr>
<td>2.6.</td>
<td>Fault signature for NPERG example</td>
</tr>
<tr>
<td>3.1.</td>
<td>Estimation and identification methods in fault diagnostics</td>
</tr>
<tr>
<td>3.2.</td>
<td>Decision logic with residual $r_A$ and $r_B$</td>
</tr>
<tr>
<td>4.1.</td>
<td>Estimation error</td>
</tr>
<tr>
<td>4.2.</td>
<td>Estimation error for different model</td>
</tr>
<tr>
<td>4.3.</td>
<td>Experimental conditions for fuel dynamics identification</td>
</tr>
<tr>
<td>4.4.</td>
<td>Model parameters for Waukesha single cylinder engine</td>
</tr>
<tr>
<td>4.5.</td>
<td>Response of hypothesis observer A and B</td>
</tr>
<tr>
<td>4.6.</td>
<td>Decision logic for isolation of fault</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1.</td>
<td>4</td>
</tr>
<tr>
<td>1.2.</td>
<td>8</td>
</tr>
<tr>
<td>1.3.</td>
<td>9</td>
</tr>
<tr>
<td>1.4.</td>
<td>10</td>
</tr>
<tr>
<td>1.5.</td>
<td>11</td>
</tr>
<tr>
<td>1.6.</td>
<td>12</td>
</tr>
<tr>
<td>1.7.</td>
<td>14</td>
</tr>
<tr>
<td>1.8.</td>
<td>17</td>
</tr>
<tr>
<td>2.1.</td>
<td>21</td>
</tr>
<tr>
<td>2.2.</td>
<td>28</td>
</tr>
<tr>
<td>2.3.</td>
<td>31</td>
</tr>
<tr>
<td>2.4.</td>
<td>35</td>
</tr>
<tr>
<td>2.5.</td>
<td>36</td>
</tr>
<tr>
<td>2.6.</td>
<td>58</td>
</tr>
<tr>
<td>2.7.</td>
<td>59</td>
</tr>
<tr>
<td>2.8.</td>
<td>61</td>
</tr>
<tr>
<td>2.9.</td>
<td>61</td>
</tr>
<tr>
<td>3.1.</td>
<td>71</td>
</tr>
<tr>
<td>3.2.</td>
<td>75</td>
</tr>
</tbody>
</table>
3.3. Responses of the discrete system observer ................................................................. 99
3.4. Estimation of continuous system responses ................................................................. 99
3.5. Schematic of hypothesis observer for actuator fault isolation ........................................ 108
4.1. Block diagram of a mean-value engine model ............................................................ 114
4.2. Air-fuel related subsystem of an IC engine .................................................................. 115
4.3. Fuel dynamic (Wall wetting model) ............................................................................ 118
4.4. Variation of $f_i$ for $NO_x$, $HC$, $CO$, $CO_2$, $O_2$ ....................................................... 122
4.5. Regression fit for mass generation parameter $f_i$ as a function of air fuel ratio ............ 123
4.6. A single degree of freedom crankshaft model ............................................................ 123
4.7. Schematics of experimental setup ................................................................................ 126
4.8. Operating conditions for steady state experiments ....................................................... 127
4.9. Measured and estimated mass flow rate of air ............................................................. 129
4.10. Measured and estimated mass flow rate of air ............................................................. 130
4.11. Computed and estimated cylinder charge efficiency .................................................... 132
4.12. Step changes in fuel flow rate and UEGO response ..................................................... 135
4.13. UEGO responses (3 different engine speeds) .............................................................. 136
4.14. UEGO responses (3 different temperatures) ............................................................... 136
4.15. Fuel evaporation time constant for various engine speed and temperature ................ 137
4.16. Fraction of fuel deposited for various engine speed and engine temperature .............. 137
4.17. Fuel evaporation characteristics with new variable $\theta_f$ ........................................... 138
4.18. Normalized fuel evaporation characteristics ............................................................... 137
4.19. Measured and predicted UEGO response ................................................................. 138
4.20. Estimation result during steady state ($\alpha = 5\%, \omega = 860 \text{ RPM, } p_m = 50\text{kPa}$) ............................................................................................................................. 142
4.21. Estimation result at steady state ($\alpha = 10\%, \omega = 1550\text{RPM, } p_m = 50\text{kPa}$) ........ 144
4.22. Estimation result during throttle tip-in ($15^\circ \rightarrow 20^\circ$ at 35th cycle) .......................... 145
4.23. Estimation result during throttle tip-in ....................................................................... 147
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.24</td>
<td>Fuel film mass estimation result during throttle tip-in and tip-out</td>
<td>149</td>
</tr>
<tr>
<td>4.25</td>
<td>Injected fuel flow rate estimation result during throttle tip-in and tip-out</td>
<td>150</td>
</tr>
<tr>
<td>4.26</td>
<td>Block Diagram for IC engine air-fuel dynamic system</td>
<td>152</td>
</tr>
<tr>
<td>4.27</td>
<td>Block Diagram equivalent to the system (4.70)-(4.71)</td>
<td>152</td>
</tr>
<tr>
<td>4.28</td>
<td>Observer responses during throttle tip-in</td>
<td>153</td>
</tr>
<tr>
<td>4.29</td>
<td>Schematic diagram of a single cylinder engine connected to a dynamometer</td>
<td>154</td>
</tr>
<tr>
<td>4.30</td>
<td>Estimation result from sliding mode unknown input observer</td>
<td>157</td>
</tr>
<tr>
<td>4.31</td>
<td>Observer configuration</td>
<td>158</td>
</tr>
<tr>
<td>4.32</td>
<td>Throttle sensor fault and the corresponding residual</td>
<td>159</td>
</tr>
<tr>
<td>4.33</td>
<td>Fuel injector fault and the corresponding residual</td>
<td>160</td>
</tr>
<tr>
<td>4.34</td>
<td>Schematic of controller configuration with HEGO sensor</td>
<td>168</td>
</tr>
<tr>
<td>4.35</td>
<td>AFR during throttle step change – simulation results</td>
<td>169</td>
</tr>
<tr>
<td>4.36</td>
<td>Schematics of controller configuration with UEGO sensor</td>
<td>171</td>
</tr>
<tr>
<td>4.37</td>
<td>Schematics of control and diagnostics for air-fuel management system</td>
<td>177</td>
</tr>
<tr>
<td>4.38</td>
<td>Injected fuel with fuel injector fault</td>
<td>178</td>
</tr>
<tr>
<td>4.39</td>
<td>AFR response with fuel injector fault</td>
<td>179</td>
</tr>
<tr>
<td>4.40</td>
<td>Estimation of fuel injector fault</td>
<td>179</td>
</tr>
<tr>
<td>4.41</td>
<td>Residual behavior</td>
<td>180</td>
</tr>
<tr>
<td>4.42</td>
<td>UEGO sensor output with fault</td>
<td>181</td>
</tr>
<tr>
<td>4.43</td>
<td>Residual behavior</td>
<td>181</td>
</tr>
<tr>
<td>4.44</td>
<td>Estimation of UEGO sensor fault</td>
<td>182</td>
</tr>
<tr>
<td>4.45</td>
<td>AFR response with UEGO sensor fault</td>
<td>182</td>
</tr>
<tr>
<td>B.1</td>
<td>Block diagram representation of the powertrain model</td>
<td>209</td>
</tr>
<tr>
<td>B.2</td>
<td>Top-level simulink engine model</td>
<td>210</td>
</tr>
<tr>
<td>B.3</td>
<td>Vehicle module</td>
<td>210</td>
</tr>
<tr>
<td>B.4</td>
<td>Throttle block</td>
<td>211</td>
</tr>
</tbody>
</table>
B.5. Intake manifold block ................................................................. 212
B.6. Pumping block ........................................................................ 212
B.7. Fuel film dynamics block ............................................................ 213
B.8. Combustion block ..................................................................... 213
B.9. Exhaust gas emissions block ...................................................... 214
B.10. Crankshaft dynamics block ...................................................... 214
B.11. Drive-train and vehicle longitudinal dynamics block ............... 215
B.12. Torque converter Block ............................................................ 215
B.13. Transmission and gear shift Block ........................................... 216
B.14. Vehicle longitudinal dynamics block ........................................ 216
B.15. Sensor module ........................................................................ 217
B.16. Controller module ................................................................... 218
B.17. Actuator module ...................................................................... 219
B.18. Throttle opening trajectory ....................................................... 224
B.19. Throttle opening trajectory ....................................................... 225
B.20. AFR response during throttle tip-in and tip-out ....................... 226
B.21. Engine speed during throttle tip-in and tip-out ......................... 226
\textbf{NOMENCLATURE}

\begin{itemize}
  \item $A_{th}$ = throttle area [$m^2$]
  \item $D$ = throttle bore diameter [$m$]
  \item $d$ = throttle shaft diameter [$m$]
  \item $a = d/D$
  \item $\alpha$ = throttle angle [rad]
  \item $\alpha_o$ = throttle angle when closed [rad]
  \item $\dot{m}_{a,sh}$ = mass flow rate of air at throttle [$kg/sec$]
  \item $C_{D,sh}$ = throttle discharge coefficient
  \item $R$ = ideal gas constant [$J/(kg\cdot K)$]
  \item $p_a$ = ambient pressure [$Pa$]
  \item $T_a$ = ambient temperature [$K$]
  \item $p_m$ = intake manifold pressure [$Pa$]
  \item $\gamma$ = specific heat ratio [$J/(kg\cdot K)$]
  \item $m_{am}$ = mass of air in intake manifold [$kg$]
  \item $\dot{m}_a$ = mass flow rate of air entering cylinder [$kg/sec$]
  \item $\eta_t$ = volumetric efficiency
  \item $t$ = time [$sec$]
\end{itemize}
\[ \theta = \text{crank angle [rad]} \]
\[ V_m = \text{intake manifold volume [m}^3\text{]} \]
\[ V_d = \text{displacement volume [m}^3\text{]} \]
\[ \omega = \text{engine speed [rad/sec]} \]
\[ \dot{m}_{\text{EGR}} = \text{mass flow rate of EGR entering cylinder [kg/sec]} \]
\[ A_{\text{eq}} = \text{effective area of EGT valve [m}^2\text{]} \]
\[ P_o = \text{pressure at EGR valve outlet [Pa]} \]
\[ T_o = \text{temperature at EGR valve outlet [K]} \]
\[ m_f = \text{mass of fuel in fuel film [kg]} \]
\[ \dot{m}_f = \text{fuel flow rate from injector [kg/sec]} \]
\[ \dot{m}_{\text{en}} = \text{fuel flow rate entering cylinder [kg/sec]} \]
\[ \tau_f = \text{fuel evaporation time constant [sec]} \]
\[ X = \text{fraction of injected fuel enter into the film} \]
\[ \tau_m = \text{fuel flow rate from film [kg/sec]} \]
\[ \tau_{\text{EGO}} = \text{EGO sensor time constant [sec]} \]
\[ \phi_m = \text{measured equivalence ratio} \]
\[ t_c = \text{cycle delay in time [sec]} \]
\[ \theta_c = \text{cycle delay in crank angle [rad]} \]
\[ t_r = \text{transportation delay in time [sec]} \]
\[ \theta_i = \text{transportation delay in crank angle [rad]} \]
CHAPTER 1

INTRODUCTION

1.1 Motivation and Objectives

Along side of the rapid progress in automatic control algorithms over last few decades, there has been a growing demand for fault tolerant control application. Fault tolerant control can be achieved both by improving the reliability of individual components in a system, and by appropriate implementation of fault detection, isolation, identification and accommodation (FDIA) schemes. At the very beginning of the automatic control era, most efforts were focused on improving the reliability of the system components for fault tolerant control. Although elaborate attempts for reliable design and manufacturing of the system units continue in practice, the FDIA concept has gained gradual acceptance for fault tolerant control application over the last two decades. The progress in FDIA theory and application is facilitated not only by the availability of complex and sophisticated automatic control algorithms but also by a growing demand for reliability and safety. This progress has also benefited from the rapid development in computing power and from advances in modeling, simulation, state estimation and parameter estimation.

The idea of FDIA has existed since the very beginning of automatic control. The earliest usage of FDIA involved straight limit checking of measurements. In the event of an out-of-range measurements, the
controller could be reconfigured, ignoring the faulty measurement, to a different (feedback or feedforward) configuration. Hardware redundancy and signal processing concepts were also introduced during the early stages of FDIA for fault detection, isolation and identification (FDI) purposes. Since the mid 1970’s, analytical redundancy approaches based on physical models of the process have become increasingly popular. The analytical redundancy concept makes use of the inherent redundancy in the mathematical model of a process and exploits it to develop a FDI strategy. Most of the approaches developed so far take advantage of linear system theory and are applied to a linear process or to a process linearized at some operating point. Not much work has been done so far in nonlinear FDI theory and design, though nonlinear control theory has seen much development. Among the nonlinear model based FDI approaches, the Nonlinear Parity Equation Residual Generation scheme (NPERG scheme, Krishnaswami and Rizzoni, 1997), developed recently, has showed promising results in fault detection and isolation. The NPERG scheme makes use of forward and inverse models of a process and utilizes analytic redundancy in the models to isolate and identify faults. However, the NPERG scheme is designed to provide isolation of a fault; accommodation of the system follows isolation of the fault is still an open question. Fault tolerant control approaches have in many cases not taken advantage of the use of model-based FDI methods, and have often been based on designing the control system to be robust in the presence of fault.

Integration of FDI and fault tolerant control is an area that has not been explored extensively until recently. In the case of linear systems, so-called 4-parameter controller approach was proposed as an integration method for FDI and fault tolerant control. The object of this study is the integration of control and diagnostics for nonlinear systems to provide fault tolerant control. To accomplish this goal, the estimation problem, the identification problem, and the control problem can be explored in parallel or in sequence. In this work, solutions to the problems are examined in the context of sliding mode control theory. One of the reasons for using the sliding mode approach is that it provides a similar design procedure for both linear and nonlinear systems and the similar design procedure can be applied, as will be shown, to both control and diagnostics problems.
Over the past decade, model-based control and fault diagnostics have been applied to automotive engines. With the possible introduction of floating point processor in automotive production in the near future, the implementation of model-based approaches to both control and diagnostics will likely accelerate. Furthermore, recently introduced technologies such as Electronic Throttle Control (ETC), Variable Valve Timing (VVT) and Direct Injection (DI) make the engine system more complex and interconnected. Design of control and diagnostics for such complex and multivariable system can benefit from modern, multivariable control theory and a model-based approach. One objective of this study is to provide a framework for design integration of control and diagnostics, so that the model-based approach in automotive engine applications can be applied more systematically and effectively.

1.2 Contributions of the Study

(1) Review of state of art in FDI and fault-tolerant control, and their application to automotive engines.

(2) Formulation of an engine model (so-called mean value engine model) for control and diagnostics application. This model is identified and validated with experimental data. A simulator based on this engine model is also implemented in SIMULINK as a computer aid for control and diagnostics design.

(3) Formulation of a design methodology for an observer with binary measurement, leading to new results concerning both estimation and control application.

(4) Formulation of an input estimation methodology based of sliding mode methodology, leading to new results in IC engine indicated torque estimation problem.

(5) Application of sliding mode observers to diagnose faults in automotive engines and experimental verification.

(6) Development and partial validation of a FDIA approach based on the sliding mode methodology.

(7) Transfer of these results to a real world production application (Lamborghini, Italy)
1.3 FDI Preliminaries

In this section we define the terminology relevant to FDI theory such as mathematical representation of system and fault, types of fault, residual. Also fundamental FDI procedures and methods are introduced.

**General system and fault representation**

Consider a block diagram representation of a dynamic system depicted in Figure 1.1. The variables and their description are listed in Table 1.1.

![Diagram](image)

**Figure 1.1:** General dynamic system and fault representation

The relations between the variables shown in Figure 1.1 can be described as follows.

\[
\begin{align*}
\dot{x} &= f(x,u,\theta) \\
y &= h(x,u,\theta)
\end{align*}
\]

(1.1)

where \( u = u_o + \Delta u \), \( \theta = \theta_o + \Delta \theta \), \( y^* = y + \Delta y \).

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(u_o)</td>
<td>Input vector</td>
</tr>
<tr>
<td>(\Delta u)</td>
<td>Input fault vector</td>
</tr>
<tr>
<td>(\theta_o)</td>
<td>Nominal parameter vector</td>
</tr>
<tr>
<td>(\Delta \theta)</td>
<td>Parameter (component) fault vector</td>
</tr>
<tr>
<td>(x)</td>
<td>State vector</td>
</tr>
<tr>
<td>(f(\cdot,\cdot,\cdot))</td>
<td>State evolution vector field</td>
</tr>
<tr>
<td>(h(\cdot,\cdot,\cdot))</td>
<td>Output measurement function</td>
</tr>
<tr>
<td>(y)</td>
<td>Actual output vector</td>
</tr>
<tr>
<td>(\Delta y)</td>
<td>Output fault vector</td>
</tr>
<tr>
<td>(y^*)</td>
<td>Measured output vector</td>
</tr>
</tbody>
</table>

**Table 1.1:** General system variables and faults
Mathematical Description of FDI process

The basic principle of fault diagnostics process is described as follows. Consider the system given in equation (1.1). Assuming that the system is free of any uncertainties (such as modeling uncertainty, disturbance and noise), the system without any fault is of the form:
\[
\begin{align*}
\dot{x}_s &= f(x_s, u_s, \theta_s) \\
y &= h(x_s, u_s, \theta_s)
\end{align*}
\]  

(1.2)

Also, the system in the presence of fault is described as follows:
\[
\begin{align*}
\dot{x} &= f(x, u + \Delta u, \theta + \Delta \theta) \\
y^* &= h(x, u + \Delta u, \theta + \Delta \theta) + \Delta y
\end{align*}
\]  

(1.3)

The objective of FDI process is to find the existence of faults ($\Delta u$, $\Delta y$, $\Delta \theta$) based on the fact that $x_s \neq x$, $y \neq y^*$ if any of faults are introduced in the system.

This problem is solvable only if the faults are linearly independent of the system states, $x(t)$. To illustrate this, a linear system is considered.
\[
\begin{align*}
\dot{x}_s(t) &= Ax_s(t) \\
y(t) &= Cx_s(t)
\end{align*}
\]  

(1.4)

The system response with initial condition $x_s(0)$ can be described in $s$-domain as follows.
\[
Y(s) = C(sI - A)^{-1}x_s(0)
\]  

(1.5)

Now consider the same system with a sensor fault
\[
\begin{align*}
\dot{x}(t) &= Ax(t) \\
y^*(t) &= Cx(t) + \Delta y(t)
\end{align*}
\]  

(1.6)

The system response with the same initial condition $x_s(0)$ in $s$-domain is expressed as:
\[
Y^*(s) = C(sI - A)^{-1}x_s(0) + \Delta Y(s)
\]  

(1.7)

As can be seen from the equations (1.5) and (1.7), $Y(s) \neq Y^*(s)$ due to the sensor fault $\Delta y(t)$. Now, consider the system response with a different initial condition $x_s(0) + \Delta x(0)$
\[
\begin{align*}
Y(s) &= C(sI - A)^{-1}(x_s(0) + \Delta x(0)) \\
&= C(sI - A)^{-1}x_s(0) + C(sI - A)^{-1}\Delta x(0)
\end{align*}
\]  

(1.8)
If \( \Delta y(t) \) is linearly dependent of \( x(t) \), we can find \( \Delta x(0) \) such that

\[
\Delta Y(s) = C(sI - A)^{-1}\Delta x(0).
\]  

From the equations (1.7)-(1.9), we know \( Y(s) = Y'(s) \). This implies that the output with a sensor fault \((y(t) = Cx(t) + \Delta y(t))\) are the same with the output \( y^*(t) \) (system response with a different initial condition \( x(0) + \Delta x(0) \)). A distinction between a sensor fault and a different initial condition can not be made if a fault is linearly dependent on system states. The same procedure can be applied to both actuator fault and parameter fault.

In summary, a fault should be independent of the system states, \( x(t) \) so that the effect of a fault and that of a initial conditions can be isolated. In most application, a fault is typically in the forms of offset, drift that can be modeled as a constant or a first order system. And they are usually independent of the system states, \( x(t) \).

**Nature of Faults**

The classification of faults is not unique and it depends on the purpose and on the diagnostics method used. The classification given below is based on the model-based diagnostic framework and is quite general.

1. **Additive measurement/actuator fault:**

Discrepancy between the true and measured system output or input. Typical examples are: sensor bias error and drift. These types of faults can be described by the following system equations:

\[
\dot{x} = f(x, u, \theta)
\]  

\[
y^* = h(x, u + \Delta u, \theta) + \Delta y
\]  

2. **Additive process fault:**

Disturbances acting on the plant which are typically due to actuator malfunction.

\[
\dot{x} = f(x, u + \Delta u, \theta) = f(x, u, \theta) + f^*(x, u, \theta)\Delta u
\]  

\[
y^* = h(x, u + \Delta u, \theta) = h(x, u, \theta) + h^*(x, u, \theta)\Delta u
\]
(3) **Multiplicative process fault:**

Changes in the plant parameter due to aging, changing manufacturing tolerance, out of range operation, etc.

\[
\dot{x} = f(x, u, \theta + \Delta \theta) \quad (1.14)
\]

\[
y^* = h(x, u, \theta + \Delta \theta) \quad (1.15)
\]

**Fault diagnosis process**

The process of fault diagnosis involves the following tasks. These can be performed sequentially or in parallel.

1. Fault detection: Presence of the fault
2. Fault Identification: Location of the fault
3. Fault Isolation: Magnitude of the fault
4. Accommodation: Corrective action of the controller

**Errors in Diagnosis**

Most plants are affected by noise and disturbances, which are sometimes difficult to distinguish from a fault. Furthermore, modeling error is unavoidable. The results of these effects are:

1. **False alarm:** Declaration of fault under normal operation
2. **Missed detection:** Fail to recognize a faulty mode of operation
3. **Misclassification:** The above two conditions are related to the binary detection of a fault (either normal or faulty operation), this is related to error in locating the size or the location of a fault

This study does not have the objective of formulating a framework that includes detection theory. For details on the latter, refer to Gerter (1998) and Soliman (1998).
**Desired properties of a diagnostic algorithm**

To decrease the probability of occurrence of the diagnostic errors mentioned above, the design/selection of a diagnostic algorithm should incorporate the following properties.

(1) Insensitivity to disturbances
(2) Noise suppression
(3) Robustness to modeling error
(4) Sensitivity to fault

An excellent review of the basic FDI concepts is found in Gertler (1998).

**Fault - Failure - Malfunction**

A fault can be defined simply to be the cause of a failure. And the failure of a system is characterized by malfunction (abnormal behavior) of the system. A block diagram description of the relationship among fault, failure and malfunction is shown in Figure 1.2

![Figure 1.2: Relationships among fault, failure and malfunction](image)

This cause-effect relationship among fault, failure and malfunction can be expanded in a hierarchical way from component level to sub-system and finally system level. For example, when a component has a fault, the subsystem (containing the defective component) could experience a failure. A failure of a subsystem could cause a malfunction of whole system (refer Figure 1.3).
The abnormal behavior (or malfunction) can be observed through appropriate measurements and corresponding inspection procedures. The basic idea of FDI is to recognize any abnormal behavior of the system through measurements and to locate the cause of the abnormal behavior. Approaches to FDI can be divided into two major categories: model-based methods and non-model-based methods. This study focuses on model-based FDI methodologies. Non-model-based FDI methodologies are introduced briefly below.

*Non-model-based FDI methodology*

Non-model-based approaches may be simple to implement and may even be cost effective. This is mainly due to the fact that non-model-based approaches do not explicitly consider the mathematical model of a plant, which often requires effort in identification and substantial computation. Non-model-based methods are usually based on empirical correlations, and are subject to uncertainty in the presence of changes in operating conditions, external disturbances, plant to plant variability, and aging. A model-based approach to fault diagnosis has the advantage of capturing the physical phenomena of the plant under consideration. This approach is based on the development of reasonably accurate models that accommodate physical intuition, while being simple enough to allow inferences to be made about system
operation through the monitoring of physics-based constants and system variables. Non-model-based methods include:

(1) Limit checking: a deviation from the normal operation can be detected by comparing the measured plant variables to preset limits (determined by off-line tuning). This is basically to check the limit of the plant hardware.

(2) Hardware redundancy: any fault in the sensors can be detected by measuring the same variable from different sensors and monitoring the discrepancies in those measurements.

(3) Methods based on signal processing techniques: many types of fault can be monitored through processing of the measured plant variables. A typical example is the use of the frequency spectrum of a measured variable. Any abnormality in the system can be reflected in the spectral content of measured variables.

(4) Rule-based methods: as one of the upcoming techniques in control application, rule-based methods such as fuzzy and genetic algorithm are expanding their domain into FDI application. However, some methods such as fuzzy algorithm, can use a model of system and can be extended to model-based method.

A typical non-model-based method can be represented in a block diagram form as depicted in Figure 1.4. The non-model-based methods listed above belong to the first block (signature generator) in Figure 1.4.

![Figure 1.4: Schematic of non-model based FDI](image-url)
Model-based FDI Methodology

The idea of model-based FDI methods is based on the use of analytical rather than physical redundancy. This analytical redundancy is inherently contained in the static and dynamic relationship amongst the input and output variables of the system. The analytical redundancy approach takes advantage of the existing redundancy by simply processing the measured information without adding any physical instrument to the system. However, there is a disadvantage, which stems from the fact that a mathematical model of plant is now required. The modeling and identification process requires not only considerable computational effort if it is designed for on-line applications, but also accuracy to avoid excessive modeling error. Therefore, the sensitivity of a diagnostic method to modeling error is one of the key issues in the application of model-based FDI methods. Most model-based FDI methods can be described by the block diagram shown in Figure 1.5.

Figure 1.5: General structure of model based FDI method

Assuming that a model of the plant is reasonably accurate to permit the use of a model-based FDI method, the general process of model-based FDI consists of the three stages depicted in Figure 1.6. These are described below in more detail.
(1) Generation of primary residual

The observations acquired through sensory measurements are compared to the analytically obtained values of the same variables. Such computation may require present and/or previous measurement of the other variables and mathematical models describing their relationship. The resulting discrepancy between the measured and calculated variable is called a primary residual. This step consists mainly of generating a primary residual that reflects the system behavior, and that has nominally zero value under normal conditions.

In summary, the above procedure can be stated as follows. *Inputs and outputs of a system are processed by an appropriate algorithm to generate a residual that has a nominal value (typically near zero) during normal operation and that deviates from the nominal value in the presence of a fault.*

(2) Secondary Residual Generation

In the ideal case, the residuals are zero, but in practice this is seldom the case. The residual deviates from zero due to factors such as noise, modeling error and faults. If noise is limited, the residual can be directly processed to analyze the faults with the aim of isolating the fault and its cause. Such analysis may be performed by pattern recognition or knowledge based inference. In the presence of significant noise, it is usually necessary to convert the primary residual into a secondary residual that can be used to distinguish faulty and normal operations of a plant. This task can be accomplished by simply filtering the primary residual, by statistical testing.
or by more complicated means such as spectral analysis. Also, any modeling error, unavoidable in practical situations, can be accommodated by similar approaches.

In summary, the above procedure can be stated as follows. The residual is examined to decide the time, location, sometimes also type, size and source of fault. This process can be based on threshold testing, statistical testing or rule based algorithms to make a decision.

Isolation of Faults

Fault isolation is the ability to distinguish a specific fault from other faults. In order to provide better fault isolability in the FDI process, residuals are generated in such a way that a fault can be distinguished from the others using one or more residuals. A systematic way of generating the residual for better fault isolation is generalized by Gertler (1998). According to his approach, such residuals can be divided into two groups:

1. structured residual
2. directional residual.

(1) Structured Residual

The residual vector \( r(t) \) is called structured when only a specific component has non zero value in response to a particular fault. Since a subset of the residual vector has non-zero value, the threshold test for each residual results in a Boolean decision, i.e., 0 for nominal and 1 for non-nominal. The combination of these Boolean decisions yields a fault code (a binary vector), and each of these for different faults denotes the corresponding fault signature. For example, the fault code under nominal operating condition has only zero components while it has different combinations of binary values for faulty situations. Therefore, any non zero component in the fault code denotes a fault in the system and a specific pattern of the non-zero components in the fault code represents the location of the fault. As a simple example, consider three different faults \( f_1, f_2, f_3 \) with three residuals \( r_1, r_2, r_3 \). The residual can be generated as in Table 1.2.
<table>
<thead>
<tr>
<th></th>
<th>$r_1$</th>
<th>$r_2$</th>
<th>$r_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Fault $f_1$</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Fault $f_2$</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Fault $f_3$</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 1.2: Example of structured residual

In Table 1.2, the fault code $(0 \ 0 \ 0)$ denotes the normal operation, and $(1 \ 1 \ 0)$ represents the occurrence of the fault $f_1$, etc.

(2) Directional Residual

Another approach for the fault isolation is the design of a directional residual. The directional residual $r(t)$ lies in a specific direction in the residual vector space in response to a fault. Therefore, the fault isolation can be achieved by investigating the direction of the residual. This means the fault signature is the direction of the residual. As an example, consider again three different faults ($f_1, f_2, f_3$).

![Directional Residual](image)

Figure 1.7: Example of directional residual

In Figure 1.7, the direction of each fault is shown (orthogonal), and a fault can be isolated by investigating the angle between the residual and each fault vectors. In this example, the residual shown indicates the presence of fault $f_1$ since it is closed to the fault direction $f_1$.  

14
1.4 Problems in Automotive Engine Control and Fault Diagnostics

The fundamental role of an automotive engine control system is to guarantee a satisfactory vehicle performance over a wide range of operating conditions. The engine is an important player in this scheme, its task being to supply the necessary torque to propel the vehicle. This objective is achieved by manipulating the input variables of an engine (see Table 1.3) in order to regulate engine output (see Table 1.3).

<table>
<thead>
<tr>
<th>Input Variables</th>
<th>Output Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Throttle Position</td>
<td>Torque</td>
</tr>
<tr>
<td>Fuel injection</td>
<td>Air-fuel ratio</td>
</tr>
<tr>
<td>Spark advance angle</td>
<td>Exhaust gas emissions</td>
</tr>
</tbody>
</table>

Table 1.3: Inputs and outputs of automotive engine

The complex and nonlinear nature of an engine coupled with a wide operating range, however, imposes difficulties in the design of engine control systems. The complex, nonlinear nature of an engine along with the typical problems faced during control design are addressed in Chapter 4. The various operating conditions of an engine can be summarized as follows.

- Engine Crank
- Cold Start
- Cold Drive-away
- Warm Drive-away
- Normal (warm) Cruise
- Part Throttle Acceleration
- Full Throttle Acceleration
- Deceleration
- Warm Idle

In addition, the broad range of operating conditions, environmental changes such as changes in altitude and ambient temperature introduce additional complexities into the engine control system.
Automotive engine control system development has been challenged by regulatory requirements in addition to the necessary task of providing customers with satisfactory vehicle performance at low cost. The regulatory requirement on the automotive engine control system can be summarized as:

- Regulating exhaust emission to meet increasingly stringent government standard
- Complying with fuel economy standard such as Corporate Average Fuel Economy (CAFE).
- Providing On-board-Diagnosis (OBD) capability to monitor emission-relevant functions and components.

Regulation on the upper limit of exhaust gas emissions will become more stringent starting in the year 2002, as shown in Table 1.4.

<table>
<thead>
<tr>
<th>Emission Category</th>
<th>HC [g/mile] 50 kmiles</th>
<th>HC [g/mile] 100 kmiles</th>
<th>NOx [g/mile] 50 kmiles</th>
<th>NOx [g/mile] 100 kmiles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year 1993</td>
<td>0.410</td>
<td></td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>TLEV(i)</td>
<td>0.125</td>
<td>0.156</td>
<td>0.4</td>
<td>0.6</td>
</tr>
<tr>
<td>LEV(ii)</td>
<td>0.075</td>
<td>0.090</td>
<td>0.2</td>
<td>0.3</td>
</tr>
<tr>
<td>ULEV(iii)</td>
<td>0.040</td>
<td>0.055</td>
<td>0.2</td>
<td>0.3</td>
</tr>
<tr>
<td>ZEV(iv)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 1.4: HC and NOx emission limits for passenger cars

(i) TLEV: Transitional Low Emission Vehicle
(ii) LEV: Low Emission Vehicle
(iii) ULEV: Ultra Low Emission Vehicle
(iv) ZEV: Zero Emission Vehicle

In addition to upper limits, fleet-average values are also required. Figure 1.8 illustrates the California emission standards for non-methane organic gas (NMOG) emission.
Figure 1.8: Required fleet-average NMOG emission

CAFE standards requires that the sales-weighted average fuel economy limit of a car manufacturer’s vehicle fleet shall meet certain limits for passenger cars (as an example, the limit is 22.5 mile/gallon for model year 97).

Another problem arises as more and more low emission vehicles appear in the market. With the current low emission levels, a defect in the engine control system would significantly affect the overall tailpipe emission levels. Early detection and warning of such defects would allow timely repair of the defective part, so that emission level can be maintained within acceptable levels. To this end, the so-called On-Board-Diagnostics (OBD) protocol has been introduced as a part of the engine control system. OBD of emissions-relevant functions and components has been mandatory since 1996. The necessary OBD functions are listed below (relatively straightforward approaches for the electrical system diagnosis are not included).
- Misfire Detection
- Catalytic Converter Monitoring
- Evaporative Estimation from Fuel Purge Canister
- Secondary Air System Monitoring
- Fuel System Monitoring
- Check of Exhaust Gas Recirculation System
- Diagnosis of all components related to the Emission Control System

Classical control techniques have been applied to the broad band of engine control problems in the past. Since most classical control techniques are based on linear, single-input-single-output system, the resulting control performance is limited to systems with weakly coupled linear subsystems. When a strong coupling is unavoidable, an empirical calibration method leading to look up tables is adopted. Substantial time and effort needs to be expended in the compilation of these look-up tables. In spite of which the performance of the overall engine control system cannot be truly optimized. However, current and upcoming regulations for engine control systems demand that the engine control system be optimized. This allows little room for sub-optimal calibrations. However, even with highly optimized calibrations and control algorithms it might still be a challenging task to satisfy future regulatory requirements. New control configurations that include hardware changes have been recently proposed. These include: Electronic Throttle Control (ECT), Variable Valve Timing (VVT), the use of new exhaust gas sensors, lean-burn operation and direct injection engines.

ECT is based on the notion of drive-by-wire throttle. The difficulties include interpretation of the driver’s command and estimation of the engine torque for feedback. An unknown input observer (UIO) can be utilized to estimate the engine torque without adding a new sensor to the engine. Implementation of UIO for engine torque estimation has been explored and validated experimentally (Kim, Rizzoni and Wang, 1997, Moskwa, 1996).

VVT utilizes an actuator mechanism to rotate the camshaft relative to the crankshaft in order to advance the cam timing with respect to the intake and exhaust strokes of an engine. This is done in order to increase the residual charge in the cylinder (internal EGR). Combustion temperature is thus decreased,
leading to a subsequent decrease in NO\textsubscript{x} formation. In addition, reduction of hydrocarbon (HC) emission may be achieved by retaining this residual gas charge through two combustion cycles.

Lean-burn engine operation has also been proposed as an approach to improve fuel economy. This allows the engine combustion to proceed at lean air-fuel ratio at some operating condition (warm, part load condition). Variable Valve Lift (VVL) can be used for this purpose. In this approach, the valve-lift for the intake valve(s) is varied, so that the resulting cylinder charge flow has more swirl motion. Increased swirl motion enhances mixing of air and fuel within the combustion chamber allowing for lean-burn operation through charge stratification.

Direct Injection (DI) engines represent a more aggressive approach to enhancing fuel economy. In this approach, fuel is injected directly into combustion chamber. Both timing and duration of fuel injection are controlled to stratify the charge in order to stabilize combustion. Conventional injection timing (so-called early injection) is adopted for full load and wide-open throttle conditions. Late injection (fuel is injected close to top-dead-center) is utilized at a part load condition. Late injection is often used to achieve lean-burn operation.

All of these novel techniques are capable of achieving desired emission levels and performance when implemented appropriately. However, they do not come without additional complexity to the already complex system from a control point of view. VCT with electronic throttle control introduces a nonlinear, multivariable system. Lean-burn engines and direct-injection engines require electronic throttle control for efficient performance. Hence as can be seen a direct result of these technologies is an increased level of complexity in the engine control system. Therefore, a more systematic, modern multivariable control approach is necessary for the design of control and diagnostic strategies.

1.5 Conclusion

Through this chapter an attempt was made to introduce the motivation and objectives and contributions of this study. The basic terminology has been referred to and a brief introduction to fault diagnosis has been presented as a prelude to later chapters. The chapter ends with a brief review of modern and current engine control and diagnostic problems.
CHAPTER 2

REVIEW

In this chapter, model-based FDI schemes, fault tolerant control methods, and their applications to automotive engine are reviewed. The first part of the chapter is devoted to a review of model based FDI methods for both linear and nonlinear systems. The general architecture of model-based FDI system is introduced first. Next, fault isolation schemes based on multiple observers are described. Diagnostic observer approaches developed to enhance isolation capabilities and/or robustness to uncertainty are examined, considering both linear and nonlinear systems. The second part of the chapter focuses on fault tolerant control, followed by a review of an integrated design of FDI and control (so-called 4-parameter controller for linear system) method. The last part of the review focuses on the application of control and fault detection methods to automotive engines.

2.1 Model-based FDI method

The schematic diagram of model-based FDI methods is illustrated in Figure 2.1. Basically, there are two different ways of utilizing analytical redundancy: state estimation (either by parity equations or by observers), and parameter estimation. The resulting residuals have different physical meaning, depending on the way they are generated. The residuals generated using observers or parity equations are related to
the inputs and outputs of a system through a mathematical model of the plant. State estimation methods are therefore best suited for sensor/actuator fault detection, usually achieved by comparing the estimated states to the measured states. On the other hand, residuals from a parameter estimator represent the corresponding plant parameters and are better suited for component fault detection. The selection of the residual generation method, therefore, is highly dependent on the type of fault that needs to be monitored.

![Diagram of FDI process]

**Figure 2.1: Model-based FDI process**

The residual evaluator is designed to decouple residuals from the effects of noise, modeling error and to declare faulty operation based on the characteristics of the fault signature, i.e., the characteristics of the residual that is associated with a specific fault. Decoupling of noise and modeling error can be done by statistical tests such as maximum likelihood test or sequential probability ratio test or by heuristic methods such as adaptive threshold test using fuzzy logic. Typically, the FDI process is completed when the occurrence and location of a fault are identified through the process mentioned above. However, one may
desired to obtain a deeper insight into a fault to determine fault type, size and cause (which may be required for subsequent fault diagnoses). In such a case, a more rigorous approach can be applied; examples are: spectral analysis, use of a detailed plant model or application of rule-based approach.

The residual generation methods in model-based FDI approaches mainly fall into three categories even though a variety of different methods are presented in the literature. They are:

1. Parity equation residual generation (PERG)
2. Observer based residual generation (OBRG)
3. Parameter identification based residual generation (PIRG)

Model-based residual generation schemes are reviewed in the next section.

Mathematical representation of System and Fault

There are numerous ways to describe the system behavior and a fault. In general, it can be described either in an input-output relation or in a state space (in a differential equation form) as listed in Table 2.1.

<table>
<thead>
<tr>
<th>Input-output form</th>
<th>Linear Time Invariant System</th>
<th>Nonlinear System</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( y(t) = Hx(t) )</td>
<td>( y(t) = F(\bar{y}(t), \bar{u}(t)) )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( \bar{y}(t) = [y(t-\Delta t), \ldots, y(t-n\Delta t)] )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( \bar{u}(t) = [u(t), \ldots, y(t-n\Delta t)] )</td>
</tr>
<tr>
<td>State space form</td>
<td>( \dot{x}(t) = Ax(t) + Bu(t) )</td>
<td>( \dot{x}(t) = f(x(t), u(t)) )</td>
</tr>
<tr>
<td></td>
<td>( y(t) = Cx(t) )</td>
<td>( y(t) = h(x(t), u(t)) )</td>
</tr>
</tbody>
</table>

Table 2.1: Mathematical Description of System

In Table 2.1, \( u = u_o + \Delta u \), \( y^* = y + \Delta y \).
2.1.1 Parity Equation Based Residual Generation (PERG)

**Basic idea of parity check with analytical redundancy**

To begin, the simplest case of parity space approach is considered to outline the basic idea. Early contributions to this approach were made by Potter and Suman (1977), Desai and Ray (1981). Desai and Ray (1981) used the algebraic measurement equation to generate residual. The use of the basic parity relation by Desai and Ray (1981) is illustrated below. Consider the following measurement equation

\[ y^*(t) = Cx(t) + \Delta y(t). \]  \hspace{1cm} (2.5)

To find \( \Delta y(t) \in R^p \), the vector \( y^*(t) \in R^p \), can be combined with a set of linearly independent parity equations given by

\[ p(t) = V y^*(t) \]  \hspace{1cm} (2.6)

where \( p(t) \in R^{(p-n)} \) is a parity vector. The projection matrix \( V \in R^{(p-n)p} \), is determined such that

\[ VC=0 \]  \hspace{1cm} (2.7)

\[ V^t V = I_p - C(C^t C)^{-1} C^t \]  \hspace{1cm} (2.8)

\[ V^t V = I_{p-n} \]  \hspace{1cm} (2.9)

i.e. the rows of \( V \) are orthogonal, \( V \) is a null space of \( C \). Hence,

\[ p(t) = V \Delta y(t) \]  \hspace{1cm} (2.10)

And the \( p \)-dimensional residual vector \( r(t) \) is given by

\[ r(t) = y^*(t) - C \hat{x}(t) \]  \hspace{1cm} (2.11)

where \( \hat{x}(t) \) is the least square estimate of \( x \). It is given by

\[ \hat{x}(t) = (C^t C)^{-1} C^t y^*(t) \]  \hspace{1cm} (2.12)

and the parity vector \( p(t) \) and the residuals \( r(t) \) are related as

\[ r(t) = V^t p(t). \]  \hspace{1cm} (2.13)

Using the above, the FDI problem can be formulated as follows. Given \( p \) redundant measurement \( y_1^*, \ldots, y_p^* \):

1. Find an estimate \( \hat{x}(t) \) of the process from a consistent subset of measurements.

2. Identify the faulty measurement using the parity-check method explained below.
Residual generation methods using the input-output form were introduced by Gertler and Singer (1985) and Gertler (1998). A residual is simply defined as the difference between the actual output and the estimated output, the latter obtained using the measured input and the transfer function of the system:

$$r(t) = y^*(t) - \hat{y}(t) = y^*(t) - H\Delta u(t)$$  \hspace{1cm} (2.14)

Although the residual (2.14) can be obtained through a simple procedure, there are difficulties in the presence of noise and uncertainties in this method. Besides, fault isolation is not easy to implement due to insufficient design freedom. To overcome these difficulties, a weighting matrix $W$ is used for further processing of the residual obtained (2.14), thus,

$$r^*(t) = Wr(t)$$  \hspace{1cm} (2.15)

By suitable choice of the weighting matrix $W$, robustness and isolation issues can be addressed. The stability problem can be solved using the factorization methods proposed by Viswanadham and Minto (1988) and Ding and Frank (1990). Gertler et al. (1990) have proposed a number of approaches for the design of the weighting matrix $W$ such as orthogonal parity equation, directional residual and structural residual.

