OBSERVER BASED PARAMETER ESTIMATION WITH APPLICATIONS TO MODEL BASED DIAGNOSTICS

A Thesis

Presented to

the Faculty of the Department of Mechanical Engineering

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Masters of Science

in Mechanical Engineering

by

Ankit Kukreja

December, 2012

OBSERVER BASED PARAMETER ESTIMATION WITH APPLICATIONS TO MODEL BASED DIAGNOSTICS

Ankit Kukreja

Approved:

Chair of the Committee,
Dr. Matthew Franchek, Professor,
Mechanical Engineering

Committee Members:

Dr. Karolos Grigoriadis, Professor,
Mechanical Engineering

Dr. Vemuri Balakotaiah, Professor,
Chemical Engineering

Dr. Suresh K. Khator, Associate Dean,
Cullen College of Engineering

Dr. Pradeep Sharma,
Professor and Chair,
Mechanical Engineering

Acknowledgements

I would like to take this opportunity to offer my heartfelt thanks to all those individuals without whom this work would not have been possible. Firstly, I am profoundly grateful to my advisor, Dr. Matthew Franchek, for his continuous guidance and motivation throughout the course of my work. It was an honor to work with him. I would like to thank Dr. Karolos Grigoriadis and Dr. Vemuri Balakotaiah for serving on my committe. I would also like to thank Cummins Inc. for financial support for the research work, and I am highly grateful to Daniel Smith of Cummins Inc. for working so closely on previous projects with us. I would also like to thank the staff of the department of Mechanical Engineering for their support.

My thanks are given to my friends and colleagues who made my experience at the University of Houston memorable. I would like to thank my lab mates Rafik, Rohit, Chris, Oliver, Fatemah and Nassim for their constant support. I would like to take the opportunity to mention Aanchal, Aniket, Nikhil, Ankit and all my friends who made my life as a graduate student fun and enjoyable. No acknowledgement could possibly be complete without thanking my parents and my sister for their affection and support throughout my endeavors.

OBSERVER BASED PARAMETER ESTIMATION WITH APPLICATIONS TO MODEL BASED DIAGNOSTICS

An Abstract

of a

Thesis

Presented to

the Faculty of the Department of Mechanical Engineering

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Masters of Science

in Mechanical Engineering

by

Ankit Kukreja

December, 2012

Abstract

Presented in this manuscript is an observer-based method for consistent parameter estimation and the utility of these estimates in real-time model based diagnostics. Parameter estimation is realized through a calculation involving the state estimates produced from the combination of two infinite time Luenberger observers with a delay. The significance of this approach is the ability of the coupled observers to have their state estimates converge in finite time to the true state values. Unique to this formulation is that consistent parameter estimates can be obtained using data limited to one sampling event provided the converged estimated states at that sampling interval meet a full rank condition similar to persistancy of excitation. The observer based parameter estimation method is then integrated with Information Synthesis to create a one-step-ahead fault detection, isolation and estimation knowledge base. The utility of the proposed method is demonstrated on a numerical example and an experimental application of engine onboard sensor diagnostics.

Table of Contents

List of Figures

List of Tables

Notations

A	System matrix in state space representation
B	Control matrix in state space representation
C	Output matrix in state space representation
D	Feed-forward matrix in state space representation
a_k	Model coefficients associated with output delay
b_k	Model coefficients associated with input delay
$Z[]$	Z-Transform
r_e	Relative degree
n	Model numerator order
m	Model denominator order
$u(k)$	Input of the object at the time samples $k = 0, 1, 2, ...,$
$y(k)$	Output of the object
$x(k)$	Vector of states of the object
$\xi(k)$	A discrete sequence
μ	A certain positive number
τ	Time constant
G	Transfer function
β	Numerator polynomial representation
α	Denominator polynomial representation
$U(z)$	Input of the objectin z-domain
$Y(z)$	Output of the object in z-domain
$u(t)$	Input of the object in continuous time
$y(t)$	Output of the object in continuous time

$x(t)$	Vector of states of the object in continuous time
$F_e(t)$	System matrix in continuous time
$G_u(t)$	Control matrix in continuous time
$G_\omega(t)$	Matrix associated with disturbance in continuous time
$O(t)$	Output matrix in continuous time
ν	Measurement noise in the system
q	Dimension of process noise array
p	Dimension of measurement noise array
ω	Input disturbance
R	Power spectral density matrix
Q	Power spectral density matrix
δ	Dirac delta function
Γ_ω	Matrix associated with disturbance in discrete time
I	Linear-in-the-parameters discrete system
ϑ	Zero mean white noise
φ	Regression vector
θ	Model paramters
S	Unknown model parameter vector
T	Model numerator vector
ε	Model prediction error
N	Number of data points
ϵ	Model prediction error matrix
V	Least squares minimization function
Y	Regressed variable vector
Φ	Regression vector

Ω	Regression matrix
θ_0	True parameter vector
σ	Standard deviation
χ	Unbiased estimate of variance
P	Unbiased estimate of variance
λ	Forgetting factor
W	Covariance matrix
L_e	Linear Transformation of Y to Φ
W_θ	Covariance matrix of parameters
$E[]$	Expectation
w	Elements of weight matrix
ξ	Instrumental variable
Z	Instrumental variable matrix
Fi	Input and output discrete filter
Υ	A discrete filter
H	Healthy model parameters
H_a	Adapted model parameters
h	Healthy model coefficient vector element
h_a	Adapted model coefficient vector element
R_e	Coefficient error vector or residual
e_j	Unit vector along the j^{th} axis
r_j	j^{th} component of R_e
F	Fault vector
γ	Fault signature code
L_{mag}	Fault signature magnitude

k	Spring constant
ma	Mass
b	Damping coefficient
F_{ex}	External force applied on the system
A_o	Nominal system matrix
C_o	Nominal output matrix
K_e	Observer gain matrix
f	Control input
E	Matrix associated with updated states
Z_k	Observer design parameter in [75]
d	Delay
κ	Radius of unit circle in z-domain

Chapter 1
INTRODUCTION AND OVERVIEW

1.1 Real Time Fault Detection, Isolation and Estimation

This research attempts to focus primarily on developing a robust fault detection, isolation and estimation (FDIE) methodology that can be applied for On-Board Diagnostics (OBD) of engineering systems. The FDIE approach presented in this manuscript is based on observer based method. The parameters of the model of the system under consideration are estimated on-line using observers and system identification methods and compared with healthy system parameters to detect, identify and estimate the size of a fault.

1.1.1 Observer Based Diagnostics

State observers have been widely used in many branches of science and engineering. According to the system characteristics, observers fall into two categories: Luenberger observers for deterministic systems [17, 24, 25, 37, 32] and Kalman filters for stochastic systems [38]. The former is limited to systems with accurate models neglecting both internal and external uncertainties; the latter considers external uncertainties as white noises. Unfortunately, modeling errors often lead to a poor performance in Kalman filtering [38], which makes the Luenberger observer more popular in deterministic system research.

Although the observer parameter estimation and adaptation theories have become rich, most of the existing methods are confined to the traditional (static) Luenberger structure. However, the main concern is the pole positions [30]. Here, the term static observer is used to denote the classic Luenberger observer, where a constant gain is used in the feedback path. From the viewpoint of frequency domain, such a

feedback gain does not have the ability of frequency shaping and puts some limitation on disturbance attenuation. It is of interest to change the observer structure and see how this would improve the parameter estimation performance. Thus, it is natural to introduce additional dynamics into the observers.

1.1.2 Motivation For The Present Study

Successful examples of identifying constant or slowly time-varying parameters have been reported in Bastin et al. (1988). The main principle is to combine integrators into a Luenberger observer [37] in order to identify constant parameters while estimating system states. Many real-world applications, however, involve time-varying parameters acting as disturbances and/or uncertainties. Estimation of constant parameters may not be sufficient for these applications. It is highly desirable to identify time-varying parameters for the purpose of feedback control and many other applications. However, most of the observers designed to do the task guarantee only that the estimation error tends to zero asymptotically. In principle, it is desirable to have observers that allow reconstructing (in the nominal case) the system state in finite time. So far there exist only a couple of observers with a finite time convergence. Examples are sliding mode based observers and moving horizon based observers. The sliding mode based observers can be applied to linear and nonlinear systems and guarantee finite time convergence of the estimation error. However, the differential equations describing the estimation error can only be rendered semi-globally stable in the sense that a bounded region of system states must be considered. Moving horizon observers are based on the online solution of a dynamic optimization problem involving the output measurements and the system model. They can be applied to linear as well as nonlinear systems. Under the assumption that the global solution to the dynamic optimization problem can be found and that a certain finite time observability assumption holds, moving horizon observers can in principle guarantee

finite time convergence of the estimation error. The main drawback of moving horizon observers is that a dynamic optimization problem must be solved online. The method applied for time-invariant systems [74] is simpler to implement as compared to moving horizon and sliding mode observer designs. Basically two observers with different speeds of convergence are used and the state estimate is constructed based on delayed and current state estimates of the identity observers. For this purpose, conditions for the (time-varying) observer feedback matrices and the delay used will lead to guaranteed finite time convergence of the observer error. However, the system is used for the purpose of continuous time parameter estimation.

Therefore, less attention has been focused on the observer problem for discrete-time systems. It was shown in [16] that certain properties, like observer error linearizability [17], are not inherited from the underlying continuous-time system. Moreover, the class of continuous-time systems that admit approximate solutions to the observer error linearization problem for their exact discretizations with sampling time T in an open interval is limited to the class of nonlinear systems that are approximately state-equivalent to a linear system and hence is very restricted. These results motivated the search for a structurally more robust approach to the observer problem for discrete systems.

1.1.3 Research Objectives

The objective of the current research work is to develop a *robust, systematic online model-based diagnostics methodology for fault detection, isolation and estimation* that uses observers for model parameter estimation. This requires the following tasks to be accomplished

- Development of online nominal models of the system.
- Estimation of model parameters using observer based method.
- Development of a robust method for performing fault detection, isolation and

estimation.

- Development of a method to quantify the sensitivity of the fault detection and diagnostics towards noise and disturbances.
- Validation of the fault detection and diagnostics method on a numerical example and experimental data.

1.2 Document Organization

The outline of this thesis is as follows. A part of chapter 2 presents the relevant literature survey of various methods in use for fault detection and diagnosis. The other part presents the relevant literature review of various approaches for parameter estimation using observers. Background on relevant concepts is also discussed at the end of chapter 2. Chapter 3 explains the various system identification techniques used for fault estimation. Consequently, it gives the concept of Information Synthesis (IS) for FDIE. In the end, validation of IS is shown to demonstrate the methodology. Chapter 4 presents the proposed observer based estimation method and validates it against an experimental sensor fault diagnostics data.

Chapter 2
LITERATURE REVIEW AND OBSERVER PRIMER

2.1 Introduction

In this chapter the literature related to fault detection and diagnosis of engine subsystems and observer based parameter estimation techniques is reviewed. For the improvement of reliability, safety and efficiency advanced methods of supervision, fault detection and fault diagnosis become increasingly important for many technical processes. This holds especially for safety related processes like aircraft, trains, automobiles, power plants and chemical plants. The classical approaches are limit or trend checking of some measurable output variables. Because they do not give a deeper insight and usually do not allow a fault diagnosis, model-based methods of fault detection were developed by using input and output signals and applying dynamic process models. These methods are based, e.g., on parameter estimation, parity equations or state observers.

An observer, which is driven by measurable system outputs and inputs, can achieve state reconstruction for the purpose of control applications when the entire state vector is not available for the use of feedback. To perform the twin tasks of state estimation and parameter identification, observers have been extensively studied since the 1970s. Various approaches taken so far for the methodologies adopted for observer based estimation are discussed in upcoming sections.

2.2 Literature Review

FDIE has received considerable attention in the past three decades and an extensive literature base exists concerning its various methods. Fruitful results can be found in several survey papers [3, 39, 40] and books [1]. In general, there are several kinds of schemes for model-based FDI: observer based, [30, 19, 15], parity space based,

[41], eigenstructure assignment based and parameter-identification based [3, 39]. In some cases, fault accommodation strategies are needed, i.e., the control algorithm must be adapted based on FDI to go on controlling the faulty system. There are essentially two FDIE categories: model-free and model-based methods. Model-free FDIE approaches consist of limit checking, special sensors, multiple sensors, frequency analysis, and expert systems. Model-based FDIE methods [3, 2] consist of parameter estimation, state estimation and parity equations. For state estimation solutions, the changes in model states are utilized to generate residuals. For parity equation solutions, an analytic model can be used to estimate a sensor output based on other sensor information. The estimated sensor output can be compared to the actual sensor output to generate a residual. These residuals are then processed to diagnose faults. In [9], detection and diagnosis of faults in automotive engines is done using a nonlinear statistical process control showing that introducing the statistical local approach into nonlinear statistical process control produces statistics that follow a normal distribution, thereby enabling a simple statistical inference for fault detection. Most model-based FDIE approaches do not utilize the size of the residuals beyond comparison to a threshold. In other words, most model-based FDIE approaches do not attempt fault estimation. Furthermore, the residuals generated are a result of combined faults, noise and modeling errors. Sensitivity and robustness properties result from these and isolability properties are mainly a result of modeling errors.

A real-time parameter identification approach is presented in [13] for diagnosing low-flow and high-flow faults in the exhaust gas recirculation system. Real-time diagnostics is achieved using a more robust version of the recursive-least-squares (RLS) method referred to as the recursive total-least-squares method. The method is used to identify the coefficients in a static orifice model of EGR valve. Literature [12] presents a method to locate and quantify engine faults due to ageing of modern diesel engines in order to help to reduce design tolerance. The sensor signals are compared with the models in order to detect a possible fault, and fault location and size are estimated

by means of a bank of extended Kalman filters. Overall efforts made in academia and industry on fault detection and isolation in automotive engine systems for a variety of component faults, actuator faults and sensor faults using various data-driven and model-based methods is presented in literature [13]. Current model based techniques like parameter estimation methods, observer-based methods and parity relations have been explained in detail.

The commonalities among these works include the generation of a residual and the processing of this residual for fault detection and isolation. For each study, a detailed dynamic model was identified so that a residual could be generated. However, the family of faults considered in these works appears during steady state operation. This simplification will be used while seeking a unifying approach to hardware faults that occur in subsystems of different applications.

The principle of *least squares* (LS), which was first invented independently by a few scientists and mathematicians such as C.F. Gauss, A.M. Legendre and R. Adrain[55, 54], is a classic approach to obtain the optimal solution of an overdetermined system to minimize the sum of squared residuals. The most popular and important interpretation of the LS approach is from the application in data fitting. That is, the best fitting in the LS sense is to approximate a set of parameters (estimands) such that the sum of squared residuals is minimized, where the residuals are differences between the measured observation values and the corresponding fitted ones. In addition, the LS problem also has various names in different disciplines [54]. For instance, in some mathematical areas, LS may be treated as a special *minimum* L_2*-norm* problem[56]. In statistics, it is also formulated as a probabilistic problem widely used in *regression analysis or correlation analysis*[57, 58]. In engineering, it is a powerful tool adopted in *parameter estimation, filtering, system identification*, and so on [59, 60]. In particular, in the area of estimation, the LS formulation can be derived from the maximum-likelihood (ML) criterion if the observation errors are normally distributed. The LS estimator can also be treated as a moment estimator.

Roughly speaking, LS problems can be classified into linear and nonlinear cases, depending on whether the involved observation quantities are linear functions of the estimand or not. It is also well known that, with linearization techniques such as Gauss-Newton methods, a nonlinear LS problem may be converted to linearized iterative refinements. More generally, the object of data fitting may be extended to minimize the sum of the weighted residual squares, which leads to the definition of least squares with weights. According to the features of their weighting matrices, linear LS problems with weights can be largely categorized from the simple to the complex as LS, weighted LS (WLS), and generalized LS (GLS), where linear-equality (LE) constraints may be imposed [61].

The above methods can be implemented recursively [62]. That is, the current solution can be obtained by updating the previously-processed one (using old observation data) with new observations. The RLS is particularly suitable for real-time applications since sequential algebraic operations at each cycle require low computation as well as fixed storage [63]. It has been widely applied to such areas as signal processing, control and communication [64]. In adaptive-fitering applications, "RLS algorithms" are even referred to RLS-based algorithms for problems with fading-memory weights [65]. As a normal-equation method, a major disadvantage of RLS methods is that they have relatively poor numerical stability, compared with direct observation-function coefficient factorization methods. Fortunately, recursive QR decomposition methods, such as Givens rotations, can be combined to improve numerical stability [66, 67, 68].

The need to study state estimators dynamical systems from a control (observers) point of view is well understood by now. For the class of finite-dimensional, time invariant linear systems, a solution to the observer problem has been known since the mid 1960s: the observer incorporates a copy of the system and uses output injection to achieve an exponentially decaying error dynamics. For the class of continuous-time nonlinear systems, the reader is referred to [15, 18, 21] and the references therein for a summary of the theory up to 1986. More recent developments include the work of

Krener et al. [42] on higher-order approximations for achieving a linearizable error dynamics. Tsinias in [43] has proposed (nonconstructive) existence theorems on nonlinear observers via Lyapunov techniques. Gauthier et al. [44] and Deza et al. [45] show how to construct high-gain, extended Luenberger- and Kalman-type observers for a class of nonlinear continuous-time systems. Tomamb [46] and Nicosia et al. [47] have proposed a continuous-time version of Newtons algorithm as a method for computing the inverse kinematics of robots; moreover, the latter paper also presents a symbiotic relationship in general between asymptotic observers and nonlinear map inversion. Finally, Bastin and Gevers [49] have designed an adaptive observer for time-varying nonlinear systems that can be converted to certain observer canonical form.

The study of online observers is traced back to the joint state-parameter estimation in adaptive control systems [23]. On one hand, the unmeasured states are evaluated for the purpose of state feedback control; on the other hand, the unknown parameters are estimated online so that the controller can be updated accordingly. When certain persistent excitation requirements are satisfied, the estimated states and parameters converge to their real values simultaneously. With the increasing reliability requirements on control systems, observers have also been applied to fault diagnosis and evaluation, where faults are modeled and evaluated as the deviation of parameters from their nominal values. In the framework of stochastic problem formulation, a joint state-parameter estimation algorithm based on a modified Extended Kalman filter (EKF), a two-stage Kalman filter, has been developed for fault diagnosis and fault tolerant control [48] applications.

The approaches of observers normally fall into two categories: (1) The unknown parameters are taken as extra states of the system; a state observer is then constructed to estimate all extended states including the unknown parameters [49, 32]. (2) Parameters estimation techniques are applied first using the input-output data;

the states of the system are then estimated with the updated information of parameters [35, 36]. A general comparison of different methods using observers is shown in [50] with a unified approach.

One concern regarding observers is the robustness to disturbances such as measurement noises, modeling inaccuracies, and unknown external inputs while providing the desired accurate estimation for both states and parameters. In [48], the performance of observers is discussed for the noise corrupted systems. It is stated in [48] that the expectation of the estimation errors is bounded if the magnitude of noises is bounded. Furthermore, the expectation of estimation errors converges to zero for systems with independent noises of zero means. Therefore, an observer is, as shown in [51], Bounded-Input Bounded-Output (BIBO) stable to unknown external signals, which means the estimation of both parameters and states are affected by disturbance inputs such that persistent estimation errors exist as functions of the disturbance. Since the disturbance is unknown and consequently the size of these errors is also unknown to the system, the accuracy and the reliability of the estimation are impaired.

For a Multiple-Input Multiple-Output (MIMO) system, the robustness to structured disturbances can be further enhanced for an observer by utilizing the measurement redundancy of the system. With proper measurements, the influence of disturbances can be decoupled from the estimation errors. One technique is to incorporate an Unknown Input Observer (UIO). A UIO [52] is a disturbance-decoupled observer for the accurate state estimation for a system with unknown disturbance inputs. In [52], a UIO has been designed to estimate the faulty parameters of an aircraft actuator. The approach, however, needs full state (n independent) measurements and hence reduces the necessity of states estimation.

Joint estimation of states and parameters in state-space systems is of practical importance for fault diagnosis and for fault tolerant control. These results are essentially for linear time invariant (LTI) systems, though some of them have been

presented for nonlinear systems that can be exactly linearized by coordinate change and output injection. More recently, observers for multi-input-multi-output (MIMO) linear time-varying (LTV) systems have been developed in [53]. Some results on nonlinear systems have also been reported in [51]. In these cited references, the unknown parameters are all assumed to be located in state equations. As a matter of fact, there is an extra difficulty for estimating parameters in output equations, since the output feedback term used in most observers then depends on unknown parameters in this case. Recently, observers for unknown parameters in output equations have been studied, both for linear systems [41, 39, 33] and for nonlinear systems [42, 44, 45]. Nevertheless, all these works consider unknown parameters either in state equations or in output equations. Until now, no globally convergent algorithm has been developed for joint estimation of states and parameters in both state and output equations. A part of [71] focuses on parameter estimation in dynamic systems with bounded time-varying parameters, so that a fault in the plant can be detected by checking the parameter variation range. A high gain observer and finite state machine based identification method is used for the purpose. As a model reference identification technique, the observer technique is used for parameter estimation. The high-gain observer shows its ability to identify the model parameter perturbations under the corruption of bounded process and measurement noises. A very similar approach in [74] for continuous time systems has been used in [75] for discrete time systems to design a finite-time functional observer. A finite-time observer in [75] comprises two rth-order infinite time functional observers and a finite delay, where r is the dimension of output estimated of linear discrete-time descriptor systems. This ensures estimated state convergence in finite time exactly.

2.3 Background

2.3.1 Observer Based Estimation

The theory of observers for discrete systems is similar to the theory of observers for continuous systems although it differs by certain specific features. In the theory of discrete observation of the state of stationary linear objects we deal with regressive models

$$y(n+k)+a_n y(n+k-1)+...+a_1 y(k)=b_{m+1}u(m+k)+...+b_1 u(k), \qquad (2.1)$$

or with dynamical models in the space of states

$$x(k+1)=Ax(k)+Bu(k), y(k)=Cx(k), \qquad (2.2)$$

where $u(k)$ is the input of the object at the time samples $k=0,1,2,...$ and $y(k)$ is the output of the object, $x(k)$ is the phase vector or the vector of state of the object from $\mathbb{R}; (a_1,...,a_n),(b_1,...,b_{m+1})$ in (2.1) and A,B,C in (2.2) are parameters of the object, scalar and matrix respectively. Here n is the order of the object, $r_e=n-m$ is the relative order of the object, for physically realizable models. Satisfying cause and effect relations we always have $r_e \geq 1$. Generally speaking, models (2.1) and (2.2) are not equivalent since (2.1) does not always imply (2.2).

Every model for the defined input sequence $u(k) \in \mathbb{R}^m$ and the arbitrary initial state

$$(u(n-1),u(n-2),...,u(0)) \quad and \quad x(0) \in \mathbb{R}^n, \qquad (2.3)$$

generates a unique solution, namely, a discrete sequence of the output $y(k) \in \mathbb{R}^p$ or the state $x(k) \in \mathbb{R}^n$ where $k \in \mathbb{I}_+$ respectively.

The discrete sequence $\varrho(k)$ which grows not faster than the degree of a certain positive number μ, i.e. $|\varrho(k)| \leq \mu^k$, $k=0,1,2,...$, can be associated with the function of the complex variable $\varrho(z)$, $z \in \mathbb{C}$, by means of the Z-*transformation* via the

expression

$$\varrho(z) = Z[\varrho(k)] = \sum_{k=0}^{\infty} \varrho(k) z^{-k} for |z| > \mu. \tag{2.4}$$

After the substitution $z = e^{\tau s}$, $\tau = \text{const} > 0$, the Z-transformation passes into the so-called *Laplace discrete transformation*. It can be immediately seen that from this formula for obtaining the inverse Z^{-1} transformation, it suffices to decompose the function $\varrho(z)$ according to the degrees z^{-k}. The coefficients in these degrees form the required sequence $\varrho(k)$.

By means of the Z-transformation the recurrence equation which connects the input and output of the object reduces to an algebraic equation. For equation (2.1) under zero initial conditions we have a relation

$$Y(z) = G(z)U(z), \tag{2.5}$$

where $Y(z) = G(z)U(z)$, $U(z) = Z[u_k]$, and $G(z)$ is a transfer function of the object in the form of a ratio of two polynomials

$$G(z) = \frac{\beta_m(z)}{\alpha_n(z)} = \frac{(b_{m+1}z^m + ... + b_1)}{(z^n + a_n z^{n-1} + ... + a_1)}. \tag{2.6}$$

By analogy, the transfer function is defined for an object given in the space of states. From (2.2) under the zero initial condition $x(0) = 0$ we have

$$G(z) = C(zI - A)^{-1}B. \tag{2.7}$$

For the so-called scalar object the output y and input u of the object are scalars and its transfer function is a scalar function of the complex variable z. Otherwise, $G(z)$ is a matrix transfer function.

