

Robust, Model-based Fault Detection for  
Commercial Transport Air Data Probes

A THESIS  
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL  
OF THE UNIVERSITY OF MINNESOTA  
BY

Paul Freeman

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR THE DEGREE OF  
MASTER OF SCIENCE

Gary J. Balas, co-advisor  
Peter Seiler, co-advisor

November 2011

© Paul Freeman 2011

# Acknowledgments

I would like to take a moment to thank those that have supported and influenced me along the way to this milestone. My family has been very supportive every step of the way, and I have to give them my gratitude for their love and encouragement.

Thanks to everyone from the lab, especially those older guys that helped answer many questions of mine over the years: Paw, Rohit, Arnar, and Abhijit (aka Ghosh).

Thanks to Mihailo Jovanovic for his engaging teaching and service on my committee. I must also thank Goodrich Sensors and Integrated Systems for its financial and technical support of this research. Thanks to Bill Kunik, Matt Sandnas, Brian Mathias, Tim Golly, Roger Foster, and others who have helped with this work.

Great thanks to Gary Balas and Pete Seiler for their guidance, encouragement, and patience throughout this project and the rest of my education at Minnesota. It has been a challenging but rewarding experience, and I have made great strides these last few years thanks to you guys.

Finally, I have to thank the sponsors of this research. This material is based upon work supported by the National Science Foundation under Grant No. 0931931 entitled CPS: Embedded Fault Detection for Low-Cost, Safety-Critical Systems. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

# Dedication

*To my family*

# Abstract

Air data probes provide essential sensing capabilities to aircraft. The loss or corruption of air data measurements due to sensor faults jeopardizes an aircraft and its passengers. To address such faults, sensor hardware redundancy is typically combined with a voting system to detect and discard erroneous measurements. This approach relies on redundancy, which may lead to unacceptable increases in system weight and cost. This thesis presents an alternative, model-based approach to fault detection for a non-redundant air data system. The model-based fault detection strategy uses robust linear filtering methods to reject exogenous disturbances, e.g. wind, and provide robustness to model errors. The proposed algorithm is applied to NASA's Generic Transport Model aircraft with an air data system modeled based on manufacturer data provided by Goodrich Sensors and Integrated Systems. The fault detection filter is designed using linearized models at one flight condition. The detection performance is evaluated at a particular reference flight condition using linear analysis and nonlinear simulations. Detection performance across the flight envelope is examined, and scheduling and blending techniques used to improve detection robustness across an expanded flight regime are explored.

# Contents

<b>List of Tables</b>	<b>vi</b>
<b>List of Figures</b>	<b>ix</b>
<b>Nomenclature</b>	<b>xii</b>
<b>Chapter 1 Introduction</b>	<b>1</b>
1.1 Fault-Tolerance in the Aerospace Industry . . . . .	2
1.2 Historical Failures due to Air Data Faults . . . . .	2
1.3 Motivation for Analytical Redundancy . . . . .	4
1.4 Model-based Approach . . . . .	5
<b>Chapter 2 Air Data</b>	<b>6</b>
2.1 Air Data Sensor Systems . . . . .	6
2.2 Fault Modes and Modeling . . . . .	10
<b>Chapter 3 Aircraft Model</b>	<b>29</b>
3.1 Generic Transport Model Longitudinal Dynamics . . . . .	29
3.2 Actuator Dynamics . . . . .	31
3.3 Aircraft Trim and Model Linearization . . . . .	32

3.4	Control Law . . . . .	33
3.5	Inertial Sensors . . . . .	37
<b>Chapter 4 Fault Detection: <math>H_\infty</math>-synthesis</b>		<b>39</b>
4.1	$H_\infty$ Problem Formulation . . . . .	39
4.2	Signal Weighting Methodology . . . . .	41
4.3	FDI Filters . . . . .	51
<b>Chapter 5 <math>H_\infty</math> Fault Detection Results</b>		<b>53</b>
5.1	Chosen Faults and Desired Filter Performance . . . . .	53
5.2	Step Fault Detection . . . . .	54
5.3	Time-Varying Fault Detection . . . . .	56
5.4	Filter Disturbance Rejection . . . . .	60
5.5	Summary . . . . .	63
<b>Chapter 6 Scheduling and Estimate Blending</b>		<b>66</b>
6.1	Control and Filter Scheduling . . . . .	67
6.2	Uncertainty Modeling and $DK$ -iteration . . . . .	74
6.3	Estimate Blending . . . . .	79
6.4	Summary . . . . .	82
<b>Chapter 7 Conclusion</b>		<b>83</b>
<b>Bibliography</b>		<b>87</b>

# List of Tables

2.1	Blockage fault effects for pitot-static probes . . . . .	26
3.1	Aircraft and environment parameters . . . . .	30
3.2	Pitch-angle tracker parameters . . . . .	35
3.3	Airspeed tracker parameters . . . . .	37
3.4	Altitude tracker specifications . . . . .	37
3.5	Inertial sensor parameters . . . . .	38

# List of Figures

1.1	Examples of pitot-static probes . . . . .	3
2.1	Pitot-static probe . . . . .	8
2.2	Air data sensor architecture . . . . .	8
2.3	Pitot-static probe with drain hole blockage . . . . .	12
2.4	Flight envelope grid points for wind tunnel experiments . . . . .	14
2.5	Comparison of nominal and faulted air data measurements for a drain hole blockage; $\alpha = -15^\circ$ . . . . .	15
2.6	Comparison of nominal and faulted air data measurements for a drain hole blockage; $\alpha = -10^\circ$ . . . . .	16
2.7	Comparison of nominal and faulted air data measurements for a drain hole blockage; $\alpha = -5^\circ$ . . . . .	17
2.8	Comparison of nominal and faulted air data measurements for a drain hole blockage; $\alpha = 0^\circ$ . Note the vertical axis change from the previous three figures. . . . .	18
2.9	Comparison of nominal and faulted air data measurements for a drain hole blockage; $\alpha = 5^\circ$ . . . . .	19
2.10	Comparison of nominal and faulted air data measurements for a drain hole blockage; $\alpha = 10^\circ$ . . . . .	20
2.11	Comparison of nominal and faulted air data measurements for a drain hole blockage; $\alpha = 15^\circ$ . . . . .	21

2.12	Pitot-static probe with pitot inlet blockage . . . . .	22
2.13	Pitot-static probe with combined pitot inlet and drain hole Blockage .	23
2.14	Pitot-static probe with static port blockage . . . . .	24
2.15	Pitot-static probe with water blockages . . . . .	25
3.1	Autopilot control law architecture . . . . .	33
3.2	Comparison of SAS frequency responses . . . . .	36
4.1	Standard system configuration for $H_\infty$ optimal control design . . . . .	40
4.2	Interconnection for generalized plant $genGTM$ . . . . .	42
4.3	Interconnection for $H_\infty$ filter synthesis . . . . .	43
4.4	$W_{fhat}$ : magnitude frequency cesponse . . . . .	45
4.5	Fault estimates for 0.1 psi step fault in static pressure signal . . . . .	46
4.6	$W_{nois}$ : magnitude frequency response . . . . .	48
4.7	$W_{fault}$ : magnitude frequency response . . . . .	50
4.8	$W_{error}$ : magnitude frequency response . . . . .	52
5.1	Fault estimation: $p_s$ step fault, linear and nonlinear GTM . . . . .	55
5.2	Fault estimation: $p_t$ step, linear and nonlinear GTM . . . . .	57
5.3	Fault estimation: simultaneous $p_s$ and $p_t$ step faults, linear and non-linear GTM . . . . .	58
5.4	Nonlinear GTM control input and state responses: simultaneous $p_s$ and $p_t$ step faults . . . . .	59
5.5	Fault estimation: $p_s$ 0.03 rad/s sinusoidal fault, linear and nonlinear GTM . . . . .	61
5.6	Fault estimation: $p_s$ 0.3 rad/s sinusoidal fault, linear and nonlinear GTM	62

5.7	Disturbance rejection: airspeed reference command steps in nonlinear GTM simulation . . . . .	64
5.8	Disturbance rejection: altitude reference command steps in nonlinear GTM simulation . . . . .	65
6.1	Fault detection: performance across airspeed swept models with $p_s$ step fault . . . . .	68
6.2	Fault detection: $p_s$ step fault with control scheduling over [75 90] knots airspeed interval . . . . .	70
6.3	Fault detection: $p_t$ step fault with control scheduling over [75 90] knots airspeed interval . . . . .	71
6.4	Fault detection: $p_s$ step with control and filter scheduling over [75 90] airspeed interval . . . . .	72
6.5	Fault detection: $p_t$ step with control and filter scheduling over [75 90] airspeed interval . . . . .	73
6.6	System configuration for uncertain plant control synthesis . . . . .	75
6.7	Multiplicative input uncertainty . . . . .	76
6.8	Fault detection performance comparison for $DK$ -iteration and $H_\infty$ -synthesis with multiplicative input uncertainty . . . . .	78
6.9	Disturbance rejection: -20 knot airspeed step with scheduled control and blended $DK$ -iteration filters . . . . .	80
6.10	Disturbance rejection with static and total pressure step faults: -20 knot airspeed step with scheduled control and blended $DK$ -iteration filters . . . . .	81

# Nomenclature

## Abbreviations and Acronyms

<i>AOA</i>	Angle of Attack
<i>EDL</i>	Entry, Descent, and Landing system
<i>FAA</i>	Federal Aviation Administration
<i>FDI</i>	Fault Detection and Identification
<i>FDIR</i>	Fault Detection, Identification, and Reconfiguration
<i>GTM</i>	Generic Transport Model
<i>LFT</i>	Linear Fractional Transformation
<i>LPV</i>	Linear Parameter Varying
<i>NASA</i>	National Aeronautics and Space Administration
<i>RVSM</i>	Reduced Vertical Separation Minimum
<i>SAS</i>	Stability Augmentation System

## List of Symbols

$\alpha$	Angle of attack, <i>deg</i>
$\alpha_m$	Angle of attack measurement, <i>deg</i>
$\Delta$	Norm-bounded uncertain matrix
$\delta_{elev}$	Elevator deflection command, <i>deg</i>
$\delta_{th}$	Throttle setting command, <i>percent</i>
$\epsilon_2, \epsilon_3$	Rotation angles from engine axis to body axis, <i>deg</i>
$\rho$	density, <i>lb/ft<sup>3</sup></i>
$\theta$	Pitch angle, <i>deg</i>
$\theta_{cmd}$	Pitch angle command, <i>deg</i>
$\theta_m$	Pitch angle measurement, <i>deg</i>

$A_0$	Speed of sound at sea level, <i>knots</i>
$A_G, B_G, C_G, D_G$	GTM state-space matrices
$A_{SAS}, B_{SAS}, C_{SAS}, D_{SAS}$	SAS state-space matrices
$\bar{c}$	Wing mean aerodynamic chord, <i>ft</i>
$C_x, C_z, C_m$	Body axis aerodynamic coefficients
$C_D, C_L, C_M$	Wind axis aerodynamic coefficients
$D$	Drag force
$\mathbf{e}$	Fault estimate errors signal
$F$	Fault detection filter
$\hat{\mathbf{f}}$	Fault estimate signal
$\mathbf{f}$	Fault signal
$f_{ps}$	Static pressure measurement fault, <i>psi</i>
$f_{pt}$	Total pressure measurement fault, <i>psi</i>
$F_L$	Lower LFT
$F_U$	Upper LFT
$g$	Gravitational constant, <i>ft/s<sup>2</sup></i>
$h$	Altitude, <i>ft</i>
$h_{cmd}$	Altitude command, <i>ft</i>
$h_m$	Altitude measurement, <i>ft</i>
$I_{yy}$	Pitch axis moment of inertia, <i>slugs-ft<sup>2</sup></i>
$K_h$	Altitude tracker
$K_V$	Airspeed tracker
$K_\theta$	Pitch angle tracker
$L$	Lift force, <i>lb</i>
$M$	Aerodynamic moment, <i>ft-lb</i>
$m$	Aircraft mass, <i>slugs</i>
$\mathbf{n}$	Noise signal
$n_{ENG}$	Number of engines
$n_\alpha$	Angle of attack measurement noise, <i>deg</i>
$n_\theta$	Pitch angle measurement noise, <i>deg</i>
$n_{ps}$	Static pressure measurement noise, <i>psi</i>
$n_{pt}$	Total pressure measurement noise, <i>psi</i>
$n_q$	Pitch rate measurement noise, <i>deg/s</i>
$p_{dyn}$	Dynamic pressure, <i>psi</i>
$p_{probe}$	Wind tunnel test probe pressure measurement, <i>psi</i>

$p_{ref}$	Wind tunnel reference probe pressure measurement, <i>psi</i>
$p_{s0}$	Static pressure at sea level, <i>psi</i>
$p_s$	True static pressure, <i>psi</i>
$p_{sm}$	Static pressure measurement, <i>psi</i>
$p_{sND}$	Nondimensionalized static pressure
$p_t$	True total pressure, <i>psi</i>
$p_{tm}$	Total pressure measurement, <i>psi</i>
$p_{tND}$	Nondimensionalized total pressure
$q$	Pitch rate, <i>deg/s</i>
$\bar{q}$	Dynamic pressure, <i>psi</i>
$\hat{q}$	Normalized pitch rate
$q_c$	Dynamic pressure, <i>psi</i>
$q_m$	Pitch rate measurement, <i>deg/s</i>
$R$	Ideal gas constant, $\frac{ft \cdot lb}{\text{R} \cdot slug}$
$\mathbf{r}$	Reference command signal
$r_x, r_z$	Thrust moment arm components, <i>ft</i>
$S$	Wing area, <i>ft<sup>2</sup></i>
$t$	Time, <i>s</i>
$T_m$	Thrust moment <i>ft-lb</i>
$T_x, T_z$	Thrust force components, <i>ft-lb</i>
$\bar{\mathbf{u}}$	GTM equilibrium input vector
$V$	Airspeed, <i>knots</i>
$V_c$	Calibrated airspeed, <i>knots</i>
$V_{cmd}$	Airspeed command, <i>knots</i>
$V_m$	Airspeed measurement, <i>knots</i>
$W_{error}$	Error weighting function
$W_{fault}$	Fault weighting function
$W_{fhat}$	Fault estimate weighting function
$W_I$	Input uncertainty weighting function
$W_{nois}$	Inertial sensor noise weighting function
$W_{ref}$	Reference command weighting function
$\bar{\mathbf{x}}$	GTM equilibrium state vector
$\mathbf{y}$	Measurements signal

# Chapter 1

## Introduction

New methods of system monitoring, fault detection, and fault diagnosis are often required to enhance the safety and reliability of increasingly advanced technological systems. These performance demands are particularly important for systems in which a malfunction can result in significant equipment damage, costly environmental harms, injuries, or deaths. Often called *safety-critical* or *life-critical* systems, these platforms require the highest resilience to malfunction that engineers can achieve. Such processes are traditionally found in energy infrastructure (e.g. power plants, nuclear reactors), medical devices (e.g. implant devices, ventilation systems, robotic surgery machines), transportation (e.g. automobiles, ships, aerospace systems), and weapons systems. Today, modern information systems governed by vast arrays of computers (e.g. telecommunications, banking and financial systems, transportation control infrastructure) are essential to the preservation of life and equipment and are breaking traditional notions of safety-critical applications. In the future, technologies utilizing elements of fault-tolerance will become more common and more relied upon by society.

