Fault Detection And Identification With Application To Advanced Vehicle Control Systems

Douglas, Randal K.
Speyer, Jason L.
Mingori, D. Lewis
et al.

Publication Date
1995

CALIFORNIA PATH PROGRAM
INSTITUTE OF TRANSPORTATION STUDIES
UNIVERSITY OF CALIFORNIA, BERKELEY

Fault Detection and Identification with Application to Advanced Vehicle Control Systems

Randal K. Douglas, Jason L. Speyer, D. Lewis Mingori, Robert H. Chen, Durga P. Malladi, Walter H. Chung

California PATH Research Report

UCB-ITS-PRR-95-26

This work was performed as part of the California PATH Program of the University of California, in cooperation with the State of California Business, Transportation, and Housing Agency, Department of Transportation; and the United States Department of Transportation, Federal Highway Administration.

The contents of this report reflect the views of the authors who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the State of California. This report does not constitute a standard, specification, or regulation.

August 1995

ISSN 1055-1425

Fault Detection and Identification With Application to Advanced Vehicle Control Systems

Award No. 65H998, M.O.U. 126

Randal K. Douglas, Jason L. Speyer, D. Lewis Mingori, Robert H. Chen,
Durga P. Malladi and Walter H. Chung

Mechanical, Aerospace and Nuclear Engineering Department
University of California, Los Angeles
Los Angeles, California 90024

Fault Detection and Identification With Application to Advanced Vehicle Control Systems

Award No. 65H998, M.O.U. 126

Randal K. Douglas, Jason L. Speyer, D. Lewis Mingori, Robert H. Chen,
Durga P. Malladi and Walter H. Chung

Mechanical, Aerospace and Nuclear Engineering Department
University of California, Los Angeles
Los Angeles, California 90024

August 3, 1995

Fault Detection and Identification With Application to Advanced Vehicle Control Systems

Award No. 65H998, M.O.U. 126

Randal K. Douglas, Jason L. Speyer, D. Lewis Mingori, Robert H. Chen,
Durga P. Malladi and Walter H. Chung

Mechanical, Aerospace and Nuclear Engineering Department
University of California, Los Angeles
Los Angeles, California 90024

August 3, 1995

Abstract

A preliminary design of a health monitoring system for automated vehicles is developed and results of tests in a high-fidelity nonlinear simulation are very encouraging. The approach is to fuse data from dissimilar instruments using modeled dynamic relationships and fault detection and identification filters. The filters are constructed so that the residual process has directional characteristics associated with the presence of a fault, that is, static patterns. Sensor noise, process disturbances, system parameter variations, unmodeled dynamics and nonlinearities all contribute to the blurring of these static patterns. A neural network residual processor is trained to form a threshold detection mechanism that announces a fault when one is present by recognizing fault patterns embedded in the residual. A health monitoring system based on this concept has been constructed for the longitudinal mode and monitors seven sensors and two actuators. Work also continues in refining a detailed nonlinear vehicle simulation which is used as a testbed for evaluating the performance of the health monitoring system.

Keywords. Automated Highway Systems, Automatic Vehicle Monitoring, Fault Detection and Fault Tolerant Control, Neural Networks, Reliability, Sensors, Vehicle Monitoring.

Fault Detection and Identification With Application to Advanced Vehicle Control Systems

Award No. 65H998, M.O.U. 126

Executive Summary

A preliminary design of a health monitoring system for automated vehicles is developed and results of tests in a high-fidelity nonlinear simulation are very encouraging. The approach is to fuse data from dissimilar instruments using modeled dynamic relationships and fault detection and identification filters. The filters are constructed so that the residual process has directional characteristics associated with the presence of a fault, that is, static patterns. Sensor noise, process disturbances, system parameter variations, unmodeled dynamics and nonlinearities all contribute to the blurring of these static patterns. A neural network residual processor is trained to form a threshold detection mechanism that announces a fault when one is present by recognizing fault patterns embedded in the residual. A health monitoring system based on this concept has been constructed for the longitudinal mode and monitors seven sensors and two actuators. Work also continues in refining a detailed nonlinear vehicle simulation which is used as a testbed for evaluating the performance of the health monitoring system.

Contents

Abstract . . . . **v**

Executive Summary . . . . **vii**

List of Figures . . . . **xi**

List of Tables . . . . **xiii**

List of Symbols . . . . **xv**

Chapter 1 Introduction **1**

Chapter 2 Vehicle Model and Simulation Development **5**

2.1 Nonlinear Model 6

2.1.1 Coordinate Systems 6

2.1.2 Rotational Equations of Motion 7

2.1.3 Translational Equations of Motion 8

2.2 Linear Model 9

2.3 Reduced-Order Model 13

2.3.1 Longitudinal Model 14

2.3.2 Lateral Model 17

Chapter 3 Fault Selection **19**
3.1 Sensor Fault Models 21
3.2 Actuator Fault Models 22

Chapter 4 Fault Detection Filter Design **25**
4.1 Fault Detection Filter Configuration 26
4.2 Eigenstructure Placement 29
4.3 Reduced-Order Observers 35

Chapter 5 Fault Detection Filter Evaluation **41**

Chapter 6 Residual Processing **47**

Chapter 7 Conclusions **59**

Appendix A Fault Detection Filter Background **63**

References **73**

List of Figures

Figure 5.1 Residuals for Fault Detection Filter One: Manifold Air Mass Sensor, Engine Speed Sensor and Forward Acceleration Sensor 43

Figure 5.2 Residuals for Fault Detection Filter Two: Pitch Rate Sensor, Forward Wheel Speed Sensor and Rear Wheel Speed Sensor 44

Figure 5.3 Residuals for Fault Detection Filter Three: Heave Acceleration Sensor, Pitch Rate Sensor and Rear Wheel Speed Sensor 45

Figure 5.4 Residuals for Fault Detection Filter Four: Throttle Actuator, Brake Actuator 46

Figure 6.1 Multi-Layer Perceptron Model 48

Figure 6.2 Residuals for Fault Detection Filter One: Manifold Air Mass Sensor, Engine Speed Sensor and Forward Acceleration Sensor Faults 55

Figure 6.3 Residuals for Fault Detection Filter Two: Pitch Rate Sensor, Forward Symmetric Wheel Speed Sensor and Rear Symmetric Wheel Speed Sensor 56

Figure 6.4 Residuals for Fault Detection Filter Three: Heave Accelerometer, Pitch Rate Sensor and Forward Symmetric Wheel Speed Sensor 57

Figure 6.5 Residuals for Fault Detection Filter Four: Throttle and Brake Actuators 58

List of Tables

Table 2.1 Eigenvalues for the Longitudinal Dynamics Using Three Model Reduction Methods 15

Table 2.2 Eigenvalues for the Lateral Dynamics Using Three Model Reduction Methods 17

List of Symbols

Vehicle Model		**Page**
$\underline{e}_x$, $\underline{e}_y$, $\underline{e}_z$	Unit vectors directed along earth-fixed coordinates	6
$\underline{a}_x$, $\underline{a}_y$, $\underline{a}_z$	Unit vectors $\underline{e}_x$, $\underline{e}_y$, $\underline{e}_z$ rotated about $\underline{e}_z$ through angle ϵ	6
$\underline{b}_x$, $\underline{b}_y$, $\underline{b}_z$	Unit vectors $\underline{a}_x$, $\underline{a}_y$, $\underline{a}_z$ rotated about $\underline{a}_y$ through angle ϕ	6
$\underline{c}_x$, $\underline{c}_y$, $\underline{c}_z$	Unit vectors directed along vehicle-fixed coordinates	6
X, Y, Z	Vehicle position in earth-fixed coordinates	6
x, y, z	Vehicle position in vehicle-fixed coordinates	6
v_x, v_y, v_z	Vehicle mass center velocity in the x, y, z direction	8
ϕ	Vehicle roll rotation about x-axis	6
θ	Vehicle pitch rotation about y-axis	6
ϵ	Vehicle yaw rotation about z-axis	6
ω_x, ω_y, ω_z	Vehicle angular velocity about the x, y, z axes	7
m_a	Intake manifold air mass	10

ω_e	Engine speed	10
ω_{fl}, ω_{fr}, ω_{rl}, ω_{rr}	Speed of the front left, front right, rear left and rear right tires	10
$\bar{\omega}_f$	Sum of the front wheel speeds: $\omega_{fl} + \omega_{fr}$	12
$\bar{\omega}_r$	Sum of the rear wheel speeds: $\omega_{rl} + \omega_{rr}$	12
$\tilde{\omega}_f$	Difference of the front wheel speeds: $\omega_{fl} - \omega_{fr}$	12
$\tilde{\omega}_r$	Difference of the rear wheel speeds: $\omega_{rl} - \omega_{rr}$	12
yr, $\dot{y}r$	Lateral deviation and lateral deviation rate from road center	10
ϵ_{des}	Desired vehicle yaw angle	10
α	Throttle angle (rad)	10
τ_b	Brake torque (Nm)	10
β	Tire steering angle	10
α_c	Commanded throttle angle	10
τ_{bc}	Commanded brake torque	10
β_c	Commanded tire steering angle	10
m	Vehicle mass	9
I_x, I_y, I_z	Moments of inertia of the sprung mass about the x, y, z axes	7
m_x, m_y, m_z	Moments tending to rotate the vehicle about the x, y, z axes	7
F_x, F_y, F_z	Force acting on the vehicle along the x, y, z axes	9
y_m	Measured engine manifold airmass (kg)	15
y_ω	Measured engine speed (rad/sec)	15
$y_{\ddot{x}}$	Measured longitudinal acceleration (m/sec^2)	15
$y_{\ddot{z}}$	Measured lateral acceleration (m/sec^2)	15
y_q	Measured pitch rate (rad/sec)	15
$y_{\omega fs}$	Measured front symmetric wheel speed (rad/sec)	15
$y_{\omega rs}$	Measured rear symmetric wheel speed (rad/sec)	15
A, B, C, D	Linearized dynamics system matrices	10
x, u, y	System state, control and output	9
$\tilde{x}$, $\tilde{u}$, $\tilde{y}$	Small perturbation of the system state, control and output	10
n, p, m	Dimensions of system state, control and output	63

Fault Detection Filter Development		**Page**
E_i	Sensor fault direction	21
F_i	Dynamics fault direction	21
$\hat{F}_i$	Complementary fault direction	71
μ_i	Sensor fault magnitude	21
m_i	Dynamics fault magnitude	63
q_i	Dimension of fault m_i	63
q	Number of dynamics faults $\{F_1, \ldots, F_q\}$	63
$\mathcal{X}$	System state space	63
$\mathcal{W}_i$	Invariant subspace associated with (C, A, F_i)	64
$\mathcal{W}_i^*$	Minimal invariant subspace associated with (C, A, F_i)	65
$\mathcal{T}_i$	Unobservability subspace associated with (C, A, F_i)	65
$\mathcal{T}_i^*$	Minimal unobservability subspace associated with (C, A, F_i)	66
$\hat{\mathcal{T}}_i^*$	Detection space associated with $(C, A, \hat{F}_i)$	71
$\mathcal{V}_i$	Space of invariant zero directions associated with (C, A, F_i)	66
$\mathcal{T}_0$	Fault detection filter complementary space	30
ν_i	Dimension of subspace $\mathcal{T}_i^*$	68
ν_0	Dimension of the complementary space $\mathcal{T}_0$	70
Λ_i, Λ_0	Set of assigned eigenvalues associated with $\mathcal{T}_i^*$, $\mathcal{T}_0$	70
$\hat{\Lambda}_i$	Set of assigned eigenvalues associated with $\hat{\mathcal{T}}_i^*$	68
λ_{i_j}, v_{i_j}	Eigenvalue, left eigenvector pair associated with $\hat{\mathcal{T}}_i^*$	69
V_{i_j}	Subspace from which the left eigenvector v_{i_j} is chosen	69
$\tilde{V}$	Matrix of fault detection filter left eigenvectors	32
L	Fault detection filter gain	20
$\hat{H}_i$	Output projection matrix associated with $\hat{\mathcal{T}}_i^*$	34
$\hat{x}$	Observer state estimate	20
e	Observer state estimate error	20
z_i	Fault detection filter residual associated with F_i	34
$\bar{\mathcal{X}}_i$	Factor space $\mathcal{X}_i/\mathcal{T}_i^*$	36

$\bar{P}_i$	Canonical projection $\bar{P}_i : \mathcal{X} \mapsto \bar{\mathcal{X}}_i$	36
$\bar{A}_i$, $\bar{C}_i$	Detection filter dynamics and output matrices induced on $\bar{\mathcal{X}}_i$	36

Neural Network Development		**Page**
$\mathcal{E}$	Average network output error over one epoch	50
N	Number of training patterns per epoch	50
d^j, y^j	Desired and measured network output for training set j	50
e^j	Network output error for training set j	50
h_0^j	Input vector to network for training set j	50
h_i^j	Input vector to network layer i for training set j	50
Φ_i^j	Bias vector to network layer i for training set j	50
W_i^j	Matrix of network weights applied at layer i for training set j	50
$S(\cdot)$	Synaptic activation function	49

Chapter 1
Introduction

A proposed transportation system with vehicles traveling at high speed, in close formation and under automatic control demands a high degree of system reliability. This requires a health monitoring and maintenance system capable of detecting a fault as it occurs, identifying the faulty component and determining a course of action that restores safe operation of the system. This report is concerned with vehicle fault detection and identification and describes a vehicle health monitoring system approach based on analytic redundancy.

Analytic redundancy methods for fault detection and identification use a modeled dynamic relationship between system inputs and measured system outputs to form a residual process. Nominally, the residual process is nonzero only when a fault has occurred and is zero at other times. For an observable system, this simple definition is met by the innovations process of any stable linear observer. A detection filter is a linear observer with the gain constructed so that when a fault occurs, the residual responds in a known and fixed direction. Thus, when a nonzero residual is detected, a fault can be announced and identified.

In applications it is unrealistic to expect that a residual process would be nonzero only when a fault has occurred. Sensor noise, process disturbances, system parameter variations, unmodeled dynamics and nonlinearities all contribute to the magnitude of a residual. There are many methods to reduce the impact of these effects on the residual but none reduce their effect to zero. This means that some threshold detection mechanism must be built.

A simple threshold detection mechanism announces a fault when the size of a residual exceeds some prescribed value. This prescribed value could be determined from empirical studies which balance a rate of false alarm against a rate of miss alarm. A more complicated residual processor might take into account the thresholds of all other residuals as well. Reasoning that if the probability of simultaneous failures is very small, no fault is announced when more than one residual exceeds a threshold. It is more likely that the nonzero residuals are caused by noise or nonlinearities or some cause other than multiple faults. A neural network residual processor is described in this report.

A complication arises when there are many possible faults because a fault detection filter can only be designed to detect a limited number of faults. This is related to the order of the vehicle dynamics. When more faults need to be identified, several fault detection filters have to be used with each filter designed to detect and identify some but not all possible faults. The vehicle fault detection system described in this report has four fault detection filters. This raises two difficult design issues. First, some and probably all faults will not be included in the design of one or more fault detection filters. When such a fault occurs, the residual of all filters will respond, even the residuals of the filters that do not have the fault included in their design. If a fault is not included in a fault detection filter design, the directional characteristics of the residual will be undefined and the fault cannot be properly identified. The challenge is to build a mechanism that recognizes when a fault detection filter is responding to a fault for which it has not been designed and then to exclude the residual of all such filters from the fault identification process. If it can be assumed that only one fault occurs at a time, then the residual processor can exclude the residual of any fault detection filters that point to two or more faults.

A second design issue is how the faults should be grouped and identification delegated among the fault detection filters. In a fault detection system that consists of a bank of fault detection filters and a residual processor such as a neural network, fault isolation is done through the combined effort of both system elements. The fault detection filter is a carefully tuned device that uses known dynamic relationships to isolate a fault. The neural network residual processor combines the residuals from several filters and resolves any ambiguity. It is suggested that identifying a fault among a group of dynamically similar faults requires the precision of and is best delegated to the fault detection filters. Furthermore, it is suggested that the reliability of the neural network training would be improved if the fault groups associated with each of the fault detection filters are dynamically dissimilar.

This paper is organized as follows. Section 2 describes the car models. Low-dimensional linear models are used for fault detection filter design. A high fidelity nonlinear model is used for evaluation and to obtain the linear models used for design. Section 3 describes the faults to be identified by the fault detection system. Section 4 describes the design of the fault detection filters. This includes how the faults are grouped for each fault detection filter design, how the fault detection filter eigenstructure placement is done and how reduced-order fault detection filters are formed. Section 5 presents an evaluation of the performance of the fault detection filters in a nonlinear simulation. Section 6 describes a fault detection filter residual processing system. Here a neural network is used to process residuals from all fault detection filters to detect and identify which if any fault has occurred. Finally, appendix A provides a very quick theoretical review of the Beard-Jones detection filter problem.

Chapter 2
Vehicle Model and Simulation Development

In this section, vehicle models are developed for the design and evaluation of fault detection filters. Three models are considered: (1) a six degree of freedom (DOF) nonlinear vehicle model, (2) a computer model obtained from the Berkeley PATH research team and derived in (Peng 1992), and (3) a linearized model used for detection filter design. The derivation of equations for the six DOF nonlinear model is independent of that used for the Berkeley simulation. The independent derivation was performed to be sure that we understood all the assumptions, definitions and issues which underlie the Berkeley simulation model. This exercise proved worthwhile in that we did uncover some differences between our model and the Berkeley model, and we have contacted them to clarify these differences. Resolution of these issues is pending.

All models can be used to describe a four-wheel-steering, four-wheel-drive vehicle. This report, however, only considers rear-wheel-drive vehicles. The road gradient and superelevation are assumed to be zero.

2.1 Nonlinear Model

Equations that describe the six degree of freedom motion of a vehicle are developed here. First, the coordinate systems are described. Then, the rotational equations of motion are developed followed by the translational equations of motion.