From the scalar object formula (2.5) makes it possible to find a simple transition from (2.1) to (2.2). Indeed, from the relation

$$G(z) = \frac{\beta(z)}{\alpha(z)}, \tag{2.8}$$

and equation (2.5), we can obtain a relation

$$\frac{Y(z)}{\beta(z)} = \frac{U(z)}{\alpha(z)}. \tag{2.9}$$

Employing this relation, we can introduce a scalar variable $x^1(k)$ using the formula

$$X^1(z) = \frac{Y(z)}{\beta(z)} = \frac{U(z)}{\alpha(z)}, \tag{2.10}$$

where $X^1(z) = Z[x^1_k]$. The latter is equivalent to the two equations

$$\begin{aligned} \alpha(z)X^1(z) &= U(z), & (2.11)\\ Y(z) &= \beta(z)X^1(z). & (2.12) \end{aligned}$$

Using now the inverse transformation Z^{-1}, we can pass from (2.11) and (2.12) first to the regressive equations

$$\begin{aligned} x^1(n+k) + a_n x^1(n+k-1) + ... + a_1 x^1(k) = u(k),\\ y(k) = b_{m+1}x^1(m+k) + ... + b_1 x^1(k), \end{aligned} \tag{2.13}$$

and then, with the use of the new state variables $x^1, x^2, x^3, ..., x^n$ related as

$$\begin{aligned} x^1(k+1) &= x^2(k),\\ x^2(k+1) &= x^3(k),\\ &\vdots\\ x(k+1)^{n-1} &= x^n(k), \end{aligned} \tag{2.14}$$

to equations in the space of states

$$\begin{aligned} x^n(k+1) = -\sum_{i=1}^{n} a_i x^i(k) + u(k),\\ y(k) = \sum_{i=1}^{n} b_i x^i(k), b_i = 0 \quad for \quad i > m+1. \end{aligned} \tag{2.15}$$

The use of vector-matrix notation

$$
\begin{aligned}
A &= \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -a_1 & -a_2 & -a_3 & \dots & -a_n \end{pmatrix}, B = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}, \\
C &= (b_1, b_2, ..., b_{m+1}, 0, ..., 0),
\end{aligned} \tag{2.16}
$$

makes the form of equation (2.9) identical to that of equation (2.2). For this reason, it makes sense to consider the regressive models and define them in the space of states.

2.3.2 True System

Figure 2.1: True System.

For dynamic systems described by a finite number of ordinary differential equations, the state of the dynamic system can be represented as a vector $x(t)$ in continuous-time and $x(k)$ in discrete-time. In either case, $x \in \mathbb{R}^n$. The vector representation of the state allows numerous tools from linear algebra to be applied to the analysis and design of state estimation and control system. The state space model in continuous

time representation is the following:

$$\dot{x}(t) = F_e(t)x(t) + G_u(t)u(t) + G_\omega(t)\omega(t), \tag{2.17}$$

$$y(t) = O(t)x(t) + \nu(t). \tag{2.18}$$

The $u \in \mathbb{R}^m$ is the deterministic control input, $\omega \in \mathbb{R}^q$ is a stochastic (process noise) vector, $y \in \mathbb{R}^p$ is the measured output, and $\nu \in \mathbb{R}^p$ is a stochastic (measurement noise) vector. The process and measurement noise vectors are each assumed to be Gaussian white noise processes:

$$\begin{aligned} E\langle\omega(t)\rangle &= 0, \\ E\langle\omega(t_1), \omega(t_2)\rangle &= Q(t_1)\delta(t_1 - t_2), \\ E\langle\nu(t)\rangle &= 0, \\ E\langle\nu(t_1), \nu(t_2)\rangle &= R(t_1)\delta(t_1 - t_2). \end{aligned}$$

The $\delta(t)$ is the Dirac delta function. The power spectral density matrices $Q(t)$ and $R(t)$ are symmetric positive definite for all t.

For implementation purposes, it is usually convenient to work with the discrete-time state-space representation:

$$x(k+1) = Ax(k) + Bu(k) + \Gamma_\omega\omega(k), \tag{2.19}$$

$$y(k) = Cx(k) + \nu(k). \tag{2.20}$$

In equation (2.19, 2.20), $x(k) = x_{kT}$, k is an integer, and T is a fixed sample period. If F_e, G_u and G_ω are time invariant over the interval $(kT, (k+1)T]$ then A, B, Γ_ω, and the statistics of the discrete-time Gaussian white noise process $\omega(k)$ can be determined so that the solutions to equation (2.17, 2.18) at the sample times $t = kT$ have the same statistics as the solution of equation (2.19, 2.20). Both $\nu(k)$ and $\omega(k)$ are discrete-time Gaussian white noise sequences with covariance matrices defined by:

$$E\langle\nu(k)\nu(j)^T\rangle = \begin{cases} R(k) & k = j, \\ 0 & otherwise; \end{cases} \tag{2.21}$$

$$E\langle \omega(k)\omega(j)^T \rangle = \begin{cases} Q(k) & k = j, \\ 0 & otherwise. \end{cases} \tag{2.22}$$

The fact that linear operations on Gaussian random variables yields Gaussian random variables and the assumptions that $u(k)$ is deterministic and $\omega(k)$ and $\nu(k)$ are Gaussian random variables, the distributions of $x(k)$ and $y(k)$ are completely characterized by their statistics, which may be time varying.

2.3.3 State Observers

A state observer estimates the state variables based on the measurements of the output and control variables. It is a subsystem to reconstruct the state vector of the plant. The mathematical model of the observer is basically the same as that of the plant, except that an additional term is included that consists the estimation error to compensate for the system inaccuracies. The estimation error or observation error is the difference between the measured output and the estimated output. The initial error is the difference between the initial state and the initial estimated state. The state of a system is the minimal set of information required to completely summarize the status of the system at an initial time t_0. Because the state is a complete summary of the status of the system, it is immaterial how the system managed to get to its state at t_0. In addition, knowledge of the state at t_0, of the inputs to the system for $t > t_0$, and of the dynamics of the system allows the state to be determined for all $t > t_0$. Therefore, the concept of state allows time to be divided into past ($t < t_0$), present ($t - t_0$), and future ($t > t_0$), with the state at t_0 summarizing the information from the past that is necessary (together with knowledge of the system inputs) to predict the future evolution of the system.

The goal of the control system designer is to specify the inputs to a dynamic system to force the system to perform some useful purpose. This task can be interpreted as forcing the system from its state at t_0 to a desired state at a future time $t_1 > t_0$. Obtaining accurate knowledge of the state at t_0 and for all states between t_0 and t_1

is often critical to the completion of this control objective. This process is referred to as state estimation.

Chapter 3
FAULT DETECTION, ISOLATION AND ESTIMATION

3.1 Introduction

Presented in this chapter is a systematic approach to data-driven modeling of a system and fault detection, isolation and estimation (FDIE). To increase the reliability, safety and economy of engineering systems, growing demands are being placed on the fault detection, isolation and estimation. In automotive systems FDIE plays a critical role in evaluating if a vehicle is healthy enough to meet the emission standards. The requirements of any FDIE approach can be characterized using the following definitions proposed by an IFAC technical committee, SAFEPROCESS (Isermann et al. 1997a):

1. **Fault:** Unpermitted deviation of at-least one characteristic property of the system from normal/standard behavior.
2. **Failure:** Permanent interruption of the system's ability to perform a required function under specified operating conditions.
3. **Fault Detection:** Detection of faults present in the system and time of detection.
4. **Fault Isolation:** Determination of kind, location and time of fault.
5. **Fault Estimation:** Determination of the size and time variant behavior of the fault.

Fault isolation and estimation collectively is also known as *fault diagnosis*. FDIE hence can be described as a process which, when applied to a system prone to unpermitted deviation of at-least one characteritic property, would detect that deviation, locate the characteristic property and estimate the size of the deviation.

3.2 Conventional Model-Free vs Model Based FDIE

The field of fault detection and diagnosis has shifted focus from conventional process model-free methods (see Gertler 1998, Isermann 1984, 1987, 1997b, 2005, Simani et al.2003). As the name suggests the model-free FDIE methods do not use a model of the system for diagnostics. The most common way of detecting a fault without using a system model is to perform the limit or threshold checking of available information from sensors. An alarm is triggered whenever the sensor signal value is found out to be out of a normal range.

Another traditional way of performing fault detection is to use the concept of **hardware redundancy**. Duplication (or even more) of sensors were employed by using more than one sensor and comparing the values of the two (or more) for detecting the fault of the system or sensor itself. The disadvantage of employing sensor redundancy is that it requires additional costs, space and hardware. In addition extra components increase the complexity of the system and may in turn require the need of additional diagnostic requirements.

Spectrum analysis also comes under the class of model-free methods. Under normal operating conditions the plant variables have a typical frequency spectrum; and a deviation from this typical behavior can be used as an indicator of abnormality.

The attraction of model-free methods lies in their simplicity. However, these methods are not suitable for detecting small magnitude of incipient fault and are not robust to noise and disturbances. Moreover in-depth fault diagnosis to isolate and estimate the size of the fault through threshold checking or hardware redundancy is not possible. To overcome these drawbacks, fault detection and diagnosis methods using the model of the system are required. Model based diagnostic methods have the following advantages over the traditional model-free methods (Nyberg 1999)

1. Fault isolation and estimation can be performed in addition to fault detection.

2. Early detection of small faults can be realized.

3. Faults can be detected during closed loop operation of the system.

4. Diagnosis can be performed over a larger operating range of the system including during transient condition.

5. The effect of disturbances and noise acting on the system can be negated in a systematic way.

Model based FDIE methods rely on the concept of *analytical redundancy*. In contrast to physical redundancy where additional sensors are used, in analytical redundancy sensory measurements are compared with analytically obtained values of a particular variable. These analytical values are obtained by employing the model of the system. The basic principle of model based FDIE method is depicted in Figure 3.1. The figure shows a typical plant being subjected to noise $v(t)$ and faults. The model based fault detection block uses the actual plant input $u(t)$ and output $y(t)$ information to generate *residuals* or *features*. These features are then compared against normal behavior of the system to detect the fault. Residual generation is followed by residual evaluation where the features are analyzed to isolate and estimate the size of the fault. Model based FDIE methods can be characterized by the way these residuals are generated and evaluated. A brief description of different model based FDIE techniques is described in the following.

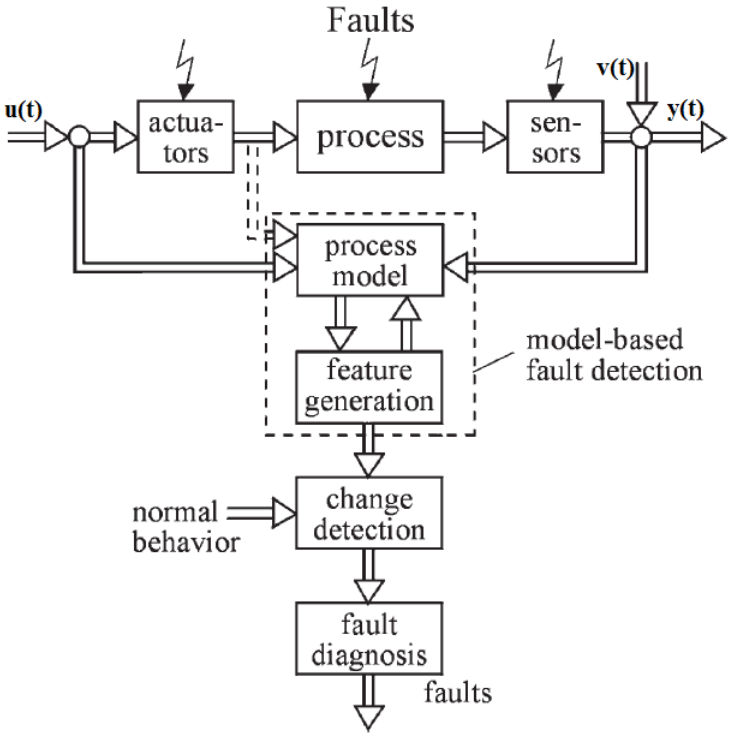

Figure 3.1: Schematic Representation of General Model Based FDIE Method (reproduced from [3]).

1. *Observer Based FDIE:*

   Observer based FDIE employs an observer for the system and compares the actual output of the system with the predicted output to generate the residual. The 'unknown input' design technique as explained in [2] can be used to decouple the residuals from the disturbances. In noisy environments a Kalman Filter can be used to generate residuals. Fault isolation is generally accomplished by using the freedom in the design of the observer or by running a number of observers for each different suspected fault.

2. *Parity Equations Based FDIE:*

   Parity equations are residual generators that generate residuals by using and

re-arranging the direct input-output model equations. Linear dynamic transformation is used on these model equations to provide disturbance decoupling and fault isolation enhancement [41].

3. *Parameter Estimation Based FDIE:*

   Parameter estimation method uses system model adaptation and generates the residual by comparing the parameters of the online adapted model with that of the healthy model. The healthy model is obtained by identifying or modeling the plant in a fault free situation ([3], [2]).

One of the major issues with the model based FDIE is the requirement of a reliable model of the system. The accuracy of this model plays a critical role in accurate fault detection and diagnosis. In actual identification of the model based diagnostic method, the critical step is the identification of the model for a complex process. Presented in subsection 3.3 is the proposed fault detection, isolation and estimation methodology. Robustness of the FDIE method will be explained through sensitivity analysis.

3.3 Information Synthesis

In the following a systematic approach to fault detection, isolation and estimation is proposed using *Information Synthesis* (IS) as presented in ([**?**]). Information Synthesis employs an on-line adaptive model-based parameter estimation technique where the model structure and the parameter of the model are determined using system identification techniques. The motivation to use the on-line adaptive model based technique is that it can be efficiently used to determine the health of the system under normal operation. In the presence of fault the input-output model of the system differs from that of the healthy system. Adapting the model parameters of the healthy system to the faulty system can capture this difference. A coefficient error vector or residual can be generated by comparing the healthy model parameters with

the adapted model parameters. The magnitude of this error vector when compared to a predefined threshold serves to detect the presence of fault and to estimate the size, provided a fault is present. The isolation of faults is performed by evaluating the angles of the error vector made with the coefficient axis in the fault space where the origin represents the healthy system. IS hence can be decomposed into the following steps

1. Estimating the nominal fault free model of the system in an off-line process,
2. Estimating the parameters of the model through an on-line parameter estimation technique, and
3. Residual generation and evaluation.

3.3.1 Recursive Least Squares Parameter Estimation

Least squares (LS) methods can be used to estimate the coefficients of the model structure. LS methods are based on minimizing the squared error as shown in Soderstrom [6] and can give consistent estimates in the presence of zero mean white noise. Let us suppose a system subjected to noise as shown in the Figure 3.2.

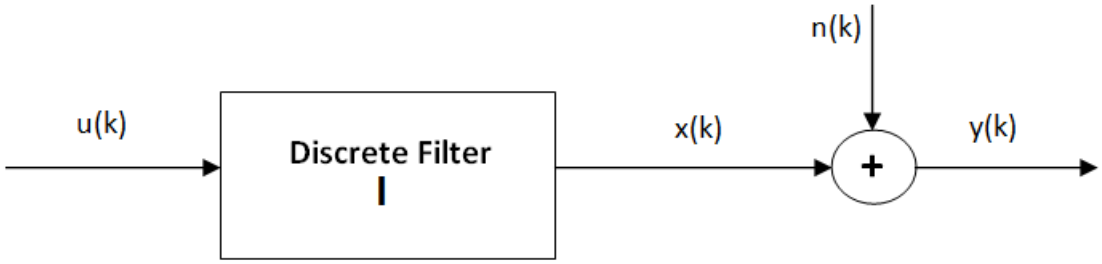

Figure 3.2: A General System.

where I is a linear-in-the-parameters discrete time system, $u(k)$ is the input signal, $x(k)$ is the noise free output of the system, $\vartheta(k)$ is a zero mean white noise and y(k) is the actual output with noise.

Now suppose the input and output relationship of the above system is given by

$$y(k) = \sum_{i=0}^{m} b_i u(k-i) - \sum_{i=1}^{n} a_i y(k-i) + v(k), \tag{3.1}$$

$$v(k) = \vartheta(k) + \sum_{i=1}^{n} a_i \vartheta(k-i). \tag{3.2}$$

where $n > m$. It can be seen that the noise after entering the system is affected by the system dynamics and becomes colored. In the matrix form the above model can be written as

$$y(i) = \varphi^T(i)\theta + \varepsilon(i), \tag{3.3}$$

where $y(i)$ is a measurable quantity whose elements are called regressed variables.

$\varphi(i)$ is an vector of known quantities called regression vector. The elements of vector $\varphi(i)$ are often called regression variables, regressors or explanatory variables.

θ is an vector of unknown parameters. It is mainly called a parameter vector or coefficient vector.

The variable i takes integer values. Sometimes i denotes a time variable but this is not necessarily the case. The linear regression model may or may not be linear in φ, but it is always linear in θ. v may represent effects of measurement errors in y or φ (un-modeled dynamics).

To understand the terminologies in (3.1) better, a model called ARX model is chosen,

$$y(i) + a_1 y(i-1) + ... + a_n y(i-n) = b_1 u(i-1) + ... + b_m u(i-m) + v(i). \quad (3.4)$$

If S represents the vector $\begin{bmatrix} 1 & a_1 & ... & a_n \end{bmatrix}$ and T represents the vector $\begin{bmatrix} b_1 & ... & b_m \end{bmatrix}$, then AR part of ARX refers to the autoregressive part $Sy(i)$ and X to the exogenous input $Tu(i)$. If the system is represented by ARX model,

$$\varphi(i) = \begin{bmatrix} -y(i-1) & -y(i-2) & ... & -y(i-n) & u(i-1) & ... & u(i-m) \end{bmatrix}^T. \quad (3.5)$$

Also,

$$\theta = \begin{bmatrix} a_1 & ... & a_n & b_1 & ... & b_m \end{bmatrix}^T. \quad (3.6)$$

Least Squares Criterion

The prediction error is

$$\varepsilon(i, \theta) = y(i) - \varphi^T(i)\theta. \quad (3.7)$$

For N data points, the regressed variable vector, regression vector and prediction error vector can be written as

$$Y = \begin{pmatrix} y(1) \\ \vdots \\ y(N) \end{pmatrix}, \Phi = \begin{pmatrix} \varphi^T(1) \\ \vdots \\ \varphi^T(N) \end{pmatrix}, \varepsilon = \begin{pmatrix} \varepsilon(1) \\ \vdots \\ \varepsilon(N) \end{pmatrix}. \quad (3.8)$$

Regression matrix is now represented as

$$\Omega = \Phi\theta, \quad (3.9)$$

and error vector as

$$\epsilon = Y - \Phi\theta. \quad (3.10)$$

The least square estimate of θ is the vector $\hat{\theta}$ that minimizes the cost or loss function,

$$V(\theta) = \frac{1}{2}\sum_{i=1}^{N}\epsilon^2(i) = \frac{1}{2}\epsilon^T\epsilon = \frac{1}{2}\|\epsilon\|^2, \tag{3.11}$$

where $\| \bullet \|$ denotes the Euclidean vector norm. Please note that whereas ϵ is a linear function of θ, $V(\theta)$ is a quadratic function of θ. Our aim is to minimize $V(\theta)$. In other words, we desire to minimize the squared errors between the actual system response, $y(i)$ and one-step ahead model prediction $\varphi^T(i)\theta$.

Lemma 3.1

Assuming $\Phi^T\Phi$ to be positive definite and invertible, the minimum value of $V(\theta)$ is obtained at

$$\hat{\theta} = (\Phi^T\Phi)^{-1}\Phi^T Y. \tag{3.12}$$

The corresponding minimum value of $V(\theta)$ is

$$\min_{\theta} V(\theta) = V(\hat{\theta}) = \frac{1}{2}[Y^TY - Y^T\Phi(\Phi^T\Phi)^{-1}\Phi^TY]. \tag{3.13}$$

Proof of Lemma:

We know that

$$\epsilon = Y - \Phi\theta, \tag{3.14}$$

and

$$\begin{aligned} V(\theta) &= \frac{1}{2}[Y - \Phi\theta]^T[Y - \Phi\theta], & (3.15)\\ &= \frac{1}{2}[\theta^T\Phi^T\Phi\theta - \theta^T\Phi^TY - Y^T\Phi\theta + Y^TY], & (3.16) \end{aligned}$$

$$V(\theta) = \frac{1}{2}[\theta - (\Phi^T\Phi)^{-1}\Phi^TY]^T(\Phi^T\Phi)[\theta - (\Phi^T\Phi)^{-1}\Phi^TY] + \frac{1}{2}[Y^TY - Y^T\Phi(\Phi^T\Phi)^{-1}\Phi^TY]. \tag{3.17}$$

By assumption, $\Phi^T\Phi$ is positive definite, so the first term can only be greater than or equal to zero. Also, the second term does not depend on θ, so $V(\theta)$ can be

minimized by keeping the first term as zero. Hence, we get the above result.

Note that Φ contains more rows than columns. The number of columns in Φ is determined by the number of coefficients to be estimated and its number of rows is based on the number of data points selected for parameter estimation.

When $\Phi^T\Phi$ is singular, Φ does not have full rank (i.e., rank $\Phi < n$). In that case, (3.12) will have infinite solutions. They span a subspace which describes a valley of minimum points of $V(\theta)$. Longer time record length corresponds to $\Phi^T\Phi$. The reason for $\Phi^T\Phi$ to be as large as possible or to use longer data sets is that we can have a more accurate estimate of the model.

Analysis Of Least Squares Estimate

Let us again consider the linear regression model structure $y(i) = \varphi(i)^T\theta + v(i)$ where θ_o is called the true parameter vector. Assume further that $v(i)$ is white noise. $v(i)$ has zero mean and variance σ^2.

Then, the following properties hold:

1. $\hat{\theta}$ is an unbiased estimate of θ_o.

2. The covariance matrix of $\hat{\theta}$ is given by

$$cov(\hat{\theta}) = \sigma^2(\Phi^T\Phi)^{-1}. \tag{3.18}$$

3. An unbiased estimate of σ^2 is given by

$$\chi^2 = 2V(\hat{\theta})/(N-n). \tag{3.19}$$

Recursive Least Squares Estimate

The recursive least squares estimation (RLSE) calculation reduces the computational requirements and improves the accuracy of the LSE. In particular, using a recursive solution to the LSE problem avoids the need of taking the inverse of the

matrix $\Phi^T\Phi$ when solving for parameter estimates θ. Apart from reducing the computational requirements, the recursive solution prevents the matrix inversion that occurs in LSE and helps improve the accuracy of the estimate. Elaborating it, let us suppose the number of parameters is x and data record length is $1000 \times x$. The construction of $\Phi^T\Phi$ takes the original size of $\Phi \in \mathbb{R}^{1000x \times x}$ and crushes it to a much smaller matrix of size $\Phi^T\Phi \in \mathbb{R}^{x \times x}$. This reduction of a much larger matrix to such a small one has chances that the estimates will not be as accurate as can be found by the recursive approach. The output matrix updates itself from $(k-1)^{th}$ to k^{th} time sample in the following way,

$$Y(k) = \begin{bmatrix} Y(k-1) \\ y(k) \end{bmatrix}. \tag{3.20}$$

Similarly, for $(k+1)^{th}$ time sample,

$$Y(k+1) = \begin{bmatrix} Y(k) \\ y(k+1) \end{bmatrix}, \Phi(k+1) = \begin{bmatrix} \Phi(k) \\ \varphi(k+1) \end{bmatrix}, \tag{3.21}$$

$$P(k) = [\Phi^T(k)\Phi(k)]^{-1}. \tag{3.22}$$

Here,

$$P(k+1)^{-1} = P(k)^{-1} + \varphi(k+1)\varphi(k+1)^T. \tag{3.23}$$

The parameter estimate,

$$\begin{aligned} \hat{\theta}(k+1) &= P(k+1)\Phi(k+1)^T Y(k+1) \\ &= [P(k)^{-1} + \varphi(k+1)\varphi(k+1)^T]^{-1}[\Phi(k)^T Y(k) + \varphi(k+1)y(k+1)] \\ &= [P(k+1)][P(k)^{-1}\hat{\theta(k)} + \varphi(k+1)y(k+1)], \end{aligned} \tag{3.24}$$

$$\hat{\theta}(k+1) = [P(k+1)][P(k+1)^{-1} - \varphi(k+1)\varphi(k+1)^T]$$
$$\hat{\theta}(k) + P(k+1)\varphi(k+1)y(k+1)$$

$$= \hat{\theta}(k) + P(k+1)\varphi(k+1)[y(k+1) - \varphi(k+1)^T\hat{\theta(k)}]. \quad (3.25)$$

Since,

$$[A + BC]^{-1} = A^{-1} - A^{-1}B(I + CA^{-1}B)^{-1}CA^{-1},$$

$$\begin{aligned} P(k+1) &= [P(k)^{-1} + \varphi(k+1)\varphi(k+1)^T]^{-1} \\ &= P(k) - P(k)\varphi(k+1)(I + \varphi(k+1)^T P(k)\varphi(k+1))^{-1}\varphi(k+1)^T P(k). \end{aligned} \quad (3.26)$$

Properties of Recursive Least Square Estimate

Let the data be generated by

$$y(k) = \varphi(k)^T\theta + v(k), \quad (3.27)$$

$$\begin{aligned} \hat{\theta}(N) &= [\sum_{k=1}^{N}\varphi(k)\varphi(k)^T]^{-1}[\sum_{k=1}^{N}\varphi(k)y(k)] \\ &= P(N)[P(N)^{-1}\theta_o + \sum_{k=1}^{N}\varphi(k)v(k)] \\ &= \theta_o + \underbrace{[\sum_{k=1}^{N}\varphi(k)\varphi(k)^T]^{-1}}_{\rho}\underbrace{[\sum_{k=1}^{N}\varphi(k)\varepsilon(k)]}_{\varsigma}, \end{aligned} \quad (3.28)$$

$\lim_{N\to\infty}\hat{\theta}(N) = \theta$ only if ρ grows larger with N or ς grows smaller with N. To make it possible, two typical cases are considered

1. $\varepsilon(k)$ is a zero mean white sequence uncorrelated with $\varphi(k)$.

2. $P(N)^{-1}$ remains non-singular and grows.

Monotonic Property

Let the performance index is defined by

$$\begin{aligned} Q(\hat{\theta}(k)) &= \frac{1}{2}[(\hat{\theta}(k) - \theta)P(k)^{-1}(\hat{\theta}(k) - \theta)] \\ &= \frac{1}{2}\tilde{\theta}(k)P(k)^{-1}\tilde{\theta}(k). \end{aligned} \quad (3.29)$$

Here, $P(k)^{-1}$ keeps on increasing. The proof that $P(k)^{-1}$ will be more than $P(k-1)^{-1}$ is as follows:

$$\begin{aligned} P(k)^{-1} &= P(k-1)^{-1} + \varphi\varphi^T \\ \Rightarrow z^T P(k)^{-1} z &= z^T P(k-1)^{-1} z + z^T \varphi\varphi^T z \\ \Rightarrow z^T P(k)^{-1} z &= z^T P(k-1)^{-1} z + \underbrace{w^T w}_{\psi}. \end{aligned} \tag{3.30}$$

ψ is always positive definite which directly implies the result.