The unique operational missions associated with each platform engender different responses to system failure. These different responses can be broadly classified as reliability regimes with each regime preserving the mission goals emphasized by the system designer. One such regime is *fail-operational*, in which a system continues to operate when control systems have failed. Fail-operational modes can be unsafe and are sometimes avoided. *Fail-safe* systems revert to a safe mode upon control system failure. One example of this system type is fire-on-command nuclear missiles. When

communication systems fail, the missiles cannot be commanded to fire, and accidental launches are avoided. Contrast this standard approach to a proposed fail-operational Soviet missile control architecture that fired missiles upon loss of communication. For high risk applications such as this, a conservative fail-safe reliability is favored. Between these two extremes – and far more difficult to successfully achieve – is *fault-tolerance*. Fault-tolerant systems avoid operational failures when faults occur by executing a reconfigured control strategy which allows the system to operate safely despite the presence of a fault. This approach requires algorithms yielding rapid, accurate *Fault Detection, Identification, and Control Reconfiguration*, or *FDIR*.

Each component of FDIR presents significant challenges and are somewhat independent problems. This thesis focuses on the development of fault detection algorithms for the critical air data sensors on commercial transport aircraft. This research works toward the goal of enhancing aircraft fault-tolerance beyond that of existing fleets. The general algorithms contained within this thesis, however, could be extended to other safety-critical aerospace applications or problems within other industries.

## 1.1 Fault-Tolerance in the Aerospace Industry

Stringent safety requirements have driven aircraft system design for decades. The system availability and integrity requirements for commercial flight control electronics are typically no more than  $10^{-9}$  catastrophic failures per flight hour [1,2]. The typical industry design solution is based extensively on physical redundancy at all levels of the design. For example, the Boeing 777 has 14 spoilers, 2 outboard ailerons, 2 flaperons, 2 elevators, one rudder and leading/trailing edge flaps [3,4]. Each of these surfaces is driven by two or more actuators, all connected to different hydraulic systems. Moreover, the control law software is implemented on three primary flight computing modules. Each computing module contains three dissimilar processors with control law software compiled using dissimilar compilers. The inertial and air data sensors have a similar level of redundancy.

## 1.2 Historical Failures due to Air Data Faults

Nearly all aircraft utilize pitot-static probes, such as those shown in Figure 1.1, to measure total and static pressure to determine airspeed and altitude. For proper operation, the probe ports and inlets must be free of blockages (e.g. icing blockages).



Figure 1.1: Examples of pitot-static probes

Failures of these probes have resulted in numerous fatal accidents of commercial, military, and general aviation aircraft. To address these failures, sensor hardware redundancy is typically combined with voting systems to detect and discard erroneous measurements. The redundancy-dependant control architectures used in the aircraft industry achieve extraordinarily high levels of availability and integrity, yet catastrophic failures do occur. Three recent examples of notable air data system failures are presented:

**FedEx/McDonnell-Douglas MD-11 crash – 1999** In October 1999, a FedEx MD-11 aircraft attempted a landing at Subic Bay, Philippines. After touchdown, the plane rolled onto another runway, through a fence, and into the bay. According to the investigators’ report, the probable cause of the incident was: “The failure of the flight crew to properly address an erroneous airspeed indication during descent and landing, their failure to verify and select the correct airspeed by checking the standby airspeed indicator, and their failure to execute a missed approach. These failures led to an excessive approach and landing speed that resulted in a runway overshoot. Contributing factors to the accident were clogged pitot tube drain holes, the MD-11’s insufficient alerting system for airspeed anomalies, and the failure of the...checklists to refer the crew to the standby airspeed indicator.” [5]

**B-2 Spirit bomber crash – 2008** In February 2008, a USAF B-2 Spirit bomber collected erroneous air data measurements that led to a crash and total loss of the \$1.4 Billion aircraft in Guam. According to investigators, moisture in the aircraft’s Port Transducer Units during air data calibration distorted the information in the bomber’s air data system, causing the flight control computers to calculate an inaccurate air-speed and a negative angle of attack upon takeoff. This led to an, “uncommanded 30 degree nose-high pitch-up on takeoff, causing the aircraft to stall and its subsequent crash.” [6]

**Air France/Airbus A330 Flight 447 crash – 2009** In June 2009, Air France Flight 447 plummeted 38,000 feet in only 3.5 minutes and crashed mysteriously over the Atlantic Ocean en route to Paris. The flight data recorders were recovered after a nearly two-year hunt. The recovered data seemed to indicate that the pilots had conflicting air speed data in the minutes leading up to the crash. The aircraft climbed to 38,000 feet when “the stall warning was triggered and the airplane stalled,” the investigators’ report says. Investigators suspect that the aircraft’s pitot probes malfunctioned due to ice at high altitude. [7]

### **1.3 Motivation for Analytical Redundancy**

While the current physical-redundancy schemes are quite reliable and these failures described are rare events, the use of physical redundancy dramatically increases system size, complexity, weight, and power consumption. Moreover, such systems are extremely expensive in terms of design and development costs as well as the unit production costs. There is an increasing demand for high-integrity, yet low cost, fault tolerant aerospace systems, e.g. unmanned aerial vehicles and fly-by-wire capabilities in lower end business and general aviation aircraft.

This demand extends beyond the atmosphere; space technology applications are strong candidates for implementation of these improved fault-tolerance algorithms because they require high reliability under extensive constraints. NASA has recently published a series of Space Technology Roadmaps highlighting technologies important to advancing American space capabilities. These roadmaps indicate that enhanced fault tolerance and health monitoring algorithms are essential for Robotics, Tele-robotics and Autonomous systems as well as Entry, Descent, and Landing (EDL)

systems. [8] [9] Substantial improvements in autonomy technology will be required in order to expand space exploration capabilities in the coming decades. Many existing advanced space platforms are ‘automated systems’, meaning that in unforeseen conditions or faulted states, they require human interaction in order to choose a course of action. On the other hand, autonomous systems are able to resolve decisions independently. This capacity increases system operations capability and improves mission robustness to uncertain environments and phenomena. Fault tolerance plays a particularly important role in autonomy since system health assessment is essential to many real-time decision-making and longer term mission-planning processes.

The demands are high; fault tolerance systems must be fast and accurate. This thesis seeks to improve robustness of fault detection algorithms so that systems maintain sufficiently accurate health monitoring across their performance envelope. Leveraging analytical tools that may make mathematical performance guarantees can assist verification and validation, reducing risk in operation of safety-critical systems.

## 1.4 Model-based Approach

The fault detection problem usually comprises a method to compute residuals and a process to declare faults based on the residuals. It is desired that the generated residual be a good representation of the fault of interest while being insensitive to process and measurement noises. Generation of residuals depends on the information available about the system. If a sufficiently accurate model of the system is available, model based methods can be used to estimate system states and outputs. See [10], [11], and [12] for a detailed treatment of model based and model-free fault detection methods. Based on these methods, this paper uses the  $H_\infty$  framework to design an analytical fault detection filter for an air data system.

This thesis has the following structure. Chapter 2 describes the model structure and operation of air data systems as well as many common fault classifications, their manifestations, and their models. Models for the aircraft, controls, and additional sensor systems are provided in Chapter 3. Chapter 4 describes the  $H_\infty$  methods used to design the robust fault detection filter. Simulation results and analysis for the initial  $H_\infty$  designs are given in Chapter 5. In Chapter 6, the fault detection algorithms are refined using scheduling and blending techniques. Finally, Chapter 7 discusses conclusions of this work and directions for future research.

# Chapter 2

## Air Data

This chapter describes the basic physical relationships used to create air data measurements, the structure and operation of a typical pitot-static air data probe, and common fault modes for such sensors. This information is used to construct fault models for use in simulations that are important to the development of fault detection algorithms.

### 2.1 Air Data Sensor Systems

The basic relationships between air data measurements and aircraft states are derived in [2]. For altitudes in the troposphere (up to  $\approx 36,000$  ft), the static pressure  $p_s$  is related to altitude  $h$  by:

$$h = \frac{T_0}{L} \left( 1 - \left( \frac{p_s}{p_{s_0}} \right)^{LR/g} \right) \quad (2.1)$$

where  $T_0 := 518.67^\circ\text{R}$  is the temperature at sea level,  $L := 0.00356 \frac{^\circ\text{R}}{\text{ft}}$  is the troposphere lapse rate,  $g := 32.17 \frac{\text{ft}}{\text{s}^2}$  is the gravity constant at sea level,  $p_{s_0} := 2116.21 \frac{\text{lb}}{\text{ft}^2}$  is the static pressure at sea level, and  $R := 1716 \frac{\text{ft}\cdot\text{lb}}{^\circ\text{R}\cdot\text{slug}}$  is the ideal gas constant.

For compressible air and subsonic speeds, the static and total (pitot) pressures,  $p_s$

and  $p_t$ , are related to the calibrated (indicated) airspeed  $V_c$  (knots) by:

$$V_c = A_0 \sqrt{5 \left( \frac{p_t - p_s}{p_{s0}} + 1 \right)^{2/7} - 5}; \quad (2.2)$$

where  $A_0 := 661.48$  knots is the speed of sound at sea level. The calibrated airspeed is equal to the true airspeed at sea level but the two airspeeds differ at altitudes above sea level. A more accurate model at high altitudes would include a model of the total air temperature sensor used to compute true airspeed [13]. This is neglected in this thesis; hence, the air data models are only valid at low altitudes. Because the aircraft model described in Chapter 3 is only operational in low-altitude flight regimes, this assumption holds for the analysis presented in this thesis.

### 2.1.1 Pitot-Static Probe Model

A fully operational pitot-static probe, as shown in Figure 2.1, measures static pressure,  $p_s$  and total pressure,  $p_t$ , using independent pressure lines and transducers. The probe captures freestream airflow via the pitot inlet port, and the moving air stagnates within the channels inside the probe. The total pressure is then measured at the pitot pressure output. A static pressure measurement is obtained via small-diameter static ports that are flush with the fuselage of the probe. The static pressure is measured by a pressure transducer at the static pressure output. Dynamic pressure,  $p_{dyn} = p_t - p_s$ , is a calculated quantity [2, 13].

Air entering the probe frequently contains significant moisture, and this moisture must be removed from the probe in order to prevent excessive accumulation of water that can lead to blockage faults. In a fully-operational probe, gravity moves water inside the probe over a small drain hole. The pressure differential between the pitot inlet (total pressure) and drain hole (static pressure) forces water out of the probe via the drain hole. The drain hole is quite small in comparison to the pitot opening, and as a result there is minimal pressure loss at the pitot pressure output.

The pressure measurement devices are modeled by inverting the functions in Equations 2.1-2.2 to obtain values of static and total pressure from the GTM altitude and airspeed states. To model sensor noise and faults in the pressure measurements, the nominal pressure signals are corrupted by white noise and faults are added to the pressure signals to yield pressure measurements.