Chow and Willsky (1984) and Lou et al. (1986) introduced a parity equation approach for a state space form. While the robustness of the residual using this method has been studied extensively by Chow and Willsky (1984) and Lou et al. (1986), the fault isolation problem was explored by Chow and Willsky (1984), Lou et al. (1986) and Frank (1987). According to Patton and Chen (1991), the parity equation primary residual generation using the state space method is equivalent to the deadbeat observer based residual generation.

In case of nonlinear systems, a similar concept can be applied, except that the estimates of output are obtained using the nonlinear functions.
\[ r(t) = y^*(t) - \hat{y}(t) = y^*(t) - F(\hat{y}(t), \hat{u}(t)) \] (2.16)

The relationship between a fault and a residual is not linear in nonlinear systems, making the fault isolation problem more complex. Krishnaswami and Rizzoni (1997) proposed a Nonlinear Parity Equation residual Generation (NPERG) scheme using both the inverse and forward models of a system and illustrated its fault isolation capability.

2.1.2 Observer-Based Residual Generation (OBRG)

*Basic idea of observer-based residual generation*

The basic idea behind observer-based residual generation is the use of an observer to estimate the outputs of the system from the measurements or subset of the measurement. The advantages of using an observer rather than an input-output model can be summarized as:

1. Compensation of the differences in initial condition
2. Provision for design freedom in primary residual generation such as de-coupling of the faults, noise cancellation and disturbance rejection.

The type of observer structure is selected to suit the goals of the diagnostics, types of fault and nature of plant. A simple approach is the use of the Luenberger observers for deterministic systems and Kalman filters for stochastic systems. Then, the estimation error (innovations for Kalman filters) becomes the primary residual containing information on the system status. In the case of a linear system in state-space representation (equation (2.2)), a full order observer is then constructed as:

\[ \dot{x}(t) = Ax(t) + Bu_s(t) + L[y^*(t) - C\hat{x}(t)] \]
\[ = (A - LC)\hat{x}(t) + B\Delta u(t) + Ly(t) + L\Delta y(t) \] (2.17)

and the output residual is defined to be:

\[ r(t) = y(t) - C\hat{x}(t) \] (2.18)
where \( \hat{x}(t) \) is the estimate of the state, \( r(t) \) is the estimation error and \( L \) is the observer gain. Introducing the estimation error \( e(t) = x(t) - \hat{x}(t) \) and the faults \( \Delta y(t) \) and \( \Delta u(t) \), we can write state equation for the error dynamics:

\[
e(t + 1) = (A - LC)e(t) + B\Delta u(t) + L\Delta y(t)
\]  
(2.19)

and

\[
r(t) = Ce(t).
\]  
(2.20)

In equations (2.19, 2.20), it can be observed that the residual is driven only by the faults \( \Delta y(t) \) and \( \Delta u(t) \) once the estimation error due to the differences in initial condition converges to zero.

In case of stochastic systems, a Kalman filter approach can be applied in a similar way. Since the Kalman filter provides an uncorrelated residual, if the Gauss-Markov model is satisfied, this approach gives a better basis for statistical testing of residuals (for example, multiple hypothesis testing and statistical innovation test) during the decision making stage for the FDI process. However, the Kalman gain is determined such that it optimizes the estimation error in the presence of random noise. This results in a loss of freedom for design. Therefore, this approach might be less effective in isolating faults, as compared to the Luenberger observer approaches (Basseville and Nikiforov, 1996). Park, Haleri and Rizzoni (1994) have proposed a suboptimal Kalman filter approach that permits achieving fault isolation as well as noise filtering.

For nonlinear systems, the design of the observer itself is not a trivial task even when nonlinearities are known exactly. Assuming that an observer for the system can be found, the residual can be obtained by computing the difference between measured and estimated output. A simple conceptual form of nonlinear estimator is:

\[
\dot{x} = f(\hat{x}, u, \theta) + K(\hat{x}, u, \theta)(y - \hat{y})
\]  
(2.21)

\[
\hat{y} = h(\hat{x}, u, \theta)
\]  
(2.22)
where \( K(\hat{x}, u, \theta) \) is a time varying observer gain matrix designed to assure convergence. Determination of \( K(\hat{x}, u, \theta) \) is non-trivial. Assuming that \( K(\hat{x}, u, \theta) \) can be found, the state estimation error equation then becomes:

\[
e(t) = [f(x, u, \theta) - f(\hat{x}, u, \theta) - K(\hat{x}, u, \theta)]e(t)
\]  \hspace{1cm} (2.23)

The estimation error of the output is

\[
e(t) = y^*(t) - \hat{y}(t) = h(x, u + \Delta u, \theta) + \Delta y - h(\hat{x}, u, \theta)
\]  \hspace{1cm} (2.24)

which can also be defined as the primary residual. Note that the additive sensor fault is linear but the actuator fault appears as nonlinear. After the nontrivial observer design procedure, the fault isolation process also requires analysis of the nonlinear behavior, making the whole FDI process for nonlinear systems very challenging.

2.2 Fault Isolation using Multiple Observers

The ideas of the preceding section do not explain how different faults can be isolated. In this section, we show that approaches based on multiple observers permit distinguishing among different faults. Clark (1978, 1987) introduced two different schemes to isolate sensor faults using a bank of observers. The two methods are named Dedicated Observer Scheme (Clark, 1978) and Generalized Observer Scheme (Clark, 1987). The use of multiple observers has explored further by Patton and Frank (1994). However, none of these methods are capable of isolating input (actuator) faults in principle. Krishnaswami and Rizzoni (1994, 1996) proposed the Nonlinear Parity Equation Residual Generation (NPERG) scheme focused on fault isolation for a general nonlinear system. They showed that both single and simultaneous faults can be isolated by appropriate configuration of observers. They also applied this approach with an input-output system representation using NARMAX (Nonlinear Auto Regressive Moving Average with Exogenous Input) model of the system. This approach make use of both forward model (state observer) and inverse model (input observer) to isolate faults in a nonlinear system and shows promising results for
Various application (Krishnaswami and Rizzoni (1994), Krishnaswami et al. (1995, 1996), Kim et al. (1997)).

Three of the schemes mentioned above, the Dedicated Observer Scheme, the Generalized Observer Scheme and the NPERG scheme are illustrated in the following sections.

2.2.1 Dedicated Observer Scheme

For the instrument fault detection, the use of dedicated observer for each sensor was introduced by Clark (1978). In this Dedicated Observer Scheme, each observer uses a measurement from a different sensor to estimate the output of the system. With this scheme, multiple simultaneous faults in sensors can be principally detected and isolated with the aid of threshold logic. The schematic of this approach is shown in the Figure 2.2.

![Figure 2.2: Schematic diagram of Dedicated Observer Scheme](image)

If all the sensors are free of faults then all $m$ of the estimated state vectors $\hat{x}^{(i)} (i = 1, 2, \ldots, m)$ will be identical to $x(t)$. If the sensor for $y_i$ is faulty, only the estimated state from observer $(1), \hat{x}^{(1)}$ will be different from the $x(t)$ and all other estimates $\hat{x}^{(2)}, \hat{x}^{(3)}, \ldots, \hat{x}^{(m)}$ will be remain identical to $x(t)$. And the fault of the sensor can be isolated through a comparison of $\hat{x}^{(1)}$ with $\hat{x}^{(2)}, \hat{x}^{(3)}, \ldots, \hat{x}^{(m)}$. 
Design Procedure - Dedicated Observer Scheme

The design procedure of the Dedicated Observer Scheme is illustrated with an example. Consider a system in the following form.

\[
\begin{align*}
\dot{x} &= Ax + Bu \\
y &= Cx
\end{align*}
\]

where \( x \in \mathbb{R}^3 \), \( u \in \mathbb{R}^l \) and \( y = [y_1 \ y_2 \ y_3]^T \). Our objective is to isolate fault for each sensor \( y_1 \), \( y_2 \) and \( y_3 \). For notational convenience, we name sensor #1 for the sensor to measure \( y_1 \), sensor #2 to the sensor measuring \( y_2 \), and so forth. First decompose the output space as

\[
\begin{align*}
y_{(1)} &= C^{(1)} x = y_1 \\
y_{(2)} &= C^{(2)} x = y_2 \\
y_{(3)} &= C^{(3)} x = y_3 \\
C &= C^{(1)} \quad C^{(2)} \quad C^{(3)}
\end{align*}
\]

As long as \( (A, C_1) \), \( (A, C_2) \) and \( (A, C_3) \) pairs are observable, three observers can be designed using each output separately as follows.

**Observer (1)**

\[
\hat{x}^{(1)} = A\hat{x}^{(1)} + Bu + L_1 (\hat{y}_1 - y_1)
\]

**Observer (2)**

\[
\hat{x}^{(2)} = A\hat{x}^{(2)} + Bu + L_2 (\hat{y}_2 - y_2)
\]

**Observer (3)**

\[
\hat{x}^{(3)} = A\hat{x}^{(3)} + Bu + L_3 (\hat{y}_3 - y_3)
\]

Note that the observer (1) is free of fault in both sensor #2 and #3, and the observer (2) is free of fault in both sensor #1 and sensor #3, and the observer (3) is free of fault in both sensor #1 and sensor #2. A primary residual can be defined using the estimates from each observer as follows.

\[
\begin{align*}
 r^{(1)} &= \hat{x}^{(1)} - \hat{x}^{(2)} \\
r^{(2)} &= \hat{x}^{(2)} - \hat{x}^{(3)} \\
r^{(3)} &= \hat{x}^{(3)} - \hat{x}^{(1)}
\end{align*}
\]

(2.27)

Characteristics of the residuals in the presence of a single sensor fault are listed in Table 2.2. Any sensor fault can be isolated by examining the behavior of the residual listed in Table 2.2. For example, when the sensor #2 is faulty, all components of \( r^{(2)} \) will have zero value since \( \hat{x}^{(1)} = \hat{x}^{(3)} \). This is due to the fact that
both observer (1) and observer (3) are free of fault in sensor #2. The same logic can be applied for fault in
the sensor #1 and the sensor #3.

<table>
<thead>
<tr>
<th>Fault in Sensor #1</th>
<th>Fault in Sensor #2</th>
<th>Fault in Sensor #3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_1^{(1)} = \hat{x}_1^{(1)} - \hat{x}_1^{(2)} \neq 0$</td>
<td>$r_1^{(1)} = \hat{x}_1^{(1)} - \hat{x}_1^{(2)} \neq 0$</td>
<td>$r_1^{(1)} = \hat{x}_1^{(1)} - \hat{x}_1^{(2)} = 0$</td>
</tr>
<tr>
<td>$r_2^{(1)} = \hat{x}_2^{(1)} - \hat{x}_2^{(2)} \neq 0$</td>
<td>$r_2^{(1)} = \hat{x}_2^{(1)} - \hat{x}_2^{(2)} \neq 0$</td>
<td>$r_2^{(1)} = \hat{x}_2^{(1)} - \hat{x}_2^{(2)} = 0$</td>
</tr>
<tr>
<td>$r_3^{(1)} = \hat{x}_3^{(1)} - \hat{x}_3^{(2)} \neq 0$</td>
<td>$r_3^{(1)} = \hat{x}_3^{(1)} - \hat{x}_3^{(2)} \neq 0$</td>
<td>$r_3^{(1)} = \hat{x}_3^{(1)} - \hat{x}_3^{(2)} = 0$</td>
</tr>
<tr>
<td>$r_1^{(2)} = \hat{x}_1^{(2)} - \hat{x}_1^{(3)} = 0$</td>
<td>$r_1^{(2)} = \hat{x}_1^{(2)} - \hat{x}_1^{(3)} \neq 0$</td>
<td>$r_1^{(2)} = \hat{x}_1^{(2)} - \hat{x}_1^{(3)} \neq 0$</td>
</tr>
<tr>
<td>$r_2^{(2)} = \hat{x}_2^{(2)} - \hat{x}_2^{(3)} = 0$</td>
<td>$r_2^{(2)} = \hat{x}_2^{(2)} - \hat{x}_2^{(3)} \neq 0$</td>
<td>$r_2^{(2)} = \hat{x}_2^{(2)} - \hat{x}_2^{(3)} \neq 0$</td>
</tr>
<tr>
<td>$r_3^{(2)} = \hat{x}_3^{(2)} - \hat{x}_3^{(3)} = 0$</td>
<td>$r_3^{(2)} = \hat{x}_3^{(2)} - \hat{x}_3^{(3)} \neq 0$</td>
<td>$r_3^{(2)} = \hat{x}_3^{(2)} - \hat{x}_3^{(3)} \neq 0$</td>
</tr>
<tr>
<td>$r_1^{(3)} = \hat{x}_1^{(3)} - \hat{x}_1^{(3)} = 0$</td>
<td>$r_1^{(3)} = \hat{x}_1^{(3)} - \hat{x}_1^{(3)} \neq 0$</td>
<td>$r_1^{(3)} = \hat{x}_1^{(3)} - \hat{x}_1^{(3)} \neq 0$</td>
</tr>
<tr>
<td>$r_2^{(3)} = \hat{x}_2^{(3)} - \hat{x}_2^{(3)} \neq 0$</td>
<td>$r_2^{(3)} = \hat{x}_2^{(3)} - \hat{x}_2^{(3)} = 0$</td>
<td>$r_2^{(3)} = \hat{x}_2^{(3)} - \hat{x}_2^{(3)} = 0$</td>
</tr>
<tr>
<td>$r_3^{(3)} = \hat{x}_3^{(3)} - \hat{x}_3^{(3)} \neq 0$</td>
<td>$r_3^{(3)} = \hat{x}_3^{(3)} - \hat{x}_3^{(3)} = 0$</td>
<td>$r_3^{(3)} = \hat{x}_3^{(3)} - \hat{x}_3^{(3)} = 0$</td>
</tr>
</tbody>
</table>

Table 2.2: Characteristics of the primary residual for Dedicated Observer Scheme example

A secondary residual can be defined for further simplification as follows.

$$e = \begin{bmatrix} e^{(1)} \\ e^{(2)} \\ e^{(3)} \end{bmatrix}$$

(2.28)

$$e^{(i)} = \sum_{k=1}^{3} \alpha_i r_i^{(i)}$$

(2.29)

where $\alpha_i$ are scaling coefficients (e.g. to normalization coefficients). The secondary residuals for each
sensor fault are listed in Table 2.3.

<table>
<thead>
<tr>
<th></th>
<th>$e^{(1)}$</th>
<th>$e^{(2)}$</th>
<th>$e^{(3)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Fault in sensor for $y_1$</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Fault in sensor for $y_2$</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Fault in sensor for $y_3$</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 2.3: Signature for the secondary residual $e$
As seen in Table 2.3, the secondary residual shows a unique signature for each sensor fault. However, at least three outputs are required for a single sensor fault isolation (note that the primary residual becomes \( r = \hat{x}^{(1)} - \hat{x}^{(2)} \) and it has the same characteristics for each sensor fault). Multiple sensor faults can be isolated with more than three measurements by comparing the residual and applying *majority vote* logical comparison. As long as the observability condition is satisfied, the *Dedicated Observer Scheme* provides a strong fault isolation capability as illustrated above.

### 2.2.2 Generalized Observer Scheme

An alternative approach, called *Generalized Observer Scheme*, was also proposed by Clark (1987). This scheme uses an observer dedicated to a specific sensor. This observer uses all measurements except the measurement from the designated sensor (see Figure 2.3). This scheme provides single sensor fault detection ability with robustness to unknown input.

![Diagram of Generalized Observer Scheme](image)

**Figure 2.3: Schematic diagram of Generalized Observer Scheme**

*Design Procedure - Generalized Observer Scheme*

The design procedure of the *Generalized Observer Scheme* is illustrated with an example. Consider the same system as in the case with the dedicated observer scheme. Again our objective is to isolate fault for each sensor \( y_1, y_2 \) and \( y_3 \). In *Generalized Observer Scheme*, we try to eliminate a sensor from the measurement. Therefore, we decompose the output space as
\[ y^{(1)} = C^{(1)} x = \begin{bmatrix} y_2 \\ y_3 \end{bmatrix}, \quad y^{(2)} = C^{(2)} x = \begin{bmatrix} y_1 \\ y_3 \end{bmatrix}, \quad y^{(3)} = C^{(3)} x = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \] (2.30)

If all the pair \((A,C^{(1)})\), \((A,C^{(2)})\) and \((A,C^{(3)})\) are observable, three observers can be designed using each output separately as follows.

\[
\begin{align*}
\hat{x} &= A\hat{x} + Bu + L_1(\hat{y}^{(1)} - y^{(1)}) & \hat{x} &= A\hat{x} + Bu + L_3(\hat{y}^{(2)} - y^{(2)}) & \hat{x} &= A\hat{x} + Bu + L_3(\hat{y}^{(3)} - y^{(3)}) \\
\end{align*}
\]

Each observer will give the estimates of the outputs and let the estimates be \(\hat{y}^{(1)}, \hat{y}^{(2)}, \hat{y}^{(3)}\) where

- Estimated output from observer (1): \(\hat{y}^{(1)} = \begin{bmatrix} \hat{y}_2^{(1)} \\ \hat{y}_3^{(1)} \end{bmatrix} \)
- Estimated output from observer (2): \(\hat{y}^{(2)} = \begin{bmatrix} \hat{y}_1^{(2)} \\ \hat{y}_3^{(2)} \end{bmatrix} \)
- Estimated output from observer (3): \(\hat{y}^{(3)} = \begin{bmatrix} \hat{y}_1^{(3)} \\ \hat{y}_2^{(3)} \end{bmatrix} \).

If the measurements are linearly independent each other, the observer (1) is free of fault in sensor #1, the observer (2) is free of fault in sensor #2 and so on. Now define a primary residuals as follows.

\[
\begin{align*}
\rho^{(1)} &= y^{(1)} - \hat{y}^{(1)} \\
\rho^{(2)} &= y^{(2)} - \hat{y}^{(2)} \\
\rho^{(3)} &= y^{(3)} - \hat{y}^{(3)} \\
\end{align*}
\] (2.31)

Characteristics of the primary residuals in the presence of a single sensor fault are listed in Table 2.4.

<table>
<thead>
<tr>
<th></th>
<th>Fault in Sensor for (y_1)</th>
<th>Fault in Sensor for (y_2)</th>
<th>Fault in Sensor for (y_3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(r^{(1)})</td>
<td>(r_2^{(1)} = y_2 - \hat{y}_2^{(1)} = 0)</td>
<td>(r_2^{(2)} = y_2 - \hat{y}_2^{(2)} = 0)</td>
<td>(r_2^{(3)} = y_2 - \hat{y}_2^{(3)} \neq 0)</td>
</tr>
<tr>
<td></td>
<td>(r_3^{(1)} = y_3 - \hat{y}_3^{(1)} = 0)</td>
<td>(r_3^{(2)} = y_3 - \hat{y}_3^{(2)} \neq 0)</td>
<td>(r_3^{(3)} = y_3 - \hat{y}_3^{(3)} = 0)</td>
</tr>
<tr>
<td>(r^{(2)})</td>
<td>(r_1^{(2)} = y_1 - \hat{y}_1^{(2)} = 0)</td>
<td>(r_1^{(2)} = y_1 - \hat{y}_1^{(2)} = 0)</td>
<td>(r_1^{(3)} = y_1 - \hat{y}_1^{(3)} \neq 0)</td>
</tr>
<tr>
<td></td>
<td>(r_3^{(2)} = y_3 - \hat{y}_3^{(2)} \neq 0)</td>
<td>(r_3^{(2)} = y_3 - \hat{y}_3^{(2)} = 0)</td>
<td>(r_3^{(3)} = y_3 - \hat{y}_3^{(3)} = 0)</td>
</tr>
<tr>
<td>(r^{(3)})</td>
<td>(r_1^{(3)} = y_1 - \hat{y}_1^{(3)} = 0)</td>
<td>(r_1^{(3)} = y_1 - \hat{y}_1^{(3)} \neq 0)</td>
<td>(r_1^{(3)} = y_1 - \hat{y}_1^{(3)} = 0)</td>
</tr>
<tr>
<td></td>
<td>(r_2^{(3)} = y_2 - \hat{y}_2^{(3)} \neq 0)</td>
<td>(r_2^{(3)} = y_2 - \hat{y}_2^{(3)} = 0)</td>
<td>(r_2^{(3)} = y_2 - \hat{y}_2^{(3)} = 0)</td>
</tr>
</tbody>
</table>

Table 2.4: Characteristics of the primary residual for Generalized Observer Scheme example
Now, a secondary residual can be defined as a function of the primary residual. For example, let us define the secondary residual as

$$
e = \begin{bmatrix} e^{(1)} \\ e^{(2)} \\ e^{(3)} \end{bmatrix}$$

$$e^{(i)} = \sum_{k=1}^{n} \alpha_k r_k^{(i)}$$

(2.32)

(2.33)

where $\alpha_k$ are scaling coefficients (e.g. to normalization coefficients). The secondary residuals for each sensor fault become to have unique signature as shown in Table 2.5.

<table>
<thead>
<tr>
<th></th>
<th>$e^{(1)}$</th>
<th>$e^{(2)}$</th>
<th>$e^{(3)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Fault in sensor for $y_1$</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Fault in sensor for $y_2$</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Fault in sensor for $y_3$</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 2.5: Secondary residual characteristics for Generalized Observer Scheme

As an extension to this scheme, Min (1987) proposed the Binary Phase Detection Filter scheme that uses two observers. In this approach, the measurement vector is divided into two groups and each observer is designed such that it only uses a group of measurement vector. As a result, any faulty sensor in one group of measurement vector can be isolated.

2.2.3. Nonlinear Parity Equation Residual Generation (NPERG) Scheme

The NPERG scheme proposed by Krishnaswami and Rizzoni (1994) was developed with a focus on (i) broad applicability (both nonlinear and linear systems) and (ii) the use of a general fault model. This scheme provides a generalized approach to the fault isolation problem, details of this method are examined below.

Consider the dynamic system represented in differential equation form.

$$x = f(x, u, \theta) = f(x, u_0 + \Delta u, \theta_0 + \Delta \theta)$$

(2.34)

$$y^* = h(x, u, \theta) + \Delta y = h(x, u_0 + \Delta u, \theta_0 + \Delta \theta) + \Delta y$$

(2.35)
Define a fault vector as follows.

\[
d(t) = \begin{bmatrix} \Delta u(t) \\ \Delta y(t) \end{bmatrix}
\]  

(2.36)

Note that if \( u \in \mathbb{R}^n \) and \( y \in \mathbb{R}^p \) obviously, \( d \in \mathbb{R}^{n+p} \).

From a physical understanding of dynamic system behavior, a fault can be detected if it has a unique non-zero effect on the available measurements. In essence, an FDI scheme is a map (linear or nonlinear) that acts upon the measurements obtained from the system under study and produces an output that is unique for each combination of faults that can occur in the system. Uniqueness guarantees the isolability of faults. Now, the object is to design a diagnostic module that can detect and isolate faults using the measurements.

**Design Procedure - NPERG**

The task of any fault detection process is to isolate the faulty outputs and inputs. This is to be done using some or all of the available measurements, while checking for consistency of the resulting solution. The following section discusses the NPERG design procedure for the system given by (2.34) and (2.35). The detail derivation and proof of NPERG scheme is described in Krishnaswami (1996).

**Step 1: Configuration of residual generator (see Figure 2.4)**

Assume that the first \( w \) elements of the output vector, \( y_f \) and the first \( v \) elements of the input vector, \( u_f \) are faulty and the other elements (\( y_{nf} \) and \( u_{nf} \)) are not faulty. Then, the non-faulty output \( y_{nf} \) can be used to estimate the input \( u_f \) that are suspected to be faulty.

Now, construct an inverse model that estimates the input \( u_f \) from a subset of outputs \( y_{nf} \) that are assumed to be non-faulty. We can then estimate the outputs \( y_f \) that are assumed faulty using the estimated input and the forward model.
This provides us with estimates of all the faults, under the assumption that a particular set (the first \( v \) inputs \( u_f \) and first \( w \) outputs \( y_f \)) of variables are faulty.

![Diagram of Residual Generator Construction](image)

**Figure 2.4: Residual generator construction**

**Step 2: Consistency Check**

Find estimates of all the other measured outputs \( y_{nf} \) using the estimated value of the assumed faulty input \( \hat{u}_f \), the nominal values of the other input \( u_{nf} \) and the forward model.

If the assumed set of faults actually occur, then the measured and the estimated values of the assumed non-faulty outputs (\( y_{nf} \) outputs of the forward model for this step) should match (within measurement and modeling errors). Isolation is thus completed since the estimates obtained in step 1 are also the correct estimates of the faulty variables (\( y_f \) and \( u_f \)). If the values do not match, then the consistency check has failed.

**Step 3:** If the consistency check fails, then it is necessary to assume a different combination of faulty elements and repeat step 2. This is done by constructing the estimator described in step 1 and the forward model of step 2 for each possible combination of faults.

Note that the consistency check described in step 2 for the above procedure is what is usually referred to in fault detection literature as *residual generation* and each set of inverse and forward models from step 1 and step 2 is referred to as a *residual generator.*
Example: 2 inputs and 2 outputs system

Assume that a system has 2 inputs and 2 outputs and there are two possible faults present in the system: 1 input fault and 1 output fault. Further let the first element of the output vector, \( y_1 \) and the first element of the input vector, \( u_1 \) be faulty. The other elements (\( y_2 \) and \( u_2 \)) are not faulty. The NPERG can be designed to detect and isolate the faults as shown in Figure 2.5.

![Figure 2.5: Residual generator construction for example](image)

Case i) If only \( y_1 \) has fault, then the residuals will have the following values

<table>
<thead>
<tr>
<th>Residuals</th>
<th>threshold test output</th>
</tr>
</thead>
<tbody>
<tr>
<td>( r_{n_1} = u_1 - \hat{u}_1 = 0 )</td>
<td>0</td>
</tr>
<tr>
<td>( r_{n_1} = y_1 - \hat{y}_1 \neq 0 )</td>
<td>1</td>
</tr>
<tr>
<td>( r_{n_2} = y_2 - \hat{y}_2 = 0 )</td>
<td>0</td>
</tr>
</tbody>
</table>

Case ii) If only \( u_1 \) has fault, then the residuals will have the following values

<table>
<thead>
<tr>
<th>Residuals</th>
<th>threshold test output</th>
</tr>
</thead>
<tbody>
<tr>
<td>( r_{n_1} = u_1 - \hat{u}_1 \neq 0 )</td>
<td>1</td>
</tr>
<tr>
<td>( r_{n_1} = y_1 - \hat{y}_1 = 0 )</td>
<td>0</td>
</tr>
<tr>
<td>( r_{n_2} = y_2 - \hat{y}_2 = 0 )</td>
<td>0</td>
</tr>
</tbody>
</table>

Case iii) If both \( y_1 \) and \( u_1 \) have faults, then the residuals will have the following values

<table>
<thead>
<tr>
<th>residuals</th>
<th>threshold test output</th>
</tr>
</thead>
<tbody>
<tr>
<td>( r_{n_1} = u_1 - \hat{u}_1 \neq 0 )</td>
<td>1</td>
</tr>
<tr>
<td>( r_{n_1} = y_1 - \hat{y}_1 \neq 0 )</td>
<td>1</td>
</tr>
<tr>
<td>( r_{n_2} = y_2 - \hat{y}_2 = 0 )</td>
<td>0</td>
</tr>
</tbody>
</table>
The fault signature for the input $u_i$ and output $y_i$ can be summarized by Table 2.6.

<table>
<thead>
<tr>
<th>Fault</th>
<th>$r_{u1}$</th>
<th>$r_{y1}$</th>
<th>$r_{y2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Actuator #1 ($u_i$)</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Sensor #1 ($y_1$)</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Actuator and Sensor#1 ($u_1, y_1$)</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Fault in other component</td>
<td>-</td>
<td>-</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 2.6: Fault signature for NPERG example

2.3 Diagnostic Observers

The application of the model-based FDI schemes introduced so far is straightforward and reasonable performance can be achieved if the model of the monitored system is accurate and all the characteristics are known a priori. However, modeling errors and disturbances are unavoidable in practice and the performance of a FDI scheme is severely limited by these uncertainties. Robustness, in the context of model-based FDI, means insensitivity to the uncertainties (modeling error and disturbances) while providing adequate sensitivity to the faults. Therefore, in a practical situation, the challenge is to design a robust residual generator.

The robustness problem has been recognized early on, and various approaches have been suggested over the years. To solve one class of robustness problem, the disturbance-decoupling concept has been used. In this approach, all the uncertainties (modeling error and disturbance) are represented by disturbances. The principle idea of disturbance decoupling can be summarized as follows:

$$\dot{x}(t) = Ax(t) + Bu(t) + E_1 d(t) + f(t)$$  \hspace{1cm} (2.37)

$$y(t) = Cx(t) + Du(t) + E_2 d(t) + f(t)$$  \hspace{1cm} (2.38)

where $E_1 d(t)$ and $E_2 d(t)$ represent the uncertainties acting on the system. The disturbance $d(t)$ is assumed to be unknown and the structure of the disturbance distribution matrices $E_1$ and $E_2$ is assumed to be known.
(i.e., the directions represented by the columns of these matrices are known). The input-output form of the above system is

\[ y(t) = G_x u(t) + G_x d(t) + G_f f(t). \]  

(2.39)

And the residual generator has the form

\[ r(t) = H_x u(t) + H_y y(t). \]  

(2.40)

Substitute (2.39) into (2.40). Since \( H_x(s) + H_y(s)G_x(s) = 0 \),

\[ r(t) = H_x G_f f(t) + H_y G_d d(t). \]  

(2.41)

Therefore, to make the residual independent of the disturbances, it is required to design the residual generator to satisfy the following relation.

\[ H_x G_d = 0 \]  

(2.42)

The above derivation illustrates a method for disturbance decoupling from residuals.

2.3.1 Diagnostic Observer for Linear System

The idea of disturbance decoupling has been adopted in several robust FDI approaches. The three main categories using this approach are:

1. Detection filter approach
2. Eigenstructure assignment approach
3. Unknown input observer approach

The principles of the above three approaches are reviewed in the following subsections.

\textit{Detection filter}

Fixed directional residual generation is the basis of the detection filter approach that was first developed by Beard (1971) and Jones (1973). The detection filter is a type of diagnostic observer whose gain is selected such that the residual vector lies in a well-defined direction in the output space in the presence of a fault. Massoumnia (1986) reformulated the design procedure in a geometrical framework, and White and Speyer (1987), and Wilbers and Speyer (1989) further expanded to provide a solution using
eigenstructure assignment. To provide an insight to this approach, the work done by Wilbers and Speyer (1989) is illustrated below. Consider a system with fault in state space form as

\[
\dot{x}(t) = Ax(t) + Bu(t) + k_i p_i(t) 
\]  
\[ y(t) = Cx(t) + k_i \Delta y_i 
\]  

(2.43)

(2.44)

where \( k_i \) is an \( n \times 1 \) design fault direction, \( i=1,2,\ldots,r \), \( r \) is the number of fault directions and \( f_i(t) \) are the vector of faults \( (p_i, \Delta y_i) \). The corresponding observer is

\[
\dot{\hat{x}}(t) = (A - LC)\hat{x}(t) + Bu(t) + Ly(t) 
\]  
\[ \hat{y}(t) = C\hat{x}(t). 
\]  

(2.45)

(2.46)

With an actuator or component fault, the state estimation error equation \( e(t) = x(t) - \hat{x}(t) \) becomes

\[
\dot{e}(t) = (A - LC)e(t) + k_i p_i(t) 
\]  
\[ r(t) = y(t) - \hat{y}(t) = Ce(t). 
\]  

(2.47)

(2.48)

And for a sensor fault, we have

\[
\dot{e}_j(t) = (A - LC)e_j(t) + h_j \Delta y_j(t) 
\]  
\[ r_j(t) = y(t) - \hat{y}(t) = Ce_j(t) + k_j \Delta y_j(t) 
\]  

(2.49)

(2.50)

where \( h_j \) is the \( j \)-th column of the detection filter gain matrix. This system can be reduced to

\[
\dot{e}_j(t) = (A - LC)e_j(t) - k_j^* \Delta y_j(t) + f_j^* 
\]  
\[ r_j(t) = Ce_j(t). 
\]  

(2.51)

(2.52)

where \( k_j^* = A f_j^* - \alpha f_j^* \) with an arbitrary scalar \( \alpha \). \( f_j^* \) is the fault direction associated with \( i \)-th sensor such that \( C f_j^* = k_j \). For isolation, the observer gain \( L \) is selected such that the actuator fault or component fault residuals lie in the direction of \( C k_i \), and sensor fault residuals lie in the plane defined by \( C f_i^* \) and \( C k_i^* \). In summary, the fault detection filter generates residuals such that the residual due to a specific fault \( f_i \) is constrained to a single line or plane in the residual space independent of the mode of \( f_i \). This is accomplished by proper choice of \( L \).
Beard (1971) and Jones (1973) used right eigenvector assignment to construct residual generators. The biggest drawback of their approach is that the design procedure is based on a generator vector, which is unknown. The spectral approach proposed by White and Speyer (1987) considers the eigenvalue and eigenvector problem directly, without requiring the selection of a generator vector. The main limitation of this method is that the eigenvalues and the eigenvectors of both the detection and the completion space must be specified simultaneously. Park (1991) showed that the eigenstructure assignment problem can be treated as two independent observer design problems, one for the detection space and one for the completion space. Park et al. (1994) and Park and Rizzoni (1994), used the approach presented in Park (1991) to formulate and solve the problem of designing a detection filter that is optimal in the sense of being robust to process and measurement noises.

*Eigenstructure assignment (ESA)*

It is well a known fact in linear state space feedback theory that the closed loop system characteristics depends entirely on the eigenvalue and eigenvector (both left and right) of a system. Patton et al. (1989) and Patton and Chen (1990) demonstrate the use of eigenstructure assignment for observer design aimed at FDI application. Both left and right eigenvector assignment may be used to design robust observer to generate the residual that maintains a fixed direction in the residual space. The design principle of the observer and its disturbance rejection property are described below. Assume that we can model the faults and also disturbance to be detected by additive vectors (and also disturbances); then

\[
\dot{x}(t) = A(t)x(t) + Bu(t) + Eq(t) + KP(t)
\]

(2.53)

\[
y(t) = Cx(t) + Du(t) + P(t)
\]

(2.54)

where \(x \in \mathbb{R}^n, u \in \mathbb{R}^r, y \in \mathbb{R}^r\) and \(P_a\) is the vector of actuator faults, \(P_i\) is the vector of sensor faults and \(q\) denotes the disturbances. This model only includes faults and disturbances that are additive. Multiplicative faults are not considered here. Construct an observer as follows.

\[
\dot{\hat{x}}(t) = (A - L C)\hat{x}(t) + (B - L D)u(t) + Ly(t)
\]

(2.55)
\[ \hat{y}(t) = C\hat{x}(t) + Du(t). \]  

(2.56)

The estimation error dynamics are
\[
e(t) = x(t) - \hat{x}(t)
\]

(2.57)

\[
\dot{e}(t) = (A - LC)e(t) + Eq(t) + KP_e(t) - LP_e(t)
\]

(2.58)

\[
= A_e e(t) + Eq(t) + KP_e(t) - LP_e(t)
\]

Define the \( p \)-dimensional residual as
\[
r(t) = W[y(t) - \hat{y}(t)]
\]

(2.59)

\[
= W[Cx(t) + Du(t) + P_e(t) - C\hat{x}(t) - Du(t)]
\]

\[
= W[e(t) + P_e(t)]
\]

\[
= WCe(t) + WP_e(t)
\]

The transfer function form of equation (2.58) gives
\[
e(t) = [\phi I - A_o]^{-1}Eq(t) + [\phi I - A_o]^{-1}KP_e(t) - [\phi I - A_o]^{-1}LP_e(t)\]

(2.60)

By substituting (2.60) into (2.59), we obtain
\[
r(t) = [W - WCE]^{-1} P_e(t) + WC[\phi I - A_o]^{-1}KP_e(t) + WC[\phi I - A_o]^{-1}Eq(t)\]

(2.61)

To decouple the residual from the effect of the disturbances, the following equation needs to be satisfied.
\[
WCE\left[\phi I - A_o\right]^{-1} E = 0
\]

(2.62)

The following eigenvector assignment algorithm can be applied to achieve this decoupling.

Case I) rank(\( CE \))=\(< m \): left eigenvector assignment

i) \( WCE=0 \)

ii) all rows of the matrix \( H=WCE \) are the left eigenvector of \( A_o \) corresponding to any eigenvalue of \( A_o \).

Case II) otherwise: right eigenvector assignment

i) \( WCE=0 \)

ii) all columns of the matrix \( E \) are the right eigenvector of \( A_o \) corresponding to any eigenvalue of \( A_o \).
Unknown input observer (UIO)

The application of the unknown input observer to the FDI problem has been extensively studied by many researchers such as Watanabe and Himmelblau (1982), Viswanadh and Srichander (1987) and Frank (1991). The task of the unknown input observer in FDI application is to keep the residuals independent of system states and dependent on the faults. This can be achieved using the unknown input observer by forcing the state estimation error to become independent of the uncertainties. Since the estimation error is decoupled from the uncertainties, the resulting residual is also decoupled from uncertainties.

The UIO design as proposed by Viswanadh and Srichander (1987) and Hou and Müller (1992), consists of transforming the system equation, such that the state vector can be divided into two parts - a part that can be directly obtained from the measurements, and the other consisting of the states to be estimated. A reduced order observer can be designed to estimate these states, and the observer gains are selected so that they decouple the observer dynamics from the unknown input. The conditions under which such a design is possible and the design procedure are given below. Consider the dynamic system represented by the following state space equations.

\[
\dot{x}(t) = Ax(t) + Bu(t) + Ed(t) + Kf(t) \quad (2.63)
\]
\[
y(t) = Cx(t) + Gf(t) \quad (2.64)
\]

where \(x \in \mathbb{R}^n\) is the state, \(u \in \mathbb{R}^p\) is a vector of known inputs, \(d(t) \in \mathbb{R}^r\) is a vector of unmeasured disturbance inputs, \(f(t) \in \mathbb{R}^s\) is a vector of unknown fault and \(y \in \mathbb{R}^m\) is a vector of measurements. The following observer design results in an observer that decouples the state estimates from the effect of the disturbance while maintaining sensitivity to the fault inputs.

If \(C\) has full row rank the system (2.63)-(2.64) can then be transformed into the following form,

\[
\begin{bmatrix}
\dot{y}(t) \\
\dot{w}(t)
\end{bmatrix} =
\begin{bmatrix}
A_{11} & A_{12} \\
A_{21} & A_{22}
\end{bmatrix}
\begin{bmatrix}
y(t) \\
w(t)
\end{bmatrix} +
\begin{bmatrix}
b_1 \\
b_2
\end{bmatrix} u(t) +
\begin{bmatrix}
E_1 \\
E_2
\end{bmatrix} d(t). \quad (2.65)
\]

Now an \((n-p)\)th order observer can be constructed to estimate \(w(t)\). This observer is given by,
\begin{align*}
\dot{z}(t) &= (A_{22}L - A_{12})z(t) + (A_{21}L - A_{11})y(t) + (b_{z}Lb_{y})u(t) + (E_{z}LE_{y})d(t) \\
\dot{\hat{\omega}}(t) &= z(t) + Ly(t).
\end{align*}

Since \(d(t)\) is not available for measurement the observer can be implemented, if

(i) \(E_{2} - LE_{1} = 0\), and

(ii) \((A_{22}L - A_{12})\) has stable eigenvalues.