Having said that, to make $Q(\hat{\theta}(k))$ bounded and reduce, $\tilde{\theta}(k)$ must keep on decreasing.

This can also be concluded by finding the difference between the previous performance index and next performance index,

$$Q(\hat{\theta}(k)) - Q(\hat{\theta}(k-1)) = \underbrace{\frac{1}{2}\varepsilon^2(k)}_{E_1} - \underbrace{\frac{1}{2}\frac{\varphi(k)^T P(k-1)\varphi(k)}{1+\varphi(k)^T P(k-1)\varphi(k)}\tilde{\theta}(k)^T\tilde{\theta}(k)}_{E_2}. \tag{3.31}$$

If $\varepsilon = 0 (E_1 = 0)$, then $Q(\hat{\theta}(k))$ is a non increasing sequence as E_2 is always positive. So, since $P(k)^{-1}$ is an increasing sequence of matrices, $\tilde{\theta}(k)$ is expected to converge monotonically. In recursive calculation, P and forgetting factor λ (described in next section) enable the estimation of time-varying parameters-coefficients for models that vary with time. One advantage of the recursive equations is that matrix inversion calculations are reduced to a series of additions and multiplications. Additionally, these simple operations can be performed as each data pair is recorded, reducing the memory that would be required to record long data sets prior to performing the matrix inversion. There are also some disadvantages with performing these calculations continually. Since data is recorded and passed through the estimation algorithm one sample at a time; there is no means to check for persistency of excitation. With new data pairs recorded and processed through the algorithm, the recursive calculation must be completed before the next data point is recorded. Since the calculations have been simplified, completing the computation would likely be a

problem for very high data sample rates.

A solution to this problem is batch mode operation. Batch mode RLSE processes a preselected segment of data that has been sampled and stored. Thus, data is not processed through the algorithm as it is recorded.

Thus far, it was assumed that the model parameters were constants when compared to the changing input and output data. The model parameters for real world systems are in fact not constant and these changing coefficients require advanced control solutions such as adaptive control. The non-recursive LSE previously outlined will calculate model coefficients based on the data to which it is applied, and new data sets generate a new set of parameters. In contrast, the recursive equations allow data to be continually passed through the estimation algorithm. As a result, a limitless amount of data can be used for the estimation. Without specific adjustments to accommodate time-varying parameters, however, the estimates will eventually converge to fixed values and future changes in system dynamics may never appear as changes in the model coefficients.

Convergence of the parameter estimates from the recursive equations occurs because the magnitudes of the terms in the matrix $P(k)$ become progressively smaller. This reduction in magnitude causes a corresponding reduction in the estimation gain. Even if model dynamics result in an error between the current model and the actual system output, the parameters will not be updated if the $P(k)$ matrix and estimation gain have already decreased to an insignificant magnitude.

There are several methods for dealing with time-varying parameters within the recursive least squares algorithm. One option is to formulate the model structure to include time varying parameters. The other two methods are less complicated and have already been included in the recursive equations stated earlier. Time-varying parameters can be addressed through the use of the forgetting factor or by the $P(k)$ matrix (called the covariance matrix) resetting. Both methods essentially accomplish the same thing; they prevent the $P(k)$ matrix and the corresponding estimation gain

from going to zero. Because the two methods differ in the way they affect the $P(k)$ matrix and the estimation gain, they have different practical applications.

Resetting the $P(k)$ matrix to a diagonal matrix with large numbers on the diagonal will cause the parameter estimates to re-converge based on new data. For larger value diagonal elements in $P(k)$, the estimation gain will be higher and the parameter estimates will respond faster to any differences between the new data and the current model.

Naturally then, the method of resetting the $P(k)$ matrix is useful for systems with parameters that experience infrequent but significant changes. When a change in the system being controlled has occurred, the $P(k)$ matrix can be reset so that the model parameters will converge to new values and the controller will be able to adapt to the new system dynamics.

Resetting the $P(k)$ matrix is not the ideal method to cope with systems that have slowly varying parameters. The matrix would need to be reset too frequently and convergence of the parameters between resets would be degraded. The forgetting factor method can be applied to systems with gradually shifting parameters. The forgetting factor accounts for time-varying parameters by preventing the values in the $P(k)$ matrix from ever getting too small. By preventing the terms in the $P(k)$ matrix from becoming insignificant in magnitude, the model parameters will always adapt to new data. Setting the forgetting factor to unity essentially turns off this method of dealing with the time-varying parameters. Decreasing the forgetting factor from unity applies the method. The further the forgetting factor is from unity, the faster the system responds to incoming data. An alternative explanation is that as the forgetting factor is decreased from unity, a smaller number of past data points are represented in the current parameter value.

If the model parameters are based on an infinite number of previous data points, additional data with different system dynamics will not be able to impact the model parameters that have already converged. For a forgetting factor less than unity, the

model parameters can adapt to changes in system dynamics because the window of data that is used to model the parameters will always be shifting ahead in time and will eventually contain only new system dynamics. For both the forgetting factor and the $P(k)$ resetting method, there must be a balance between responding to parameter variation and converging to good parameter estimates. Resetting the $P(k)$ matrix often, or having a forgetting factor much less than unity, will create a fast response to changes in the system dynamics. But, this increases the impact of current data on the estimated parameters and will cause more variability in the parameter estimates.

Combining the forgetting factor and $P(k)$ resetting methods provides particular advantages. A forgetting factor close to unity will provide good parameter convergence while still allowing the model estimate to drift over time and respond to a slowly changing system. Including the option to reset the $P(k)$ matrix adds rapid adaptation capability to the algorithm. The $P(k)$ matrix reset can be scheduled to occur periodically to let the model parameters re-converge after an extended period of time. The $P(k)$ matrix reset can also be manually triggered so that adaptation can occur when a known change in the system dynamics has taken place.

Persistency of excitation is a paramount topic associated with system identification. Input signals which do not sufficiently excite the system dynamics will not produce accurate models. The lack of a persistently exciting signal manifests itself in the matrix $\varphi^T\varphi$. Specifically, the matrix $\varphi^T\varphi$ will be either singular (noninvertible) or will be ill condition (near singular) producing numerical errors. When such a condition exists, the parameter estimates are inaccurate. Although there are methods that deal with near singular $\varphi^T\varphi$ matrices, such situations serve as evidence to an inadequate design of experiments. Resorting to the recursive least squares solution does not compensate for a lack of a persistently exciting input. Instead, the application of recursive least squares estimation will result in some of the parameter estimates diverging. Input signal design, as it pertains to the system identification algorithms presented here, must receive a detailed development. The design of these signals will

satisfy the persistency of excitation conditions thereby eliminating one problem area during parameter identification. It is tempting to develop a simple check of the system input sequence to evaluate if the data record satisfies persistency of excitation. For example, comparing the maximum and minimum value of the measured system input sequence is one simple check and may indeed work for specific applications. This simple check could be implemented for online parameter estimates and could be used to identify datasets to reject. However, such a simple method does not in any way guarantee persistency of excitation.

Weighted Recursive Least Square Estimate

We now put weights W^{-1} in error at previous instances of time in the least square estimate equation. If the parameter is constant, a constant weight given at all the instances of time is justifiable. But, if the parameters are slowly changing, then there is a need to define weight in such a way that weight to the error at recent past data is more than that of relatively older data. In this way, the recent variation in recent parameters will be given more weight as compared to the past parameters. This enables the estimator to track the adapted parameters with time. The cost function will become,

$$V(\theta) = \frac{1}{2}[Y - \Phi\theta]^T W^{-1}[Y - \Phi\theta]. \tag{3.32}$$

The solution of the above cost function in order to minimize it is given by

$$\Phi^T W^{-1}\Phi\hat{\theta} \;\; = \;\; \Phi^T W^{-1} Y \Rightarrow \hat{\theta} = (\Phi^T W^{-1}\Phi)^{-1}\Phi^T W^{-1} Y, \tag{3.33}$$

where W is the covariance matrix which is defined by expectation of error squared,

$$W = E[\varepsilon\varepsilon^T]. \tag{3.34}$$

For white noise, $W = \sigma^2 I$where σ^2 is the variance of error. Old measurements are represented by $\Phi_0\theta = Y_0$ and new measurements are represented by $\Phi_1\theta = Y_1$. In

block form,

$$\begin{bmatrix} \Phi_0 \\ \Phi_1 \end{bmatrix} \theta = \begin{bmatrix} Y_0 \\ Y_1 \end{bmatrix}. \tag{3.35}$$

It should be noted here that the error may or may not be correlated for the present state, but it is assumed that the error of the present state is uncorrelated as compared to the error of previous state, which can be represented by W as,

$$W = \begin{bmatrix} W_0 & 0 \\ 0 & W_1 \end{bmatrix}. \tag{3.36}$$

The dimensions of W_0 and W_1 may or may not be the same and they may or may not be diagonal matrices. But, as explained earlier, the past and present errors are uncorrelated.

Our aim is to generate a matrix with as small a variance as possible, so that a heavier weight can be provided to the present measurements.
From (3.33),

$$\hat{\theta} = \underbrace{[(\Phi^T W^{-1} \Phi)^{-1} \Phi^T W^{-1}]}_{L_e} Y, \tag{3.37}$$

where L is a linear transformation of Y to Φ.

Let W_θ and P denotes the covariance matrix of parameters.

$$\begin{aligned} P = W_\theta &= E[(\theta - \hat{\theta})(\theta - \hat{\theta})^T] = E[(L\varepsilon)(L\varepsilon)^T] = LE[\varepsilon\varepsilon^T]L^T = LWL^T \\ &= [(\Phi^T W^{-1} \Phi)^{-1} \Phi^T W^{-1}] W [(\Phi^T W^{-1} \Phi)^{-1} \Phi^T W^{-1}]^T \\ &= (\Phi^T W^{-1} \Phi)^{-1}. \end{aligned} \tag{3.38}$$

As P is updated, it keeps on decreasing. Let us find out what the update of P will be

$$P^{-1} = \begin{bmatrix} \Phi_0^T & \Phi_1^T \end{bmatrix} \begin{bmatrix} W_0^{-1} & 0 \\ 0 & W_1^{-1} \end{bmatrix} \begin{bmatrix} \Phi_0 \\ \Phi_1 \end{bmatrix} = \Phi_0^T W_0^{-1} \Phi_0 + \Phi_1^T W_1^{-1} \Phi_1. \tag{3.39}$$

So, the update formula for P becomes,

$$P_1^{-1} = P_0^{-1} + \Phi_1^T W_1^{-1} \Phi_1, \tag{3.40}$$

and the update formula for θ becomes,

$$\theta_1 = P_1(\Phi_0^T W_0^{-1} Y_0 + \Phi_1^T W_1^{-1} Y_1). \tag{3.41}$$

This can be simplified to,

$$\hat{\theta}_1 = \hat{\theta}_0 + K(Y_1 - \Phi\hat{\theta}_0), \tag{3.42}$$

where K is the gain matrix, defined by,

$$K = P_1 \Phi_1^T W_1^{-1}. \tag{3.43}$$

Exponential Weighting

If W^{-1} is a diagonal $w(i)$,

$$V(\theta) = \frac{1}{2}\sum_{i=1}^{N} w(i)[y_i - \varphi(i)^T\theta]^2, \tag{3.44}$$

$$V(\hat{\theta}(k)) = \sum_{i=1}^{k} \lambda^{k-i}(y(i) - \varphi(i)^T\hat{\theta}(k))^2. \tag{3.45}$$

In this equation, $\hat{\theta}(k)$ has been generated and $y(i)$ is being estimated in such a way that the estimate of $y(i)$ will rely more on the recent information of estimated parameters. In order to do so, λ^{k-i} has been introduced which takes values less than or equal to one and is always positive. This means that with increasing time the measurement obtained previously are discounted. The smaller the value of λ, the quicker the information in previous data will be forgotten. The explanation is directly evident from the above expression.

So, the estimate is modified as,

$$\hat{\theta}(k) = \hat{\theta}(k-1) + P(k)\varphi(k)\Xi(k), \tag{3.46}$$

where,

$$\Xi(k) = y(k) - \varphi(k)^T\hat{\theta}(k-1), \tag{3.47}$$

$$P(k) = \frac{1}{\lambda}\left(P(k-1) - \frac{P(k-1)\varphi(k)\varphi(k)^T P(k)}{\lambda + \varphi(k)^T P(k)\varphi(k)}\right). \tag{3.48}$$

3.3.2 Instrumental Variable Method

Instrumental variable is one of the variants of least squares estimation. LSE results in unbiased estimate only when φ and noise v are uncorrelated. But it suffers from bias if they are correlated.

IV method takes this problem into account by selecting a vector ξ which is

1. uncorrelated with v, and

2. very strongly correlated with v.

Let the estimate for the nominal system $Y = \theta_o \Phi + \nu$ using IV method is

$$\begin{aligned} \hat{\theta}_{IV} &= [\sum_{k=1}^{N} \xi(k)\varphi^T(k)]^{-1} \sum_{k=1}^{N} \xi(k)y(k), \\ &= [Z^T\Phi]^{-1}Z^TY, \\ &= \theta_o + [Z^T\Phi]^{-1}Z^T\nu. \end{aligned} \tag{3.49}$$

The aim while selecting the vector ξ is to minimize the second term of calculated IV estimate (3.49). It is realized when $Z^T\nu \rightarrow 0$ while keeping the $[Z^T\Phi]^{-1}$ as invertible.

Nonparametric modeling methods will be used in the selection of model structure and the validation of the identified model, but parametric methods are better suited to online identification and control. Parametric models condense a large amount of system information into a comparatively small array of parameters. A well known, simple, and still very useful linear parameter identification technique is the least squares algorithm. Instrumental variable methods improve the least squares estimates by reducing bias generated by noise or un-modeled disturbances.

The four-step instrumental variable approach is an algorithm for estimating the parameters of a linear model, and it is closely related to the least squares estimation method. Parameters estimated through the least squares technique are improved upon using both instrumental variables and a noise-specific estimated filter. Prior

to implementing the four-step instrumental variable method, the order of the model must be determined. As a discrete system, the model can be written in the form of a linear difference equation

$$y(t) + a_1 y(t-1) + a_2 y(t-2) + ... + a_n y(t-n) = \\ b_0 u(t) + b_1 u(t-1) + b_2 u(t-2) + ... + b_m u(t-m). \quad (3.50)$$

The parameters $a_1 ... a_n$ and $b_0 ... b_m$ are constants, not functions of time. Although the parameters may change with time, it is assumed that the change is sufficiently slow so that the parameters can be treated as constants compared to the variables $y(t)$ and $u(t)$. By taking the z-transform of the linear difference equation, the expression can be rewritten in terms of a discrete transfer function

$$G(z) = \frac{Y(z)}{U(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + ... + b_m z^{-m}}{1 + a_1 z^{-1} + a_2 z^{-2} + ... + a_n z^{-n}} = \frac{\beta(z)}{\alpha(z)}, \quad (3.51)$$

with the orders n and m known. The four-step instrumental variable approach can be implemented to estimate the values of the parameters $b_0 ... b_m$ and $a_1 ... a_n$. A shorthand notation that will be used to indicate the passing of vectors of time dependent data through a discrete filter or model is

$$\alpha(z) y(t) = \beta(z) u(t) \quad (3.52)$$

The basis of the instrumental variable algorithm is to use instrumental variables, which are correlated to the system input and uncorrelated to the system noise, to produce consistent parameter estimates. One method to generate these instruments is to use a previous estimate of the system model.

Step 1: Least Squares Parameter Estimation

The first step of the four-step instrumental variable approach uses the least squares algorithm to determine a first estimate of the parameters. These parameter estimates will be biased because of noise content in the measured output signal. The solution

to LS is taken from equation 3.12) where the initial parameter estimates are denoted as

$$\hat{\theta}_1 = [\ \hat{a}_1 \quad \hat{a}_2 \quad ... \quad \hat{a}_n \quad \hat{b}_0 \quad \hat{b}_1 \quad \hat{b}_2 \quad ... \quad \hat{b}_m\ ]^T. \tag{3.53}$$

Replacing the actual parameters in equation (3.51) with the estimated parameters from equation (3.53), the transfer function of the same form as $G(z)$ using the estimated parameters is referred to as $G_1(z)$

$$G_1(z) = \frac{\hat{b}_0 + \hat{b}_1 z^{-1} + \hat{b}_2 z^{-2} + ... + \hat{b}_m z^{-m}}{1 + \hat{a}_1 z^{-1} + \hat{a}_2 z^{-2} + ... + \hat{a}_n z^{-n}} = \frac{\beta_1(z)}{\alpha_1(z)}. \tag{3.54}$$

The modeled system output results from passing the known input vector $u(t)$ through the model transfer function to produce

$$x_1(t) = G_1(z)u(t). \tag{3.55}$$

Step 2: Instrumental Variable Parameter Estimation

Errors result in the least squares parameter estimate when the measured system output $y(t)$ contains noise. With a noise term $v(t)$ included, the equations for $y(t)$ become the following:

$$\begin{aligned} y(t) = -a_1 y(t-1) - a_2 y(t-2) - ... - a_n y(t-n) + \\ b_0 u(t) + b_1 u(t-1) + b_2 u(t-2) + ... + b_m u(t-m) + v(t), \end{aligned} \tag{3.56}$$

or equivalently

$$\alpha(z)y(t) = \beta(z)u(t) + v(t) \tag{3.57}$$

or

$$y(t) = \varphi^T(t)\theta + v(t). \tag{3.58}$$

Because there is a correlation between the noise $v(t)$ and the system output $y(t)$ in the vector of variables $\varphi(t)$, the parameter vector that is the solution of equation (3.12) no longer approaches the real system parameters. To correct for this noise problem, the instrumental variable vector $\xi(t)$ is introduced. Because the instrumental variables

are paramount to reducing the effects of the noise, the selection of the instrumental variables is important. The instrumental variable vector is a vector that must be correlated to the input vector $u(t)$ and uncorrelated to the noise in the system. Two criteria for the instrumental variable vector (where E[] indicates the expected value) are

(c1) $lim_{N\to\infty}\frac{1}{N}\sum_{t=1}^{N} E[\xi(t)\varphi^T(t)]$ be nonsingular

and

(c2) $lim_{N\to\infty}\frac{1}{N}\sum_{t=1}^{N} E[\xi(t)v(t)] = 0.$

A simple method to generate instruments which are correlated to the system input and uncorrelated (weakly correlated) with the noise is to pass the known input $u(t)$ through the estimated system model $G_1(z)$, see equation (3.54). The estimated system output is then used in place of the actual system output producing the instrumental variable vector below

$$\begin{aligned}\xi(t) = [-x_1(t-1) \quad -x_1(t-2) \quad ... \quad -x_1(t-n) \\ u(t) \quad u(t-1) \quad u(t-2) \quad ... \quad u(t-m)]^T.\end{aligned} \quad . \quad (3.59)$$

This instrumental variable vector has the same form as the variable vector $\varphi(t)$, with the estimated system output substituted for the measured system output. Multiplying equation (3.58) by the instrumental variable vector gives

$$\xi(t)y(t) = \xi(t)\varphi^T(t)\theta + \xi(t)v(t). \quad (3.60)$$

Equation (3.60) clearly shows the reason for the two conditions given in (c1) and (c2). Solving for the parameter vector θ requires inverting the product $\xi(t)\varphi^T(t)$, hence the condition in (c1). The noise in equation (3.60) is eliminated if the condition (c2) holds. Using the instrumental variable vector $\xi(t))$ instead of the variable vector $\varphi(t)$, the least square error equation differentiated and putting to zero can be rewritten as follows

$$\frac{2}{N}[\sum_{t=1}^{N}\xi(t)(y(t) - \varphi^T(t)\hat{\theta})]. \quad (3.61)$$

Solving the above equation for the parameter vector produces equation (3.63) below, which will be referred to as the **instrumental variable parameter estimation formula**. These estimates of the parameters and discrete transfer function have been designated as the second estimates of the four-step algorithm

$$\hat{\theta}_2 = [\sum_{t=1}^{N} \xi(t)\varphi^T(t)]^{-1}[\sum_{t=1}^{N} \xi(t)y(t)], \tag{3.62}$$

$$\hat{\theta}_2 = [\ \hat{a}_1 \quad \hat{a}_2 \quad ... \quad \hat{a}_n \quad \hat{b}_0 \quad \hat{b}_1 \quad \hat{b}_2 \quad ... \quad \hat{b}_m\ ]^T, \tag{3.63}$$

and

$$G_2(z) = \frac{\hat{b}_0 + \hat{b}_1 z^{-1} + \hat{b}_2 z^{-2} + ... + \hat{b}_m z^{-m}}{1 + \hat{a}_1 z^{-1} + \hat{a}_2 z^{-2} + ... + \hat{a}_n z^{-n}} = \frac{\beta_2(z)}{\alpha_2(z)}. \tag{3.64}$$

Once the instruments have been generated using $G_1(z)$ and the parameters of the transfer function $G_2(z)$ have been calculated using the instrumental variable parameter estimation formula, an improved vector of the simulated system output can be generated

$$x_2(t) = G_2(z)u(t). \tag{3.65}$$

Step 3: Filter Determination

The use of the instrumental variables in step two of the four-step algorithm improves the initial least squares parameter estimate from step one, but does not completely eliminate the effects of the noise. Because the system model $G_1(z)$ used to generate the instruments includes the effects of noise, there is still some correlation between the instruments and the noise in the output signal. The effects of this noise can be further attenuated through the addition of a filter, denoted $Fi(z)$. The measured output of the system can be passed through a filter prior to the calculation of the parameters using the instrumental variable algorithm. To keep the filter from influencing the input-output relationship of the data, the known system input vector must also be passed through the same filter. Even if the same filter is applied to both the input and output data, it can still affect the estimated model if the filter

bandwidth is low enough to attenuate system dynamics as well as the noise. To reduce the likelihood that the noise filter would attenuate system dynamics, the filter is estimated based on the model error passed through the model transfer function denominator rather than just the model error. Passing the model error through the model transfer function denominator attenuates the model dynamics relative to the high frequency noise prior to the estimation of the filter. In this manner, the filter can be designed to attenuate noise in the measured output signal $y(t)$, but will not change the input-output relationship. The discrete filter $Fi(z)$ appears in the equations for the filtered input and output signals below

$$u_{Fi}(t) = Fi(z)u(t), \tag{3.66}$$

and

$$y_{Fi}(t) = Fi(z)y(t). \tag{3.67}$$

To aid in the discussion of noise modeling, the noise term $v(t)$ in equation (3.56)-(3.58) will be rewritten in terms of a discrete transfer function and white noise. Letting $e(t)$ represent zero-mean white noise, the noise signal is written as the result of passing white noise through a discrete filter $\Upsilon(z)$

$$y(t) = G(z)u(t) + \Upsilon(z)e(t). \tag{3.68}$$

An alternative expression for equation (3.68) is as follows

$$\alpha(z)y(t) = \beta(z)u(t) + \alpha(z)\Upsilon(z)e(t). \tag{3.69}$$

A comparison of equations (3.57) and (3.69) determines an expression for the noise signal $v(t)$

$$v(t) = \alpha(z)\Upsilon(z)e(t). \tag{3.70}$$

Substituting this expression for $v(t)$ into equation (3.58) gives the following equation

$$y(t) = \Phi^T(t)\theta + \alpha(z)\Upsilon(z)e(t). \tag{3.71}$$

With the addition of the filter, the following equation describes the filtered system output signal

$$y_{Fi}(t) = Fi(z)y(t) = Fi(z)\Phi^T(t)\theta + Fi(z)\alpha(z)\Upsilon(z)e(t). \tag{3.72}$$

Because the final parameter estimate will be generated using the instrumental variable algorithm, it is useful to examine the instrumental variable criteria when selecting the desired filter. Modifying equation (c2) to account for the filtered signal $y_{Fi}(t)$ results in the following expression

$$lim_{N\to\infty}\frac{1}{N}\sum_{t=1}^{N} E[\xi(t)Fi(z)\alpha(z)\Upsilon(z)e(t)] = 0. \tag{3.73}$$

Since $e(t)$ is a zero-mean white noise signal, it has an expected value of zero. To cancel the effects of the discrete filters $\alpha(z)$ and $\Upsilon(z)$, which color the white noise signal; the filter $Fi(z)$ is designed as their inverse, resulting in the following equations

$$Fi(z) = \frac{1}{\alpha(z)\Upsilon(z)} \tag{3.74}$$

and

$$lim_{N\to\infty}\frac{1}{N}\sum_{t=1}^{N} E[\xi(t)Fi(z)\alpha(z)\Upsilon(z)e(t)] = lim_{N\to\infty}\frac{1}{N}\sum_{t=1}^{N} E[\xi(t)e(t)] = 0. \tag{3.75}$$

Using the best estimate of the system transfer function determined to this point, along with equations (3.64) and (3.68), the noise signal can be estimated as the error between the measured output signal and the modeled output

$$\hat{\Upsilon}(z)e(t) = y(t) - x_2(t). \tag{3.76}$$

The noise signal used for the filter estimate is not just the model error from equation (3.76), but it is the product of the model denominator and that error. For the purposes of implementing the filter, it is useful to model the system noise as a discrete filter or linear difference equation. Combining equations (3.69) and (3.70), the noise signal can be written as

$$v(t) = \alpha(z)\Upsilon(z)e(t) = \alpha(z)y(t) - \beta(z)u(t). \tag{3.77}$$

An auto-regressive model structure will be used for $v(t)$, and the order of the model is selected based on the sum of the order of the numerator and denominator of the plant model $(n+m+1)$

$$v(t) = \alpha(z)\Upsilon(z)e(t) = \frac{1}{1 + c_1 z^{-1} + c_2 z^{-2} + ... + c_{n+m+1} z^{-1-n-m}} e(t), \tag{3.78}$$

$$v(t) = -c_1 v(t-1) - c_2 v(t-2) ... - c_{n+m+1} v(t-n-m-1) + e(t). \tag{3.79}$$