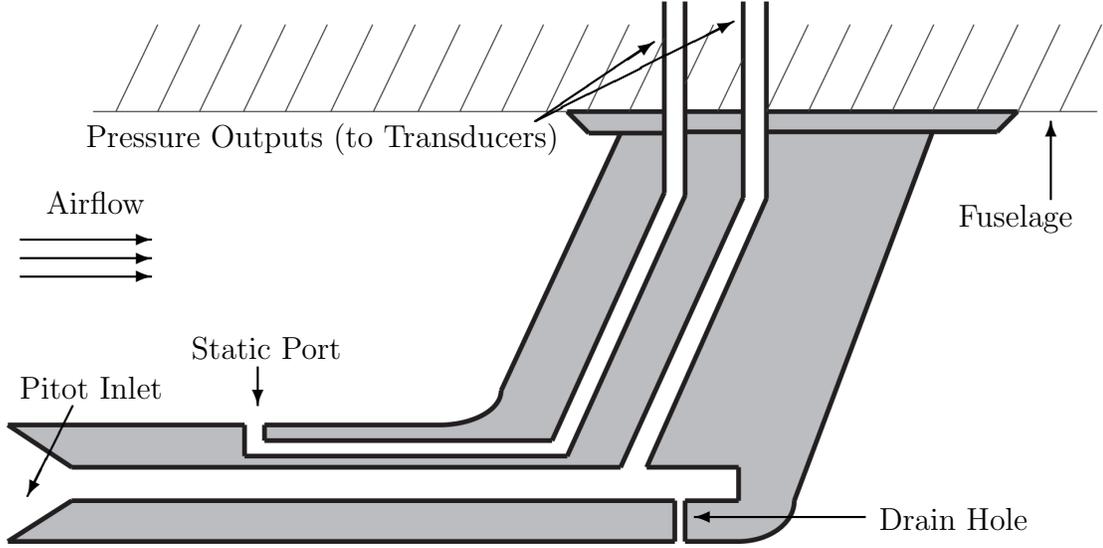


Figure 2.1: Pitot-static probe

### 2.1.2 Pressure Measurement Processor Model

Pressure transducer measurements are processed onboard the aircraft to yield derived altitude and airspeed measurements for feedback to the control loops and to the pilot. Equations 2.1-2.2 are applied in the simulation to produce altitude and airspeed measurements from pressure measurements. In this thesis, all pressure signals have units of psi (lbs/in<sup>2</sup>). The air data system architecture model is depicted in Figure 2.2. Linearizing the Air Data Conversion equations provides insight into

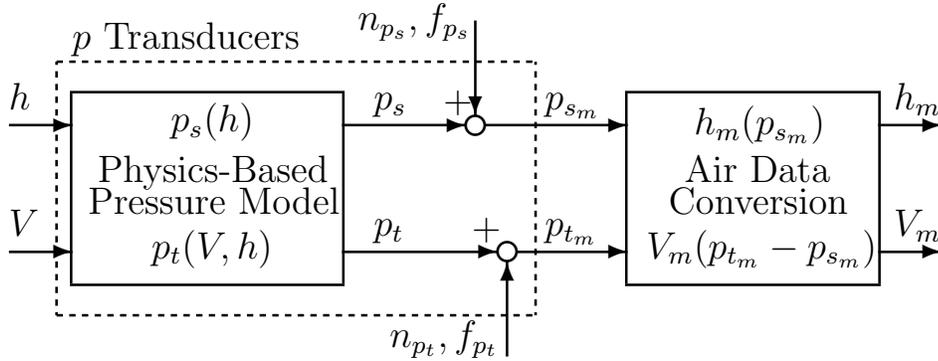


Figure 2.2: Air data sensor architecture

appropriate magnitudes for injected faults:

$$\begin{bmatrix} \delta h_m \\ \delta V_m \end{bmatrix} = \begin{bmatrix} -1911 & 0 \\ -281.4 & 281.4 \end{bmatrix} \begin{bmatrix} \delta p_s \\ \delta p_t \end{bmatrix} \quad (2.3)$$

From Equation 2.3, a fault (error) of magnitude 0.01 psi in  $p_s$  would yield an  $h_m$  error of -19.11 ft and -2.81 knots in  $V_m$ . A fault of the same size in  $p_t$  will yield an  $V_m$  error of 2.81 knots. Equation 2.3 shows that faults injected on  $p_s$  and  $p_t$  both influence  $V_m$ . If the faults are simultaneous in both  $p_s$  and  $p_t$  and of equal magnitude,  $V_m$  will be unaffected.

### 2.1.3 Industry Standards

The accuracy specifications of air data systems are driven primarily by vehicle operational requirements. Commercial and general aviation aircraft are subject to civil air traffic controllers and Federal Aviation Administration (FAA) regulations in the United States. With ever-increasing flight traffic and limited airport and air traffic control resources, regulators are establishing stringent requirements for air data systems in order to ensure that aircraft can safely maintain small separations in the airspace. One example of such requirements involves vertical aircraft separation. In 2005, regulators amended the Reduced Vertical Separation Minima (RVSM) standard of 2000 feet to a 1000 feet standard in the United States with a goal of increasing airspace capacity and allowing aircraft to operate closer to optimum flight trajectories and ensure fuel savings. Thus, a RVSM-certified aircraft must maintain a minimum of 1000 feet vertical separation between another RVSM aircraft and 2000 feet between a non-RVSM aircraft. As of 2011, this standard has been implemented nearly worldwide. Russia, currently in the process of planning a transition to the new standard, is the last major nation yet to implement the protocol.

Certification for RVSM compliance requires particular avionics systems that each have their own specifications. In the European Union, an operator shall ensure that airplanes operated in RVSM airspace are equipped with:

1. Two independent altitude measurement systems;
2. An altitude alerting system;
3. An automatic altitude control system; and
4. A secondary surveillance radar (SSR) transponder with altitude reporting system that can be connected to the altitude measurement system in use for altitude keeping.

The United States maintains similar standards. [14] Additionally, in order to ensure that aircraft can safely operate in close proximity, the maximum allowable altimetry system error is 245 feet. Thus a static pressure fault as large as  $\pm 0.1$  psi would lead to an altitude error of nearly 200 feet; an error this size would certainly be cause for concern for an RVSM-certified aircraft.

The FAA seeks certifiable fault-tolerance capabilities for low-cost, unmanned aircraft that will allow for safe operation within controlled airspace. As described in Chapter 1, this research seeks effective fault-tolerance approaches predicated upon analytical redundancy. Small, low-cost UAVs may not meet the avionics requirements for RVSM certification, but analytical approaches may aid in achieving a similar degree of performance.

These standards will inform fault modeling to ensure that the fault detection algorithms are tested against realistically severe air data system faults.

## **2.2 Fault Modes and Modeling**

This chapter details common fault modes for pitot-static probes, their potential causes, and their effects on air data measurements. There are three broad classifications of fault mechanisms in air data probes: blockage faults, airflow disruption faults, and heater-related faults. Some faults induce the same qualitative measurement errors across all flight conditions, while other faults will manifest themselves differently depending upon aircraft maneuvering. Some of these common blockage fault modes are targeted for this thesis. Airflow disruption faults and heater-related faults require advanced modeling beyond the focus of this thesis, yet remain relevant problems in the aerospace industry. A description of the different fault modes follows.

### **2.2.1 Blockage and Leak Faults**

Pitot-static probes are exposed to the elements. This exposure makes these probes vulnerable to blockage faults; the impact on the aircraft depends on the location of the blockage(s) and sometimes the maneuvers the aircraft happens to be undergoing. The static and pitot pressures internal to the probe are conveyed to pressure transducers via pneumatic lines. Faults in the pneumatic lines can significantly alter air data measurements.

Blockage faults can occur as a result of a variety of causes:

- Exterior Icing (unpowered heater, failed heater, conditions exceed anti-icing capabilities)
- Water or ice accumulation in pressure line
- Insects (e.g. Mud Dauber Wasp)
- Bird Strike
- Debris
- Protective cover left in place

An overview of common blockage faults is now provided.

#### 2.2.1.1 Drain Hole Blockage

With a blocked drain hole and an open pitot inlet (Figure 2.3), the air data measurements will be slightly erroneous. The air data transducers are calibrated to account for the pitot pressure losses that occur with a nominally-performing (i.e. open) drain hole. In the event that the hole experiences a blockage, this calibration induces a small error – a small false increase in airspeed. Moreover—and perhaps more seriously—the probe has a high probability of collecting water which can enter the pressure lines and corrupt measurements. The static pressure measurement, and hence the altitude measurement, will be unaffected by a drain hole blockage.

To illustrate these effects with an example using experimental data, a pitot-static probe with a blocked drain hole was tested in a subsonic wind tunnel on a  $[V_s - \alpha]$  grid in order to simulate a sizeable flight envelope. Figure 2.4 shows the airspeed and angle-of-attack combinations used during the experiment. Comparisons of nominal and faulted static and total pressure measurements across airspeed sweeps for each angle of attack tested are shown in Figures 2.5-2.11. The pressures are presented in nondimensional form; the error between the test probe measurement  $p_{probe}$  and a calibrated tunnel reference probe measurement  $p_{ref}$  is nondimensionalized by the

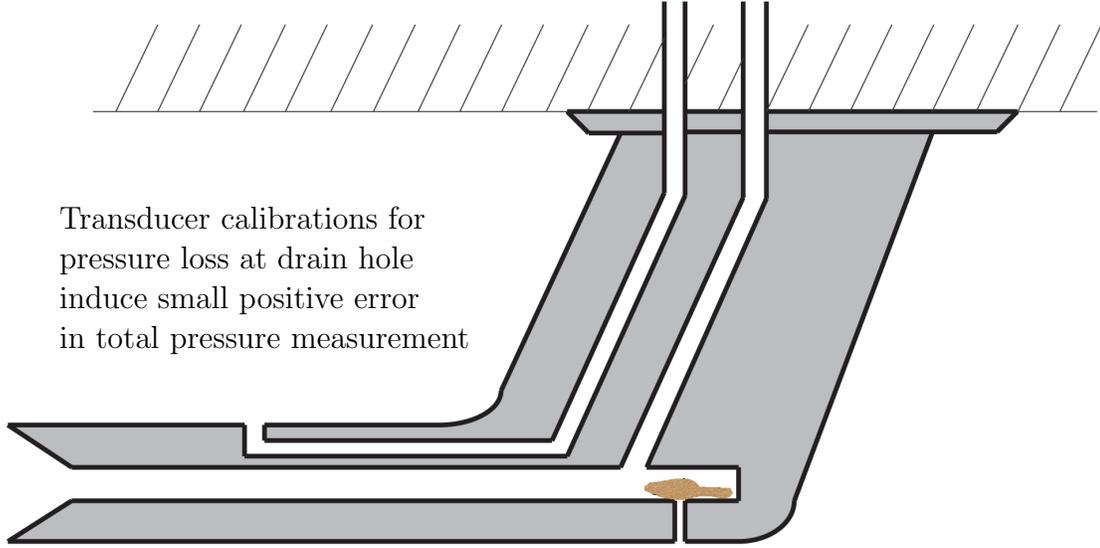


Figure 2.3: Pitot-static probe with drain hole blockage

tunnel dynamic pressure  $q_c$  as shown in Equations 2.4-2.5.

$$p_{s_{ND}} = \frac{(p_{s_{probe}} - p_{s_{ref}})}{q_c} \quad (2.4)$$

$$p_{t_{ND}} = \frac{(p_{t_{probe}} - p_{t_{ref}})}{q_c} \quad (2.5)$$

Nondimensionalizing in this way allows for an understanding of the quality of a set of air data measurements that spans a broad flight regime.

Examining Figures 2.5-2.11, it is clear that a drain hole blockage has little to no impact on the static pressure measurement indicated by the probe. This is true across different airspeeds and angles of attack. At negative angles of attack, however, the geometry of the probe is such that the static port captures a component of the dynamic pressure; this is reflected in elevated static pressure indicated by the probe relative to the tunnel reference static pressure. This effect diminishes as the angle of attack increases, and for  $\alpha \geq 0$ , this behavior is no longer evident in the experimental results.

As predicted, the plots also indicate a small false increase in the total pressure measured by the faulted probe. Since the static measurements in the nominal and faulted probes do not deviate, the difference in the total pressure measurements would, to

an aircraft avionics suite, indicate differing airspeeds. In this case, the faulted probe would indicate an erroneously high airspeed. This effect is fairly consistent throughout the envelope tested, but its magnitude is slightly diminished for near-zero angles of attack and is exacerbated for larger magnitude angles of attack.

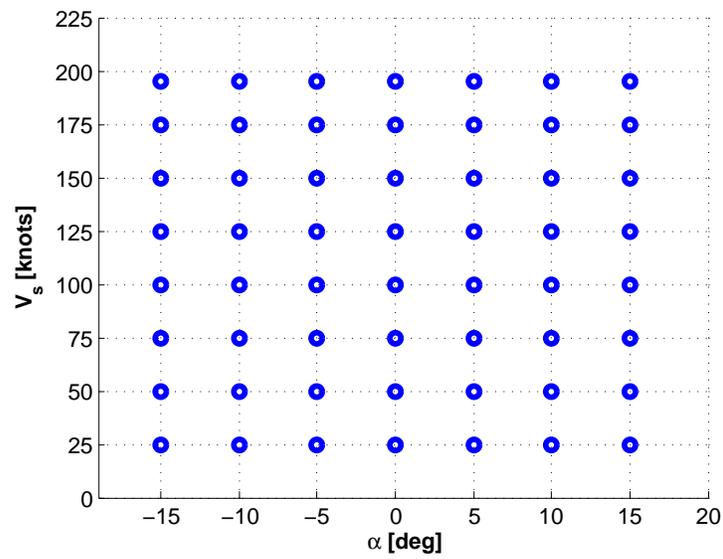


Figure 2.4: Flight envelope grid points for wind tunnel experiments

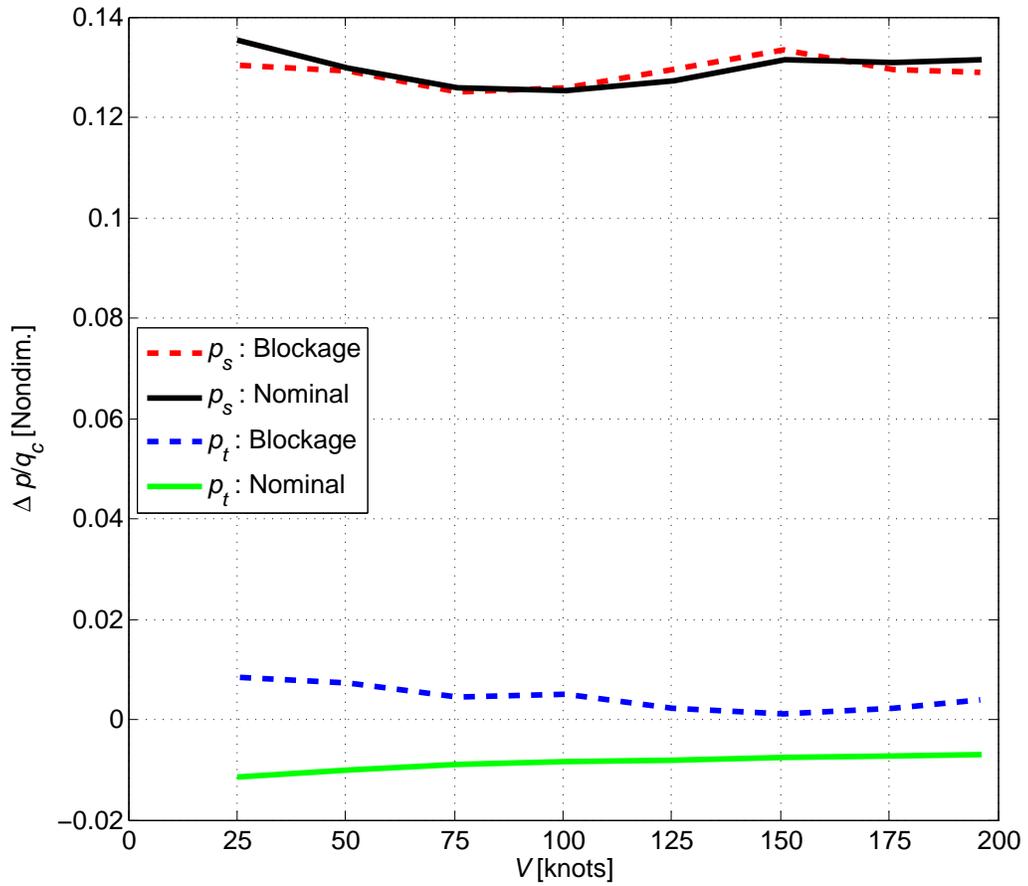


Figure 2.5: Comparison of nominal and faulted air data measurements for a drain hole blockage;  $\alpha = -15^\circ$