2.1.1 Coordinate Systems

The motion of the vehicle will be referred to an Earth-fixed reference frame E which is described by a right handed orthogonal axis system (X, Y, Z) fixed on the Earth. The unit vectors along the X, Y, Z-axes are $\underline{e}_x$, $\underline{e}_y$ and $\underline{e}_z$, respectively. A second reference frame C fixed in the sprung mass of the vehicle is described by a right handed orthogonal axis system (x, y, z) fixed along the central principal axes of the vehicle. The origin is at the vehicle mass center where x points in the forward direction, y points to the left, and z points upward. We assume that x and y are horizontal when the vehicle is at rest. Unit vectors $\underline{c}_x$, $\underline{c}_y$, and $\underline{c}_z$ are directed along x, y, and z, respectively. The orientation of C with respect to E is given by a sequence of three angular rotations. First, there is a yaw rotation ϵ about the aligned Z and z-axes. Let $\underline{a}_x$, $\underline{a}_y$ and $\underline{a}_z$ be unit vectors along the displaced x, y, z-axes. Then there is a roll rotation ϕ about the displaced y-axis $\underline{a}_y$. Let $\underline{b}_x$, $\underline{b}_y$ and $\underline{b}_z$ describe the directions of x, y, z-axes after this roll rotation. Last, there is a pitch rotation θ about the displaced x-axis $\underline{b}_x$. The unit vectors $\underline{c}_x$, $\underline{c}_y$ and $\underline{c}_z$ describe the final orientation of C. The relationships among the various unit vectors are

$$\begin{bmatrix} \underline{a}_x \\ \underline{a}_y \\ \underline{a}_z \end{bmatrix} = \begin{bmatrix} \cos\epsilon & \sin\epsilon & 0 \\ -\sin\epsilon & \cos\epsilon & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \underline{e}_x \\ \underline{e}_y \\ \underline{e}_z \end{bmatrix} \tag{2.1a}$$

$$\begin{bmatrix} \underline{b}_x \\ \underline{b}_y \\ \underline{b}_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \underline{a}_x \\ \underline{a}_y \\ \underline{a}_z \end{bmatrix} \tag{2.1b}$$

$$\begin{bmatrix} \underline{c}_x \\ \underline{c}_y \\ \underline{c}_z \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \underline{b}_x \\ \underline{b}_y \\ \underline{b}_z \end{bmatrix} \tag{2.1c}$$

2.1.2 Rotational Equations of Motion

The angular velocity of the car relative to the Earth is

$$\underline{\omega} = \dot{\epsilon}\underline{e}_z + \dot{\phi}\underline{a}_x + \dot{\theta}\underline{b}_y$$

Using the coordinate system transformations (2.1) the angular velocity is also given by

$$\begin{aligned}\underline{\omega} &= (\dot{\phi}\cos\theta - \dot{\epsilon}\cos\phi\sin\theta)\underline{c}_x + (\dot{\theta} + \dot{\epsilon}\sin\phi)\underline{c}_y + (\dot{\phi}\sin\theta + \dot{\epsilon}\cos\phi\cos\theta)\underline{c}_z \\ &= \omega_x\underline{c}_x + \omega_y\underline{c}_y + \omega_z\underline{c}_z\end{aligned}$$

Thus, the angular velocities of the car expressed in vehicle fixed axes, which are measured numbers, become

$$\begin{aligned}\begin{bmatrix}\omega_x \\ \omega_y \\ \omega_z\end{bmatrix} &= \begin{bmatrix}0 \\ \dot{\theta} \\ 0\end{bmatrix} + \begin{bmatrix}\cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta\end{bmatrix}\begin{bmatrix}\dot{\phi} \\ 0 \\ 0\end{bmatrix} + \begin{bmatrix}\cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta\end{bmatrix}\begin{bmatrix}1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi\end{bmatrix}\begin{bmatrix}0 \\ 0 \\ \dot{\epsilon}\end{bmatrix} \\ &= \begin{bmatrix}\cos\theta & 0 & -\cos\phi\sin\theta \\ 0 & 1 & \sin\phi \\ \sin\theta & 0 & \cos\phi\cos\theta\end{bmatrix}\begin{bmatrix}\dot{\phi} \\ \dot{\theta} \\ \dot{\epsilon}\end{bmatrix}\end{aligned}$$

If this expression is solved for the angular rates $\dot{\phi}$, $\dot{\theta}$ and $\dot{\epsilon}$, one obtains the rotational kinematic equations of motion:

$$\begin{bmatrix}\dot{\phi} \\ \dot{\theta} \\ \dot{\epsilon}\end{bmatrix} = \begin{bmatrix}\cos\theta & 0 & \sin\theta \\ \sin\theta\tan\phi & 1 & -\cos\theta\tan\phi \\ -\sin\theta\cos^{-1}\phi & 0 & \cos\theta\cos^{-1}\phi\end{bmatrix}\begin{bmatrix}\omega_x \\ \omega_y \\ \omega_z\end{bmatrix} \tag{2.2}$$

The rotational dynamic equations governing the angular motions of the vehicle are obtained from the Euler equations:

$$\dot{\omega}_x = \frac{m_x}{I_x} + \omega_y\omega_z\frac{I_y - I_z}{I_x} \tag{2.3a}$$

$$\dot{\omega}_y = \frac{m_y}{I_y} + \omega_z\omega_x\frac{I_z - I_x}{I_y} \tag{2.3b}$$

$$\dot{\omega}_z = \frac{m_z}{I_z} + \omega_x\omega_y\frac{I_x - I_y}{I_z} \tag{2.3c}$$

The applied moments m_x, m_y and m_z come from aerodynamic forces and interaction forces between the tires and pavement. Expressions for these moments are discussed in a later section.

2.1.3 Translational Equations of Motion

The position vector from an Earth-fixed point to the center of mass of the car may be described in terms of the earth-fixed unit vectors $\underline{e}_x$, $\underline{e}_y$ and $\underline{e}_z$ or the vehicle-fixed unit vectors $\underline{c}_x$, $\underline{c}_y$ and $\underline{c}_z$. Thus,

$$\begin{aligned} \underline{p} &= X\underline{e}_x + Y\underline{e}_y + Z\underline{e}_z \\ &= x\underline{c}_x + y\underline{c}_y + z\underline{c}_z \end{aligned}$$

The velocity of the center of mass of the car then becomes,

$$\begin{aligned} \underline{v} &= \dot{X}\underline{e}_x + \dot{Y}\underline{e}_y + \dot{Z}\underline{e}_z \\ &= (\dot{x}\underline{c}_x + \dot{y}\underline{c}_y + \dot{z}\underline{c}_z) + (\omega_x\underline{c}_x + \omega_y\underline{c}_y + \omega_z\underline{c}_z) \times (x\underline{c}_x + y\underline{c}_y + z\underline{c}_z) \\ &= (\dot{x} - y\omega_z + z\omega_y)\underline{c}_x + (\dot{y} - zw_x + xw_z)\underline{c}_y + (\dot{z} - xw_y + yw_x)\underline{c}_z \\ &= v_x\underline{c}_x + v_y\underline{c}_y + v_z\underline{c}_z \end{aligned}$$

Solving for $\dot{x}$, $\dot{y}$, and $\dot{z}$ in terms of x, y, z and ω_x, ω_y, ω_z one obtains:

$$\dot{x} = v_x + y\omega_z - z\omega_y \tag{2.4a}$$

$$\dot{y} = v_y + z\omega_x - x\omega_z \tag{2.4b}$$

$$\dot{z} = v_z + x\omega_y - y\omega_x \tag{2.4c}$$

The acceleration of the center of mass of the car in both earth-fixed and vehicle-fixed axes is

$$\begin{aligned} \underline{a} &= \ddot{X}\underline{e}_x + \ddot{Y}\underline{e}_y + \ddot{Z}\underline{e}_z \\ &= (\dot{v}_x\underline{c}_x + \dot{v}_y\underline{c}_y + \dot{v}_z\underline{c}_z) + (\omega_x\underline{c}_x + \omega_y\underline{c}_y + \omega_z\underline{c}_z) \times (v_x\underline{c}_x + v_y\underline{c}_y + v_z\underline{c}_z) \\ &= (\dot{v}_x + \omega_y v_z - \omega_z v_y)\underline{c}_x + (\dot{v}_y + \omega_z v_x - \omega_x v_z)\underline{c}_y + (\dot{v}_z + \omega_x v_y - \omega_y v_x)\underline{c}_z \end{aligned}$$

Expressing the forces acting on the vehicle $\underline{F}$ as

$$\underline{F} = F_x\underline{c}_x + F_y\underline{c}_y + F_z\underline{c}_z$$

and using Newtons 2nd Law, $\underline{F} = m\underline{a}$, leads to the following dynamic translational equations of motion.

$$\dot{v}_x = \omega_z v_y - \omega_y v_z + \frac{F_x}{m} \tag{2.5a}$$

$$\dot{v}_y = \omega_x v_z - \omega_z v_x + \frac{F_y}{m} \tag{2.5b}$$

$$\dot{v}_z = \omega_y v_x - \omega_x v_y + \frac{F_z}{m} \tag{2.5c}$$

As before, the applied forces F_x, F_y and F_z come from gravity, aerodynamic forces and interaction forces between the tires and pavement. Equations (2.2), (2.3), (2.4) and (2.5) describe the motion of the car provided the applied forces and moments are known. These expressions would be required if our objective were to construct a complete analytical model or a computer simulation. At the present time, we have not taken this next step, and have instead used the Berkeley simulation model for subsequent work. More work on force and moment models may be attempted at a later date.

2.2 Linear Model

The nonlinear model in the previous section was generated primarily to better understand and verify the Berkeley model. In this section, we generate a linearized model directly from the Berkeley model. This will be done numerically rather than analytically. The procedure is as follows.

First, a computer run is made in which the car goes straight at a constant speed of 25 m/s ($\simeq$ 56 mph) to obtain steady state values for each state. The nonlinear model is then linearized about this nominal operating point using the central difference method. The use of an analytical approach, that is taking partial derivatives, is impractical because this model is too complicated.

The nonlinear model has the form :

$$\dot{x} = f(x, u) \tag{2.6a}$$

$$y = Cx + D\dot{x} \tag{2.6b}$$

Suppose the nominal operating point is $(x_0,\ u_0)$ where $f(x_0, u_0) = 0$. Take perturbations $\tilde{x}$, $\tilde{u}$ about the nominal point, that is, let

$$\begin{aligned} x &= x_0 + \tilde{x} \\ u &= u_0 + \tilde{u} \end{aligned}$$

Also approximate $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial u}$ as

$$\begin{aligned} \frac{\partial f}{\partial x} &\approx \frac{\Delta f}{\Delta x} = \left.\frac{f(x+\tilde{x},u) - f(x-\tilde{x},u)}{2\tilde{x}}\right|_{x=x_0,u=u_0} \\ \frac{\partial f}{\partial u} &\approx \frac{\Delta f}{\Delta u} = \left.\frac{f(x,u+\tilde{u}) - f(x,u-\tilde{u})}{2\tilde{u}}\right|_{x=x_0,u=u_0} \end{aligned}$$

Equation (2.6a) may now be approximated as

$$\dot{x}_0 + \dot{\tilde{x}} = f(x_0, u_0) + \left.\frac{\partial f}{\partial x}\right|_{x=x_0,u=u_0} \tilde{x} + \left.\frac{\partial f}{\partial u}\right|_{x=x_0,u=u_0} \tilde{u} + \cdots$$

Dropping out higher order terms and using the approximations given above for the partial derivatives, one obtains

$$\begin{aligned} \dot{\tilde{x}} &= A\tilde{x} + B\tilde{u} \\ \tilde{y} &= C\tilde{x} + D\dot{\tilde{x}} \\ &= (C + DA)\tilde{x} + DB\tilde{u} \end{aligned}$$

where

$$\begin{aligned} \tilde{x} &= [m_a \quad w_e \quad v_x \quad x \quad v_y \quad y \quad v_z \quad z \quad \phi \quad \dot{\phi} \quad \theta \quad \dot{\theta} \quad \epsilon \quad \dot{\epsilon} \\ &\qquad w_{fl} \quad w_{fr} \quad w_{rl} \quad w_{rr} \quad X \quad Y \quad yr \quad \dot{y}r \quad \epsilon_{des} \quad \alpha \quad \tau_b \quad \beta]^T \\ \tilde{y} &= [m_a \quad w_e \quad \dot{v_x} \quad \dot{v_y} \quad \dot{v_z} \quad \dot{\phi} \quad \dot{\theta} \quad \dot{\epsilon} \quad w_{fl} \quad w_{fr} \quad w_{rl} \quad w_{rr}]^T \\ \tilde{u} &= [\alpha_c \quad \tau_{bc} \quad \beta_c]^T \\ A &= \left.\left[\frac{\Delta f}{\Delta x}\right]\right|_{x=x_0,u=u_0} \\ B &= \left.\left[\frac{\Delta f}{\Delta u}\right]\right|_{x=x_0,u=u_0} \end{aligned}$$

and where A is a 26×26 real matrix and B is a 26×3 real matrix. Symbols in $\tilde{x}$, $\tilde{y}$ and $\tilde{u}$ are defined in the list of symbols.

Several sizes of perturbations must be taken to find one that gives the best approximation of the partial derivatives. If the perturbation is too small, there is a truncation error in computing the difference $f(x+\tilde{x},u)-f(x-\tilde{x},u)$. If the perturbation is too large, a roundoff error occurs in computing $f(x+\tilde{x},u)$ and $f(x-\tilde{x},u)$; also nonlinearities become important. According to our experience, $\frac{\tilde{x}}{x}$ and $\frac{\tilde{u}}{u} \approx 10^{-4}$ is a good rule for selecting the size of the perturbation when using the central differences method. The resulting linear model can then be tested in a simulation to see how well it describes the nonlinear model over the speed range of 23 m/s to 27 m/s. When this was done, we found that the errors were under 10%.

The linear model generated as described above was intended for use in designing the fault detection filters. This model has 26 states. Before using the model for filter design, we decided to try to simplify the model to the extent possible without significant loss of accuracy. The model simplification was accomplished in two steps, the first of which resulted in no loss of accuracy.

By inspection of the equations, it was found possible to rearrange the sequence of states such that the linearized equations assume the following partitioned form:

$$\begin{aligned}
\dot{\tilde{x}} &= \begin{bmatrix} \dot{\tilde{x}}_3 \\ \dot{\tilde{x}}_4 \end{bmatrix} = \begin{bmatrix} A_1 & 0 \\ A_2 & A_3 \end{bmatrix} \begin{bmatrix} \tilde{x}_3 \\ \tilde{x}_4 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \tilde{u} \\
\tilde{y} &= \begin{bmatrix} C_1 & 0 \end{bmatrix} \begin{bmatrix} \tilde{x}_3 \\ \tilde{x}_4 \end{bmatrix}
\end{aligned}$$

where

$$\begin{aligned}
\tilde{x}_3 &= [m_a \quad w_e \quad v_x \quad v_z \quad z \quad \theta \quad \dot{\theta} \quad w_{fl} \quad w_{fr} \quad w_{rl} \quad w_{rr} \quad \alpha \quad \tau_b \quad v_y \quad \phi \quad \dot{\phi} \quad \dot{\epsilon} \quad \beta]^T \\
\tilde{x}_4 &= [x \quad X \quad y \quad \epsilon \quad Y \quad yr \quad \dot{yr} \quad \epsilon_{des}]^T
\end{aligned}$$

In this form, we see that both $\tilde{x}_3$ and $\tilde{y}$ are independent of $\tilde{x}_4$. Thus $\tilde{x}_4$ can be deleted from the model without affecting the transfer function from $\tilde{u}$ to $\tilde{y}$. Based on this observation,

$\tilde{x}_4$ is removed from the model, which then becomes

$$\begin{aligned} \dot{\tilde{x}}_3 &= A_1\tilde{x}_3 + B_1\tilde{u} \\ \tilde{y} &= C_1\tilde{x}_3 \end{aligned}$$

where A_1 is an 18×18 matrix, B_1 is an 18×3 matrix and C_1 is a 12×18 matrix.

If the four wheel speed state variables w_{fl}, w_{fr}, w_{rl}, w_{rr} are replaced by four new state variables $\bar{w}_f$, $\bar{w}_r$, $\tilde{w}_f$, $\tilde{w}_r$ defined as:

$$\begin{aligned} \bar{w}_f &= w_{fl} + w_{fr} \\ \bar{w}_r &= w_{rl} + w_{rr} \\ \tilde{w}_f &= w_{fl} - w_{fr} \\ \tilde{w}_r &= w_{fl} - w_{rr} \end{aligned}$$

then the model exactly decouples into two subsystems. These are the longitudinal and lateral dynamics, that is,

$$\begin{bmatrix} \dot{\tilde{x}}_1 \\ \dot{\tilde{x}}_2 \end{bmatrix} = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{bmatrix} + \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix} \begin{bmatrix} \tilde{u}_1 \\ \tilde{u}_2 \end{bmatrix}$$

where

$$\begin{aligned} \tilde{x}_1 &= [m_a \quad w_e \quad \dot{x} \quad \dot{z} \quad z \quad \theta \quad \dot{\theta} \quad \bar{w}_f \quad \bar{w}_r \quad \alpha \quad \tau_b]^T \\ \tilde{x}_2 &= [\tilde{w}_f \quad \tilde{w}_r \quad \dot{y} \quad \phi \quad \dot{\phi} \quad \dot{\epsilon} \quad \beta]^T \\ \tilde{u}_1 &= [\alpha_c \quad \tau_{bc}]^T \\ \tilde{u}_2 &= \beta_c \end{aligned}$$

Therefore, the longitudinal model becomes:

$$\dot{\tilde{x}}_1 = A_1\tilde{x}_1 + B_1\tilde{u}_1$$

and the lateral model becomes:

$$\dot{\tilde{x}}_2 = A_2\tilde{x}_2 + B_2\tilde{u}_2$$

2.3 Reduced-Order Model

Previous manipulation has not involved any approximation. For further model simplification, some approximation must occur. First, the actuator dynamics are neglected because they are relatively fast and also they are in series with the other dynamics. At this point, we are more concerned about simplifying the highly coupled dynamics and will return to consider the actuator dynamics later. Hence, the actuator dynamic states are deleted from the model. So the states for the longitudinal model are $\tilde{x}_1$ and for the lateral model are $\tilde{x}_2$.

$$\begin{aligned} \tilde{x}_1 &= [m_a \quad w_e \quad \dot{x} \quad \dot{z} \quad z \quad \theta \quad \dot{\theta} \quad \bar{w}_f \quad \bar{w}_r]^T \\ \tilde{x}_2 &= [\tilde{w}_f \quad \tilde{w}_r \quad \dot{y} \quad \phi \quad \dot{\phi} \quad \dot{\epsilon}]^T \end{aligned}$$

After the linear models are derived, the first thing one should do is check the eigenvalues. Then, three approaches are presented to get reduced-order models. The first approach one may consider to reduce the model is to set the derivatives of certain fast states to zero. Using this philosophy, states with large negative eigenvalues can be dropped. However, a correction should be made using the deleted states to remove the steady state error. Consider a linear system modeled as :

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned}$$

Suppose this model is written in a partitioned form.

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u \\ y &= \begin{bmatrix} C_1 & C_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + Du \end{aligned}$$

where x_2 contains the 'fast states'. Set the derivative of x_2 to zero and solve the resulting equations for x_2 as a function of x_1 and u. This leads to

$$x_2 = -A_{22}^{-1} A_{21} x_1 - A_{22}^{-1} B_2 u$$

Substitute this result into the expressions for $\dot{x}_1$ and y to obtain the reduced order model:

$$\begin{aligned} \dot{x}_1 &= \left[A_{11} - A_{12}A_{22}^{-1}A_{21}\right] x_1 + \left[B_1 - A_{12}A_{22}^{-1}B_2\right] u \\ y &= \left[C_1 - C_2A_{22}^{-1}A_{21}\right] x_1 + \left[D - C_2A_{22}^{-1}B_2\right] u \end{aligned}$$

this model preserves the static input-output relationships.