The existence conditions for \(L\) such that the above conditions are satisfied are,

(i) \((A,C)\) is observable and

(ii) \(\text{rank}(CE) = \text{rank}(E) = r\) and \(m \geq r\) (for proof see Kudva et al., 1980)

The eigenspectrum of \((A_{22}L - A_{12})\) must contain the invariant zeros of the triple \((A,C,E)\) (see Kudva et al., 1980). Then it can be shown that the error dynamic equation in the absence of any faults is,

\begin{equation}
\dot{e}(t) = (A_{22}L - A_{12})e(t) + \dot{\hat{\omega}}(t).
\end{equation}

In the presence of a failure the error dynamics become

\begin{equation}
\dot{e}(t) = (A_{22}L - A_{12})e(t) + [L \ I]Kf(t)
\end{equation}

where \(I\) is an identity matrix of appropriate dimension, and the error is non zero provided, \([-L \ I]K \neq 0\).

Thus the error residual is sensitive to the fault but not to the unknown input.

The unknown input method of residual generation can also be implemented using the generalized observer scheme as described by Frank and Wünnenberg (1989). Hou and Müller (1991) extended the UIO to systems where the unknown input enters the measurement equation. The UIO may then be used to isolate actuator faults by constructing a bank of observers, each of which treats some subset of the faulty inputs as unknown. If a fault is present, only those observers that do not treat this fault as an unknown input produce
nonzero residuals. A proper choice of the number of observers, and of a set of faults decoupled from the residual, results in a unique signature corresponding to each fault.

Rather than decoupling the unknown input from the system during estimation, it is also possible to estimate the unknown input using a similar approach. In this approach, the UIO estimates the unknown states and unknown inputs using the output $y$ and the known input $u$. To examine this concept, the design procedure developed by Hou and Müller (1992) is shown below. Consider a linear time-invariant system described by

$$\dot{x}(t) = Ax(t) + Bu(t) + Dd(t)$$  \hspace{1cm} (2.70)

$$y(t) = Cx(t)$$  \hspace{1cm} (2.71)

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^r$, $d \in \mathbb{R}^e$, $y \in \mathbb{R}^m$ are the state vector, the known input vector, the unknown input vector, and the measurement vector, respectively. $A$, $B$, $D$ and $C$ are constant matrices of appropriate dimensions. The UIO for a linear time invariant system can be formulated as follows. The design procedure given below was developed mainly by Müller and Hou (1992), and is included here for completeness.

Under the assumption that rank $D=q$ (the significance of this assumption will be explained later), it is possible to choose a nonsingular matrix $T$ such that

$$T = \begin{bmatrix} N & D \end{bmatrix}, \quad N \in \mathbb{R}^{(m-q) \times q}$$  \hspace{1cm} (2.72)

Using this nonsingular matrix, the system given by Equations (2.70, 2.71) can be transformed as

$$\dot{\bar{x}} = \bar{A}\bar{x} + \bar{B}u + \bar{D}d$$  \hspace{1cm} (2.73)

$$y = \bar{C}\bar{x}$$  \hspace{1cm} (2.74)

where

$$\bar{x} = T\bar{x} = T\begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix}, \quad \bar{A} = T^{-1}AT = \begin{bmatrix} \bar{A}_{11} & \bar{A}_{12} \\ \bar{A}_{21} & \bar{A}_{22} \end{bmatrix},$$

$$\bar{B} = T^{-1}B = \begin{bmatrix} \bar{B}_1 \\ \bar{B}_2 \end{bmatrix}, \quad \bar{D} = T^{-1}D = \begin{bmatrix} 0 \\ I_q \end{bmatrix},$$

$$\bar{C} = CT = [CN \quad CD]$$

with $\bar{x}_1 \in \mathbb{R}^{m-q}$ and $\bar{x}_2 \in \mathbb{R}^{e}$.
In equation (2.73) and (2.74), it is possible to decouple the system into two parts: one directly related to the unknown input, the one which does not see the unknown input. The latter system can now be defined as:

\[
\begin{bmatrix}
I_{n-q} & 0
\end{bmatrix} \hat{x} = \begin{bmatrix}
\overline{A}_n \\
\overline{A}_i
\end{bmatrix} \bar{x} + \overline{B}_i u
\]

\[y = \begin{bmatrix} CN & CD \end{bmatrix} \bar{x}
\]

under the assumption that \( \bar{x}_2 \) can be measured. Also, if the matrix \( CD \) has full column rank, then there exists a nonsingular matrix

\[U = \begin{bmatrix} CD & Q \end{bmatrix} \quad \text{with} \quad Q \in \mathbb{R}^{m(m-q)}
\]

where

\[U^{-1} = \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} \quad \text{with} \quad U_1 \in \mathbb{R}^{m \times m}, \quad U_2 \in \mathbb{R}^{m \times q}
\]

By pre-multiplying both sides of measurement equation (2.74) by \( U^{-1} \),

\[U_1 y = U_1 CN \bar{x}_1 + \bar{x}_2
\]

\[U_2 y = U_2 CN \bar{x}_1
\]

and by substituting equation (2.80) into equation (2.76) and combining it with equation (2.81), it is possible to obtain the system.

\[\hat{x}_i = \tilde{A}_i \bar{x}_i + \overline{B}_i u + E_i y
\]

\[\bar{y} = \tilde{C}_i \bar{x}_i
\]

where

\[\tilde{A}_i = \overline{A}_n - \overline{A}_i U_1 CN, \quad E_i = \overline{A}_n U_1, \quad \tilde{C}_i = U_2 CN, \quad \bar{y} = U_2 y
\]

If the pair \( \{\tilde{A}_i, \tilde{C}_i\} \) is observable or detectable, the conventional observer design procedure can be applied:

\[\dot{w} = (\tilde{A}_i - L \tilde{C}_i) w + \overline{B}_i u + L^* y, \quad w \in \mathbb{R}^{n-q}
\]

where \( L \in \mathbb{R}^{(n-q)(m-q)} \) and \( L^* = LU_2 + E_i \). With appropriate choice of \( L \), \( w \to \hat{x}_i \) as \( t \to \infty \). We have
\[
\hat{x} = T \hat{x} = T^w \left[ \begin{array}{c} w \\ U_1 y - U_1 CNw \end{array} \right]
\]

(2.86)

where \( \hat{x} \rightarrow x \) as \( t \rightarrow \infty \). Furthermore, using the result obtained above, it is possible to estimate the unknown input by

\[
\hat{d} = U_1 \hat{y} + G_1 w + G_2 y + G_3 u
\]

(2.87)

where

\[
G_1 = U_1 CNLU_2 CN + U_1 CN\bar{A}_{12}U_1 CN - U_1 CN\bar{A}_{11} - \bar{A}_{21} + \bar{A}_{22}U_1 CN
\]

(2.88)

\[
G_2 = -U_1 CNLU_2 - U_1 CN\bar{A}_{12}U_1 - \bar{A}_{22}U_1
\]

\[
G_3 = -U_1 CN\bar{B}_{1} - \bar{B}_2
\]

The existence condition of UIO for the system described by equation (2.85) can be summarized as

(a) \( \text{rank } CD = \text{rank } D \)

(2.89)

(b) \( \text{rank } \left[ \begin{array}{cc} sI_{n-k} - \bar{A}_{n} & -\bar{A}_{11} \\ CN & CD \end{array} \right] = n \)

(2.90)

The proof of these conditions can be found in Müller and Hou (1992). The physical interpretation of the conditions is presented in the following paragraphs. The first condition (2.89) states the required number of measurements and the quantities that need to be measured. To satisfy this condition, the number of measurements in the output vector \( y \) must be equal to or greater than the number of unknown input vector \( d \). Also the state that is directly coupled to the unknown inputs must be obtainable from the measurement vector. The second condition (2.90) states that the transmission zeros of the triple \((C, A, D)\) must be stable. This condition stems from the fact that the UIO performs an implicit inversion of the system and the transmission zeros of the system become the poles of the UIO. Therefore, to obtain a stable UIO, the transmission zeros of the system should be stable.
2.3.2 Diagnostic Observer for Nonlinear Systems

Diagnostics for nonlinear systems can be designed using the robust linear FDI scheme addressed so far, either by treating nonlinearities as uncertainties in the system model, or as disturbances. However, not only is the extension of the linear observer design to nonlinear system difficult, but also less reliable in the presence of severe nonlinearities. Therefore, the use of the nonlinear observer seems to be more attractive even though observer theory for nonlinear systems is still subject of research. Several approaches have been proposed and they are categorized as follows:

(1) Nonlinear Identity Observer

(2) Tau Type Observer

(3) Nonlinear Fault Detection Filter

(4) Nonlinear Unknown Input Observer

(5) Adaptive Nonlinear Observer

Some of the methods above utilize extension of well known diagnostic methods for linear systems and some of them apply a nonlinear observer design philosophy.

**Thau Type Nonlinear Observer**

For a nonlinear system of the following form,

\[
\dot{x} = Ax + Bu + f(x)
\]  
(2.91)

\[y = Cx.
\]  
(2.92)

Himmelspach (1992) showed the Thau type nonlinear observer can be used to detect and isolate sensor faults in a system. The Thau type observer is constructed as follows.

\[
\dot{\hat{x}} = A \hat{x} + Bu + f(\hat{x}) + L(y - \hat{y})
\]  
(2.93)

\[
\hat{y} = C \hat{x}
\]  
(2.94)

where \(L\) is a constant observer gain matrix.
The observer existence conditions are:

1. $(C,A)$ is observable

2. $f(x)$ is continuously differentiable on a domain $L$ of the state space.

3. $f(x)$ is locally Lipschitz on $L$, i.e.,

$$\|f(x_1) - f(x_2)\|_2 = L_{2f} \|x_1 - x_2\|_2, \quad \forall x \in L$$ \hspace{1cm} (2.95)

with Lipschitz constant $L_{2f}$

Solution Procedure – design of $L$

The following conditions must be satisfied to construct an asymptotic observer.

$$\text{Real}\{\sigma(A-LC)\} < 0$$ \hspace{1cm} (2.96)

$$L_{2f} < \frac{\lambda(Q)}{2\lambda - (P)}$$ \hspace{1cm} (2.97)

where $\sigma(.)$ denotes eigenvalues, $\lambda$ and $\lambda'$ denote the minimum and maximum eigenvalue respectively. $P$ is the positive definite symmetric solution to the Lyapunov equation

$$(A-DC)\text{T}P + P(A-DC) = -Q.$$ \hspace{1cm} (2.98)

where $Q$ is an arbitrary positive definite and symmetric matrix.

The Thau type observer combined with a binary phase detection filter was applied to sensor fault isolation by Himmelspach (1991). He showed that the Thau type nonlinear observer works for small bias type faults meanwhile it has poor performance for larger faults. This is mainly due to the fact that the Thau type observer is based on the assumption that $f(x)$ is locally Lipschitz on $L$.

Nonlinear Identity Observer

This approach was first developed by Hengy and Frank (1986) with an aim to detect and isolate component faults. Frank (1987) extended this approach to a more general class of faults. Recently, Adjallah et al. (1994) proposed a new design procedure to provide better stability of the observer.
For a nonlinear system without modeling uncertainty,

\[ \dot{x} = f(x, u) \quad (2.99) \]
\[ y = h(x, u) \quad (2.100) \]

We design an observer with the following structure.

\[ \hat{x} = f(\hat{x}, u) + K(\hat{x}, u)(y - \hat{y}) \quad (2.101) \]
\[ \hat{y} = h(\hat{x}, u) \quad (2.102) \]

Now, use a Taylor expansion for \( f(x,u) \) and \( h(x,u) \) around \( x = \hat{x} \) to obtain

\[ \dot{x} = f(\hat{x}, u) + \left. \frac{\partial f(x, u)}{\partial x} \right|_{x=\hat{x}} (x - \hat{x}) + h.o.t \quad (2.103) \]
\[ y = h(\hat{x}, u) + \left. \frac{\partial h(x, u)}{\partial x} \right|_{x=\hat{x}} (x - \hat{x}) + h.o.t \quad (2.104) \]

where \( h.o.t \) denotes the remaining high order terms. Now, define the estimation error \( e = x - \hat{x} \). The error dynamics become:

\[ \dot{e} = F(\hat{x}, u)e - K(\hat{x}, u)H(\hat{x}, u)e + O_i(e^i, t) \quad (2.105) \]

where

\[ F(\hat{x}, u) = \left. \frac{\partial f(x, u)}{\partial x} \right|_{x=\hat{x}}, H(\hat{x}, u) = \left. \frac{\partial h(x, u)}{\partial x} \right|_{x=\hat{x}} \quad (2.106) \]

and \( O_i(e^i, t) \) is the high order term with respect to error \( e \). The residual generator can be obtained as

\[ r = y - \hat{y} = H(\hat{x}, u)e + O_j(e^j, t) \quad (2.107) \]

where \( O_j(e^j, t) \) is the high order term from the output equation. We now need to find \( K(\hat{x}, u) \) such that it stabilizes the error dynamics.
As a solution for this problem, Adjallah et al. (1994) proposed the following procedure with the assumption that $h(x,u) = Cx$, $\ker C \neq 0$. Note that the assumption made here is not too restrictive since in many applications the output relationship is linear.

**Solution Procedure** – design of $K(\hat{x}, u)$

The matrix $K(\hat{x}, u)$ takes the form

$$K(\hat{x}, u) = P^{-1} \hat{F}(\hat{x}, u) C^T Q$$

(2.108)

where

$P = P^T > 0$ is positive definite matrix

$$K = -\frac{\partial f(x,u)}{\partial x} \bigg|_{x=\hat{x}}$$

$\bar{K}$ is a negative definite matrix

$\bar{K}$ is the highest rank right orthogonal matrix to $C$

$$\hat{F}(\hat{x}, u) = diag \left\{ \frac{1}{2} \sum_{j=x}^{n} \psi_{ij} \psi_{ij} \right\} \quad i = 1, \cdots, n$$

(2.109)

$\psi_{ij}$ is the $ij$th element of the matrix $P \frac{\partial f(x,u)}{\partial x} \bigg|_{x=\hat{x}}$

$Q$ is a matrix satisfying $C^T QC - I \geq 0$.

With this selection of matrix $K(\hat{x}, u)$, the error dynamics is asymptotically stable (Adjallah et al., 1994).

Note that terms of order higher than one can be taken into consideration in the observer design (Hengy and Frank 1987) at the expense of greater computation. Even with the first order approximation, computational burden poses a great challenge for on-line application. For some nonlinear systems in a special form, a constant gain matrix $K$ can be used instead of $K(\hat{x}, u)$ as given above. This is especially true for a system that can be represented as $\dot{x} = Ax + g(x)u$ and the function $g(x)$ satisfies the Lipschitz condition. A constant gain matrix $K$ can then be selected to stabilize the observer. In this case, the design procedure for $K$ is similar to the Thau type observer (Thau, 1973). Also, Gauthier et al. (1992) showed that the observer with constant gain matrix $K$ can be used when a system has the form, $\dot{x} = f(x) + g(x)u$.
Nonlinear Fault Detection Filter

As an extension of linear fault detection methods to nonlinear systems, Garg and Hedrick (1995) combined the Thau type observer and the fault detection filter method (White and Speyer, 1987) for fault detection in a nonlinear system. For a nonlinear system represented as

\[ \dot{x} = Ax + f(x) + Bu + \sum_{i=1}^{s} F_i \phi_i(x, u) f_i \]  
\[ y = Cx \]

(2.110)  
(2.111)

where $F_i \in \mathbb{R}^s$, $s$ is the number of parameter($f_i$) changes and $\phi_i$ is a function of $x$ and/or $u$ depending on the parameter under consideration. Under the assumptions given below

1. the nonlinear term $f(x)$ is Lipschitz, i.e. $\|f(x_i) - f(x_j)\| \leq \gamma \|x_i - x_j\|$;
2. $(A, C)$ is observable;
3. $CF = \begin{bmatrix} CF_1 & CF_2 & \cdots & CF_s \end{bmatrix}$ is of rank $s$, i.e. the failure modes are linearly independent in the output space;

the following structure of Thau type observer can be designed:

\[ \dot{\hat{x}} = A\hat{x} + f(\hat{x}) + Bu + H(y - C\hat{x}) \]

(2.112)

where the matrix $H$ is the observer gain matrix that is selected to stabilize the observer.

Solution Procedure – design of $H$

As for the detection filter in linear system, the observer gain $H$ is chosen such that the outputs of observer produce the directional residuals. To this end, $H$ is selected as follows.

\[ H = Q_x [CF] + H_x [I - CF][CF]^T C \]

\[ Q_x = \begin{bmatrix} AF_i - \sum_{j=1}^{s} \alpha_i \lambda_i \nu_j & \cdots & AF_i - \sum_{j=1}^{s} \alpha_i \lambda_i \nu_j \end{bmatrix} \]

\[ F_i = \sum_{j=1}^{s} \alpha_i \nu_j \]

(2.113)

\[ H_x = \frac{1}{2e} PC^T (I - CF)[CF]^T C \]
where \( \lambda_i, v_i \) are the eigenvalues and eigenvectors of the matrix \( A_h = A - HC \). \( \epsilon > 0 \) is a positive constant such that there exists a symmetric and positive definite matrix \( P \) such that the modified Riccati equation

\[
0 = A_d P + P A_d^T + P[I y^2 - \frac{1}{\epsilon} C^T C] P + I + I_e
\]  

(2.114)

is satisfied, \( \alpha_i \) are constants and

\[
F_e = [I - CF][CF]^T C
\]  

(2.115)

and the superscript * means the pseudo-inverse of a matrix.

**Nonlinear Unknown Input Observer**

One attractive property (disturbance decoupling) of the unknown input observer for linear systems propelled its generalization into nonlinear systems. The extension of the linear UIO to nonlinear systems are found in many publications such as Krener and Respondek (1985), Zeitz (1987), Xia and Gao (1989), Wün menberg (1990) and Seliger and Frank (1991). Two different design procedures are introduced here, a nonlinear UIO proposed by Wün menberg (1990) and disturbance decoupling nonlinear observer by Seliger and Frank (1991).

Wün menberg (1990) considered a nonlinear system that is in observable canonical form (Bestle and Zeitz, 1983) as follows.

\[
\dot{x} = Ax + B(y, u) + Ed + K(x, u)f
\]  

(2.116)

\[
y = Cx + K_e(x, u) f
\]  

(2.117)

where \( d = d(x, u, \theta) \) represents the unknown input and \( f \) denotes either actuator or component fault to be detected. The nonlinear unknown input observer for the system given above is

\[
\dot{x} = F\dot{x} + J(y, u) + Gy
\]  

(2.118)
Solution Procedure – selection of $F, J$ and $G$

The following conditions for the observer matrices are necessary in order to decouple the unknown input $d$ and to provide sensitivity to the fault vector $f$.

\[ TA - FT = GC, \quad F \text{ stable} \]

\[ J(y,u) = TB(y,u) \]

\[ L_1 T + L_2 C = 0, \quad TE = 0 \quad (2.119) \]

\[ \text{rank}(TK(x,u)) = \text{rank}(K(x,u)) \]

\[ \text{rank}\left( \begin{bmatrix} G \\ L_2 \end{bmatrix} K_1(x,u) \right) = \text{rank}(K_1(x,u)) \]

With the above conditions satisfied, the estimation error ($e = Tx - \hat{x}$) dynamics become,

\[ \dot{e} = Fe - GK(x,u)f + TK_1(x,u)f, \quad (2.120) \]

Note that the estimation error dynamics is not affected by the unknown input $d$. By defining the residual as

\[ r = L_1 \hat{x} + L_2 y \quad (2.121) \]

and substituting $\hat{x} = Tx - e$ into the above equation, we have

\[ r = L_1 e + L_2 K_1(x,u)f, \quad (2.122) \]

where the residual does not see the unknown input $d$ and is only sensitive to the fault $f$. The drawback of this approach is the difficulty of transforming the system into the system represented by the form given in equations (2.106) and (2.107), and the fact that the existence conditions are restrictive in most application.

Instead of using the linear transformation used in above method, a nonlinear transformation is adopted to design UIO for the more general nonlinear system. Krener and Respondek (1985), Zeit (1987), Xia and Gao (1989) proposed a two step state transformation procedure that allows the construction of an UIO for the system (2.106) and (2.107). However, the two step transformation procedure is not trivial to obtain and its existence conditions are restrictive. Seliger and Frank (1991) propose an alternative
technique that requires a single transformation that directly decouples the disturbance or unknown inputs.

Consider the nonlinear dynamic system represented by the following state space equations.

\[ \dot{x} = f(x) + g(x)u + p(x)d + q(x)f \]  
\[ y(x) = h(x) \]  

(2.123)  

(2.124)

Apply a transformation \( z = T(x) \) that satisfies the following equations.

\[ \frac{\partial T(x)}{\partial x} f(x) = FT(x) + \Phi_d h(x) \] 

\[ \frac{\partial T(x)}{\partial x} g(x) = \Phi_f h(x) \] 

\[ \frac{\partial T(x)}{\partial x} p(x) = 0 \]  

(2.125)  

\[ \text{rank} \left( \frac{\partial T(x)}{\partial x} q(x) \right) = \text{rank}(q(x)) \]  

\[ R(T(x), h(x)) = 0 \]

If a stable matrix \( F \) and output transformations \( \Phi_d \) and \( \Phi_f \) can be found such that this set of linear partial differential equations can be solved simultaneously for \( T(x) \), a robust fault detection observer can be designed for the system as follows.

\[ \dot{\hat{z}} = F \hat{z} + \Phi_d h(x) + \Phi_f h(x)u \]  
\[ r = R(\hat{z}, y) \]  

(2.126)  

(2.127)

Then the residual error evolves according to

\[ \dot{e} = Fe - \frac{\partial T(x)}{\partial x} q(x)f \]  
\[ r = R(T(x) + e, h(x)) \]  

(2.128)  

(2.129)

It can be seen that the residual is unaffected by the unknown inputs \( d \) and is affected by the fault \( f \). The authors presented a systematic method for solving the set of simultaneous partial differential equations (2.125).
Adaptive Nonlinear Observer

When the plant parameter changes or parameter uncertainty is presented in the system, observer based methods for fault detection are not as effective as a system that is completely known. To enhance fault detection capabilities, the adaptive observer based residual generator approaches have been explored by Billman and Isermann (1987), Bastin and Gevers (1988). This approach was further explored for more general case by Ding and Frank (1992, 1993) and recently by Yandg and Saif (1995). We now discuss the approach used by Ding and Frank (1992).

Consider a nonlinear system

$$\dot{x} = a(x) + q(x,u) + Q(x,u)\theta + G(x,u)f + g(t) \quad (2.130)$$

$$y = c(x), \quad x_0 = x(0) \quad (2.131)$$

where the output is considered scalar for the sake of simplicity and $a: R^n \to R^n$, $Q: R^n \times R^n \to R^{n \times d}$, $G: R^n \times R^n \to R^{m}$, $g: R \to R^n$, and $c: R^n \to R$ are assumed to be known and smooth enough, $f$ represents an abrupt fault and $\theta \in R^d$ is unknown and represents a time varying parameters. Also assume that the time rate of change in the parameter $\theta$ is bounded such that $0 < \|\dot{\theta}\| \leq M << \infty$.

To design an adaptive observer, a transformation $\xi = T(x) \in R^k$, $k \leq n$ has to be found, defined on a neighborhood of the initial state $x_0$ such that

$$\dot{\xi} = F \xi + \Psi (y,u) + \Psi (y,u)\theta + \Xi (x,u) f \quad (2.132)$$

$$y = \begin{bmatrix} 0 & \ldots & 0 & 1 \end{bmatrix} \xi \quad (2.133)$$

where rank$[\Xi (y,n)] = l$. The existence condition of this transformation is found in Ding and Frank (1992).

Now, an adaptive nonlinear observer has the following structure

$$\dot{\xi} = F \xi + \Psi (y,u) + \Psi (y,u)\hat{\theta} + L_\gamma \begin{bmatrix} V(t) \\ 0 \end{bmatrix} \dot{\theta} \quad (2.134)$$

where
\[
\dot{\theta} = \Gamma \phi^T(t) r(t)
\]
\[
\dot{V}(t) = RV(t) + \Psi(y, u), \quad V(0) = 0
\]
\[
\phi(t) = k^T V(t) + \psi(y, u)
\]

where $\Gamma$ is a positive definite matrix and $R$ is a stable matrix.

Solution Procedure

The elements $l_j$ of $L$ are assigned such that $s^s + l_s s^{s-1} + \cdots + l_1 s + l_0 = 0$ is Hurwitz, $k^T = [0 \cdots 0 1]$.

Let a matrix $Q$ satisfy

\[
QFQ^{-1} = \begin{bmatrix} R & \times \\ k^T & \times \end{bmatrix}
\]

then

\[
Q\Psi(y, u) = \begin{bmatrix} \Psi_n(y, u) \\ \Psi_a(y, u) \end{bmatrix} \in R^{n+1}
\]

and the conditions

1. $\phi(t)$ and $\dot{\phi}(t)$ are bounded
2. $\exists \alpha, \beta$ such that $0 < \alpha t \leq \int_0^t \phi(\tau) \phi'(\tau) d\tau$
3. $\exists M_i$ such that $|\dot{V}(t)\dot{\theta}| \leq M_i < \infty$

are required to guarantee $|y - \xi| < K < \infty, \forall t$.  

\[\Box\]
2.4 Fault Tolerant Control

2.4.1 Fault Tolerant Control in General

The design of reliable control systems (fault tolerant control) can be categorized by the approach used: a passive approach and an active approach. By passive approach, we mean that the system has limited fault tolerance achieved, through appropriate selection of the controller structure and feedback gain. Passive approaches generally adopt the robust control methodology to ensure that the system remains insensitive to certain faults and that the controller and system keep the same structure even in the presence of the fault. Therefore, the effectiveness of the passive approach strongly depends on the robustness of the nominal system (fault free system). The reason for the limited fault tolerance stems from the fact that the only design freedom left in passive approach is the controller parameter, without use of the on-line fault information. Typical methods for robust controller design are listed below:

- $\mu$ analysis and synthesis - Doyle (1984), Packard and Doyle (1993)
- Variable structure (Sliding mode) control - Utkin (1992)

In contrast to the passive approaches, active approaches to fault tolerant control are based on the reconfiguration of a new control module to achieve desirable system responses in the presence of certain faults. In order to achieve this goal, active approaches require either $a$ priori information on the expected fault types, or a mechanism for detecting and isolating unanticipated faults. In the latter case, the mechanism for detecting and isolating the faults can be the FDI scheme introduced earlier in this chapter. The active approaches are divided into two types: on-line automatic controller reconfiguration and projection based method. The projection-based methods select a new pre-computed (off-line) and stored control law to respond to the type of fault occurred. The on-line automatic controller reconfiguration methods are often referred to as reconfiguration method. This approach calculates the new controller
parameters in response to a certain fault in the system, and permits changes in the structure of the controller to accommodate the fault. Various techniques have been proposed for reconfiguration control and this is still an area of active research. We list:

Modeling following method - Morse (1990), Huang and Stengel (1990)

The diagram depicted in Figure 2.6 illustrates the area of the fault tolerant control system design. The passive approaches are attacking the combined area of FDI and Robust control denoted by area A and the active fault tolerant control approaches investigate into the area B.

![Figure 2.6: Area of fault tolerant control system](image)

From Figure 2.6, it is obvious that there are interactions between FDI and controller reconfiguration (in the active approach) and between FDI and robust control (in the passive approaches). In active approaches, FDI module is responsible for locating the fault through residual generation and decision making. Control system can then be reconfigured either by occupying a new pre-designed controller or by changing the controller parameters. Therefore, the diagnostic information provided by the FDI module is crucial to the effectiveness of the active fault tolerant control approaches. On the other hand, the interaction between FDI
and robust control can be characterized by trade-off between them. For robust control, the controller is designed such that the system response is insensitive to certain noise or disturbances that may also be a type of fault. In contrast, sensitivity to fault needs to be maximized in the FDI module. Therefore, there exists an inherent trade-off between FDI and robust control. For effectiveness of the fault tolerant control design, this interaction and trade-off represented by area A in Figure 2.6 should be optimized. The next section illustrates one approach to fault tolerant control design for linear systems.

2.4.2 Integrated Design of Diagnostics and Control (4 parameter controller)

A typical schematic diagram of the model based FDI scheme is illustrated in Figure 2.7. The task of FDI is in general to check the consistency between the input and output using the input and output relationship.

![Figure 2.7: Closed loop control system with open loop FDI](image)

As can be seen in Figure 2.7, the relation between the input $u(t)$ and output $y(t)$ can be described by the open loop system model. In other words, most of the model based FDI schemes are designed in such a way that they have no effect on control function. Even though the open loop FDI in Figure 2.7 does not have effect on the controller, the control signal $u(t)$ influences the FDI residuals whenever there is modeling uncertainty in the system. To see the effect of the control signal on the open loop FDI residual, consider a system with modeling uncertainty below.
\[ \dot{x} = (A + \Delta A)x + (B + \Delta B)u + Ff \]
\[ y = (C + \Delta C)x + (D + \Delta D)u + Gf \]

where \( x \) is the state vector, \( u \) is the input, \( y \) is measurement and \( f \) is a fault and \( A, B, C, D, F \) and \( G \) are the matrices of appropriate dimension. Then the observer can be designed as:

\[ \dot{\hat{x}} = (A - LC)\hat{x} + Bu + Ly \]
\[ \hat{y} = C\hat{x} + Du \]

(2.139)

Now, define the estimation error \( e \) as \( x - \hat{x} \). Then the error dynamics become:

\[ \dot{e} = x - \hat{x} \]
\[ = (A + \Delta A)x + (B + \Delta B)u + Ff - (A - LC)\hat{x} + Bu + Ly \]
\[ = (A - LC)e + (F - LG)f + (\Delta B - L\Delta D)u + (\Delta A - L\Delta C)x \]

(2.140)

Define the residual \( r = We \) where \( W \) is a weighting matrix. In transfer function form,

\[ r = We \]
\[ = W \left[ G + C(sI - A + LC)^{-1}(F - LG) \right] f(s) \]
\[ + WC(sI - A + LC)^{-1}(\Delta B - L\Delta D)u(s) \]
\[ + WC(sI - A + LC)^{-1}(\Delta A - L\Delta C)x(s) \]

(2.141)

From equation (2.141), it can be seen that the control \( u(t) \) is related to the residual \( r(t) \) through \( (\Delta B - L\Delta D) \).

The residuals thus become less sensitive to the fault due to the modeling uncertainty. Therefore, a robust controller designed to enhance robustness to modeling uncertainty could possibly reduce sensitivity of the residuals to faults. Hence the effectiveness of FDI scheme would be seriously compromised. To overcome this difficulty, in addition to the standard robust control design criteria, additional design specifications can be considered so that the FDI scheme becomes sensitive to faults and at the same time insensitive to control signals and modeling uncertainty.

A solution for this problem is to design the robust control and FDI scheme simultaneously by optimizing both control and FDI objectives. The first approach on this idea was proposed by Nett et al.
(1988) and called 4 degree-of-freedom controller. The structure of the 4 degree of freedom controller is depicted in Figure 2.8.

![Diagram of 4 degree-of-freedom controller](image)

Figure 2.8: Four-parameter controller approach

In Figure 2.8, $G$ is the transfer function matrix, $u$ and $z$ denote system input and output, $w$ is the exogenous input, $y$ is the measured output, $z^*$ is the diagnostic signal and $w^*$ is the reference input. The control parameter is denoted by $K_g$ and $n_a=f_a+u_n$, $n_s=f_s+w_s$ represent actuator fault + input noise and sensor fault + sensor noise, respectively. Note that with $K_{11}=K_{12}=0$ ($z^*=0$), the controller becomes a conventional 2 parameter control structure. The control is determined by the following relationship.

$$ u^*(s) = [K_{22} \quad K_{12}] \begin{bmatrix} y^*(s) \\ w^*(s) \end{bmatrix} $$  \hspace{1cm} (2.142)

The diagnostic signal is obtained through:

$$ z^*(s) = [K_{11} \quad K_{12}] \begin{bmatrix} w^*(s) \\ y'(s) \end{bmatrix} $$  \hspace{1cm} (2.143)

The controller is parameterized to further explore its relationship with the plant $G(s)$ as shown in Figure 2.9.

![Diagram of parameterized controller](image)

Figure 2.9: Parameterized controller
The four-parameter controller has an implicit internal model of the plant as seen in Figure 2.9. And $x=y^*-Gu^*$ where $y^*$ is the measured plant output and $Gu^*$ is the implicit model output. Without model uncertainty ($x=0$), there is neither feed-forward nor feedback from $x$. However, if $x\neq0$, then feed-forward (from $x$) can be used to deduce the effect of uncertainty and feedback (from $x$) can be utilized to reduce the effect of the uncertainty. With this physical insight, the control parameter can be selected to obtain the following characteristics: i) the plant output tracks the reference command and is insensitive to actuator faults; ii) the diagnostic output tracks actuator faults; iii) both the above properties are maintained with bounded uncertainty. Several approaches for the design of the control parameter $K_y$ have been proposed. Nett et al. (1988), Jacobson and Nett (1991) and Kilsgaard et al. (1996) applied the $H^\infty$ optimization technique to achieve the characteristics given above, Tyleer and Morari (1994) adopted the $H^2$ optimal control technique and Juarez and Ajbar (1996) introduced the $l^\infty$ optimal control method.

In this four parameter approach, the closed loop signals (reference input and output) are utilized for fault detection, which seems reasonable at first glance. However, the interaction between the robustness effect of the controller and fault estimation limit the overall performance of the system in the presence of simultaneous modeling error and fault. With this integrated design approach, it is possible to achieve a limited range of fault tolerance design. For some significant faults, a different approach such as reconfiguration, restructuring or rescheduling of the controller is required. Even though the integrated design approach is a passive way to achieve a fault tolerant system, it still provides a reasonable baseline robust controller. To accommodate various types of faults, a bank of such base line robust controllers with model based FDI is recommended.
2.5 Application of modern control and estimation theory to automotive engine AFR control and diagnosis

2.5.1 Control Application

The task of the engine controller in the context of AFR regulation is to maintain the air fuel ratio at a stoichiometric value over a wide range of engine operating conditions. Current production engine control schemes primarily employ both feed-forward and feedback control to provide very precise AFR control in both steady state and under slowly varying engine operating conditions. Feed-forward control, which is relatively faster than feedback control, is utilized to accommodate any transients introduced. Meanwhile, steady state error or disturbances are primarily handled by the feedback from the HEGO sensor.

The feed-forward terms are determined by the amount of air entering into a cylinder per engine cycle. Once the amount of air entering a cylinder is known, the amount of fuel to be injected (feed-forward term) can be calculated simply as:

\[
m_{\text{feedforward}} = \frac{\dot{m}_a}{\text{Stoichiometric}}
\]  

(2.144)

The most challenging task in this process arises from the fact that the amount of air entering each cylinder is difficult to measure directly. Two different approaches have been employed to estimate this, either using a mass flow meter or manifold pressure sensor (the so called speed density method). However, since both of these are primarily based on steady state calibration and empirical compensation, only relatively large AFR excursion (more than 15%) can be observed under transient conditions (Cho, 1991; Cho and Oh, 1993). This is part of the reason that production lean-burn engines operate lean only at steady states and employ the conventional AFR control (air fuel ratio at stoichiometry) during transient operation.

In addition to the uncertainty involved in the process of estimating air flow rate, fuel film dynamics (also called wall wetting phenomenon) introduce another obstacle during transient operations. During transient operation, not all the fuel injected enters a cylinder. Some of the fuel condenses on the intake wall, evaporates later and is inducted into the cylinder over the next few cycles. The fuel film dynamics are also
difficult to measure, hence a feed-forward compensation based on a lookup table is adopted in production vehicles. Furthermore, typical problems encountered in control applications, such as noise, disturbances and other uncertainties can also be seen in the process of the engine AFR control implementation.

The difficulties mentioned above form a basis for applying modern control and estimation theories to the engine control problem. The feasibility of using estimation theory for AFR control was postulated by Athans (1978). Athans categorized the engine control problem into 3 different groups; i) a static and dynamic optimization; ii) a stochastic multi-variable control system design; iii) a system design with reliability. Although he did not explore nor propose detailed control strategies, he described and delineated the engine control problem in the context of model control theory. Since then, a wide variety of works has been reported which are explained in the later parts of this section. Most of these are focused on improving the AFR regulation during transient conditions by replacing the conventional, empirical, feed-forward and feedback control with a physical, model-based approach. Although these methods vary in their approach, the fundamental concept behind the theory is to estimate both air dynamics and fuel dynamics and use these estimates to improve AFR regulation using various control theories. Typically, an estimator based on a mathematical model is utilized. The model should contain the airflow dynamics in the intake manifold and the wall wetting dynamics for fuel. Some models may also include the engine process delays and transient response of feedback sensors.

In reviewing previous works in this area, the focus was on three specific areas: the objective of the estimator (estimation of air dynamics or estimation of fuel dynamics or both); the differences in the model (e.g., time-based versus event-based); the differences in the control strategies used. Model based AFR control research was contributed by several research groups such as Powell (Stanford University), Hedrick (U.C. at Berkley), Grizzle and Cook (University of Michigan and Ford Motor Co., respectively), Geering (Swiss Federal Institute of Technology) and Hendricks (Technical University of Denmark). Without ignoring the contributions of other research groups, this review focuses primarily on reported work from these five groups, while significant contributions from other groups are introduced as and when necessary.
An event based AFR control approach has been an area of major research at Stanford University (Chang, 1993; Chang et al., 1993; Amstutz et al., 1994; Chang et al., 1995). Their engine models include intake manifold air dynamics, fuel wall wetting dynamics, and delays inherent in 4 cycle engine and exhaust oxygen sensor dynamics. The sampling period is synchronized with crank angle (so called event-based model) as opposed to the conventional sampling in constant time interval (so called time-based). In their early work (Chang et al., 1993), the intake air dynamics is considered instantaneous so that the intake air flow is assumed to be a static function of throttle opening and engine speed. For estimator and controller design, a Kalman filter/observer is adopted combined with an integral control to compensate the bias introduced by unknown disturbance or uncertainties involved in the modeling process. For precise control of airflow, an electronically controlled DBW (Drive By Wire) throttle is used. This approach is then experimentally demonstrated on a single cylinder CFR engine. Amstutz et al. (1994) extended this approach by using the switch type EGO sensor, used in production engine, rather than using the UEGO sensor which provides linear information about the exhaust air fuel ratio. More complicated air dynamics in the intake manifold were introduced by Ault (1994) and Fekete (1994, 1995). They used a look-up Table to estimate the steady state air mass as a function of throttle position and engine speed. Then, the steady state air mass flow is used as an input to the intake manifold filling and emptying model which is modeled as a first order lag. The time constant for this first order lag model is also mapped as a function of air mass flow and engine speed. For the identification of intake manifold time constant, Ault utilizes measurements from the UEGO sensor during steady fuel injection by dithering the throttle opening. In contrast, Fekete employs intake manifold pressure measurements, during small throttle steps, to determine the time constant. Effectiveness of these off-line calibration techniques is then demonstrated for air fuel ratio control on a multi-cylinder engine. Both techniques require extensive quantities of data, since finer quantization of the time constant map provides increased accuracy of the estimate of the air mass in cylinder. To apply their method to multi-cylinder engines, they also introduced the exhaust mixing model as a first order lag. Fekete (1995) extended this approach further and demonstrated its possible application to multi-cylinder lean-burn engines.
Estimator based AFR control strategies using continuous time domain model have been developed by the research group at Technical University of Denmark (Hendricks and Sorensen, 1991, Hendricks et al., 1992, Kaidantzis et al., 1993, Hendricks et al., 1996). Their model consists of intake manifold air dynamics and fuel film dynamics. For the filling and emptying model of airflow into a cylinder, they adopted the speed density approach through the use of volumetric efficiency look-up table as a function of manifold pressure and engine speed. Their approaches mainly focused on the estimation of airflow into the cylinder and the estimation of fuel dynamics for transient fuel compensation. Hendricks et al. (1992) proposed the use of Constant Gain Extended Kalman Filtering to estimate the airflow and the fuel flow from fuel film. This method was compared to the application of a sliding mode observer to the same problem in (Kaidantzis et al., 1993). The transient fuel compensation scheme used in their approach is reported in Hendricks et al. (1993). Kaidantzis et al. (1993) also shows a comparison between sliding mode control and a conventional PI feedback controller.

Due to its robustness and simplicity, the sliding mode control methodology has been attracting the control engineer's attention, especially for nonlinear systems applications. Cho and Hedrick (1991) first introduced the sliding mode control to the engine AFR control problem. Three state variables are used to describe the dynamics involved in AFR control: engine speed, mass of air in the intake manifold and mass of fuel for fuel dynamics. They used the sliding mode methodology as a robust integral feedback scheme, which can replace the PI feedback loop presently used in production. The estimation of air mass is based on the speed density method and no estimation is done for the fuel dynamics. Thanks to the property of robustness of the sliding mode control method, this technique allows less modeling effort, requires very little calibration and relies only on the HEGO sensor. However, this method results in large magnitudes of chattering (Cho and Oh, 1993) and, like conventional controllers, is limited by the accuracy of the required feed-forward controller. Choi and Hedrick (1993) introduced an observer based sliding mode control for AFR control. They showed that faster response and a reduction of chattering can be achieved with introduction of an observer. In addition to the estimator for the fuel, an additional transient fuel compensation algorithm using neural network was proposed by Won et al. (1995)
In Turin and Geering (1993), an on-line identification algorithm based on the extended Kalman filter is presented. They modeled intake air dynamics, fuel wall wetting dynamics and oxygen sensor dynamics. The extended Kalman filter is then applied to determine fuel model parameters using a fuel dither with fixed airflow. The large size of the fuel dither combined with the fact that important delays were assumed to be known make this technique impractical. Turin and Geering (1994) also extended this approached to a model reference adaptive control structure. The performance of the model reference controller is relatively poor and the stability remains to be investigated.