This selection of model order ensures that the order of the filter will increase and decrease as the complexity of the plant model changes, but also does not lead to excessive calculation demands. In the same way that the linear difference equation for the plant was rewritten in terms of vectors, $v(t)$ can be rewritten

$$v(t) = \Phi_v^T(t)\theta_v, \tag{3.80}$$

$$\theta_v = \begin{bmatrix} c_1 & c_2 & ... & c_{n+m+1} & 1 \end{bmatrix}^T, \tag{3.81}$$

and

$$\Phi_v(t) = \begin{bmatrix} -v(t-1) & -v(t-2) & ... & -v(t-n-m-1) & e(t) \end{bmatrix}^T. \tag{3.82}$$

The noise $v(t)$ can be calculated using equation (3.76), equation (3.77), and the most accurate plant transfer function estimate to this point

$$v(t) = \alpha_2(z)y(t) - \beta_2(z)u(t) = \alpha_2(z)[y(t) - x_2(t)]. \tag{3.83}$$

The parameter vector for the noise model $v(t)$ can be estimated using the least squares parameter estimation formula. Soderstrom proves that the addition of a zero mean white noise signal does not affect a least squares estimate, so $e(t)$ is neglected in the application of least squares and equations (3.81) and (3.82) become

$$\theta_v = \begin{bmatrix} c_1 & c_2 & ... & c_{n+m+1} \end{bmatrix}^T, \tag{3.84}$$

and

$$\varphi_v(t) = \begin{bmatrix} -v(t-1) & -v(t-2) & ... & -v(t-n-m-1) \end{bmatrix}^T. \tag{3.85}$$

By examining equations (3.74) and (3.78), the parameters calculated least squares estimation can be used to specify the estimated discrete filter

$$\hat{Fi}(z) = \frac{1 + c_1 z^{-1} + c_2 z^{-2} + ... + c_{n+m+1} z^{-1-n-m}}{1}, \tag{3.86}$$

Step 4: Filtered Instrumental Variable Parameter Estimation

The final step of the four step process is to recalculate the parameters using the instrumental variable parameter estimation formula while implementing the filter calculated from equation (3.86). The filter must be applied to the input and output signals wherever they occur in the instrumental variable formula to ensure that the input-output relationship is maintained

$$\hat{\theta}_4 = [\sum_{t=1}^{N} \xi_{Fi}(t) \Phi_{Fi}^T(t)]^{-1} [\sum_{t=1}^{N} \xi_{Fi}(t) y_{Fi}(t)] \tag{3.87}$$

$$\begin{aligned} \xi_{Fi}(t) = [-x_{Fi}(t-1) \quad & -x_{Fi}(t-2) \quad ... \quad -x_{Fi}(t-n) \quad u_{Fi}(t) \\ & u_{Fi}(t-1) \quad u_{Fi}(t-2) \quad ... \quad u_{Fi}(t-m)]^T. \end{aligned} \tag{3.88}$$

and

$$\begin{aligned} \Phi_{Fi}(t) = [-y_{Fi}(t-1) \quad & -y_{Fi}(t-2) \quad ... \quad -y_{Fi}(t-n) \quad u_{Fi}(t) \\ & u_{Fi}(t-1) \quad u_{Fi}(t-2) \quad ... \quad u_{Fi}(t-m)]^T. \end{aligned} \tag{3.89}$$

Filtered input and output signals can be calculated according to equations (3.66) and (3.67) with the estimated noise filter from equation (3.86) used in place of the ideal filter $Fi(z)$

$$u_{Fi}(t) = \hat{Fi}(z) u(t), \tag{3.90}$$

and

$$y_{Fi}(t) = \hat{Fi}(z) y(t). \tag{3.91}$$

The filtered instruments $x_{Fi}(t)$ can be calculated by filtering the best estimate of the model output $x_2(t)$ or by passing the filtered input $u_{Fi}(t)$ through the best estimate of the plant transfer function

$$x_{Fi}(t) = \hat{Fi}(z)x_2(t) = G_2(z)u_{Fi}(t). \tag{3.92}$$

3.3.3 Residual Generation and Evaluation

Residual generation and evaluation aims to detect and isolate the location and estimate the size of the fault. This can be done by comparing the RLS adapted model parameters with healthy parameters and analyzing the difference.

Fault Detection

A coefficient error vector can be formed by comparing the healthy model parameters (H) with the adapted model parameters (H_a) which may or may not contain a fault. Let the healthy model coefficient vector be defined as

$$H = [\ h_1 \quad h_2 \quad ... \quad h_n\ ] \in R^{n\times 1}, \tag{3.93}$$

where, n is the number of model parameters and let the adapted model coefficient vector be defined as

$$H_a = [\ h_{a1} \quad h_{a2} \quad ... \quad h_{an}\ ] \in R^{n\times 1}. \tag{3.94}$$

The coefficient error vector, or the residual, is then obtained as

$$R_e = D(H - H_a), \tag{3.95}$$

where,

$$D = diag\left[\frac{1}{h_1}, \frac{1}{h_2}, ..., \frac{1}{h_n}\right]. \tag{3.96}$$

The matrix D is used to normalize each coefficient. This addresses the cases where there is a large difference in the magnitude of coefficients. Detecting the presence of a failure can be accomplished by evaluating the magnitude of R_e and ensuring that

this error is not due to the standard variability of the system. Hence, a failure would be detected if

$$\|R_e\|_2 > \eta, \tag{3.97}$$

where,

$$\|R_e\|_2 = \sqrt{\sum_{i=1}^{n} |R_e(i,1)|^2}, i \in I_+. \tag{3.98}$$

Here η is the threshold value that represents the degree of variability of the system. This threshold can be calculated by statistical analysis of the estimated coefficients of the model.

In the present study a number of estimates of the coefficients are calculated from the simulated data. The mean of the values obtained from these datasets represents the true value of the coefficients and the standard deviation of each of the coefficient is used to determine η given by

$$\eta = \| \ 3\sigma_{ha1} \quad ... \quad 3\sigma_{han} \ \|_2. \tag{3.99}$$

The value of σ_{hai} corresponds to a 99.73% confidence interval assuming a normal distribution of obtained values of coefficients from the data.

Fault Isolation

The fault isolation method employed utilizes the direction of the coefficient error vector by evaluating the angles which the error vector forms with different axes of the n-dimensional fault space, where n axes correspond to n parameters of the model. A generalized representation of 3-dimensional fault space is shown in Figure 3.3 which shows three distinct fault spaces which could correspond to three different possible faults of the system. The origin of the fault space represents the healthy system and the radius of each fault space is a result of system variability possibly due to, among other things, sensor noise and accuracy of parameter estimation.

The fault vector can be written as

$$F = \sum_{j=1}^{n} r_j e_j, \tag{3.100}$$

where, r_j is the j^{th} component of R_e and e_j is the unit vector along the j^{th} axis of the fault space. The angles that this fault vector makes with different axes can be computed from the following equation

$$\gamma_j = \cos^{-1}\left(\frac{F.e_j}{|F|}\right), \tag{3.101}$$

where, γ_j is the angle that the fault vector makes with the j^{th} axis of the orthogonal system.

Then a fault signature code for a particular fault (say fault-1) becomes

$$\gamma_1^f, \gamma_2^f, ..., \gamma_n^f$$

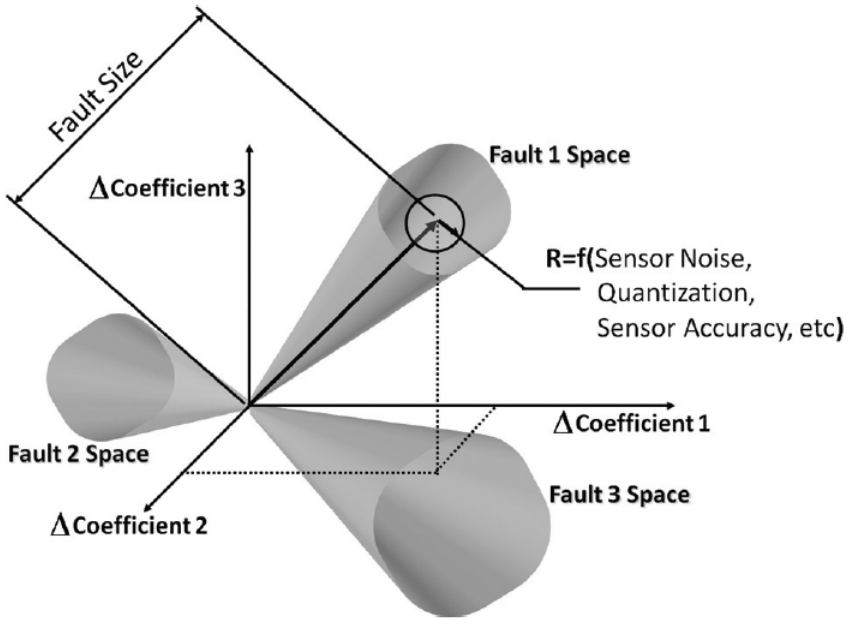

Figure 3.3: Fault Space.

This fault signature code is then compared with a set of predefined fault signature codes to isolate the fault.

Uniqueness of Fault Signature Code: The presence of a fault in the system changes the input output relationship by affecting the parameters of the system which in turn affects the fault error vector and the angles this vector makes with different axes of the fault space. The uniqueness of the fault signature code can be explained in terms of fault isolability which signifies that in order for a fault to be isolable its effect on the model parameter set should be different from other faults. This implies that isolable faults have unique fault signature codes.

Fault Signature Code Table: The fault signature code table consists of the faults and their fault signature codes and serves to identify the location of the fault. To create this table we need to have heuristic knowledge of the effect of the fault on different regressors of the model.

If a fault does not affect a particular regressor the error vector would be in a plane perpendicular to the coefficient axis corresponding to that fault and hence the angle would be 90 degrees otherwise the angle would be different from 90. In the presence of these faults we need to evaluate the exact magnitude of the angles of the fault signature code. To perform this we need to have knowledge of the physical equations governing the system and/or results of simulations and experiments. This information can be used to determine the exact direction of the coefficient error vector and hence angles produced by these faults.

Fault Estimation

Fault estimation is realized using the length of the coefficient error vector. The greater the length of the error vector, the greater or more severe is the size of the fault.

The length of the fault vector can be computed by

$$L_{mag} = \sqrt{\sum_{j=1}^{n} r_j^2}, \tag{3.102}$$

where r_j are the components of the coefficient error vector. Here each component of the fault vector has been normalized.

3.3.4 Sensitivity Analysis

A sensitivity analysis can be performed to quantify the effect of the uncertainities in the parameter estimates due to noise and the disturbances acting on the system. The variance of the parameters will have an effect on the angles that the fault vector forms with different axes and the length of the vector itself. Sensitivity analysis gives a measure of the robustness of the FDIE methodology. The smaller the effect of parameter variance on the fault angles and the fault length, the greater is the robustness with respect to noise and disturbances. These effects can be computed by calculating the partial differentiation of the angles and the length vectors with respect to each of the parameter vectors.

Obtaining the relation from equation (3.101) for the angles,

$$\gamma_j = \cos^{-1}\left(\frac{F.e_j}{|F|}\right). \tag{3.103}$$

Here,

$$F.e_j = r_j, \tag{3.104}$$

and

$$|F| = \sqrt{\sum_{j=1}^{n} r_j^2}. \tag{3.105}$$

Hence, by differentiating the angle equation with respect to the parameter h_{ai} we obtain,

$$\frac{\partial \gamma_j}{\partial h_{ai}} = -\frac{1}{\sqrt{1-\left(\frac{r_j}{|F|}\right)^2}} \frac{\partial\left(\frac{r_j}{|F|}\right)}{\partial h_{ai}}. \tag{3.106}$$

Using equation (3.106) and the above definition of $|F|$, we get,

$$\frac{\partial \gamma_j}{\partial h_{ai}} = -\frac{1}{\sqrt{1-\left(\frac{r_j}{|F|}\right)^2}} \left[-\frac{h_j \delta_{ij}}{|F|} - \frac{r_j \times r_i}{\sqrt[1.5]{\sum_{j=1}^{n} (r_j)^2}}\right], \tag{3.107}$$

where, δ_{ij} is kronecker delta that is, $\delta_{ij} = 1$ for $i = j$ and $\delta_{ij} = 0$ for $i \neq j$.

Hence, we can calculate the influence of all parameters h_{ai} for $i = (1, ..., n)$ on each γ_j.

Then the effect of the variance of parameters on angle γ_j is given by

$$\Delta(\gamma_j) = \sum_{i=1}^{n} \left[\frac{\partial \gamma_j}{\partial h_{ai}} \Delta h_{ai} \right], \tag{3.108}$$

for $j = (1, ..., n)$, where $\Delta h_{ai} = 3\sigma_{hai}$.

This provides the sensitivity of angles of the fault vector with the variation in the estimated parameters. Similarly the sensitivity of the length of fault vector can be calculated in the following manner

$$\frac{\partial L_{mag}}{\partial h_{ai}} = -\frac{r_i}{\sqrt{\sum_{j=1}^{n} r_j^2}}, \tag{3.109}$$

and

$$\Delta(L_{mag}) = \sum_{i=1}^{n} \left[\frac{\partial L}{\partial h_{ai}} \Delta h_{ai} \right], \tag{3.110}$$

where again, $\Delta h_{ai} = 3\sigma_{hai}$.

This provides the sensitivity of the fault vector length with the variation in the estimated parameters.

3.3.5 Method Summary

The fault detection, isolation and estimation method proposed in the present work is realized using the on-line model and analyzing the effect of faults on the parameters of the model of the system. A flow chart of the FDIE methodology and its implementation is shown in Figure 3.4. A number of healthy data sets are used to form a family of healthy model parameters and to calculate the standard deviation and hence the threshold (equation 3.97). Once the healthy model parameters are obtained it is used as baseline model against which all the future models are compared. If a fault is present in the system a warning signal will be flagged by the algorithm, which

for the case of OBD in automobiles, is known as malfunction indicator light (MIL) and is situated at the vehicle dashboard. On the other hand if IS does not diagnose a fault, new on-line data will be used to perform the algorithm once again. This process can be applied in a continuous fashion where the algorithm will continuously estimate the parameters of the system model.

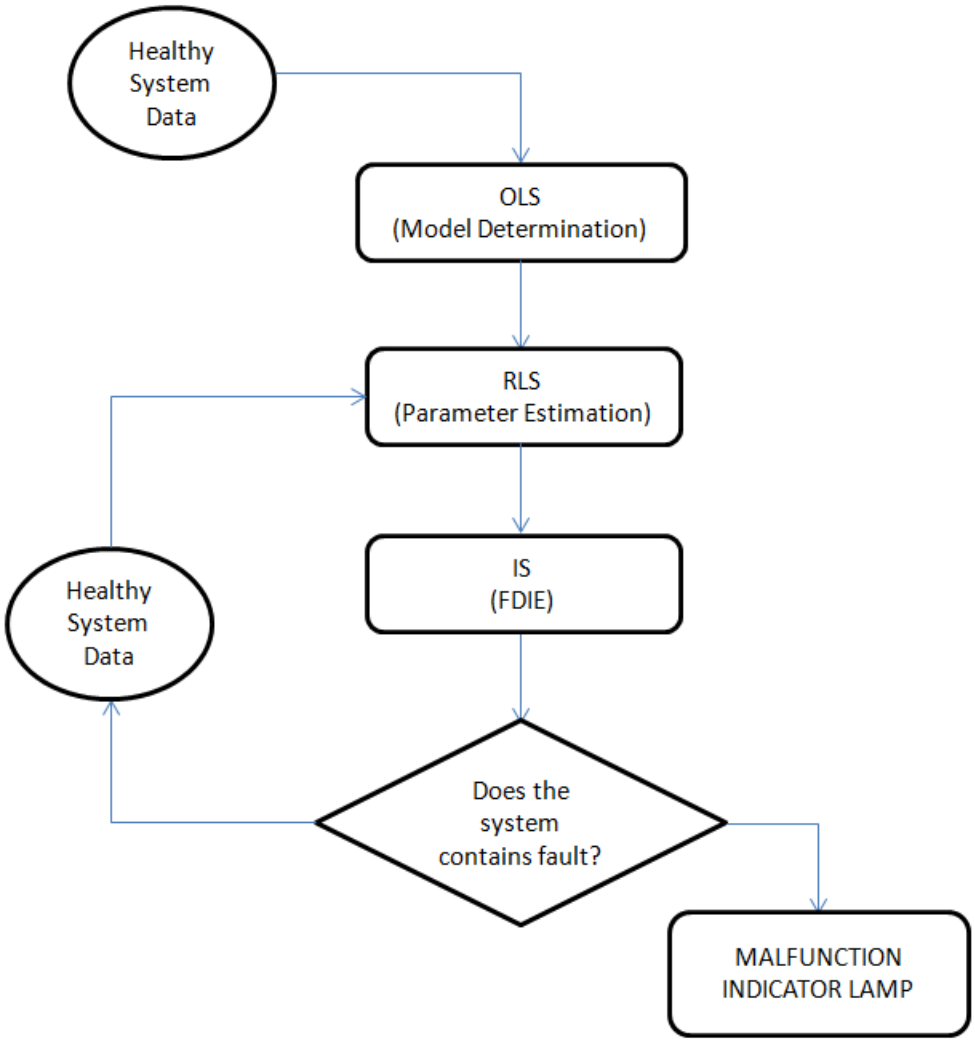

Figure 3.4: Flow Chart of IS Based Model Development and FDIE.

3.4 Adaptive Model Based FDIE Illustration

A double mass-spring-damper system is represented in Figure 3.5. The force $F_{ex}(t)$ is the input $u(t)$ of the system. The left and right mass equal ma_1 and ma_2, respectively. The output of the system consists of both the displacement $x_1(t)$ of mass ma_1 and the displacement $x_2(t)$ of mass ma_2. The spring constants are k_1 and k_2. The damper constants are b_1 and b_2.

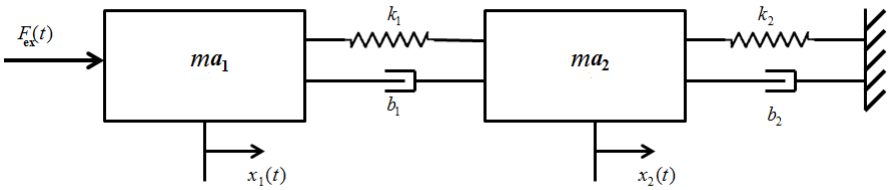

Figure 3.5: Mass-Spring-Damper System.

The equations of motion resulting from force balance are:

$$F_{ex} - k_1(x_1 - x_2) - b_1(\dot{x}_1 - \dot{x}_2) = ma_1\ddot{x}_1, \tag{3.111}$$

$$k_1(x_1 - x_2) + b_1(\dot{x}_1 - \dot{x}_2) - k_2x_2 - b_2\dot{x}_2 = ma_2\ddot{x}_2. \tag{3.112}$$

Written in matrix notation,

$$\begin{bmatrix} ma_1 & 0 \\ 0 & ma_2 \end{bmatrix} \begin{bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{bmatrix} + \begin{bmatrix} b_1 & -b_1 \\ -b_1 & b_1 + b_2 \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} + \begin{bmatrix} k_1 & -k_1 \\ -k_1 & k_1 + k_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} F_{ex} \\ 0 \end{bmatrix}, \tag{3.113}$$

with

$$ma_1 = 5 \quad k_1 = 25 \quad b_1 = 5,$$
$$ma_2 = 3 \quad k_2 = 100 \quad b_2 = 5.$$

Here, the system's uncertain parameters are usually k_1 and k_2. The goal of proposed FDIE methodology in this context is to detect the change in the system due

to uncertainities and find the direction and magnitude of change. In this case, the spring constants k_1 and k_2 take the values as shown in *Table* 3.1.

Table 3.1: Fault Codes and Corresponding Spring Constants.

Fault type	Fault code	Spring constant(k_1)	Spring constant(k_2)
Healthy system	k_0	25	100
Fault 1 - k_1	k_1^1	50	100
Fault 2 - k_1	k_1^2	60	100
Fault 3 - k_2	k_2^1	25	125
Fault 4 - k_2	k_2^2	25	150

Taking laplace transform,

$$\begin{bmatrix} ma_1s^2 + b_1s + k_1 & -b_1s - k_1 \\ -b_1s - k_1 & ma_2s^2 + (b_1 + b_2)s + (k_1 + k_2) \end{bmatrix} \begin{bmatrix} X_1(s) \\ X_2(s) \end{bmatrix} = \begin{bmatrix} F_{ex}(s) \\ 0 \end{bmatrix},$$

$$\begin{bmatrix} X_1(s) \\ X_2(s) \end{bmatrix} = \begin{bmatrix} ma_1s^2 + b_1s + k_1 & -b_1s - k_1 \\ -b_1s - k_1 & ma_2s^2 + (b_1 + b_2)s + (k_1 + k_2) \end{bmatrix}^{-1} \begin{bmatrix} F_{ex}(s) \\ 0 \end{bmatrix}.$$

With external input, the mass oscillates. As the system is stable, it converges to steady state value. To satisfy the persistence of excitation lemma, the input is randomly changing. On the other hand, output noise is also present in the system. Let the displacement of mass m_1 which is $x(t)$ be the output signal. In this example, two second order systems are coupled, the effect of which is that there are four poles. So, the individual transfer function of $X_1(s)$ and $X_2(s)$ is fourth order. From the derivation to solve for $X_1(s)$ and $X_2(s)$, it is noted that four coefficients for each transfer function are dependent on k_1 and k_2 as

$$X_1(s) = \frac{a_1's^2 + a_2's + a_3'}{a_4's^4 + a_5's^3 + a_6's^2 + a_7's + a_8'} F_{ex}(s), \tag{3.114}$$

$$X_2(s) = \frac{a_1''s^2 + a_2''s + a_3''}{a_4''s^4 + a_5''s^3 + a_6''s^2 + a_7''s + a_8''} F_{ex}(s), \tag{3.115}$$

where a'_3, a'_6, a'_7 and a'_8 in (3.114) and a''_3, a''_6, a''_7 and a''_8 in (3.115) depend on k_1 and k_2. The healthy model coefficients for x_1 are calculated using recursive least squares algorithm and are shown in Figure 3.6. The estimated output is shown to precisely track the actual output (Figure 3.7).

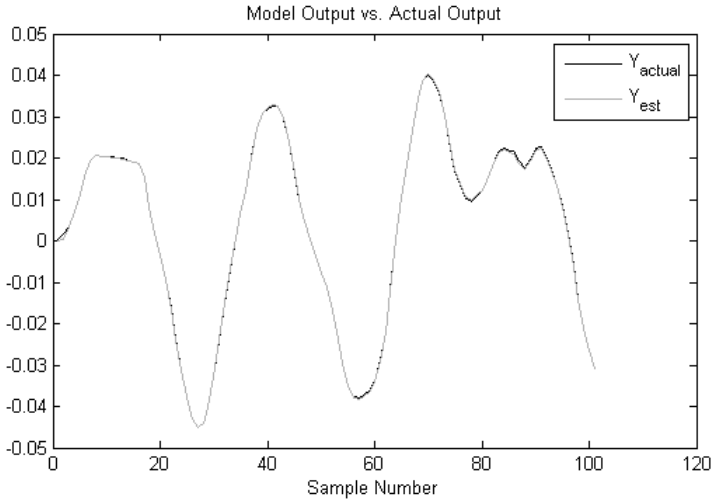

Figure 3.6: Plot of Actual Output vs Model Output - Healthy System.

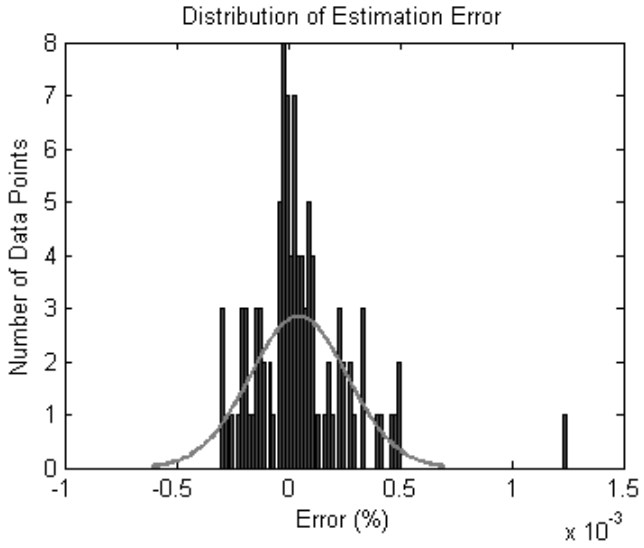

Figure 3.7: Error Histogram for Healthy System.

3.4.1 Information Synthesis

The diagnostic model derived above can now be used to detect, isolate and estimate the fault by adapting the parameters of the model and analyzing the changes. The results of IS on a double mass-spring-damper system is presented.

While considering two faults by changing k_1 and two faults by changing k_2, it produces four sets of model coefficients for each case apart from the healthy system coefficient set. The mean value in each set was considered to be the true value and standard deviation was calculated for the coefficients of the heathy system. Using the least squares estimation technique described earlier, the healthy and adapted coefficients are calculated for (3.114) as shown in *Table* 3.2.

Table 3.2: Coefficients for x_1.