A second approach is to use balanced realization before implementing the method just described. Balancing refers to an algorithm which finds a realization that has equal and diagonal controllability and observability grammians. The diagonal of the joint grammian $g(i)$ can be used to reduce the order of the model. Since $g(i)$ reflects the combined controllability and observability of individual states, it is reasonable to remove those states from the model that have a small $g(i)$. Elimination of these states retains the most important input-output characteristics of the original system. After balanced realization has been done, the first method is used to obtain a reduced order model.

A third approach is a little different from the second one. After balanced realization has been done, a truncation is used instead of the first method. For example, if the full-order model is

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u \\ y &= \begin{bmatrix} C_1 & C_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + Du \end{aligned}$$

then, the reduced-order model is

$$\dot{x}_1 = A_{11}x_1 + B_1u$$

This is the approach originally proposed by Moore (Moore 1981). Using this approach it is possible to calculate a bound on the error introduced by deleting states.

2.3.1 Longitudinal Model

At the end of the previous section, section 2.2 which deals with the linear model, a decoupled longitudinal model is developed. Its eigenvalues are -212.11, -166.04, -31.46, -26.27, -0.04, $-2.3 \pm 6.65i$ and $-1.53 \pm 5.69i$. Observe that two of these eigenvalues are significantly larger than the rest. From this we conclude that at least two state variables can be dropped. In method one, by looking at the eigenvectors corresponding to the large eigenvalues, we find that the two fast mode states are the sum of the front wheel speeds $\bar{w}_f$ and the sum of the rear wheel speeds $\bar{w}_r$. So, these two states are dropped to get a seventh-order model. In methods 2 and 3, two states with smallest grammians are dropped. These methods combine the states in such a way that they lose their physical significance, so we can not explicitly identify the states that are being deleted. Here are some results using the three methods for model reduction described earlier.

Order Reduction Method	Eigenvalues				
Method 1.	-33.05	-25.85	-0.0484	$-2.26 \pm 6.71i$	$-1.57 \pm 5.67i$
Method 2.	-31.58	-26.23	-0.0449	$-2.32 \pm 6.65i$	$-1.54 \pm 5.69i$
Method 3.	-32.08	-26.05	-0.0449	$-2.25 \pm 6.04i$	$-1.78 \pm 5.82i$

Table 2.1: Eigenvalues for the Longitudinal Dynamics Using Three Model Reduction Methods.

The eigenvalues of each reduced-order model are given in table 2.1. The second method is the best because the eigenvalues are closer to the true eigenvalues. This method uses the balanced realization and drops unimportant states by letting their derivatives be zero. One can also perform another test to see which method is best. That method is based on frequency response. Bode diagrams for each input to each output are plotted to see their responses to frequencies from 10^{-1} to 10^{2} rad/s. The reason for choosing this frequency range is that it roughly corresponds to that of the control inputs to a car. The Bode diagrams also show that the frequency response of a model obtained with the second method is closest to that of the full-order model.

The seven-state model involves the longitudinal dynamics only. No lateral dynamics and no actuator dynamics are included. The states have no physical significance because they are derived from the balanced realization as stated in section 2.3. The measured outputs are

y_m	Engine manifold air mass (kg).

y_ω Engine speed (rad/sec).

$y_{\ddot{x}}$ longitudinal acceleration (m/sec^2).

$y_{\ddot{z}}$ heave acceleration (m/sec^2).

y_q Pitch rate (rad/sec).

$y_{y_{\text{fs}}}$ Forward symmetric wheel speed (rad/sec).

$y_{y_{\text{rs}}}$ Rear symmetric wheel speed (rad/sec).

and the control inputs are

α Throttle angle (deg).

β Brake torque (Nm).

The system matrices are given by

$$A = \begin{bmatrix} -0.0514 & -0.2203 & 0.2670 & -0.0102 & 0.0145 & 0.0084 & -0.0074 \\ -0.2984 & -7.7825 & 18.5490 & -0.9359 & 0.1522 & 0.2418 & 0.0463 \\ -0.3247 & -19.1948 & -49.4179 & -3.2002 & -4.9689 & -2.3224 & -0.0652 \\ 0.0440 & 2.2616 & 14.8614 & -2.1396 & 6.4462 & -0.2283 & 0.0394 \\ 0.0216 & 1.0707 & 8.3103 & -7.1707 & -0.6642 & -0.2614 & 0.9221 \\ 0.0116 & 0.5739 & 3.6890 & -1.0911 & -0.6573 & -1.0090 & 5.9643 \\ 0.0150 & 0.7490 & 4.6068 & -1.4672 & -1.0353 & -6.5849 & -2.5807 \end{bmatrix} \tag{2.7a}$$

$$B = \begin{bmatrix} 0.9509 & -0.0341 \\ 2.8861 & -0.0107 \\ 2.9813 & 0.0082 \\ -0.4068 & 0.0116 \\ -0.2001 & 0.0185 \\ -0.1069 & 0.0040 \\ -0.1389 & 0.0109 \end{bmatrix} \tag{2.7b}$$

$$C = \begin{bmatrix} 0.0080 & 0.4605 & 0.3771 & 0.1010 & 0.0541 & 0.0340 & -0.0129 \\ 0.7411 & 2.8381 & -2.9156 & 0.1484 & -0.0552 & -0.0503 & -0.0048 \\ 0.0027 & 0.1650 & -0.2533 & 0.0732 & -0.0157 & 0.0094 & -0.0005 \\ 0.0000 & -0.0006 & -0.0007 & -0.0207 & -0.0500 & -0.0446 & 0.0694 \\ -0.0000 & -0.0024 & 0.0049 & 0.0108 & 0.0207 & -0.0026 & 0.0009 \\ 0.4222 & -0.1429 & 0.0360 & 0.2242 & -0.1731 & -0.0138 & 0.1051 \\ 0.4217 & 0.1241 & -0.4239 & -0.2778 & -0.0356 & -0.0740 & 0.0579 \end{bmatrix} \tag{2.7c}$$

$$D = \begin{bmatrix} 0.0000 & -0.0000 \\ -0.0005 & 0.0004 \\ 0.0010 & -0.0020 \\ -0.0000 & 0.0001 \\ 0.0000 & -0.0000 \\ -0.0001 & -0.0008 \\ 0.0013 & -0.0010 \end{bmatrix} \tag{2.7d}$$

2.3.2 Lateral Model

At the end of previous section, section 2.2 dealing with the Linear Model, we also have a decoupled lateral model. Its eigenvalues are -205.91 , -133.45 , $-3.29 \pm 5.96i$, $-8.39 \pm 1.43i$. This model also contains two high frequency modes, so we again conclude that two state variables can be dropped. By looking at the corresponding eigenvectors, we learn that the two fast mode states are the difference of the front wheel speeds $\tilde{w}_f$ and the difference of the rear wheel speeds $\tilde{w}_r$. So following the procedure of method 1, these two states are dropped to get a fourth-order model. In methods 2 and 3, two states with the smallest grammians are dropped. Here are some results by using these three methods for model reduction.

Order Reduction Method	Eigenvalues	
Method 1.	$-2.95 \pm 5.78i$	$-8.80 \pm 2.02i$
Method 2.	$-3.29 \pm 6.02i$	$-8.31 \pm 2.04i$
Method 3.	$-4.18 \pm 5.63i$	$-9.49 \pm 6.97i$

Table 2.2: Eigenvalues for the Lateral Dynamics Using Three Model Reduction Methods.

The eigenvalues of each lateral reduced-order model are given in table 2.2. Once again, the second method produces the best result. That is where we use balanced realization and drop states by letting their derivatives be zero. Bode diagrams for each input to each output also were plotted to see their responses to frequency from 10^{-1} to 10^{2} rad/s. Looking at the Bode diagrams confirmed that the second method is best.

Chapter 3
Fault Selection

Analytic redundancy is an approach to health monitoring that compares dissimilar instruments using a detailed model of the system dynamics. Therefore, to detect a fault in a given sensor, there must be a dynamic relationship between the sensor and other sensors or actuators. That is, the information provided by a monitored sensor must, in some form, also be provided by other sensors. Analytic redundancy also can be used to effectively monitor the health of system actuators and even the dynamic behavior of the system itself. But, as with sensors, if some part of the vehicle is to be monitored for proper operation, then that part has to produce some observable dynamic effect.

In automated vehicles, these requirements preclude monitoring, for example, nonredundant sensors such as obstacle detection sensors or lane position sensors. The information provided by a radar or infrared sensor designed to detect objects in the vehicle's path has no dynamic correlation with other sensors on the vehicle. A sensor that detects the vehicle's position in a lane is the only sensor that can provide this information. Similarly, the health of the actuator that controls the position of the driver's window is easily monitored by the driver. But, unless specialized sensors are installed, no other part of the car is affected by the operation of this actuator and there is no analytic redundancy.

Before describing how faults are modeled, it is necessary to describe how a fault detection filter works. Most of the detail is left to appendix A. For a thorough background, several references are available, a few of which are (Douglas 1993), (White and Speyer 1987) and (Massoumnia 1986). Consider a linear time-invariant system with q failure modes and no disturbances or sensor noise

$$\dot{x} = Ax + Bu + \sum_{i=1}^{q} F_i m_i \tag{3.1a}$$

$$y = Cx + Du \tag{3.1b}$$

The system variables x, u, y and the m_i belong to real vector spaces and the system maps A, B, C, D and the F_i are of compatible dimensions. Assume that the input u and the output y both are known. The F_i are the failure signatures. They are known and fixed and model the directional characteristics of the faults. The m_i are the failure modes and model the unknown time-varying amplitude of faults. The m_i do not have to be scalar values.

A fault detection filter is a linear observer that, like any other linear observer, forms a residual process sensitive to unknown inputs. Consider a full-order observer with dynamics and residual

$$\dot{\hat{x}} = (A + LC)\hat{x} + Bu - Ly \tag{3.2a}$$

$$r = C\hat{x} + Du - y \tag{3.2b}$$

Form the state estimation error $e = \hat{x} - x$ and the dynamics and residual are

$$\begin{aligned} \dot{e} &= (A + LC)e - \sum_{i=1}^{q} F_i m_i \\ r &= Ce \end{aligned}$$

In steady-state, the residual is driven by the faults when they are present. If the system is (C, A) observable, and the observer dynamics are stable, then in steady-state and in the absence of disturbances and modeling errors, the residual r is nonzero only if a fault has occurred, that is, if some m_i is nonzero. Furthermore, when a fault does occur, the residual is nonzero except in certain theoretically relevant but physically unrealistic situations. This means that any stable observer can detect the presence of a fault. Simply monitor the residual and when it is nonzero a fault has occurred.

In addition to detecting a fault, a fault detection filter provides information to determine which fault has occurred. An observer such as (3.2) becomes a fault detection filter when the observer gain L is chosen so that the residual has certain directional properties that immediately identify the fault. The gain is chosen to partition the residual space where each partition is uniquely associated with one of the design fault directions F_i. A fault is identified by projecting the residual onto each of the residual subspaces and then determining which projections are nonzero.

Before the fault detection filter design (3.2) can begin, a system model with faults has to be found with the form (3.1). Seven sensors and two actuators are associated with the linearized longitudinal vehicle dynamics described in section 2.3.1. The sensors measure the engine manifold airflow and engine speed, the vehicle forward and heave accelerations, the pitch rate and the averaged speed of the forward wheels and the averaged speed of the rear wheels. The actuators control the engine throttle and the brake torque.

3.1 Sensor Fault Models

Sensor faults can be modeled as an additive term in the measurement equation

$$y = Cx + E_i \mu_i \tag{3.3}$$

where E_i is a column vector of zeros except for a one in the i^{th} position and where μ_i is an arbitrary time-varying real scalar. Now, for the fault detection filter design, faults are expressed as additive terms to the system dynamics as in (3.1). Sensor faults may be expressed in this way, as explained in (Douglas 1993), where the fault E_i in (3.3) is equivalent to a two-dimensional fault F_i

$$\dot{x} = Ax + F_i m_i \qquad \text{with } F_i = \left[F_i^1, F_i^2\right]$$

and where the directions F_i^1 and F_i^2 are given by

$$\begin{aligned} E_i &= CF_i^1 && (3.4a) \\ F_i^2 &= AF_i^1 && (3.4b) \end{aligned}$$

Using the linearized longitudinal dynamics of section 2.3.1, an engine manifold airflow measurement is given by the first element of the system output (2.7). Therefore, any fault in the engine manifold airflow sensor can be modeled as an additive term in the measurement equation as in (3.3)

$$y = Cx + Du + E_{y_m}\mu_{y_m}$$

where

$$E_{y_m} = \begin{bmatrix} 1, & 0, & 0, & 0, & 0, & 0, & 0 \end{bmatrix}^T$$

and where μ_{y_m} is an arbitrary time-varying real scalar. An equivalent two-dimensional fault F_{y_m} found by solving (3.4) is

$$F_{y_m} = \begin{bmatrix} 0.1145 & 0.0232 \\ 0.9439 & 8.0234 \\ 0.3365 & -94.5225 \\ -2.8676 & 32.1570 \\ 3.5965 & 30.7195 \\ 21.5405 & 73.7392 \\ 15.5799 & -179.3062 \end{bmatrix}$$

Other vehicle sensor fault directions are found in the same way.

3.2 Actuator Fault Models

A fault in a control input is modeled as an additive term in the system dynamics. In the case of a fault appearing at the input of an actuator, that is the actuator command, the fault has the same direction as the associated column of the system B matrix. A fault appearing at the output of an actuator, the actuator position, has the same direction as the associated column of the system A matrix.

For the vehicle longitudinal dynamics developed in section 2.3, the actuator dynamics are relatively fast and, in an approximation, are removed from the system model. Thus, the control inputs are applied directly to the car dynamics through a column of the B matrix and to the sensor outputs through a column of the feedforward D matrix. So, for this system a control input fault has three directions. One fault direction is the B matrix column. The other two directions come from treating the D matrix column as if it were a sensor fault which is explained above.

The engine throttle control is the first element of the system input so one direction of an engine throttle control fault is the first column of the B matrix from (2.7)

$$F_\alpha^1 = \begin{bmatrix} 0.9509 \\ 2.8861 \\ 2.9813 \\ -0.4068 \\ -0.2001 \\ -0.1069 \\ -0.1389 \end{bmatrix} \tag{3.5}$$

Because the linear model (2.7) has a control feedforward term, a throttle control fault also shows up directly in the system outputs in a direction given by the first column of the D matrix, that is,

$$E_\alpha = \begin{bmatrix} 1.1894e-07 \\ -4.6361e-04 \\ 1.0324e-03 \\ -3.6799e-05 \\ 2.0846e-06 \\ -9.9046e-05 \\ 1.3114e-03 \end{bmatrix}$$

As with a sensor fault, this direction E_α leads to a two-dimensional dynamics fault direction given by solving (3.4). Together with (3.5), an engine throttle fault is modeled as a three-dimensional dynamics fault

$$F_\alpha = \begin{bmatrix} 0.9509 & -0.0023 & -0.0002 \\ 2.8861 & -0.0021 & -0.0462 \\ 2.9813 & -0.0074 & -0.0911 \\ -0.4068 & -0.0220 & 0.0911 \\ -0.2001 & 0.0300 & 0.1458 \\ -0.1069 & 0.1758 & 0.5580 \\ -0.1389 & 0.1273 & -1.5206 \end{bmatrix}$$

A fault model for the brake torque is developed in the same way and is given by

$$F_\beta = \begin{bmatrix} -3.4075 & 0.0423 & 0.0004 \\ -1.0743 & 0.0370 & 0.0831 \\ 0.8178 & 0.1236 & 0.1275 \\ 1.1552 & 0.3200 & -0.1217 \\ 1.8522 & -0.4697 & -0.2077 \\ 0.3980 & -2.8685 & -0.9034 \\ 1.0870 & -2.0732 & 2.4854 \end{bmatrix}$$

Chapter 4
Fault Detection Filter Design

The fault detection filter design process consists of three steps. First, determine how many fault detection filters are needed and if more than one, which filters will detect and identify which faults. In a detection filter, the state estimation error in response to a fault in the direction F_i remains in a state subspace $\mathcal{T}_i^*$, a detection space. The ability to identify a fault, to distinguish one fault from another, requires for an observable system that the detection spaces be independent. Therefore, the number of faults that can be detected and identified by a fault detection filter is limited by the size of the state space and the sizes of the detection spaces associated with each of the faults. If the problem considered has more faults than can be accommodated by one fault detection filter, then a bank of filters will have to be constructed. The vehicle health monitoring system described in this report considers nine system faults: seven sensor faults and two actuator faults. Since the linearized longitudinal dynamics have only seven states, clearly more than one fault detection filter is needed. In fact, a bank of four fault detection filters is built.

Second, design the fault detection filters using eigenstructure assignment while making sure that the eigenvectors are not ill-conditioned. The essential feature of a fault detection filter is the detection space structure embedded in the filter dynamics. An eigenvector assignment design algorithm explicitly places eigenvectors to span these subspaces and provides a mechanism to ensure that the eigenvectors are as well-conditioned as possible. System disturbances, sensor noise and system parameter variations are not considered in the fault detection filter designs described in this report. For such a benign environment, the filter designs are based on spectral considerations only; there is little else that can be used to distinguish a good design from a bad design. The next design phase will include models of a more realistic operating environment.

Third, form the reduced-order detection filters so that later, when noise and disturbance models are developed, they can be included in the filters without need for a complete redesign. A product of the detection space structure is that with respect to each fault there is a large unobservable subspace. If the observable subspace that remains is factored out, a reduced-order fault detection filter is formed which has no special subspace structure. One reduced-order filter is formed for each fault in the fault detection filter design. Now, because there is no special structure, the gains of the reduced-order filters can be chosen arbitrarily. For example, the gains could minimize the residuals response to disturbances or sensor noise in an $\mathcal{H}_2$ or $\mathcal{H}_\infty$ norm. The important point is that the form of the reduced-order fault detection filters is determined independently of and without any need for disturbance or sensor noise models. Furthermore, once these models are designed, the reduced-order filters are easily adjusted.

4.1 Fault Detection Filter Configuration

To determine how many and which faults may be included in a fault detection filter design, the detection spaces for each of the faults, also called unobservability subspaces, are formed. A detection space for a fault F_i is denoted by $\mathcal{T}_i^*$. First, the dimensions of the detection spaces are needed. Since the detection spaces are independent subspaces, the sum of their dimensions for any given fault detection filter cannot exceed the dimension of the state-space. Second, the detection spaces for any given fault detection filter have to be output separable and mutually detectable. These concepts are described in detail in appendix A but briefly, output separability means that the output subspaces $C\mathcal{T}_i^*$ are independent. Mutual detectability means that the sum of the detection spaces $\sum \mathcal{T}_i^*$ is an unobservability subspace. This condition ensures that the spectrum of the detection filter can be assigned arbitrarily.