Grizzle and Cook (1988, 1991) tried to implement a individual cylinder AFR control using a single EGO sensor. Their efforts are primarily focused on the modeling of the EGO sensor, delay and the exhaust gas mixing for multi-cylinder engine. To compensate for nonlinearity and delay involved in the process, they applied a linear periodic control. Grizzle et al. (1994) proposed an air charge estimator to improve the transient AFR control, which utilizes a continuous air dynamic model. Recent works include more advanced AFR control application such as control with variable cam timing (Stefanopoulou et al., 1995). For more advanced engine control systems, Cook and Johnson (1995) have summarized the issues of current research and future research directions.

2.5.2 Diagnostics Application

Regulations on the exhaust emission level for passenger vehicles lead to the development of highly efficient engine control systems. As more vehicles with newer technology appear on the road, the overall exhaust emission from vehicles is greatly reduced. Hence emission from defective vehicle becomes significant. This is the reason that On-Board-Diagnosis (OBD) of all the emission related functions and components was deemed necessary from the year 1996. Kiencke (1996) explained the conceptual approach for diagnosis in production vehicles. They include: misfire detection, catalytic converter monitoring, estimation of evaporation from fuel canister, secondary air system monitoring, fuel system monitoring and a check of the exhaust re-circulation system.
Model based diagnostics for the automotive engine has been explored since the late 80's. Rizzoni and Min (1988) applied the fault detection filter to diagnose sensor and actuator faults in electronically controlled engines. They used a linearized engine model and proposed binary phase detection filters to detect and isolate the throttle position sensor and manifold pressure sensor faults. Park and Rizzoni (1989) later used a similar engine model and employed a detection filter design by the eigenvector assignment method for the same sensor failure diagnosis. For robust isolation of faults, Krishnaswami and Rizzoni (1994, 1997) proposed the nonlinear parity equation residual generation (NPERG) scheme and applied it to engine sensor and actuator fault diagnosis. This scheme requires both forward (output observer) and inverse (input observer) models of the system and provides a systematic way of configuring the corresponding models for diagnosis. Krishnaswami (1996) applied the NPERG scheme and nonlinear sliding mode observer for automotive engine diagnosis and experimentally demonstrated its capability. In an effort to improve detection accuracy (less false alarm and missed detection), Soliman (1998) introduced fuzzy logic in addition to NPERG scheme.

As an alternative of using physics based model, Gertler et al. (1995) proposed the use of the nonlinear auto regressive moving average (NARMAX) model and parity equation based approach for the detection of automotive engine faults. Even though they demonstrated the detection capability of the proposed algorithm for various faults in an engine, these algorithms suffer from the lack of adaptability to uncertainties such as vehicle to vehicle variation and parameter change due to aging. Similar approaches have also been proposed by Krishnaswami et al. (1995) where they combined the NARMAX model and NPERG scheme for automotive engine diagnosis.
2.6 Conclusion

This chapter has reviewed model-based FDI methods, fault isolation schemes, fault-tolerant control methods, and their applications to automotive engines. The four-parameter controller method established a method of integrating control and diagnostics. This method mainly focuses on linear systems with input actuator fault. For general cases such as nonlinear systems and system with both actuator and sensor fault, the integration of control and diagnostics is still on-going research area. The following chapter explores this problem to provide a framework for the integration of control and diagnostics problem.
CHAPTER 3

INTEGRATION OF CONTROL AND FAULT DIAGNOSTICS

In this section we consider the integration of control and diagnostics with the objective of achieving fault tolerant control. Fault tolerant control, in this context, refers to a strategy such that the desired stability and performance of the control system are guaranteed in the presence of faults. Fault detection and isolation form an integral part of this strategy thus allowing an early detection of faulty components (actuator and/or sensor). The three issues necessary for achieving this goal are addressed in this chapter: i) problems in fault tolerant control, ii) estimation and identification and iii) integration of control and diagnostics for fault tolerant control. The first part of this chapter discusses the salient features of strategies involving both fault tolerant control and fault diagnostics. The second part of the chapter explores the estimation and identification problem using sliding mode methodology. Some of well-known estimation and identification schemes are illustrated to explore principles of sliding mode estimation methods. In the third part, some new estimation schemes are proposed. These schemes deal with special estimation problems such as estimation binary measurement and input estimation. Both estimation and identification schemes are utilized in control and fault diagnostics. Finally, an approach for fault tolerant control is proposed based on the first two parts.
3.1 Problems Involved in Fault Tolerant Control and Fault Diagnostics

As discussed in Chapter 2, design of a fault tolerant control system encompasses knowledge of fault diagnostics, robust control and controller reconfiguration. Figure 3.1 illustrates the interaction between them (Figure 2.6 is re-introduced for convenience).

![Diagram of fault tolerant control system]

**Figure 3.1**: Area of fault tolerant control system

The discussion of the fault tolerant control in Chapter 2 can be summarized as follows. In the passive approach (Robust Control + FDI, Area A in Figure 3.1), an inherent trade-off between controller robustness (to disturbance - that may also be a fault) and sensitivity of the FDI scheme (to fault or disturbance) limits the effectiveness of fault tolerant control. In the active approach (FDI + Controller reconfiguration, Area B in Figure 3.1), the control module is re-configured based on a decision made by the FDI module. Therefore, diagnostic information from the FDI module is crucial to the performance of active approach. One of the issues to be resolved in this approach is system stability with respect to detection delay. In both approaches, FDI schemes are designed based on the assumption that the system is an open-loop control system. The 4-parameter controller approach (reviewed in Chapter 2) utilizes a closed loop signal in diagnostics design. A limitation in this approach is that it is only applicable to linear systems with input faults. The area D in Figure 3.1 has not been explored in much detail now, possibly due to the complexity of the problem.

Based on observations from the existing literature (review on fault tolerant control in chapter 2), three facts are clear. First, most FDI schemes consider systems as open-loop system and do not use the closed-loop signal in fault diagnostics design. Second, most of the active approach (robust control + FDI)
focuses on systems with input faults. Finally, most of the FDI schemes involve estimations of inputs and outputs that may be used for control. Designing a scheme encompassing all of the above (Area D in Figure 3.1) presents a challenging task. The approach proposed here considers the integration of estimation-based FDI algorithm with estimation-based control in closed-loop system.

3.1.1 Closed-loop Control System and Fault

As mentioned earlier, most fault diagnostic studies focus mainly on the open-loop control systems. However, a closed-loop system may be a reasonable choice for fault tolerant control since disturbance rejection via feedback is after all, the prime motive. A similar approach could be used to compensate faults in a system with feedback control. Therefore, the relationship between closed-loop control systems and faults is examined in this section before we proceed to propose a methodology for a fault tolerant control systems.

Closed-loop Control and Actuator Fault

Consider a feedback control system as follows.

\[
\begin{align*}
\dot{x} &= f(x,t) + B(x,t)u \\
u &= g(y,t), \quad y = h(x)
\end{align*}
\]

(3.1)

where, \( u \) is a feedback control designed such that the output \( y \) tracks a desired trajectory \( y_d \). Assume that the control \( u \) is designed to be robust to disturbances satisfying the so-called matching condition. Let \( u_o \) and \( y_o (=y_d) \) be input and output under normal conditions (without any fault). Then, the system above can be expresses as:

\[
\begin{align*}
\dot{x} &= f(x,t) + B(x,t)u_o \\
u_o &= g(y_o,t), \quad y_o = h(x)
\end{align*}
\]

(3.2)

Now consider the same system with an actuator fault, expressed by the term \( \Delta u \):

\[
\begin{align*}
\dot{x} &= f(x,t) + B(x,t)(u^* + \Delta u) \\
u^* &= g(y^*,t), \quad y^* = h(x)
\end{align*}
\]

(3.3)
Since the matching condition holds for the above system (the matching condition is always satisfied for actuator fault), the control \( u^* \) will force the output \( y^* \) to track \( y_d (=y_o) \). This implies that the control \( u^* \) may be represented as
\[
  u^* = u_o + u_r, \tag{3.4}
\]
where \( u_o \) is the control in absence of the actuator fault and \( u_r \) is an additional control for rejecting the actuator fault. From equation (3.2), (3.3) and (3.4), we can say that
\[
  u_r = -\Delta u \tag{3.5}
\]
Therefore, an actuator fault can be compensated by robust controller and may be detected by monitoring the closed-loop control value, \( u_r \). The difficulty arises in how to decompose the control \( u^* \) into \( u_o \) and \( u_r \). However, once this is done, an inversion of the system is no longer necessary for actuator fault isolation (for example, the use of an input observer with restrictive existence conditions). Another remarkable thing is that the system output in the presence of fault are the same as the output without fault in the system, i.e., \( y^* = y = y_o \). According to Krishnaswami and Rizzoni (1996), strong fault isolation is possible only if the map between the input fault and the output is one-to-one and onto. This is not the case for the above system. This explains why it is more difficult to diagnose a closed-loop system compared to an open-loop system.

**Closed-loop Control and Sensor Fault**

Consider the feedback control system of equation (3.1) again. Let \( u_o \) and \( y_o (=y_d) \) be the input and output under normal conditions (without any fault). Then, the system can be expressed as shown in equation (3.2). Now consider the same system with a sensor fault expressed by the term \( \Delta y \):
\[
\begin{align*}
  \dot{x} &= f(x,t) + B(x,t)x^* \\
  u^* &= g(y^*, t), \quad y^* = h(x) + \Delta y \tag{3.6}
\end{align*}
\]
The control \( u^* \) will force the output \( y^* \) to track \( y_d \). This implies that the true output \( y = h(x) \) does not track the desired trajectory \( y_d \) but tracks \( y_d - \Delta y \), i.e.,
\[
  y^* = y_d \iff h(x) + \Delta y = y_d \iff y = y_d - \Delta y. \tag{3.7}
\]
The controller does not attempt to compensate the sensor fault since it only sees $y^*$ which is different from the desired trajectory $y_d$. Hence, for a sensor fault, an effort should be made to bring the true output $y$ to track the desired trajectory $y_d$. This involves an estimation of the true output $y$ (or sensor fault $Ay$) and feed the estimate to the controller. With this additional step, the controller is reformulated to

$$\hat{u} = g(\hat{y}, t)$$

(3.8)

where $\hat{y}$ is an estimate of the true output $y$ and $\hat{u}$ is a modified control to reject the sensor fault. Two facts need to be noted here. First fact is that the control $u^*$ is a function of $y^*$ (that includes sensor fault $Ay$). By comparing the control $u_o$ (control under normal condition) and $u^*$, it may be possible to detect a fault. However, no distinction can be made between an actuator fault and a sensor fault. Therefore, an additional step is necessary for the isolation of a sensor fault as discussed above. Second fact is that estimation of the output is required to compensate a sensor fault. The estimation of the true output $y$, however, is not an extra task since it is a necessary step in standard fault diagnostic procedures. Although the control structure is not completely altered, it is suitably reconfigured to use an estimate of the output rather than the measured output.

In summary, following remarks can be made here.

i) An actuator fault always satisfies the matching condition and it may be compensated by a feedback control.

ii) A sensor fault induces a deviation of the true output from the desired trajectory hence the need for estimation of the true output for modified control action or controller reconfiguration.

iii) Detection of fault may be possible by monitoring a closed-loop control signal. The difficulty lies in distinguishing the fault induced control action from a normal control action.

iv) For isolation of a fault, output estimation remains to be a necessary step.

In a later section (Section 3.5), an algorithm for the integration of control and fault diagnostics is explored focusing on the remarks listed above. Figure 3.2 depicts the interactions between robust control, fault
diagnostics and controller reconfiguration in the algorithm to be developed. The basic idea of the algorithm can be briefly summarized as follows. An integral sliding mode control method is adopted as a robust control tool. This will provide both fault detection and input fault compensation capabilities. Various estimation methods will be used for fault diagnosis. The main objectives of the estimation procedure are fault isolation and identification. While fault isolation aims at distinguishing fault from one another, fault identification aims to estimate the magnitude of each fault. The controller is reconfigured to compensate an output fault using the information obtained from the estimation process.

![Fault Tolerant Control Diagram](image)

**Figure 3.2: Fault tolerant control**

### 3.1.2. Estimation and Identification Problems in Fault Tolerant Control and Diagnostics

This study focuses on fault tolerant control through an integration of control and fault diagnosis. The design procedure utilizes mathematical models based on physical principles. Fault diagnostics for a model-based approach requires some analytical redundancies. Analytical redundancy requires estimation and/or identification as discussed in Chapter 2. These redundancies are not only useful for diagnostics but
are also helpful for control design/applications. Typical estimation/identification methods employed in this process are listed in Table 3.1.

<table>
<thead>
<tr>
<th>State Estimation</th>
<th>Linear Sliding Mode Observer (Section 3.2, Utkin)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Nonlinear Sliding Mode Observer (Section 3.2, Drakunov, Krishnaswami)</td>
</tr>
<tr>
<td></td>
<td>Observer using Binary Measurement (Section 3.3, Kim, Utkin, Drakunov)</td>
</tr>
<tr>
<td>Input Estimation</td>
<td>Sliding Mode Input Observer (Section 3.3, Kim, Drakunov, Wang)</td>
</tr>
<tr>
<td>Parameter Identification</td>
<td>Sliding Mode Identification (Section 3.2, Utkin)</td>
</tr>
</tbody>
</table>

Table 3.1: Estimation and identification methods in fault diagnosis

Some well-known estimation and identification procedures are introduced in Section 3.2. They are linear/nonlinear sliding mode observer and sliding mode identification. New estimation algorithms developed this study are listed (in italic character) in Table 3.1. These are:

i) A method for observer design using binary measurements

ii) A sliding mode input observer

These algorithms are explored in detail in Section 3.3 and their application to an engine air-fuel management system is discussed in Chapter 4.

3.2. Estimation and Identification Problems in Sliding Mode

Fundamental aspects of sliding mode methodology are explained in this section. The early part deals with the sliding mode control methodology. The so-called equivalent control method, which is one of the key concepts in sliding mode methodology, is explained. Later parts introduce observer design methods for both linear and nonlinear systems. Also, parameter identification methods based on sliding mode are illustrated. The contents of this section are already well known in the literatures, for example, Utkin (1978, 1992) Draknov (1995) and Krishnaswami (1996). Some of the concepts introduced here are utilized in
following sections. Also various observer/identification design methods explained in this section are applied to control and estimation problems in Chapter 4. Therefore, the objectives of this section can be summarized as an introduction to the fundamental philosophy behind sliding methodology and to observer/identification design procedures.

3.2.1 Introduction to Sliding Mode Methodology

The basic idea of sliding mode control is to constrain the motion of the state trajectories to some predefined manifolds (sliding manifold) in the system state space. To this end, a switching control law determined by the equation of the sliding manifold is applied to the system as the control input. A major advantage of sliding mode control is the use of a switching control action, which is simple to implement. In addition, properties such as order reduction, decoupling of the design procedure from system parameters, disturbance rejection and insensitivity to parameter variations can be achieved.

To introduce the general form of sliding mode control, consider a system described by

\[ \dot{x} = f(x,u,t) \]  (3.9)

where \( x \in \mathbb{R}^n \), \( u \in \mathbb{R}^m \). The control design consists of two stages: i) selection of a sliding manifold \( s_i(x)=0 \) as a function of the state vector \( x \); and ii) the design of continuous control functions \( u_i^+ \) and \( u_i^- \) away from the sliding manifold \( s_i(x)=0 \). Control action is defined as a function of the sliding manifold:

\[ u_i = \begin{cases} u_i^+(x,t) & \text{if } s_i(x,t) > 0 \\ u_i^-(x,t) & \text{if } s_i(x,t) < 0 \end{cases} \quad i = 1, \ldots, m. \]  (3.10)

Existence of a sliding mode is guaranteed as long as the following pair of inequalities hold:

\[ \lim_{t \to \infty} \dot{s}_i > 0 \quad \text{and} \quad \lim_{t \to \infty} \dot{s}_i < 0. \]  (3.11)

A detailed derivation and proof can be found in Utkin (1978, 1992).
3.2.2 Sliding Mode Control Design by Equivalent Control Method

To assign the desired (stable) dynamics to the sliding manifold, sliding mode equation(s) should be found first. If the control that can drive the state trajectories to the sliding manifold is known, a formal technique called equivalent control method is used to find the sliding mode equation(s). In this method, a time derivative of the vectors \( s \) along the system trajectory is set equal to zero, and the resulting algebraic system is solved for the control vector. The solution, called equivalent control \( u_{eq} \), is then substituted into the original system equation (3.1); the resulting equations are the sliding mode equation(s). From a geometrical point of view, the above procedure corresponds to replacing a discontinuous control by a continuous one which drives the time derivative of the manifold \( s \) to origin along the intersection of the discontinuity surfaces.

To illustrate the design of sliding mode control using the equivalent control method, the following system is considered.

\[
\dot{x} = f(x,t) + B(x,t)u
\]  

(3.12)

where \( x \in \mathbb{R}^n \), \( u \in \mathbb{R}^n \) and the control \( u \) is determined by (3.10). The time derivative of sliding manifold \( s(x) \in \mathbb{R}^n \) can be then expressed as follows.

\[
\dot{s} = \frac{\partial s}{\partial x} \dot{x} = G(x)[f(x,t) + B(x,t)u] = G(x)f(x,t) + G(x)B(x,t)u(t)
\]  

(3.13)

\[
G(x) = \frac{\partial s}{\partial x}
\]  

(3.14)

The equivalent control is obtained by setting \( \dot{s} = 0 \), and solving for the control:

\[
u_{eq} = -[G(x)B(x,t)]^{-1}G(x)f(x,t)
\]  

(3.15)

To find the sliding mode equation, substitute (3.15) into (3.12).

\[
\dot{x} = f(x,t) - B(x,t)[G(x)B(x,t)]^{-1}G(x)f(x,t)
\]  

\[
= \left[I - B(x,t)[G(x)B(x,t)]^{-1}G(x)\right]f(x,t)
\]  

(3.16)

From \( s(x)=0 \), we find the relationship between \( x_2 \) and \( x_i \):

\[
x_2 = s_0(x_i) \in \mathbb{R}^n
\]  

(3.17)
From (3.16) and (3.17), the equation of sliding mode can be obtained as
\[ \dot{x}_i = f_i(x_i, t, s_i), \quad x_i \in \mathbb{R}^{n \times m} \] (3.18)

By looking at the sliding mode equation (3.18), several attractive features of sliding mode control can be observed:

1. the order of the control problem is reduced (to dimension \( n \cdot m \))
2. the sliding mode does not depend on the control \( u \)
3. the sliding mode depends on the sliding manifold equation, and is therefore not dependent on system parameters

Invariance of the sliding mode with respect to plant parameter changes and/or disturbances can be seen by considering the following system.
\[ \dot{x} = f(x, t) + B(x, t)u + d(x, t) \] (3.19)

For invariance to disturbance \( d(t) \), the following condition needs to be satisfied - referred as Invariance condition or Matching condition,
\[ d \in \text{range}B \] (3.20)

If equation (3.20) holds, then there exist \( \lambda \) such that \( d = B\lambda \), and equation (3.19) can be written as
\[ \dot{x} = f(x, t) + B(x, t)(u + \lambda) \] (3.21)

Use of the equivalent control method to find the sliding mode equation results in
\[ \dot{s} = G(x)f(x, t) + B(x, t)(u + \lambda) = 0, \quad G(x) = \frac{\partial s}{\partial x}. \] (3.22)

Solving equation (3.22) with respect to the control,
\[ u_{sl} + \lambda = -[G(x)B(x, t)]^{-1}G(x)f(x, t) \] (3.23)

we substitute (4.23) into (4.21) to find the sliding mode equation:
\[ \dot{z} = f(x, t) - B(x, t)[G(x)B(x, t)]^{-1}G(x)f(x, t). \] (3.24)

The sliding mode equation (3.24) does not depend on disturbance \( d(x, t) \). Therefore, the resulting system response is invariant to the disturbance.
3.2.3 Estimation and Identification Problems in Sliding Mode

A similar approach to sliding mode control can also be applied to identification and estimation problems. The sliding mode approach to identification and estimation in dynamic systems consists of designing a model with discontinuous parameters and enforcing sliding modes such that the model and plant outputs coincide. Then, the average values of the discontinuous parameters depend on the unknown states, parameters and disturbances, and may be used for their estimation.

The general principle of a sliding mode estimator can be stated for the system (3.25) as follows.

\[ \dot{x} = f(x, t, \theta) \]  
\[ (3.25) \]

where \( x \in \mathbb{R}^n \) is known state vector, \( \theta \) can be an unknown vector parameter and disturbances. Define a model of the form

\[ \dot{\hat{x}} = f(\hat{x}, t, \theta_0) + V \]  
\[ (3.26) \]

with \( \hat{x} \) is the estimate of the state \( x \), and \( \theta_0 \) is nominal values of \( \theta \). The auxiliary input \( V \) is discontinuous function of the mismatch \( s = x - \hat{x} \) as given below:

\[ V = -M \text{sign}(s), \quad M > \sup |f(\hat{x}, t, \theta_0) - f(x, t, \theta)| \]  
\[ (3.27) \]

Sliding mode occurs after a finite time with \( s = 0 \); then \( \hat{x} \rightarrow x \).

In addition to state estimation, an equivalent value of the discontinuous function \( V_{eq} = f(\theta_0) - f(\theta) \) can be obtained by low-pass filtering \( V \) to filter out the high frequency components. This function \( V_{eq} \) depends on the unknown parameters or disturbances and it can be used to estimate the unknown parameters and disturbances. Theoretical details can be found in Utkin (1992).
Sliding Mode Observer for Linear Systems

Two different design procedures for the linear system observer are considered. The conventional design procedure is rather straightforward, as in the case with the Luenberger type observer design. The second design procedure requires the concept of equivalent control in the observer and the so-called equivalent filter.

Conventional Method

Consider a linear time invariant system described in state space as follows.

\[ \dot{x} = Ax + Bu, \quad x \in \mathbb{R}^{n} \]  \hspace{1cm} (3.28)

and the measured output variables

\[ y = Cx, \quad y \in \mathbb{R}^{p}, \quad \text{rank} C = p \]  \hspace{1cm} (3.29)

It is assumed that the matrices \( A, B \) and \( C \) are known, and that the pair \( \{A, C\} \) is observable. We write equation (3.28) for the new variables \( x_1 \) and \( y \):

\[ \dot{y} = A_{11}y + A_{12}x_1 + Bu \]  \hspace{1cm} (3.30)

\[ \dot{x}_1 = A_{21}y + A_{22}x_1 + Bu \]  \hspace{1cm} (3.31)

An observer to estimate \( x_1 \) and \( y \) in the system (3.28) is of the form

\[ \dot{\hat{y}} = A_{11}\hat{y} + A_{12}\hat{x}_1 + Bu + L_1v \]  \hspace{1cm} (3.32)

\[ \dot{\hat{x}}_1 = A_{21}\hat{y} + A_{22}\hat{x}_1 + Bu + L_2v \]  \hspace{1cm} (3.33)

\[ v = \text{sign}(e_y), \quad [\text{sign}(e_y)]^T = [\text{sign}(e_1) \cdots \text{sign}(e_r)] \]  \hspace{1cm} (3.34)

where \( L_1 \) and \( L_2 \) are observer gain matrices and the estimation error is defined as \( e_y = y - \hat{y} \) and \( e_1 = x_1 - \hat{x}_1 \). From equation (3.30)-(3.34), we obtain equations for the error dynamics:

\[ \dot{e}_y = A_{11}e_y + A_{12}e_1 - L_1v \]  \hspace{1cm} (3.35)

\[ \dot{e}_1 = A_{21}e_y + A_{22}e_1 - L_2v \]  \hspace{1cm} (3.36)

For sufficiently large \( L_1 \), sliding motion will begin on a sliding manifold \( e_y = 0 \), i.e.,

81
\[ L_t > |A_{te_i}| + |A_{se_i}| \]  

(3.37)

and when in sliding mode, the equivalent control becomes

\[ \nu_m = L_t A_{se_i} \]  

(3.38)

Substituting the equivalent control into equation (3.36), we obtain the sliding motion equation as

\[ \dot{e}_i = (A_{22} - L_t L_t^i A_{12}) e_i \]  

(3.39)

The sliding motion equation (3.39) is order of (n-p) and a Luenberger estimator design can be utilized to define the dynamics of \( e_i \) since we still have a design freedom in \( L_t \). Note that observability of the pair \( (A_{22}, A_{12}) \) follows from observability of the pair \( (A, C) \). Thus, after sliding motion occurs, \( e_i \) approaches zero (\( \hat{x}_i \) converges to \( x_i \)). The vector \( x_2 \), which together with \( x_i \) constitutes the desired state vector \( x \), is determined from (3.36)

**Design based on Equivalent Control method**

Consider the system representation in equation (3.30) and (3.31). Let \( \text{rank}(A_{12}) = m_j \leq m \), then the matrix \( A_{12} \) can be decomposed into

\[ A_{12} = D_j C_1 \quad D_j \in \mathbb{R}^{m \times k}, \quad C_1 \in \mathbb{R}^{n \times (n-k)} \]  

(3.40)

Introduce a new variable

\[ y_1 = C_1 x_1 \quad C_1 = D_j^T A_{21} \]  

(3.41)

where the superscript + denotes a left pseudo inverse. Now equation (3.30) can be divide into two parts as follows:

\[ \dot{y}_1 = A_{11} y_1 + A_{12} x_2 + B_{11} u \]  

(3.42)

\[ \dot{x}_2 = A_{21} y_1 + A_{22} x_2 + B_{21} u \]  

(3.43)

The observer for (3.42) is constructed as shown below

\[ \dot{\hat{y}}_1 = A_{11} \hat{y}_1 + A_{12} \hat{x}_2 + B_{11} u + L_n \text{sign}(y_1 - \hat{y}_1) \]  

(3.44)

Equation (3.44) requires the measurement of \( y_1 \), which is not available. However, from equations (3.38), (3.40) and (3.41), we know that
\[ y_t - \hat{y}_t = D^*_t (L_t \text{sign}(y - \hat{y}))_{eq} \]  

(3.45)

Using this information, the observer becomes

\[ \dot{\hat{y}}_t = A_{1t} \hat{y}_{t-1} + A_{2t} \hat{x}_t + B_{u} + L_t \text{sign}(D^*_t (L_t \text{sign}(y - \hat{y})))_{eq} \]  

(3.46)

By repeating this procedure \( n \) times (since \( x \in \mathbb{R}^n \)), the system can be represented as

\[ \dot{\hat{y}} = A_{11} \hat{y} + D_{21} y_1 + B_{u} \]
\[ \dot{\hat{y}}_1 = A_{12} \hat{y}_1 + D_{22} y_2 + B_{u} \]
\[ \vdots \]
\[ \dot{\hat{y}}_{n,t} = A_{1n} \hat{y}_{n,t-1} + D_{2n} y_{n} + B_{u} \]  

(3.47)

and the observer is of the form

\[ \hat{y} = A_{11} \hat{y} + D_{21} \hat{x}_1 + B_{u} + v_1 \]
\[ \dot{\hat{y}}_1 = A_{12} \hat{y}_1 + D_{22} \hat{x}_2 + B_{u} + v_2 \]
\[ \vdots \]
\[ \dot{\hat{y}}_{n,t} = A_{1n} \hat{y}_{n,t-1} + D_{2n} \hat{x}_n + B_{u} + v_n \]  

(3.48)

where

\[ v_i = L_i \text{sign}(z_i), \quad L_i > 0 \]  

(3.49a)

\[ v_i = L_i \text{sign}(z_i), \quad L_i > 0, \quad i = 2, \ldots, n \]  

(3.49b)

\[ z_i = y - \hat{y} \]  

(3.49c)

\[ \tau_i \dot{z}_i + z_i = D^*_i v_{i-1}, \quad \tau_i > 0, \quad i = 2, \ldots, n \]  

(3.49d)

For example, if the system only has two states, then the observer becomes

\[ \hat{y} = A_{11} \hat{y} + A_{12} \hat{x}_1 + B_{u} + v_1, \quad v_1 = L_1 \text{sign}(y - \hat{y}) \]  

(3.50a)

\[ \dot{\hat{y}}_1 = A_{21} \hat{y}_1 + A_{22} \hat{x}_2 + B_{u} + v_2, \quad v_2 = L_2 \text{sign}(z_2) \]  

(3.50b)

\[ \tau_i \dot{z}_i + z_i = v_i, \quad \tau_i > 0, \quad i = 2, \ldots, n \]  

(3.50c)

since in this case, we have \( y_1 = x_1, \quad D_2 = A_{12}, \quad D_4 = A_{22}, \quad A_{11} = A_{22}, \quad B_4 = B_2 \) and \( D^*_2 = 1 \).


Sliding Mode Observer for Nonlinear Systems

As in the case of linear systems, two approaches to the design of sliding mode observer for nonlinear systems are illustrated. They are a conventional method, and a design based on the equivalent control method.

Conventional Method

Consider a nonlinear system linear in the control, as follows.

\[ \dot{x} = f(x) + g(x)u \]
\[ y = Cx \]  \hspace{1cm} (3.51)

where \( x \in \mathbb{R}^n \) and the output of the system \( y \in \mathbb{R}^p \).

Define the following observer for the system in equation (3.51)

\[ \dot{\hat{x}} = f(\hat{x}) + g(\hat{x})u + Lz + Mv \]  \hspace{1cm} (3.52)

where \( L \) and \( M \) are the \( (n \times p) \) gain matrices to be defined, and discontinuous function \( v \) is

\[ v = \begin{bmatrix} \text{sign}(z_1) & \text{sign}(z_2) & \cdots & \text{sign}(z_p) \end{bmatrix}^\top \]  \hspace{1cm} (3.53)

with

\[ z_i = y_i - c_i \hat{x} \]  \hspace{1cm} (3.54)

and \( c_i \) is the \( i \)th row of the \( C \) matrix. Then, the estimation error dynamics become

\[ \dot{e} = \Delta f + \Delta gu - Lz - Mv \]  \hspace{1cm} (3.55)

where \( e = x - \hat{x} \), \( \Delta f = f(x) - f(\hat{x}) \) and \( \Delta g = g(x) - g(\hat{x}) \).

Define a sliding manifold as

\[ s = y - C\hat{x} = C(x - \hat{x}) \]  \hspace{1cm} (3.56)

The sliding mode on the \( p \)-dimensional sliding manifold \( s=0 \) is enforced if \( ss < 0 \).

\[ s = C\hat{x} - \hat{C}\hat{x} \]
\[ = Cf(x) + Cg(x)u - Cf(\hat{x}) - Cg(\hat{x})u - CLz - CMv \]  \hspace{1cm} (3.57)
\[ = C\Delta f + C\Delta gu - CLs - CM\text{sign}(s) \]
The gain matrix $M$ is selected such that
\[ M_i > |\Delta f_i| + |\Delta g_i u|, \quad i = 1, \cdots, p \] (3.58)

The linear part of the observer gain $L$ is selected to provide more damping in the observer (Hedrick, 1987).

To examine the sliding motion, first find the equivalent control by letting $s = \dot{s} = 0$
\[ \dot{s} = 0 = C\Delta f + C\Delta g u - CMv_{eq} \] (3.59)
and solve for $v_{eq}$.
\[ v_{eq} = (CM)^{+}(C\Delta f + C\Delta g u) \] (3.60)

By substituting equation (3.60) into equation (3.55), the sliding motion equation is obtained
\[ \dot{e} = \Delta f + \Delta g u - Lz - M(CM)^{+}(C\Delta f + C\Delta g u) \]
\[ = -LCe + \left( I - M(CM)^{+}C \right)\Delta f + \left( I - M(CM)^{+}C \right)\Delta g u \] (3.61)

By multiplying the matrix $C$ on both sides of equation (3.61), we obtain
\[ \dot{e} = -LCe + \left( I - M(CM)^{+}C \right)\Delta f + \left( I - M(CM)^{+}C \right)\Delta g u \] (3.62)
and
\[ \dot{e} = -LCe \] (3.63)

providing that $\Delta f$ and $\Delta g u$ are small or zero. In some cases, the nonlinear functions $f(x)$ and $g(x)$ can be approximated with combination of linear part and nonlinear part. For example, the first 2 terms in Taylor expansion of $f(x)$ can used instead of $f(x)$. The modeling error $\Delta f$ in this case is the high order terms greater than second order and convergence of the observer depends on $\Delta f$.

**Design based on Equivalent Control method**

A design procedure based on the equivalent control is examined for a nonlinear system. Again, consider a nonlinear system of the form
\[ \dot{x} = f(x) + g(x)u \] (3.64)
where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^p$ denotes state and input respectively. The output measurement equation is
\[ y = h(x) \] (3.65)

85
where $y \in \mathbb{R}^q$. A sliding mode observer of the following form (Krishnaswami, 1996) can be used to estimate state trajectories:

$$
\dot{x} = \left( \frac{\partial H(x)}{\partial x} \right)^{-1} M(x) \text{sign}(V(t) - H(\dot{x})) + B(\dot{x})u
$$

(3.66)

where

$$
H(x) = \text{col} \begin{pmatrix} h_1(x) & \cdots & h_n(x) \end{pmatrix}
$$

(3.67a)

$$
h_i(x) = h(x)
$$

(3.67b)

$$
h_i(x) = \frac{\partial h_i(x)}{\partial x} f(x) \quad i = 1, \ldots, n
$$

(3.67c)

$$
V(t) = \begin{bmatrix} v_1(t) & v_2(t) & \cdots & v_n(t) \end{bmatrix}
$$

(3.67d)

$$
v_i(t) = y(t)
$$

(3.67e)

$$
v_{i+1}(t) = (m_i(\dot{x}) \text{sign}(v_i(t) - h_i(\dot{x}(t))))_u, \quad i = 1, \ldots, n - 1
$$

(3.67f)

$$
M(\dot{x}) = \text{diag}(m_1(\dot{x}) \quad m_2(\dot{x}) \quad \cdots \quad m_n(\dot{x}))
$$

(3.67g)

The following conditions should be satisfied to assure the convergence of the observer.

1. Functions $f(x)$ and $h(x)$ are smooth enough that all the partial derivatives introduced above exist and are continuous.

2. The Jacobian of the diffeomorphism $H: \mathbb{R}^n \rightarrow \mathbb{R}^n$ is nondegenerating in the domain $X \subset \mathbb{R}^n$, i.e.:

$$
\left[ \begin{array}{c}
\det \frac{\partial H(x)}{\partial x} \\
\frac{\partial H(x)}{\partial x} B(x) \\
\end{array} \right] \geq \delta > 0
$$

(3.68)

for some $\delta > 0$ and every $x \in X$. This means the map $H$ is an injection and provides a global observability condition.

3. For any $x \in X$

$$
\frac{\partial}{\partial x} \left[ \frac{\partial H(x)}{\partial x} B(x) \right] = 0 \quad \forall x
$$

(3.69)

Provided that the above condition is satisfied, the convergence of the observer can be guaranteed. The proof of the convergence is found in Drakunov(1993) and is rather lengthy, and thus not included here. A summary of the procedure can be stated as follows. Since the map $H: X \rightarrow \mathbb{R}^n$ is an injection, it is sufficient
to show that the modified estimation error \( e(t) = H(x(t)) - H(\hat{x}(t)) \) converges to zero in finite time. After defining the estimation error dynamics, a similar procedure for the linear case can be adopted to prove the convergence step by step.

For systems that do not satisfy the equation (3.61), Krishnaswami(1996) proposed an extension of the sliding observer scheme, assuming derivatives of the inputs are available. Consider once again the system in equations (3.64) and (3.65), and define,

\[
y(t) = h_i(x)
\]

\[
y(t) = \frac{\partial h(x)}{\partial x} \frac{dx}{dt} + \frac{\partial h(x)}{\partial u} \frac{du}{dt} = h_i(x, u)
\]

\[
y(t) = \frac{\partial h_i(x, u)}{\partial x} \frac{dx}{dt} + \frac{\partial h_i(x, u)}{\partial u} \frac{du}{dt} = h_i(x, u, \hat{u})
\]

\[
\vdots
\]

\[
y^{(i)}(t) = \frac{\partial h_i(x, u, \cdots, u^{i-1})}{\partial x} \frac{dx}{dt} + \frac{\partial h_i(x, u, \cdots, u^{i-1})}{\partial u} \frac{du}{dt} + \cdots + \frac{\partial h_i(x, u, \cdots, u^{i-1})}{\partial u^{i-2}} \frac{du^{i-2}}{dt} = h_i(x, u, \cdots, u^{i-1})
\]

Let,

\[
H(x, u, \cdots, u^{i-1}) = \begin{bmatrix} h_1(x) & h_2(x, u) & \cdots & h_i(x, u, \cdots, u^{i-1}) \end{bmatrix}
\]

Then the observer can be constructed as follows.

\[
\dot{x} = \left( \frac{\partial \hat{H}}{\partial \hat{x}} \right)^{-1} \left( M(\hat{x}) \text{sign}(V(t) - \hat{H}) - \left[ \frac{\partial \hat{H}}{\partial \hat{u}} \hat{u} + \frac{\partial \hat{H}}{\partial \hat{u}} \hat{\hat{u}} + \cdots + \frac{\partial \hat{H}}{\partial \hat{u}^{i-1}} \hat{u}^{i-1} \right] \right)
\]

where,

\[
\hat{H} = H(\hat{x}, u, \cdots, u^{i-1})
\]

\[
V(t) = \begin{bmatrix} v_1(t) & v_2(t) & \cdots & v_i(t) \end{bmatrix}
\]

\[
v_i(t) = y(t)
\]

\[
v_{i,a}(t) = \left( m_i(\hat{x}) \text{sign}(V(t) - h_i(\hat{x}(t))) \right)_{a_i}, \quad i = 1, \cdots, n - 1
\]
\[ M(\hat{x}) = \text{diag}(m_1(\hat{x}), m_2(\hat{x}), \ldots, m_s(\hat{x})) \]  

(3.73e)

where, \( M \in \mathbb{R}^{m \times m} \) and the \( m_i \)'s are positive elements. The subscript 'eq' in equation (3.73d) denotes equivalent control value, meaning that the right hand side of the equation is low pass filtered. By appropriate choice of \( M \), the observer converges (\( \hat{x} \to x \)) in finite time interval. Proof of convergence of this observer can be found in Drakunov (1992) and Krishnaswami (1996).

**Parameter Identification using Sliding Mode**

In order to determine the elements of the matrices \( A \) and \( B \) in (3.28) from the measured state vector \( x \) and control vector \( u \) with linearly independent components, the following identification algorithm can be used.

\[ \dot{y} = \hat{A}x + \hat{B}u + \nu \]  

(3.74)

where

\[ \hat{A} = -\lambda \nu x^T, \quad \lambda > 0 \]  

(3.75a)

\[ \hat{B} = -\lambda \nu u^T \]  

(3.75b)

\[ \nu = \Psi^A x + \Psi^B u, \quad \Psi^A = \| \Psi^A \|, \quad \Psi^B = \| \Psi^B \|, \quad y, \nu \in \mathbb{R}^s \]  

(3.75c)

\[ \Psi^A_{ij} = \begin{cases} \alpha_{ij}^A & \text{for } x_i s_j > 0, \\ \beta_{ij}^A & \text{for } x_i s_j < 0, \end{cases} \]  

(3.75d)

\[ \Psi^B_{ij} = \begin{cases} \alpha_{ij}^B & \text{for } u_i s_j > 0, \\ \beta_{ij}^B & \text{for } u_i s_j < 0, \end{cases} \]  

(3.75e)

and \( s_i = x_i - y_i \), \( \| \| \) indicates the \( L_2 \) norm. For the dynamic system in equations (3.28) and (3.74), a sliding mode appears on the manifold \( s = 0 \) (\( s = x - y \)), and its mathematical description in equation (3.75) requires insertion of an equivalent control \( \nu_{eq} = \Delta A x + \Delta B u \), which is the solution for \( \nu \) of the equation \( \dot{s} = 0 \) (\( \Delta A = A - \hat{A}, \Delta B = B - \hat{B} \)). Denoting \( \Delta C = (\Delta A, \Delta B) \) and \( \tilde{z} = (x^T, u^T) \), we construct a Lyapunov function of the form.
\[ V = \frac{1}{2} \text{tr}(\Delta C \Delta C^T) \geq 0 \]  \quad (V = 0, \text{if } \Delta A = 0, \Delta B = 0) \tag{3.76} \\

We calculate the time derivative of \( V \) on the trajectories of system (3.28) or the system \( \dot{\Delta}C = -\lambda \varepsilon \Delta C \) after the sliding mode occurs (i.e., when \( v = v_{eq} = \Delta C z \)):

\[ \dot{V} = -\lambda \text{tr}(\Delta C z \varepsilon^T \Delta C^T) = -\lambda \|w\|^2, \quad w = \Delta C z \tag{3.77} \]

Since the components of the vector \( z \) are linearly independent, we have \( \dot{V} < 0 \) for \( z \neq 0 \), and therefore \( V \) is a monotonically decreasing function. If we impose additional conditions such that for any \( \Delta C = \text{constant} \), the integral \( \int_0^\infty \|w\|^2 \, dt \) diverges, then

\[ \lim_{t \to \infty} V = 0, \quad \lim_{t \to \infty} \Delta C = 0 \quad \text{or} \quad \lim_{t \to \infty} \Delta A = 0, \quad \lim_{t \to \infty} \Delta B = 0 \tag{3.78} \]

Consequently, the components of the matrices \( \hat{A} \) and \( \hat{B} \) tend to the values of the parameters of the matrices \( A \) and \( B \) and, as a result, the identification problem can be solved. Unlike methods based on use of reference models, these methods are not assumed to possess asymptotic stability and accordingly the Lyapunov function is independent of the mismatch between the state vector \( s \) of the system and the supplementary model (in our case, the supplementary model plays the role of the reference model). In order to realize the algorithm proposed here, there is no necessity for direct measurement of the derivative of the state vector or for approximate differentiation with the aid of physically realizable components.