Fault code	a'_1	a'_2	a'_3	a'_4	a'_5	a'_6	a'_7	a'_8
k_0	0.2	0.6	8.5	1	4.5	48.1	41.6	166.5
k_1^1	0.2	0.5	8.4	1	2.8	51.4	39.2	312.1
k_1^2	0.2	0.6	8.6	1	2.6	54.4	40.2	367.8
k_2^1	0.2	0.7	10.1	1	4.8	55.7	52.4	207.4
k_2^2	0.2	0.8	11.9	1	5.4	65.2	64.8	254.3

The threshold for the dataset is given by equation (3.99) for normalized adapted coefficients a'_3, a'_6, a'_7 and a'_8 and is calculated as 0.05878. Now the error vector defined in equation (3.95). For k_1^1 fault, the error vector is given by $R = [0.01070.05110.04040.6428]^T$ and $\|R\|_2 = 0.6462$. Since $\|R\|_2 > \varepsilon$, a fault is present. For a k_1^2 fault, the error vector is given by $R = [0.01340.09710.02380.8887]^T$ and $\|R\|_2 = 0.8945$. Since $\|R\|_2 > \varepsilon$, a fault is present. Similarly, $\|R\|_2 = 0.3144$ and 0.6821 for k_2^1 and k_2^1 faults respectively and hence, the fault is present.

Shown in Figure 3.8 is the error vector plot of the faulty systems in the fault space. Failure estimation can be performed by analyzing the magnitude of the error

vector as can be seen in the dataset. The magnitude of fault k_1^2 is larger than the fault k_1^1 and the magnitude of fault k_2^2 is larger than the fault k_2^1. Shown in Figure 3.9 is the 4-D (3-D with color bar) plot of the coefficients of all the system models for the dataset.

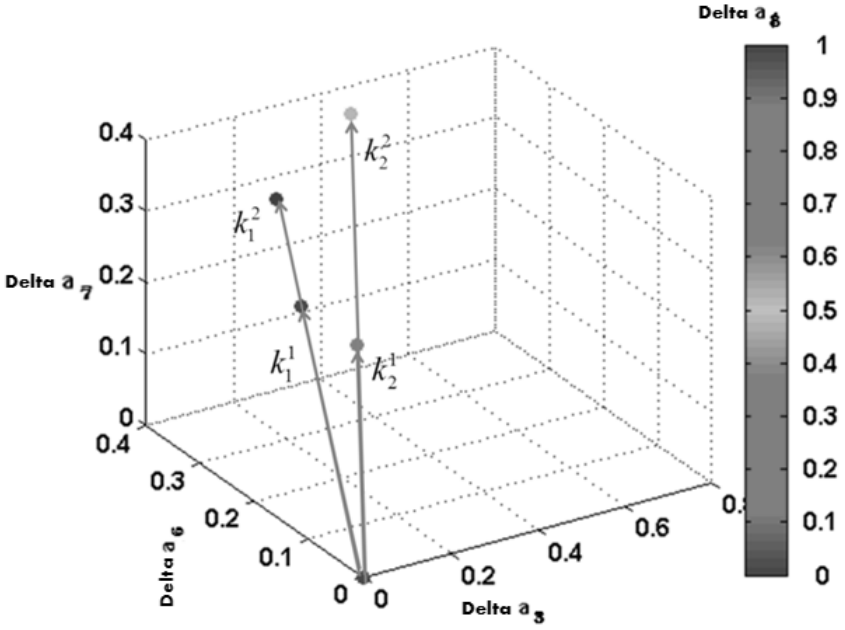

Figure 3.8: Error Vector in Fault Space.

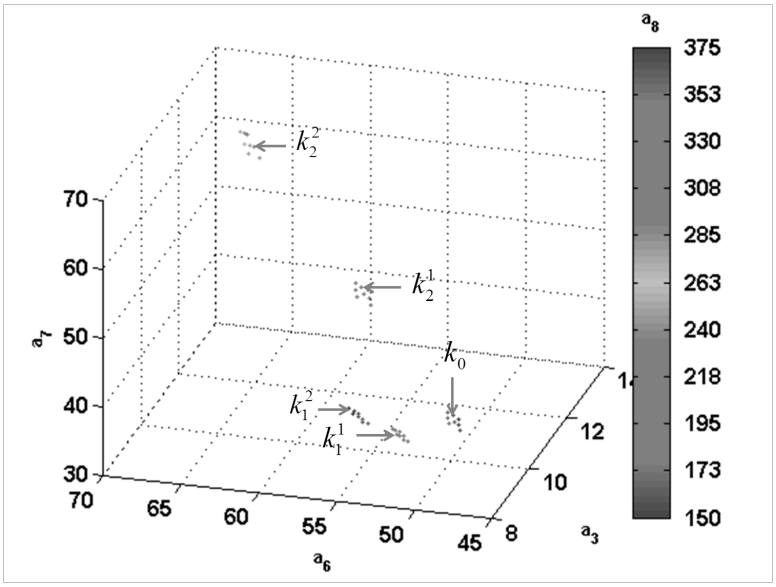

Figure 3.9: Plot of Adapted Model Coefficients.

3.4.2 Sensitivity Analysis

Sensitivity is used to evaluate the robustness of the proposed FDIE methodology. Sensitivity analysis gives a quantitative picture of the effect of unwanted variables such as sensor noise and external disturbances on the detection and diagnostic process. The variance of the estimated parameters caused by these unwanted variables in turn affects the fault signature code angles and the error vector magnitude. Shown in *Table* 3.3 the fault signature code of the spring constant faults calculated using equation (3.101).

Table 3.3: Fault Signature Code - Spring Constant Faults.

Fault code	$\gamma_{a_3'}(^o)$	$\gamma_{a_6'}(^o)$	$\gamma_{a_7'}(^o)$	$\gamma_{a_8'}(^o)$
k_1^1	89.1	85.4	86.4	5.8
k_1^2	89.2	83.7	88.5	6.4
k_2^1	63.8	67.9	54.6	54.9
k_2^2	63.4	70.6	55.1	55.3

The sensitivities of the angles and error vector length to variation in model parameters are shown in *Table* 3.4 and *Table* 3.5 and are calculated using equations (3.108) and (3.110) respectively. These sensitivity values show the uncertainty in the angles and the error vector length due to uncertainty in the parameter estimation.

Table 3.4: Fault Signature Code Sensitivity - Spring Constant Faults.

Fault code	$\Delta\gamma_{a'_3}(^o)$	$\Delta\gamma_{a'_6}(^o)$	$\Delta\gamma_{a'_7}(^o)$	$\Delta\gamma_{a'_8}(^o)$
k_1^1	1.0	0.7	2.1	2.8
k_1^2	1.4	1.1	3.1	1.6
k_2^1	0.5	0.3	1.3	0.5
k_2^2	6.1	4.1	7.1	6.2

Table 3.5: Error Vector Length Sensitivity - Spring Constant Faults.

Fault code	$\|\Delta R\|_2 \times 100$
k_1^1	9.2
k_1^2	11.1
k_2^1	5.6
k_2^2	37.2

It can be seen from *Table* 3.4 that the sensitivity of angles towards adapted coefficients is more with increase in the magnitude of fault. Similar observation is made from *Table* 3.5. The small values of angle and magnitude sensitivity proves the robustness of FDIE method towards parametric uncertainty and variations. Consequently, it is also shown that the error vector contains the information of the size of fault.

Chapter 4
OBSERVER BASED ESTIMATION AND APPLICATIONS

In this chapter, an observer-based method for consistent parameter estimates and the utility of these estimations in real-time model based diagnostics is presented. The chapter is divided into two main parts: finite time state estimation in discrete systems and observer based parameter estimation. The class of discrete systems considered includes linear piecewise-continuous systems that are observable. Parameter estimation is realized through a calculation involving the states from the combination of Luenberger observers. After the estimation, Information Synthesis will be shown to be an accurate diagnostic method to perform fault detection, isolation and estimation tasks and that meets the requirement of On Board Diagnostics. The utility of the proposed method is demonstrated on an experimental application of an engine onboard sensor diagnostics.

4.1 Finite Time State Estimation in Discrete Systems

4.1.1 System Description

The class of discrete systems considered include linear observable piecewise-continuous systems. In the case when the system is not Hurwitz, the system must also be controllable. The piecewise continuous condition arises from the diagnostics application of this work. In particular, the sudden failure of a subsystem produces a discontinuous change in the system dynamics thereby requiring the piecewise condition. The slowly time varying changes in the system dynamics is due to evolutionary-type of system aging and is thus continuous.

Let the true system model whose parameters are to be identified be defined as the

linear discrete system expressed in *controllable canonical* form

$$\begin{aligned} x(i+1) &= Ax(i) + Bu(i), \\ y(i) &= Cx(i), \end{aligned} \tag{4.1}$$

where

$$A = [A_o + \Delta A], \tag{4.2}$$

$$C = [C_o + \Delta C], \tag{4.3}$$

$A, A_o \in \mathbb{R}^{n\times n}$ are Hurwitz, $[A, C]$ and $[A_o, C_o]$ are observable and $C, C_o \in \mathbb{R}^{1\times n}$. A_o and C_o denote the nominal healthy system dynamics, and the perturbations ΔA and ΔC are due to system uncertainties, failures and/or aging. The description in (4.1) relies on the usual discrete system notation of

$$x(i) = x(t)|_{t=iT}, \tag{4.4}$$

where T is the sampling interval in units of time and $i \in I_+$. The remaining variables in (4.1) have dimensions that are consistent with those defined above.

The full state observer for (4.1) is defined as

$$\begin{aligned} \hat{x}(i+1) &= A_o\hat{x}(i) + Bu(i) + K_e(y(i) - C_o\hat{x}(i)), \\ \hat{y}(i) &= C_o\hat{x}(i), \end{aligned} \tag{4.5}$$

where $\hat{x}(i)$ is the estimated state and $C_o\hat{x}(i)$ is the estimated output. It looks like a copy of the state equation driven by an error term $y(i)-\hat{y}(i)$ that enters the dynamics through an $n \times m$ *observer gain matrix* K_e (n=Number of states; p=Number of outputs). This error term is intended to drive the state estimate $\hat{x}(i)$ to the actual state $x(i)$ over time. To further explore this key convergence issue, we define the *estimation error* $\tilde{x}(t) = x(t) - \hat{x}(t)$ in terms of which we derive the *error dynamics*. Regardless of the system given in (4.1), the observer will be chosen as stable, namely $A_o - K_eC_o$ is Hurwitz. The only unknowns appearing in (4.2) and (4.3) are namely

ΔA and ΔC.

Convergence of the observer defined in (4.5) is shown by developing the state estimation error vector

$$\begin{aligned} \hat{e}(i+1) &= x(i+1) - \hat{x}(i+1), \\ &= (A_o - K_e C_o)e(i) + (\Delta A - K_e \Delta C)x(i), \end{aligned} \tag{4.6}$$

where $(A_o - K_e C_o)$ is Hurwitz by the design of K_e [74],[75]. Thus provided $|x(i)| < \infty$ then $|\hat{e}(i)| < \infty$. Moreover, if the eigenvalues of the observer are faster than the system dynamics, state convergence can be achieved.

4.1.2 Primer: Finite Time Observers

Reference [75] considers the problem of finite-time functional observer design for linear discrete-time descriptor systems, which converges in finite time exactly. Consider the following linear discrete-time singular system:

$$\begin{aligned} Ex(i+1) &= Ax(i) + Bu(i) + Df(i), \\ y(i) &= Cx(i), \\ z(i) &= Lx(i), x(0) = x_0, \end{aligned} \tag{4.7}$$

with state $x \in \mathbb{R}^n$, control input $u \in \mathbb{R}^m$, control input $f \in \mathbb{R}^q$, measurable output $y \in \mathbb{R}^p$, and estimated output $z \in \mathbb{R}^r$. All matrices are real and of appropriate dimensions. Reader is refered to [75] for explaination of all the associated notations. In order to design an observer which converges in finite time, the following functional observer for system (4.7) in infinite time interval:

$$\begin{aligned} \zeta(i+1) &= \hat{A}\zeta(i) + \hat{B}y(i) + \hat{J}u(i), \\ \hat{z}(i) &= \zeta(i) + \hat{D}y(i), \end{aligned} \tag{4.8}$$

is defined where $\zeta \in \mathbb{R}^r$ is the state vector and $\hat{z} \in \mathbb{R}^r$ is the estimate of z. $\hat{A}, \hat{B}, \hat{J}$ and $\hat{D}$ are unknown matrices of appropriate dimensions, which are to be determined

such that $\hat{z}$ asymptotically converges to z.

In order to prove that the infinite time functional observer states error will asymptotically converge to zero, a matrix M should exist such that the following conditions hold.

(c1) $\hat{A}ME + \hat{B}C = MA$

(c2) $\hat{J} = MA$

(c3) $MD = 0$

(c4) $ME = L - \hat{D}C$

(c5) $\zeta(i+1) = \hat{A}\zeta(i)$

Under these conditions, there exists a matrix parameter Z such that $\hat{A} = \Lambda - Z\Gamma$ is Hurwitz, where,

$$\Lambda = LAH_1 - \Theta\Sigma^+ \begin{bmatrix} CAH_1 \\ CH_1 \\ EAH_1 \end{bmatrix}, \tag{4.9}$$

$$\Gamma = (I - \Sigma\Sigma^+) \begin{bmatrix} CAH_1 \\ CH_1 \\ EAH_1 \end{bmatrix}. \tag{4.10}$$

Here, the terminology is explained below,

$$E = E - I, \tag{4.11}$$

$$\begin{bmatrix} H_1 & E_1 \end{bmatrix} = \begin{bmatrix} L \\ G \end{bmatrix}^{-1}, \tag{4.12}$$

$$\Sigma = \begin{bmatrix} CAE_1C & 0 & \\ CE_1 & 0 & 0 \\ \bar{E}AE_1 & E & D \end{bmatrix}, \tag{4.13}$$

$$\Theta = \begin{bmatrix} LAE_1 & L & 0 \end{bmatrix}, \tag{4.14}$$

and Σ^+ is the generalized inverse matrix of Σ as,

$$\Theta\Sigma^+\Sigma = \Theta. \tag{4.15}$$

By selecting the Z_1 and Z_2 such that observer state matrices $\hat{A}_k = \Lambda - Z_k\Gamma$ are stable, where $k = 1, 2$, they have designed two infinite time functional observers:

$$\begin{aligned} \zeta_k(i+1) &= \hat{A}_k\zeta_k(i) + \hat{B}_k y(i) + \hat{J}_k u(i), \\ \hat{z}_k(i) &= \zeta_k(i) + \hat{D}_k y(i), \end{aligned} \tag{4.16}$$

where $k = 1, 2$ and

$$\hat{A} := \begin{bmatrix} \hat{A}_1 & 0 \\ 0 & \hat{A}_2 \end{bmatrix}, \hat{B} := \begin{bmatrix} \hat{B}_1 \\ \hat{B}_2 \end{bmatrix}, \hat{J} := \begin{bmatrix} \hat{J}_1 \\ \hat{J}_2 \end{bmatrix},$$

$$\hat{D} := \begin{bmatrix} \hat{D}_1 \\ \hat{D}_2 \end{bmatrix}, N := \begin{bmatrix} \hat{I}_r \\ \hat{I}_r \end{bmatrix}, \hat{z} := \begin{bmatrix} \hat{z}_1 \\ \hat{z}_2 \end{bmatrix}, \zeta := \begin{bmatrix} \zeta_1 \\ \zeta_2 \end{bmatrix}.$$

Key result: The two infinite observers are combined, and with use of delay d, a new estimate $\check{z}$ is generated.

$$\zeta(i+1) = \hat{A}\zeta(i) + \hat{B}y(i) + \hat{J}u(i), \tag{4.17}$$

$$\hat{z}(i) = \zeta(i) + \hat{D}y(i), \tag{4.18}$$

$$\check{z}(i) = K[\hat{z}(i) - \hat{A}^d\hat{z}(i-d)]. \tag{4.19}$$

Theorem 4.1 Assume $\hat{A}$ and d can be chosen such that

(a1) $\hat{A}$ is stable

(a2) $det[N\hat{A}^dN] \neq 0$.

Then, (4.19) with $K = [\ I_r \ \ 0\ ][\ N \ \ e^{\hat{A}d}N]^{-1}$, is an observer for system (4.7), whose output $\check{z}(i)$ converges to $z(i)$ in finite time d.

Theorem 4.2 If Z_1, Z_2 are chosen such that $\hat{A}_1$ is nonsingular and

$$Re\lambda_j(\hat{A}_2) < \kappa < Re\lambda_j(\hat{A}_1), j = 1, 2, ..., r. \tag{4.20}$$

for some $\kappa \in (0,1)$, then, there exists a positive integer d_o such that for $d > d_o$, $det[N\hat{A}^d N] \neq 0$.

Since $\hat{z}_k(i+1) - z(i+1) = \hat{A}_k(\hat{z}_k(i) - z(i))$ for $k = 1, 2$, we have

$$\hat{z}(i+1) - Nz(i+1) = \hat{A}(\hat{z}(i) - Nz(i)), i \geq 0, \tag{4.21}$$

Therefore

$$\hat{z}(i) - Nz(i) = \hat{A}^d[\hat{z}(i-d) - Nz(i-d)], k \geq d. \tag{4.22}$$

Using the fact $KN = I$ and $K\hat{A}^d N = 0$, we get

$$\begin{aligned} \check{z}(i) &= K[\hat{z}(i) - \hat{A}^d\hat{z}(i-d)], & (4.23)\\ &= z(i) + K[\hat{z}(i) - Nz(i)] - K\hat{A}^d[\hat{z}(i-d) - Nz(i-d)]. & (4.24) \end{aligned}$$

This implies $\check{z}(i) = \hat{z}(i)$ for $i \geq d$.

The proofs for the theorems are presented in Appendix B.

4.1.3 Proposed Finite Time Observer

The problem with using the method of two infinite functional observers is that the matrices A and C are unknown. In order to determine them, the calculated estimate is expressed in the form of Luenberger observer with stable observer state matrices $\hat{A}_k = A_o - K_{ek}C_o$. It should be noted that the proposed method is analoguous to that presented in [75](presented in section 4.1.2). The analogy exists because of the fact that the key result and the two theorems described above are valid for the proposed approach as well. For defining the observer state matrix, the matrices are replaced as

$$\Lambda \rightarrow A_o, \tag{4.25}$$

$$\Gamma \rightarrow C_o, \tag{4.26}$$

$$Z_k \rightarrow K_{ek}. \tag{4.27}$$

The infinite observers can be redefined in Luenberger observer form as:

$$\begin{aligned} \hat{x}(i+1) &= \hat{A}\hat{x}(i) + K_e y(i) + Bu(i), \\ \hat{z}(i) &= \hat{x}(i) - \hat{D}y(i), & (4.28) \end{aligned}$$

where $\hat{x} \in \mathbb{R}^r$ is the state vector and $\hat{y}(k) \in \mathbb{R}^r$ is the estimate of y. $\hat{A}$, B and $\hat{D}$ are unknown matrices of appropriate dimensions, which are to be determined such that $\check{z}$ asymptotically converges to z.

By selecting the K_{e1} and K_{e2} such that observer state matrices $\hat{A}_k = A_o - K_{ek}C_o$ are stable, where $k = 1, 2$, two infinite time Luenberger observers are designed:

$$\begin{aligned} \hat{x}_k(i+1) &= \hat{A}_k\hat{x}_k(i) + K_{ek}y(i) + B_k u(i), \\ \hat{z}_k(i) &= \hat{x}_k(i) - \hat{D}_k y(i), \end{aligned} \tag{4.29}$$

where $k = 1, 2$ and

$$\hat{A} := \begin{bmatrix} \hat{A}_1 & 0 \\ 0 & \hat{A}_2 \end{bmatrix}, \hat{B} := \begin{bmatrix} \hat{B}_1 \\ \hat{B}_2 \end{bmatrix}, \hat{D} := \begin{bmatrix} \hat{D}_1 \\ \hat{D}_2 \end{bmatrix},$$

$$N := \begin{bmatrix} I_r \\ I_r \end{bmatrix}, \hat{z} := \begin{bmatrix} \hat{z}_1 \\ \hat{z}_2 \end{bmatrix}, \hat{x} := \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix}.$$

The two infinite observers are combined and with use of delay d, a new estimate $\check{Z}$ is generated.

$$\begin{aligned} \hat{x}(i+1) &= \hat{A}\hat{x}(i) + K_e y(i) + Bu(i), \\ \hat{z}(i) &= \hat{x}(i) + \hat{D}y(i), \\ \check{Z}(i) &= K\left[\begin{bmatrix} \hat{z}_1(i) \\ \hat{z}_2(i) \end{bmatrix} - \begin{bmatrix} (A_o - K_{e1}C_o)^d & 0 \\ 0 & (A_o - K_{e2}C_o)^d \end{bmatrix}\begin{bmatrix} \hat{z}_1(i-d) \\ \hat{z}_2(i-d) \end{bmatrix}\right]. \end{aligned} \tag{4.30}$$

Again, $\hat{A}$ and d can be chosen such that $\hat{A}$ is stable and $det[N\hat{A}^d N] \neq 0$. Then, with $K = [\ I_r \ \ 0\ ][\ N \ \ e^{\hat{A}d}N]^{-1}$, it is an observer for system (4.7), whose output $\check{Z}(i)$ converges to $z(i)$ in finite time d. The choice for K_{e1}, K_{e2} is done such that

$$Re\lambda_j(\hat{A}_2) < \kappa < Re\lambda_j(\hat{A}_1), j = 1, 2, ..., r. \tag{4.31}$$

For some $\kappa \in (0, 1)$, then, there exists a positive integer d_o such that for $d > d_o$, $det[N\hat{A}^d N] \neq 0$.

In order to show the observed states' convergence, the above method is simulated.

The estimate thus converges in finite time equivalent to delay d as shown in Figure 4.1.

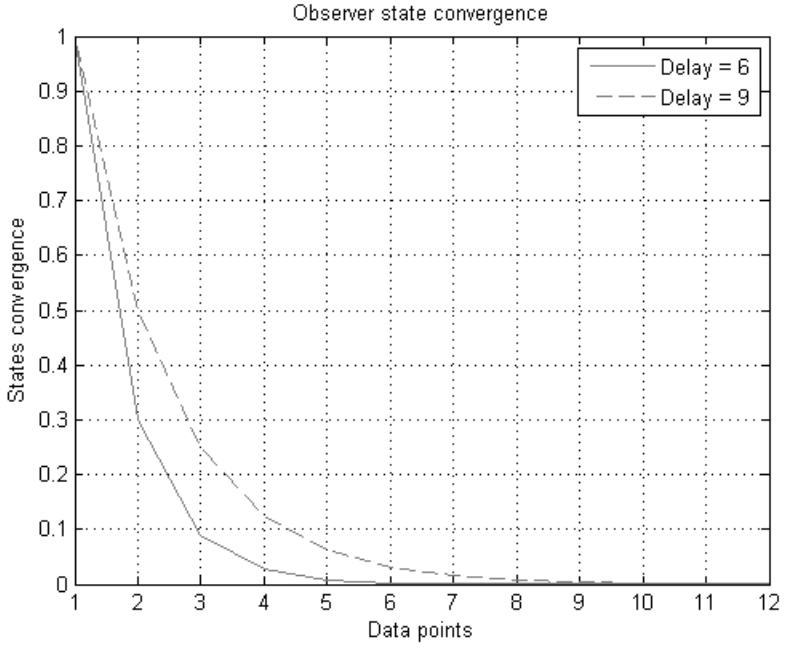

Figure 4.1: The Observer States Convergence.

The convergence is same as that of using the method of [75] which implies that by using the two infinite-time Luenberger observers, an exact convergence in finite time can be achieved. With the finite-time convergence of known estimated states, parametric uncertainty quantification can now be discussed.

4.1.4 Computation of Parametric Uncertainty

This parametric uncertainty is based on simplify the expressions for $\Delta\bar{A}$ and ΔC done in the following development. To identified the parametric uncertainties in (4.1), the estimated states defined in (4.5) are used to develop the following observed equation,

$$\begin{aligned} \hat{x}(i+1) &= A\hat{x}(i) + Bu(i), \\ y(i) &= C\hat{x}(i), \end{aligned} \tag{4.32}$$

where the state estimates have converged in finite time (Section 4.1.2) and are known. Using definition of A and C from (4.2) and (4.3),

$$\begin{aligned} \hat{x}(i+1) &= (A_o + \Delta A)\hat{x}(i) + Bu(i), \\ y(i) &= (C_o + \Delta C)\hat{x}(i). \end{aligned} \tag{4.33}$$

From (4.33), the parametric uncertainty in the state matrices, ΔA and ΔC, can be calculated as

$$\Delta A = (\hat{x}(i+1) - A_o\hat{x}(i) - Bu(i))\hat{x}^T(i)(\hat{x}(i)\hat{x}^T(i))^{-1}, \tag{4.34}$$

$$\Delta C = (y(i) - C_o\hat{x}(i))\hat{x}^T(i)(\hat{x}(i)\hat{x}^T(i))^{-1}, \tag{4.35}$$

provided the condition that $|\hat{x}(i)\hat{x}^T(i)| \neq 0$, that is $[\hat{x}(i)\hat{x}^T(i)]^{-1}$ is invertible, persists for ΔA and ΔC calculation.

Remark: Note that the estimated state perturbation matrix ΔA can be calculated from the observer states using the data at one time sample. The impact of this means that the online computational needs for the parametric uncertainty calculation are reduced to that associated with the observer and the matrix calculation at one time sample in (4.34). Furthermore the persistency of excitation condition required for consistent parameter estimates has been reduced to the data at one time sample. In contrast, standard least squares estimation solutions require persistency of excitation over the data segment when estimating model parameters. Finally, note that the computations in (4.34) can be performed recursively.

4.1.5 Sensitivity Analysis

Sensitivity towards noise and input can be calculated for systems consisting of observers by first deriving the transfer function between control effort and observer error. The system is shown in *Figure* 4.2.