In practice it is just as easy to find a basis for the detection space as it is to find only the dimension. The method used here is suggested for numerical stability in (Wonham 1985) and is described in appendix A. Briefly, for a fault F_i, the approach is to find the minimal (C, A)-invariant subspace $\mathcal{W}_i^*$ that contains F_i and then to find the invariant zero directions of the triple (C, A, F_i), if any. If the invariant zero directions are denoted by $\mathcal{V}_i$, then the minimal unobservability subspace $\mathcal{T}_i^*$ is given by

$$\mathcal{T}_i^* = \mathcal{W}_i^* + \mathcal{V}_i$$

The longitudinal linear model of section 2.3 has seven states, seven sensors and two controls. As explained in section 3, each sensor and each actuator, that is, each control, is to be monitored for a fault. It turns out that for all seven sensor faults and the two control input faults described in section 3, the detection spaces are given by the fault directions themselves. That is, for each fault,

$$\mathcal{T}_i^* = \operatorname{Im} F_i$$

Since the detection space associated with each of the sensor faults is two-dimensional, one fault detection filter can detect and identify at most three sensor faults. Therefore at least three fault detection filters are needed to identify all seven sensor faults. The detection spaces associated with the two actuator faults also are given by the fault directions themselves and are three-dimensional. As with the sensor faults, the detection spaces associated with the two actuator faults are given by the fault directions themselves and are three-dimensional. Together, there are seven two-dimensional sensor faults and two three-dimensional actuator faults. These should be combined in output separable and mutually detectable groups with seven or fewer directions. Many configurations are possible. The configuration chosen for this design involves a bank of four fault detection filters and requires some explanation.

In a fault detection system that consists of a bank of fault detection filters and a residual processor such as a neural network, fault isolation is done through the combined effort of both system elements. The fault detection filter is a carefully tuned device that uses known dynamic relationships to isolate a fault. The neural network residual processor combines the residuals from several filters and resolves any ambiguity. It is suggested that identifying a fault among a group of dynamically similar faults requires the precision of and is best delegated to the fault detection filters. Furthermore, it is suggested that the reliability of the neural network training should be improved if the fault groups associated with each of the fault detection filters are dynamically dissimilar. Given the above considerations, fault detection filters are designed for the following groups of faults:

Fault detection filter 1.

F_{y_m} : Manifold air mass sensor.

F_{y_ω} : Engine speed sensor.

$F_{y_{\ddot{x}}}$: Forward acceleration sensor.

Fault detection filter 2.

$F_{y_{\ddot{z}}}$: Heave acceleration sensor.

$F_{y_{rs}}$: Rear symmetric wheel speed sensor.

$F_{y_{fs}}$: Forward symmetric wheel speed sensor.

Fault detection filter 3.

F_{y_θ} : Pitch rate sensor.

$F_{y_{\ddot{z}}}$: Heave acceleration sensor.

$F_{y_{rs}}$: Rear symmetric wheel speed sensor.

Fault detection filter 4.

F_{u_α} : Throttle angle actuator.

F_{u_β} : Brake torque actuator.

To show that these fault sets are output separable, show that the columns of the CT_i^* matrices are independent where T_i^* is any basis for the detection space $\mathcal{T}_i^*$. Since for all faults it turns out that the detection spaces are just $\mathcal{T}_i^* = \operatorname{Im} F_i$, output separability requires that the columns of the CF_i matrices for each fault detection filter be independent. For example, for the first fault detection filter, the columns of the matrix

$$[CF_{y_m}, CF_{y_\omega}, CF_{y_{\ddot{x}}}]$$

are independent.

Showing that the fault sets are mutually detectable involves calculating invariant zeros of each triple $(C, A, F_1), \dots, (C, A, F_q)$ and then showing that these are the same invariant zeros as of the triple $(C, A, [F_1, \dots, F_q])$. For example, for the first fault detection filter, define the sets of invariant zeros

$$\begin{aligned} \Omega_{y_m} &= \Omega(C, A, F_{y_m}) \\ \Omega_{y_\omega} &= \Omega(C, A, F_{y_\omega}) \\ \Omega_{y_{\ddot{x}}} &= \Omega(C, A, F_{y_{\ddot{x}}}) \\ \Omega_y &= \Omega(C, A, [F_{y_m}, F_{y_\omega}, F_{y_{\ddot{x}}}]) \end{aligned}$$

where $\Omega(C, A, F_i)$ means the set of invariant zeros of the triple (C, A, F_i). The first fault detection filter is mutually detectable because

$$\Omega_y = \Omega_{y_m} + \Omega_{y_\omega} + \Omega_{y_{\ddot{x}}}$$

4.2 Eigenstructure Placement

The fault detection filters are found using a left eigenvector assignment algorithm described in appendix A. Since the calculations are somewhat long and they are the same for each detection filter, the calculation details are given for only one of the fault detection filters. Apply Algorithm A.1 to the design of a fault detection filter for the first fault group: the manifold air mass sensor, the engine speed sensor and the forward acceleration sensor.

The first step is to find the dimension of each detection space. This was discussed in section 4.1 where it was shown that the detection spaces are given by the fault directions themselves, that is, $\mathcal{T}_i^* = \operatorname{Im} F_i$. The fault directions assigned to the first fault detection filter are all sensor faults and all have dimension two

$$\begin{aligned}
\nu_{y_m} &= \dim \mathcal{T}_{y_m}^* = 2 \\
\nu_{y_\omega} &= \dim \mathcal{T}_{y_\omega}^* = 2 \\
\nu_{y_{\ddot{x}}} &= \dim \mathcal{T}_{y_{\ddot{x}}}^* = 2
\end{aligned}$$

The dimension of the fault detection filter complementary space $\mathcal{T}_0$ is also needed. The complementary space is any subspace independent of the detection spaces that completes the state-space. Thus, for the first fault detection filter

$$\mathcal{X} = \mathcal{T}_{y_m}^* \oplus \mathcal{T}_{y_\omega}^* \oplus \mathcal{T}_{y_{\ddot{x}}}^* \oplus \mathcal{T}_0$$

and the dimension of $\mathcal{T}_0$ is one

$$\begin{aligned}
\nu_0 &= n - \nu_{y_m} - \nu_{y_\omega} - \nu_{y_{\ddot{x}}} \\
&= 7 - 2 - 2 - 2 \\
&= 1
\end{aligned}$$

Next define the complementary faults sets. There are three faults F_{y_m}, F_{y_ω} and $F_{y_{\ddot{x}}}$ so there are four complementary fault sets which are:

$$\hat{F}_{y_m} = [F_{y_\omega}, F_{y_{\ddot{x}}}] \tag{4.1a}$$

$$\hat{F}_{y_\omega} = [F_{y_m}, F_{y_{\ddot{x}}}] \tag{4.1b}$$

$$\hat{F}_{y_{\ddot{x}}} = [F_{y_m}, F_{y_\omega}] \tag{4.1c}$$

$$\hat{F}_0 = [F_{y_m}, F_{y_\omega}, F_{y_{\ddot{x}}}] \tag{4.1d}$$

Now choose the fault detection filter closed-loop eigenvalues. Since the system model includes no sensor noise, no disturbances and no parameter variations, there is little basis for preferring one set of detection filter closed-loop eigenvalues over another. The poles are chosen here to give a reasonable response time but are not unrealistically fast. The assigned eigenvalues are

$$\begin{aligned} \Lambda_{y_m} &= \{-3, -6\} \\ \Lambda_{y_\omega} &= \{-4, -7\} \\ \Lambda_{y_{\ddot{x}}} &= \{-5, -8\} \\ \Lambda_0 &= \{-9\} \end{aligned}$$

The next step is to find the closed-loop fault detection filter left eigenvectors. For each eigenvalue $\lambda_{i_j} \in \Lambda_i$, the left eigenvectors v_{i_j} generally are not unique and must be chosen from a subspace as $v_{i_j} \in V_{i_j}$ where V_{i_j} and another space W_{i_j} are found by solving

$$\begin{bmatrix} A^T - \lambda_{i_j} I & C^T \\ \hat{F}_i^T & 0 \end{bmatrix} \begin{bmatrix} V_{i_j} \\ W_{i_j} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{4.2}$$

The V_{i_j} are as follows and show that all but one left eigenvector are chosen from a three-dimensional subspace.

$$V_0 = \begin{bmatrix} -0.7663 \\ 0.2813 \\ -0.1156 \\ -0.5620 \\ -0.0102 \\ -0.0451 \\ -0.0477 \end{bmatrix}$$

$$V_{y_{m_1}} = \begin{bmatrix} 0.3937 & -0.7842 & -0.1471 \\ 0.6822 & 0.3075 & 0.4324 \\ 0.5448 & 0.3266 & -0.2730 \\ 0.2407 & -0.0918 & -0.6586 \\ -0.1232 & 0.4136 & -0.5268 \\ 0.0978 & -0.0631 & 0.0460 \\ -0.0136 & -0.0198 & -0.0591 \end{bmatrix} \qquad V_{y_{m_2}} = \begin{bmatrix} 0.8277 & 0.0947 & -0.3123 \\ 0.0246 & 0.7596 & 0.4115 \\ -0.1720 & 0.6151 & -0.2647 \\ 0.0404 & 0.1735 & -0.6844 \\ -0.5226 & 0.0103 & -0.4365 \\ 0.0996 & 0.0715 & 0.0254 \\ 0.0026 & -0.0214 & -0.0601 \end{bmatrix}$$

$$V_{y_{\omega_1}} = \begin{bmatrix} 0.5353 & 0.7323 & -0.1351 \\ -0.4034 & -0.1608 & 0.1191 \\ -0.1345 & -0.0320 & -0.2896 \\ -0.5238 & 0.3196 & -0.7240 \\ 0.4917 & -0.5645 & -0.5954 \\ -0.1281 & 0.1248 & 0.0435 \\ -0.0095 & 0.0217 & -0.0560 \end{bmatrix} \qquad V_{y_{\omega_2}} = \begin{bmatrix} 0.6574 & 0.6199 & -0.1569 \\ -0.4388 & -0.0851 & 0.0546 \\ -0.0776 & -0.0843 & -0.2997 \\ -0.3144 & 0.1954 & -0.8739 \\ 0.5075 & -0.7364 & -0.3393 \\ -0.1135 & 0.1443 & -0.0133 \\ 0.0053 & 0.0089 & -0.0599 \end{bmatrix}$$

$$V_{y_{\ddot{x}_1}} = \begin{bmatrix} 0.8761 & -0.3783 & -0.0931 \\ -0.2425 & 0.1176 & 0.1382 \\ -0.0755 & -0.0399 & -0.2410 \\ -0.3401 & -0.5629 & -0.7047 \\ 0.2197 & 0.7022 & -0.6384 \\ -0.0608 & -0.1755 & 0.0656 \\ -0.0193 & -0.0266 & -0.0756 \end{bmatrix} \qquad V_{y_{\ddot{x}_2}} = \begin{bmatrix} 0.8669 & -0.4000 & -0.0884 \\ -0.2522 & 0.0978 & 0.1363 \\ -0.0491 & 0.0203 & -0.2501 \\ -0.2781 & -0.3670 & -0.8468 \\ 0.3156 & 0.8138 & -0.4326 \\ -0.0741 & -0.1817 & 0.0166 \\ -0.0116 & -0.0076 & -0.0812 \end{bmatrix}$$

As explained in (Douglas and Speyer 1995b), to help desensitize the fault detection filter to parameter variations, the left eigenvectors are chosen from $v_{i_j} \in V_{i_j}$ as the set with the greatest degree of linear independence. The degree of linear independence is indicated by the smallest singular value of the matrix formed by the left eigenvectors. Upper bounds on the singular values of the left eigenvectors are given by the singular values of $V = [V_0, V_{y_{m_1}}, V_{y_{m_2}}, V_{y_{\omega_1}}, V_{y_{\omega_2}}, V_{y_{\ddot{x}_1}}, V_{y_{\ddot{x}_2}}]$. These singular values are

$$\sigma(V) = \{2.6458, 2.4312, 2.0705, 1.3067, 0.3067, 0.0242, 0.0130\} \tag{4.3}$$

If the left eigenvector singular value upper bounds were small, then all possible combinations of detection filter left eigenvectors would be ill-conditioned and the filter eigenstructure would be sensitive to small parameter variations. Since (4.3) indicates that the upper bounds are not small, continue by looking for a set of fault detection filter left eigenvectors that are reasonably well-conditioned. For this case, the best-conditioned set of left eigenvectors from the set V nearly meets the upper bound and is given by

$$\tilde{V} \quad = \quad \left[v_0, v_{y_{m_1}}, v_{y_{m_2}}, v_{y_{\omega_1}}, v_{y_{\omega_2}}, v_{y_{\ddot{x}_1}}, v_{y_{\ddot{x}_2}}\right]$$

$$= \begin{bmatrix} -0.7663 & 0.2481 & -0.3779 & -0.3262 & -0.3840 & 0.4370 & -0.3755 \\ 0.2813 & 0.8149 & 0.0602 & 0.3474 & 0.0545 & -0.0314 & 0.1077 \\ -0.1156 & 0.4356 & 0.5242 & 0.1223 & 0.2732 & -0.2215 & 0.0543 \\ -0.5620 & -0.0202 & 0.4288 & 0.5909 & 0.4855 & -0.5556 & 0.5520 \\ -0.0102 & -0.2693 & 0.6254 & -0.6199 & 0.7284 & -0.6627 & -0.7139 \\ -0.0451 & 0.1022 & -0.0565 & 0.1560 & -0.0866 & 0.0847 & 0.1720 \\ -0.0477 & -0.0355 & 0.0233 & 0.0149 & 0.0346 & -0.0629 & 0.0236 \end{bmatrix}$$

where once again

$$\begin{array}{rclcrcl} v_0 & = & V_0 & & & & \\ v_{y_{m_1}} & \in & V_{y_{m_1}} & & v_{y_{m_2}} & \in & V_{y_{m_2}} \\ v_{y_{\omega_1}} & \in & V_{y_{\omega_1}} & & v_{y_{\omega_2}} & \in & V_{y_{\omega_2}} \\ v_{y_{\ddot{x}_1}} & \in & V_{y_{\ddot{x}_1}} & & v_{y_{\ddot{x}_2}} & \in & V_{y_{\ddot{x}_2}} \end{array}$$

The singular values of this set of detection filter left eigenvectors are

$$\sigma(\tilde{V}) = \{1.7191, 1.4641, 1.0000, 0.9240, 0.2168, 0.0171, 0.0092\}$$

Since the difference between the largest and the smallest singular values is only two orders of magnitude, the detection filter gain will be small and the filter eigenstructure should not be sensitive to small parameter variations.

The fault detection filter gain L is found by solving

$$\tilde{V}^T L = \tilde{W}^T \tag{4.4}$$

where $\tilde{V}$ is the matrix of left eigenvectors as found above, and $\tilde{W}$ is a matrix of vectors w_{i_j} which satisfy (a.13)

$$\begin{bmatrix} A^T - \lambda_{i_j} I & C^T \\ \hat{F}_i^T & 0 \end{bmatrix} \begin{bmatrix} v_{i_j} \\ w_{i_j} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

If the left eigenvector v_{i_j} is a linear combination of the columns of V_{i_j}, w_{i_j} is the same linear combination of the columns of W_{i_j} where V_{i_j} and W_{i_j} are from (4.2). The $\tilde{W}$ for the first fault detection filter is

$$\begin{aligned} \tilde{W} &= \left[w_0, w_{y_{m_1}}, w_{y_{m_2}}, w_{y_{\omega_1}}, w_{y_{\omega_2}}, w_{y_{\ddot{x}_1}}, w_{y_{\ddot{x}_2}} \right] \\ &= \begin{bmatrix} 0.0000 & 17.8806 & 23.0005 & 0.0000 & -0.0000 & -0.0000 & -0.0000 \\ -0.0000 & -0.0000 & 0.0000 & -0.7504 & 2.2110 & 0.0000 & 0.0000 \\ 0.0000 & -0.0000 & -0.0000 & 0.0000 & -0.0000 & -20.2591 & -14.0152 \\ -17.7474 & 6.4188 & -7.7985 & -6.7710 & -10.2386 & 18.7932 & -5.7161 \\ 220.5377 & 106.4884 & -115.0230 & -0.8805 & -174.3199 & 323.5220 & -35.9330 \\ 12.8111 & -6.7819 & 10.1691 & -9.6822 & 12.4763 & -11.6560 & -7.8435 \\ 5.4270 & 3.2914 & -6.7082 & 17.4639 & -11.7092 & 2.4288 & 13.3306 \end{bmatrix} \end{aligned}$$

The detection filter gain is found from (4.4) and is given by

$$L = \tag{4.5}$$

$$\begin{bmatrix} -0.9444 & -1.5605 & 18.3898 & 20.3959 & -33.4718 & -14.9397 & -0.1545 \\ -16.0816 & -0.9793 & 46.3621 & -15.7918 & -97.3527 & -1.4476 & 14.1137 \\ 97.3771 & -16.7452 & 78.5443 & -8.8050 & 28.0068 & -12.8347 & 47.5303 \\ -11.0019 & -4.0789 & 39.0600 & 3.2445 & 213.9492 & -20.9184 & 27.7123 \\ -61.4309 & 31.0335 & -204.1174 & 31.4592 & -213.9106 & 45.9121 & -126.8465 \\ -251.1954 & 149.9014 & -1127.4460 & 176.9975 & -936.7994 & 175.4977 & -575.9418 \\ 64.6381 & -40.4210 & 437.0497 & -239.8124 & -6321.3974 & 64.8115 & 101.8300 \end{bmatrix}$$

To complete the detection filter design, output projection matrices $\hat{H}_{y_m}$, $\hat{H}_{y_\omega}$ and $\hat{H}_{y_{\ddot{x}}}$ are needed to project the residual along the respective output subspaces $C\hat{\mathcal{T}}^*_{y_m}$, $C\hat{\mathcal{T}}^*_{y_\omega}$ and $C\hat{\mathcal{T}}^*_{y_{\ddot{x}}}$. What this means is that, for example, $\hat{\mathcal{T}}^*_{y_m}$ becomes the unobservable subspace of the pair $(\hat{H}_{y_m}C, A + LC)$. Remember that by the definition of the complementary faults (4.1), faults F_{y_ω} and $F_{y_{\ddot{x}}}$ lie in $\hat{\mathcal{T}}^*_{y_m}$ and fault F_{y_m} does not. The effect is that the projected residual is driven by fault F_{y_m} and only fault F_{y_m}.