**Simultaneous Estimation and Identification**

For a system represented in canonical form as

\[ \dot{x}_i = x_{i+1}, \quad i = 1, \ldots, n - 1 \tag{3.79a} \]
\[ \dot{x}_n = a^T x + bu \tag{3.79b} \]
\[ y = x_1 \tag{3.80} \]

where \( x^T = [x_1 \quad x_2 \quad \ldots \quad x_n] \) and \( a^T = [a_1 \quad a_2 \quad \ldots \quad a_n] \), \( u \) is a control input, \( y \) is an output. The parameter vector \( a^T \) and parameter \( b \) are unknown.
The identification algorithm can be formulated as

\[
\begin{align*}
\dot{x}_i &= \dot{x}_{i+1} + v_i & i = 1, \ldots, n-1 \\
\dot{x}_n &= \hat{a} \dot{x} + \hat{b} u + v_x 
\end{align*}
\] (3.81a, 3.81b)

with

\[
\begin{align*}
v_i &= M_i \text{sign}(z_i) & M_i > 0 & i = 1, \ldots, n \\
\tau_i \dot{z}_i + z_i &= v_{r,i} & \tau_i > 0 & i = 2, \ldots, n \\
z_i &= x_i - \hat{x}_i \\
\hat{a} &= -\lambda v_x \dot{x} & \lambda > 0 \\
\hat{b} &= -\lambda v_x u
\end{align*}
\] (3.82a, 3.82b, 3.82c, 3.82d, 3.82e)

where \( \dot{x}, \hat{a}, \hat{b} \) represent the estimate of state \( x \), parameter \( a \) and \( b \) respectively. Define the estimation error \( e = x - \dot{x} \), \( \Delta a = a - \hat{a} \), \( \Delta b = b - \hat{b} \), then the error dynamics become:

\[
\begin{align*}
\dot{e}_i &= e_{i+1} - v_i & i = 1, \ldots, n-1 \\
\dot{e}_x &= \hat{a} \dot{e} + \Delta a \dot{x} + \Delta b u - v_x 
\end{align*}
\] (3.83a, 3.83b)

Now, the design parameter \( M_i \) and \( r_i \) are chosen such that the error \( e \), \( \Delta a \) and \( \Delta b \) become zero. This can be achieved by following the \( n \) step procedure below.

First, starts with \( e_x \). Select \( M_1 \) such that sliding manifold \( e_x = x - \dot{x} \) can be reached using \( v_i = M_1 \text{sign}(x - \dot{x}) \), i.e., \( M_1 > |e_x| \). Then, after sliding mode occur, the equivalent value of the discontinuous function \( v_1 \) can be obtained as \( v_{xq} = e_x \). This value can be obtained by a filter defined in equation (3.82) with time constant \( r_1 \) and \( v_{xq} = z_2 \).

Next, consider \( e_2 \). Again, we select \( M_2 \) such that the sliding manifold \( e_2 = x - \dot{x} \) can be reached using \( v_i = M_2 \text{sign}(z_i) \), i.e., \( M_2 > |e_2| \). After sliding mode is achieved, the equivalent value of the discontinuous function \( v_2 \) becomes \( v_{2q} = e_2 \). This value can be obtained by the filter defined in equation (3.82) with time constant \( r_2 \) and \( v_{2q} = z_3 \).
This step is repeated until sliding motion along \( e_{e+1} = x_{e+1} - \hat{x}_{e+1} \) is achieved. Finally, these estimates are used in conjunction with the equation (3.75) to estimate the parameter vectors \( a \) and \( b \).

3.3. Estimation Methods for Special Cases

Two special estimation problems are explored in this section. An input estimation is introduced first. An estimation structure for input is generally known as unknown input observer. Here, an input estimation algorithm using sliding mode methodology is developed. This approach has been applied to engine indicated torque (or in-cylinder pressure) estimation problem (Drakunov et. al, 1994; Wang et. al, 1995). Another special type of observer is proposed for estimation with binary type measurement. This observer structure is useful for IC engine application where HEGO sensor (Heated Exhaust Gas Oxygen sensor, sensor response is almost binary type) is only available measurement for air-fuel ratio.

3.3.1. Input Estimation using Sliding Mode

Methods of an input estimation using the sliding mode method are described in this section.

Consider a simple first order system:

\[
\dot{x} = f(x) + g(x)u \tag{3.84}
\]

where \( u \) is the input to be identified.

Define a supplementary model as

\[
\dot{y} = f(y) + g(y)v \tag{3.85}
\]

where

\[
v = -M\text{sign}(e) \quad e = x - y, \quad M > 0 \tag{3.86}
\]

The error dynamics are then:

\[
\dot{e} = \dot{x} - \dot{y} = f(x) - f(y) + g(x)u - g(y)v \tag{3.87}
\]

With a proper choice of gain \( M \), a sliding mode will occur on the sliding manifold \( s = e = x - y \). The sliding mode \( (s=0, x=y) \) can be described by
\[ \dot{s} = g(y)[u - \nu] \]  

(3.88)

After sliding mode occurs, the average value of the discontinuous function \( \nu \) approaches to the control input \( u \). The average value of \( \nu_{eq} \) can be obtained by low pass filtering the discontinuous function \( \nu \).

\[ \tau \dot{z} + z = \nu \]  

(3.89)

where \( z = \nu_{eq} = u \).

Let us consider a second order example typically found in mechanical system as follows.

\[
\begin{align*}
J \ddot{x} + g(x, \dot{x}, u) + f(x, \dot{x}) &= 0 \\
y &= x
\end{align*}
\]  

(3.90)

where \( J \) is a inertia, \( g \) is known function (e.g. nonlinear stiffness, nonlinear damping, control input) and \( f \) is unknown input to be estimated. The system in equation (3.90) can be represented in state space as

\[
\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= -J^{-1}g(x_1, x_2, u) - J^{-1}f(x_1, x_2) \\
y &= x_1
\end{align*}
\]  

(3.91)

where \( x_1 = x \) and \( x_2 = \dot{x} \).

An unknown input observer using sliding mode method can be designed as follows. First design an observer for \( x_i \) using the first equation in (3.91)

\[ \dot{\hat{x}}_i = \dot{\hat{x}}_i + L_i \text{sign}(y - \hat{x}_i) \]  

(3.92)

The estimation error \( (e_i = x_i - \hat{x}_i, \ i = 1, 2) \) dynamics become

\[ \dot{e}_i = e_i - L_i \text{sign}(e_i) \]  

(3.93)

If the gain \( L_i > |e_i| \), then a sliding mode occurs on a sliding manifold \( s_i = e_i = 0 \). And the equivalent value of \( L_i \text{sign}(e_i) \) satisfies

\[ L_i \text{sign}(e_i)_{eq} = e_i. \]  

(3.94)

The equivalent value of \( L_i \text{sign}(e_i) \) can be obtained using a low pass filter (also called as equivalent filter) as follows.

\[ \tau_i \dot{v}_i + v_i = L_i \text{sign}(e_i) \]  

(3.95)
where $\tau_i$ is a filter time constant and $v_i = e_i$ is the equivalent value of $L_i \text{sign}(e_i)$. Now design an observer for $x_2$ using $v_i$ as follows.

$$\dot{\hat{x}_2} = -J^{-1}g(\hat{x}_1, \hat{x}_2, u) + L_2 \text{sign}(v_i) \tag{3.96}$$

The estimation error dynamics for $x_2$ becomes:

$$\dot{e}_2 = -J^{-1}g(x_1, x_2, u) + J^{-1}g(\hat{x}_1, \hat{x}_2, u) - J^{-1}f(x_1, x_2) - L_2 \text{sign}(e_2) \tag{3.97}$$

If the gain $L_2$ is large enough, i.e., $L_2 > | -J^{-1}g(x_1, x_2, u) + J^{-1}g(\hat{x}_1, \hat{x}_2, u) - J^{-1}f(x_1, x_2) |$, then a sliding mode can be enforced along the sliding manifold $s_2 = e_2 = 0$. As $e_2$ approaches zero, $\hat{x}_2$ approaches $x_2$.

Furthermore, the equivalent value of $L_2 \text{sign}(e_2)$ becomes:

$$L_2 \text{sign}(e_2)_w = -J^{-1}f(x_1, x_2) \tag{3.98}$$

Therefore, an estimate of unknown input, $f(x_1, x_2)$ can be obtained by applying another equivalent filter to $L_2 \text{sign}(e_2)$.

$$\tau_i \dot{e}_2 + v_2 = L_2 \text{sign}(e_2) \tag{3.99}$$

where $\tau_i$ is a filter time constant and $v_2$ is the equivalent value of $L_2 \text{sign}(e_2)$. Finally, the estimate of $f(x_1, x_2)$ is then expressed as:

$$\hat{f}(x_1, x_2) = -Jv_2 \tag{3.100}$$

As shown above, both state and unknown input can be estimated using unknown input sliding mode approach.

The main advantage of using this estimation method is that it does not require a complex gain selection procedure for the general type of nonlinear observer, nor differentiation of measured variables. Furthermore, convergence can be achieved in a finite time.
3.3.2. Observer using Binary Measurement

As long as a system is observable, the state variables of a system can be estimated using the conventional observer even though the robustness to disturbance and modeling uncertainty still needs to be considered further. However, the conventional observer approach is no longer valid when using a binary sensor since information is limited to the sign of the output.

A design methodology for such systems was introduced by Drakunov and Utkin (1995). They suggested using a two-stage observer consisting of a discrete time varying system observer to estimate the states at each sensor-switching instant, and of a continuous system observer which estimate the state trajectory between the switching instants. Generally speaking, the observer equations (first stage observer for a discrete time varying system) are time varying and the conventional eigenvalue placement methodology is not applicable. To overcome this difficulty, a new observer design approach is explored next to provide a guaranteed convergence of the observer.

*Design of Observer*

Consider the system

\[
\begin{align*}
\dot{x} &= Ax + Bu \\
y &= \text{sign}(x)
\end{align*}
\]  

(3.101)

where \( x \) is the state, \( u \) is the control and \( y \) is the output. Let \( z_k \) be the states at sensor switching instants and \( \tau_k \) be the interval between switching instants:

\[
\begin{align*}
z_k &= x(t_k) \\
\tau_k &= t_k - t_{k-1}
\end{align*}
\]  

(3.102)

If the \( z_k \)'s are known exactly for the entire process, the estimation of the states becomes an initial-value problem with known initial conditions:

*If all the states are available at every sensor switching event*

\[
y = z_k
\]  

(3.103)
then estimation of the continuous state variables during the sensor switching interval becomes an initial value problem with the state variables at the switching instant \( t_k \) as the initial condition; consequently the state trajectory until the next sensor switching instant \( t_{k+1} \) can be calculated by

\[
x(t) = e^{A(t-t_k)}z_k + \int_{t_k}^{t} e^{A(t-\tau)}Bu(\tau)d\tau
\]  

(3.104)

However, if information on the \( z_k \)'s is limited, that is the case with the binary sensor measurement, a discrete observer is needed to estimate the discrete sequences \( z_k \). Then, the continuous states \( x(t) \) can be predicted by using the discrete estimation of \( x(t_k) \) as the initial value at the sensor switching event (say, time between \( t_k \) and \( t_{k+1} \)) according to equation (3.104). Now the measurement equation becomes

\[
y = \text{sign}(x)
\]  

(3.105)

As mentioned above, the observer design procedure is divided into two steps. The first step consists of estimating the state variables at each switching instant of the binary sensor, i.e., \( \hat{z}_k \) = estimation of \( z_k \) at time \( t_k \). The second step consists of estimating the states between the sensor switching times, i.e., \( \hat{x}(t) \) = estimation of \( x(t) \) for the time interval between \( t_k \) and \( t_{k+1} \). This two-step procedure leads to a continuous estimate of the state variables for the entire process.

Using the definition given in (3.103), the discrete system (defined only at sensor switching instant) can be expressed as follows:

\[
z_k = e^{A_{k-1}}z_{k-1} + \int_{t_{k-1}}^{t_k} e^{A(t-\tau)}Bu(\tau)d\tau
\]  

(3.106)

A discrete observer can be designed to estimate the sequence \( z_k \). In this case, the updates of the estimates of these discrete state variables occur in non-uniformly distributed time intervals (time between \( t_{k-1} \) and \( t_k \)) which are the intervals between binary sensor switching times. The discrete observer can be constructed as shown in (3.107):

\[
\hat{z}_k = e^{A_{k-1}}\hat{z}_{k-1} + \int_{t_{k-1}}^{t_k} e^{A(t-\tau)}Bu(\tau)d\tau + L_{k-1}(z_{k-1} - \hat{z}_{k-1})
\]  

(3.107)
The selection of the gain for this type of observer is not a trivial task since the system in question is time-varying (because the sensor switching interval \( \tau \) varies in time). The approach based on eigenvalue placement is not applicable to the time varying system. For notational simplicity, redefine the observer given by equation (3.107) as

\[
\dot{\hat{z}}_k = F_{k-1} \hat{z}_{k-1} + G_{k-1} \{u(t)\}^{k}_{\zeta_k} + L_{k-1}(\hat{z}_{k-1} - z_{k-1})
\]  

(3.108)

where

\[
F_{k-1} = e^{\lambda_{k-1} \tau}
\]  

(3.109a)

\[
G_{k-1} \{u(t)\}^{k}_{\zeta_k} = \int_{\zeta_k}^{\tau} e^{A(\tau-\tau')}Bu(\tau)d\tau
\]  

(3.109b)

To provide convergence of the observer in general cases, a deadbeat observer structure is selected with transformation of the system into a specific canonical system representation. Transform the system as

\[
w_k = \tilde{F}_{k-1}w_{k-1} + \tilde{G}_{k-1} \{u(t)\}^{k}_{\zeta_k}
\]  

(3.110)

with transformation matrix \( T \)

\[
w_{k-1} = T_{k-1}x_{k-1}
\]  

(3.111a)

\[
\tilde{F}_{k-1} = T_{k-1}F_{k-1}T^{-1}_{k-1}
\]  

(3.111b)

\[
\tilde{G}_{k-1} \{u(t)\}^{k}_{\zeta_k} = G_{k-1} \{u(t)\}^{k}_{\zeta_k} T^{-1}_{k-1}
\]  

(3.111c)

such that the system transition matrix has the following structure

\[
\tilde{F}_{k-1} = \begin{bmatrix}
0 & \cdots & 0 & f_{1k-1} \\
1 & \ddots & 0 & 0 \\
\vdots & \ddots & 1 & 0 \\
0 & \cdots & 1 & f_{\alpha k-1}
\end{bmatrix}
\]  

(3.112)
where the $f_{k,i}$'s depend on $F_{k,i}$. Then, the gain is selected such that the effect of the time varying parameter in the state transition matrix is canceled by feedback, i.e., the deadbeat observer can be designed by selecting the gain $L_{k,i}$ as

$$L_{k,i} = \begin{bmatrix} -f_{u,i} \\ \vdots \\ -f_{d,i} \end{bmatrix}$$

(3.113)

The resulting observer has the following structure

$$\hat{w}_{k} = \bar{F}_{k,i} \hat{w}_{k-1} + \bar{G}_{k,i} \{u(t)\}_{k,i} + L_{k,i} (\hat{w}_{k-1} - w_{k-1})$$

(3.114)

This deadbeat observer converges with $n$ step where $n$ is the dimension of the system. Now the discrete event system is estimated using the observer designed above. The next step is to construct the continuous states between the discrete intervals. Knowing the estimated states at each switching instant, the continuous states (states between switching) can be calculated as described by equation (3.104).

**Proof of convergence**

Define the estimation error of the discrete sequence $z_k$ as

$$\tilde{z}_k = z_k - \hat{z}_k$$

(3.115)

then,

$$\tilde{z}_k = (F_{k,i} - L_{k,i}) \tilde{z}_{k-1}$$

(3.116)

since the discrete observer has a deadbeat structure, the estimation error $\tilde{z}_k$ will converge to zero in $n$ (dimension of the system) steps. As $\tilde{z}_k \to 0$ after $n$ steps, $\hat{z}_k \to z_k$. Now, define the estimation error for continuous state variable $x(t)$ as

$$\tilde{x}(t) = x(t) - \hat{x}(t)$$

(3.117)

then again,

$$\tilde{x}(t) = e^{A(t)-nI} \tilde{z}_k$$

(3.118)

As $\tilde{z}_k \to 0$ after $n$ steps, the estimation error also approaches to zero, i.e., $\tilde{x}(t) \to 0$. Therefore, $\hat{x}(t) \to x(t)$ after $n$ step during the time interval $t_{nk}$ and $t_{nk+1}$.  

97
The observer structure developed above is also applicable to systems in which only discrete information on outputs is available. It should be noted that the convergence of the proposed observer is affected by parameters of the system. Since the observer gains are determined based on the system parameters and the sensor-switching interval, exact information on those quantities is crucial.

**Example: Observer with binary measurement**

Consider a linear system where only binary information on the output is available as follows.

\[
\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u, \quad y = \text{sign}(x), \quad u = -10 y
\]  
(3.119)

First, find the discrete system defined only at sensor switching point) using equations (3.108) and (3.109).

\[
z_k = F_{k,1} z_{k-1} + G_{k,1} \begin{bmatrix} u(t) \end{bmatrix}_{k-1}, \quad F_{k,1} = \begin{bmatrix} 1 & \tau_{k-1} \\ 0 & 1 \end{bmatrix}, \quad G_{k,1} \begin{bmatrix} u(t) \end{bmatrix}_{k-1} = \begin{bmatrix} \int_{t_{k-1}}^{t_k} (\tau - t_{k-1}) u(\tau) d\tau \\ \int_{t_{k-1}}^{t_k} u(\tau) d\tau \end{bmatrix}
\]  
(3.120)

Since the matrix $F_{k,1}$ is already in the form of (3.112), we don’t need to transform the system (note that with measurement of $x_i$, the row of the matrix $F_{k,1}$ in equation (3.112) should be in reverse order to have a deadbeat structure).

Second, design an observer for the discrete system using relation given in equations (3.112) and (3.113).

\[
\hat{z}_k = F_{k,1} \hat{z}_{k-1} + G_{k,1} \begin{bmatrix} u(t) \end{bmatrix}_{k-1} + L_{k-1} (\hat{z}_{k-1} - z_{k-1}), \quad L_{k-1} = \begin{bmatrix} -\tau_{k-1} \\ -1 \end{bmatrix}
\]  
(3.121)

Third, find the estimates of the continuous states using equation (3.113).

\[
\hat{x}(t) = \begin{bmatrix} 1 & (t - t_k) \\ 0 & 1 \end{bmatrix} \hat{z}_k(k) + \begin{bmatrix} \int_{t_k}^{t} (\tau - t_k) u(\tau) d\tau \\ \int_{t_k}^{t} u(\tau) d\tau \end{bmatrix}
\]  
(3.122)

Figure 3.3 illustrates the discrete estimator responses for this example. As noted earlier, the discrete states converged in two steps (for $n$-dimensional system, $n$ steps are required for convergence).
The estimates of the continuous states are depicted in Figure 3.4. As soon as the discrete states converges, the continuous states also converges to the actual states.
3.4 Integration of Control and Fault Diagnosis – Methodology

3.4.1 Integral Sliding Mode Control

The integral sliding mode control (ISM C) method was proposed initially by Zanasi (1993). A systematic design procedure was then generalized by Utkin and Shi (1996), and the design procedure based on Ackermann’s formula for linear systems was established by Ackermann and Utkin (1998). By extending the order of the sliding motion equation to the order of the system rather than reducing it to order of the control, the sliding mode is enforced from the initial time instant, and robustness is enhanced throughout the entire response of the system.

Design Procedure

Consider a nonlinear system represented by the following equation

\[ \dot{x} = f(x) + g(x)u \quad (3.123) \]

where \( x \in \mathbb{R}^n \), \( u \in \mathbb{R}^m \) and suppose there exists an ideal control \( u_0 \) (such as a state feedback \( u_0 = u(x) \)) such that the system can be stabilized in a desired way. With this ideal control input, the system is

\[ \dot{x}_o = f(x) + g(x)u_0 \quad (3.124) \]

where \( x_o \) is the state trajectory under the ideal control \( u_o \). When the above system experiences a disturbance or parameter uncertainty, the system behavior is described by

\[ \dot{x} = f(x) + g(x)u + h(x,t) \quad (3.125) \]

where \( h(x,t) \) denotes the perturbation caused by the disturbance or uncertainty in the system. Assume that the matching condition holds and the perturbation is bounded, i.e.,

\[ h(x,t) \in \text{span}\{g(x)\} \quad \Rightarrow \quad h(x,t) = g(x)\lambda(x,t) \quad (3.126a) \]

\[ \|h_i(x,t)\| \leq k_i(x,t) \quad i = 1, \ldots, n \quad (3.126b) \]
where $k(x,t)$ is a known scalar function. The above conditions (3.126) are quite restrictive but they are mostly satisfied in many applications. Once the conditions (3.126) are satisfied, we can define the disturbance $h(x,t) = g(x)\dot{\lambda}(x,t)$ as shown in equation (3.126).

Now the integral sliding mode control is designed to make the state trajectory $x = x_o$ from the initial time instant. The first step is to divide the control into 2 parts:

$$u = u_o + v$$  \hspace{1cm} (3.127)

where $u_o$ is the ideal control defined in equation (3.124) and $v$ is the discontinuous control to be chosen to reject the perturbation $h(x,t)$. The second step is the design of the sliding manifold. Let the sliding manifold be

$$s = s_o(x) + z \quad s, s_o(x), z \in \mathbb{R}^n$$  \hspace{1cm} (3.128)

where $s_o(x)$ is the manifold that can be designed with the ideal control $u_o$ and $z$ is the part that induces the integral term. Since the discontinuous control $v$ introduced in equation (3.127) is intended for the rejection of a perturbation, its equivalent value $\nu_{eq}$ should be in a form of

$$\nu_{eq} = \dot{\lambda}(x, t) \quad \Leftrightarrow \quad h(x, t) = g(x)\nu_{eq}$$  \hspace{1cm} (3.129)

The integral term $z$ that satisfying the equation (3.129) can be found by differentiating equation (3.128) with time, let the control be ideal, and solve for $\dot{z}$ by letting $\dot{s} = 0$ as

$$\dot{s} = \dot{s}_o(x) + \dot{z}$$
$$= \frac{\partial s}{\partial x} \dot{x}_o + \dot{z}$$
$$= \frac{\partial s}{\partial x} [f(x) + g(x)u_o] + \dot{z}$$  \hspace{1cm} (3.130)

Therefore, the integral term $z$ is selected as

$$\dot{z} = -\frac{\partial s}{\partial x} [f(x) + g(x)u_o], \quad z(0) = -s_o(x(0))$$  \hspace{1cm} (3.131)

Now, the new sliding manifold with control $u_o + v$ becomes
\[ \dot{s} = \dot{s}_s(x) + \dot{\lambda} \]
\[ = \frac{\partial s_s}{\partial x} [f(x) + g(x)(u_s + v) + h(x, t)] - \frac{\partial s_s}{\partial x} [f(x) + g(x)u_s] \]
\[ = \frac{\partial s_s}{\partial x} [g(x)v + h(x, t)] \]
\[ = \frac{\partial s_s}{\partial x} [g(x)v + g(x)\lambda(x, t)] \]
\[ = \frac{\partial s_s}{\partial x} g(x)[v + \lambda(x, t)] \]  

(3.132)

By the definition of equivalent control (let \( \dot{s} = 0 \) and solve for control \( v \)),

\[ v_{eq} = \lambda(x, t) \iff h(x, t) = g(x)v_{eq} \]  

(3.133)

And the sliding motion equation is

\[ \dot{x} = f(x) + g(x)(u_s + v_{eq}) + h(x, t) \]
\[ = f(x) + g(x)u_s \]  

(3.134)

As shown in equation (3.134), the order of the sliding motion equation is the same as the order of the system and the state trajectory is under control of the ideal control \( u_s \). To enforce the sliding mode, the discontinuous control \( v \) should be designed such that \( \dot{s} < 0 \). To this end, consider the equation (3.132) with the discontinuous control \( v = -M\text{sign}(s) \) and \( h(x, t) = g(x)\lambda(x, t) \),

\[ \dot{s} = -\frac{\partial s}{\partial x} g(x)[M\text{sign}(s) + \lambda(x, t)] \]  

(3.135)

If the following conditions are satisfied, then we have achieved the reaching condition \( \dot{s} < 0 \).

\[ \frac{\partial s}{\partial x} g(x) \text{ be positive definite.} \]  

(3.136a)

\[ M > |\lambda(x, t)| \]  

(3.136b)

To relax the condition in equation (3.136), Utkin and Shi (1996) proposed the so-called decoupling control method that uses a transformation of the sliding manifold.
3.4.2 Use of Integral Sliding Mode Control for Fault Detection and Compensation

A main purpose of using the ISMC is to compensate perturbations introduced in a system. When the perturbation is an input (actuator) fault, it can be compensated by the integral sliding mode control. When the perturbation is an output (sensor) fault, the integral sliding mode control drives the system out of a desired trajectory. In contrast to input fault case, an additional consideration is required to compensate the output fault.

An observation can be made from the equation (3.133); the discontinuous control in the ISMC method is directly related to perturbations. Using sliding mode terminology, we can state that a perturbation can be estimated using the equivalent value of the discontinuous control. Especially when the perturbation is either an actuator fault or a sensor fault, the fault can be detected by monitoring the equivalent value of the integral sliding mode control term. Therefore, detection of a fault may be accomplished even without using any observer.

As explained above, there are two advantages in using the ISMC as control method. It can be designed to compensate an input fault and it can be utilized to detect both a sensor and an actuator fault. To illustrate the idea, both an actuator fault and a sensor fault are considered below.

*Actuator fault detection and compensation*

The system given in equation (3.123) with actuator fault is represented as

\[
\dot{x} = f(x) + g(x)(u + \Delta u)
\]  

(3.137)

where \(\Delta u\) represents an actuator fault and is bounded. Use the same approach considered above and divide control into 2 parts \(u = u_o + v\) where \(u_o\) is selected such that it stabilizes the system or forces the state trajectory follows a desired trajectory (usually \(u_o = u(x)\), i.e. state feedback control). Consider the case with output \(y\) to be zero where the desired manifold for control \(u_o\) becomes as follows.

\[
S_o = Cx
\]  

(3.138)

Now, select an additional state using
\[
\dot{z} = -\frac{\partial S_s}{\partial x} \left( f(x) + g(x)u_s \right) \quad z(0) = -S_s(x(0))
\] (3.139)

Define a new sliding surface as

\[ S = S_s + z \] (3.140)

and the derivative of the sliding manifold is

\[
\dot{S} = \frac{\partial S_s}{\partial x} \dot{x} + \dot{z}
\]

\[ = \frac{\partial S_s}{\partial x} \left[ f(x) + g(x)(u_s + \nu) + g(x)\Delta u \right] - \frac{\partial S_s}{\partial x} \left[ f(x) + g(x)u_s \right] \]

\[ = \frac{\partial S_s}{\partial x} g(x) [\nu + \Delta u] \] (3.141)

With appropriate choice of the gain, a sliding mode can be enforced. As long as sliding occurs, the equivalent value of \( \nu \) becomes

\[ \nu_{eq} = \Delta u \] (3.142)

As a result, an actuator fault is compensated using integral sliding mode control, while it can be detected by monitoring the equivalent value of the integral sliding mode control.

Sensor fault detection and compensation

The system given in equation (3.123) with a sensor fault is represented as

\[ \dot{x} = f(x) + g(x)u \] (3.143)

\[ y = Cx + \Delta y \] (3.144)

where \( \Delta y \) represents the sensor fault and is bounded. Again use the same approach considered above with the control \( u = u_s + \nu \). And, the desired manifold for control \( u_s \) becomes

\[ S_s' = Cx \] (3.145)

Now, choose an additional state using

\[ \dot{z} = -\frac{\partial S_s}{\partial x} \left( f(x) + g(x)u_s \right) \quad z(0) = -S_s(x(0)) \] (3.146)

Define a new sliding surface as
\[ S = S_s + z \]  

and the derivative of the sliding manifold

\[ \dot{S} = \frac{\partial S_s}{\partial x} \dot{y} + \dot{z} \]

\[ = \frac{\partial S_s}{\partial x} [f(x) + g(x)(u_s + v) + \Delta y] \frac{\partial S_s}{\partial x} [f(x) + g(x)u_s] \]

\[ = \frac{\partial S_s}{\partial x} [g(x)v + \Delta y] \]  

(3.148)

As in the case with actuator fault, a sliding mode can be enforced with a proper selection of the gain. Using the equivalent value of \( v \), the following relationship can be obtained and we can calculate \( \Delta \dot{y} \).

\[ g(x)v_{eq} + \Delta \dot{y} = 0 \quad \Leftrightarrow \quad \Delta \dot{y} = -g(x)v_{eq} \]  

(3.149)

As in the actuator fault case, the presence of a sensor fault can be detected by monitoring the equivalent value of integral sliding control term. Equation (3.149) requires the additional assumption that the sensor fault should be continuous. Other than intermittent faults, most of the sensor faults such as offset, drift occur gradually, and hence are continuous.

A notable fact is that the system is forced to track (for a tracking problem) the faulty sensor measurement due to the integral sliding mode control term, i.e., the output including a fault \( (C \dot{x} + \Delta y) \) will track a desired trajectory. This implies that true output \( (C \dot{x}) \) deviates from the desired trajectory by the amount of a sensor fault \( (\Delta y) \), i.e., the integral sliding mode control drives the system out of a desired trajectory. Therefore, an additional action is required to compensate the sensor fault. Two approaches can be considered to compensate the sensor fault: i) One approach is the use of decoupling, and ii) another approach is based on cancellation of the fault through direct estimation of the sensor fault.

i) Decoupling of a faulty sensor

The decoupling approach is application in the case when a system is still observable without the measurement from a faulty sensor. As long as a system remains observable without the measurement from a faulty sensor, the system output can be estimated from an observer that does not use the faulty sensor. When a faulty sensor is decoupled from an observer, the estimated
output is free of the sensor fault. And the output from the faulty sensor is replaced by the estimated output in feedback. As a result, the controller does not see the faulty sensor and the system remains in a desired trajectory. The fault isolation schemes (Generalized Observer Scheme, Dedicated Observer Scheme, NPERG scheme) explained in Chapter 2 can be applied for this objective.

ii) Direct estimation of sensor fault

The above decoupling approach cannot be used when a system becomes unobservable by removing a faulty sensor in the system outputs. In this case, the sensor fault ($\Delta y$) should be estimated to obtain the estimate of true output ($\hat{C}x = y - \Delta y$). Then, the estimated output can be fed back to the controller so that the system follows a desired trajectory. Estimation of a sensor fault is not a trivial task, and the existence of a suitable algorithm depends on the characteristics of the system under consideration. However, the time derivative of a sensor fault can be estimated through the equivalent value of the sliding mode control term as shown in equation (3.164). This additional information can be utilized further to estimate the sensor fault as will be illustrated in the next section.

3.4.3 Fault Isolation Using Observers with Hypothesis Testing

As explained in the previous section, the presence of a fault can be detected by monitoring the equivalent value of an integral sliding mode control (ISMC) term, i.e., when there is either an actuator fault or a sensor fault, the equivalent value of ISMC term will deviate from zero. However, a fault isolation problem still remains, i.e., it is possible for either an actuator or a sensor fault to exist. Applying the NPERG (Nonlinear Parity Equation Residual Generation) scheme is a reasonable approach for fault isolation (See Chapter 2 for details). However, some characteristics of the fault can be estimated using integral sliding mode control. If an actuator fault is introduced in a system, the equivalent value of ISMC term represents the actuator fault itself. Therefore, an input observer is no longer necessary to isolate an
actuator fault, as was the case in the NPERG scheme. Typically, the existence condition for an input observer is quite strict and often requires time derivative of output. In the proposed approach, the use of an input observer can be avoided and a state observer can be utilized for the isolation of the actuator fault. If a sensor fault exists, the time derivative of the sensor fault can be estimated using the equivalent value of ISMC term. For sensor fault isolation, the NPERG scheme without the input observer can still be utilized as long as the system remains observable by eliminating some measurement. If the system becomes unobservable after removing the faulty sensor, the sensor fault should be estimated using the available measurements and the estimated time derivative of the sensor fault. Estimates of the sensor fault, then, can be used for both isolation and compensation. To accomplish the objectives mentioned above, observers with hypothesis testing are proposed as follows.

Before proposing a design for observer with hypothesis testing, let us analyze the behavior of a sliding mode observer for a healthy plant. Consider an affine nonlinear system as follows.

\[ \dot{x} = f(x) + g(x)u \]
\[ y = Cx \] (3.150)

Represent the system with new variable \( x_t \) and \( y \):

\[ \dot{x}_t = f_t(x_t, y) + g_t(x_t, y)u \]
\[ \dot{y} = f_z(x_t, y) + g_z(x_t, y)u \] (3.151)

An observer for the system in equation (3.151) can be designed as:

\[ \dot{x}_t = f_t(x_t, \hat{y}) + g_t(x_t, \hat{y})u + L_1(y - \hat{y}) + M_1 \text{sign}(y - \hat{y}) \]
\[ \dot{\hat{y}} = f_z(x_t, \hat{y}) + g_z(x_t, \hat{y})u + L_2(y - \hat{y}) + M_2 \text{sign}(y - \hat{y}) \] (3.152)

where \( L_1, L_2, M_1 \) and \( M_2 \) represent the observer gain and are selected to provide the observer with convergence. Define the estimation error as \( e_t = x_t - \hat{x}_t, e_y = y - \hat{y} \). The error dynamics become:

\[ \dot{e}_t = f_t(e_t, e_y) + g_t(e_t, e_y)u + L_1 e_y + M_1 \text{sign}(e_y) \]
\[ \dot{e}_y = f_z(e_t, e_y) + g_z(e_t, e_y)u + L_2 e_y + M_2 \text{sign}(e_y) \] (3.153)

The sliding manifold \( s = e_y \) in this case, and the time derivative of the sliding manifold is:
\[
\dot{s} = \dot{e}_r + f_2(e_1, e_r) + g_2(e_1, e_r)u + L_1 e_r + M_2 \text{sign}(e_r) .
\] (3.154)

With a proper choice of gain $L_1$ and $M_2$, a sliding mode will occur, and the equivalent value of discontinuous function $\text{sign}(e_r)$ becomes

\[
\text{sign}(e_r)_{eq} = -M_2^{-1}(f_2(e_1, 0) + g_2(e_1, 0)u).
\] (3.155)

By substituting the equation (3.155) into the first equation of the equation (3.153), we have the following sliding mode equation:

\[
\dot{e}_1 = f_1(e_1, 0) + g_1(e_1, 0)u - M_1 M_2^{-1}(f_2(e_1, 0) + g_2(e_1, 0)u)
\] (3.156)

where again the gain $M_1$ is selected to provide a stability of the above sliding mode equation (3.156). The selection of the gains mentioned above is non-trivial and depends on nonlinearities ($f_1, f_2, g_1$ and $g_2$) in the system.

Next, let us examine the behavior of the observer in the presence of either actuator or sensor fault, and propose a design approach for the observer with hypothesis testing for fault isolation.

**Hypothesis A: Estimated fault represents an actuator fault.**

Design an observer using measured outputs, commanded inputs, and the equivalent value of ISMC term as an estimated actuator fault. The schematic diagram of such an observer is depicted in Figure 3.5.

![Figure 3.5: Schematic of observer with hypothesis testing for actuator fault isolation](image)

If an actuator fault is present, then the observer will converge and $\hat{y}$ will approach $y$. On the other hand, if a sensor fault is present, the observer will not converge. Accordingly, a residual $r_a = y - \hat{y}$
with a sensor fault. Therefore, an actuator fault and a sensor fault can be isolated by monitoring the residual $r_A$.

Consider the system given in equation (3.151) to illustrate this idea. The system given in equation (3.151) with an actuator fault becomes:

$$
\dot{x}_i = f_1(x_i, y) + g_1(x_i, y)(u + \Delta u)
$$
$$
\dot{y} = f_2(x_i, y) + g_2(x_i, y)(u + \Delta u)
$$

(3.157)

Design an observer for the actuator fault as follows.

$$
\dot{\hat{x}_i} = f_1(\hat{x}_i, \hat{y}) + g_1(\hat{x}_i, \hat{y})(u + v_{eq}) + L_1(y - \hat{y}) + M_1 \text{sign}(y - \hat{y})
$$
$$
\dot{\hat{y}} = f_2(\hat{x}_i, \hat{y}) + g_2(\hat{x}_i, \hat{y})(u + v_{eq}) + L_2(y - \hat{y}) + M_2 \text{sign}(y - \hat{y})
$$

(3.158)

If an actuator exists ($\Delta u \neq 0$), then the estimation error dynamics becomes

$$
\dot{e}_i = f_1(e_i, e_y) + g_1(e_i, e_y)(u + \Delta u - v_{eq}) + L_1 e_y + M_1 \text{sign}(e_y)
$$
$$
\dot{e}_y = f_2(e_i, e_y) + g_2(e_i, e_y)(u + \Delta u - v_{eq}) + L_2 e_y + M_2 \text{sign}(e_y)
$$

(3.159)

Since $v_{eq} = \Delta u \approx \Delta u$, the error dynamics in equation (3.159) become the same as the one without actuator fault as:

$$
\dot{e}_i = f_1(e_i, e_y) + g_1(e_i, e_y)u + L_1 e_y + M_1 \text{sign}(e_y)
$$
$$
\dot{e}_y = f_2(e_i, e_y) + g_2(e_i, e_y)u + L_2 e_y + M_2 \text{sign}(e_y)
$$

(3.160)

and the observer will converge as designed. Furthermore, the equivalent value of discontinuous function $\text{sign}(e_y)$ will approach zero after $e_i$ approaches zero.

$$
\psi_{eq} = \text{sign}(e_y)_{eq} = -M_2^{-1}(f_2(e_i, 0) + g_2(e_i, 0)u) \rightarrow 0 \quad \text{as} \quad e_i \rightarrow 0
$$

(3.161)

If the detected fault is a sensor fault, the estimation error dynamics become

$$
\dot{e}_i = f_1(e_i, e_y) + g_1(e_i, e_y)(u - v_{eq}) + L_1 e_y + M_1 \text{sign}(e_y)
$$
$$
\dot{e}_y = f_2(e_i, e_y) + g_2(e_i, e_y)(u - v_{eq}) + L_2 e_y + M_2 \text{sign}(e_y)
$$

(3.162)

and the observer convergence is not guaranteed due to the term $v_{eq}$. Also, the equivalent value of discontinuous function $\text{sign}(e_y)$ will deviate from zero. Therefore, a distinction between an actuator fault and a sensor fault can be made by monitoring $\psi_{eq}$ from the observer.
Hypothesis B: Estimated fault represents a sensor fault.

When the decoupling approach can be applied as explained in the previous section, isolation of a sensor fault is rather straight-forward and estimation of the actual output is possible. All fault isolation schemes using multiple observers (introduced in Chapter 2) are suitable for this case. Without losing generality, the decoupling approach is not explained in detail here. When a system becomes unobservable by eliminating a sensor from output equation, the specific sensor fault should be estimated for both isolation and compensation. Design of such an observer is proposed and its existence conditions are examined in this subsection.