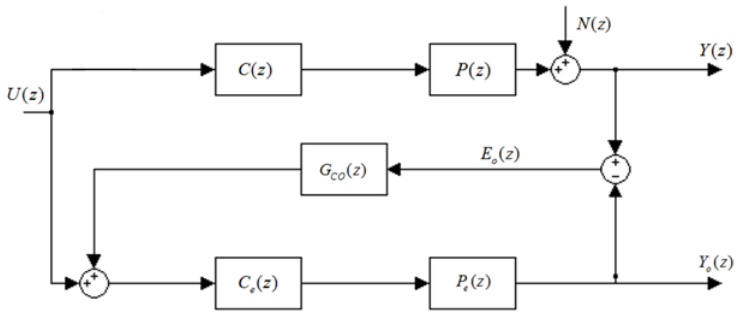

Figure 4.2: The Luenberger Observer in the Presence of Sensor Noise.

Simplification of the above block diagram gives the transfer function as

$$E_o(z) = \frac{C(z)P(z) - C_e(z)P_e(z)}{1 + G_{CO}(z)C_e(z)P_e(z)}U(z) + \frac{1}{1 + G_{CO}(z)C_e(z)P_e(z)}N(z). \tag{4.36}$$

Derivation: The equations directly obtained from the block diagram are:

(a1) $E_o(z) = Y(z) - Y_o(z)$

(a2) $U(z) + G_{CO}(z)E_o(z) = C_e(z)P_e(z)Y_o(z)$

(a3) $N(z) + C(z)P(z)U(z) = Y(z)$

By using (a2) and (a3), the equations for $Y_o(z)$ and $Y(z)$ are put in (a1),

$$E_o(z) = N(z) + C(z)P(z)U(z) + \frac{U(z) + G_{CO}(z)E_o(z)}{C_e(z)P_e(z)} \tag{4.37}$$

After simplifying the above equation, we get the same result as (4.33).

Note: Both $G_{CO}(z)$ and K_e are equivalent. The transformation from observer transfer function to state space form will change the $G_{CO}(z)$ and K_e and vice versa.

From this schematic diagram, the transfer function for observer is derived as,

$$Y_o(z) = \frac{P(z)C(z)}{1 + G_{CO}(z)P(z)C(z)}U(z) \tag{4.38}$$

This transfer function can be obtained from state space as,

$$C_o(zI - (A_o - K_eC_o))^{-1}B \tag{4.39}$$

Thus, these two are related as,

$$\frac{P(z)C(z)}{1+G_{CO}(z)P(z)C(z)} \Leftrightarrow C_o(zI-(A_o-K_eC_o))^{-1}B \tag{4.40}$$

So, $G_{CO}(z)$ and K_e can be found if any one of it is known.

So, the transfer functions related to low and high frequency respectively are

$$T_U(z) = \frac{C(z)P(z)-C_e(z)P_e(z)}{1+G_{CO}(z)C_e(z)P_e(z)}, \tag{4.41}$$

$$T_N(z) = \frac{1}{1+G_{CO}(z)C_e(z)P_e(z)}. \tag{4.42}$$

It is important to note that at lower frequencies, the difference in the numerator of $T_U(z)$ is non-zero due to slow tracking whereas at higher frequencies, the difference essentially becomes zero due to rapid tracking. Due to this fact, the estimated process $P_e(z)$ and the actual process $P(z)$ is the same at higher frequencies.

The derivative $dT(z)/dP(z)$ is examined[70],

$$\frac{dT_U}{dP} = \frac{C(z)}{1+G_{CO}(z)C_e(z)P_e(z)}, \tag{4.43}$$

$$\frac{dT_N}{dP} = \frac{G_{CO}(z)C(z)}{(1+G_{CO}(z)C_e(z)P_e(z))^2}. \tag{4.44}$$

Again, because the relative deviation from nominal is of interest, the logarithmic derivative is considered,

$$\frac{d\log T_U}{d\log P} = S_U(z) = \frac{C(z)P(z)}{1+G_{CO}(z)C_e(z)P_e(z)} \tag{4.45}$$

$$\frac{d\log T_N}{d\log P} = S_N(z) = \frac{G_{CO}(z)C_e(z)P_e(z)}{1+G_{CO}(z)C_e(z)P_e(z)} \tag{4.46}$$

As is noticed in (4.45) it depends on the compensator and hence, the lower bound of the compensator is essentially designed only on the basis of fast observer transient response (ten times the bandwidth of plant) for faster tracking. On the other hand in (4.46), increasing observer gain amplifies the observer noise and an upper bound is hence desired.

4.2 Validation on Double Mass-Spring-Damper System

Before proceeding toward the experimental data analysis, the attention is drawn back to the numerical example, the double mass-spring-damper system explained in the previous chapter. The calculation of K_e is validated on this example. Since the observer is attempting to estimate the values of state variables which are themselves changing, it is desired that the dynamics of the observer be significantly faster than the dynamics of the closed-loop system without the observer. A common guideline is to make the estimator poles (eigenvalues of $A_o - K_e C_o$) 4-10 times faster than the slowest plant pole (eigenvalues of A_o). The K_e is calculated on the basis of pole placement by using Ackermann formula presented in Appendix C. Making the estimator poles too fast can be problematic if the measurement is corrupted by noise or there are errors in the sensor measurement in general. The plant poles are at location $[-1.9298 \pm 6.1572i, -0.3142 \pm 1.9749i]$. Based on the poles found above, the observer poles are placed at $[-5.7893 \pm 18.4717i, -0.9427 \pm 5.9247i]$ such that the observer bandwidth is approximately ten times more than plant bandwidth. These poles can be modified later, if necessary. The observer gain matrix is obtained as

$$K_{ek} = \begin{bmatrix} 5628.1 \\ -416.75 \\ -59.71 \\ 15.51 \end{bmatrix}. \tag{4.47}$$

Observer and plant bandwidth in bode plot (Figure 4.3) is shown to demonstrate the noise attenuation while making the observer fast enough to achieve fast error tracking performance. The observer bandwidth is 26.4585 rad/sec which is almost ten times more than plant bandwidth, 3.0397 rad/sec. Error tracking shows that in the presence of noise, the estimates track the actual system fast enough while the

observer bandwidth attenuates the noise.

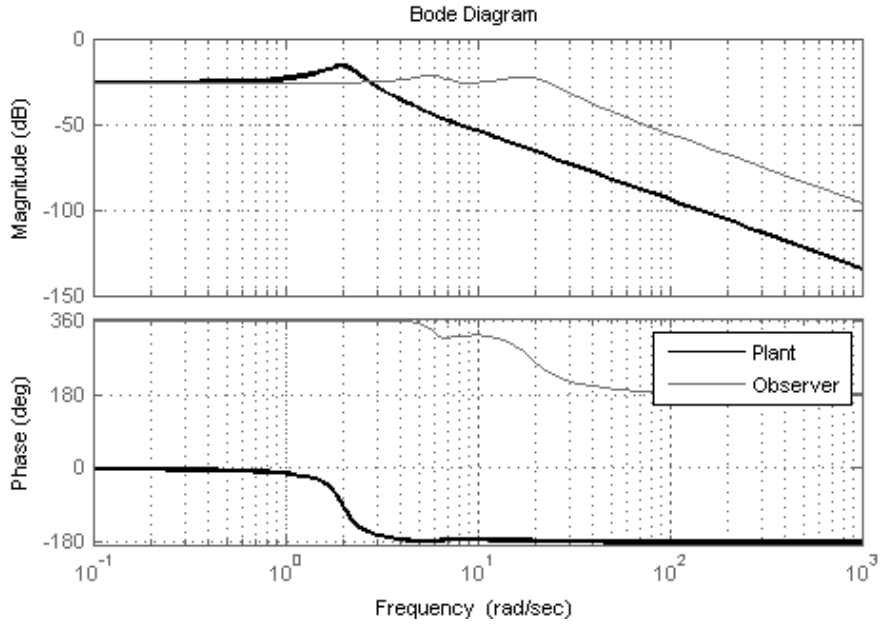

Figure 4.3: Bode Plot of the Observer and the Plant.

4.3 Validation on Sensor Fault Diagnostics Experimental Data

Coming back to the experimental data study, the data used was obtained from tests performed on a Ford gasoline engine. The test bench is comprised of a dynamometer capable of running steady state and transient speed load maps.

Previously, the data has been recorded for diagnostics of delay faults in engines which is also termed as six fault diagnostics.

4.3.1 UEGO Sensor And Control Strategy

In order to measure the air/fuel ratio (AFR) in the exhaust system and control it, UEGO sensors have been introduced in the automotive industry as an alternative to the heated exhaust oxygen sensors (HEGO) which may only indicate if the AFR is

above or below stoichiometry. In addition to giving the information on the operation of the system, these devices are used to measure the mass air flow in the exhaust line and determine the control action accordingly. An inventory of the common sensors used for the AFR control has been drawn in [77] along with their characteristics. UEGO sensors are fairly characterized by a nonlinear response but may usually be approximated by a first order transfer function with a delay. In [78] the sensor dynamics were described by a second order model changing from under-damped to over-damped with increasing load. An important factor which impacts the bandwidth of the AFR control system is the delay. This factor is behind the degenerate performance of the AFR control system especially at low speed where the bandwidth becomes very low for a tight transient control.

4.3.2 Diagnostics Of The UEGO Sensor

Apart from the control related issues, the diagnostics of the UEGO operation is nowadays a very challenging problem and is among the priorities of research as it is closely related to the overall performance of the AFR control system. In fact, Many types of faults may affect the operation of this UEGO sensor and potentially engender an increase in the emissions and a decrease in the fuel economy. These faults are typically characterized by a change in the sensor parameters or in its operational characteristics. Several methods and approaches have been proposed that diagnose the operation of the UEGO sensors namely the data-driven, model-based and statistics-based approaches. Reports issued by California air regulation board state that this sensor is subject to six possible faults that may affect its normal operation. These faults can be either an additional lag or delay in the transition of the response from lean to rich, rich to lean or both direction (see details in section 4.1.6). In [79] it has been reported that "asymmetrical faults may result in emission failures at much lower thresholds than symmetrical faults. Therefore, the logic to determine fault versus no fault conditions should determine the type of fault present."

4.3.3 Illustration Of The Six Possible Faults

CARB regulations incite automakers to monitor the operation of the UEGO sensor against any possible malfunction possibly affecting the transition from lean to rich, from rich to lean or both. These faults may cause instability of the closed-loop system and lead to degradation of the performance of the AFR regulation loop. It has been interpreted that the monitor should be able to detect six discreet faults [79] as shown in Figure 4.4.

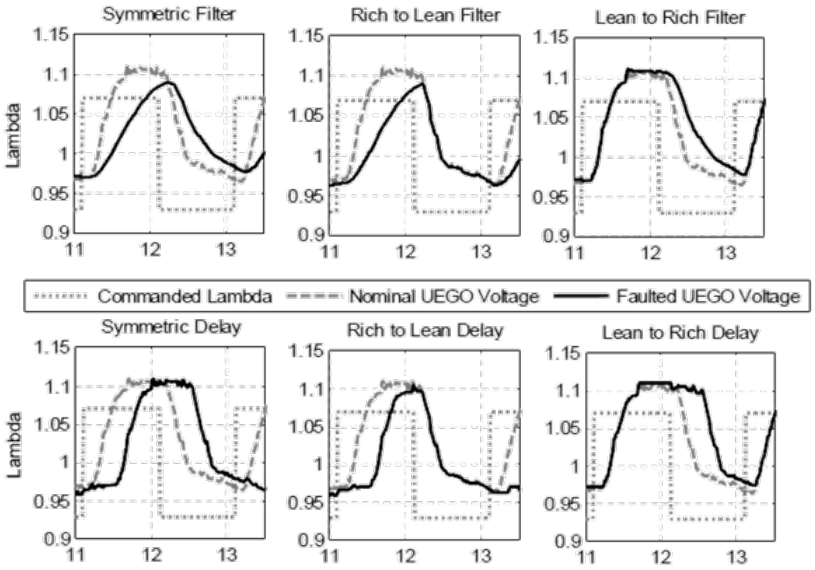

Figure 4.4: Faulted/Unfaulted λ Measurements Due to Faults in UEGO Sensor(Courtesy: Ford Motor Company).

The second and third plots show both lean to rich and rich to lean filter faults. These faults are due to asymmetry in the time constant T_c between lean and rich modes; time delays T_d in both burn modes are the same. The last two plots show both lean to rich and rich to lean delay faults. These faults are only due to asymmetry in the time delay; the time constants are the same in both burn modes. The first and fourth plots show both symmetric filter faults and symmetric delay faults. These faults have been induced using a dedicated software in the loop (SIL) that receives

the UEGO sensor measurement, induces the specific fault with a desired magnitude and direction, then feeds the signal back to the closed-loop system.

In this work, the experimental data is obtained from UEGO sensor and system identification is done in order to obtain the nominal plant. The main objective of this section is to demonstrate the workability of the proposed methodology and compare the results with parameter estimation done using the recursive least squares approach.

4.3.4 No Faults (Fault code - 0)

The nominal system for which the data is fitted with first order approximation is obtained as,

$$Y(s) = \frac{0.02943}{z - 0.9708}. \tag{4.48}$$

With a single input as air/fuel ratio to the plant and single output as air/fuel ratio measured by sensor, the estimated and actual output using the proposed methodology is shown in Figure 4.5. It can be easily seen from the figure that the estimated output traces the actual output quite well. The error histogram in Figure 4.6 shows estimated output having minimal percentage deviation from the actual output.

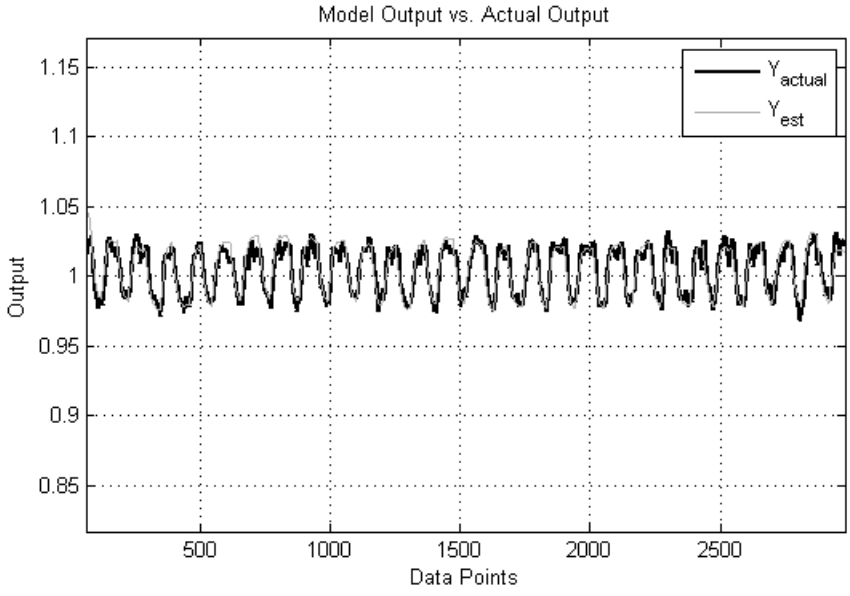

Figure 4.5: Comparison of the Actual and Estimated Air/Fuel Ratio Using Observers.

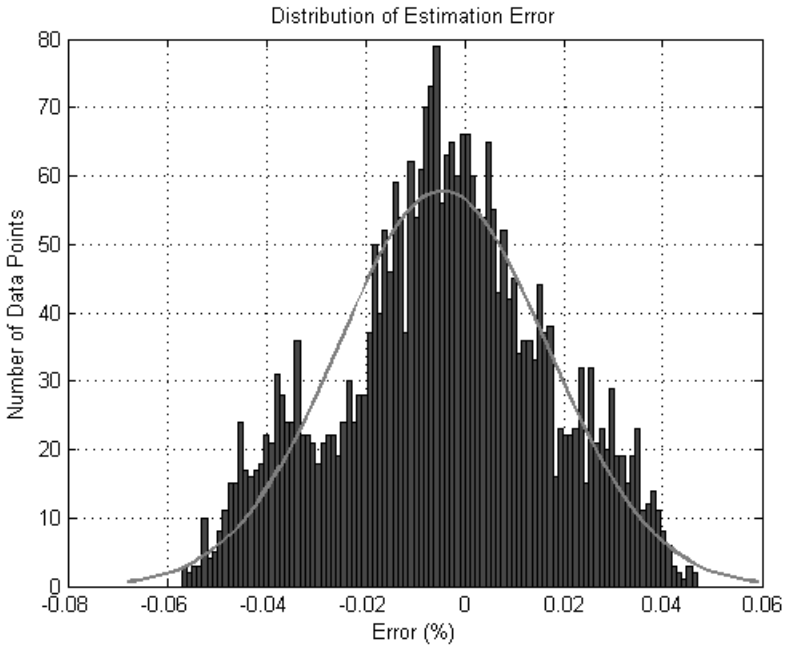

Figure 4.6: Error Histogram Using Observers.

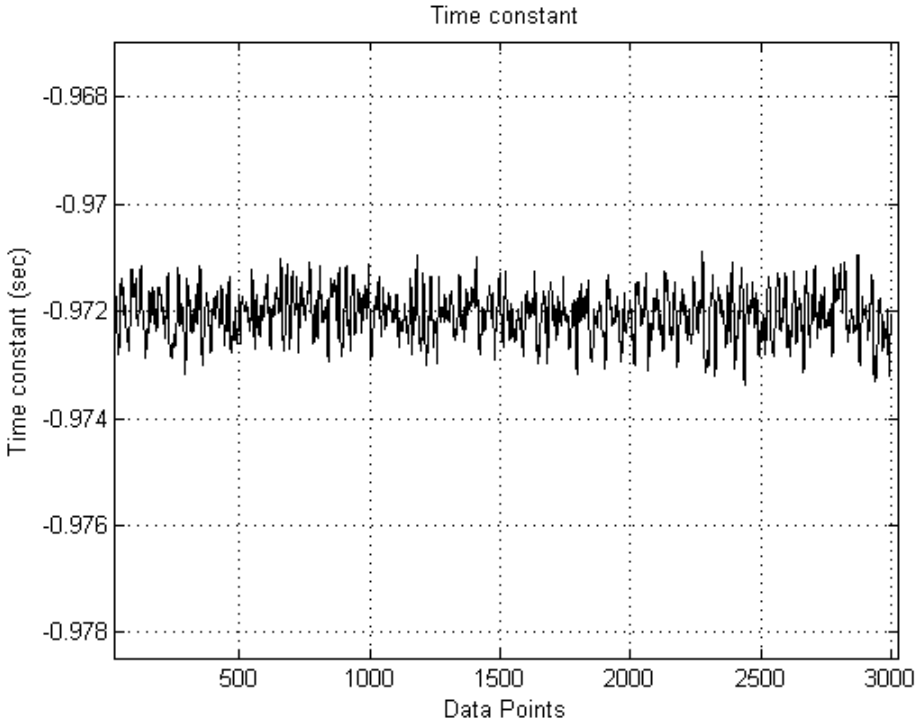

Figure 4.7: Time Constant (sec) for No Fault Case (Fault Code - 0) Using Observers.

Figure 4.7 shows the variations of the time constant. The trajectory is uniquely calculated within the system so that an exact value of estimated output can be obtained at each iteration. The estimation of the time constant is fairly accurate because the perturbations have been within very small bounds of ± 0.003. In order to compare the performance of methodology, the same data has been adapted using the recursive least square approach (Figure 4.8 and 4.9) and then compared. The comparison of the two methodologies on the basis of error histogram shows that the proposed methodology adapts more efficiently. Furthermore the variations (Figure 4.7) in the observer based method calculated time constant is less than that obtained from the RLS calculated time constant (Figure 4.10).

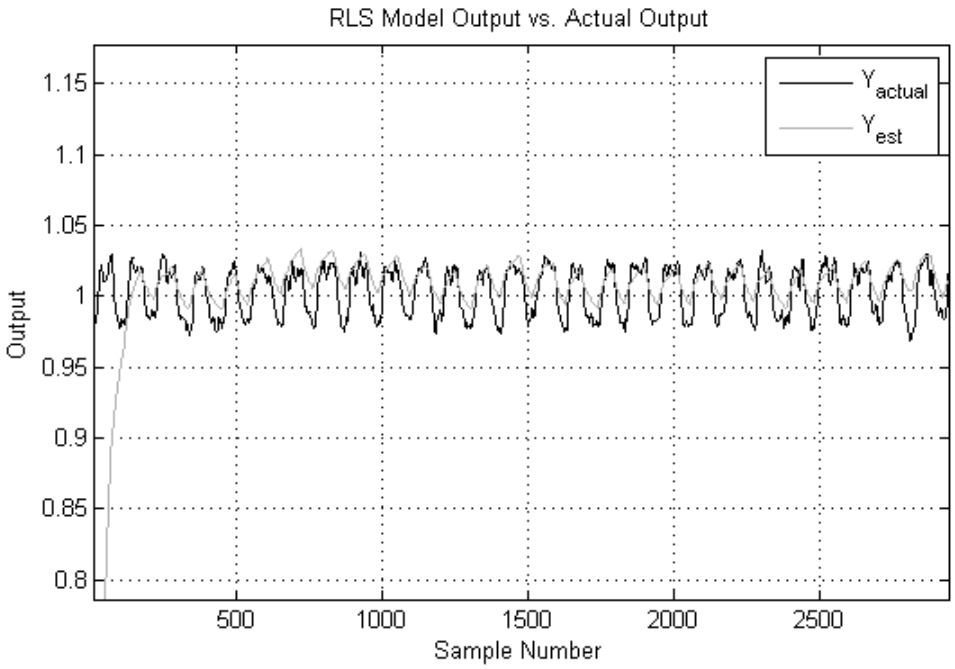

Figure 4.8: Comparison of the Actual and Estimated Air/Fuel Ratio Using Least Squares Technique.

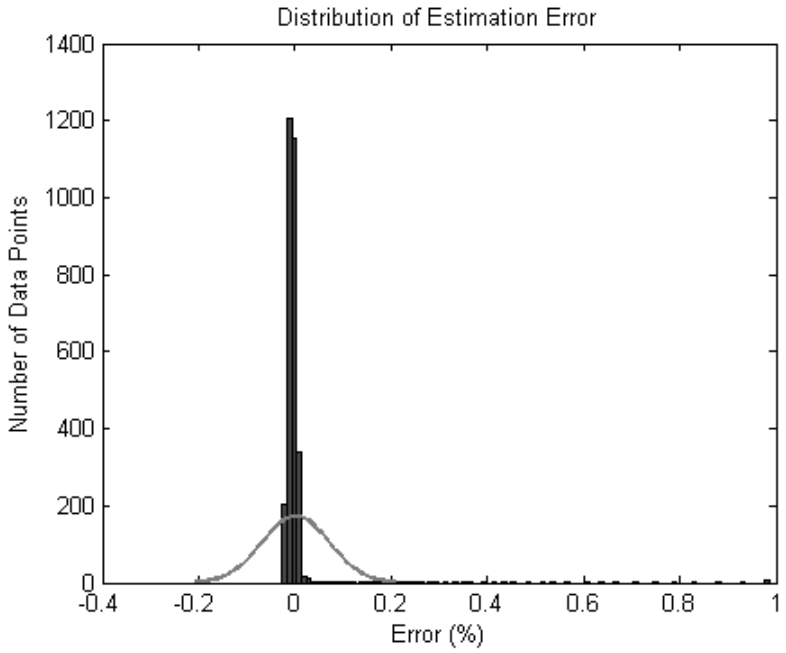

Figure 4.9: Error Histogram Using Least Squares Technique.

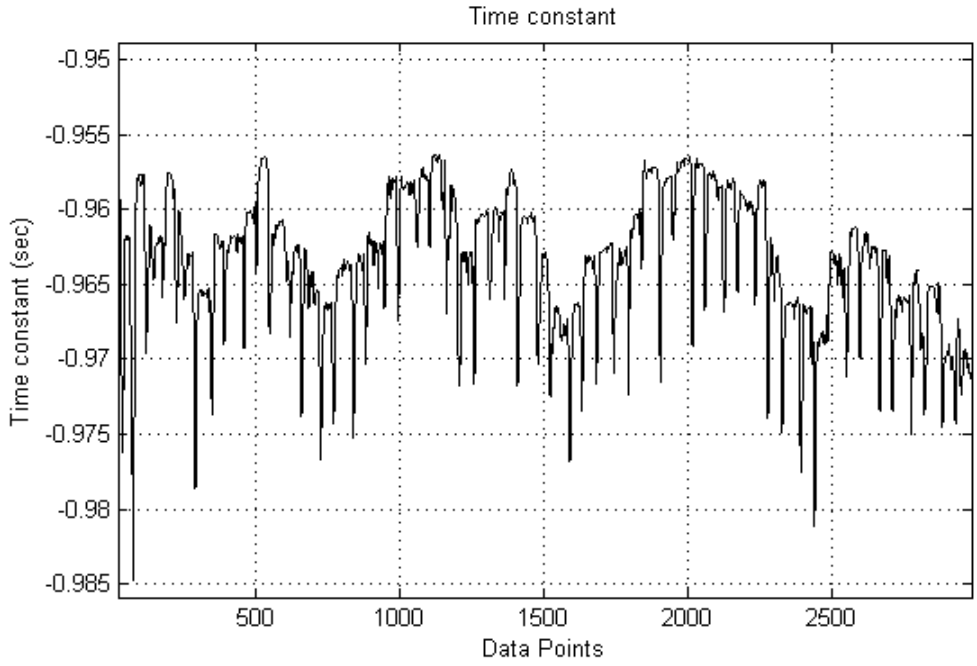

Figure 4.10: Time Constant (sec) for No Fault Case (Fault Code - 0) Using Least Squares Technique.

4.3.5 Symmetric Delayed Faults (Fault code - 1)

The data that consists of symmetric faults is analyzed in this section using the proposed methodology. The nominal system, when the data is fitted with first order approximation, is obtained as

$$Y(s) = \frac{0.006668}{z - 0.9932}. \tag{4.49}$$

Given the nominal system and the output signal, the estimated output using observers is obtained as shown in Figure 4.11. It can be noted here that the coefficients are well adapted with each data point such that the estimated output approximates the actual output. The deviation from actual trajectory can be noted as shown in Figure 4.12. The error is well within the range of $\pm$ 0.05 %. which positively concludes the discussed approach.