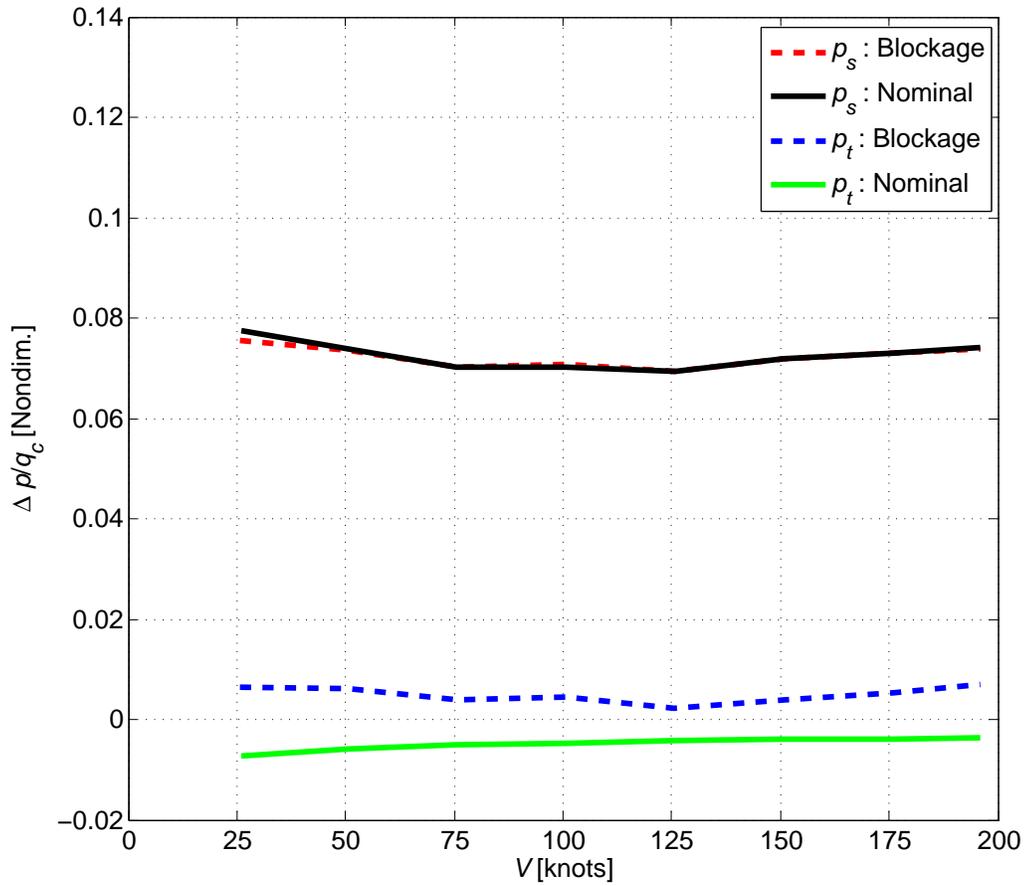


Figure 2.6: Comparison of nominal and faulted air data measurements for a drain hole blockage;  $\alpha = -10^\circ$

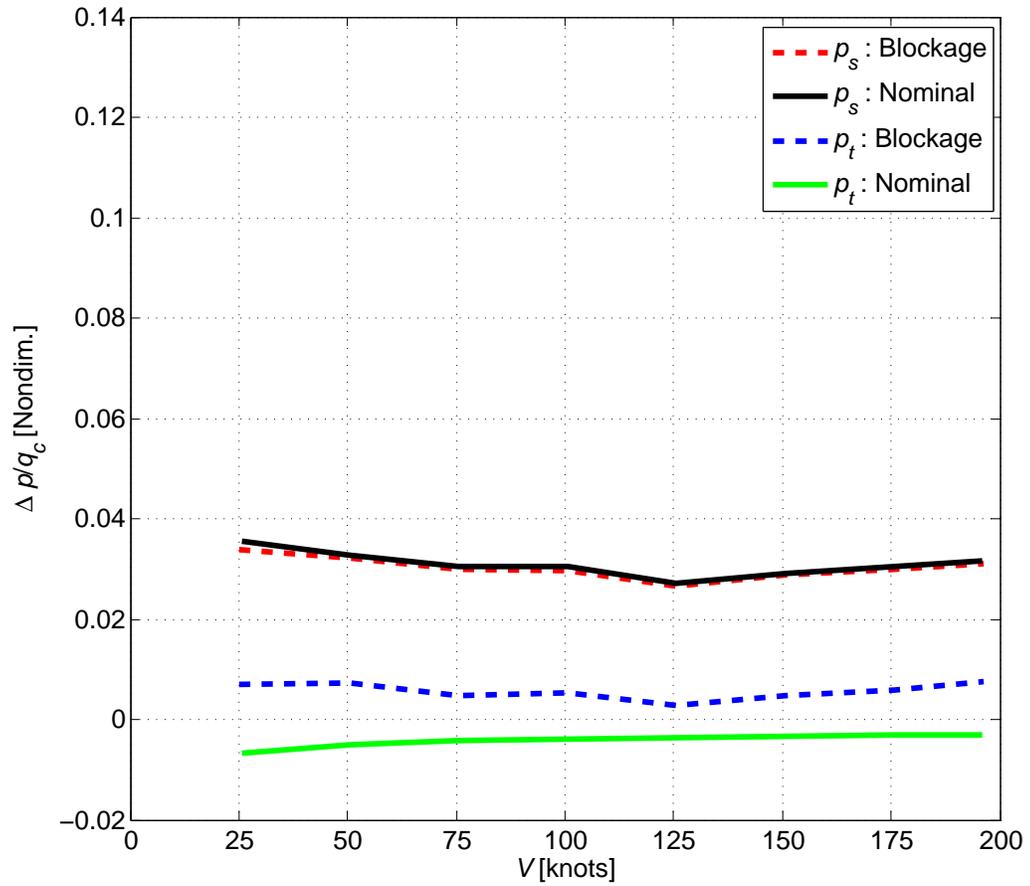


Figure 2.7: Comparison of nominal and faulted air data measurements for a drain hole blockage;  $\alpha = -5^\circ$

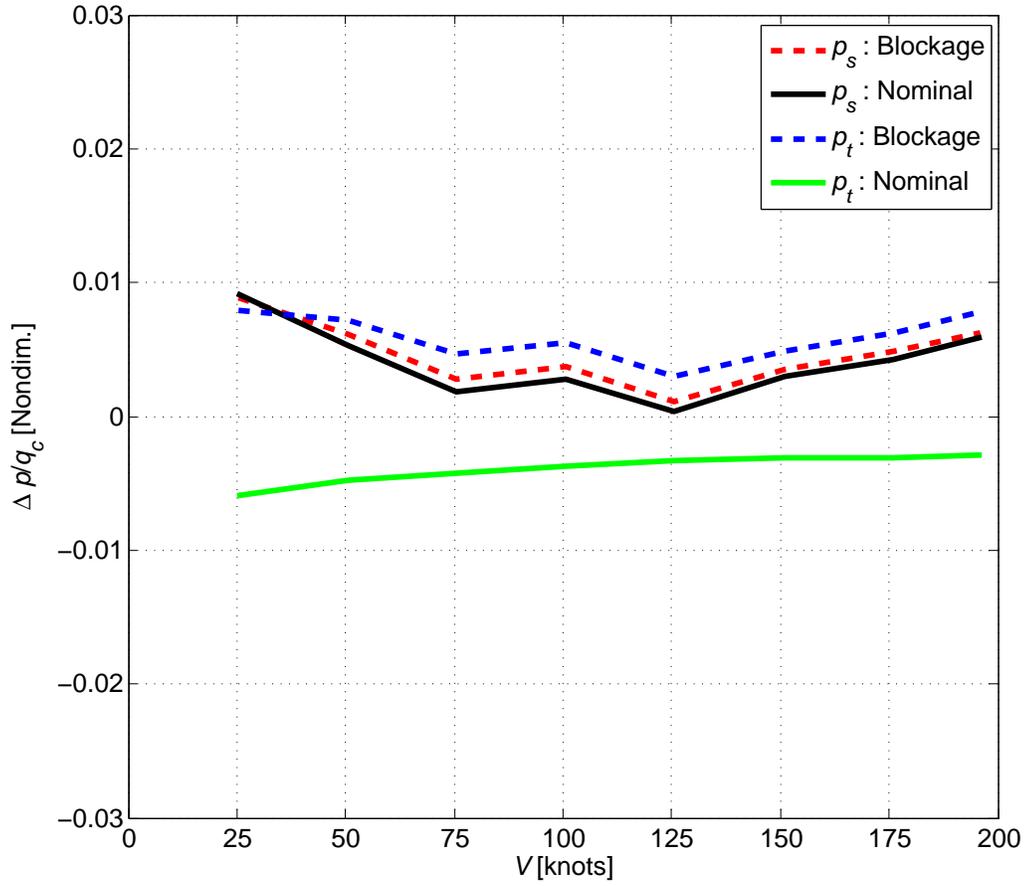


Figure 2.8: Comparison of nominal and faulted air data measurements for a drain hole blockage;  $\alpha = 0^\circ$ . Note the vertical axis change from the previous three figures.

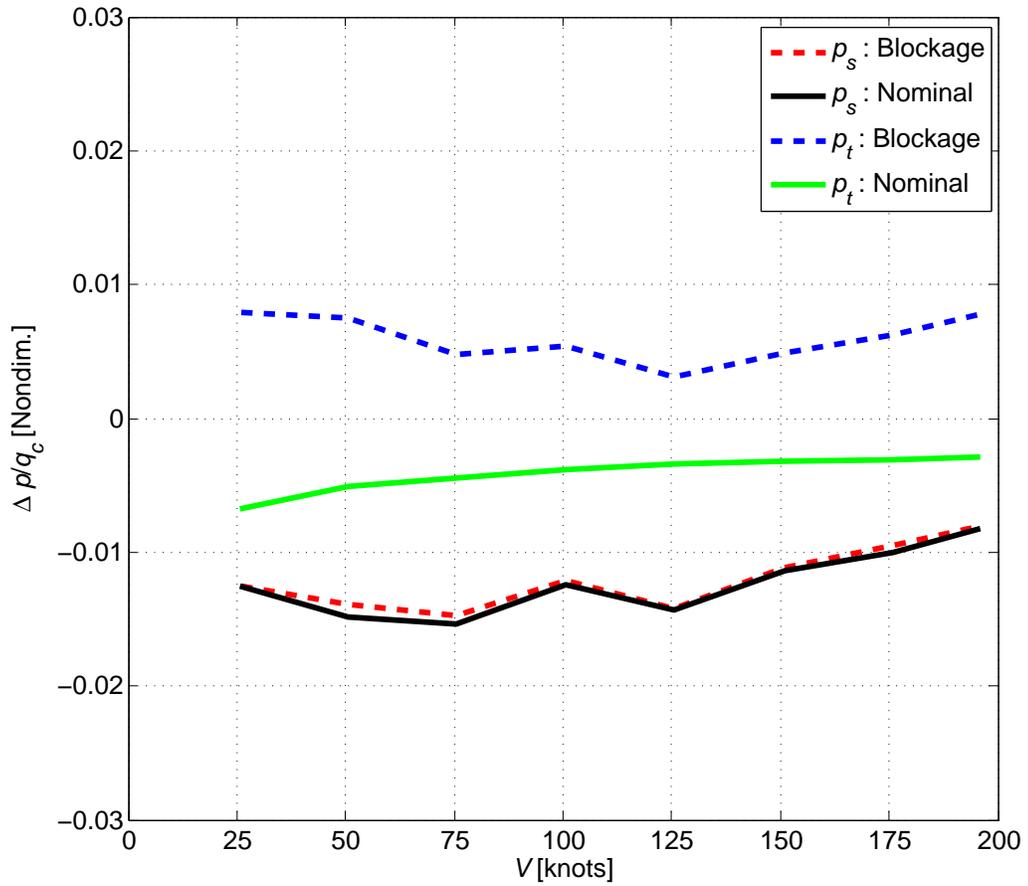


Figure 2.9: Comparison of nominal and faulted air data measurements for a drain hole blockage;  $\alpha = 5^\circ$