Let us consider a special case where the system only has one output with fault.

\[
\begin{align*}
\dot{x}_1 &= f_1(x_1, x_2) + g_1(x_1, x_2)u \\
\dot{x}_2 &= f_2(x_1, x_2) + g_2(x_1, x_2)u \\
y &= x_2 + \Delta y
\end{align*}
\]  

(3.163)

where \( x_f \) is a vector of appropriate dimension and \( x_2 \in \mathbb{R}^1 \). The following observer structure is proposed for the above system:

\[
\begin{align*}
\dot{\hat{x}}_1 &= f_1(\hat{x}_1, \hat{y}) + g_1(\hat{x}_1, \hat{y})u \\
\dot{\hat{y}} &= f_2(\hat{x}_1, \hat{y}) + g_2(\hat{x}_1, \hat{y})u + L_\Delta (y - \hat{y}) + M_\Delta \text{sign}(y - \hat{y})
\end{align*}
\]  

(3.164)

If a sensor fault is exists (\( \Delta y \neq 0 \)), then the error dynamics becomes

\[
\begin{align*}
\dot{e}_1 &= f_1(e_1, e_r) + g_1(e_1, e_r)u \\
\dot{e}_r &= \hat{y} - \hat{y} \\
&= C\hat{x} + \Delta \hat{y} - \hat{y} \\
&= f_1(x_1, x_2) + g_2(x_1, x_2)u + \Delta \hat{y} - f_1(\hat{x}_1, \hat{y}) - g_2(\hat{x}_1, \hat{y})u - L_\Delta e_r - M_\Delta \text{sign}(e_r) \\
&= f_1(x_1, y - \Delta y) + g_2(x_1, y - \Delta y)u + \Delta \hat{y} - f_1(\hat{x}_1, \hat{y}) - g_2(\hat{x}_1, \hat{y})u - L_\Delta e_r - M_\Delta \text{sign}(e_r)
\end{align*}
\]  

(3.165)

Define new functions as follows.

\[
\begin{align*}
\tilde{f}_2(e_1, e_r, \Delta y) &= f_2(x_1, y - \Delta y) - f_2(x_1, \hat{y}) \\
\tilde{g}_2(e_1, e_r, \Delta y) &= g_2(x_1, y - \Delta y) - g_2(x_1, \hat{y})
\end{align*}
\]  

(3.166)

Then, the equation (3.165) become:
\begin{align}
\dot{e}_1 &= f_1(e_1, e_1) + g_1(e_1, e_1)u \\
\dot{e}_2 &= \tilde{f}_2(e_2, e_2, \Delta y) + \tilde{g}_2(e_2, e_2, \Delta y)u + \Delta \hat{y} - L_2e_2 - M_2\text{sign}(e_2).
\end{align}
\tag{3.167}

With appropriate choice of gain \(L_2\) and \(M_2\), a sliding mode can be enforced on the sliding manifold \(s = e_2\).

And the equivalent value of discontinuous function \(\text{sign}(e_2)\) can be obtained as follows.

\[\psi_{e_2} = \text{sign}(e_2)u = M_2^{-1}(\tilde{f}_2(e_2, 0, \Delta y) + \tilde{g}_2(e_2, 0, \Delta y)u + \Delta \hat{y}).\]
\tag{3.168}

As long as the error dynamics for \(e_1\) is stable, \(e_1\) will approach to zero in finite time, i.e., if \(\dot{e}_1 = f_1(e_1, 0) + g_1(e_1, 0)u\) is stable, then \(e_1 \to 0\). Then the equation (3.168) will have the following form.

\[\psi_{e_2} = M_2^{-1}(\tilde{f}_2(0, 0, \Delta y) + \tilde{g}_2(0, 0, \Delta y)u + \Delta \hat{y}).\]
\tag{3.169}

Note that the equation (3.169) is a nonlinear algebraic equation where \(\psi_{e_2}, M_2, u\) are known and \(\Delta y, \Delta \hat{y}\) are unknown. The estimates of \(\Delta y\) can be obtained by substituting \(\Delta \hat{y} = -g(x)v_{eq}\) into \(\Delta \hat{y}\) in equation (3.169) and solving for \(\Delta y\). Now, for sensor fault compensation, the sensor output \((y + \Delta y)\) is modified using the estimated output fault \((\Delta \hat{y})\) and the modified output \((y + \Delta y - \Delta \hat{y} = y)\) is fed to the controller.

As shown above, for a single output system, a sensor fault can be estimated as long as the unobservable part is stable. In this case, fault isolation can be achieved as follows. As long as the fault introduced to system is a sensor fault, the observer will converge to the measured output. Therefore, a residual from the above observer can be defined as \(r_B = y - \hat{y}\). A decision logic for the system above can be selected using both residual \(r_A\) and \(r_B\) as in Table 3.2.

<table>
<thead>
<tr>
<th></th>
<th>Residual (r_A)</th>
<th>Residual (r_B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Fault</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Actuator Fault</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Sensor Fault</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 3.2 Decision logic with residual \(r_A\) and \(r_B\)

For multiple outputs system where the decoupling approach cannot be used, either the Dedicated Observer Scheme or the Generalized Observer Scheme can still be applied as long as the unobservable part
remains stable. Otherwise, the above mentioned observer can be designed for each sensor resulting in a bank of observers.

3.5 Conclusion

In this chapter, we have addressed some issues related to fault tolerant control. In particular, the relation between closed-loop system and fault has been explored and a few observations have been made. Based on these observations, a framework for integration of the control and fault diagnostic has been proposed using the Integral Sliding Model Control (ISMC) and a bank of observers with hypotheses. The ISMC is utilized to detect a fault (both actuator and sensor fault) in a system. The ISMC is also utilized to compensate an actuator fault. In the case of a sensor fault, the ISMC control forces the system to deviate from a desired trajectory. Therefore, an estimation of the true output (not affected by sensor fault) is required to feed the true output to controller. To this end, multiple observers can be used to isolate and to estimate the true output. For the case of a single output system, estimation of the sensor fault is accomplished using the equivalent values from both controller and observer.

The sliding mode methodology has also been explored for various kinds of estimation and identification problems to illustrate the theoretical background before applying these methods to the automotive engine problem.

One original contribution of this study includes an observer design method with a binary sensor. Even though the approach was initially proposed by Drakunov and Utkin(1995), it was yet incomplete. By adding a transformation so that the discrete observer becomes a dead-beat structure, the application of the observer has been further extended. Also, design of a sliding mode input observer has been introduced in this chapter, showing that differentiation of output is not necessary for this type of input observer.

In the following Chapter, the estimation and identification methods studied in this chapter are applied to the automotive engine problem and are validated using experimental data. Also, the proposed framework for the integration of control and diagnostics is validated with a simulation model of the engine.
CHAPTER 4

Application to IC Engine Air and Fuel Management System

Various algorithms introduced in Chapter 3 are applied to the air and fuel management system of an internal combustion (IC) engine. At the beginning of the chapter, an IC engine model, the so-called mean value engine model, is described. Some parameters of the engine model are then identified using the experimental data. And based on the model identified, the estimation problems are explored by utilizing the estimation methods mentioned in Chapter 3. After each estimation problem is solved, some of the estimator is configured based on the NPERG scheme for the fault diagnostic application. The performance of the above model based FDI approach is then illustrated with the experimental data. As a model-based control application, two estimation approaches (the on-line parameter estimation and the estimator using a binary sensor) are adopted to implement a control law for air fuel ratio regulation. The simulation model is used to validate this estimator-based control. The algorithm for the integration of control and diagnostics is then implemented in simulation and validated with case studies.
4.1 Internal Combustion Engine Model – Mean Value Engine Model

A spark ignition (SI) internal combustion (IC) engine is a highly complex system that consists of several sub-systems, and can be (partially) described by a combination of nonlinear algebraic and differential equations. For model-based control and diagnostic purposes, the so-called mean-value models, have been proposed in Dobner (1986), Powell and Cook (1987), Hendricks and Sorenson (1989), Moskwa and Hedrick (1989). A mean-value engine model describes the engine dynamics with limited bandwidth, this is equivalent to considering the mean behavior of the state variables over a few engine cycles. Mean-value engine models have also been applied for engine control strategy development and model-based control application (Amstutz et al., 1994; Beaumont et al., 1992; Chang et al., 1995; Cho and Hedrick, 1988; Fekete et al, 1995; Grizzle et al, 1991; Hendricks et al., 1992).

Figure 4.1: Block diagram of a mean-value engine model
The structure of a subset of a typical mean-value engine model is illustrated in Figure 1 in block diagram form. The various sub-systems of the mean-value engine model considered are: filling-and-emptying of intake manifold, fuel film dynamics (wall wetting model), exhaust mixing and transportation delay, and exhaust sensor dynamics. These subsystems are emphasized in the block diagram of Figure 1 (blocks with thick lines). The engine model derivation is based on physical principles, and on some empirical correlations. In our case, the model is formulated in the crank angle domain, i.e.: the independent variable is crank angle instead of time. The reason for this transformation is that many important engine events, such as fuel injection, occur quasi-periodically with respect to crank angle (Chang et al., 1995; Chin and Coats, 1986). The development of the each of the model blocks in Figure 4.1 is explained in greater detail in (Hendricks and Sorenson, 1991; Moskwa and Hedrick, 1989; Powell and Cook, 1987); we give a brief summary below. All relevant terms are defined in the Nomenclature.

Figure 4.2: Air-fuel related subsystem of an IC engine
4.1.1 Model Description

In this section, the engine model depicted in Figure 4.2 is described in detail. Much focus is placed on the air fuel related subsystem as shown in Figure 4.2. In addition, a drive train consists of automatic transmission, gear-shift logic and vehicle longitudinal dynamics is also described. The model explained here is implemented in Simulink™ and Simulink model (PTSIM) description is in Appendix B.

**Throttle Area**

The throttle area with a given throttle angle \( \alpha \) can be represented by a nonlinear equation given below.

\[
A_a = \frac{\pi D^2}{4} \left( 1 - \frac{\cos \alpha}{\cos \alpha_o} \right) + \frac{2}{\pi} \frac{a}{\cos \alpha} \left( \cos^2 \alpha - a^2 \cos^2 \alpha_o \right)^{1/2} - \\
\cos \alpha \sin^{-1} \left( \frac{a \cos \alpha_o}{\cos \alpha} \right) - a \left( 1 - a^2 \right)^{1/2} + \sin^{-1} a
\]  

(4.1)

**Mass flow rate of air at throttle**

The mass flow rate of air at throttle can be described as flow depends on the flow condition.

For unchoked flow:

\[
\dot{m}_{un} = C_{D,a} A_a \rho \left( \frac{p_a}{p_s} \right)^{1/\gamma} \left[ 2 \gamma - 1 \left( \frac{p_a}{p_s} \right)^{\gamma/(\gamma+1)} \right]^{1/2}, \quad \frac{p_a}{p_s} > \left[ \frac{2}{\gamma + 1} \right]^{\gamma/(\gamma-1)}
\]  

(4.2)

for choked flow:

\[
\dot{m}_{ch} = C_{D,a} A_a \rho \left( \frac{2}{\gamma + 1} \right)^{1/\gamma} \left( \frac{p_a}{p_s} \right)^{\gamma/(\gamma-1)}, \quad \frac{p_a}{p_s} \leq \left[ \frac{2}{\gamma + 1} \right]^{\gamma/(\gamma-1)}
\]  

(4.3)
Intake Manifold

Applying the principle of conservation of mass in the intake manifold, we obtain the following equation:

\[ \frac{dm_{in}}{dt} = \dot{m}_{ah} - \dot{m}_{sc} \]  \hspace{1cm} (4.4)

Using the Ideal Gas law, (4.4) can be reformulated as follows:

\[ \frac{dp_n}{dt} + \frac{\eta V_e \omega}{4 \pi V_n} p_n = \frac{RT_n}{V_n} \dot{m}_{ah} \]  \hspace{1cm} (4.5)

Equation (4.5) can be transformed into the crank angle domain using the chain rule of differentiation, and employing the crank angle variable, \( \theta \), as the independent variable:

\[ \omega \frac{dp_n}{d\theta} + \frac{\eta V_e \omega}{4 \pi V_n} p_n = \frac{RT_n}{V_n} \dot{m}_{ah} \]  \hspace{1cm} (4.6)

In the presence of the exhaust gas recirculation (EGR), one must take into account the presence of a diluent mass (Moskwa and Hedrick, 1989). The additional mass results in a modified version of (4.6):

\[ \frac{dp_n}{d\theta} = -\frac{\eta V_e \omega}{4 \pi V_n} p_n + \frac{RT_n}{V_n \omega} (\dot{m}_{ah} + \dot{m}_{EGR}) \]  \hspace{1cm} (4.7)

The mass flow rate evaluation in the EGR duct is obtained using the compressible flow equation:

\[ \dot{m}_{EGR} = A_n \sqrt{\frac{p_{in}}{RT_{in}}} \left[ \left( \frac{P_{in}}{p_{in}} \right)^{\frac{2n}{n-1}} - \left( \frac{P_{in}}{P_{sw}} \right)^{\frac{n+1}{n-1}} \right] \]  \hspace{1cm} (4.8)

where the \( n \) exponent replaces the \( k=c_p/c_v \) value because of the heat transfer through the duct. Under sonic flow conditions the pressure ratio is substituted by the critical value:

\[ \frac{p_{in}}{p_{sw}} = \left( \frac{2}{n+1} \right)^{\frac{n}{n-1}} \]  \hspace{1cm} (4.9)

Actual mass flow rate of air into cylinder can be computed using a cylinder charge efficiency, resulting in the so-called speed density equation:

\[ \dot{m}_w = \frac{\eta PV_e \omega}{4 \pi RT_n} - \dot{m}_{EGR} \]  \hspace{1cm} (4.10)
Fuel Dynamic (Wall Wetting Model)

A commonly used model of the condensation and evaporation dynamics of fuel in an intake port (Acquino, 1981; Chang et al., 1995; Shayler et al., 1995) is shown in Figure 4.3

![Fuel Dynamic Diagram](image)

Figure 4.3: Fuel dynamic (Wall wetting model)

This model is based on the notion that a fraction of the fuel injected into the inlet form will condense to form a liquid fuel film on the inlet port walls. The mass of liquid fuel in the fuel film will then change 1) due to the addition of fuel from the injection process and 2) due to evaporation of the condensed fuel according to:

\[
\frac{dm_g}{dt} = -\frac{1}{\tau_f} m_g + X \dot{m}_g
\]

\[
\dot{m}_g = \frac{1}{\tau_f} m_g + (1 - X) \dot{m}_g
\]  

(4.11)

Again, equation (4.11) can be transform into the crank angle domain:

\[
\frac{dm_g}{d\theta} = -\frac{1}{\tau_f \omega} m_g + \frac{X}{\omega} \dot{m}_g
\]

\[
\dot{m}_g = \frac{1}{\tau_f} m_g + (1 - X) \dot{m}_g
\]  

(4.12)

118
Air/Fuel Ratio in Cylinder

Using the mass flow rate of air and fuel from equation (4.10) and (4.12), the fuel/air ratio in the cylinder can be expressed as the ratio of in-cylinder fuel mass to air mass:

\[
\frac{\dot{m}_f}{\dot{m}_{ac}} = \frac{\dot{m}_f}{\tau_f \dot{m}_{ac}} + (1 - X) \frac{\dot{m}_g}{\dot{m}_{ac}}
\]  

(4.13)

Let the air-fuel ratio in cylinder be \( AFR = \frac{\dot{m}_g}{\dot{m}_f} \), then (4.13) becomes

\[
\frac{1}{AFR} = \frac{1}{\tau_f} \frac{\dot{m}_f}{\dot{m}_{ac}} + (1 - X) \frac{\dot{m}_g}{\dot{m}_{ac}}
\]

(4.14)

Introducing the equivalence ratio in cylinder \( \phi_c = I/AFR \),

\[
\phi_c = \frac{1}{\tau_f} \frac{\dot{m}_f}{\dot{m}_{ac}} + (1 - X) \frac{\dot{m}_g}{\dot{m}_{ac}}
\]

(4.15)

Delay and Sensor dynamics

An important aspect of the dynamics of a spark-ignition engines is the inherent delay of the combustion process and of the transport of the exhaust gas between the exhaust valve and the oxygen sensor (Powell and Cook, 1987; Chang et al, 1995). The total delay is the sum of combustion delay and transport delay. In addition, the sensor used to measure the concentration of oxygen in the exhaust gas can – to a first approximation – be modeled as a first order lag with time constant \( \tau_m \). These concepts are summarized in (4.16).

\[
\tau_m \frac{d\phi}{dt} + \phi = \phi(\theta - t_e - t_c)
\]

(4.16)

In the crank angle domain, the equation (4.16) can be expressed in radians:

\[
\frac{d\phi}{d\theta} = -\frac{1}{\tau_m \omega} \phi + \frac{1}{\tau_m \omega} \phi(\theta - \phi - \theta_c)
\]

(4.17)
Torque Production (Combustion torque model)

Engine torque is generated through the combustion of air fuel mixture in a cylinder. The variables, which similarly influence the combustion process, are the ignition of a cylinder charge of air, fuel and residual gas as well as other variables and parameters which also affect the combustion efficiency, i.e., in-cylinder flow motion, engine temperature and so on. Modeling of the combustion process is not a trivial task at all and requires an object-oriented approach since the complexity of model basically limit the capability of the model in application (Dawson, 1998). For example, for analyses such as combustion efficiency, cylinder design and breathing process, a detail combustion model, which takes account of the effect of engine parameters and variables at every crank angle, is required. In this study, an approach originally developed by Powell and Cook (1987) is adopted since this model provides reasonable accuracy for torque production in the mean value engine model based control analysis. This model approximates the engine torque as a nonlinear algebraic equation as a function of mass flow rate of air into cylinder, air fuel ratio, spark advance, engine speed and EGR flow rate as shown in equation (4.18). This model can be obtained through the regression analysis of dynamometer data.

\[
T_e = f\left(\dot{m}_{\text{ac}}, AFR, \delta, \omega, \dot{m}_{\text{EGR}}\right) = 32.748 + 15.1744\dot{m}_{\text{ac}} + 0.8764AFR - 0.034AFR^2 + 1.04\delta - 0.0112\delta^2
\]
\[
+ 0.0011\omega - 4.28 \times 10^{-6}\omega^3 + 1.92 \times 10^{-5}\omega\delta + 0.102\delta m_{\text{ac}} - 0.002\delta m_{\text{ac}} - 1183\delta \dot{m}_{\text{EGR}}
\]

(4.18)

Exhaust Emission

Another combustion by-product is the exhaust emission which is the result of chemical reaction (combustion) of the mixture of fuel vapor (hydrocarbon) and air within the cylinder. Under ideal condition (stoichiometric air fuel mixture and complete combustion), the combustion equation of a general hydrocarbon fuel \((C_aH_b)\) with air is:

\[
C_aH_b + (a + b/4)(O_2 + 3.773N_2) \rightarrow aCO_2 + b/2H_2O + 3.773(a + b/4)N_2
\]

(4.19)

As shown in equation (4.19), the combustion products consist of \(CO_2\), \(H_2O\), \(N_2\) for a stoichiometric mixture. More complicated combustion equation which takes into account of incomplete combustion and non-stoichiometric mixture is:
The purpose of the emission model in this study is to predict emission as a function of crank angle (or time) during the actual simulations under different air fuel control strategies or faulty operating condition. To serve this purpose, rather than include all the effects of combustion related parameters, a few assumptions are made to simplify the exhaust emission modeling. The approach applied here has initially proposed by Yuen and Servati (1984). First, the chemical composition of fuel is considered only for \( C_8H_{18} \). Even though all gasolines are known to have different hydrocarbon composition, most of the chemical properties can be approximated by \( C_8H_{18} \). Second, complete combustion is assumed so that there is no misfire or incomplete combustion. Also, air fuel ratio is the only parameter that affects exhaust emissions. Third, combustion in cylinder consists of one gram of air and \( 1/AFR \) grams of fuel. With these assumptions, the combustion equation can be simplified as:

\[
\frac{1}{114AFR} C_8H_{18} + \frac{1}{138}(O_2 + 3.77N_2) \rightarrow a_iCO + a_iCO + a_iNO + a_iO_2 + a_iN_2 + a_iH_2O + a_iCH_i
\]  

(4.21)

Yuen and Servati (1984) determined the seven constants appearing in equation (4.21) using atomic balance and dynamometer data with different air fuel ratios. Then they defined mass generation ratio \( f_i \) by

\[
f_i = a_iA_i \quad i = NO_x, HC, CO_2, CO, O_2
\]  

(4.22)

where \( A_i \) is the molecular weight of species \( i \). Parameter \( f_i \) can be interpreted as mass generation of species \( i \) per gram of air used in combustion. Finally, the exhaust emission is obtained by

\[
m_i = m_{at} \times f_i \quad i = NO_x, HC, CO_2, CO, O_2
\]  

(4.23)

Variation of \( f_i \) for each species for different air fuel ratio is depicted in Figure 4.4.
In this study, rather than repeating the same procedure given above with the dynamometer data, the mass generation parameter $f_i$ is modeled with polynomial function of air fuel ratio as shown in the equation (4.24).

\[
f_{\text{soy}} = 1.0762 \times 10^{-4} AFR^4 - 7.6086 \times 10^{-7} AFR^5 + 2.0784 \times 10^{-3} AFR^3 - 2.7399 \times 10^{-2} AFR^2 \\
+ 1.7477 \times 10^{-5} AFR - 4.3506 \times 10^{-5}
\]

\[
f_{\text{hc}} = 9.8468 \times 10^{-7} AFR^4 - 7.8574 \times 10^{-7} AFR^5 + 2.3061 \times 10^{-3} AFR^3 - 2.9647 \times 10^{-2} AFR - 1.4190 \times 10^{-1}
\]

\[
f_{\text{co}} = 3.1871 \times 10^{-3} AFR^4 - 1.7442 \times 10^{-3} AFR^3 + 3.1292 \times 10^{-2} AFR^2 - 1.8540 \times 10^{-1} AFR + 1.9685 \times 10^{-1}
\] (4.24)

\[
f_{\text{co2}} = 1.6158 \times 10^{-3} AFR^4 - 1.2918 \times 10^{-3} AFR^3 + 4.042 \times 10^{-2} AFR^2 - 6.1497 \times 10^{-2} AFR + 4.5028 AFR - 12.457
\]

\[
f_{\text{o2}} = -1.8399 \times 10^{-3} AFR^4 + 1.0623 \times 10^{-3} AFR^3 - 2.1646 \times 10^{-2} AFR^2 + 1.8755 \times 10^{-1} AFR - 5.8855 \times 10^{-1}
\]

Figure 4.5 illustrates the regression fit result with the data given in Figure 4.4.
Figure 4.5: Regression fit for mass generation parameter \( f_i \) as a function of air fuel ratio

Then, the flow rate of the exhaust emission is calculated by:

\[
\dot{m}_x = \dot{m}_\nu \times f_x \quad \quad X = NO_x, HC, CO_2, CO, O_2
\]

\[\text{(4.25)}\]

\textit{Crankshaft Dynamics}

The simplest model to describe the rotating dynamics of the crankshaft is a single degree of freedom model, in which it is assumed that the rotating assembly only experience rigid body motion. A single degree of freedom model is depicted in Figure 4.6 where all the torque acting on crankshaft is illustrated.

Figure 4.6: A single degree of freedom crankshaft model
Applying Newton's second law yields:

\[ J \frac{d\omega}{dt} = T_e - T_f - T_a - T_p \]  

(4.26)

where \( T_e \) is the engine torque produced through combustion, \( T_f \) is the friction and pumping loss which also include loss due to damping, \( T_a \) is the torque required to drive the engine accessories such as air condition compressor and power steering pump, \( T_p \) is the torque required to drive torque converter pump which corresponds to load torque. The friction torque, \( T_f \), is difficult to model precisely since it depends on the viscosity of oil, temperature, engine speed and manifold pressure, etc. It is shown that it largely depends on the engine speed and manifold pressure as reported by Ciuuly (1994). Here it is simplified to a polynomial function of engine speed and manifold pressure as:

\[ T_f = 48.814 + 9.639 \times 10^{-3} \omega + 1.6151 \times 10^{-4} \omega^2 - 3.8638 \times 10^{-1} p_m + 7.7083 \times 10^{-4} \omega^1 p_m \]  

(4.27)

In crank angle domain, equation (4.27) can be described by:

\[ \frac{d\omega}{d\theta} = \frac{1}{J\omega} \left( T_e - T_f - T_a - T_p \right) \]  

(4.28)

**Torque Converter and Transmission**

The most commonly used model of a torque converter is mainly divided into a static model and a dynamic model. The dynamic modeling of torque converter itself has been an active research area (Kotwicki, 1982; Runde, 1986; Hrovat and Tobler, 1985). Since the dynamic model is not available, the static model of Kotwicki (1982) is employed for a torque converter. It is reported that the static model still provides reasonable accuracy over a fairly wide range of operating conditions (Kotwicki, 1982; Hrovat and Tobler, 1985). The static model is a quadratic regression fit of data and requires pump speed and turbine speed as inputs. The outputs of the model are the turbine torque and the pump torque and they have different regression fit model depend on the speed ratio \( \omega_t/\omega_p \).
• Coupling mode (\(\omega_i / \omega_k \geq 0.9\))

\[
T_p = T_i = -6.7644 \times 10^{-3} \omega_i^3 - 32.0084 \times 10^{-3} \omega_r \omega_i - 25.2441 \times 10^{-3} \omega_r^2
\]  \hspace{1cm} (4.29)

• Converter mode (\(\omega_i / \omega_k < 0.9\))

\[
T_p = 3.4325 \times 10^{-3} \omega_i^3 + 2.2210 \times 10^{-3} \omega_r \omega_i - 4.6041 \times 10^{-3} \omega_r^2
\]  \hspace{1cm} (4.30a)

\[
T_i = 5.7656 \times 10^{-3} \omega_r^3 + 0.3107 \times 10^{-3} \omega_r \omega_i - 5.4323 \times 10^{-3} \omega_r^2
\]  \hspace{1cm} (4.30b)

**Vehicle Longitudinal Dynamics**

Simple but frequently used vehicle longitudinal dynamics is chosen where only vehicle mass in considered and ignored the wheel slip (Wong, 1979, Cho and Hedrick, 1988). Unless precise wheel slip is required such as anti-lock brake system or traction control system analysis, the model given below meets with purpose of this study.

\[
M_v \frac{dV}{dt} = F_{\text{friction}} - F_{\text{roadload}}
\]  \hspace{1cm} (4.31)

where

\[
F_{\text{friction}} = T_i \times N_{\text{gear}} \times N_{\text{final}} \times \frac{1}{R_{\text{wheel}}}
\]  \hspace{1cm} (4.32)

\[
F_{\text{roadload}} = \frac{1}{2} C_{\text{air}} A F V^2 + C_{\text{rolling}} M_g
\]  \hspace{1cm} (4.33)

Again, in crank angle:

\[
\frac{dV}{d\theta} = \frac{1}{M \omega} (F_{\text{friction}} - F_{\text{roadload}})
\]  \hspace{1cm} (4.34)
4.2 Model Identification

Among the subsystems of the mean value engine model derived above, subsystems related to air dynamics and fuel dynamics are identified using experimental data. The identified parameters are the throttle discharge coefficient $C_{ds}$, the cylinder charge efficiency $\eta_c$, for the intake air dynamics, the fuel evaporation time constant $\tau_f$ and the fraction of fuel entering cylinder $X_f$ for the fuel dynamics (wall wetting model). The experimental setup and identification procedures are illustrated in the sequel.

4.2.1 Experimental Setup

The engine under investigation is Ford V8 4.6l, 8-valve engine connected to dynamometer through a 4 speed automatic transmission. The dynamometer is controller by a DYNE LOC throttle and dynamometer controller. For data acquisition, an IBM Pentium PC equipped with two National Instruments AT-MIO-E2 data acquisition boards is used. The AT-MIO-E2 is linked to the breakout box of the production engine control unit and related sensors. An IBM 486 PC equipped with a Texas instrument DSP TMS-320C30 board is also used to control the fuel injection directly, without using the production engine control unit. In addition to the sensors already installed to the engine by the manufacturer, an optical encoder, manifold pressure sensor and linear oxygen sensor (UEGO) are used.

![Diagram](image)

Figure 4.7: Schematics of experimental setup
The schematic diagram for the experimental setup is shown in Figure 4.7 and the detail specifications for the engine and sensor calibrations are listed in Appendix A.

4.2.2 Identification of Intake Air Dynamics

The mass flow rate through the throttle can be represented by a nonlinear algebraic function (either for choked or unchoked flow) of the intake manifold pressure multiplied by the throttle discharge coefficient. Also, the cylinder charge efficiency is derived based on the assumption that the intake manifold pressure is constant, i.e., the intake air dynamics are in steady state condition. Therefore, the parameters related to the intake air dynamics are identified using data from steady state experiments. To this end, a series of steady state experiments over various engine operation conditions was performed. Due to the limitation of the experimental set-up, it was impossible to cover the whole operating range of the engine. However, the operation conditions for the steady state experiments encompass the engine operating range from the idle condition to part-load conditions covering much of the operating conditions encountered during an EPA FUDS test as shown in Figure 4.8.

![Figure 4.8: Operating conditions for steady state experiments](image)

127
Identification of throttle discharge coefficient - $C_{bh}$

To determine the mass flow rate of air through the throttle, it is required to identify the discharge coefficient of the throttle. Reconsider equation (4.2) and (4.3),

for unchoked flow

$$m_{\text{un}} = \frac{C_{bh}A_{in}p_{m}}{\sqrt{RT_a}}\left(\frac{p_m}{p_s}\right)^{\frac{1}{\gamma}}\left[\frac{2\gamma}{\gamma-1}\left[1 - \left(\frac{p_m}{p_s}\right)^{\frac{(\gamma-1)}{\gamma}}\right]\right]^{\frac{1}{2}}$$

$$\frac{p_m}{p_s} > \left[\frac{2}{\gamma+1}\right]^{\gamma/(\gamma-1)}$$

(4.2)

for choked flow

$$m_{\text{ch}} = \frac{C_{bh}A_{in}p_m}{\sqrt{RT_a}}\gamma\left(\frac{2}{\gamma+1}\right)^{(\gamma+1)/(\gamma-1)}$$

$$\frac{p_m}{p_s} \leq \left[\frac{2}{\gamma+1}\right]^{\gamma/(\gamma-1)}$$

(4.3)

These two equations can be simplified as

$$m_{\text{un}} = C_{bh}A_{in}\psi_1(\cdot)$$

(4.35)

$$m_{\text{ch}} = C_{bh}A_{in}\psi_2(\cdot)$$

(4.36)

where $\psi_1(\cdot)$ is the flow function for unchoked flow and $\psi_2(\cdot)$ is the flow function for choked flow. Since the quantities in the flow functions can be measured (intake manifold pressure and the pressure of air coming in) and the throttle area $A_{in}$ also be computed using measured variables (refer to equation (4.1)), the throttle discharge coefficient can be identified by a least squares algorithm using the data from the steady state experiments mentioned above.

The discharge coefficient identified using equation (4.35) and (4.36) is $C_{bh} = 0.5665$. With the identified discharge coefficient, the mass flow rate of air through throttle is estimated. Both the estimated and the measured mass flow rate of air at throttle are plotted in Figure 4.9.
Figure 4.9: Measured and estimated mass flow rate of air

As shown in the Figure 4.9, the estimation error shows an offset over the entire range of throttle opening. To remove the offset, the structure of equation (4.35) and (4.36) is modified by introducing the offset as shown in equation (4.37) and (4.38).

\[
\dot{m}_{\text{ah}} = C_{\text{ao}} + C_{\text{ao}}A_o y_1() \tag{4.37}
\]

\[
\dot{m}_{\text{ah}} = C_{\text{ao}} + C_{\text{ao}}A_o y_2() \tag{4.38}
\]

where \(C_{\text{ao}}\) is an offset. The identified discharge coefficient and its offset are \(C_{\text{ao}}=0.56727\) and \(C_{\text{ao}}=0.00203\) respectively. Again, with the identified discharge coefficients, the mass flow rate of air at throttle is estimated. The estimated and the measured mass flow rate of air are plotted in Figure 4.10.
Figure 4.10: Measured and estimated mass flow rate of air

As shown in Figure 4.10, the estimation error is unnoticeable. The estimation errors for the initial identification and the identification with modified structure are compared in Table 4.1.

<table>
<thead>
<tr>
<th></th>
<th>Absolute mean error</th>
<th>RMS error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial estimation</td>
<td>0.0019256</td>
<td>0.00055803</td>
</tr>
<tr>
<td>Modified estimation</td>
<td>-0.0001375</td>
<td>0.00054932</td>
</tr>
</tbody>
</table>

Table 4.1 Estimation error

The modified structure (4.37) and (4.38) is reasonable since there is still some flow of the air when the throttle is closed (throttle opening = 0%). And this corresponds to offset $C_o$. 
Identification of cylinder charge efficiency ($\eta_c$)

To compute the mass flow rate of air into cylinders, it is important to identify the cylinder charge efficiency as described in the speed density equation (4.10).

$$\dot{m}_{\text{air}} = \frac{\eta_c V_{\text{d}} \omega}{4\pi RT_s}$$

(4.10)

In the conventional approaches, the cylinder charge efficiency is stored in a look-up table as a function of engine operating conditions such as the engine speed and the intake manifold pressure. Instead of using a look-up table, a polynomial model structure is adopted for the cylinder charge efficiency and the parameter of the polynomial is then identified by the least squares method using the steady state experimental data set. Several different polynomials are examined:

Type A: $\eta_c = a_0 + a_1 \omega + a_2 P_n$

(4.39a)

Type B: $\eta_c = a_0 + a_1 \omega + a_2 \omega^2 + a_3 P_n$

(4.39b)

Type C: $\eta_c = a_0 + a_1 \omega + a_2 \omega^2 + a_3 P_n + a_4 P_n^2$

(4.39c)

Type D: $\eta_c = a_0 + a_1 \omega + a_2 \omega^2 + a_3 P_n + a_4 \alpha$

(4.39d)

Type E: $\eta_c = a_0 + a_1 \omega + a_2 \omega^2 + a_3 P_n + a_4 \alpha + a_5 \alpha^2$

(4.39e)

The estimation errors for the different polynomials are compared in Table 4.2.

<table>
<thead>
<tr>
<th>Type</th>
<th>Absolute mean error</th>
<th>RMS error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type A</td>
<td>0.0279</td>
<td>0.0339</td>
</tr>
<tr>
<td>Type B</td>
<td>0.0269</td>
<td>0.0316</td>
</tr>
<tr>
<td>Type C</td>
<td>0.0238</td>
<td>0.0303</td>
</tr>
<tr>
<td>Type D</td>
<td>0.0281</td>
<td>0.0370</td>
</tr>
<tr>
<td>Type E</td>
<td>0.0088</td>
<td>0.0114</td>
</tr>
</tbody>
</table>

Table 4.2 Estimation error for different model
As shown in Table 4.2, the polynomial of the type E gives the best estimate of the cylinder charge efficiency. The parameter and structure of type E is:

\[
\eta_r = a_0 + a_1 \omega + a_2 \omega^2 + a_3 P_n + a_4 \alpha + a_5 \alpha^2
\]  

(4.40)

where

\[
\begin{align*}
  a_0 & = 0.67661467277010 \\
  a_1 & = -0.00327492921013 \\
  a_2 & = 0.00000613532992 \\
  a_3 & = -0.0005492984235 \\
  a_4 & = 0.05345607570316 \\
  a_5 & = -0.00080695504720.
\end{align*}
\]

The computed cylinder charge efficiency using the speed density method and the estimated cylinder charge efficiency using the polynomial (4.40) are depicted in Figure 4.11.

![Graphs of computed and estimated cylinder charge efficiency](image)

Figure 4.11: Computed and estimated cylinder charge efficiency

(throttle opening, engine speed and manifold pressure vs. cylinder charge efficiency)
4.2.3 Identification of Fuel Dynamics (Wall Wetting Model)

Having completed the identification of the air dynamics using steady state experiments, parameters related to the fuel dynamics (wall wetting model) are considered next. Recalling the fuel dynamic model in equation (4.12),

\[
\frac{dm_f}{d\theta} = -\frac{1}{\tau_f \omega} m_f + \frac{X}{\omega} \dot{m}_f
\]

\[
\dot{m}_f = \frac{1}{\tau_f} m_f + (1 - X) \dot{m}_f
\]

(4.12)

where the parameter \( \tau_f \) and \( X \) are to be identified. There is a difficulty in this identification problem since the mass of fuel in the fuel film can not be easily measured. This difficulty can be resolved by combining equation (4.12) with equation (4.10), (4.15) and (4.17) since the equivalence ratio is measured with the UEGO sensor, i.e.,

\[
\frac{dm_f(\theta)}{d\theta} = -\frac{1}{\tau_f \omega} m_f(\theta) + \frac{X}{\omega} \dot{m}_f(\theta)
\]

\[
\frac{d\phi_n(\theta)}{d\theta} = -\frac{1}{\tau_n \omega} \phi_n(\theta) + \frac{1}{\tau_f \tau_n \omega} \frac{m_f(\theta - \theta_t - \theta_c)}{\dot{m}_f(\theta - \theta_t - \theta_c)}
\]

\[
\frac{d\phi_n(\theta)}{d\theta} = \frac{1}{\tau_f} \phi_n(\theta) + \frac{1}{\tau_n \omega} \frac{m_f(\theta - \theta_t - \theta_c)}{\dot{m}_f(\theta - \theta_t - \theta_c)} + \frac{1 - X}{\tilde{\tau}_n \omega} \frac{\dot{m}_f(\theta - \theta_t - \theta_c)}{\dot{m}_f(\theta - \theta_t - \theta_c)}
\]

(4.41)

(4.42)

where \( \phi_n \) and \( \omega \) is measured, \( \dot{m}_f \) is known and \( \dot{m}_f \) can be calculate using the pumping equation (4.10).

The parameter \( \tau_n \) is also known and the cycle delay \( \theta_c \) can be computed from the valve timing of the given engine. In order to remove the transport delay and the possible exhaust gas mixing in the exhaust manifold, the UEGO sensor is placed on the just downstream of the exhaust valve. With this simplification, the model for the fuel dynamics identification can be represented as

\[
\frac{dm_f(\theta)}{d\theta} = -\frac{1}{\tau_f} m_f(\theta) + Xc_i \dot{m}_f(\theta)
\]

\[
\frac{d\phi_n(\theta)}{d\theta} = -c_2 \phi_n(\theta) + \frac{1}{\tau_f} m_f(\theta - \theta_c) + (1 - X)c_i \dot{m}_f(\theta - \theta_c)
\]

\[
\gamma(\theta) = \phi_n(\theta)
\]

(4.43)

(4.44)

(4.45)
where \( c_1 = \frac{1}{\omega} \), \( c_2 = \frac{1}{\tau_n \omega} \), and \( c_3 = \frac{1}{\tau_n \omega m_w} \).

The identification problem is formulated based on the Output Error Minimization (OEM) method. And it can be stated as follows. First, define the output error \( e(\theta) = y(\theta) - \hat{y}(\theta) \) where \( y(\theta) = \phi_n(\theta) \) is the measured value and its estimate \( \hat{y}(\theta) = \hat{\phi}_n(\theta) \) is calculated using the model described in equations (4.43)-(4.44) with the parameter \( \tau_j \) and \( X \) which is to be adjusted.

\[
\frac{d\hat{m}_n(\theta)}{d\theta} = -\frac{1}{\hat{\xi}_j} c_1 \hat{\dot{m}}_n(\theta) + \hat{\xi}_j c_3 \hat{m}_n(\theta) \tag{4.46}
\]

\[
\frac{d\hat{\phi}_n(\theta)}{d\theta} = -c_2 \hat{\dot{\phi}}_n(\theta) + \frac{1}{\hat{\xi}_j} c_3 \hat{m}_n(\theta - \theta_c) + (1 - \hat{X}) c_3 \hat{m}_n(\theta - \theta_c) \tag{4.47}
\]

\[
\hat{y}(\theta) = \hat{\phi}_n(\theta) \tag{4.48}
\]

Then, the square sum of the estimation error over the data range is defined as a cost function to minimize as given in equation (4.49).

\[
J = \int [e(\theta)]^2 d\theta \tag{4.49}
\]

The cost function \( J \) is minimized with respect to the parameter \( \tau_j \) and \( X \). The minimization is the based on the nonlinear least squares method.

It is reported that both \( \tau_j \) and \( X \) are depend on the intake wall temperature and the engine speed (Chang et al., 1995, Shayler et al., 1995). To take into account the effects of both intake wall temperature and engine speed, the experiment described above has been repeated with different engine coolant temperatures and engine speeds. The engine coolant temperature is used instead of the intake manifold wall temperature since it is the commonly used variable in the engine control unit and close to the intake manifold temperature compared to the other temperature (e.g. air charge temperature) used in an engine. The experiments are repeated with three different engine coolant temperatures and 6 different engine speeds as listed in Table 4.3.
<table>
<thead>
<tr>
<th>Engine Speed</th>
<th>Throttle Opening [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>43°C (110°F)</td>
<td>780 RPM 950 RPM 1050 RPM 1450 RPM 1650 RPM</td>
</tr>
<tr>
<td>66°C (150°F)</td>
<td>780 RPM 950 RPM 1050 RPM 1450 RPM 1650 RPM</td>
</tr>
<tr>
<td>82°C (180°F)</td>
<td>780 RPM 950 RPM 1050 RPM 1450 RPM 1650 RPM</td>
</tr>
</tbody>
</table>

Table 4.3 Experimental conditions for fuel dynamics identification

To excite the fuel dynamics only while keeping the other dynamics steady, it is required to keep the engine load and the throttle at a constant value. The fuel pulse width is manipulated so that the actual fuel flow through the fuel injector has a step change (step-up and then step-down fashion). The fuel flow rate and the response of the UEGO sensor to the step change in fuel flow are shown in Figure 4.12. This figure illustrates the typical input and output used in the identification procedure.

![Figure 4.12: Step changes in fuel flow rate and UEGO response](image)

Figure 4.12: Step changes in fuel flow rate and UEGO response

Figure 4.13 shows typical responses of the UEGO sensor for 3 different engine speeds. In these experimental data sets, the engine coolant temperature is kept constant and 82°C. Note that at the higher engine speed, the UEGO response is faster compared with the case when the engine speed is slow. In
Figure 4.14, the responses of the UEGO sensor for 3 different engine coolant temperatures are depicted. During the experiments for these cases, the engine speed is maintained as 1100 RPM with 10% throttle opening. It can be observed that the response of the equivalence ratio gets slower as the engine temperature decreases.