An important point to understand is the variation of the time constant. The methodology provides a series of values of the time constant that gives a correct output estimate at each iteration. They are shown in Figure 4.13.

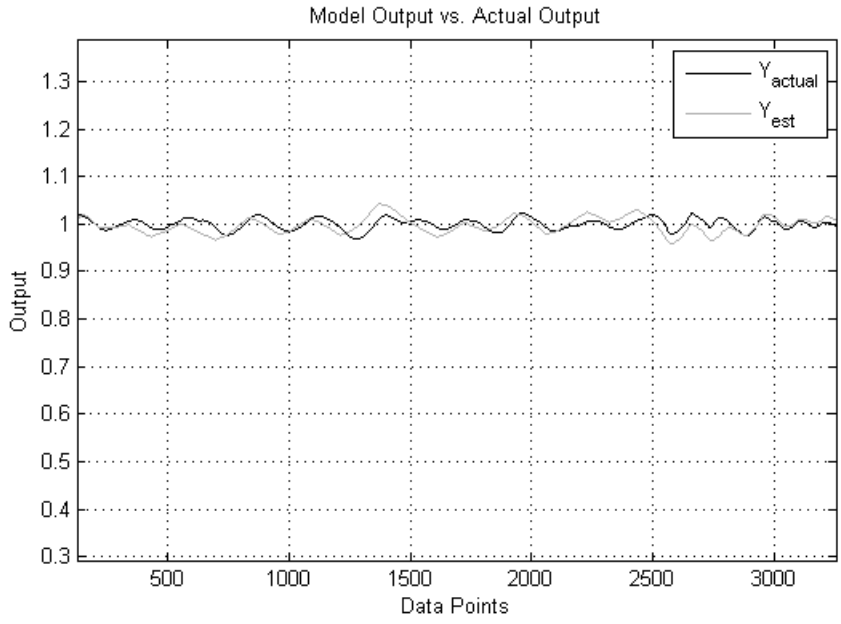

Figure 4.11: Comparison of the Actual and Estimated Air/Fuel Ratio Using Observers.

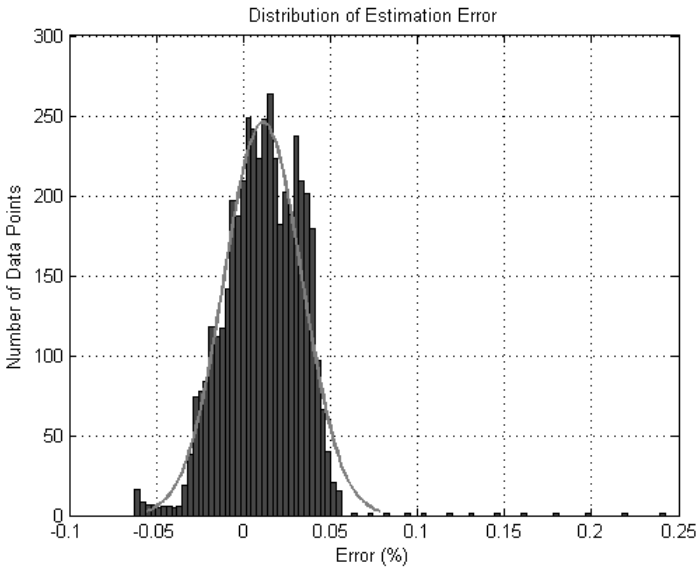

Figure 4.12: Error Histogram Using Observers.

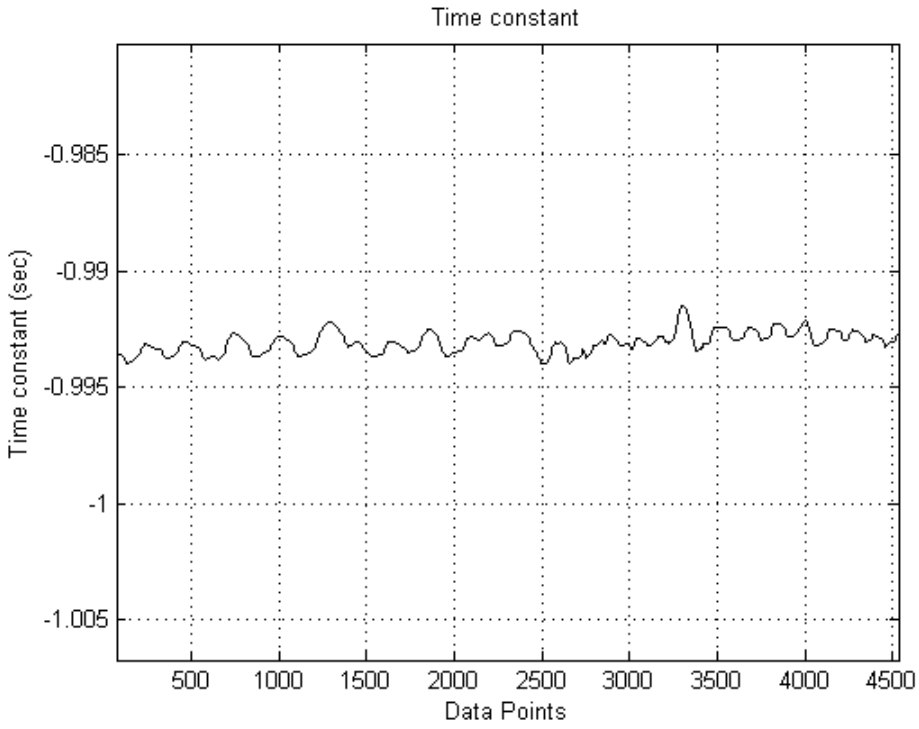

Figure 4.13: Time Constant (sec) for Symmetric Fault Case (Fault Code - 1) Using Observers.

With a step to further investigate the performance of the approach, a comparison

has been made between the error histogram obtained from the proposed approach and the other obtained from the RLS technique (Figure 4.12 and 4.15). The plot of estimated and actual output obtained from the RLS technique for the same data is shown in Figure 4.14. The comparison shows that the observer based method works well for symmetric fault diagnostics data that is not very well adapted by the RLS technique and shows a better approach for parameter estimation.

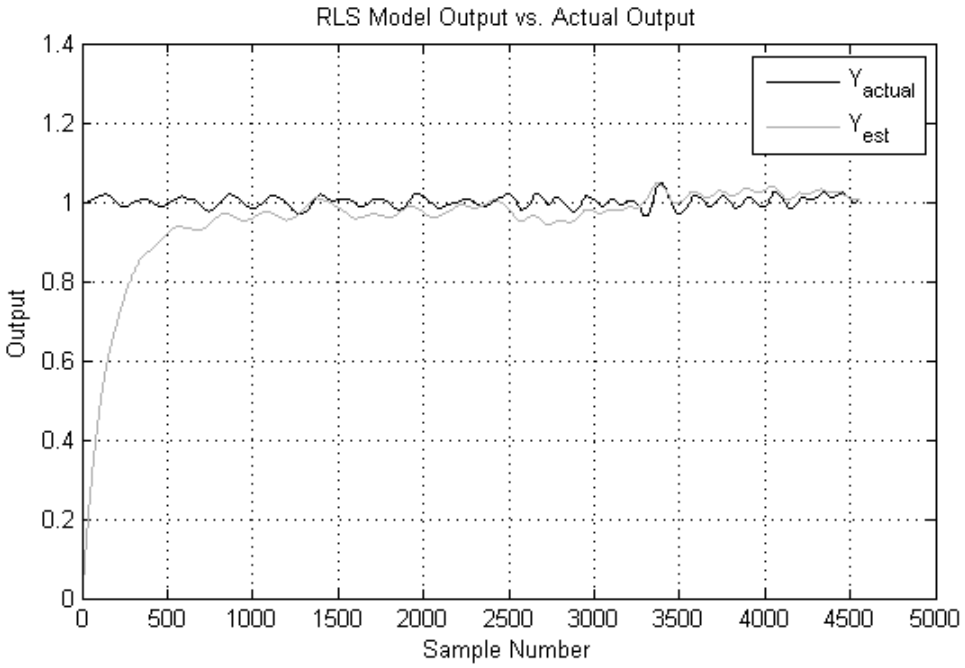

Figure 4.14: Comparison of the Actual and Estimated Air/Fuel Ratio Using Least Squares Technique.

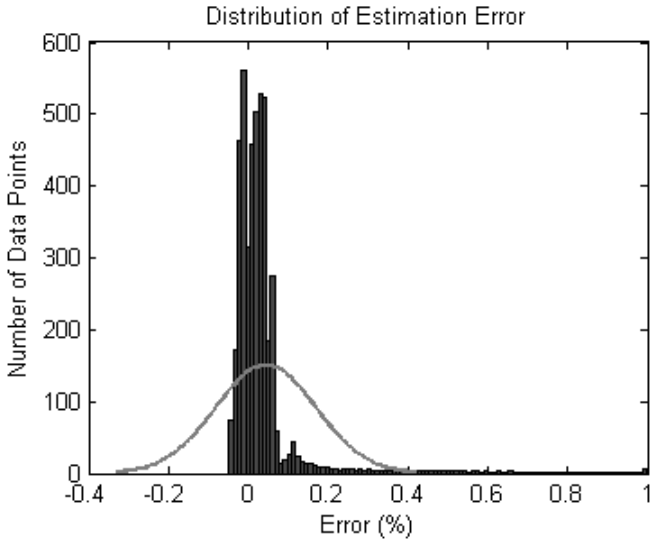

Figure 4.15: Error Histogram Using Least Squares Technique.

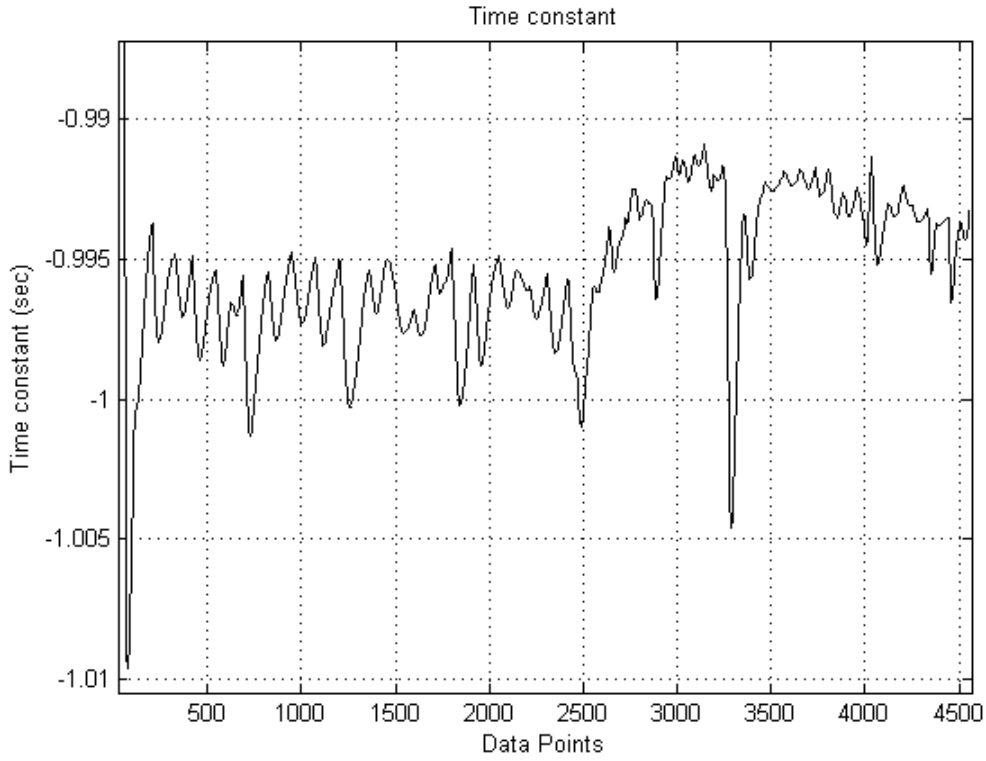

Figure 4.16: Time Constant (sec) for Symmetric Fault Case (Fault Code - 1) Using Least Squares Technique.

4.3.6 Asymmetric Rich to Lean Delayed Faults (Fault code - 6)

Similar analysis has been done for asymmetric faults (six faults) as done for symmetric faults. The nominal system, when the data is fitted with first order approximation, is obtained as

$$Y(s) = \frac{0.007779}{z - 0.992}. \tag{4.50}$$

Given the nominal system and the output signal, the estimated output using observers is obtained as shown in Figure 4.17. The coefficients are rapidly adapted with each data point such that the estimated output approximates the actual output. The deviation from actual trajectory can be noted as shown in Figure 4.18. The error is well within the range of ± 0.05 % in this case also.

The variation of the time constant is shown to be small enough to accurately estimate the parameter within the bounds of ±0.003. They are shown in Figure 4.19.

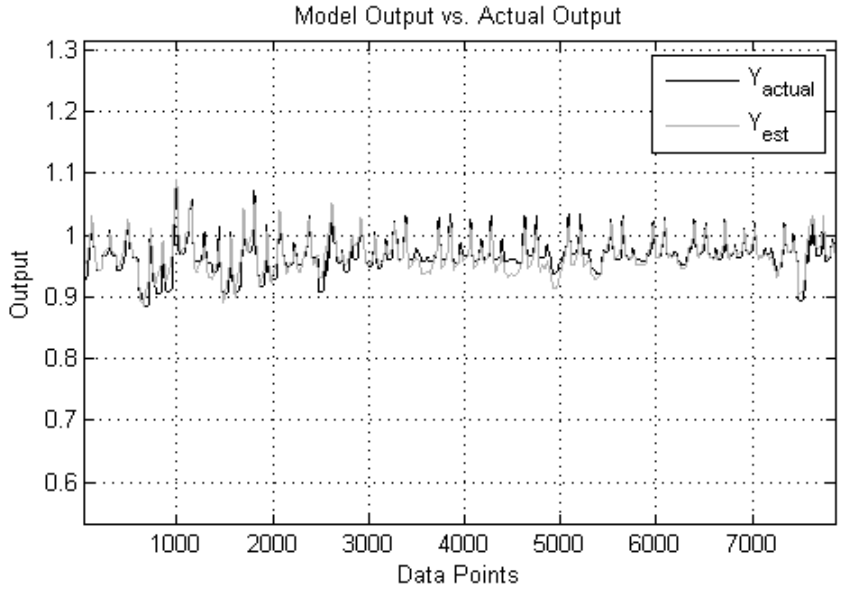

Figure 4.17: Comparison of the Actual and Estimated Air/Fuel Ratio Using Observers.

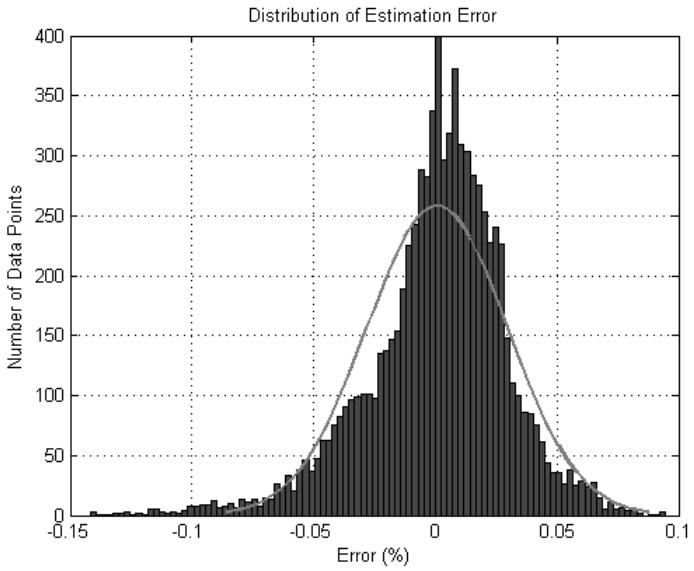

Figure 4.18: Error Histogram Using Observers.

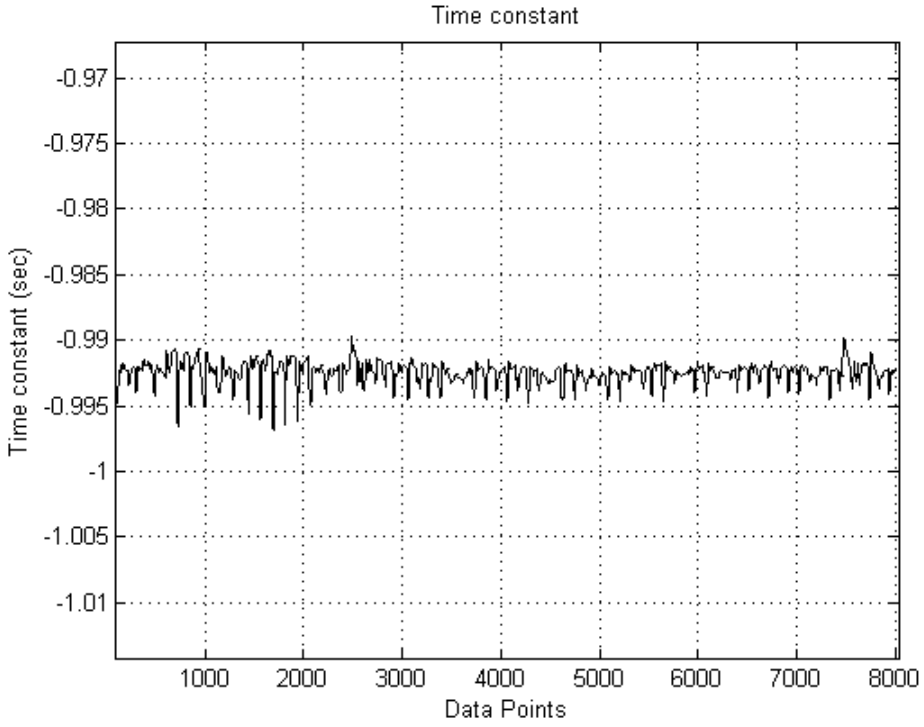

Figure 4.19: Time Constant (sec) for Asymmetric Fault Case (Fault Code - 6) Using Observers.

The comparison between the error histogram obtained from the proposed approach

and the other obtained from the RLS technique is shown (Figure 4.18 and 4.21). The plot of estimated and actual output obtained from the RLS technique for the same data is shown in Figure 4.20. The comparison shows that the observer based method works well for six fault diagnostics data that is not very well adapted by the RLS technique and shows a better approach for parameter estimation in this case also.

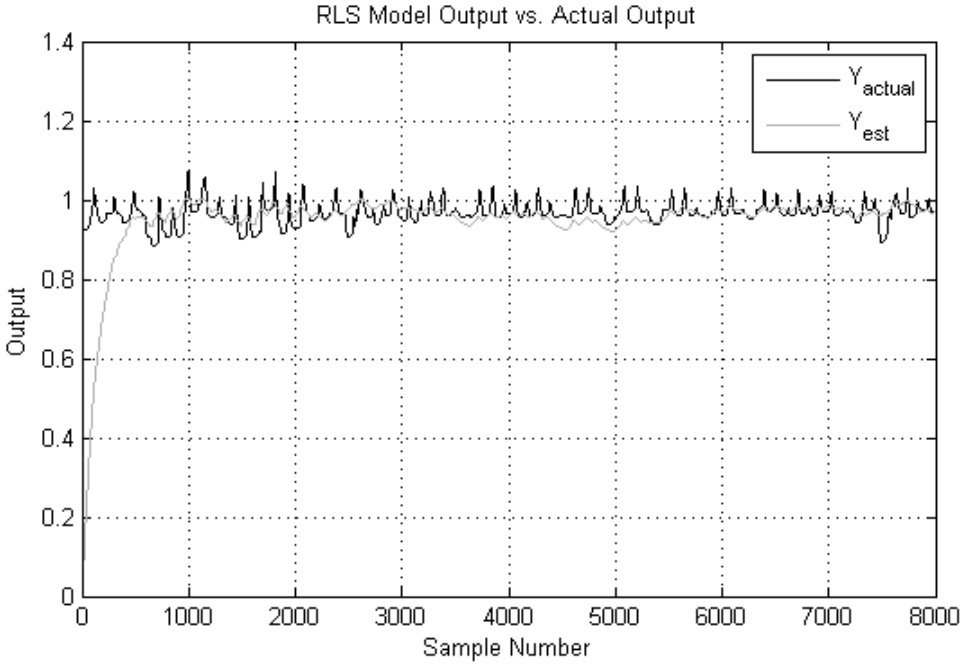

Figure 4.20: Comparison of the Actual and Estimated Air/Fuel Ratio Using Least Squares Technique.

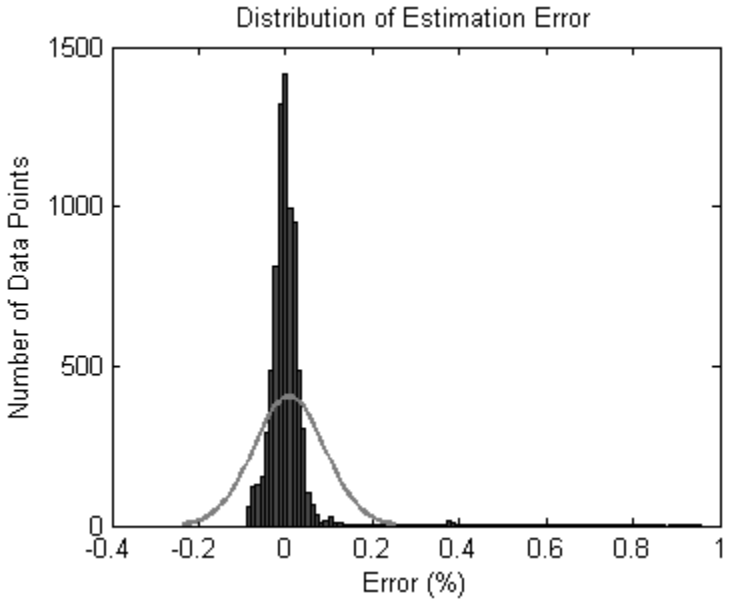

Figure 4.21: Error Histogram Using Least Squares Technique.

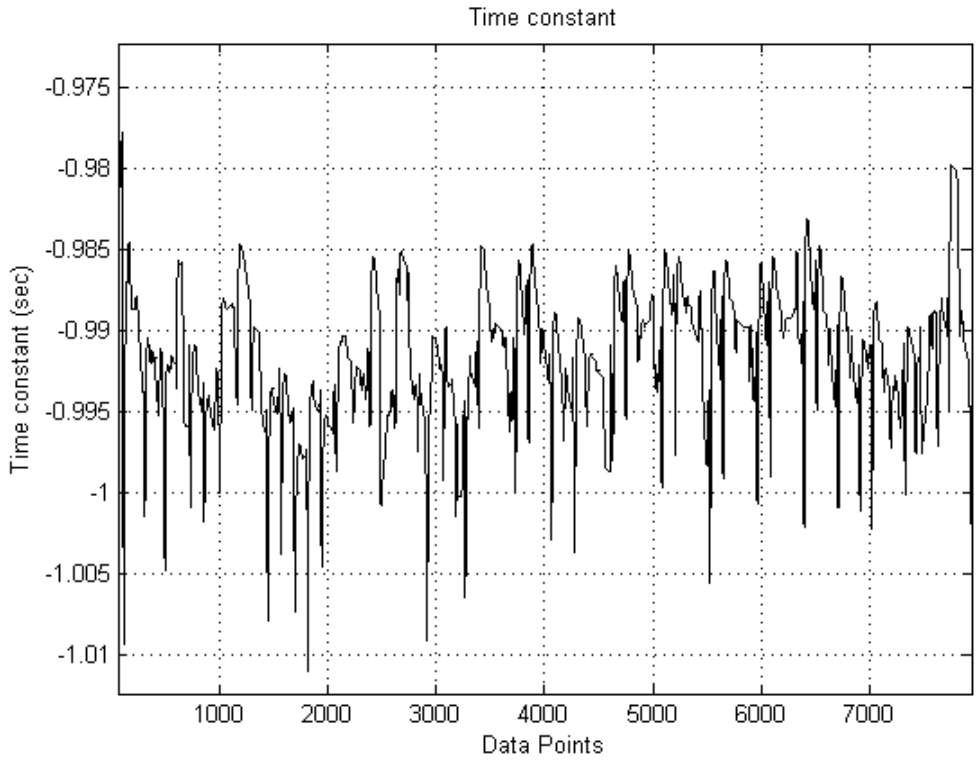

Figure 4.22: Time Constant (sec) for Asymmetric Fault Case (Fault Code - 6) Using Least Squares Technique.

4.3.7 Summary

With the above validation and diagnostics of different fault codes of UEGO sensor, it has been observed that observer based method results in a fast and accurate adaptation of air/fuel ratio (output of sensor) to track the perturbations as compared to least squares technique. Apart from that, the parameter estimation of time constant is faster and within smaller bounds than least squares technique. Table (4.1) and (4.2) compares the average and standard deviation of the parameter (Time constant) estimate. The average value proves the accurate estimation by both the methods and standard deviation observers the accuracy of quantification of parametric uncertainty. In the case of observer based method, the lower deviation concludes its better quantification ability that least squares method.

Table 4.1: Average And Standard Deviation Of Time Constant Using Observer.

Fault	Actual τ	$\bar{\tau}$	σ_τ
No Fault (Code 0)	0.9708	0.9711	0.0053
Symmetric Fault (Code 1)	0.9932	0.9919	0.00451
Asymmetric Fault (Code 6)	0.992	0.9923	0.00327

Table 4.2: Average And Standard Deviation Of Time Constant Using Least Squares.