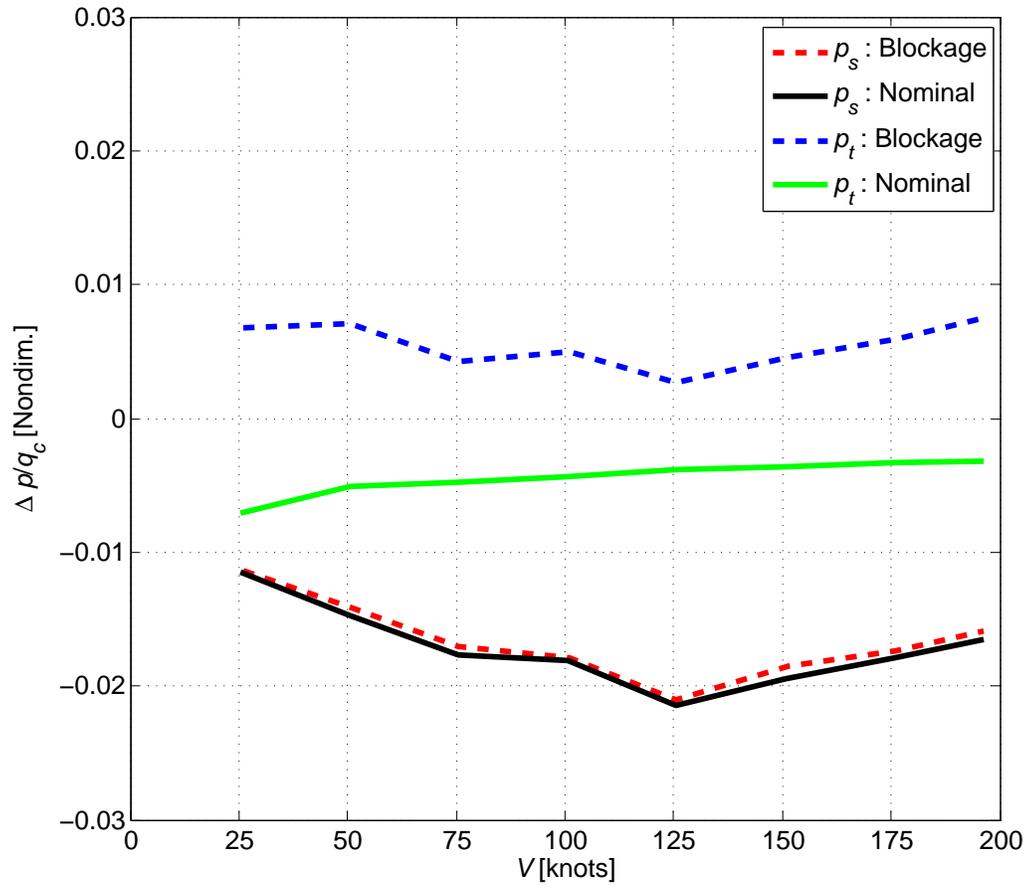


Figure 2.10: Comparison of nominal and faulted air data measurements for a drain hole blockage;  $\alpha = 10^\circ$

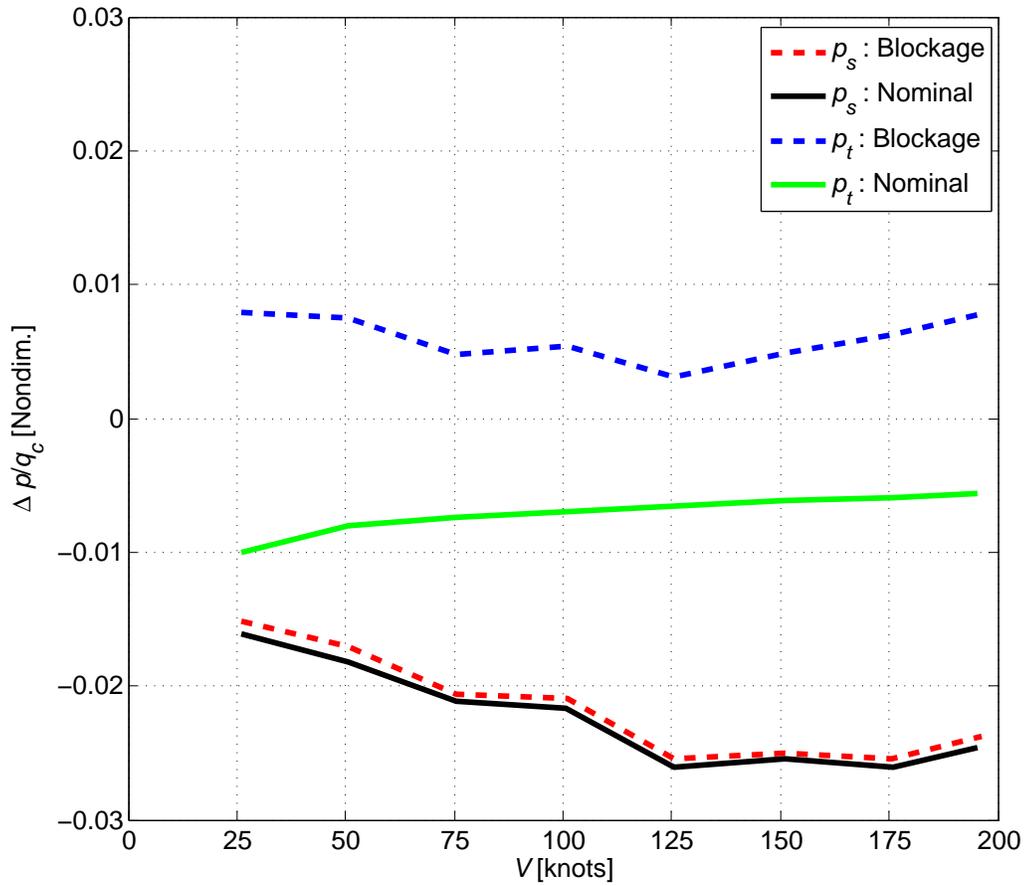


Figure 2.11: Comparison of nominal and faulted air data measurements for a drain hole blockage;  $\alpha = 15^\circ$

### 2.2.1.2 Pitot Inlet Blockage

With a blocked pitot inlet and an open drain hole (Figure 2.12), the drain hole becomes a static port. Therefore, the total pressure output will match the static pressure output. A pitot inlet blockage, however, is not evident at the beginning of a takeoff roll because the pilots and aircraft avionics are not anticipating a dynamic pressure indicative of a moving aircraft.

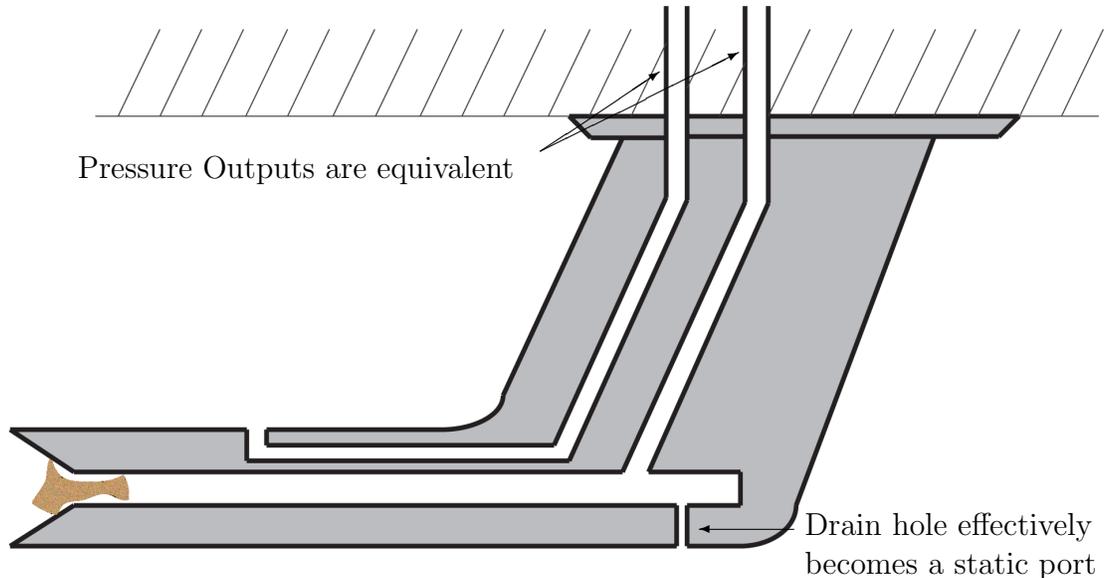


Figure 2.12: Pitot-static probe with pitot inlet blockage

### 2.2.1.3 Pitot Inlet and Drain Hole Blockages

A pitot inlet blockage combined with a drain hole blockage (Figure 2.13, presents a more complex set of effects. In this case, the pitot pressure channel becomes a closed system, and the pressure therefore remains constant during all aircraft maneuvers. The indicated airspeed will be misleading in different ways (depending upon the particular aircraft maneuver).

When the aircraft climbs, the static pressure correctly decreases while the dynamic pressure incorrectly increases. In a descent, the static pressure correctly increases while the dynamic pressure incorrectly decreases. For an aircraft increasing or decreasing in airspeed, the dynamic pressure incorrectly remains constant. Table 2.1 describes the effects of a drain hole blockage combined with a pitot inlet blockage on the airspeed and altitude measurements.

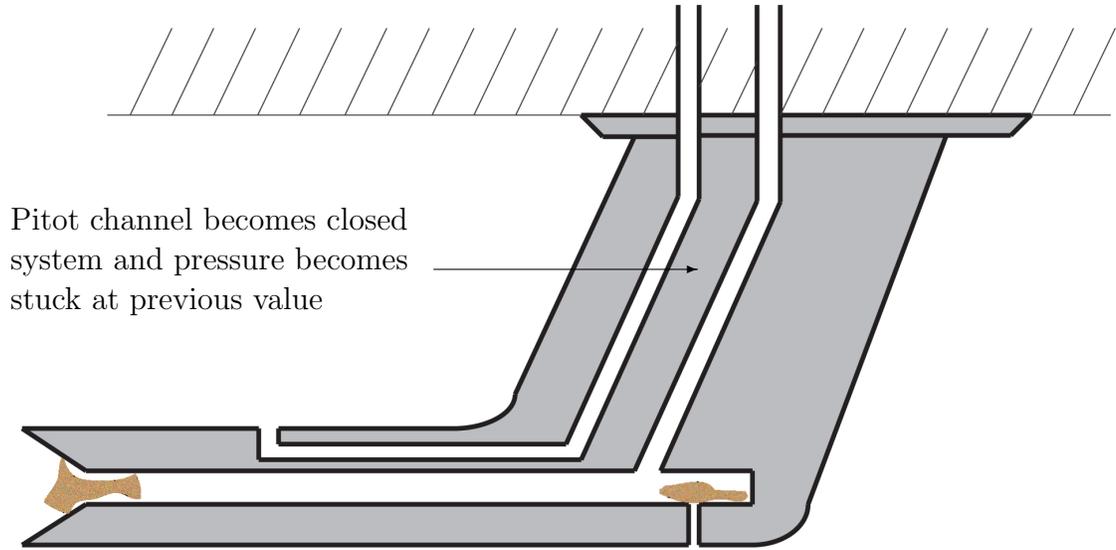


Figure 2.13: Pitot-static probe with combined pitot inlet and drain hole Blockage

#### 2.2.1.4 Static Port Blockage

A blocked static port (Figure 2.14) directly impacts the altitude measurement and adversely impacts the dynamic pressure (and thus indicated airspeed) in a manner similar to a pitot inlet blockage. When the aircraft climbs, the static pressure incorrectly remains constant while the dynamic pressure incorrectly decreases. In a descent, the static pressure incorrectly remains constant while the dynamic pressure incorrectly increases. When increasing or decreasing in airspeed while maintaining a particular altitude, the static and dynamic pressure are correct. Table 2.1 describes the effects of a static port blockage on the airspeed and altitude measurements.

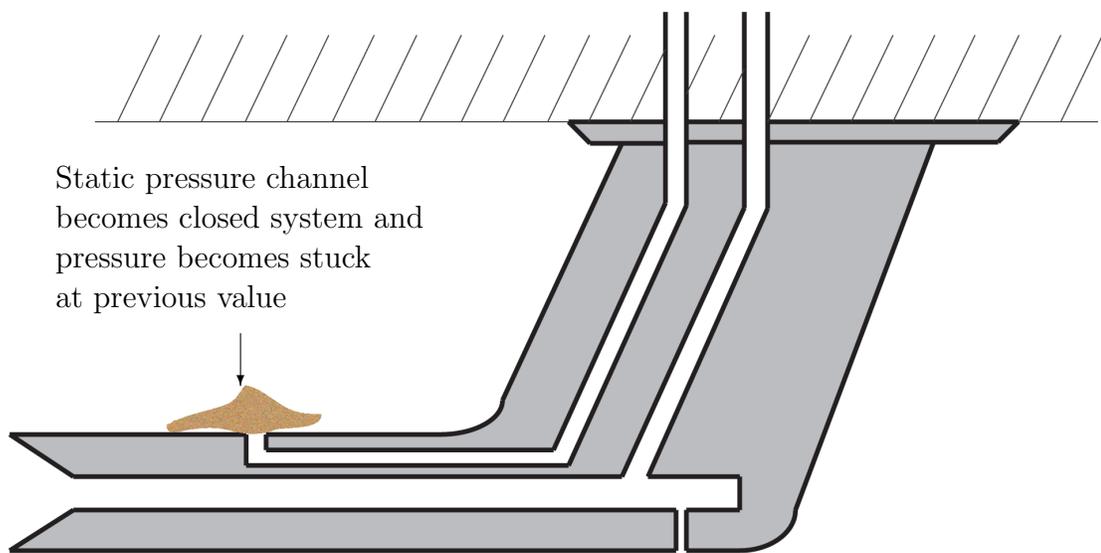


Figure 2.14: Pitot-static probe with static port blockage

### 2.2.1.5 Water in Probe or Pressure Lines

The presence of water within an air data probe or its attached pressure lines (Figure 2.15) can lead to significant fluctuations in pressure measurements. A meniscus formed within a pressure line will cause increased pneumatic lag due to the weight of the water present in the line. The magnitude of the pressure fluctuations and the associated pneumatic lag is influenced by the mass and location of the water in the line. Unlike with port blockages, potential impacts on the altitude and airspeed measurements are not as clearly defined.