A projection $\hat{H}_i$ is computed by first finding a basis for the range space of $C\hat{T}^*_i$ where again, $\hat{T}^*_i$ is any basis for the detection space $\hat{\mathcal{T}}^*_i$. This is done by finding the left singular vectors of $C\hat{T}^*_i$. Denote this basis for now as h_i. Then $\hat{H}_i$ is given by

$$\hat{H}_i = I - h_i h_i^T$$

Output projections for the first fault detection filter are

$$\hat{H}_{y_m} = \begin{bmatrix} 0.9986 & 0.0000 & 0.0000 & 0.0099 & -0.0009 & 0.0170 & -0.0322 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0099 & 0.0000 & 0.0000 & 0.6362 & 0.0065 & -0.4776 & -0.0571 \\ -0.0009 & 0.0000 & 0.0000 & 0.0065 & 0.9995 & 0.0108 & -0.0189 \\ 0.0170 & 0.0000 & 0.0000 & -0.4776 & 0.0108 & 0.3594 & 0.0418 \\ -0.0322 & 0.0000 & 0.0000 & -0.0571 & -0.0189 & 0.0418 & 0.0064 \end{bmatrix} \tag{4.6a}$$

$$\hat{H}_{y_\omega} = \begin{bmatrix} 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0087 & 0.0000 & 0.0295 & -0.0004 & 0.0384 & -0.0792 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0295 & 0.0000 & 0.7062 & 0.0005 & -0.3580 & -0.2801 \\ 0.0000 & -0.0004 & 0.0000 & 0.0005 & 1.0000 & 0.0006 & 0.0004 \\ 0.0000 & 0.0384 & 0.0000 & -0.3580 & 0.0006 & 0.5637 & -0.3410 \\ 0.0000 & -0.0792 & 0.0000 & -0.2801 & 0.0004 & -0.3410 & 0.7214 \end{bmatrix} \tag{4.6b}$$

$$\hat{H}_{y_{\ddot{x}}} = \begin{bmatrix} 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.5276 & 0.1876 & -0.0015 & 0.2370 & -0.3973 \\ 0.0000 & 0.0000 & 0.1876 & 0.6790 & 0.0025 & -0.4029 & -0.1429 \\ 0.0000 & 0.0000 & -0.0015 & 0.0025 & 1.0000 & 0.0032 & 0.0010 \\ 0.0000 & 0.0000 & 0.2370 & -0.4029 & 0.0032 & 0.4943 & -0.1772 \\ 0.0000 & 0.0000 & -0.3973 & -0.1429 & 0.0010 & -0.1772 & 0.2992 \end{bmatrix} \tag{4.6c}$$

In summary, a fault detection filter for the system with sensor faults E_{y_m}, E_{y_ω} and $E_{y_{\ddot{x}}}$ as in (3.3)

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du + E_{y_m}\mu_{y_m} + E_{y_\omega}\mu_{y_\omega} + E_{y_{\ddot{x}}}\mu_{y_{\ddot{x}}} \end{aligned}$$

is equivalent to a fault detection filter for the system with faults F_{y_m}, F_{y_ω} and $F_{y_{\ddot{x}}}$ as in (3.4)

$$\begin{aligned} \dot{x} &= Ax + Bu + F_{y_m}\mu_{y_m} + F_{y_\omega}\mu_{y_\omega} + F_{y_{\ddot{x}}}\mu_{y_{\ddot{x}}} \\ y &= Cx + Du \end{aligned}$$

and has the form

$$\begin{aligned} \dot{\hat{x}} &= (A + LC)\hat{x} + Bu - Ly \\ z_{y_m} &= \hat{H}_{y_m}(C\hat{x} + Du - y) \\ z_{y_\omega} &= \hat{H}_{y_\omega}(C\hat{x} + Du - y) \\ z_{y_{\ddot{x}}} &= \hat{H}_{y_{\ddot{x}}}(C\hat{x} + Du - y) \end{aligned}$$

with L and the $\hat{H}_{y_m}$, $\hat{H}_{y_\omega}$ and $\hat{H}_{y_{\ddot{x}}}$ given by (4.5) and (4.6). Calculations for the detection filters for the other three fault groups are carried out in the same way and are not shown here.

4.3 Reduced-Order Observers

An essential feature of a fault detection filter is that for each pair $(\hat{H}_iC, A + LC)$, there is an unobservable subspace $\hat{\mathcal{T}}_i^*$ that contains the complementary fault $\hat{F}_i$. This means that any fault in $\hat{F}_i$ does not affect the residual when when the residual is operated upon by the projector $(\hat{H}_i)$. Consider a detection filter

$$\begin{aligned} \dot{\hat{x}} &= (A + LC)\hat{x} + Bu - Ly \\ z_i &= \hat{H}_i(C\hat{x} - y) \end{aligned}$$

If the state estimation error is $e = \hat{x} - x$, then the error dynamics are

$$\dot{e} = (A + LC)e - \sum_{i=1}^{q} F_i m_i \tag{4.7a}$$

$$z_i = \hat{H}_i C e \tag{4.7b}$$

Now $\hat{\mathcal{T}}_i^*$ is $(\hat{H}_i C, A{+}LC)$ unobservable means that $\hat{\mathcal{T}}_i^*$ is $(A{+}LC)$-invariant and $\hat{H}_i C \hat{\mathcal{T}}_i^* = 0$. Therefore, define the factor space $\bar{\mathcal{X}}_i = \mathcal{X}/\hat{\mathcal{T}}_i^*$ and let $\bar{P}_i$ be the canonical projection $\bar{P}_i : \mathcal{X} \mapsto \bar{\mathcal{X}}_i$. Compute $\bar{P}_i$ using singular value decomposition to find the null space of $(\hat{T}_i^*)^T$

$$\bar{P}_i \hat{T}_i^* = 0 \tag{4.8}$$

Now let $\bar{A}_i$ and $\bar{C}_i$ be maps induced on the factor space $\bar{\mathcal{X}}_i$ by $(A + LC)$ and $\hat{H}_i C$. These are given by

$$\begin{aligned} \bar{P}_i(A + LC) &= \bar{A}_i \bar{P}_i \\ \hat{H}_i C &= \bar{C}_i \bar{P}_i \end{aligned}$$

Also, since $\hat{\mathcal{T}}_i^*$ is the unobservable subspace of $(\hat{H}_i C, A + LC)$, it follows that

$$\begin{aligned} \bar{P}_i F_{j \neq i} &= 0 \\ \bar{P}_i F_i &\neq 0 \end{aligned}$$

The reduced-order fault detection filter error dynamics equivalent to (4.7) are given by

$$\dot{\bar{e}}_i = \bar{A}_i \bar{e}_i - \bar{P}_i F_i m_i \tag{4.9a}$$

$$z_i = \bar{C}_i \bar{e}_i \tag{4.9b}$$

where $\bar{e}_i \in \bar{\mathcal{X}}_i$.

Besides the lower-order, an important property of the reduced-order fault detection filter (4.9) is that there is no special structure imposed on the dynamics. No $\bar{A}_i$-invariant subspaces need to be defined. Therefore, the reduced-order filter dynamics are arbitrarily modified with a gain $\bar{L}_i$ as

$$\dot{\bar{e}}_i = (\bar{A}_i + \bar{L}_i\bar{C}_i)\bar{e}_i - \bar{P}_i F_i m_i \tag{4.10a}$$

$$z_i = \bar{C}_i\bar{e}_i \tag{4.10b}$$

This gain may be chosen to satisfy any secondary filter criterion such as to minimize disturbances or sensor noise or the effect of parameter uncertainty. A reduced-order fault detection filter with the error dynamics (4.10) is given by

$$\dot{\bar{x}}_i = (\bar{A}_i + \bar{L}_i\bar{C}_i)\bar{x}_i + \bar{P}_i B u - (\bar{P}_i L + \bar{L}_i) y \tag{4.11a}$$

$$z_i = \hat{H}_i(\bar{C}_i\bar{x}_i - y) \tag{4.11b}$$

The reduced-order fault detection filter (4.11) detects and identifies only the one fault F_i out of the group of q design faults $\{F_1, \dots, F_i, \dots, F_q\}$. Therefore, a family of q reduced-order filters are needed to detect and identify all faults in the design set. In general, the order of all q reduced-order filters is higher than the order of the full-state fault detection filter.

The fault detection filter designed in section 4.2 has three faults F_{y_m}, F_{y_ω} and $F_{y_{\ddot{x}}}$. The complementary faults $\hat{F}_{y_m}$, $\hat{F}_{y_\omega}$ and $\hat{F}_{y_{\ddot{x}}}$ each produce a fourth-order detection space. Therefore, each associated factor space will be third-order. Canonical projections for each factor space are computed using singular value decomposition to solve (4.8). They are given by

$$\bar{P}_{y_m} = \begin{bmatrix} -0.1426 & -0.8445 & -0.3770 & 0.1389 & 0.3061 & -0.0968 & 0.0442 \\ 0.7495 & -0.1743 & -0.3539 & -0.0444 & -0.5243 & 0.0760 & 0.0057 \\ 0.4577 & -0.0567 & 0.4589 & 0.6920 & 0.3086 & 0.0229 & 0.0457 \end{bmatrix}$$

$$\bar{P}_{y_\omega} = \begin{bmatrix} -0.3373 & -0.0641 & 0.1332 & -0.0431 & 0.9144 & -0.1624 & 0.0090 \\ -0.8472 & 0.4218 & -0.0201 & -0.1513 & -0.2805 & 0.0356 & -0.0312 \\ -0.0975 & 0.1439 & 0.2912 & 0.9359 & -0.0107 & 0.0789 & 0.0514 \end{bmatrix}$$

$$\bar{P}_{y_{\ddot{x}}} = \begin{bmatrix} -0.1295 & -0.0390 & 0.1473 & 0.1254 & 0.9561 & -0.1697 & 0.0371 \\ -0.4027 & 0.0494 & 0.1933 & 0.8670 & -0.1814 & 0.0962 & 0.0643 \\ 0.8605 & -0.2963 & 0.0794 & 0.4021 & 0.0430 & 0.0276 & 0.0358 \end{bmatrix}$$

The reduced-order fault detection filters are as follows. Note that each is third-order and the combined system is ninth-order. This is two more than the full-order fault detection filter of section 4.2.

Reduced-order fault detection filter to detect F_{y_m} from the set of faults F_{y_m}, F_{y_ω} and $F_{y_{\ddot{x}}}$.

$$\begin{aligned}
\dot{\bar{x}} &= \begin{bmatrix} -7.0685 & -0.4295 & -0.2956 \\ -0.4295 & -5.8586 & -2.7384 \\ -0.2956 & -2.7384 & -8.0729 \end{bmatrix} \bar{x} + \begin{bmatrix} -3.8172 & 0.0187 \\ -0.7269 & -0.0248 \\ 1.2773 & 0.0170 \end{bmatrix} u + \\
& \begin{bmatrix} 16.1573 & 0 & 0 & 3.9056 & 148.1616 & -5.2199 & 4.5504 \\ 18.3944 & 0 & 0 & -16.5934 & 22.8931 & 15.8321 & -2.6829 \\ -15.7902 & 0 & 0 & -11.2426 & 225.2354 & 5.9689 & 7.5751 \end{bmatrix} y \\
z_{y_m} &= \begin{bmatrix} -0.5029 & -0.2457 & 0.2491 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ -0.0236 & -0.2090 & -0.1581 \\ 0.0098 & -0.0169 & 0.0233 \\ 0.0056 & 0.1507 & 0.1252 \\ 0.0177 & 0.0266 & 0.0058 \end{bmatrix} \bar{x} + \begin{bmatrix} -0.0000 & 0.0000 \\ 0.0000 & -0.0000 \\ 0.0000 & -0.0000 \\ -0.0001 & 0.0005 \\ -0.0000 & 0.0000 \\ 0.0000 & -0.0004 \\ 0.0000 & -0.0000 \end{bmatrix} u + \\
& \begin{bmatrix} -0.9986 & 0 & 0 & -0.0099 & 0.0009 & -0.0170 & 0.0322 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.0099 & 0 & 0 & -0.6362 & -0.0065 & 0.4776 & 0.0571 \\ 0.0009 & 0 & 0 & -0.0065 & -0.9995 & -0.0108 & 0.0189 \\ -0.0170 & 0 & 0 & 0.4776 & -0.0108 & -0.3594 & -0.0418 \\ 0.0322 & 0 & 0 & 0.0571 & 0.0189 & -0.0418 & -0.0064 \end{bmatrix} y
\end{aligned}$$

Reduced-order fault detection filter to detect F_{y_ω} from the set of faults F_{y_m}, F_{y_ω} and $F_{y_{\ddot{x}}}$.

$$\begin{aligned}
\dot{\bar{x}} &= \begin{bmatrix} -6.0954 & -0.2736 & 1.6189 \\ -0.2736 & -9.4614 & 0.8762 \\ 1.6189 & 0.8762 & -7.4432 \end{bmatrix} \bar{x} + \begin{bmatrix} -0.2832 & 0.0317 \\ 0.4874 & -0.0034 \\ 0.8006 & 0.0179 \end{bmatrix} u + \\
& \begin{bmatrix} 0 & -2.2024 & 0 & 9.3196 & 88.3783 & -18.3873 & 17.2499 \\ 0 & 0.2481 & 0 & 19.3040 & -178.1764 & -6.8141 & -12.8500 \\ 0 & -0.7344 & 0 & 2.4798 & 198.6644 & 5.3856 & -2.9772 \end{bmatrix} y \\
z_{y_\omega} &= \begin{bmatrix} 0 & 0 & 0 \\ -0.0029 & 0.0098 & 0.0349 \\ 0 & 0 & 0 \\ 0.1133 & 0.2433 & 0.0309 \\ 0.0198 & -0.0092 & 0.0109 \\ -0.1122 & -0.1260 & 0.2248 \\ 0.0243 & -0.0932 & -0.3161 \end{bmatrix} \bar{x} + \begin{bmatrix} 0 & 0 \\ -0.0001 & 0.0001 \\ 0 & 0 \\ -0.0004 & 0.0006 \\ 0.0000 & -0.0000 \\ -0.0005 & -0.0001 \\ 0.0010 & -0.0005 \end{bmatrix} u +
\end{aligned}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.0087 & 0 & -0.0295 & 0.0004 & -0.0384 & 0.0792 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.0295 & 0 & -0.7062 & -0.0005 & 0.3580 & 0.2801 \\ 0 & 0.0004 & 0 & -0.0005 & -1.0000 & -0.0006 & -0.0004 \\ 0 & -0.0384 & 0 & 0.3580 & -0.0006 & -0.5637 & 0.3410 \\ 0 & 0.0792 & 0 & 0.2801 & -0.0004 & 0.3410 & -0.7214 \end{bmatrix} y$$

Reduced-order fault detection filter to detect $F_{y_{\ddot{x}}}$ from the set of faults F_{y_m}, F_{y_ω} and $F_{y_{\ddot{x}}}$.

$$\begin{aligned} \dot{\bar{x}} &= \begin{bmatrix} -7.9585 & -1.4027 & 0.2690 \\ -1.4027 & -7.1497 & -0.5553 \\ 0.2690 & -0.5553 & -9.8918 \end{bmatrix} \bar{x} + \begin{bmatrix} -0.0145 & -0.0268 \\ 0.0116 & 0.0148 \\ 0.0103 & 0.0004 \end{bmatrix} u + \\ & \begin{bmatrix} 0 & 0 & -24.6267 & 11.7615 & 240.9060 & -13.9996 & 9.8338 \\ 0 & 0 & -0.5869 & 11.9707 & 257.8946 & 1.9564 & -8.1247 \\ 0 & 0 & 0.1929 & -20.4744 & 173.3469 & 12.7158 & 7.0952 \end{bmatrix} y \\ z_{y_{\ddot{x}}} &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -0.0063 & 0.2182 & 0.0244 \\ 0.0586 & 0.0431 & -0.2381 \\ 0.0216 & 0.0059 & 0.0080 \\ -0.0512 & 0.1255 & 0.2075 \\ 0.0046 & -0.1642 & -0.0178 \end{bmatrix} \bar{x} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -0.0000 & -0.0009 \\ 0.0000 & 0.0001 \\ 0.0000 & -0.0000 \\ -0.0000 & -0.0007 \\ 0.0000 & 0.0006 \end{bmatrix} u + \\ & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.5276 & -0.1876 & 0.0015 & -0.2370 & 0.3973 \\ 0 & 0 & -0.1876 & -0.6790 & -0.0025 & 0.4029 & 0.1429 \\ 0 & 0 & 0.0015 & -0.0025 & -1.0000 & -0.0032 & -0.0010 \\ 0 & 0 & -0.2370 & 0.4029 & -0.0032 & -0.4943 & 0.1772 \\ 0 & 0 & 0.3973 & 0.1429 & -0.0010 & 0.1772 & -0.2992 \end{bmatrix} y \end{aligned}$$

Chapter 5
Fault Detection Filter Evaluation

Fault detection filter performance is evaluated using the nonlinear vehicle simulation discussed in section 2. Sensor fault detection and identification performance is evaluated by introducing a sensor bias into the data provided by the nonlinear simulation to the fault detection filters. In the most benign test, the nonlinear vehicle simulation is run in steady state at 25 meters per second forward speed with no turns while a bias is added to one of the sensor outputs. In this test, the operating point is the same as that used to derive the linearized dynamics for the fault detection filter design. Furthermore, the vehicle dynamics are not stimulated resulting in data that is essentially linear. Thus, the fault detection filter is operating in a nominal environment and the test does not provide much useful information.

In a more useful test, the filters operate at an off-nominal condition, that is, the vehicle operates in a steady state condition but not the same one used to generate the linearized dynamics. This is achieved by increasing the throttle two degrees from the nominal value causing the steady state vehicle speed to be about two meters per second faster than the nominal. If the vehicle dynamics were linear, the increased throttle setting would have only a transient effect, if any, on the linear fault detection filter state estimates. The state estimate errors and the filter residuals would asymptotically go to zero. Since the vehicle dynamics are not linear and the vehicle operating condition is not the same as it would be if the dynamics were linear, the filter state estimates and the residuals are not zero. The top-left plot of figures 5.1-5.4 shows the magnitude of each reduced-order fault detection filter residual as it responds to the off-nominal steady state condition.

Since most residuals are not zero, as is to be expected, the natural question to ask is what magnitude residual should be considered small. The answer lies in comparing the size of a nonzero residual due to non-linearities and the size of a nonzero residual due to a fault. A residual scaling factor is chosen such that when a fault is introduced into the *linearized* dynamics the magnitude of the corresponding reduced-order fault detection filter residual is one. Since most residuals generated by the off-nominal operating condition have magnitude less than about 0.2, they should not be easily mistaken for residuals generated by a fault.