Figure 4.13: UEGO responses from the fuel identification experiments (3 different engine speeds)

Figure 4.14: UEGO responses from the fuel identification experiments (3 different temperatures)
Figure 4.14: UEGO responses from the fuel identification experiments (3 different temperatures)

With the experimental data from the different engine operating conditions as listed in Table 4.3, both $\gamma$ and $X$ are identified. The identification results are shown in Figure 4.15 and 4.16.

Figure 4.15: Fuel evaporation time constant for various engine speed and engine temperature

Figure 4.16: Fraction of fuel deposited into fuel film with different operating conditions
Figure 4.16: Fraction of fuel deposited in fuel film for various engine speed and engine temperature

It can be noted that both \( \tau_f \) and \( X \) decrease as the engine temperature increases and have smaller value at the higher engine speed. In case of the fuel evaporation time constant \( \tau_f \), it varies almost linearly with the engine speed at the same engine temperature. To examine the possibility of removing the effect of varying engine speed, define a new variable by multiplying the engine speed by the fuel evaporation time constant, i.e.,

\[
\theta_f = \tau_f \times \omega
\]  

(4.50)

Now, the new variable \( \theta_f \) that can be obtained by multiplying the engine speed by the data shown in Figure 4.15 is depicted in Figure 4.17

![Fuel evaporation time constant vs engine speed](image)

Figure 4.17: Fuel evaporation characteristics with new variable \( \theta_f \)

As shown in Figure 4.17, the dependence of \( \theta_f \) on the engine speed is negligible, and \( \theta_f \) only depends on the engine temperature. To illustrate this fact, both \( \tau_f \) and \( \theta_f \) shown in Figure 4.15 and Figure 4.17 are normalized (divided by their maximum value) and depicted in Figure 4.18.
One of the difficulties in air-fuel control system is the dependence of the fuel evaporation time constant on the engine speed, since the system becomes the time varying system. However, this problem can be solved by using a crank-angle based fuel dynamic model (Rizzoni, 1998) since the variable $\theta_e = \tau_e \times \omega$ is not depend on the engine speed anymore as illustrated in Figure 4.18. This result needs to be further verified over a wide range of engine operating conditions, since the result obtained above based on a limited set of engine operating conditions.

To validate the identification results, the measured UEGO response and the predicted UEGO response using the identified parameters (both $\tau_e$ and $X$) are depicted in Figure 4.19. In Figure 4.19, the rms (root mean square) error denotes the root mean square error between the measured and the predicted equivalence ratio.
Figure 4.19: Measured and predicted UEGO response
4.3 Experimental Study on Estimation and Identification

In this section, the various estimation and identification algorithms illustrated in Chapter 3 are applied to IC engine. Application results are demonstrated with experimental data.

4.3.1 Estimation of Cylinder Air Charge and Efficiency

One of the major obstacles to the AFR control problem is the estimation of the mean mass air flow rate entering the engine, $\dot{m}_a$, since this quantity is impossible to measure directly during transients. Using the well-known speed-density method, it is possible to estimate $\dot{m}_a$ using a table of experimentally determined values of charge efficiency $\eta_a$. The sliding mode identification algorithm introduced earlier is implemented to estimate $\eta_a$ on-line, and to provide a comparison with the steady-state estimate of air charge.

Rearrange the intake manifold dynamics model (4.7) in the form

$$\frac{dp}{dQ} = \eta_a p_a + b m_{at}$$

$$a = -\frac{V_d}{4\pi V_p}, \quad b = \frac{RT_p}{\omega V_p}$$

and define the model

$$\dot{y} = \hat{a}x + bu + v$$

(4.51)

where

$$v = -M\text{sign}(e) \quad M > 0$$

(4.52)

$$e = y - p_a$$

(4.53)

Let

$$\dot{\hat{\eta}} = -\lambda e$$

(4.54)

After finite time, a sliding mode occurs and as $e$ becomes zero, $\hat{\eta}$, approaches to $\eta_a$. Finally, the estimated mean mass flow rate of air into the engine is given by
\[ \dot{m}_{sc} = \dot{\bar{\eta}} \frac{\omega V_m}{RT_s} p_m \]  

(4.55)

The estimation results are compared with measured data obtained from a production mass air flow meter (only possible at steady-state), and with the charge efficiency predicted by an empirical model in a form of a polynomial function of engine speed \( \omega \), manifold pressure \( p_m \) and throttle opening \( \alpha \).

\[ \eta_s = 1.9751 \times 10^{-1} - 3.6075 \times 10^{-3} \omega + 7.0125 \times 10^{-6} \omega^2 - 5.3226 \times 10^{-8} p_m + 4.8898 \alpha \]  

(4.56)

Figure 4.20 compares the charge efficiency \( \eta_s \), estimated from the sliding mode observer with the estimate obtained using the polynomial approximation for two different operating conditions. Figure 4.21 repeats the comparison for the air charge variable, adding also the measured air charge, for the same experimental conditions. The estimation results of the same variables during throttle transient are illustrated in Figure 4.22. Unfortunately, it is impossible to obtain a reliable measurement of the air flow rate during a transient.
Figure 4.20: Estimation result during steady state ($\alpha = 5\%$, $\omega = 860$ RPM, $p_m = 50$ kPa)
Figure 4.21: Estimation result at steady state ($\alpha=10\%, \omega = 1550\text{RPM}, p_m = 50\text{kPa}$)
Figure 4.22: Estimation result during throttle tip-in (throttle step change $15^\circ \rightarrow 20^\circ$ at 35th cycle)
4.3.2 Estimation of Throttle Angle

In some diagnostic applications, it may be useful to estimate the throttle opening angle, $\alpha$, using a measurement of manifold pressure, $p_m$. Since the throttle angle is an input to the intake manifold dynamic system, we adopt the nonlinear input estimator design approach described by equations (3.70)-(3.75).

Define the model

$$\dot{y} = ap_n + bv$$  \hspace{1cm} (4.57)

where $a = \eta_a V_a / 4 \pi V_m$ and $b = RT_a / \omega V_m$.

The discontinuity function $\nu$ is chosen as follows.

$$\nu = -M \text{sign}(e) \quad M > 0$$  \hspace{1cm} (4.58)

$$e = y - p_m$$  \hspace{1cm} (4.59)

Then, the sliding mode can be described by

$$\dot{e} = b[m_{an} - \nu].$$  \hspace{1cm} (4.60)

After finite time, sliding occurs and the average value of the discontinuous function $\nu$ approaches the control $u$ which corresponds to $m_{an}$. After low-pass filtering the discontinuous function $\nu$, $\zeta$ converges to $m_{an}$. Finally, the throttle opening angle $\alpha$ can be obtained using the nonlinear algebraic relation (4.1) using the estimated $m_{an}$ and the measured manifold pressure $p_m$.

Experimental estimation results are depicted in Figure 4.23, showing reasonably accurate estimation of throttle angle during step change in throttle angle.
Figure 4.23: Estimation result during throttle tip-in
4.3.3 Estimation of Fuel Film Dynamics and AFR using Linear Oxygen Sensor

To illustrate the fuel film and AFR estimation approach presented in Chapter 3, we first make use of the proportional measurement of air-fuel ratio provided by the UEGO sensor (Yamada et al., 1992). In the presence of such a measurement, the plant equation becomes

\[
\frac{d\phi_{\text{UEGO}}(\theta)}{d\theta} = -\frac{1}{\tau_{\text{UEGO}}\omega} \phi_{\text{UEGO}}(\theta) + \frac{1}{\tau_{\text{UEGO}}\omega} \phi_1(\theta - \theta_d) \tag{4.61}
\]

with the output of \( y = \phi_{\text{UEGO}} \).

The estimation problem is approached by first estimating the in-cylinder AFR with delay, \( \hat{\phi}_1(\theta - \theta_d) \). Then, the estimate of fuel film mass \( \hat{m}_f(\theta - \theta_d) \) can be obtained knowing \( \hat{m}_f(\theta - \theta_d) \). An observer for the estimation of \( \phi_1 \) is constructed as follows.

\[
\frac{d\hat{\phi}_{\text{UEGO}}(\theta)}{d\theta} = -\frac{1}{\tau_{\text{UEGO}}\omega} \hat{\phi}_{\text{UEGO}}(\theta) + \frac{1}{\tau_{\text{UEGO}}\omega} v \tag{4.62}
\]

\[
v = M \text{sign}(\phi_{\text{UEGO}} - \hat{\phi}_{\text{UEGO}}) \tag{4.63}
\]

After finite time, sliding occurs and the equivalent value of \( v \) approaches \( \phi_1(\theta - \theta_d) \). The in-cylinder AFR estimate can be obtained through a low pass filtering the discontinuous function \( \Psi \):

\[
\hat{\phi}_1(\theta - \theta_d) = v_{eq} = z \tag{4.64}
\]

After calculating \( \hat{m}_f(\theta - \theta_d) \) using \( \hat{\phi}_1(\theta - \theta_d) \) and \( \hat{m}_f(\theta - \theta_d) \), \( \hat{m}_f(\theta) \) is then estimated by the convolution of the fuel film dynamics from \( \theta - \theta_d \) to \( \theta \) in the form of a predictor:

\[
\hat{m}_f(\theta) = e^{-\frac{1}{\tau_{\text{UEGO}}} \theta} \hat{m}_f(\theta - \theta_d) + \int_{\theta - \theta_d}^{\theta} e^{-\frac{1}{\tau_{\text{UEGO}}} \tau} \frac{d}{d\tau} \hat{m}_f(\tau) d\tau \tag{4.65}
\]

Figure 4.24 depicts the estimation result based on this method. The fuel flow rate through the fuel injector is estimated by augmenting an additional state variable for \( \hat{m}_f \) using the UEGO sensor measurement.

Since the commanded fuel flow rate for the injector is known, just additive (possibly injector fault) fuel flow rate is estimated and then added to the commanded fuel flow rate. Without a presence of the injector fault, the additional state variable has zero value and constant disturbance can be obtained with a constant
injector fault (Kim et al., 1997; Krishnaswami, 1996). This estimation can be used for diagnostic for the fuel injector. Estimation results are depicted in Figure 4.25.

Figure 4.24: Fuel film mass estimation result during throttle tip-in and tip-out
Figure 4.25: Injected fuel flow rate estimation result during throttle tip-in and tip-out
4.3.4 Estimation of Fuel Film Dynamics and AFR using Binary Sensor

Estimation of fuel flow from fuel film, \( \dot{m}_f \), can be accomplished by using the observer proposed in Chapter 4 - observer with binary measurement. With this observer, in addition to the estimate of \( \dot{m}_f \), the linear AFR can also be estimated. Observer design procedure is illustrated below.

Combine equations related to fuel dynamic (4.9), Air to fuel ratio in a cylinder (4.12) and delay and sensor dynamic (4.14) as:

\[
\frac{dx_1(\theta)}{d\theta} = -\frac{1}{\tau_f \omega} x_1(\theta) + \frac{X}{\omega} u(\theta) \tag{4.66}
\]

\[
\frac{dx_2(\theta)}{d\theta} = -\frac{1}{\tau_a \omega} x_2(\theta) + \frac{1}{\tau_a \omega} \frac{x_1(\theta - \theta_d)}{m_\omega(\theta - \theta_d)} + \frac{1 - X}{\tau_a \omega} \frac{u(\theta - \theta_d)}{m_\omega(\theta - \theta_d)} \tag{4.67}
\]

where the state variable is \( x = [\dot{m}_f, \phi_a] \) and input \( u = \dot{m}_f \). To simplify the expressions, let

\[
a_1 = \frac{1}{\tau_f \omega}, \quad a_{12} = \frac{1}{\tau_a \tau_f \omega m_\omega(\theta - \theta_d)}, \quad a_2 = \frac{1}{\tau_a \omega} \tag{4.68}
\]

\[
b_1 = \frac{X}{\omega}, \quad b_2 = \frac{1 - X}{\tau_a \omega m_\omega(\theta - \theta_d)} \tag{4.69}
\]

then, equation (4.66) and (4.67) become

\[
\dot{x}_1(\theta) = -a_1 x_1(\theta) + b_1 u(\theta) \tag{4.70}
\]

\[
\dot{x}_2(\theta) = -a_2 x_2(\theta) + a_{12} x_1(\theta - \theta_d) + b_2 u(\theta - \theta_d) \tag{4.71}
\]

and due to the switching characteristics of exhaust oxygen sensor, measurement equation becomes

\[
y = \text{sign}(x_2) \tag{4.72}
\]

The system given in (4.70)-(4.71) can be expressed in block diagram form as depicted in Figure 4.26.
Assuming that all the parameters in equation (4.70)-(4.71) are constant or at least slowly varying between sensor switching, the system given by equation (4.70)-(4.71) become equivalent to the system representation shown in Figure 4.27.

The equivalent system in Figure 4.27, the delay can be treated as measurement delay while in system shown in Figure 4.26 the delay is in the process. With newly introduced system, new state space representation can be defined as

\[ \dot{x}_1(\theta) = -a_1 x_1(\theta) + b_1 u(\theta) \]  

(4.73)

\[ \dot{x}_2(\theta) = -a_2 x_2(\theta) + a_{12} x_1(\theta) + b_2 u(\theta) \]  

(4.74)

\[ y(\theta) = sign(x_1(\theta)) = sign(x_2(\theta - \theta_d)) \]  

(4.75)

Now, the system does not have the process delay, it only has a measurement delay. If the delay is known, the observer can be constructed based on this system since the difficulty introduced by the measurement delay can be resolved by estimating the states up to instant \( \theta \) after the system is at the instant \( \theta + \theta_d \). Figures 4.28 shows the estimation of the fuel flow in fuel film and linear AFR. After an initial transient due to the
difference in initial conditions, the estimated states converges to the actual states within a reasonable accuracy except during throttle tip-in.

Figure 4.28: Observer responses during throttle tip-in
4.3.5 Estimation of Indicated Torque for an IC Engine

The system under consideration is an internal combustion engine connected to a dynamometer. An objective is to estimate an indicated torque of the engine. Measurements are the angular position of flywheel and dynamometer. The schematic diagram of the engine-dynamometer is shown in Figure 4.29.

![Schematic diagram of a single cylinder engine connected to a dynamometer](image)

Figure 4.29: Schematic diagram of a single cylinder engine connected to a dynamometer

The crankshaft model shown in Figure 4.29 can be described as follows.

\[ J_e \ddot{\theta}_e = -K(\theta_e - \theta_d) - B(\dot{\theta}_e - \dot{\theta}_d) - T_i - T_f + T_r \]

\[ J_d \ddot{\theta}_d = -K(\theta_d - \theta_e) - B(\dot{\theta}_d - \dot{\theta}_e) - T_r \]  

(4.76)

where \( J_e \) is the inertia of crankshaft and flywheel, \( J_d \) is the inertia of dynamometer, \( K \) is the torsional stiffness of crankshaft, \( B \) is the structural damping coefficient. The reciprocating torque \( T_r \) is,

\[ T_r = M_{eq} r^3 f_i(\theta_e) \left[ f_i(\theta_e) \dot{\theta}_e + f_z(\theta_e) \dot{\theta}_e^2 \right] \]  

(4.77)

where \( M_{eq} \) is the reciprocating mass of piston and part of connecting rod, \( r \) is the crank radius. The crank geometric function \( f_i \) and \( f_z \) are,

\[ f_i(\theta_e) = \sin \theta_e + \frac{\lambda \sin(2\theta_e)}{2\sqrt{1 - \lambda^2 \sin^2 \theta_e}} \]  

(4.78)

\[ f_z(\theta_e) = \frac{\lambda \cos(\theta_e)}{\sqrt{1 - \lambda^2 \sin^2 \theta_e}} + \lambda^3 \frac{\sin^2(2\theta_e)}{4(1 - \lambda^2 \sin^2 \theta_e)^{3/2}} + \cos \theta_e \]  

(4.79)
where $\lambda = n/\ell$ and $\ell$ is the connecting rod length. In this model, the friction loss and pumping loss is ignored except the viscous friction loss, so the friction torque $T_f$ is modeled as:

$$T_f = B \dot{\theta}_s$$

(4.80)

The indicated torque $T_I$ to be estimated, is:

$$T_I = P_i A_p r f_i(\theta_s)$$

(4.81)

where $P_i$ is an indicated cylinder indicated pressure, $A_p$ is a piston area. And $T_e$ is a load torque and is known. Input to the system. Substituting the equations from (4.77) to (4.81) into the equation (4.76) and rearranging them, we obtain a two degree-of-freedom nonlinear crankshaft model as:

$$\dot{\theta}_s = J \dot{\theta}_s - K(\theta_s - \theta_e) - B(\dot{\theta}_s - \dot{\theta}_e) - M_\alpha r^2 f_i(\theta_s) f_i(\theta_e) \dot{\theta}_e^2 + T_e$$

(4.82)

Define the state variables as $x_1 = \theta_s$, $x_2 = \dot{\theta}_s$, $x_3 = \theta_e$, $x_4 = \dot{\theta}_e$, input $u = T_f$ and output $y = [x_1 \quad x_2]^T$, this model can be written in a state space form as:

$$\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= [J + M_\alpha r^2 f_i^2(x_1)]^{-1} \left[-K(x_1 - x_3) - B(x_2 - x_4) - M_\alpha r^2 f_i(x_1) f_i(x_2) x_2^2 - T_e \right] \\
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= J^{-1} \left[-K(x_3 - x_4) - B(x_3 - x_4) - T_e \right]
\end{align*}$$

(4.83)

Now, design an observer for the state space model in equation (4.83) as follows. First, consider the states $x_i$ and $x_j$ that are measured in equation (4.83). The observer for $x_i$ and $x_j$ are

$$\begin{align*}
\dot{\hat{x}}_1 &= \hat{x}_2 + \lambda \text{sign}(x_1 - \hat{x}_1) \\
\dot{\hat{x}}_3 &= \hat{x}_4 + \lambda \text{sign}(x_3 - \hat{x}_3)
\end{align*}$$

(4.84)

Define the estimation error $e_i = x_i - \hat{x}_i$, $i = 1, \cdots, 4$. The error dynamics satisfies

$$\begin{align*}
\dot{e}_1 &= e_2 + \lambda \text{sign}(e_1) \\
\dot{e}_3 &= e_4 + \lambda \text{sign}(e_3)
\end{align*}$$

(4.85)
If the gain $L_i > |e_i|$, then a sliding mode occurs on a sliding manifold $s_i = e_i = 0$. Also if the gain $L_i > |e_i|$, then a sliding mode occurs on a sliding manifold $s_i = e_i = 0$. And the equivalent values of $L_i \text{sign}(e_i)$ and $L_i \text{sign}(e_i)$ satisfies

$$
\begin{align*}
\nu_1 & = L_i \text{sign}(e_i) e_i = e_i \\
\nu_2 & = L_i \text{sign}(e_i) e_i = e_i \\
\end{align*}
$$

(4.86)

where $\nu_1$ and $\nu_2$ can be obtained using equivalent filters. Now, apply this new information to estimate the $T_i$ and $x_i$ and follows.

$$
\begin{align*}
\dot{x}_1 &= [J_x + M_{a r} r^2 f_x(y_i)]^{-1} \{-K(y_i - y_i) - B(\dot{x}_i - \dot{x}_i) - F \dot{x}_2 \\
&- M_{a r} r^2 f_x(y_i) \dot{x}_i + L_i \text{sign}(\nu_i)\} \\
\dot{x}_i &= J_i \{-K(y_i - y_i) - B(\dot{x}_i - \dot{x}_i) - T_i + L_i \text{sign}(\nu_i)\}
\end{align*}
$$

(4.87)

The estimation error dynamics for $x_2$ and $x_3$ are expressed as

$$
\begin{align*}
\dot{e}_e &= [J_x + M_{a r} r^2 f_x(y_i)]^{-1} \{-B(e_2 - e_2) - B(e_2 - M_{a r} r^2 f_x(y_i)f_x(y_i)e_2^2 - T_i - L_i \text{sign}(\nu_i)\} \\
\dot{e}_e &= J_i \{-B(e_2 - e_2) - L_i \text{sign}(\nu_i)\}
\end{align*}
$$

(4.88)

With an appropriate choice of the gain $L_i > |B(e_2 - e_2) - T_i|$, a sliding mode occurs on a sliding manifold $s_j = e_j = 0$. After a finite time, the estimation error $e_j$ approaches to zero. Finally, a sliding manifold $s_j = e_j = 0$ can be reached using the gain $L_i$ such that

$$
L_i > |B(e_2 - e_2) - B(e_2 - M_{a r} r^2 f_x(y_i)f_x(y_i)e_2^2 - T_i|
$$

(3.89)

After the sliding mode is achieved, we have $e_i \to 0$, $i = 1, \ldots, 4$ and the equivalent value of $L_i \text{sign}(\nu_i)$ approaches to the indicated torque.

$$
\nu_2 = L_i \text{sign}(\nu_i) e_i \approx T_i
$$

(3.90)

Finally, the estimate of indicated torque $T_i$ is obtained using an equivalent filter

$$
\begin{align*}
\tau \ddot{v}_2 + \dot{v}_2 = L_i \text{sign}(\nu_i) \\
v_2 = \ddot{T}_i
\end{align*}
$$

(3.91)

where $\tau$ is an equivalent filter time constant to be selected.
The observer design just illustrated was applied to experimental data obtained from an engine-dynamometer setup. The experiment was set up for an engine coupled to a general electric DC dynamometer, as well as a DyneSystems Dyn-Loc IV dynamometer/throttle controller. Data was collected using an IBM/PS2 equipped with a National Instruments MC-MIO-16 A/D converter. Measured data includes; in-cylinder combustion pressure, engine rotation, dynamometer torque and rotation. The engine model parameters are listed in Table 4.4.

<table>
<thead>
<tr>
<th>Model Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inertia</td>
</tr>
<tr>
<td>Stiffness</td>
</tr>
<tr>
<td>Damping</td>
</tr>
<tr>
<td>Geometric Parameter</td>
</tr>
</tbody>
</table>

Table 4.4: Model parameters for Waukesha single cylinder engine

Figure 4.30 illustrates an estimation result using an experimental data. In this test, the engine was running at 800 RPM. The dynamometer generated a varying load torque. After a short transient, the estimated indicated torque converges to the indicated torque computed from a measurement of indicated pressure (Rizzoni, 1998).

![Figure 4.30: Estimation result from sliding mode unknown input observer](image-url)
4.3.6 Diagnosis Using NPERG Scheme with Observers

We consider a throttle position sensor fault and a fuel injector fault to demonstrate the diagnostic application of the estimation methods presented here. These are throttle sensor fault and fuel injector fault. Both faults have a strong influence on air-to-fuel ratio control performance. The throttle sensor fault is generated by an electronic circuit that reduces the sensor output. For the throttle sensor fault, the sensor output is reduced to 75% of the actual sensor output. A fuel injector fault was introduced by disabling one of the fuel injectors. The NPERG observer configuration depicted in Figure 4.31 is adopted to diagnose these two faults.

![Observer configuration diagram](image)

Figure 4.31: Observer configuration

The desired observers are designed based on the sliding mode approaches as illustrated in Chapter 3 and the previous Sections. The throttle sensor fault is generated by an electronic circuit that changes the calibration of the sensor output to 75% of the actual sensor output. A fuel injector fault is introduced by intermittently disabling one of the fuel injectors. Figure 4.32 shows the measured throttle angle and the corresponding residual for a throttle sensor fault introduced at approximately 100th engine cycle (Figure 4.32(a)). The immediate response of the throttle sensor residual to the throttle sensor fault is observed in Figure 4.32(b). The fuel injector command signal and the residuals for the fuel injector fault are depicted in Figure 4.33. The fuel injector fault occurs at 115th engine cycle after the initial step throttle change at 60th engine cycle. Figure 4.33(b) shows the prompt response of the fuel injector residual to the fuel injector fault.
Figure 4.32: Throttle sensor fault and the corresponding residual
Figure 4.33: Fuel injector fault and the corresponding residual
4.4 Simulation Study on Integrated Control and Diagnostics for Air and Fuel Management System

Integration of control and diagnostics in the context of powertrain control has not been formally studied thus far, although some general results on reliable or fault-tolerant control are available in the literature Jacobson and Nett (1991), Juarez et al. (1996), Kilsgaard et al. (1996), Patton (1997), Stengel (1991), Veillette and Perkins (1992) and Wu (1992). In this section, the integration of control and diagnostics for automotive engine air and fuel management system is explored based on the method proposed in Chapter 3 (use of Integral Sliding Mode Control and hypothesis observers). Problem formulation for both control and diagnostic perspective is introduced first. Then, the control design based on the estimation and identification is followed. At the end, the integration of control and diagnostics for the air and fuel management system is illustrated with simulation.

4.4.1 Problem Formulation

The subsystems related to air and fuel management system are the intake air dynamics, the pumping, the fuel dynamics (wall wetting model), the delay (both cycle delay and transportation delay) and the air fuel ratio dynamics. These systems are represented by the following equations.

\[
\frac{dp_n}{d\theta} = -\eta_n \frac{dV_n}{d\theta} p_n + \frac{RT}{V_n \omega} \dot{m}_{an} \tag{4.1}
\]

\[
\dot{m}_{an} = \frac{\eta_n p_n V_n \omega}{4\pi RT_o} \tag{4.2}
\]

\[
\frac{dm}{d\theta} = \frac{1}{\tau_f \omega} m + \frac{X}{\omega} \dot{m}_g \tag{4.3}
\]

\[
\dot{m}_g = \frac{1}{\tau_f} m_g + (1 - X) \dot{m}_g
\]

\[
\phi = \frac{1}{\tau_f} \dot{m}_g + (1 - X) \frac{\dot{m}_g}{\dot{m}_s}
\]

\[
\frac{d\phi_n}{d\theta} = -\frac{1}{\tau_n \omega} \phi_n + \frac{1}{\tau_n \omega} \phi_n (\theta - \theta_c - \theta_d) \tag{4.5}
\]

For notational simplicity, the above equations are converted into state space representation as follows.
\[ \dot{x}_1(\theta) = a_1 x_1(\theta) + b_1 u_1(\theta) \]
\[ \dot{x}_2(\theta) = a_2 x_2(\theta) + b_2 u_2(\theta) \]  \tag{4.92}
\[ \dot{x}_3(\theta) = a_3 x_3(\theta) + \frac{a_1 a_2}{b_1} x_2(\theta - \theta_d) + \frac{a_1 a_3}{b_1} x_3(\theta - \theta_d) \]

where \( x_1 = p_n \), \( x_2 = m_f \), \( x_3 = \phi_n \), \( u_1 = \dot{m}_a \), \( u_2 = \dot{m}_f \) and the parameters are:

\[ a_1 = -\frac{\eta V_d}{4\pi V_a}, \quad b_1 = \frac{RT_a}{V_a \omega}, \quad a_2 = -\frac{1}{r_f \omega}, \quad b_2 = \frac{X}{\omega}, \quad a_3 = -\frac{1}{r_a \omega}, \quad a_3 = \frac{1}{r_f}, \quad b_3 = 1 - X \]  \tag{4.93}

Also by introducing the intermediate variable \( z = \dot{m}_a \),

\[ z(\theta) = -\frac{a_3}{b_3} x_3(\theta) \]  \tag{4.94}

The system can be represented as

\[ \text{TYPE B} \]
\[ \dot{x}_1(\theta) = a_1 x_1(\theta) + b_1 u_1(\theta) \]
\[ \dot{x}_2(\theta) = a_2 x_2(\theta) + b_2 u_2(\theta) \]  \tag{4.95}
\[ \dot{x}_3(\theta) = a_3 x_3(\theta) + \frac{a_1 a_2}{z(\theta - \theta_d)} x_2(\theta - \theta_d) + \frac{a_1 b_3}{z(\theta - \theta_d)} x_3(\theta - \theta_d) \]

And the output equation for both types becomes

\[ y = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \]  \tag{4.96}

The reason for using the intermediate variable \( z \) is to consider the cylinder charge efficiency as an unknown (meaning it needs to be identified on-line rather than to be obtained either from look-up table or polynomial as explained earlier). From now on, Type B model is adopted, i.e., the cylinder charge efficiency is to be identified on-line (the parameter \( a_3 \) is unknown and to be identified on-line).
One objective of the AFR control problem is to keep the AFR as close to the stoichiometric ratio as possible under various operating conditions. By maintaining the AFR near to stoichiometric ratio, the catalytic converter operates near its maximum conversion efficiency, and exhaust emissions can be minimized. To keep AFR at the stoichiometric ratio (approximately 14.6 for gasoline), the fuel flow through a fuel injector, $\dot{m}_f$, is manipulated against any transient introduced by various disturbances such as throttle changes induced by driver (eventually resulting in changes in $\dot{m}_a$), load changes and gear shifts, and the action of the EGR and the canister purge control systems. These disturbances may cause a deviation of AFR from the stoichiometric ratio (denoted as $AFR_s$).

Best control performance may be achieved if accurate information on both $\dot{m}_a$ and $\dot{m}_f$ is available. Unfortunately, these variables cannot be easily measured in current production engines. To overcome this difficulty, most current production control strategies adopt off-line tuning and calibration methods that require significant effort. By taking advantage of a model of the relevant dynamics, it may be possible to estimate both $\dot{m}_a$ and $\dot{m}_f$ using available measurements such as manifold pressure, air flow rate through throttle, oxygen sensor voltage, etc. Therefore, accurate estimation of both quantities is a necessary pre-requisite to the control problem. The control goal is then specified as

$$x_3 - x_{3d} = x_3 - \phi_i = 0 \quad x_{3d} = \phi_i = \text{constant} \quad (4.97)$$

On-board diagnostic regulations require that all major components and subsystems related to the emissions control system be monitored, and that faults and failures be declared in a timely fashion to permit prompt repair. Thus, the diagnostic goal for air and fuel management system is to generate residuals for the faults in either sensor or actuator. The faults are represented as

$$y_1 = x_1 + \Delta y_1 \quad (4.98a)$$

$$y_2 = x_2 + \Delta y_2 \quad (4.98b)$$

$$u_1 = u_{1e} + \Delta u_1 \quad (4.98c)$$

$$u_2 = u_{2e} + \Delta u_2 \quad (4.98d)$$
The residuals corresponding to the fault defined in equation (4.98) are

\[
    r_1(t) = \begin{cases} 
        0 & \text{if } \Delta y_1 = 0 \\
        \Delta y_1 & \text{if } \Delta y_1 \neq 0 
    \end{cases} \tag{4.99a}
\]

\[
    r_2(t) = \begin{cases} 
        0 & \text{if } \Delta y_2 = 0 \\
        \Delta y_2 & \text{if } \Delta y_2 \neq 0 
    \end{cases} \tag{4.99b}
\]

\[
    r_3(t) = \begin{cases} 
        0 & \text{if } \Delta u_1 = 0 \\
        \Delta u_1 & \text{if } \Delta u_1 \neq 0 
    \end{cases} \tag{4.99c}
\]

\[
    r_4(t) = \begin{cases} 
        0 & \text{if } \Delta u_2 = 0 \\
        \Delta u_2 & \text{if } \Delta u_2 \neq 0 
    \end{cases} \tag{4.99d}
\]

In what follows, two control algorithms are designed for air fuel ratio regulation. One is for the use of nonlinear oxygen sensor (HEGO) and another is for the use of linear oxygen sensor (UEGO) sensor. In both algorithms, the cylinder charge efficiency will be identified on-line and the fuel film dynamics will be estimated to improve the transient AFR performance. At the end, an integrated control and diagnostics strategy is designed in which the linear oxygen sensor (UEGO) is utilized and the faults are introduced into the system. The design is implemented and is validated in the simulation environment (for details, see PTSIM in Appendix B.).

4.4.2 AFR Control via Estimation and Identification

Air fuel ratio regulation in most current engine control units is achieved by feedforward control (open-loop, determined by either mass flow meter method or speed-density method) and feedback correction using PI (Proportional and Integral) control. Any additional disturbances or uncertainties are attenuated by an additional corrective control action that is predetermined through an elaborate and costly calibration process.

\[
    \dot{m}_f = \dot{m}_{feedforward} + \dot{m}_{feedback} + \dot{m}_{adder} \tag{4.100}
\]

Each control action defined in equation (4.100) is explained below accompanied by difficulty involved. The feedforward term is determined by:

164
\[ \dot{m}_{\text{feedforward}} = \frac{\dot{m}_{\text{egr}}}{AFR_i} \]  \hspace{1cm} (4.101)

One difficulty involved in the calculation of feedforward term is that the mass flow rate of air entering a cylinder is not measured directly. Therefore, either the feedforward term or the mass flow rate of air entering a cylinder is usually stored in a look-up table for various operating conditions. The engine control unit determines the feedforward term from the look-up table for the specific operating condition.

The \textit{feedback correction term} can be expressed as

\[ \dot{m}_{\text{feedback}} = K_f (AFR_i - AFR) + K_i \int (AFR_i - AFR) dt \]  \hspace{1cm} (4.102)

A challenge encountered in determining the feedback term is the still prevalent use of a nonlinear heated exhaust gas oxygen (HEGO), with nearly binary characteristics [51]. The voltage output form the HEGO sensor can be approximately modeled as follows.

\[ V_{\text{HEGO}} = 1 - \text{sign} \left( AFR_i - \frac{\dot{m}_{\text{egr}}}{\dot{m}_{\text{p}}}, \right) \]  \hspace{1cm} (4.103)

Therefore, the feedback term based on the HEGO sensor can be determined through the following relation

\[ \dot{m}_{\text{feedback}} = K_f (0.5 - V_{\text{HEGO}}) + K_i \int (0.5 - V_{\text{HEGO}}) dt \]  \hspace{1cm} (4.104)

One of the typical examples of the \textit{additive corrective action} is the compensation of the transient fuel caused by the fuel film dynamics during throttle or load transient. As shown in equation (4.12), the actual fuel entering a cylinder consists of the fraction of fuel injected and the fuel evaporated from the fuel film.

\[ \dot{m}_r = \frac{1}{\tau_f} m_g + (1 - X) \dot{m}_g. \]  \hspace{1cm} (4.12)

Note that the feedforward term defined above is based on the assumption that all of the injected fuel enters the cylinder. This assumption is only valid for steady state. Therefore, air fuel ratio deviates from stoichiometry during throttle transient. The compensation for this transient fuel is not straightforward for
implementation since the fuel film mass is not easily measured. Most production engine control units also employ the look-up table approach for this application.

In summary, the challenges in AFR regulation addressed above are:

- Estimation of air entering cylinder for feed-forward term
- Use of nonlinear heated exhaust gas oxygen (HEGO) sensor for feedback
- Estimation of fuel film mass for transient fuel compensation

In what follows, an air-fuel ratio control algorithm is proposed to overcome the difficulties mentioned above.

*Air Fuel Ratio Control Using Nonlinear Oxygen Sensor*

Use of Heated Exhaust Gas Oxygen sensor for AFR regulation is explored here. Both an identification problem and an estimation problem are involved in the design process. The identification algorithm is required to identify the cylinder charge efficiency and to estimate the air entering a cylinder. In this approach, a manifold pressure sensor is utilized. Meanwhile, an estimation algorithm is implemented to estimate the fuel film dynamics using HEGO sensor.

The identification algorithm (shown in Chapter 3, from equation (3.114) to equation (3.127)) is employed to identify the cylinder charge efficiency and estimate the air entering a cylinder. The validation of this approach with experimental data is already explored earlier in this Chapter (section 4.3.1). The same algorithm is implemented for the engine simulation.

The estimation problem of fuel dynamics was extensively explored and illustrated earlier in this Chapter, and is not repeated here. For details, refer to Section 3.4 for the algorithm in general, and to Section 4.3.4 for implementation and experimental validation.
Now, since a reasonably accurate estimation of $z = \hat{m}_{\text{ac}}$ and $x_i = m_f$ is available, a new control input $u_i = \hat{m}_{\text{ac}}$ can be computed as follows. As shown in (4.12), the mean fuel mass flow rate entering the cylinder, $\bar{m}_f$, is

$$\bar{m}_f = \frac{1}{\tau_f} \hat{m}_f + (1 - X) \hat{m}_f.$$  \hspace{1cm} (4.12)

To keep the AFR near stoichiometry, we first define a base fuel required using the estimate of $\hat{m}_{\text{ac}}$ as

$$\hat{m}_{\text{base}} = \frac{\hat{m}_{\text{ac}}}{AFR}.$$  \hspace{1cm} (4.105)

Substituting (4.105) into (4.12) and solving for $\hat{m}_f$, the fuel flow rate for the transient fuel compensation is

$$\hat{m}_{\text{comp}} = \frac{X}{1 - X} \hat{m}_{\text{base}} - \frac{1}{(1 - X)\tau} \hat{m}_f.$$  \hspace{1cm} (4.106)

and the total fuel needed is the summation of the base and the compensated fuel, i.e.,

$$\hat{m}_f = \hat{m}_{\text{base}} + \hat{m}_{\text{comp}} = \frac{X}{1 - X} \hat{m}_{\text{base}} - \frac{1}{(1 - X)\tau} \hat{m}_f.$$  \hspace{1cm} (4.107)

In addition to this newly found control, the feedback (PI control) is also added in the engine simulator. The resulting control law for the AFR regulation is of the form

$$\hat{m}_f = \hat{m}_{\text{feedforward}} + \hat{m}_{\text{feedback}} = \frac{1}{1 - X} \frac{\hat{m}_{\text{ac}}}{AFR} - \frac{1}{(1 - X)\tau} \hat{m}_f + K_p (0.5 - \hat{V}_{\text{EGO}}) + K_i \int (0.5 - \hat{V}_{\text{EGO}}) d\Theta.$$  \hspace{1cm} (4.108)

Figure 4.34 illustrate the schematic diagram of the controller configuration based on the algorithm proposed above.
Figure 4.34: Schematic of controller configuration with nonlinear oxygen (HEGO) sensor

Figure 4.35 shows a comparison between the response from a conventional control method (feedforward + PI) and the response with transient fuel compensation using the method described above. As can be seen in Figure 4.35, the AFR excursion during transient is improved remarkably with transient fuel compensation.
Figure 4.35: AFR during throttle step change – simulation results
4.4.3 Integration of Control and Diagnostic

In this section, design of a fault tolerant control strategy through integration of control and
diagnostics is applied to the engine air and fuel management system. The algorithm was explored in
Chapter 3 and this section focuses on validation of the algorithm through simulation. A baseline control
design is introduced at the beginning. Then, a robust controller is developed using the integral sliding mode
control method. This controller also serves as a tool for fault detection. Next, observers with hypothesis
testing are designed for the purpose of fault isolation and identification. Finally, all of the above designs
are implemented in simulation and the effectiveness of the proposed design is examined.

Baseline Air-Fuel Ratio Controller Design

As a baseline AFR controller, the structure of the AFR controller designed in the previous section
is adopted. Accordingly, the approach used for the identification of cylinder charge efficiency and air
entering a cylinder is utilized again. However, instead of using the nonlinear oxygen (HEGO) sensor, the
proportional oxygen (UEGO) is utilized in the feedback and estimation process. Then, an integral sliding
mode control action is appended to the baseline AFR controller. The system with baseline controller is
repeated here for convenience.

\[
\dot{x}_1(\theta) = a_1 x_1(\theta) + b_1 u_1(\theta) \\
\dot{x}_2(\theta) = a_2 x_2(\theta) + b_2 u_2(\theta) \\
\dot{x}_3(\theta) = a_3 x_3(\theta) + \frac{a_3 a_{22}}{z(\theta - \theta_e)} x_2(\theta - \theta_e) + \frac{a_3 b_{22}}{z(\theta - \theta_e)} u_2(\theta - \theta_e) \\
z(\theta) = -\frac{a_3}{b_3} x_3(\theta)
\]

And the output equation is

\[
y = [x_1] \quad \text{(4.111)}
\]
The baseline AFR controller determines the control action by the following equation.

\[ u_i = \dot{m}_{\text{af}} \]

\[ = \dot{m}_{\text{feedforward}} + \dot{m}_{\text{feedback}} \]

\[ = \frac{1}{1 - X \ AFR_i} \frac{1}{(1 - X) \tau} \dot{m}_{\text{af}} + K_p (\phi_i - \phi) + K_i \int (\phi_i - \phi) d\theta \]

(4.112)

\[ = \frac{1}{1 - X \ AFR_i} \frac{1}{(1 - X) \tau} \dot{z}_i + K_p (\phi_i - y_i) + K_i \int (\phi_i - y_i) d\theta \]

Fundamentally, the above control has two components: a feedforward term determined through estimation and identification and a feedback term using the UEGO sensor. The configuration for the baseline AFR controller is depicted in Figure 4.36.