Fault	Actual τ	$\bar{\tau}$	σ_τ
No Fault (Code 0)	0.9708	0.9620	0.0205
Symmetric Fault (Code 1)	0.9932	0.9946	0.0174
Asymmetric Fault (Code 6)	0.992	0.9915	0.0146

Overall, this gives a better and robust estimation method. This process can be applied in a continuous fashion where the algorithm will continuously estimate the parameters of the system model and will perform fault diagnosis efficiently.

Chapter 5
CONCLUSIONS AND FUTURE WORK

Presented in Chapter 1 was the introduction and overview of the motivation for on-board diagnostics using parameter based and observer based methods and the diagnostic requirements. In chapter 2 various diagnostic methods proposed in the literature and those being currently used were reviewed and the concept of observers was described. Chapter 3 described the fault detection, isolation and estimation method proposed in the present study based on analyzing the coefficient change of a diagnostic model. The model parameter estimation through recursive least squares was also presented. Chapter 4 proposed the observer based estimation method and also validated the method on sensor fault diagnostics data.

5.1 Fault Detection, Isolation and Estimation Using Observers

5.1.1 Summary

A proposed scheme involves the online parameter estimation for applications such as system fault detection and diagnosis. A combination of two Luenberger observers with different decay rates and a delay in between is applied in the system for real time estimation of the system coefficients in finite time. Discussion about approaches adopted for defining bounds on the observer gain matrix, and hence designing the observer, is done for achieving noise attenuation and one-step estimation. The observer based method has been validated on simulated data of a double-mass-spring-damper system and on experimental data for sensor fault diagnostics.

In this work the application of a parametric adaptive model based technique (Information synthesis) for fault detection, isolation and estimation(FDIE) is presented. This methodology can be used onboard to alert the vehicle operator in case of failure

of a system that results in emission exceeding the legal limits. The diagnostic model parameters are updated by using the observer based method. The model identified is further validated by correlating it with the physics based model of the system. To perform FDIE the changes in the parameters of the model of the system are analyzed. A coefficient error vector is produced by comparing the healthy model parameters with the adapted model parameters. The magnitude of the coefficient error vector may be compared to a threshold, to be used for fault detection. Fault isolation is shown to be possible using fault signature codes which are unique to faults that affect the system dynamics differently. These fault signatures are the angles made by the error vector in the fault space. The effect of variability of the system parameters on the fault signature codes and the magnitude of the fault vector are shown through the sensitivity analysis. The most important feature of the FDIE method presented in this work is the robustness to modeling errors and the handling of noise acting on the system. This robustness is required to prevent false detection of failure or to miss the failure detections. The proposed approach has been successfully implemented and validated to diagnose faults in engine subsystems using experimentally obtained data.

5.1.2 Contributions and Publications

This outcome of research is a novel way of performing the parameter estimation for various practical systems. Utilizing state estimation techniques and performing algebraic manipulations, coefficient estimation and real time diagnosis and monitoring of practical situations can be achieved. Consequently, a novel way of performing the diagnostics of systems is proposed. Synthesizing the information contained in the coefficient changes of the diagnostic model, accurate fault detection, isolation and estimation can be achieved.

A summary of major contributions:

1. Observer design: Following a deterministic approach of coupling the two infinite

time Luenberger observers, an observer design is proposed that enables finite time exact convergence.

2. Quantification of parametric uncertainty: With known estimated states, the parametric uncertainty is quantified while satisfying the conditions of state persistency of excitation and observer convergence.

3. Design of observer gain matrix: The window of selecting the observer gain matrix is shown to ensure noise attenuation while mandating faster tracking.

4. Real time validation: The task of online estimation of model parameters has been performed using engine sensor data, which conclusively validates the proposed methodology.

5. Fault detection, isolation and estimation: The magnitude of the coefficient error vector against a threshold is compared and the fault signature code for fault isolation is analyzed.

Publication

1. Kukreja. A., Franchek. M.A., Grigoriadis. K., Gupta A., Real Time Adaptive Model Based FDIE of Diesel Engine Air Handling System. DSCC 2012, FL.

5.1.3 Scope for Future Work

A promising way of performing observer based estimation and fault diagnostics is presented in this work. Some directions in which this work can be further extended are listed below:

1. Further validation of the proposed approach can be accomplished by developing a real time parameter estimation algorithm and testing on other real time experimental data for different applications.

2. A routine to calculate the bias and the delay are to be included in the algorithm.

3. After validation, the approach may be utilized for highly demanded areas in today's world such as fault detection and diagnosis, condition and health monitoring and so forth.

Bibliography

[1] Janos, J.G., Fault Detection and Diagnosis In Engineering Systems, Marcel Dekker Inc. 1998.

[2] Simani, S., Fantuzzi, C., Patton, R.J., Model based Fault Diagnosis In Dynamic Systems Using Identification Techniques, Springer-Verlag London Ltd, 2003.

[3] Isermann, R., *Model Based fault detection and Diagnosis- Status and Applications*, Annual Reviews in Control 29, pp. 71-85, 2005.

[4] Korenberg, M.J., Billings, S.A., Liu, Y.P., McILROY, P.J., *Orthogonal parameter estimation algorithm for non-linear stochastic systems*, International Journal of Control, Vol 48, pp. 193-210, 1988.

[5] Ljung, L., System Identification- Theory for the User, 2nd Edition. Prentice Hall, 1999.

[6] Soderstrom, T., Stoica, P., System Identification, Prentice Hall International,U.K, 1989.

[7] Heywood, J.B., Internal Combustion Engine Fundamentals, Mc Graw Hill, New York, 1988.

[8] Guzzela, L., Onder, C.H., Introduction to Modeling and Control of Internal Combustion Engine Systems, Springer, Heidalberg, Germany, 2004.

[9] Xun Wang; Kruger, U.; Irwin, G.W.; McCullough, G.; McDowell, N.; , *Nonlinear PCA With the Local Approach for Diesel Engine Fault Detection and Diagnosis*, Control Systems Technology, IEEE Transactions on , vol.16, no.1, pp.122-129, Jan. 2008.

[10] King, P., Vinsonneau, J., Shields, D., and Burnham, K., 2002, *Improved SI Engine Modeling Techniques With Application to Fault Detection*, in Proceedings of the 2002 IEEE International Conference on Control Applications, Glasgow, Scotland, U.K., pp. 719-24.

[11] Mohammadpour, J., K.M. Grigoriadis, M.A. Franchek, and B.J. Zwissler, *Real-time diagnosis of the exhaust recirculation in diesel engines using least-squares parameter estimation*, ASME Journal of Dynamic Systems and Controls, Vol. 132, Jan. 2010.

[12] Schilling, A., Amstutz, A., Guzzella, L., *Model-based detection and isolation of faults due to ageing in the air and fuel paths of common-rail direct injection diesel engines equipped with a lambda and a nitrogen oxides sensor*, Proceedings of the institution of mechanical engineers part d-journal of automobile engineering SN 0954-4070, Jan 2008 VL 222 IS D1, BP 101, EP 117.

[13] J. Mohammadpour, K.M. Grigoriadis, M.A. Franchek, *A survey on diagnostic methods for automotive engines*, International Journal of Engine Research February 2012 13: 41-64, first published on November 21, 2011.

[14] Michael Mangold, Gtz Lauschke, Johanna Schaffner, Michael Zeitz, Ernst-Dieter Gilles, *State and parameter estimation for adsorption columns by nonlinear distributed parameter state observers*, Journal of Process Control, Volume 4, Issue 3, August 1994, Pages 163-172, ISSN 0959-1524, 10.1016/0959-1524(94)85008-9.

[15] Bin Jiang, Fahmida N. Chowdhury, *Parameter fault detection and estimation of a class of nonlinear systems using observers*, Journal of the Franklin Institute, Volume 342, Issue 7, November 2005, Pages 725-736, ISSN 0016-0032, 10.1016/j.jfranklin.2005.04.007.

[16] Khalifa Al-Hosani, Vadim I. Utkin, *Parameters estimation using sliding mode observer with shift operator*, Journal of the Franklin Institute, Volume 349, Issue 4, May 2012, Pages 1509-1525, ISSN 0016-0032, 10.1016/j.jfranklin.2011.05.002.

[17] M. Perrier, S.Feyo de Azevedo, E.C. Ferreira, D. Dochain, *Tuning of observer-based estimators: theory and application to the on-line estimation of kinetic parameters*, Control Engineering Practice, Volume 8, Issue 4, April 2000, Pages 377-388, ISSN 0967-0661, 10.1016/S0967-0661(99)00164-1.

[18] D. Ghosh, *Nonlinear-observerbased synchronization scheme for multiparameter estimation*, 2008 EPL 84 40012, doi:10.1209/0295-5075/84/40012.

[19] A. Alawi, *Real-time fault diagnosis using observer and physical parameter estimation*, ProQuest Dissertations and Theses, University of Dar-Es-Salaam, 2007.

[20] I. Kolmanovsky, I. Sivergina, J. Sun, *Simultaneous input and parameter estimation with input observers and set membership parameter bounding: Theory and an automotive application*, Int. J. Adapt. Control Signal Process. 2006; 20:225246 DOI:10.1002/acs.899.

[21] Tyukin, Ivan Y.; Steur, Erik; Nijmeijer, Henk; van Leeuwen. Cees, *Adaptive Observers and Parameter Estimation for a Class of Systems Nonlinear in the Parameters*, 17-th IFAC World Congress, 6-11 Seoul, 2008.

[22] Zhongyu Zhao, Wen-Fang Xie, Henry Hong and Youmin Zhang, *A disturbance-decoupled adaptive observer and its application to faulty parameters estimation of a hydraulically driven elevator*, International Journal of Adaptive Control and Singal Processing, 2011; 25:519534 DOI: 10.1002/acs.1216.

[23] Xiuliang Li, Qinghua Zhang, and Hongye Su, *An adaptive observer for joint estimation of states and parameters in both state and output equations*, International Journal of Adaptive Control and Singal Processing, 2011; 25:831842 DOI: 10.1002/acs.1244.

[24] Wen Chen, Fahmida N. Chowdhury (2007): *Simultaneous identification of time-varying parameters and estimation of system states using iterative learning observers*, International Journal of Systems Science, 38:1, 39-45.

[25] Hasan, SMN and Husain, I, *A Luenberger-Sliding Mode Observer for Online Parameter Estimation and Adaptation in High-Performance Induction Motor Drives*, IEEE Transactions on industry applications, VOL. 45, NO. 2, MARCH/APRIL 2009.

[26] Moraal, P.E. and Grizzle, J.W., *Observer Design for Non-Linear-Systems With Discrete-Time Measurements*, IEEE Transactions on Automatic Control, VOL. 40, NO. 3, MARCH 1995.

[27] Karafyllis, L and Kravaris, C, *On the observer problem for discrete-time control systems*, IEEE Transactions on Automatic Control, VOL. 52, NO. 1, JANUARY 2007.

[28] M. Farza, H. Hammouri, S. Othman, K. Busawon, *Nonlinear observers for parameter estimation in bioprocesses*, Chemical Engineering Science, Volume 52, Issue 23, December 1997, Pages 4251-4267, ISSN 0009-2509, 10.1016/S0009-2509(97)00190-5.

[29] N. Sheibat-Othman, D. Peycelon, S. Othman, J.M. Suau, G. Fvotte, *Nonlinear observers for parameter estimation in a solution polymerization process using infrared spectroscopy*, Chemical Engineering Journal, Volume 140, Issues 13, 1 July 2008, Pages 529-538, ISSN 1385-8947, 10.1016/j.cej.2007.11.039.

[30] Takaji Umeno, Katsuhiro Asano, Hideki Ohashi, Masahiro Yonetani, Toshiharu Naitou, Takeyasu Taguchi, *Observer based estimation of parameter variations and its application to tyre pressure diagnosis*, Control Engineering Practice, Volume 9, Issue 6, June 2001, Pages 639-645, ISSN 0967-0661, 10.1016/S0967-0661(01)00037-5.

[31] Roset, B and Nijmeijer, H, *Observer-based model predictive control*, International Journal of Control, ISSN 0020-7179, 11/2004, Volume 77, Issue 17, pp. 1452 - 1462.

[32] Price, M. G. *System parameter estimation using an extended luenberger observer*. University of Virginia. ProQuest Dissertations and Theses, 1978, pp. 131.

[33] Kolka, G. K. G. *Parameter estimation using linear observers*. University of Warwick (United Kingdom)). PQDT - UK, Ireland,1981.

[34] Maybhate, A. and Amritkar, R.E., *Use of synchronization and adaptive control in parameter estimation from a time series*, Physical Review E, ISSN 1063-651X, 01/1999, Volume 59, Issue 1, pp. 284 - 293.

[35] Hsu, LY and Chen, TL, *Vehicle Full-State Estimation and Prediction System Using State Observers*, IEEE Transactions on Vehicular Technology, ISSN 0018-9545, 07/2009, Volume 58, Issue 6, pp. 2651 - 2662.

[36] Ibarra-Junquera, V., Rosu, H.C., Cornejo-Prez, O., *Supersymmetric Free-Damped Oscillators: Adaptive Observer Estimation of the Riccati Parameter*, International Journal of Theoretical Physics, ISSN 0020-7748, 11/2006, Volume 45, Issue 11, pp. 2084 - 2092.

[37] D. Luenberger, *Observer for multivariable systems*, IEEE Transactions on Automatic Control - IEEE TRANS AUTOMAT CONTR , vol. 11, no. 2, pp. 190-197, 1966.

[38] R.E. Kalman, *A new approach to linear filtering and prediction problems*, Transactions of the ASME - Journal of basic engineering, 82 (Series D): 35-45.

[39] M. Kinnaert, *Fault diagnosis based on analytical models for linear and nonlinear systemsA tutorial*, Proceedings of IFAC Safeprocess, Washington, USA, 2003, 3750.

[40] D.M. Frank, S.X. Ding, B. Koppen-Seliger, *Current developments in the theory of FDI*, Proceedings of IFAC Safeprocess, Budapest, Hungary, 2000, 1627.

[41] J.J. Gertler, *All linear methods are equal and extendable to nonlinearities*, Proceedings of IFAC Safeprocess, Budapest, Hungary, 5263.

[42] A. J. Krener and W. Respondek, *Nonlinear observer with linearizable error dynamics*, SIAMJ. Contr., vol. 47, pp. 1081-1100, 1988.

[43] J. Tsinias, *Further results on the observer design problem*, Syst. Contr. Lett., vol. 14, pp. 411-418, 1990.

[44] J. P. Gauthier, H. Hammouri, and S. Othman, *A simple observer for nonlinear systems; applications to bioreactors*, IEEE Trans. Automat. Contr., vol. 37, pp. 875-880, June 1992.

[45] F. Deza, E. Busvelle, J. P. Gauthier, and D. Rakotopara, *High gain estimation for nonlinear systems*, Syst. Contr. Lett., vol. 18, pp. 295-299, 1992.

[46] A. TomamM, *An asymptotic observer for solving the inverse kinematics problem*, in Proc. Amer. Contr. Con, San Diego, May 1990, pp. 1774 1779.

[47] S. Nicosia, A. Tomambe, and P. Valigi, *Use of observers for nonlinear map inversion* Syst. Contr. Lett., vol. 16, pp. 447-455, 1991.

[48] Zhang YM, Jiang J. *An active fault-tolerant control system against partial actuator failures.* IEEE ProceedingsControl Theory and Applications 2002; 149(1):95104.

[49] Bastin G, Gevers MR. *Stable adaptive observers for nonlinear time-varying systems.* IEEE Transactions on Automatic Control 1998; 33(7):650658.

[50] Zhang Q. *Revisiting different adaptive observers through a unified formulation.* The 44th IEEE Conference on Decision and Control and European Control Conference, Seville, Spain, December 2005.

[51] Marino R, Santosuosso GL, Tomei P. *Robust adaptive observers for nonlinear systems with bounded disturbances.* IEEE Transactions on Automatic Control 2000; 46(6):967972.

[52] Wang D, Lum KY. *Adaptive unknown input observer approach for aircraft actuator fault detection and isolation.* International Journal of Adaptive Control and Signal Processing 2007; 21:3148.

[53] Zhang Q. *Adaptive observer for multiple-input-multiple-output (MIMO) linear time varying systems.* IEEE Transactions on Automatic Control 2002; 47(3):525529.

[54] Y. Zhu and X. R. Li, *Recursive least squares with linear constraints* Communication in Information and Systems, vol. 7, no.3, pp. 287-312, 2007.

[55] S. M. Stigler, The History of Statistics: The Measurement of Uncertainty Before 1900, Cambridge, Belknap Press of Harvard University Press, MA, 1986.

[56] A. Bjorck, Numerical Methods for Least Squares Problems, SIAM, Philadelphia, PA, 1996.

[57] D. A. Freedman, Statistical Models: Theory and Practice, Cambridge University Press, 2005.

[58] D. G. Kleinbaum, L. L. Kupper, A. Nizam, and K. E. Muller, Applied Regression Analysis and Multivariable Methods, 4th Edition, Duxbury Press, Belmont, CA, 2007.

[59] H. W. Sorenson, Parameter Estimation, Marcel Dekker, NY, 1980.

[60] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, Estimation with Applications to Tracking and Navigation, John Wiley and Sons, NY, 2001.

[61] Wiki02: http://en.wikipedia.org/wiki/Generalizedleastsquares.

[62] J. A. Apolinario, QRD-RLS Adaptive Filtering, 1st Edition, Springer, NY, 2009.

[63] M. Zhou, V. A. Centeno, J. S. Thorp and A. G. Phadke, *An alternative for including phasor measurements in state estimators*, IEEE Trans. Power Syst., vol. 21, no. 4, pp. 1930-1937, Nov. 2006.

[64] J. Mikles and M. Fikar, Process modeling, identification, and control II: identification and optimal control, Springer-Verlag Berlin Heidelberg, NY, 2007.

[65] S. Haykin, Adaptive Filter Theory, 4th Edition, Prentice-Hall, Englewood Cliffs, NJ, 2001.

[66] I. K. Proudler, J. G. McWhirter, and T. J. Shepard, "Fast QRD-based Algorithms for least-squares linear prediction," in IMA Conf. Proc., Warwick, England, 1988.

[67] J. M. Cioffi, *The fast Householder RLS Adaptive Filter*, in Proc. ICASSP, Albuquerque, NM. Apr. 1990.

[68] J. M. Cioffi, *The fast adaptive ROTORs RLS algorithm*, IEEE Trans. Acoustic, Speech Signal Processing, vol. ASSP-38, pp. 631-653, 1990.

[69] Ricardo G. Sanfelice, and Laurent Praly. 2011. *On the performance of high-gain observers with gain adaptation under measurement noise.* Automatica 47, 10 (October 2011), 2165-2176. DOI=10.1016/j.automatica.2011.08.002.

[70] Ellis, George (2002). Observers in Control Systems - A Practical Guide. Elsevier.

[71] Dai, Xuewu (2008). Observer-based parameter estimation and fault detection. Thesis for doctor of philosophy. School of Electrical and Electronic Engineering, University of Manchester.

[72] S. Daley, H. Wang, "Application of a high gain observer to fault detection," Second IEEE Conference on Control Applications, 1993 Vancouver, B.C..

[73] D. Yoo, S. S.-T. Yau, Z. Gao, *Optimal fast tracking observer bandwidth of the linear extended state observer*, International Journal of Control, Vol. 80, No. 1, 2007, 102-111.

[74] P. H. Menold, R. Findeisen, and F. Allgower, *Finite time convergent observers for linear time-varying systems*, Proceedings of the 42nd IEEE Conference on Decision and Control, Hawaii, 2003, pp. 56735678.

[75] Feng Jun-E, *Finite time functional observers for discrete-time singular systems with unknown inputs*, Control Conference (CCC), 2010 29th Chinese, vol., no., pp.65-70, 29-31 July 2010.

[76] Kukreja. A., Franchek. M.A., Grigoriadis. K., Gupta A., "Real Time Adaptive Model Based FDIE of Diesel Engine Air Handling System". Dynamic Systems and Control Conference 2012, Florida.

[77] A. Chevalier, C.W. Vigild, and E. Hendricks. *Predicting the Port Air Mass Flow of SI Engines in Air/Fuel Ratio Control Applications.* SAE Special Publications, 2000.

[78] R.C. Turin and H.P. Geering. *Online Identification of Air-to-Fuel Ratio Dynamics in a Sequentially Injected SI Engine.* SAE Special Publications, 955:41-51, 1993.

[79] G. Soriano, T. Clark, and J. Kerns. *OBD UEGO Upstream Response Rate (6 pattern) Monitor.* Ford Technical Report, May 13, 2009.

Appendix A

FINITE TIME OBSERVER CONVERGENCE

Theorem 1 Assume $\hat{\bar{A}}$ and d can be chosen such that

(a1) $\hat{\bar{A}}$ is stable

(a2) $det[\bar{N}\hat{\bar{A}}^d\bar{N}] \neq 0$.

Then, (4.19) with $\bar{K} = [\ \bar{I}_r \quad 0\ ][\ \bar{N} \quad e^{\hat{\bar{A}}^d}\bar{N}]^{-1}$, is an observer for system (4.7), whose output $\check{\bar{z}}(i)$ converges to $\bar{z}(i)$ in finite time d.

Proof Since $\hat{\bar{z}}_k(i+1) - \bar{z}(i+1) = \hat{\bar{A}}_k(\hat{\bar{z}}_k(i) - \bar{z}(i))$ for $k = 1, 2$, we have

$$\hat{\bar{z}}(i+1) - \bar{N}\bar{z}(i+1) = \hat{\bar{A}}(\hat{\bar{z}}(i) - \bar{N}\bar{z}(i)), i \geq 0, \tag{A.1}$$

Therefore

$$\hat{\bar{z}}(i) - \bar{N}\bar{z}(i) = \hat{\bar{A}}^d[\hat{\bar{z}}(i-d) - \bar{N}\bar{z}(i-d)], k \geq d. \tag{A.2}$$

Using the fact $\bar{K}\bar{N} = \bar{I}$ and $\bar{K}\hat{\bar{A}}^d\bar{N} = 0$, we get

$$\begin{aligned} \check{\bar{z}}(i) &= \bar{K}[\hat{\bar{z}}(i) - \hat{\bar{A}}^d\hat{\bar{z}}(i-d)] \\ &= \bar{z}(i) + \bar{K}[\hat{\bar{z}}(i) - \bar{N}\bar{z}(i)] - \bar{K}\hat{\bar{A}}^d[\hat{\bar{z}}(i-d) - \bar{N}\bar{z}(i-d)] \end{aligned} \tag{A.3}$$

This implies $\check{\bar{z}}(i) = \bar{z}(i)$ for $i \geq d$.

Theorem 2 If $\bar{Z}_1, \bar{Z}_2$ are chosen such that $\hat{\bar{A}}_1$ is nonsingular and

$$Re\lambda_j(\hat{\bar{A}}_2) < \sigma < Re\lambda_j(\hat{\bar{A}}_1), j = 1, 2, ..., r. \tag{A.4}$$

for some $\sigma \in (0, 1)$, then, there exists a positive integer d_o such that for $d > d_o$, $det[\bar{N}\hat{\bar{A}}^d\bar{N}] \neq 0$.

Proof Since we have

$$\begin{aligned} \left[\ \bar{N} \quad \hat{\bar{A}}^d\bar{N}\ \right] &= \begin{bmatrix} \bar{I} & \hat{\bar{A}}_1^d \\ \bar{I} & \hat{\bar{A}}_2^d \end{bmatrix} \\ &= \begin{bmatrix} \bar{I} & 0 \\ \bar{I} & -\bar{I} \end{bmatrix} \begin{bmatrix} \bar{I} & \hat{\bar{A}}_1^d \\ 0 & \hat{\bar{A}}_1^d - \hat{\bar{A}}_2^d \end{bmatrix} \end{aligned} \tag{A.5}$$

and using the nonsingularity of matrix $\hat{\bar{A}}_1$ obtain

$$\begin{aligned} det\left[\begin{array}{cc} \bar{N} & \hat{\bar{A}}^d\bar{N} \end{array}\right] &= (-1)^n det(\hat{\bar{A}}_1^d - \hat{\bar{A}}_2^d) \\ &= (-1)^n det(\hat{\bar{A}}_1^d) det[I - ((\hat{\bar{A}}_1^{-1})^d \hat{\bar{A}}_2^d)] \\ &= (-1)^n det(\hat{\bar{A}}_1^d) det[I - ((\hat{\bar{A}}_1^{-1})^d (\sigma\bar{I})^d (\sigma^{-1}\bar{I})^d \hat{\bar{A}}_2^d)] \\ &= (-1)^n det(\hat{\bar{A}}_1^d) det[\bar{I} - (\sigma\hat{\bar{A}}_1^{-1})^d (\sigma^{-1}\hat{\bar{A}}_2)^d]. \end{aligned} \tag{A.6}$$

The first determinant on the right-hand side is nonzero for any positive integer. The second one tends to $+1$ as $d \to +\infty$, due to the assumption of the lemma, $(\sigma\hat{\bar{A}}_1^{-1})^d(\sigma^{-1}\hat{\bar{A}}_2)^d \to 0$ as $d \to +\infty$. Consequently there exists a positive integer d_o such that $det\left[\begin{array}{cc} \bar{N} & \hat{\bar{A}}^d\bar{N} \end{array}\right] \neq 0$ for all integer $d > d_o$. The proof ends.

Appendix B
ACKERMANN FORMULA

The observer gain matrix is defined using ackermann formula as,

$$\bar{K}_e = \phi(\bar{A}_0)\bar{S}_0^{-1}\begin{bmatrix} 0 & 0 & \dots & 1 \end{bmatrix}^T, \tag{B.1}$$

where

$$\bar{S}_0 = [\ \bar{C}_0 \quad \bar{C}_0\bar{A}_0 \quad \dots \quad \bar{C}_0\bar{A}_0^{n-1}\ ]^T, \tag{B.2}$$

and

$$\phi(\bar{A}_0) = (\bar{A}_0 - a\bar{I})^n, \tag{B.3}$$

where n is the dimension of $\bar{A}_0$ matrix and a is the desired repeated observer pole.