Of course, the size of the residual is proportional to the size of the fault. The size of the fault used for finding the residual scaling factors is determined as follows. For most sensors, the size of the fault is given by the difference in magnitude between the sensor output at the nominal and off-nominal steady state operating conditions. For some sensors, such as the accelerometers and the pitch rate sensor, the output is zero in any steady state condition and another method has to be used. For the longitudinal accelerometer, the size of the fault is given by the largest transient value of the sensor output as the two-degree step throttle command takes the vehicle from the nominal to the off-nominal condition. For the heave accelerometer, even the transient is small during an acceleration maneuver so a another value is chosen that represents a vehicle heave acceleration that is reasonably encountered during normal vehicle operation. The pitch rate sensor is treated the same way as the heave accelerometer.

Figure 5.1 shows the magnitudes of the residuals for the three reduced-order fault

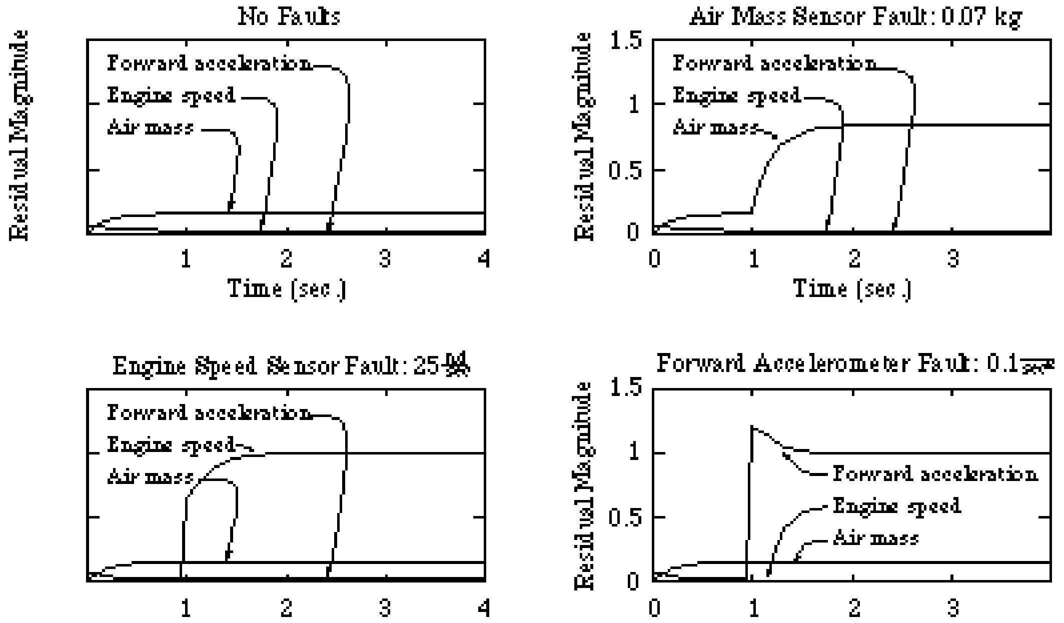

Figure 5.1: Residuals for Fault Detection Filter One: Manifold Air Mass Sensor, Engine Speed Sensor and Forward Acceleration Sensor.

detection filters derived from the first fault design group: an air mass sensor fault, an engine speed sensor fault and a longitudinal accelerometer fault. A sensor bias fault is added after one second when filter initialization errors have died out. Only one sensor fault is added at a time; simultaneous faults are not allowed. It is important to note that when any of the sensor faults from the first fault design group occur, the residuals associated with a fault detection filter designed for other faults have no meaning. This is why only three residuals are shown in each plot of figure 5.1 while eleven residuals are generated by the entire fault detection system. Distinguishing a meaningful residual from a non-meaningful residual is left to the residual processing neural network described in the next section. All reduced-order fault detection filter residuals respond closely to their linear counterparts. The residual associated with the fault quickly approaches one and other residuals *in the fault group* remain unaffected.

Figures 5.2 and 5.3 show the residuals for the three reduced-order fault detection filters

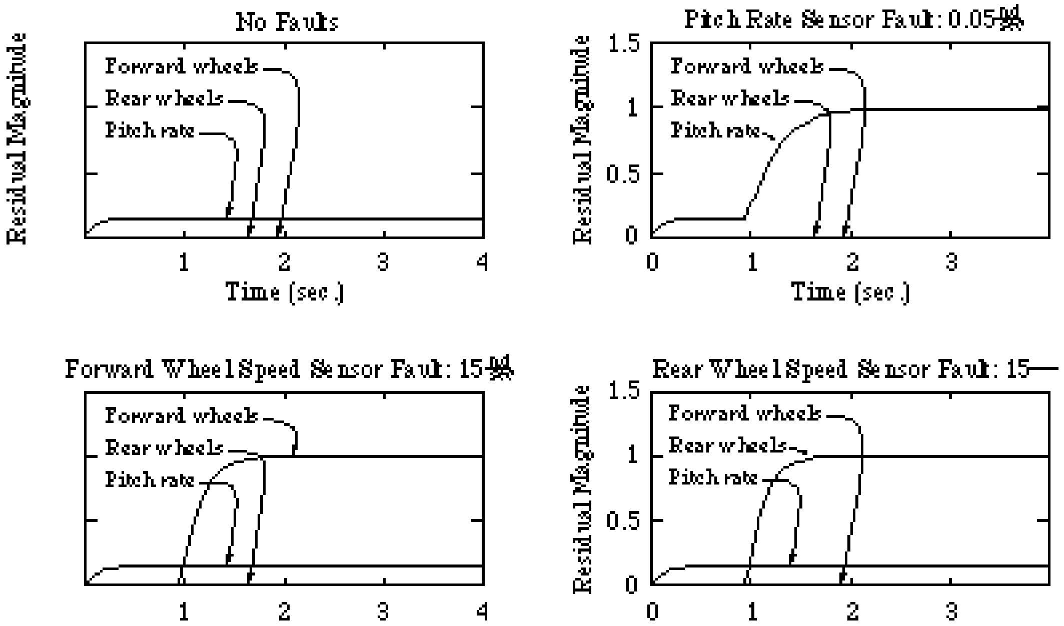

Figure 5.2: Residuals for Fault Detection Filter Two: Pitch Rate Sensor, Forward Wheel Speed Sensor and Rear Wheel Speed Sensor.

derived from the second and third sensor fault design groups. Residual scaling factors are chosen in the same way as for the first fault design group. The reduced-order fault detection filter performance indicated by figures 5.2 and 5.3 is the same as that indicated by figure 5.1.

Actuator fault identification performance is shown in figure 5.4. A throttle fault is simulated by sending a two-degree step throttle command to the nonlinear simulation but not to the fault detection filter. Even though a throttle fault stimulates the vehicle nonlinear dynamics and the residual associated with the brake fault, figure 5.4 shows that both positive and negative throttle faults are clearly identifiable from a brake fault. A brake fault is simulated by applying a brake torque just large enough to slow the vehicle from 25 m/s to 21 m/s. This changes the vehicle steady state operating point by the same amount as a four-degree throttle fault. Figure 5.4 shows that the brake fault is clearly identified.

It is interesting to note how well the throttle and brake faults are identified. Intuitively,

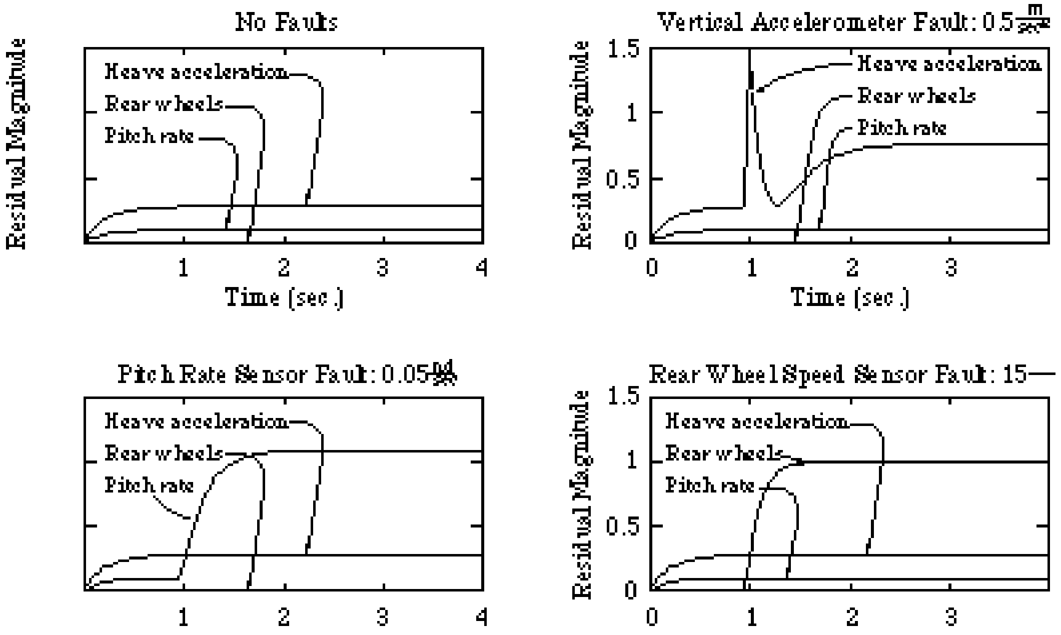

Figure 5.3: Residuals for Fault Detection Filter Three: Heave Acceleration Sensor, Pitch Rate Sensor and Rear Wheel Speed Sensor.

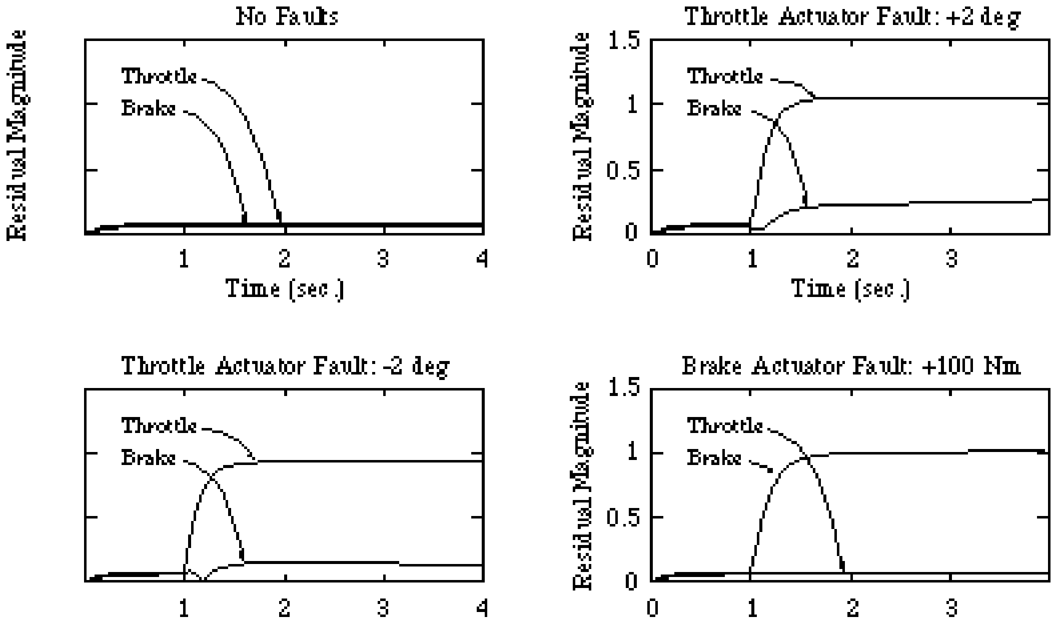

Figure 5.4: Residuals for Fault Detection Filter Four: Throttle Actuator, Brake Actuator.

there should not be much difference in the vehicle behavior when the brake is applied or when the throttle is decreased. Yet, the fault detection filter, using the vehicle dynamics, finds there is enough difference between them to clearly distinguish one from the other.

Chapter 6
Residual Processing

The essential feature of a residual processor is to analyze the residues generated by all fault detection filters and announce whether or not a fault has occurred. Nominally, the residual process is zero in the absence of a fault and non-zero otherwise. However, when driven by nonlinearities, the residual process can fail to go to zero even in the absence of faults. This is noted in the simulation studies of section 5. Furthermore, the residual may be nonzero when a fault occurs for which the fault detection filter is not designed. In this case, the residual directional properties are not defined; the fault detection filter detects but cannot isolate the fault. These are examples of ambiguities that are to be resolved by the residual processor.

The approach taken here is to consider that the residuals from all fault detection filters constitute a pattern, a pattern that contains information about the presence or absence of a fault. Hence, residual processing is treated as a static pattern recognition problem. This class of problems is ideally suited for application to a neural network.

Multilayer feedforward networks, also referred to as multilayer perceptrons, are an important class of neural networks that have proved extremely successful in pattern recognition problems. Typically, the network consists of a set of source nodes or neurons that constitute the input layer, one or more hidden layers of computation nodes, and an output layer of computation nodes. The input signal propagates through the network layer-by-layer. Each node in the network has a set of synapses, each of which is characterized by a weight matrix and a nonlinear differentiable activation function. The activation function limits the amplitude of the neuron output.

The neural networks described in this section each are applied to the residuals from only one fault detection filter. That is, in its present form, when a fault occurs, the neural network does not attempt to determine which fault detection filters are responding to a design fault and which are responding to some unknown input. Thus, the present system has four residual processors. Each processor is a multilayer perceptron with the following characteristics:

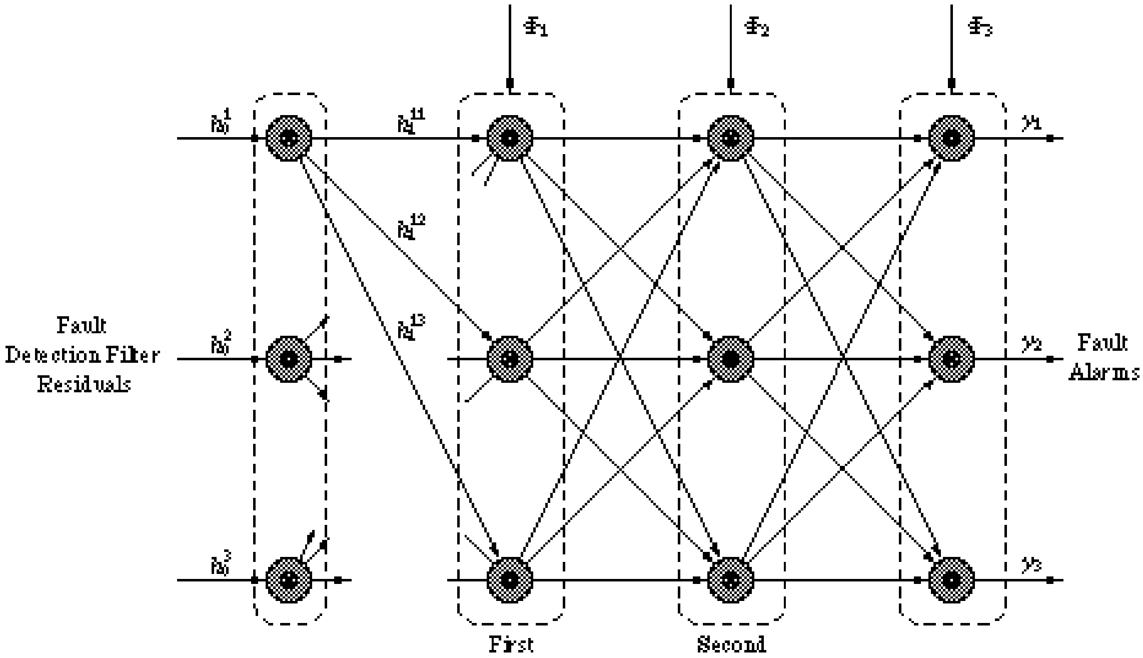

Figure 6.1: Multi-Layer Perceptron Model.

1. Each network considers the residual from only one filter.

2. Each network has one input layer, two hidden layers, and one output layer.

3. For the first three fault groups, the sensor fault groups, the input vector is $u \in \mathrm{R}^3$ and the output vector is $y \in \mathrm{R}^3$. For the fourth fault group, the actuator fault group, $u \in \mathrm{R}^2$ and $y \in \mathrm{R}^2$.

4. The activation function is a sigmoidal function. This has a smooth nonlinearity which is required later for a gradient vector calculation.

$$S(x) = 10\frac{e^x - 1}{e^x + 1} \tag{6.1}$$

The training process involves the determination of the synaptic weights and the bias vectors of the network. The backpropagation algorithm is the most widely used supervised learning algorithm in neural network applications. It consists of two passes through the different layers of the network: a forward pass and a backward pass. In the forward pass, an input vector is applied to the input layer and its effect propagates through the network to produce an output. In the present application, the fault detection filter residuals are applied to the input layer. During this pass, the synaptic weights of the network are held constant. In the backward pass, the synaptic weights are adjusted in accordance with an error-correction rule. Here, the network output is compared with a desired network output and an error vector is formed. As the error vector propagates backward through the network, the synaptic weights are adjusted to minimize the error. In the present application, the presence or absence of a fault announcement forms the desired output. Together, the detection filter residuals and fault announcements form a neural network training set.

Since the learning rate for a conventional backpropagation algorithm can become excessively slow, the learning phase of the neural network is viewed as a case of a nonlinear unconstrained parameter optimization problem. The cost function is defined as the error between the actual and the desired output over the entire training set. This is also known as batch mode training, wherein weight updating is done after the presentation of the entire set of training examples to the network. Each training example is also called a pattern and one complete presentation of all training sets is called an epoch. The learning process continues epoch-by-epoch until the synaptic weights and the bias vectors of the network stabilize and the cost converges to some minimum value. The order of presentation of the training examples within an epoch is randomized from one epoch to the next to make the search in weight space stochastic over the learning cycles. This avoids the possibility of limit cycles in the evolution of the synaptic weights.

The back propagation training algorithm requires that a gradient of the cost function with respect to the synaptic weights and bias vectors be calculated. The required partial derivatives may be calculated analytically as follows. Begin with an expression for the cost function, the average error over one epoch given by

$$\mathcal{E} = \sum_{j=1}^{j=N} \frac{e^j}{N}$$

where $\mathcal{E}$ is the average error over one epoch, e^j is the error for the j^{th} pattern or training set in the epoch and N is the number of patterns per epoch. The error e^j is given by

$$e^j = \frac{1}{2}(d^j - y^j)^T(d^j - y^j)$$

where d^j is the desired network output and y^j is the actual output Note that d^j and $y^j \in \mathrm{R}^3$ or R^2, depending on the network.

For the network shown in figure 6.1, the network output is given by

$$y^j = S(h_3^j)$$

where h_3^j is the input vector to the third layer for the j^{th} training pattern and $S(\cdot)$ is the activation function embedded in each node. The input vectors to the third layer and other layers are given by

$$h_i^j = W_i^j \cdot S(h_{i-1}^j) + \Phi_i^j \tag{6.2}$$

where h_i^j, W_i^j and Φ_i^j are the input vector, weighting matrix and associated bias vector for layer i and training pattern j. Note that h_i^j, Φ_i^j and $S(h_i^j) \in \mathrm{R}^3$ and W_i^j is a 3×3 matrix.