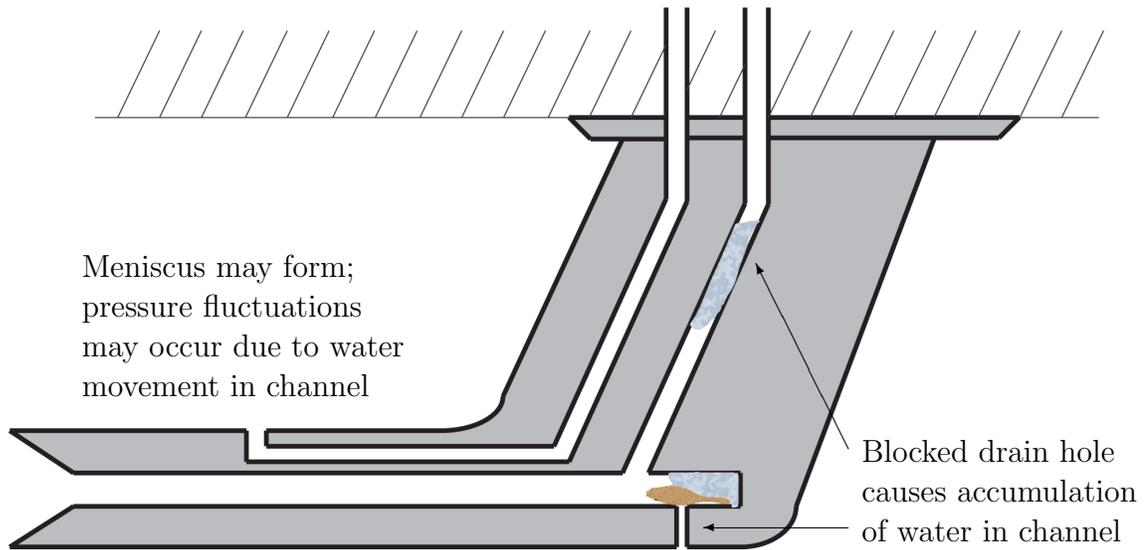


Figure 2.15: Pitot-static probe with water blockages

A summary of blockage fault effects is presented in Table 2.1. When a fault does not impact the validity of a particular measurement, the corresponding table entry indicates that the measurement is 'Correct.'

Table 2.1: Blockage fault effects for pitot-static probes

Fault	Aircraft Maneuver	$V_s$ Measurement	$h$ Measurement	Figure
Drain Hole Blockage	All	Small false increase in airspeed	Correct	Figure 2.3
Pitot Inlet Blockage	All	Airspeed drops to zero; Remains stuck at zero	Correct if only the pitot inlet affected. Incorrect if static measurement also affected (e.g. bird strike, protective cover, etc.)	Figure 2.12
Pitot Inlet and Drain Hole Blockage	Climb	False increase in airspeed	Correct	Figure 2.13
	Descent	False decrease in airspeed	Correct	
	Increasing airspeed	Airspeed falsely remains constant	Correct	
	Decreasing airspeed	Airspeed falsely remains constant	Correct	
Static Port Blockage	Climb	False decrease in airspeed	Altitude remains incorrectly constant. Altitude rate (Vertical Speed Indicator) remains stuck at zero	Figure 2.14
	Descent	False increase in airspeed		
	Increasing airspeed	Correct if altitude remains constant		
	Decreasing airspeed	Correct if altitude remains constant		
Water in Probe or Pressure Lines	All	Various	Various	Figure 2.15

#### 2.2.1.6 Pneumatic Line Leakage

Air pressures are conveyed to pressure transducers via pneumatic lines attached to the air data probe. Leakages in the pneumatic lines corrupt pressure measurements. The particular effects of such a fault depend upon which system is leaking (pitot or static), leakage severity, and the ambient pressure in the location of the leak. Leaks could occur in a pressurized cabin compartment or in an unpressurized bay; the ambient pressure in the location of the leak influences the magnitude and direction of any errors in the air data measurements.

### 2.2.2 Airflow Disruption Faults

Pitot-static probes are placed in locations on an aircraft fuselage which are expected to experience laminar airflow that is unlikely to be fouled in-flight by moving control surfaces or by the fuselage itself. While this provides for maximum sensor accuracy, airflow disruptions that do occur may still adversely impact measurements collected by the air data probes. The most common example of such airflow disruptions can occur with aircraft icing. Icing-related blockage faults internal to pitot-static probes were previously discussed; ice accretion on the fuselage or the exterior of the probe can significantly disrupt airflow in the vicinity of the static ports, introducing errors into the static pressure measurement. In such a case, the pitot pressure measurement is unlikely to be significantly impacted.

### 2.2.3 Heater Faults

Pitot-static probes contain heaters embedded within their surfaces which discourage ice accretion on the surface of the probe and melt ice crystals which may have entered the pitot inlet, allowing for proper drainage and preventing blockages. Heater elements can completely fail (in which they short circuit and no longer provide any heat) or experience slow-acting faults such as resistance drift. Changing resistances of the heater element cause fluctuations in heat output.

### 2.2.4 Faults Studied and Future Modeling Work

Because blockage faults in air data probes are common, and since their effects are more easily modeled than the effects related to airflow disruption or heater faults, the focus of this thesis is blockage fault detection. Blockage faults tend to be fast-

acting and cannot necessarily be predicted via regular monitoring and maintenance (although consistent probe cleaning minimizes risk). Blockage faults also tend to exhibit steady-state behaviors – particularly for full blockages – yet there are instances where a partial blockage or the presence of water in the pressure lines could lead to time-varying adverse effects on the air data measurements. Faults relating to airflow disruptions (e.g. from icing) or heater faults involve more complex studies of icing aerodynamics and heater element construction and control. In order to investigate fault-tolerance approaches for these failure scenarios, higher fidelity models of the air data probes are necessary. One way to achieve a better understanding of time-varying behaviors would be to perform experiments using high-frequency pressure transducers. Most current air pressure transducers often operate at frequencies near 1 Hz, and data that could be used to describe time-varying effects may be lost at such low frequencies.

# Chapter 3

## Aircraft Model

The platform considered for this research is the NASA Generic Transport Model (GTM), a remote-controlled, 5.5 percent scale commercial aircraft [15]. The aircraft weighs approximately 50 lbs. and can reach airspeeds of up to 200 miles per hour. Because the aircraft is also dynamically-scaled, it is a useful platform for software and control systems design in NASA's Safety and Security Program. NASA constructed a high fidelity 6 degree-of-freedom Simulink model of the GTM [16] with the aerodynamic coefficients provided as look-up tables. The GTM Simulink model captures key flight dynamics characteristics of the GTM aircraft and the commercial transport aircraft it is designed after and provides a useful starting point for this analysis.

### 3.1 Generic Transport Model Longitudinal Dynamics

This section describes the longitudinal dynamics of the GTM aircraft. Important GTM aircraft geometric and mass parameters – as well as relevant constants – are provided in Table 3.1. Using this information, the nonlinear equations of motion can be written.

Table 3.1: Aircraft and environment parameters

Wing Area, $S$	5.902 ft <sup>2</sup>
Mean Aerodynamic Chord, $\bar{c}$	0.9153 ft
Mass, $m$	1.795 slugs
Pitch Axis Moment of Inertia, $I_{yy}$	4.655 slugs-ft <sup>2</sup>
Air Density, $\rho$	0.002377 slugs/ft <sup>3</sup>
Gravity Constant, $g$	32.17 ft/s <sup>2</sup>

The longitudinal dynamics of the GTM are described by a standard five-state longitudinal model [17]:

$$\dot{V} = \frac{1}{m} (-D - mg \sin(\theta - \alpha) + T_x \cos \alpha + T_z \sin \alpha) \quad (3.1)$$

$$\dot{\alpha} = \frac{1}{mV} (-L + mg \cos(\theta - \alpha) - T_x \sin \alpha + T_z \cos \alpha) + q \quad (3.2)$$

$$\dot{q} = \frac{(M + T_m)}{I_{yy}} \quad (3.3)$$

$$\dot{\theta} = q \quad (3.4)$$

$$\dot{h} = V \sin(\theta - \alpha) \quad (3.5)$$

where  $V$  is air speed (knots),  $\alpha$  is angle of attack (deg),  $q$  is pitch rate (deg/s),  $\theta$  is pitch angle (deg), and  $h$  is altitude (ft). The control inputs are the elevator deflection  $\delta_{elev}$  (deg) and engine throttle  $\delta_{th}$  (percent).

The drag force  $D$  (lbs), lift force  $L$  (lbs), and aerodynamic pitching moment  $M$  (lb-ft) are given by:

$$D = \bar{q} S C_D(\alpha, \delta_{elev}, \hat{q}) \quad (3.6)$$

$$L = \bar{q} S C_L(\alpha, \delta_{elev}, \hat{q}) \quad (3.7)$$

$$M = \bar{q} S \bar{c} C_m(\alpha, \delta_{elev}, \hat{q}) \quad (3.8)$$

where  $\bar{q} := \frac{1}{2} \rho V^2$  is the dynamic pressure (lbs/ft<sup>2</sup>) and  $\hat{q} := \frac{\bar{c}}{2V} q$  is the normalized pitch rate (unitless).  $C_D$ ,  $C_L$ , and  $C_m$  are unitless aerodynamic coefficients computed from look-up tables provided by NASA. NASA provided raw look-up table data for the aerodynamic coefficients in the airframe body axes, i.e. the raw data is provided for  $C_x$ ,  $C_z$ , and  $C_m$ . This notation refers to standard aircraft body axis conventions [17].  $x$  is directed to the front along the longitudinal axis of the aircraft and  $z$  is directed down.  $X$  and  $Z$  represent the aerodynamic forces along the  $x$  and  $z$  axes,

respectively. The body-axis look-up tables  $C_X$  and  $C_Z$  were transformed into lift and drag coordinates via a rotation of angle  $\alpha$  about the lateral axis.

The GTM has two engines, one on the port side and the other on the starboard side of the airframe. Equal thrust settings for both engines is assumed. The thrust from a single engine  $T$  (lbs) is a function of the throttle setting  $\delta_{th}$  (percent).  $T(\delta_{th})$  is specified as a ninth-order polynomial in NASA's high fidelity GTM simulation model.  $T_x$  (lbs) and  $T_z$  (lbs) denote the projection of the total engine thrust along the body x-axis and body-z axis, respectively.  $T_m$  (lbs-ft) denotes the pitching moment due to both engines.  $T_x$ ,  $T_z$  and  $T_m$  are given by:

$$T_x(\delta_{th}) = n_{ENG}T(\delta_{th}) \cos(\epsilon_2) \cos(\epsilon_3) \quad (3.9)$$

$$T_z(\delta_{th}) = n_{ENG}T(\delta_{th}) \sin(\epsilon_2) \cos(\epsilon_3) \quad (3.10)$$

$$T_m(\delta_{th}) = r_z T_x(\delta_{th}) - r_x T_z(\delta_{th}) \quad (3.11)$$

$n_{ENG} = 2$  is the number of engines.  $\epsilon_2 = 1.98$  deg and  $\epsilon_3 = 2.23$  deg are angles that specify the rotation from engine axes to the airplane body axes.  $r_x = 0.4223$  ft and  $r_z = 0.3336$  ft specify the moment arm of the thrust.

## 3.2 Actuator Dynamics

The actuator dynamics are modeled as linear systems. The elevator actuator for the longitudinal GTM has a 5Hz bandwidth, first-order system with a 10 ms delay and is shown in Equation 3.12. The engine dynamics are modeled as the second order system shown in Equation 3.13.

$$Act_{elev} = e^{-0.01s} \frac{31.42}{s + 31.42} \quad (3.12)$$

$$Act_{th} = \frac{-0.147s + 0.731}{s^2 + 1.36s + 0.731} \quad (3.13)$$

### 3.3 Aircraft Trim and Model Linearization

A steady, level reference flight condition is chosen within the GTM flight envelope. The GTM, including actuator dynamics, is trimmed at the following condition:

$$\bar{\mathbf{x}} = \begin{bmatrix} V \\ \alpha \\ q \\ \theta \\ h \end{bmatrix} = \begin{bmatrix} 75 \text{ knots} \\ 5.63 \text{ deg} \\ 0 \text{ deg/s} \\ 5.63 \text{ deg} \\ 500 \text{ ft} \end{bmatrix}, \quad \bar{\mathbf{u}} = \begin{bmatrix} \delta_{th} \\ \delta_{elev} \end{bmatrix} = \begin{bmatrix} 33.098 \% \\ 0.072 \text{ deg} \end{bmatrix} \quad (3.14)$$

The nonlinear GTM dynamics are linearized about this trim condition to yield a 8-state system  $G$  with two inputs and five outputs (Equations 3.15 and 3.16).

$$\begin{aligned} \dot{\mathbf{x}} &= A_G \mathbf{x} + B_G \mathbf{u} \\ \mathbf{y} &= C_G \mathbf{x} + D_G \mathbf{u} \end{aligned} \quad (3.15)$$

where

$$A_G = \begin{bmatrix} -0.0724 & -0.1509 & 0.0022 & -0.3327 & 0 & -0.0128 & 0.0635 & -0.2007 \\ -0.3848 & -2.3677 & 0.9428 & 0.0000 & 0 & 0.0006 & -0.0031 & -1.8647 \\ -0.8623 & -39.4439 & -3.2854 & 0 & 0 & -0.1525 & 0.7569 & -310.2071 \\ 0 & 0 & 1.0000 & 0 & 0 & 0 & 0 & 0 \\ -0.0000 & -1.3090 & 0 & 1.3090 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1.3360 & -0.7314 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -31.4159 \end{bmatrix}$$

$$B_G = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 4 \end{bmatrix}, \quad C_G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \quad D_G = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (3.16)$$

This resulting linear model is used for control law development, initial filter synthesis, and simulation.

### 3.4 Control Law

The GTM longitudinal axis flight control law is an airspeed/altitude hold autopilot designed using a combination of classical loop-at-a-time and  $H_\infty$  techniques. The design allows for fault detection simulation and analysis while holding the GTM at a cruise condition or performing simple longitudinal maneuvers. These control laws serve merely to provide a closed-loop aircraft model that approximates the flight characteristics of a true aircraft for the purposes of simulation. The full control law interconnection is shown in Figure 3.1.

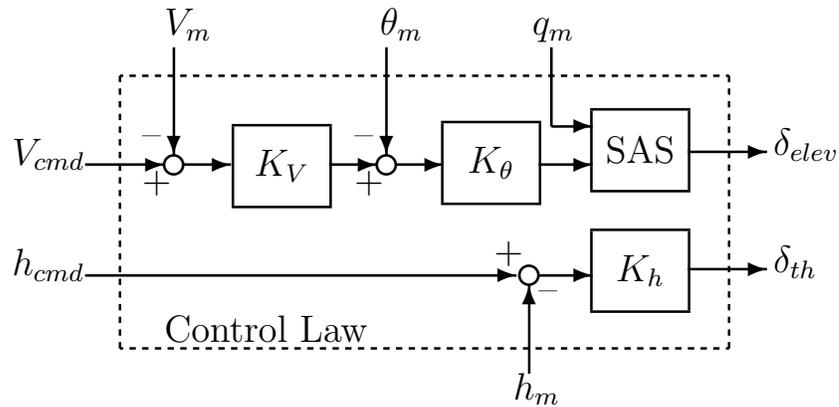


Figure 3.1: Autopilot control law architecture

### 3.4.1 Airspeed Hold Autopilot

The Airspeed Hold Autopilot consists of three sequentially closed loops: a stability augmentation system (SAS), a pitch angle (deg) tracking controller, and finally an airspeed (knots) tracking controller. Together, these three loops generate the appropriate elevator deflection command (deg).