Figure 4.36: Schematic of controller configuration with proportional oxygen (UEGO) sensor
Design of Integral Sliding Mode Controller

In addition to the baseline controller, it is required to use the integral sliding control action for the detection and the compensation of a fault in the system. To design an integral sliding control, first find the integral term denoted by $\psi$. From the equation (3.154),

$$ \dot{\psi} = -\frac{\partial s_s}{\partial x} \left[ f(x) + g(x)u_s \right], \quad \psi(0) = -s_s(x(0)) $$  \hspace{1cm} (4.113)

where $s_s = x_1 - x_{3d}$ and $u_s = u_2$ for AFR control problem. Since, $\frac{\partial s_s}{\partial x} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$, equation (4.113) becomes

$$ \dot{\psi}(\theta) = -a_1 x_1(\theta) + \frac{a_2 a_{22}}{z(\theta - \theta_d)} x_1(\theta - \theta_d) + \frac{a_1 b_{32}}{z(\theta - \theta_d)} u_s(\theta - \theta_d), \quad \psi(0) = -x_1(0) + x_{3d}(0) $$  \hspace{1cm} (4.114)

And the sliding manifold for the integral sliding control is

$$ s = s_s + (\psi - \psi_d) $$  \hspace{1cm} (4.115)

with the integral sliding control term as

$$ v(\theta) = M \text{sign}(s) = M \text{sign}(s_s + (\psi - \psi_d)) $$  \hspace{1cm} (4.116)

To enforce the reaching condition with the above discontinuous control, two conditions (equation 3.150 and 3.151) should be satisfied. For the system under consideration,

$$ \frac{\partial s_s}{\partial x} g(x) = \frac{a_1 b_{32}}{z(\theta - \theta_d)} $$ is positive definite. \hspace{1cm} (4.117)

And the gain of the discontinuous control $M$ can be selected such that

$$ M > \left| \frac{h(x, \theta)}{g(x)} \right| $$  \hspace{1cm} (4.118)

where $h(x, \theta)$ is the fault to be considered later. To generalize the above condition, instead of $h(x, \theta)$, the upper bound of the fault may be used.

The system with integral sliding control is then represented as

172
\[ \dot{x}_1(\theta) = a_1 x_1(\theta) + b_1 u_1(\theta) \]
\[ \dot{x}_2(\theta) = a_2 x_2(\theta) + b_2 u_2(\theta) + v(\theta) \quad (4.119) \]
\[ \dot{x}_3(\theta) = a_3 x_3(\theta) + \frac{a_3 a_{32}}{z(\theta - \theta_x)} x_2(\theta - \theta_x) + \frac{a_3 b_{32}}{z(\theta - \theta_x)} u_2(\theta - \theta_x) + v(\theta - \theta_x) \]
\[ z(\theta) = -\frac{a_3}{b_3} x_1(\theta) \quad (4.120) \]

where the baseline AFR control \( u_2 \) and the integral sliding control \( v \)

\[ u_2 = \dot{m}_{\text{inlet}} + m_{\text{cork}} \]
\[ = \frac{1}{1 - X} \frac{z}{A F R_1} - \frac{1}{(1 - X)X} \dot{x}_2 + K_p (\phi_j - y_2) + K_i \int (\phi_j - y_2) d\theta \quad (4.121) \]
\[ v = M_{\text{sign}}(s) = M_{\text{sign}}((x_1 - x_{id}) + (\psi - \psi_d)) \quad (4.122) \]

As shown in Chapter 3, the equivalent value of the integral sliding control can be used to detect either fuel injector fault or UEGO sensor fault. In case of fuel injector fault, the equivalent value of \( v \) becomes

\[ v_{eq} = \Delta u_2 \quad (4.123) \]

On the other hand, with the UEGO sensor fault, the equivalent value of \( v \) becomes

\[ v_{eq} = -\frac{z}{a_3 b_{32}} \Delta \dot{y}_2 \quad (4.124) \]

And the equivalent value of \( v \) is used for the hypothesis observer for the fault isolation in following section.

Two tasks are accomplished in this section. First, an input (actuator) fault is compensated through the ISMC control. Second, the presence of a fault can be detected by monitoring the equivalent control \( v_{eq} \). Once a decision is made (on the existence of a fault), then this equivalent control is forwarded to hypothesis observers for isolation and identification of the fault.

\textit{Fault Isolation using Observers with Hypothesis}
The diagnostic objectives defined earlier include the detection of faults in the actuators and sensors and the corresponding residual generation. The faults to be identified are: \( \Delta y_1 \) (intake manifold pressure sensor fault), \( \Delta y_2 \) (UEGO sensor fault), \( \Delta u_1 \) (throttle actuator fault), \( \Delta u_2 \) (fuel injector fault). However, it is assumed that the cylinder charge efficiency is unknown and is to be identified on-line using the measured manifold pressure. Since the identification of cylinder charge efficiency requires healthy operation of the intake manifold pressure, the intake manifold pressure fault is not considered here. Furthermore, since the engine controller does not have any control authority on the throttle actuator (unless electronic throttle control is used), the diagnostics for the throttle actuator fault is limited to the detection and identification without fault accommodation. In this case, an input observer illustrated in Section 4.3.2 can serve as a diagnostic observer. Therefore, only two hypotheses are required: one for fuel injector fault isolation (Hypothesis A: Fuel injector is faulty) and one for the UEGO sensor fault isolation (Hypothesis B: UEGO sensor is faulty). Both observer structures are shown below.

**Observer with Hypothesis A: faulty fuel injector**

\[
\dot{x}_2(\theta) = a_x \dot{x}_1(\theta) + b_x u_2(\theta) + L_z(y_2 - \hat{y}_2) + M_z \text{sign}(y_2 - \hat{y}_2) + b_x \Delta u
\]  

(4.125)

\[
\dot{y}_2(\theta) = a_{y2} \dot{x}_2(\theta) + b_{y2} \Delta \dot{u} = \frac{a_x a_{y2}}{\dot{z}(\theta - \theta_d)} \dot{x}_2(\theta - \theta_d) + \frac{a_x b_{y2}}{\dot{z}(\theta - \theta_d)} u_2(\theta - \theta_d) + L_z(y_2 - \hat{y}_2) + M_z \text{sign}(y_2 - \hat{y}_2) + \frac{a_x b_{y2}}{\dot{z}(\theta - \theta_d)} \Delta \dot{u}(\theta - \theta_d)
\]  

(4.126)

**Observer with Hypothesis B: faulty UEGO sensor**

\[
\dot{x}_2(\theta) = a_x \dot{x}_1(\theta) + b_x u_2(\theta)
\]  

(4.127)

\[
\dot{y}_2(\theta) = a_{y2} \dot{x}_2(\theta) + b_{y2} \Delta \dot{u} = \frac{a_x a_{y2}}{\dot{z}(\theta - \theta_d)} \dot{x}_2(\theta - \theta_d) + \frac{a_x b_{y2}}{\dot{z}(\theta - \theta_d)} u_2(\theta - \theta_d) + L_z(y_2 - \hat{y}_2) + M_z \text{sign}(y_2 - \hat{y}_2)
\]  

(4.128)

The responses of two observers in the presence of either actuator fault or sensor fault can be summarized as in Tables 4.5 and 4.6.
Table 4.5 Response of hypothesis observer A and B

Residuals for input and output faults are defined as follows.

\[ r_A = y - \hat{y}, \quad \text{from hypothesis observer A} \]  
\[ r_B = y - \hat{y}, \quad \text{from hypothesis observer B} \quad (4.129a) \]

Using the residuals defined above, an isolation of fault can be accomplished by threshold test of the residuals and the decision logic in Table 4.6.

Table 4.6 Decision logic for isolation of fault

<table>
<thead>
<tr>
<th>Fault Condition</th>
<th>Residual ( r_A )</th>
<th>Residual ( r_B )</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Fault</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Actuator Fault ((\Delta u_2))</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Sensor Fault ((\Delta y_2))</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

If the fault appears to be a sensor fault (i.e., UEGO sensor fault), then the magnitude of the fault should be identified for compensation. Since we only have a single measurement, any of the isolation schemes (DOS, GOS, NPERG in Chapter 2) can not be applied here. However, it may be possible to identify the UEGO sensor fault using the method described in Section 3.5 (equations 3.163 – 3.169). Let’s examine the estimation error dynamics for the hypothesis observer B.

\[ \dot{e}_{s2}(\theta) = a_2 e_{s2}(\theta) \quad (4.130) \]
\( e_{s2}(\theta) = \hat{y}_2 - \hat{\hat{y}}_2 \)

\[
= \dot{x}_2(\theta) + \Delta \hat{y}_2(\theta) - \hat{y}_2 \\
= a_1(y_2(\theta) - \Delta y_2(\theta)) + \frac{a_1 a_{32}}{z(\theta - \theta_d)} z_1(\theta - \theta_d) + \frac{a_1 b_{32}}{z(\theta - \theta_d)} u_1(\theta - \theta_d) + \Delta \hat{y}_2(\theta) \tag{4.131}
\]

\[
- a_2 \dot{y}_2(\theta) - \frac{a_1 a_{32}}{z(\theta - \theta_d)} \ddot{x}_2(\theta - \theta_d) - \frac{a_1 b_{32}}{z(\theta - \theta_d)} u_1(\theta - \theta_d) \\
= a_2 e_{s2}(\theta) - a_1 \Delta y_2(\theta) + \Delta \hat{y}_2(\theta) + \frac{a_1 a_{32}}{z(\theta - \theta_d)} e_{s2}(\theta - \theta_d) - L_c e_{s2} - M_\delta \text{sign}(e_{s2})
\]

First, check the stability of the error dynamics for \( e_{s2} \). The coefficient \( a_2 = -(\tau \omega)^{-1} \) is negative since the fuel evaporation time constant and engine speed are positive. Therefore, the \( e_{s2} \) will approach zero in a finite time. Second, find the equivalent value of discontinuous function \( M_\delta \text{sign}(e_{s2}) \) from the error dynamics for \( e_{s2} \) (equation (4.131)). We can select the observer gain \( M_\delta \) such that a sliding occurs on \( e_{s2} = 0 \). Then, the equivalent value of \( M_\delta \text{sign}(e_{s2}) \) becomes

\[
\psi_{eq} = -a_1 \Delta y_2(\theta) + \Delta \hat{y}_2(\theta) \tag{4.132}
\]

We know that the estimate of \( \Delta \hat{y}_2(\theta) \) can be obtained using the equivalent control \( \psi_{eq} \) from the controller in case of the sensor fault. Therefore, we can find the estimate of the sensor fault by substituting

\[
\Delta \hat{y}_2 = -\frac{a_1 b_{32}}{z} \psi_{eq}
\]

into equation (4.132) and solve for \( \Delta y_2 \) as below.

\[
\Delta \hat{y}_2 = \frac{1}{a_1} (\Delta \hat{y}_2 - \psi_{eq}) \tag{4.133}
\]

The controller is configured to use estimated output rather than measured UEGO output. The estimated UEGO output is calculated using both the estimate of UEGO sensor fault, \( \Delta \hat{y}_2 \) and the measured UEGO output. (\( y_2 = x_3 + \Delta y \)). And the estimated output is defined by equation (4.134).

\[
\hat{y}_2 = y_2 - \Delta \hat{y}_2 \approx x_3 \tag{4.134}
\]

Now we completed the design of both control and diagnostics. The design is validated with simulation in following section.

Validation of Algorithm with Simulation

176
The baseline controller, the ISMC control and hypothesis observers are implemented in the simulation (PTSIM). The schematic diagram of the integrated control and diagnostics for air and fuel management system is shown in Figure 4.37.

Two cases are considered to examine the effectiveness of the design. The first case considers an actuator (fuel injector) fault and the second case examines a sensor (UEGO) fault.

Case A: Input (actuator, fuel injector) Fault

As an input fault, fuel injector fault is considered. The fuel injector injects 15% more fuel than the commanded value. Figure 4.38 depicts the actual fuel flow rate from the fuel injector. The injector fault is introduced around 36th engine cycle. The AFR response (UEGO sensor output) is illustrated in Figure 4.39. As the fuel injector fault is introduced, the AFR becomes rich. However, as soon as the ISMC control is activated, the AFR returns to stoichiometric value. The estimated injector fault is illustrated in
Figure 4.40. As explained earlier, the estimate of fuel injector fault is obtained using the equivalent control from the ISMC controller. Since an equivalent filter (low pass filter) is used to filter out the high frequency components, the delay in the estimation is unavoidable. The residuals defined in equation (4.129) are conditioned through a threshold tester (0 if the residual is within threshold, 1 if greater than threshold, -1 if smaller than threshold). Determination of the threshold itself is not a trivial task and it is beyond the scope of this study. Here, the threshold is chosen such that it can eliminate the effect of high frequency components induced by switching function in ISMC. The conditioned residuals (in Figure 4.41) show that the fault introduced at 38th engine cycle is an input (fuel injector) fault (see Table 4.6). The conditioned residuals are non-zero at the first few engine cycles due to the initial estimation error from the observers. Once the actuator fault is isolated and identified, the residual $r_A$ becomes zero as expected. And the residual $r_B$ becomes and remains non-zero.

![Figure 4.38: Injected fuel with fuel injector fault](image)
Figure 4.39: AFR response with fuel injector fault

Figure 4.40: Estimation of fuel injector fault
Case B: Output (UEGO sensor) Fault

Now, an UEGO sensor fault is considered. In simulation, the UEGO sensor output is modified to include 15% additive fault at 36th engine cycle. Figure 4.42 shows the UEGO sensor output where the sensor fault ($\Delta y_2=2$) is introduced at 36th engine cycle. As soon as the fault is detected by monitoring the equivalent control from the ISMC controller, the residuals from the observers are examined. Figure 4.43 illustrates the behavior of the conditioned residuals. From Figure 4.43 and Table 4.6, we can say that the fault introduced here is an output (UEGO sensor) fault. Once the fault is identified as a sensor fault, it is identified using the equivalent value of the ISMC control and the observer (equation 4.124). Figure 4.44 compares the UEGO sensor fault and the estimate fault. The identified fault ($\Delta y_2$) is then forwarded to controller for compensation. Figure 4.45 compares the UEGO response and the actual AFR (true AFR). Even after the UEGO sensor fault is introduced, the UEGO output remains at 16.7. This means that the actual (true) AFR stays at stoichiometric (1.47).
Figure 4.42: UEGO sensor output with fault

Figure 4.43: Residual behavior
Figure 4.44: Estimation of UEGO sensor fault

Figure 4.45: AFR response with UEGO sensor fault
4.5 Conclusion

This chapter has explored four different areas. First, an IC engine model, the so-called the mean value engine model, was described and implemented in Simulink™ (PTSIM in Appendix B). Some parameters of the engine model are identified through experiments. Second, various estimation methods mentioned in Chapter 3 are applied to the engine model. The estimation results illustrate that they are reasonably accurate to be used for control and diagnostics. Third, some of the estimator is utilized for the estimation approaches (the on-line parameter estimation and the estimator using a binary sensor) are implemented for air fuel ratio regulation. It is shown that the transient AFR response can be improved as a result of control based on the estimation. Finally, the algorithm for the integration of control and diagnostics is implemented in simulation. Two case studies (input fault case and output fault case) show that the proposed algorithm offer a fault tolerant control with fault diagnostic capability.
CHAPTER 5

CONCLUSIONS

On Fault Tolerant Control Design via Integration of Control and Diagnostics

1. A fault tolerant control is achieved through an integrated design of control and diagnostics. Information obtained in control and diagnostics modules are utilized such that a fault can be detected, isolated and compensated. In summary, role of control and diagnostics modules can be stated as follows.

- Control Module
  - Performs primary control task
  - Detection of fault using additive control terms (equivalent value of integral control term)
  - Input(actuator) fault compensation

- Diagnostics Module
  - Generation of analytic redundancy (estimation using observers)
  - Fault isolation
    i) Input(actuator) fault is isolated by observer using both output and estimated fault.
ii) Output(sensor) fault is isolated by decoupling a faulty sensor or by direct estimation of the fault. In the latter case, information obtained from controller (time derivative of sensor fault is obtained using integral sliding mode control term)

- For output (sensor) fault, it is necessary to modify either sensor output or desired trajectory. In broad sense, this can be described as controller reconfiguration.

2. Simulation study illustrates that the proposed approach can be a possible candidate for fault tolerant control using a model-based application.

3. Integral sliding mode control used in this study is sensitive to system parameter change. To accommodate parametric fault, further analysis is required to extend the applicability of the proposed approach.

4. When a fault is introduced to system, ISMC controller response is instantaneous. Immediate control response is good for an input fault. However, it is desirable for the controller to ignore an output fault. Also the system become sensitive to disturbance and noise. To resolve this problem, a delay in control intervention may be considered, i.e., let the control wait until fault isolation is completed.

On IC Engine Application

1. An IC engine air and fuel dynamics are identified using experimental data obtained through steady and transient test. A notable result is observed during fuel dynamics (wall wetting phenomenon) identification. It is well known that the fuel evaporation time constant is a function of intake wall temperature and engine speed. The identified fuel evaporation time constant matched well with this fact. Compensation of the transient fuel due to this fuel dynamics thus requires both intake wall temperature (in experiment, engine coolant temperature is used) and engine speed. However, by looking at the fuel dynamics in different domain (crank angle domain), it was possible to simplify this problem. A new constant in crank angle domain is thus defined by multiplying the engine speed to fuel evaporation time constant. This new constant turns out to be only dependant on engine coolant
temperature. This is one of the advantages in crank angle domain approach in engine modeling and control.

2. An online estimation of cylinder air charge is attempted using a parameter identification technique based on sliding mode methodology. Although it is validated with simulation, it will reduce significant amount of time and effort involved in open loop fuel calibration if it is implemented in onboard control application.

3. An estimation method using binary sensor (HEGO sensor) is developed and applied for fuel and AFR dynamics system. The initially proposed observer design procedure did not guarantee the convergence of observer. By transformation of the system and appropriate gain selection procedure, the observer becomes a deadbeat structure observer with guaranteed stability.

4. IC engine indicated torque estimation problem is explored. A nonlinear input observer is developed using a sliding mode methodology. Upcoming technologies in automotive IC engine (such as electronic throttle control, direction injection engine, etc.) requires an estimation of engine indicated torque. Hence, the estimation method examined in this study can be one of approaches to be considered for such application.

5. The location of HEGO sensor and the geometry of exhaust manifold of the engine under consideration bring a challenging problem. The HEGO sensor is located at far down stream from the exhaust valves and the geometry of exhaust manifold allows mixing of exhaust gases from each cylinder before it reaches the HEGO sensor. Although attempts are made to identify the delay, it was not possible to complete it in any sense. For complete identification of delay and exhaust mixing phenomenon, a thermal and fluid modeling and experiments is recommended and further simplification of the thermal-fluid model for control application is also recommended.

6. For IC engine application, one of obvious area to be explored extensively for future studies will be integration of different domain and different scale problems. Many processes in IC engine can be interpreted/utilized different ways by looking at different domains or scales. For example, some of the
phenomenon can be analyzed either in crank angle domain or in time domain. Some of them can be better described in discrete domain since they are discontinuous in nature. In some cases, even the same process is to be examined in different scale. For example, the dynamics in intake/exhaust manifold can be simplified in crank angle domain. A typical example of discrete process is fuel injection since fuel injector spray fuels for a finite time during one engine cycle. As an example of using different scaling will be problems involved in engine indicated torque. An indicated torque can be divided in DC component and AC component. For studies like longitudinal vehicle dynamics and anti-lock brake system, a DC component of indicated torque is enough and we can model the torque production process only with cycle-by-cycle variation. In contrast, for a study like engine misfire detection, an AC component is necessary and it can only be seen by looking at a resolution in crank angle within one engine cycle. Until recently, the amount of fuel injected is varied only cycle-by-cycle and so-called mean value engine model can be used for control design and analysis. However, new technologies such as direct injection and variable valve timing require a stratified fuel injection within one engine cycle. Accordingly, the new engine controllers should be design considering dynamics faster than mean value engine dynamics. Different scaling problems are expected to be increasing in near future. Therefore, control and diagnostics problems for IC engine should be examined to integrate the different domain and scaling problems.
BIBLIOGRAPHY


Jones, H. L., (1973) Failure detection in linear systems, Ph.D. dissertation, Department of Aeronautics & Astronautics, MIT.


Utkin, V. I., (1978) Sliding modes and their application in variable structure systems. Moscow, MIR.


Appendix A

Specification of FORD V12 4.6liter Engine, Sensors and Actuators

- Cylinder Displacement: 4.6L
- Number of Cylinder: 8
- Bore and Stroke: 90.2mm \times 90.0mm
- Throttle Shaft Diameter: 9.9mm
- Throttle Bore Diameter: 64.7mm
- 'Intake Manifold Volume: 01215 m³
- Firing Order: 1-3-7-2-6-5-4-8
- Valve Timing: Intake Valve Opening (12.1 BTDC)
  Intake Valve Closing (64.0 ATDC)
  Exhaust Valve Opening (63.1 BBDC)
  Exhaust Valve Closing (21.4 ATDC)
- Cycle Delay: 489°
- Mass Air Flow Rate Meter Calibration

\[ \dot{m} = -0.0629V^1 + 1.987V^1 + 0.1782V^2 + 5.9229V - 0.033 \]

where
\( \dot{m} \): mass flow rate of air [g/sec]
\( V \): measured voltage [V]

- Throttle Position Sensor Calibration

\[ \alpha = 21.3343V - 12.8229 \]

where
\( \alpha \): throttle position [degree]
\( V \): measured voltage [V]
- **ECT / ACT Sensor Calibration**

\[ T = 2.625V^4 - 24.5608V^3 + 83.0623V^2 - 142.9351V + 152.4436 \]

where

- \( T \): temperature [°C]
- \( V \): measured voltage [V]

- **Fuel Injector Pulse Width Calibration**

\[ PW = 1.1942V + 1.8883 \]

where

- \( PW \): pulse width [mmsec]
- \( V \): measured voltage [V]
Idle By Pass Air Calibration

\[ m_{\text{air}} = 43345V^3 - 4.2509V^2 - 8.016V - 0.1035 \, [\text{g/sec}] \]

where
\( m_{\text{air}} \): mass flow rate of air [g/sec]

\( V \): measured voltage [V]

Engine Speed Circuit Calibration

\[ \omega = C_1 \cdot V + C_2 \]

where
\( \omega \): engine speed [RPM]

\( C_1 \): slope \quad \( C_2 \): offset \quad \( V \): measured voltage [V]

200
Fuel Injector Calibration

\[ m_{f} = (PW - C_{i}) \cdot C_{2} \cdot C_{3}/1000 \]

where
- \( m_{f} \): injected fuel [gram]
- \( C_{i} \): slope
- \( C_{2} \): offset
- \( C_{3} \): unit conversion factor [lb to gram]
- \( PW \): fuel pulse width [mmsec]

UEGO Sensor Calibration

\[ A/F = C_{1} \cdot V / 5 + C_{2} \]

where
- \( A/F \): air to fuel ratio
- \( C_{1} \): slope (span for UEGO setting)
- \( C_{2} \): offset (zero for UEGO setting)
- \( V \): measured voltage [V]

201
Manifold Pressure Sensor Calibration

\[ P_m = P_{\text{atm}} - C \cdot V \]

where

- \( P_m \): manifold pressure [Pa]
- \( P_{\text{atm}} \): atmospheric pressure [Pa]
- \( C \): calibration constant
- \( V \): measured voltage [V]
Appendix B

Powertrain Simulator (PTSIM)

User Manual

For

Powertrain Simulator (PTSIM)
Getting Started

1) Execute MATLAB

2) From the MATLAB prompt change to the directory containing the powertrain model via the "cd" command (i.e. cd models).

3) Make sure you have copied the following files to that directory:
   (A) ptsim.mdl
   (B) ptpara.m

4) From the MATLAB prompt type "ptsim". A window should come up described as the "ptsim".

5) From the MATLAB prompt type "ptpara" to load the engine simulation parameters.

6) You are ready to begin!
Model Structure

In this section the mean value engine model as implemented in Simulink is discussed at the sub-model level. It is assumed that the reader has brought up the Simulink file "ptsim.mdl" at this point. Each sub-model is accessed by double clicking on it.

Further, the reader will be able to locate parameters for monitoring or plotting as necessary from information contained in this section. The reader is advised at this point to go through the ptsim.mdl file block by block to verify the individual inputs, outputs and independent variables. Figure B.1 shows the structure of the model in a block diagram form.

![Block Diagram](image)

Figure B.1: Block diagram representation of the powertrain model

A top-level simulink powertrain model containing four blocks (vehicle, sensors, controllers and actuators, we will call the 4 block as module) as shown in Figure B.2.
In what follows, each module (vehicle module, sensor module, controller module and actuator module) will be explained in detail.

B1. Vehicle Module

Vehicle module carries 8 blocks: throttle, intake air dynamics, pumping, fuel film dynamics, combustion, crankshaft dynamics, drive-train and vehicle, exhaust gas emissions. All inputs to this block come from the actuator modules, and all outputs are connected to the sensor modules.
B1.1 Throttle

Figure B.4: Throttle block

- inputs: throttle angle, manifold pressure
- outputs: throttle mass flow rate

a. Throttle Area Computation

Computes the projected open area of the throttle plate.
- inputs: throttle shaft diameter, throttle bore diameter, closed throttle angle, throttle angle
- outputs: throttle area

b. Throttle Mass Flow Computation

In this block, mass air flow rate through the throttle is computed based on the compressible flow equation.
- inputs: throttle area, throttle discharge coefficient, ambient stagnation pressure, manifold pressure, ideal gas constant for air, ambient stagnation temperature, specific heat ratio for intake air
- outputs: throttle mass flow rate

207
B1.2 Intake Air Dynamics

Conservation of mass and ideal gas laws are used to compute intake manifold pressure.

![Diagram of Intake manifold block]

- **inputs**: engine speed, throttle mass flow rate
- **outputs**: manifold pressure

B1.3 Pumping

In this block, mass airflow rate into the cylinders through the intake valves is computed based on the mean value approach utilizing volumetric efficiency.

![Diagram of Pumping block]

- **inputs**: manifold pressure, engine speed
- **outputs**: air flow rate to cylinder
B1.4 Fuel Film Dynamics

Evaporation/Condensation equations are used to compute gaseous fuel flow into the cylinder.

- inputs: fuel injected rate, engine speed
- outputs: fuel flow rate to cylinder, fuel mass in fuel film

B1.5 Combustion

Calculates engine torque as a function of mass flow rate of air entering cylinder, air fuel ratio, engine speed, EGR flow rate, and spark advance.

- inputs: air mass flow rate into cylinder, APR, engine speed, spark advance, EGR flow rate
- outputs: engine torque.
B1.6 Exhaust Gas Emissions

Flow rate of exhaust emission gases is calculated as a function of mass flow rate of air into cylinder and air fuel ratio.

- inputs: air mass flow rate to cylinder, air fuel ratio
- outputs: flow rate of exhaust emission gases (NO_x, CO, CO_2, HC, O_2)

B1.7 Crankshaft Dynamics

Newton's 2nd law is applied to a single-degree-of-freedom crankshaft model. Engine friction torque is modeled as a polynomial function of manifold pressure and engine speed. The ability to put a disturbance load from the power steering pump is also included in this block diagram.

- inputs: engine torque, pump torque, idle flag, power steering pump flag
- outputs: engine speed
B1.8 Drive-train and Vehicle Longitudinal Dynamics

This block contains torque converter, transmission, gear shift logic, and vehicle longitudinal dynamics.

![Diagram of Drive-train and Vehicle Longitudinal Dynamics Block](image)

Figure B.11: Drive-train and vehicle longitudinal dynamics block

- inputs: engine speed
- outputs: vehicle speed, pump torque (torque converter)

B1.9 Torque Converter

A static torque converter model is implemented in this block. The block calculates the pump torque and turbine torque as a function of pump speed and torque speed.

![Diagram of Torque Converter Block](image)

Figure B.12: Torque converter Block

- inputs: pump speed, turbine speed
- outputs: pump torque, turbine torque
B1.10 Transmission and Gear Shifts

Two different gear shift logics are available in this block. By selecting the manual switch, user can fix the gear ratio at a specific value. Otherwise, the up and down shift is selected by the engine speed defined in up and down speed block.

![Diagram of Transmission and Gear Shifts](image_url)

Figure B.13: Transmission and gear shift Block

- inputs: engine speed
- outputs: overall gear ratio

B1.11 Vehicle Longitudinal Dynamics

Vehicle longitudinal dynamics are modeled in this block. Aerodynamic drag and rolling resistance are considered as vehicle load. Wheel slip is ignored for modeling simplicity.

![Diagram of Vehicle Longitudinal Dynamics](image_url)

Figure B.14: Vehicle longitudinal dynamics block

- inputs: engine speed, turbine torque, gear ratio
- outputs: vehicle speed, turbine speed, pump speed

212
B2. Sensor Module

Two sensors are considered: HEGO sensor and UEGO sensor. Available variables are listed in the Section III. User can use a scope or XY-graph in Simulink to monitor variables during the simulation. For now, two scopes are included to monitor the engine speed and air fuel ratio (UEGO sensor output).

![Sensor Module Diagram]

Figure B.15: Sensor module

B2.1 HEGO Sensor Dynamics

Time delayed air fuel ratio is passed through a first order system, then pass through a switching filter to give a binary EGO output.

- inputs: air to fuel ratio from combustion block, engine speed
- outputs: EGO sensor binary output

B2.2 UEGO Sensor Dynamics

UEGO sensor dynamics are modeled as the first order system with a time constant defined in egpre.m file.

- inputs: air to fuel ratio from combustion block, engine speed
- outputs: UEGO sensor output (linear air fuel ratio)
B3. Controller Module

All available variables are input the controller module so that users can design their own control strategy using those variables. For now, the controller module has 7 outputs:

- On/Off command to power steering pump
- On/Off command to idle speed controller
- Throttle angle profile versus crank angle
- Idle bypass valve command
- EGR flow rate
- Spark Advance
- Fuel injection rate

Figure B.16: Controller module
B1.2 Air Fuel Ratio Controller

Both feedforward and feedback control is implemented. Feedforward fuel term is decided by air flow rate into cylinder and the stoichiometric ratio of the given fuel. The feedback consists of PI control which uses HEGO sensor measurement. Transient fuel compensation algorithm is also implemented assuming the fuel film mass is measurable.

- inputs: Air flow rate into cylinder, EGO sensor output, fuel film mass
- outputs: fuel flow rate

B1.3 Idle Speed Controller

To regulate the engine idle speed around set point, two PI controllers are used to determine the spark advance and idle by pass air. To simulate engine idle condition, user has to set the idle flag to 1. To apply disturbance load to engine (engine accesary load such as power steering pump), disturbance load model is also included.

- inputs: engine speed
- outputs: spark advance, idle by pass air

B4. Actuator Module

In case a more complex model of an actuator is required, users can include the actuator model in this module. For now, the input to each actuator model is directly connected to the output.
### Variables used in simulation

<table>
<thead>
<tr>
<th>Variable</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>afr</td>
<td>-</td>
<td>Air to fuel ratio in cylinder</td>
</tr>
<tr>
<td>afrm</td>
<td>-</td>
<td>Air to fuel ratio at exhaust (measured by UEGO)</td>
</tr>
<tr>
<td>alpha</td>
<td>rad</td>
<td>Throttle angle</td>
</tr>
<tr>
<td>CO</td>
<td>kg/sec</td>
<td>CO flow rate at exhaust</td>
</tr>
<tr>
<td>CO₂</td>
<td>kg/sec</td>
<td>CO₂ flow rate at exhaust</td>
</tr>
<tr>
<td>dma_c</td>
<td>kg/sec</td>
<td>Mass flow rate of air entering cylinder</td>
</tr>
<tr>
<td>dma_isc</td>
<td>kg/sec</td>
<td>Mass flow rate of air through idle by pass valve</td>
</tr>
<tr>
<td>dma_th</td>
<td>kg/sec</td>
<td>Mass flow rate of air at throttle</td>
</tr>
<tr>
<td>dm_egr</td>
<td>kg/sec</td>
<td>Mass flow rate of exhaust gas through egr valve</td>
</tr>
<tr>
<td>dmf_c</td>
<td>kg/sec</td>
<td>Mass flow rate of fuel entering cylinder</td>
</tr>
<tr>
<td>dmf_i</td>
<td>kg/sec</td>
<td>Mass flow rate of fuel through fuel injector</td>
</tr>
<tr>
<td>EGO</td>
<td>-</td>
<td>EGO sensor output</td>
</tr>
<tr>
<td>etav</td>
<td>-</td>
<td>Cylinder charge efficiency</td>
</tr>
<tr>
<td>far</td>
<td>-</td>
<td>Fuel to air ratio</td>
</tr>
<tr>
<td>Tf</td>
<td>Nm</td>
<td>Engine friction torque</td>
</tr>
<tr>
<td>gr</td>
<td>-</td>
<td>Gear ratio</td>
</tr>
<tr>
<td>HC</td>
<td>kg/sec</td>
<td>HC flow rate at exhaust</td>
</tr>
<tr>
<td>mff</td>
<td>kg</td>
<td>Mass of fuel in fuel film</td>
</tr>
<tr>
<td>NO₅</td>
<td>kg/sec</td>
<td>NO₅ flow rate at exhaust</td>
</tr>
<tr>
<td>O₂</td>
<td>kg/sec</td>
<td>O₂ flow rate at exhaust</td>
</tr>
<tr>
<td>omega</td>
<td>rad/sec</td>
<td>Engine speed</td>
</tr>
<tr>
<td>pm</td>
<td>Pa</td>
<td>Intake manifold pressure</td>
</tr>
<tr>
<td>SA</td>
<td>degree</td>
<td>Spark advance</td>
</tr>
<tr>
<td>t</td>
<td>second</td>
<td>Time</td>
</tr>
<tr>
<td>theta</td>
<td>rad</td>
<td>Crank angle</td>
</tr>
<tr>
<td>Tp</td>
<td>Nm</td>
<td>Pump torque</td>
</tr>
<tr>
<td>Te</td>
<td>Nm</td>
<td>Engine torque</td>
</tr>
<tr>
<td>Tt</td>
<td>Nm</td>
<td>Turbine torque</td>
</tr>
<tr>
<td>Vspd</td>
<td>m/sec</td>
<td>Vehicle speed</td>
</tr>
<tr>
<td>wp</td>
<td>rad/sec</td>
<td>Pump speed</td>
</tr>
<tr>
<td>wt</td>
<td>rad/sec</td>
<td>Turbine speed</td>
</tr>
</tbody>
</table>
PTPARA.M

%===============================================================================
% PTPARA  Initialize the simulation model and its parameters.
%
%
%-----------------------------------------------------------------------------
clear

% ... Constants
d2r=pi/180; % Conversion from Degrees to Radians
R2w=2*pi/60; % Conversion from RPM to Radians/sec
Ra=8314.3/28.846; % Ideal Gas Constant for Air
Stoich=14.6; % Stoichiometric Air/Fuel Ratio for Gasoline (Heywood)
Ga=1.40; % Specific Heat Ratio for Air at 295K [dimless.]

% ... throttle
d=0.009931; % throttle shaft diameter [m]
D=0.064719; % throttle bore diameter [m]
a=d/D;
alpo=6.18*d2r; % throttle when closed [rad]
C_d=0.567; % throttle discharge coefficient
P_crit=(2/(Ga+1))^((Ga/(Ga-1))); %Critical Pressure Ratio for Air

% ... for Intake Dynamics
Pa=1.01325e5; % One Atmosphere [N/m^2]
Ta=295; % Atmospheric Temperature [K]
V_m=0.0029; % Intake Manifold Volume [m^3]
T_m=295; % Intake Manifold Mean Operating Temperature [K]
V_d=0.0046; % Displacement Volume [m^3] for all 8 cylinder

% ... coeff. for Cylinder Charge Efficiency
vc1=2.5574e-01;
vc2=2.0844e-03;
vc3=-2.2649e-06;
vc4=3.4471e-06;

% ... for INERTIAL DYNAMIC
J=0.0843; % Engine Rotational Inertia [Nms^2]
B_damp=0.592; % Engine Damping Constant [Nms]
t_d=pi; % Delay of torque production

% ... for Road Load
rad_wheel=.3175; % Vehicle wheel radius [m]
C_roll=0.007; % Coefficient of rolling resistance [dimensionless]
C_drag=0.34; % Coefficient of drag [dimensionless]
A_veh=2.26; % Frontal Area of Vehicle [m^2]
M_v=2154.6; % Vehicle mass including passengers and fuel[kg]

% ... for transmission and axle
N_t(1)=2.4; % 1st gear ratio
N_t(2)=1.47; % 2nd gear ratio
N_t(3)=1; % 3rd gear ratio
N_t(4)=0.67; % 4th gear ratio
N_f=2.73; % final gear ratio
N_g=N_t*N_f;

217
tauf=0.25; % time constant for fuel dynamic [sec]
X=0.6; % faction of fuel to be condensed

Kinj=0.002;
L_o=0.2;

tau_uego=0.0434; % time constant for UEGO sensor [sec]

tde=8*(pi/2);

bf1=4.8814e+1;
bf2=9.6390e-3;
bf3=1.6151e-4;
bf4=-3.8683e-1;
bf5=7.7083e-4;

P_NOx=[1.0762e-06 -7.6086e-05 2.0784e-03 -2.7399e-02 1.7477e-01 -4.3306e-01];
P_CO=[1.6158e-05 -1.2918e-03 4.0420e-02 -6.1497e-01 4.5028e+00 -1.2457e+01];
P_HC=[9.8468e-07 -7.8574e-05 2.3061e-03 -2.9647e-02 1.4190e-01];
P_CO2=[3.1871e-05 -1.7442e-03 3.1292e-02 -1.8540e-01 1.9685e-01];
P_O2=[-1.8399e-05 1.0623e-03 -2.1646e-02 1.8755e-01 -5.8855e-01];

d_th=pi/2;
r_theta=[0 3 4 7 8 10]*100*d_th;
r_alpha=[1 1 2 2 1 1]*5;

f_theta=max(r_theta); nosamp=f_theta/d_th;
th=[0:(nosamp-1)]*d_th;

icflag=r_alpha(1);
if icflag==0
  % ... initial value for variables (throttle = 0 deg; idle)
tq_eng=40.5185;
ppm=9.6623e+03;
omega=535.3532*R2w;
mff=1.1724e-06;
dma_c0=1.1625e-04;
Vspd0=omega0/Ng(1)*rad_wheel;
elseif icflag==5
  % ... initial value for variables (throttle = 5 deg)
tq_eng=53.5325;
ppm=3.7898e+04;
omega=742.9*R2w;
mff=8.7441e-06;
dma_c0=8.4856e-04;

218
Vspd0=3.5;
else if icflag==10
  % ... initial value for variables (throttle = 10 deg)
tq_eng0=73.8083;
pm0=5.2029e+04;
omega0=1064.6*R2w;
mff0=2.0523e-05;
dma_c0=0.002;
Vspd0=5.5;
else if icflag==15
  % ... initial value for variables (throttle = 15 deg)
tq_eng0=101.0017;
pm0=5.5872e+04;
omega0=1493.6*R2w;
mff0=3.6336e-05;
dma_c0=0.0035;
Vspd0=omega0/Ng(1)*rad_wheel;
else if icflag==20
  % ... initial value for variables (throttle = 20 deg)
tq_eng0=134.8407;
pm0=5.9691e+04;
omega0=2022.7*R2w;
mff0=5.6026e-05;
dma_c0=0.0055;
Vspd0=omega0/Ng(1)*rad_wheel;
else if icflag==25
  % ... initial value for variables (throttle = 25 deg)
tq_eng0=175.4378;
pm0=5.9908e+04;
omega0=2651.2*R2w;
mff0=7.9667e-5;
dma_c0=0.0078;
Vspd0=omega0/Ng(1)*rad_wheel;
else if icflag==30
  % ... initial value for variables (throttle = 30 deg)
tq_eng0=222.3295;
pm0=5.9914e+04;
omega0=3368.8*R2w;
mff0=1.0701e-4;
dma_c0=0.0104;
Vspd0=omega0/Ng(1)*rad_wheel;
else
  clc
  msg=['\n\n*** Initial condition is not defined, please find and specify\n'];
  fprintf(msg);
end

% ... CONTROLLER - air fuel ratio controller parameter
Pgain=1e-6;
Igain=1e-8;
**Tutorial - Simulation**

Simulation of the vehicle's response to a change in the throttle angle as prescribed below. As an input, define throttle angle versus crank angle in the egpre.m file as follows:

Crank angle vector

\[ r_{\text{theta}} = [0 \ 4 \ 5 \ 10 \ 11 \ 15] \times 100 \times d_{\text{th}}; \]

Throttle angle vector

\[ r_{\text{alpha}} = [5 \ 5 \ 10 \ 10 \ 5 \ 5]; \]

To see the throttle trajectory against crank angle, plot in Matlab as follows:

```matlab
plot(r_theta, r_alpha)
```

You will get a figure shown in Figure B.18.

![Throttle trajectory graph](image)

**Figure B.18: Throttle opening trajectory**

Now that the Throttle Angle Input Profile has been specified, it is time to run the simulation. To run the simulation, click on “Start” under the “Simulation” menu located in the title bar of any of the windows of the ptsim simulation.
The running “Clock” of the simulation can be opened to view the simulation’s progress, by double-clicking on any one of the clocks in any part of the simulation. This simulation has been developed in the Crank Angle domain so the clock displays the elapsed crank angle radians, not time. The time (in seconds) is calculated and stored in the variable \( t \).

*Tip: The simulation will run faster if the clock is not open.*

Once the simulation has completed running, plots can be created by entering commands from the Matlab window.

The following commands can be issued at the Matlab prompt:

```matlab
» plot(theta,alpha)
```

![Throttle opening trajectory](image)

*Figure B.19: Throttle opening trajectory*
Figure B.20: AFR response during throttle tip-in and tip-out

Figure B.21: Engine speed during throttle tip-in and tip-out