Again, $S(\cdot)$ is the activation function given by (6.1). From (6.2), the network output is given by information from the second layer as

$$y^j = S(W_3^j \cdot S(h_2^j) + \Phi_3^j)$$

Thus, the error function gradient is calculated backwards as

$$\begin{aligned} \frac{\partial y^j}{\partial h_i^j} &= \frac{\partial y^j}{\partial h_{i+1}^j} \cdot W_{i+1}^j \cdot \frac{\partial S(h_i^j)}{\partial h_i^j} \\ \frac{\partial e^j}{\partial h_i^j} &= \frac{\partial e^j}{\partial h_{i+1}^j} \cdot W_{i+1}^j \cdot \frac{\partial S(h_i^j)}{\partial h_i^j} \end{aligned}$$

and with the boundary condition

$$\frac{\partial e^j}{\partial h_3^j} = (y^j - d^j)^T \cdot \frac{\partial S(h_3^j)}{\partial h_3^j}$$

Hence, the error function gradient for the j^{th} pattern is

$$\begin{aligned} \frac{\partial e^j}{\partial W_i^j} &= \frac{\partial e^j}{\partial h_i^j} \cdot S(h_{i-1}) \\ \frac{\partial e^j}{\partial \Phi_i^j} &= \frac{\partial e^j}{\partial h_i^j} \end{aligned}$$

Finally the gradient vector for the cost is

$$\begin{aligned} \frac{\partial \mathcal{E}}{\partial W_i} &= \frac{\sum_{j=1}^{j=N} \frac{\partial e^j}{\partial W_i^j}}{N} \\ \frac{\partial \mathcal{E}}{\partial \Phi_i} &= \frac{\sum_{j=1}^{j=N} \frac{\partial e^j}{\partial \Phi_i^j}}{N} \end{aligned}$$

A Davidon-Fletcher-Powell algorithm is used to solve the unconstrained parameter optimization problem. The algorithm converges in exactly n steps for a quadratic cost and uses a rank two update for the Hessian matrix to ensure that at the end of every iteration, the Hessian is positive definite. The algorithm is outlined below and with the following notation:

x_k : Parameter vector at the k^{th} iteration.

F : Objective function.

S_k : Direction of search at the k^{th} iteration.

H_k : Hessian matrix at the k^{th} iteration

Algorithm 6.1 (Davidon-Fletcher-Powell Parameter Optimization).

1) Set $k = 0$, x_0 to the initial parameter vector and $H_0 = I$.

2) Find the one-dimensional search direction $S_k = -H_k \cdot \nabla F^T(x_k)$.

3) Minimize along S_k to get x_{k+1}, that is, do a line search

$$\min_{\alpha_k} F(x_k + \alpha_k S_k)$$

4) Set

$$\begin{aligned} p_k &= x_{k+1} - x_k = \alpha_k S_k \\ y_k &= \nabla F^T(x_{k+1}) - \nabla F^T(x_k) \end{aligned}$$

5) Update the Hessian matrix:

$$H_{k+1} = H_k + \frac{p_k p_k^T}{y_k^T p_k} - \frac{H_k y_k y_k^T H_k^T}{y_k^T H_k y_k}$$

6) Check for convergence. If convergence is not achieved, set $k = k + 1$ and go to step (2).

In the present application the convergence check used is whether the change in the Hessian matrix is small. Since the cost is not quadratic, the algorithm takes more than n steps to converge for an n-dimensional optimization problem. The trained networks have been tested and the results are summarized in figures 6.2 through 6.5. It is seen from all figures that the trained neural networks are quite effective in announcing a fault very soon after one occurs.

Figure 6.2 shows residuals from the first fault detection filter, the one that considers sensor faults for the manifold air mass sensor, the engine speed sensor and the forward acceleration sensor. The faults are applied sequentially and one at a time. The decision function of the network announces a zero if there is no fault and a one if there is a fault. Note that all residuals are scaled by a factor of ten before being processed by the network. This scaling can be avoided by scaling the weights of the first layer by the same magnitude. The synaptic weights W and the bias vectors Φ for the first network are:

$$W = \begin{bmatrix} 0.155804 & 0.134528 & -0.151691 \\ -0.154178 & 0.065559 & -0.227977 \\ -0.052750 & 0.187670 & 0.235507 \\ 0.071434 & -0.234352 & 0.001146 \\ -0.001592 & 0.107922 & -0.255098 \\ -0.225173 & -0.104908 & 0.127320 \\ 0.384729 & 0.026842 & -0.194035 \\ -0.176569 & -0.383187 & -0.290295 \\ 0.164612 & -0.155957 & 0.346983 \end{bmatrix} \qquad \Phi = \begin{bmatrix} -0.411395 & 0.586518 & -0.167796 \\ -0.071276 & 0.062116 & -0.014957 \\ 0.535025 & 0.302601 & 1.098216 \end{bmatrix}$$

The first 3 rows of the weight matrix correspond to the first layer of the network. Rows 4, 5 and 6 correspond to the second layer and rows 7, 8 and 9 correspond to the third layer. Similarly, the first row of Φ corresponds to the bias vector of the first layer nodes, and so on.

Figure 6.3 shows residuals from the second fault detection filter, the one that considers sensor faults for the pitch rate sensor, the forward symmetric wheel speed sensor and the rear symmetric wheel speed sensor. The faults again are applied sequentially and one at a time as for the first filter but the residuals are not scaled. The synaptic weights W and bias vectors Φ for the second network are:

$$W = \begin{bmatrix} 0.456071 & 0.149415 & -0.104004 \\ -0.418542 & 0.094649 & -0.164763 \\ -0.184857 & 0.197932 & 0.159792 \\ 0.327377 & -0.444597 & -0.109918 \\ -0.038376 & 0.057827 & -0.254133 \\ -0.339349 & 0.068775 & 0.114895 \\ 0.641334 & 0.029250 & -0.353188 \\ -0.170035 & -0.374427 & -0.328512 \\ 0.135355 & -0.122547 & 0.277558 \end{bmatrix} \qquad \Phi = \begin{bmatrix} 0.049039 & 0.994333 & -0.800334 \\ 0.716702 & 0.147230 & -0.395788 \\ 0.844156 & 0.568572 & 1.471435 \end{bmatrix}$$

Figure 6.4 shows residuals from the third fault detection filter, the filter that considers sensor faults for the heave accelerometer, the pitch rate sensor and the forward symmetric wheel speed sensor. The faults again are applied sequentially and one at a time as for the first filter and again the residuals are not scaled. The synaptic weights W and bias vectors Φ for the third network are:

$$W = \begin{bmatrix} 0.427967 & 0.199235 & -0.134513 \\ -0.397269 & 0.124967 & -0.186824 \\ -0.175831 & 0.283523 & 0.167301 \\ 0.281785 & -0.441553 & -0.126375 \\ -0.088793 & 0.027302 & -0.328136 \\ -0.361911 & 0.031584 & 0.068693 \\ 0.612911 & 0.032435 & -0.342027 \\ -0.183339 & -0.434450 & -0.343878 \\ 0.139504 & -0.135555 & 0.274775 \end{bmatrix} \qquad \Phi = \begin{bmatrix} -0.045264 & 1.124374 & -0.586397 \\ 0.938002 & 0.033609 & -0.384049 \\ 1.030029 & 0.488223 & 1.378053 \end{bmatrix}$$

Finally, figure 6.5 shows residuals from the fourth fault detection filter, the filter that considers actuator faults for the throttle and brake actuators. Note that two fault cases are given for the throttle fault. The synaptic weights W and bias vectors Φ for the fourth network are:

$$W = \begin{bmatrix} 0.377897 & 0.031862 \\ -1.148921 & 0.465636 \\ -0.466482 & 0.999481 \\ 0.059375 & 0.727932 \\ -0.289151 & 0.004747 \\ -0.025186 & 0.283940 \end{bmatrix} \qquad \Phi = \begin{bmatrix} -1.744053 & -0.463958 \\ 1.975143 & -1.321055 \\ 2.845561 & 2.588340 \end{bmatrix}$$

Figure 6.2: Residuals for Fault Detection Filter One: Manifold Air Mass Sensor, Engine Speed Sensor and Forward Acceleration Sensor Faults.

Figure 6.3: Residuals for Fault Detection Filter Two: Pitch Rate Sensor, Forward Symmetric Wheel Speed Sensor and Rear Symmetric Wheel Speed Sensor.

Figure 6.4: Residuals for Fault Detection Filter Three: Heave Accelerometer, Pitch Rate Sensor and Forward Symmetric Wheel Speed Sensor.

Figure 6.5: Residuals for Fault Detection Filter Four: Throttle and Brake Actuators.

Chapter 7
Conclusions

Analytic redundancy is a viable approach to vehicle health monitoring. The fault detection filters developed here are small, third-order linear filters, which should not be a significant computational burden. Evaluating their performance in a high-fidelity nonlinear simulation shows that the filter residuals quickly and clearly respond to the introduction of a fault even in the presence of vehicle nonlinearities. A neural network residual processing system effectively automates fault announcement by examining the fault detection filter residuals for activity characteristic of a static pattern associated with a fault. Faults are announced by the neural network very soon after they are introduced in the vehicle simulation.

By directing development of the project components in parallel and seeing significant progress in all areas, we are able to identify several important areas for future work: model refinement, robust fault detection filter design, health monitoring system evaluation, residual processing and neural network development, and platoon health monitoring.

Model Refinement: Through a good working relation with the Berkeley PATH researchers, development and refinement of the nonlinear simulation will continue. The research division of the Ford Motor Company will be contributing a new tire model. In addition to addressing the fidelity of the vehicle nonlinear dynamics model, simulation development will also involve uncertainty models associated with process disturbances such as winds and roads, sensor measurement uncertainty, system parameter uncertainty and unmodeled dynamics. Fidelity of the modeled nonlinearities and uncertainties is very important for a realistic assessment of any health monitoring system performance.

Robust Fault Detection Filter Design: Development of robust fault detection filters will continue with three directions of investigation. First, the physical system will be examined for the possibility of treating nonlinearities and disturbances as pseudo-fault directions. This approach effectively decouples the nonlinearity or disturbance from fault identifying residuals. Second, analysis will continue to develop a methodology for determining which faults are assigned to which fault detection filters. This approach is incorporated into the longitudinal fault detection filter preliminary designs. Faults are to be grouped based upon their fault directions, residual directions or eigenvalue characteristics to enhance robust performance. Third, parameter uncertainty in the linearized vehicle dynamics is modeled as an input-output decomposition. This allows model uncertainty to be treated as a disturbance.

Evaluation: As the vehicle nonlinear simulation and fault detection filter development continues, evaluation of the health monitoring system will be extended to include a range of operating points. Both longitudinal and lateral modes of vehicle dynamics will be included as will the uncertainties discussed above.

Residual Processing - Neural Networks: The neural network residual processor developed for the preliminary design will be extended to include the entire fault set. Presently, one neural network is designed and trained for each fault detection filter or family of reduced-order fault detection filters. When a fault occurs, a neural network will correctly identify the fault if the fault belongs to the fault detection filter design set. Results are undefined when the fault does not belong to the design fault set. Extending the neural network to include the entire fault set will allow the network, when a fault occurs, to identify first the correct fault design set or sets and then the fault.

A second area of investigation is to research methods for determining probabilities of false and miss alarms in the presence of system uncertainties and nonlinearities. One feature of the reduced-order fault detection filters is that the error dynamics have no special structure. This allows the filter gains to be chosen arbitrarily much like a Kalman filter. With a reduced-order fault detection filter treated as a Kalman filter, modified forms of techniques such as the Shiryayev Sequential Probability Ratio Test (SPRT) are applicable to the residual analysis.

An essential feature of schemes such as the Shiryayev SPRT is to produce thresholds that announce in minimum time whether a fault has occurred within a given probability of false alarm. However, currently, residual tests such as the Shiryayev SPRT are developed only for special stochastic processes. Extending schemes of this type to include residuals from the fault detection filter could proceed by allowing a failure to produce a change in several probability density functions rather than just one.

Platoon Health Monitoring: Work should begin towards extending the health monitoring system for one vehicle to include the presence of multiple vehicles in a controlled platoon configuration. Sensors required for control such as distance measurements will be included in the fault set. Transmission of vehicle sensor outputs will be transmitted to all vehicles. Feasibility and performance of an expanded health monitoring system will be evaluated in an extended nonlinear simulation.

Appendix A
Fault Detection Filter Background

A linear time-invariant system with q failure modes and no disturbances or sensor noise can be modeled (Beard 1971), (White and Speyer 1987), (Massoumnia 1986) by

$$\begin{aligned} \dot{x} &= Ax + Bu + \sum_{i=1}^{q} F_i m_i && \text{(a.1a)} \\ y &= Cx. && \text{(a.1b)} \end{aligned}$$

All system variables belong to real vector spaces $x \in \mathcal{X}$, $u \in \mathcal{U}$, $y \in \mathcal{Y}$ and $m_i \in \mathcal{M}_i$ with $n = \dim \mathcal{X}$, $p = \dim \mathcal{U}$, $m = \dim \mathcal{Y}$ and $q_i = \dim \mathcal{M}_i$. The input $u \in \mathcal{U}$ is known as is the output $y \in \mathcal{Y}$. The failure modes $m_i \in \mathcal{M}_i$ are vectors that are unknown and arbitrary functions of time and are zero when there is no failure. The failure signatures $F_i : \mathcal{M}_i \mapsto \mathcal{F}_i \subseteq \mathcal{X}$ are maps that are known, fixed and unique. A failure mode m_i models the time-varying amplitude of a failure while a failure signature F_i models the directional characteristics of a failure. Assume the F_i are monic so that $m_i \neq 0$ implies $F_i m_i \neq 0$. Actuator and plant faults are modeled with F_i as the appropriate direction from

A or B. For example, a stuck actuator is modeled with F_i as the column of A associated with the actuator dynamics and with $m_i(t) = -u_i(t) + u_{ic}$ where u_{ic} is some constant. A sensor fault can also fit this model with no need for additional dynamics (Beard 1971), (White and Speyer 1987).

Consider a full-order observer of the form

$$\dot{\hat{x}} = (A + LC)\hat{x} + Bu - Ly \tag{a.2a}$$

$$z = C\hat{x} - y. \tag{a.2b}$$

The state estimation error $e = \hat{x} - x$ dynamics are

$$\dot{e} = (A + LC)e - \sum_{i=1}^{q} F_i m_i \tag{a.3}$$

If (C, A) is observable and L is chosen so that $A + LC$ is stable, then in steady-state and in the absence of disturbances and modeling errors, the residual r is nonzero only if a failure mode m_i is nonzero and is almost always nonzero whenever m_i is nonzero. It follows that any stable observer can detect the occurrence of a fault. Simply monitor the residual z and when it is nonzero a fault has occurred. A more difficult task is to determine which fault has occurred and that is what a fault detection filter is designed to do.

A fault detection filter is an observer with the property that when $m_i(t) \neq 0$, the error $e(t)$ remains in a (C, A)-invariant subspace $\mathcal{W}_i$ which contains the reachable subspace of $(A + LC, F_i)$. Thus, the residual remains in the output subspace $C\mathcal{W}_i$. Furthermore, the output subspaces $C\mathcal{W}_1, \ldots, C\mathcal{W}_q$ are independent so that $z \in \sum_{i=1}^{q} C\mathcal{W}_i$ has a unique representation $z = z_1 + \cdots + z_q$ with $z_i \in C\mathcal{W}_i$. The fault is identified by projecting z onto each of the output subspaces $C\mathcal{W}_i$. The following statement of the detection filter problem, sometimes called the Beard-Jones detection filter problem (BJDFP), is essentially the same as that found in (Beard 1971) and (White and Speyer 1987) but is stated in the geometric language of (Massoumnia 1986).

Definition A.1 (Detection Filter Problem). Given the system (a.1), with state-space $\mathcal{X}$ and measurement-space $\mathcal{Y}$, the detection filter problem is to find a set of subspaces

$\mathcal{W}_i \subseteq \mathcal{X}$, $i = 1, \ldots, q$ such that for some map $L : \mathcal{Y} \mapsto \mathcal{X}$ the following conditions are met:

$$\begin{aligned} (A + LC)\mathcal{W}_i \subseteq \mathcal{W}_i & \qquad \text{Subspace invariance.} \\ \mathcal{F}_i \subseteq \mathcal{W}_i & \qquad \text{Failure inclusion.} \\ C\mathcal{W}_i \cap (\sum_{j \neq i} C\mathcal{W}_j) = 0 & \qquad \text{Output separability.} \end{aligned}$$

It can be shown (Massoumnia 1986), (White and Speyer 1987) that when (C, A) is observable, the last condition, output separability, implies that the subspaces $\mathcal{W}_1, \ldots, \mathcal{W}_q$ are independent.

It should be pointed out that for any subspace $\mathcal{F}_i \subseteq \mathcal{X}$ there is a minimal (C, A)-invariant subspace $\mathcal{F}_i \subseteq \mathcal{W}_i^* \subseteq \mathcal{X}$. A method suggested by (Wonham 1985) for computing a minimal invariant subspace is a recursive algorithm, the (C, A)-invariant subspace algorithm.

$$\mathcal{W}_i^* \quad = \quad \lim \mathcal{W}_i^k$$

where

$$\mathcal{W}_i^{k+1} \quad = \quad \operatorname{Im} F_i + A \left(\mathcal{W}_i^k \cap \operatorname{Ker} C \right)$$

and where the recursion begins with

$$\mathcal{W}_i^0 \quad = \quad 0$$

To ensure stability, the invariant subspaces $\mathcal{W}_i$ are usually chosen as a set of mutually detectable, minimal unobservability subspaces or detection spaces (Beard 1971) as they are also called in the context of fault detection. An unobservability subspace $\mathcal{T} \subseteq \mathcal{X}$ or UOS is a subspace with the property that $\mathcal{T}$ is the unobservable subspace of the pair $(HC, A+LC)$ for some L and H. This means not only that $\mathcal{T}$ is (C, A)-invariant but also that the spectrum of $(A + LC)$ induced on the factor space $\mathcal{X}/\mathcal{T}$ may be placed arbitrarily within a conjugate symmetry constraint and with respect to L such that $(A + LC)\mathcal{T} \subseteq \mathcal{T}$. Furthermore, when (C, A) is observable, the entire spectrum of $(A + LC)$ is arbitrary. If $\mathcal{T}(\mathcal{F})$ is the set of

(C, A)-unobservability subspaces that contain $\mathcal{F}$, then it can be shown that $\mathcal{T}(\mathcal{F})$ has a smallest element denoted $\mathcal{T}^*$ (Willems 1982). The detection space is usually found as a minimal UOS, $\mathcal{T}^*$, because there is no known parameterization of all UOS and algorithms exist to compute the minimal UOS (White and Speyer 1987), (Massoumnia 1986).