The innermost controller is a SAS that increases damping in the longitudinal axis oscillatory modes. It takes measured pitch rate and pitch angle commands as measurements and has elevator deflection as its output. The SAS is designed using the  $H_\infty$  framework for robust control. An ideal model with additional damping is defined in the frequency domain using the GTM transfer function from elevator input to pitch rate. This model has the same low-frequency characteristics as the GTM in open-loop. A model-matching problem is formulated, and a controller is synthesized such that the closed-loop GTM dynamics match those of the ideal model. The resulting inner loop behaves like the open-loop aircraft: with significantly damped phugoid and short period oscillations. [18]

Unfortunately, the SAS formulated in this manner is a 15-state system; such a large system adds too much complexity to the overall aircraft control law. A reduced-order system can be fit to match the frequency response of the full system in order to streamline the control law. In this case, a five state model is found to exhibit a nearly identical frequency response up to frequencies exceeding 10 rad/s. At higher frequencies, the original SAS provides higher signal attenuation than the reduced version and a phase discrepancy arises. At such frequencies, however, the dynamics of the GTM roll off, and the impact of this discrepancy is minimized. A frequency response comparison of the full-state and reduced-order SAS is shown in Figure 3.2, and the state-space matrices for the reduced-order SAS are shown in Equation 3.17. The reduced-state stability augmentation is suitable to use given its equivalent system response within the important range of frequencies.

$$A_{SAS} = \begin{bmatrix} -20.1728 & 21.1728 & -21.1728 & 21.1728 & -10.5864 \\ 28.9184 & -29.9184 & 30.9184 & -30.9184 & 15.4592 \\ -7.9238 & 7.9238 & -8.9238 & 9.9238 & -4.9619 \\ -6.0079 & 6.0079 & -6.0079 & 5.0079 & -2.0039 \\ 6.2792 & -6.2792 & 6.2792 & -6.2792 & 2.1396 \end{bmatrix},$$

$$B_{SAS} = \begin{bmatrix} 0.9729 & 0.3040 \\ -1.2429 & -0.2936 \\ -0.2833 & 0.0736 \\ 1.1815 & 0.0534 \\ -1.0579 & -0.0323 \end{bmatrix}, \quad C_{SAS} = \begin{bmatrix} 14.9714 \\ -14.9714 \\ 14.9714 \\ -14.9714 \\ 7.4857 \end{bmatrix}^\top, \quad D_{SAS} = \begin{bmatrix} -0.0156 \\ -0.0354 \end{bmatrix}^\top \quad (3.17)$$

A pitch angle tracker  $K_\theta$  is closed around the inner-loop SAS. It is designed using proportional-integral control and pitch angle measurement (deg) feedback to provide errorless steady-state pitch command tracking. The controller has a step response rise time of 0.785 s and 15.5% overshoot. The coefficients of the transfer function are negative due to the sign convention for elevator deflection. The controller is given by:

$$K_\theta = -0.60 - \frac{0.72}{s} \quad (3.18)$$

Time-domain parameters for the closed-loop step response from  $\theta_{cmd}$  to  $\theta$ , as well as stability margins, are presented in Table 3.2.

Table 3.2: Pitch-angle tracker parameters

Rise Time	0.785 s
Overshoot	15.5 %
Settling Time	34.2 s
Steady-state Error	0
Bandwidth	1.90 rad/s
Gain Margin	7.76 dB @ 7.11 rad/s
Phase Margin	51.6 deg @ 1.44 rad/s

The airspeed tracker  $K_V$  is closed around the pitch-angle tracking and SAS loops. It is designed using proportional-integral control and airspeed measurement (knots) feedback. The sign convention for the elevator deflection again requires that the coefficients of the transfer function are negative. The airspeed tracker has a step response rise time of 1.74 s and 13.0% overshoot, and is given by:

$$K_V = -1.44 - \frac{0.30}{s} \quad (3.19)$$

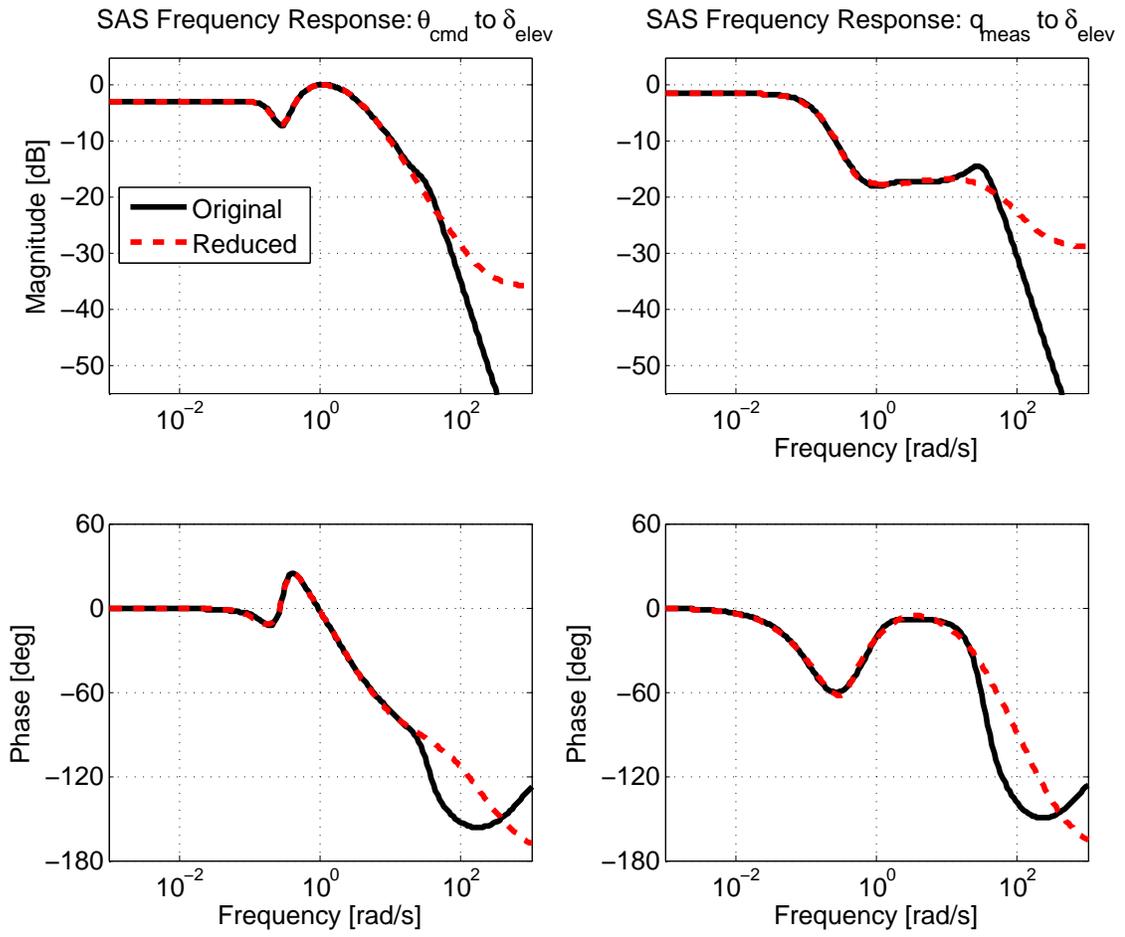


Figure 3.2: Comparison of SAS frequency responses

Time-domain parameters for the closed-loop step response from  $V_{cmd}$  to  $V$ , as well as stability margins, are presented in Table 3.3.

Table 3.3: Airspeed tracker parameters

Rise Time	1.74 s
Overshoot	13 %
Settling Time	16.6 s
Steady-state Error	0
Bandwidth	0.88 rad/s
Gain Margin	13.9 dB @ 2.18 rad/s
Phase Margin	66.6 deg @ 0.561 rad/s

### 3.4.2 Altitude Hold Autopilot

The altitude hold autopilot consists of a single loop utilizing altitude measurement (ft) feedback and lead compensation with low-pass filtering. The controller  $K_h$  generates the appropriate throttle command (percent), and it has a step response rise time of 5.94 s and 18.3% overshoot. The controller is given by:

$$K_h = \frac{0.088s + 0.44}{0.017s^2 + 0.32s + 1} \quad (3.20)$$

Time-domain specifications for the closed-loop step response from  $h_{cmd}$  to  $h$ , as well as stability margins, are presented in Table 3.4.

Table 3.4: Altitude tracker specifications

Rise Time	5.94 s
Overshoot	18.3 %
Settling Time	41.7 s
Steady-state Error	0
Bandwidth	0.24 rad/s
Gain Margin	7.76 dB @ 2.18 rad/s
Phase Margin	66.6 deg @ 0.561 rad/s

## 3.5 Inertial Sensors

Sensors for angle of attack, pitch rate, and pitch angle are modeled as unity with additive white noise and bias on the true states. Sensor dynamics are neglected in

the model. The sensor noise parameters are derived from sensor data on the NASA GTM T2 aircraft and are presented in Table 3.5 [16]. Equations 3.21-3.23 are used

Table 3.5: Inertial sensor parameters

Measurement Noise	$n_\alpha$	$n_q$	$n_\theta$
Standard Deviation	0.031 deg	0.4737 deg/s	0.05 deg
Bias	0 deg	-0.0115 deg/s	0 deg
Scale Factor	1	1	1

to represent the state measurements to be fed back to the autopilot controllers and fault detection filters:

$$\alpha_m = \alpha + n_\alpha \tag{3.21}$$

$$q_m = q + n_q \tag{3.22}$$

$$\theta_m = \theta + n_\theta \tag{3.23}$$

# Chapter 4

## Fault Detection: $H_\infty$ -synthesis

The  $H_\infty$ -synthesis framework is used to design filters to estimate disturbances, e.g. faults, at the plant input.  $H_\infty$  methods offer a number of advantages over traditional Kalman filtering, including superior performance in the presence of model uncertainty and the ability to filter process noise and exogenous disturbances without necessarily having a statistical model of those inputs [19].

### 4.1 $H_\infty$ Problem Formulation

The  $H_\infty$  filtering problem is formulated as a variant of a standard  $H_\infty$  optimal control problem. The principles used to design controllers can be used to develop a filter that estimates parameters of interest – in this case, pressure measurement faults.

#### 4.1.1 $H_\infty$ Control Formulation

$H_\infty$  optimal control methods are used to synthesize controllers that can achieve robust performance, mitigating the impact of exogenous disturbances to the system. The standard  $H_\infty$  control formulation, shown in Figure 4.1, consists of a plant  $P$  and controller  $K$ , with signal  $u$  containing the control variables,  $y$  the measured variables,  $w$  the exogenous signals such as reference commands and disturbances, and  $z$  the error signals.  $H_\infty$  methods provide a tool to minimize the error signals in a way that achieves the desired control objectives.

The closed-loop system can be represented with a partitioned plant as shown in

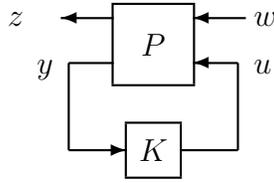


Figure 4.1: Standard system configuration for  $H_\infty$  optimal control design

Equations 4.1-4.2.

$$\begin{bmatrix} z \\ y \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \begin{bmatrix} w \\ u \end{bmatrix} \quad (4.1)$$

$$u = Ky \quad (4.2)$$

In this general formulation, the closed-loop transfer function from the exogenous signals  $w$  to the errors  $z$  can be represented by the linear fractional transformation

$$F_L(P, K) = P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21} \quad (4.3)$$

Minimizing the infinity norm of this LFT minimizes the gain across all frequencies from exogenous inputs to errors. Thus, for a given  $\gamma$ , an  $H_\infty$  algorithm will generate a stabilizing controller for which  $\| F_L(P, K) \|_\infty < \gamma$ . The value of  $\gamma$  is reduced iteratively until a controller can no longer be found at a  $\gamma$ -value sufficiently less than the value of the previously successful iteration. [20] [21]

#### 4.1.2 $H_\infty$ Filter Formulation

Using the framework from the  $H_\infty$  control problem, a filtering problem can be constructed. A  $H_\infty$  filter is synthesized from the linearized GTM model to estimate faults associated with the static and total pressure measurements. Unmodeled dynamics and model uncertainty are first neglected for this analysis, but the framework is easily extended later to include uncertainty. Those considerations will be important as the fault detection algorithm is refined in the next chapter.

The linearized aircraft dynamics are connected with the autopilot (Figure 3.1), inertial sensor models, and air data sensor architecture (Figure 2.2) to form the generalized plant. The interconnection for this generalized plant,  $genGTM$ , is shown

in Figure 4.2 and has the following inputs: the autopilot reference signals  $\tilde{\mathbf{r}} = [V_{cmd} \ h_{cmd}]^\top$ , the inertial measurement noises,  $\tilde{\mathbf{n}} = [n_\alpha \ n_q \ n_\theta]^\top$ , and the injected pitot faults  $\tilde{\mathbf{f}} = [f_{ps} \ f_{pt}]^\top$ . The errors  $\tilde{\mathbf{e}}$  are the difference between the injected faults and estimated faults  $\hat{\mathbf{f}} = [\hat{f}_{ps} \ \hat{f}_{pt}]^\top$ . The generalized GTM plant has measurement outputs  $\mathbf{y} = [p_{sm} \ p_{tm} \ \alpha_m \ q_m \ \theta_m]^\top$ . These are the measurements that will be available to the fault detection filter.