One method for computing $\mathcal{T}^*$ is suggested by (Wonham 1985) as a numerically stable method for finding supremal controllability subspaces. These are the dual of minimal unobservability subspaces or detection spaces. There are two steps. First, for a fault F_i, find the minimal (C, A)-invariant subspace $\mathcal{W}_i^*$ using the recursive (C, A)-invariant subspace algorithm as explained above. Next, calculate the invariant zero directions of the triple (C, A, F_i), if any. Denote the invariant zero directions as $\mathcal{V}_i$. Then

$$\mathcal{T}_i^* = \mathcal{W}_i^* \oplus \mathcal{V}_i$$

Detection space calculations are described in detail in (Wonham 1985) with amplification and examples given in (Douglas 1993).

Finally, a mutually detectable set of UOS $\{\mathcal{T}_1^*, \ldots, \mathcal{T}_q^*\}$ is one which satisfies Definition A.1 such that the sum $\sum_{i=1}^q \mathcal{T}_i^*$ is also an UOS. While for any one UOS $\mathcal{T}_i$, the spectrum of $(A + LC)$ induced on $\mathcal{X}/\mathcal{T}_i$ may be placed arbitrarily with respect to L, it is not necessarily true that the factor space spectrum is arbitrary when several UOS are considered simultaneously. When a set of UOS $\mathcal{T}_1^*, \ldots, \mathcal{T}_q^*$ is mutually detectable, the spectrum of $(A + LC)$ induced on $\mathcal{X}/\sum_{i=1}^q \mathcal{T}_i^*$ is arbitrary and, when (C, A) is observable, the entire spectrum of $(A + LC)$ is arbitrary.

Once the detection spaces are found, the next step is to find a fault detection filter gain. The gain is not unique and several methods exist for finding one. Eigenstructure assignment algorithms are described in (White and Speyer 1987) and (Douglas and Speyer 1995b) and an $\mathcal{H}_\infty$-bounded fault detection filter is described in (Douglas and Speyer 1995a). The procedure applied in this report is a left eigenvector assignment algorithm introduced in (Douglas 1993) and (Douglas and Speyer 1995b). This procedure is used because it extends directly to one that hedges against sensitivity to parameter uncertainty. Noise robustness algorithms such as the $\mathcal{H}_\infty$-bounded fault detection filter of (Douglas and Speyer 1995a) are not used here because disturbances and sensor noise are not yet included in the vehicle model. Furthermore, later, when they are included, the reduced-order fault detection filters provide a natural way to accommodate noise without the need for redesigning the filter.

The left eigenvector assignment algorithm works by assigning an eigenstructure in the dual space to a set of intersecting detection space annihilators. This means that left eigenvectors, which annihilate the detection spaces, are placed rather than right eigenvectors, which span the detection spaces as is done in (White and Speyer 1987). Since these annihilators intersect, care must be taken to ensure that the assigned eigenvectors are consistent.

Before proceeding, it is necessary to establish a dual relation between unobservability and controllability subspaces. First, introduce the following notation. $\mathcal{X}'$ denotes the dual space of $\mathcal{X}$ and if $C : \mathcal{X} \mapsto \mathcal{Y}$, then C' denotes the dual map $C'\mathcal{Y}' \mapsto \mathcal{X}'$. Writing C^T, the transpose of matrix C, for the dual map C' implies that bases have been chosen for $\mathcal{X}$ and $\mathcal{Y}$. Now, in (Wonham 1985) it is shown that if $\mathcal{T} \subseteq \mathcal{X}$ is a (C, A)-unobservability subspace then the annihilator of $\mathcal{T}$ denoted here by $\mathcal{T}^\perp \subseteq \mathcal{X}'$ is an (A', C')-controllability subspace in the dual system. Second, if $\mathcal{T}$ is a (C, A)-unobservability subspace, the observable part of the system is characterized by the factor space $\mathcal{X}/\mathcal{T}$ and the induced system maps. Furthermore, for any subspace $\mathcal{T} \subseteq \mathcal{X}$, the annihilator of $\mathcal{T}$ and the factor space $\mathcal{X}/\mathcal{T}$ are isomorphic, $\mathcal{T}^\perp \simeq (\mathcal{X}/\mathcal{T})'$.

The dual relation between unobservability and controllability subspaces is useful because any result found for controllability subspaces can be applied easily to the unobservability subspaces of a detection filter. Consider the results of (Moore and Laub 1978) which are paraphrased as follows. The first statement describes a set of vectors in the kernal of C that can be assigned as closed-loop eigenvectors.

Theorem A.1. Let $A : \mathcal{X} \mapsto \mathcal{X}$, $B : \mathcal{U} \mapsto \mathcal{X}$ and $C : \mathcal{X} \mapsto \mathcal{Y}$. Then a set of linearly independent vectors $\{v_1, \ldots, v_k \mid v_i \in \operatorname{Ker} C \subseteq \mathcal{X}\}$ satisfies $(A + BK)v_i = \lambda_i v_i$ for some $K : \mathcal{X} \mapsto \mathcal{U}$ and distinct self-conjugate complex numbers $\lambda_1, \ldots, \lambda_k$ if and only if v_i and v_j

are conjugate pairs when λ_i and λ_j are and there exists a set of vectors $\{w_1, \ldots, w_k | w_i \in \mathcal{U}\}$ such that

$$\begin{bmatrix} A - \lambda_i I & B \\ C & 0 \end{bmatrix} \begin{bmatrix} v_i \\ w_i \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

It follows immediately that for a monic B, a set of vectors $\{v_1, \ldots, v_k\}$ satisfies theorem A.1 if and only if $Kv_i = w_i$.

The second result also from (Moore and Laub 1978) characterizes the set of all eigenvectors that span a supremal (A, B)-controllability subspace $\mathcal{R}^*$.

Theorem A.2. Let $\lambda_1, \ldots, \lambda_k$ be a set of distinct, self-conjugate complex numbers that satisfy

1) $k \geq \dim(\mathcal{R}^*)$ where $\mathcal{R}^*$ is the supremal (A, B)-controllability subspace in $\operatorname{Ker} C$

2) at least one λ_i is real

3) no λ_i or $\operatorname{Re}(\lambda_i)$ is a transmission zero of (C, A, B)

Let V_i and W_i solve

$$\begin{bmatrix} A - \lambda_i I & B \\ C & 0 \end{bmatrix} \begin{bmatrix} V_i \\ W_i \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Then $\mathcal{R}^* = \operatorname{Im} V_1 + \cdots + \operatorname{Im} V_k$.

Given the dual relationship between controllability and unobservability subspaces, the application of Theorems A.1 and A.2 to detection filter design is immediate. First, consider just one detection space $\mathcal{T}_i^*$. Characterize the left eigenvectors that annihilate $\mathcal{T}_i^*$ and find a detection filter gain L_i that produces $\mathcal{T}_i^*$. Next establish a consistency requirement on a detection filter gain L that is to produce q detection spaces $\mathcal{T}_1^*, \ldots, \mathcal{T}_q^*$.

If $\mathcal{T}_i^* \subseteq \mathcal{X}$ with dimension ν_i is a detection space for fault F_i, the annihilator $(\mathcal{T}_i^*)^\perp$ is the supremal controllability subspace of the dual system with $(\mathcal{T}_i^*)^\perp \subseteq \operatorname{Ker} F_i'$ and has dimension $n - \nu_i$. Let $\hat{\Lambda}_i = \{\lambda_{i_1}, \ldots, \lambda_{i_{n-\nu_i}}\}$ be a set of distinct self-conjugate complex numbers that does not include any of the invariant zeros of the triple (F_i', A', C'). By Theorem A.2 the annihilator of $\mathcal{T}_i^*$ satisfies

$$(\mathcal{T}_i^*)^\perp = \operatorname{Im} V_{i_1} + \cdots + \operatorname{Im} V_{i_{n-\nu_i}}$$

where the V_{i_j} are found, along with W_{i_j}, by solving

$$\begin{bmatrix} A^T - \lambda_{i_j} I & C^T \\ F_i^T & 0 \end{bmatrix} \begin{bmatrix} V_{i_j} \\ W_{i_j} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{a.4}$$

where $j = 1, \ldots, n - \nu_i$ and where $\lambda_{i_j} \in \hat{\Lambda}_i$. A set of linearly independent closed-loop left eigenvectors $v_{i_1}, \ldots, v_{i_{n-\nu_i}}$ that spans $(\mathcal{T}_i^*)^\perp$ satisfies Theorem A.1 and is found by solving

$$\begin{bmatrix} A^T - \lambda_{i_j} I & C^T \\ F_i^T & 0 \end{bmatrix} \begin{bmatrix} v_{i_j} \\ w_{i_j} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{a.5}$$

Since $v_{i_j} \in \operatorname{Im} V_{i_j}$ (a.4), the left eigenvectors may not be unique but they are constrained to be arranged in conjugate pairs when the given closed-loop eigenvalues λ_{i_j} are in conjugate pairs.

Now find a detection filter gain L_i. By the remark following Theorem A.1, L_i^T satisfies

$$L_i^T v_{i_j} = w_{i_j} \tag{a.6}$$

and $(A^T + C^T L_i^T) v_{i_j} = \lambda_{i_j} v_{i_j}$ for each $j = 1, \ldots, n - \nu_i$. Form two matrices $\hat{V}_i$ and $\hat{W}_i$

$$\hat{V}_i = \left[v_{i_1}, \ldots, v_{i_{n-\nu_i}} \right] \tag{a.7a}$$

$$\hat{W}_i = \left[w_{i_1}, \ldots, w_{i_{n-\nu_i}} \right] \tag{a.7b}$$

and solve $L_i^T \hat{V}_i = \hat{W}_i$. A real solution for L_i^T always exists because the v_{i_j} are linearly independent and the assigned closed-loop poles λ_{i_j} and eigenvectors v_{i_j} when complex are arranged in conjugate pairs. Finally, L_i, the detection filter gain found as the transpose

$$\hat{V}_i^T L_i = \hat{W}_i^T \tag{a.8}$$

satisfies $(A + L_i C)\mathcal{T}_i^* \subseteq \mathcal{T}_i^*$ and places the spectrum of $(A + L_i C)$ induced on $\mathcal{X}/\mathcal{T}_i^*$ as $\sigma(A + L_i C | \mathcal{X}/\mathcal{T}_i^*) = \hat{\Lambda}_i$.

Because the detection filter has q detection spaces $\mathcal{T}_1^*, \ldots, \mathcal{T}_q^* \subseteq \mathcal{X}$, the detection filter gain L has to satisfy (a.8) for $i = 1, \ldots, q$ or

$$L^T \left[\hat{V}_1, \ldots, \hat{V}_q \right] = \left[\hat{W}_1, \ldots, \hat{W}_q \right] \tag{a.9}$$

Since the $\hat{V}_i$ and $\hat{W}_i$ represent $\sum_{i=1}^{q}(n-\nu_i)$ pairs of vectors (v_{i_j}, w_{i_j}), care must be taken to construct the $\hat{V}_i$ and $\hat{W}_i$ conformably. If (a.9) is to have a solution for L, there can be no more than n distinct pairs (v_{i_j}, w_{i_j}) and of these, the v_{i_j} must be linearly independent and arranged in conjugate pairs if a solution is to be unique and real.

Finding a set of left eigenvectors consistent with (a.9) is not difficult but requires careful bookkeeping. Since $(\mathcal{T}_i^*)^\perp$ and $(\mathcal{X}/\mathcal{T}_i^*)'$ are isomorphic, the closed-loop spectrum induced on the factor space $\mathcal{X}/\mathcal{T}_i^*$ is

$$\sigma(A+L_iC|\mathcal{X}/\mathcal{T}_i^*) = \sigma(A'+C'L_i'|(\mathcal{T}_i^*)^\perp) = \hat{\Lambda}_i$$

If Λ_i is the spectrum of $(A+L_iC)$ restricted to the invariant subspace $\mathcal{T}_i^*$

$$\Lambda_i = \sigma(A+LC|\mathcal{T}_i^*)$$

then the spectrum of $(A+L_iC)$ is just

$$\Lambda = \sigma(A+L_iC) = \Lambda_i \cup \hat{\Lambda}_i \tag{a.10}$$

Now, the subspaces $\mathcal{T}_1^*, \ldots, \mathcal{T}_q^*$ are independent when the faults are output separable and (C, A) is observable (Massoumnia 1986), (White and Speyer 1987), so

$$\Lambda = \Lambda_1 \cup \cdots \cup \Lambda_q \cup \Lambda_0$$

where Λ_0 is a set of $\nu_0 = n - \nu_1 - \cdots - \nu_q$ eigenvalues associated with the complementary space $\hat{\mathcal{X}}_0 = \mathcal{X}/\sum_{i=1}^{q}\mathcal{T}_i^*$, $\nu_0 = \dim(\hat{\mathcal{X}}_0)$,

$$\Lambda_0 \quad = \quad \sigma(A+LC|\mathcal{X}/\sum_{i=1}^{q}\mathcal{T}_i^*)$$

It follows from (a.10) that

$$\hat{\Lambda}_i \quad = \quad \bigcup_{\substack{k=0 \\ k\neq i}}^{q} \Lambda_k \tag{a.11}$$

Since the sets of assigned closed-loop poles $\hat{\Lambda}_i$ intersect, the sets of vectors v_{i_j} and w_{i_j} that solve (a.5) should also form intersecting sets compliant with (a.11). By (a.11), if $\lambda_{i_j} \in \Lambda_i$ for $i \neq 0$, then $\lambda_{i_j} \in \hat{\Lambda}_{k \neq i}$ and the v_{i_j} and w_{i_j} that satisfy (a.5) now must satisfy

$$\begin{aligned} 0 &= (A^T - \lambda_{i_j} I)v_{i_j} + C^T w_{i_j} \\ 0 &= F_1^T v_{i_j} \\ &\vdots \\ 0 &= F_{i-1}^T v_{i_j} \\ 0 &= F_{i+1}^T v_{i_j} \\ &\vdots \\ 0 &= F_q^T v_{i_j} \end{aligned}$$

For $i = 0$ and $\lambda_{i_j} \in \Lambda_0$, then $\lambda_{i_j} \in \hat{\Lambda}_k$ for $k = 1, \dots, q$ and the v_{i_j} and w_{i_j} that satisfy (a.5) now must satisfy

$$\begin{aligned} 0 &= (A^T - \lambda_{i_j} I)v_{i_j} + C^T w_{i_j} \\ 0 &= F_1^T v_{i_j} \\ &\vdots \\ 0 &= F_q^T v_{i_j} \end{aligned}$$

The detection filter gain computation algorithm suggested by (a.5)-(a.9) and modified to force consistency among eigenvectors which span the intersecting detection space annihilators, is as follows.

Algorithm A.1.

1) Find the dimensions of the detection spaces $\nu_i = \dim \mathcal{T}_i^*$ for $i = 1, \dots, q$ and the dimension of the complementary space $\nu_0 = n - \sum_{i=1}^q \nu_i$.

2) Define the complementary fault sets

$$\hat{F}_i = \begin{cases} [F_1, \dots, F_q] & \text{for } i = 0 \\ [F_1, \dots, F_{i-1}, F_{i+1}, \dots, F_q] & \text{for } 1 \leq i \leq q \end{cases} \tag{a.12}$$

Define $(q+1)$ sets of distinct self-conjugate complex numbers $\Lambda_0, \Lambda_1, \dots, \Lambda_q$ where $\dim \Lambda_i = \nu_i$ and where no elements of Λ_i are zeros of the triple $(C, A, \hat{F}_i)$. By the remarks at the end of section A, each of these sets may be specified arbitrarily except for conjugate symmetry when (C, A) is observable and when the detection spaces $\mathcal{T}_i^*$ are mutually detectable.

3) For $i = 0, \ldots, q$ and $j = 1, \ldots, \nu_i$ and for $\lambda_{i_j} \in \Lambda_i$ solve

$$\begin{bmatrix} A^T - \lambda_{i_j} I & C^T \\ \hat{F}_i^T & 0 \end{bmatrix} \begin{bmatrix} v_{i_j} \\ w_{i_j} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{a.13}$$

for pairs (v_{i_j}, w_{i_j}) where the v_{i_j} are linearly independent for all i, j. Let

$$\tilde{V}_i = \left[v_{i_1}, \ldots, v_{i_{\nu_i}} \right] \tag{a.14a}$$

$$\tilde{W}_i = \left[w_{i_1}, \ldots, w_{i_{\nu_i}} \right] \tag{a.14b}$$

4) Solve for the detection filter gain L as

$$\left[\tilde{V}_0, \tilde{V}_1, \ldots, \tilde{V}_q \right]^T L = \left[\tilde{W}_0, \tilde{W}_1, \ldots, \tilde{W}_q \right]^T \tag{a.15}$$

References

Beard, R. V. 1971. *Failure Accommodation in Linear Systems Through Self-reorganization.* PhD thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA.

Douglas, R. K. 1993. *Robust Fault Detection Filter Design.* PhD thesis, The University of Texas at Austin, Austin, TX.

Douglas, R. K. and J. L. Speyer 1995a. An $\mathcal{H}_\infty$ Bounded Fault Detection Filter. In *Proceedings of the ACC.*

Douglas, R. K. and J. L. Speyer 1995b. Robust Fault Detection Filter Design. In *Proceedings of the ACC.*

Massoumnia, M.-A. 1986. A Geometric Approach to the Synthesis of Failure Detection Filters. *IEEE Transactions on Automatic Control*, Vol. AC-31, No. 9.

Moore, B. C. 1981. Principal Component Analysis in Linear Systems: Controllability, Observability, and Model Reduction. *IEEE Transactions on Automatic Control*, Vol. AC-26, No. 1.

Moore, B. C. and A. J. Laub 1978. Computation of Supremal (A, B)-Invariant and Controllability Subspaces. *IEEE Transactions on Automatic Control*, Vol. AC-23, No. 5.

Peng, H. 1992. *Vehicle Lateral Control for Highway Automation.* PhD thesis, Department of Mechanical Engineering, University of California, Berkeley, CA.

White, J. E. and J. L. Speyer 1987. Detection Filter Design: Spectral Theory and Algorithms. *IEEE Transactions on Automatic Control*, Vol. AC-32, No. 7.

Willems, J. C. 1982. Almost Invariant Subspaces: An Approach to High Gain Feedback Design – Part II: Almost Conditionally Invariant Subspaces. *IEEE Transactions on Automatic Control*, Vol. AC-27, No. 5:1071–1085.

Wonham, W. M. 1985. *Linear Multivariable Control: A Geometric Approach.* Springer-Verlag, New York, 3rd edition.