The objective of the  $H_\infty$  filter synthesis is to generate a stable filter  $F$  which minimizes norm between the disturbances and the errors. Because this formulation has a generalized plant that is already closed-loop, the filter can take advantage of the expected closed-loop system dynamics to generate fault estimates. These estimates are more accurate than estimates that would result from the common open-loop synthesis approach that fails to model the dynamics associated with the expected operation of a controlled system in the field. Note that the pitot noise enters the system in the same manner as pitot faults (see Figure 2.2). As a result,  $F$  tracks the sensor noise as well as the faults. Algorithms for processing the estimated fault information generated by  $F$  can compensate for this noise.

Weighting functions are used to describe the expected frequency content of the inputs and the desired frequency content of the errors, the normalized inputs  $[\tilde{\mathbf{r}} \ \tilde{\mathbf{n}} \ \tilde{\mathbf{f}}]^\top$  and outputs  $\tilde{\mathbf{e}}$ . Figure 4.3 shows the desired interconnection of the filter with the generalized plant *genGTM* with signal weights and filter  $F$ . Input and output signals with tildes represent their respective normalized signals in physical units.

For fault detection, the disturbances are the autopilot reference signals  $\mathbf{r}$  and the inertial measurement noises  $\mathbf{n}$ . The filter seeks to track the injected faults  $\mathbf{f}$  with the fault estimates  $\hat{\mathbf{f}}$  while rejecting inertial measurement noise  $\mathbf{n}$  and reference commands  $\mathbf{r}$ . Similar  $H_2$  and  $H_\infty$  model matching approaches to FDI filter design have been applied in [22–27].

## 4.2 Signal Weighting Methodology

In signal-based  $H_\infty$  control design, the size of signals entering and exiting the system – across all frequencies – becomes the primary focus. Different signals can inherently have different magnitudes depending upon the physical parameter represented. Sometimes the magnitude of a signal is constant across all frequencies, and sometimes it is highly frequency dependent. In order to account for these inherent differences

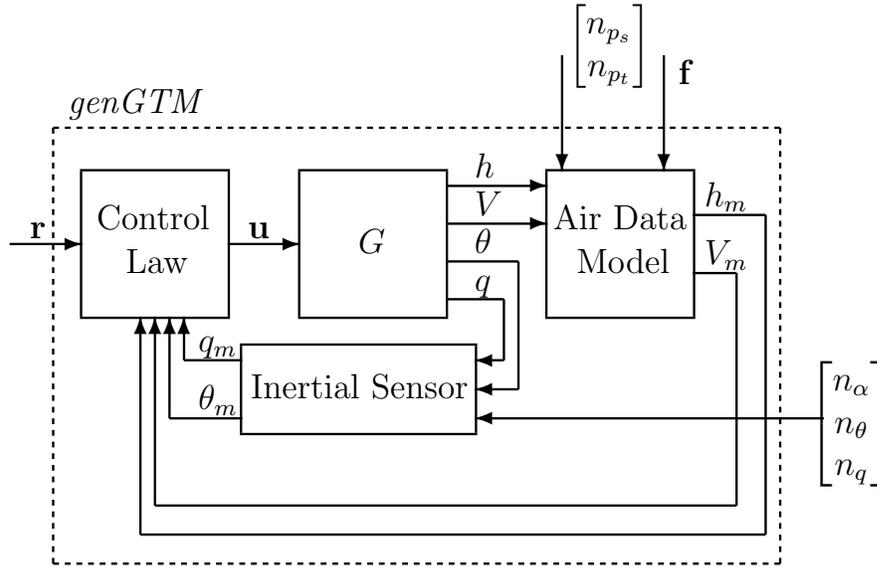


Figure 4.2: Interconnection for generalized plant  $genGTM$

in a way that makes the formulated problem amenable to optimization algorithms, the control designer must use weighting functions to describe the expected or known frequency content of exogenous signals. Additionally, the designer may desire error signals to have particular frequency content; weighting the error signals introduces those specifications into the problem formulation. Weights are also used for uncertainty modeling in order to reflect model accuracy across frequencies; system models are often more or less accurate in particular frequency ranges, and it is prudent to introduce uncertainty wherever the model dynamics are less accurate.

#### 4.2.1 Simple Weighting Approach

Because the synthesized  $H_\infty$  filter will have states equal in number to the generalized plant for which it is optimized, adding complexity to the generalized plant rapidly increases the complexity of the combined plant-filter system. As a first check, it is desirable to ascertain how well faults can be estimated using constant weighting on the input signals. Because the weights should represent the relative size or importance of exogenous signals, knowledge of these signal sizes informs the initial choices. A dynamic weighting function on the filter output,  $W_{fhat}$ , is initially used as the sole performance tuning mechanism.

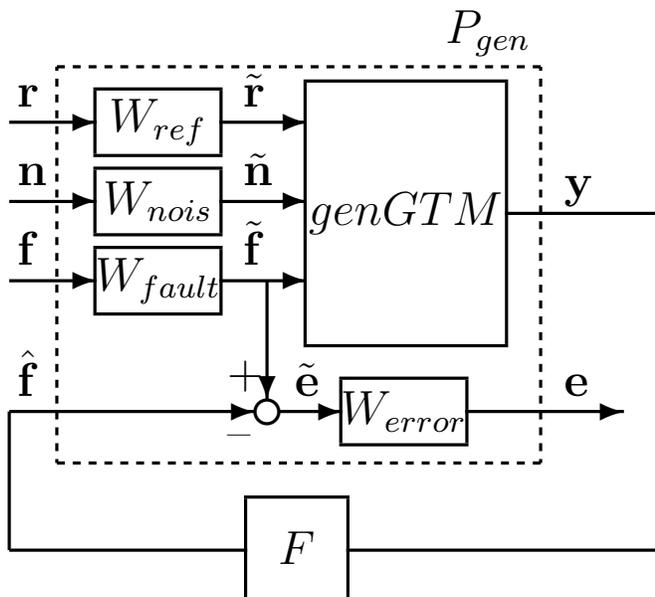


Figure 4.3: Interconnection for  $H_\infty$  filter synthesis

$W_{ref}$

The reference weighting function defines relative size of the autopilot reference commands. The airspeed reference command for the GTM trim condition (Equation 3.14) is 75 knots and the altitude reference command is 500 feet. Thus, we choose the weight to be:

$$W_{ref} = \begin{bmatrix} 1 & 0 \\ 0 & 6.6667 \end{bmatrix} \quad (4.4)$$

This ensures that the weighted generalized plant properly reflects the relationship between the two reference command channels.

$W_{nois}$

The standard deviations of the noise in each channel of the inertial sensor measurements are shown in Table 3.5. The entries in  $W_{nois}$  are selected to reflect those differing noise levels in each measurement.

$$W_{nois} = \begin{bmatrix} 0.031 & 0 & 0 \\ 0 & 0.4737 & 0 \\ 0 & 0 & 0.05 \end{bmatrix} \quad (4.5)$$

$W_{fault}$

Static and total pressures are measured in the same units with similar transducers; it is reasonable to assume that the maximum fault would be equivalent in each channel. Thus, the weighting function should reflect that. Section 2.1.3 explained that a 0.1 psi static pressure fault would be significant for a commercial aircraft (and an extremely large deviation for the low-flying GTM aircraft), so that value is selected.

$$W_{fault} = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} \quad (4.6)$$

$W_{fhat}$

A single dynamic weight on the filter output can be used to ensure that the filter does not respond too rapidly to faults when detected. Without a dynamic weight limiting the frequency of the response, the  $H_\infty$  algorithm will generate a filter  $F$  that is like a pseudoinverse of *genGTM*, responding almost instantaneously to an injected fault. Such a filter would be prohibitively sensitive if implemented on a real aircraft; thus, measures must be taken to prevent such a filter from generated.

A fault estimate weight is chosen that provides almost no penalty to low frequency fault estimates but penalizes high frequency responses by rising sharply in magnitude above 4 radians per second as shown in Figure 4.4.

$$W_{fhat} = \frac{80(s + 357.5)}{s + 357.8} \quad (4.7)$$

### Simple Approach Results

To gauge the efficacy of the simple approach with constant weights on the inputs and a dynamic weight on the fault estimate penalizing high frequency responses, a filter is synthesized and simulated with the linear GTM model. The fault injected is a static pressure step of magnitude 0.1 psi at time  $t = 1$  second. The static pressure fault estimate should track the injected fault and the total pressure fault estimate should indicate that no fault is present. The results of the simulation are shown in Figure 4.5. The static fault estimate responds rapidly – but not too rapidly – with a rise time of approximately 1 second. There is a significant error, however, between the actual

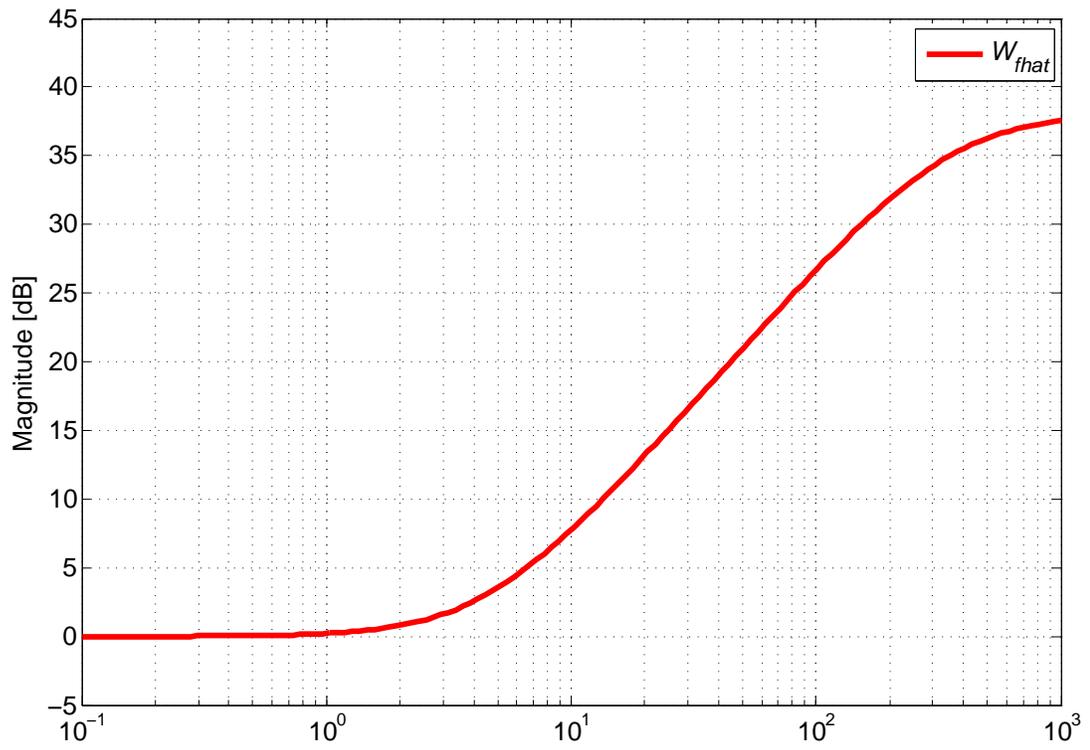


Figure 4.4:  $W_{\hat{f}}$ : magnitude frequency response

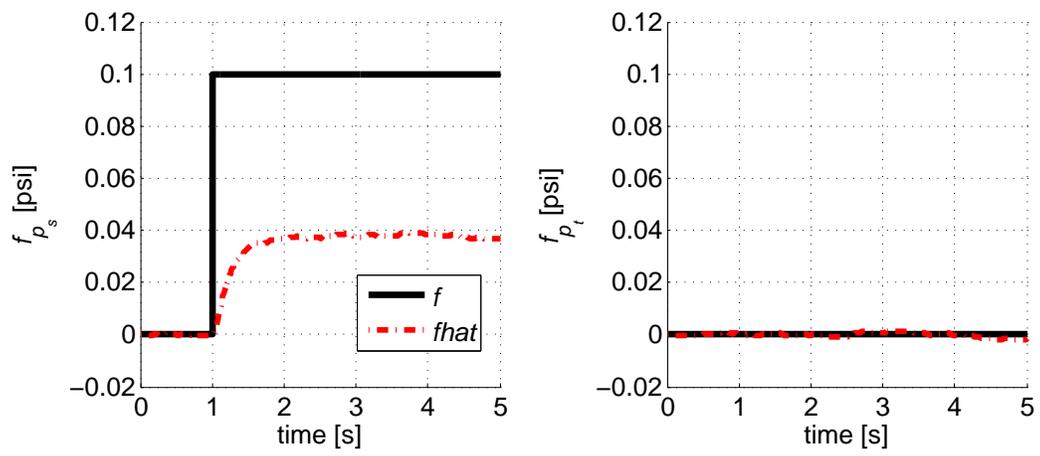


Figure 4.5: Fault estimates for 0.1 psi step fault in static pressure signal

fault and the estimated fault in the static channel. Such an error is unacceptably large ( 50%), and a more comprehensive weighting strategy is required. Notably, the simple approach correctly indicates that no fault is occurring in the total pressure channel.

#### 4.2.2 Multi-weight Approach

While the simple weighting approach shows some promise, a more comprehensive methodology may can achieve stronger results.  $W_{ref}$  does not need dynamic weighting, but the other exogenous signals have expected frequency content that can be included in the problem formulation by revising the weights.

$W_{nois}$

For mid-grade inertial sensors that may be utilized on a small UAV, the sensor noise magnitude is greater at high frequencies while the noise magnitude is reduced at low to intermediate frequencies.

$$W_{nois} = \frac{3.5s + 0.5248}{s + 349.9} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.8)$$

The magnitude frequency responses for  $W_{nois}$  are shown in Figure 4.6. The weighting on the inertial sensor noise is chosen iteratively so that the transfer function from  $\tilde{\mathbf{n}}$  to  $\hat{\mathbf{f}}$  on the unweighted *genGTM* interconnection with  $F$  has gain less than 1 for all frequencies. The weighting on the  $n_q$  signal is an order of magnitude larger than the other weightings due to the higher noise level on the pitch rate sensor output. The transfer functions from inertial sensor noise to pressure fault estimates show stronger attenuation in the  $\hat{f}_{p_s}$  channel; this fact is important for analysis of the filter performance.

$W_{fault}$

Faults are injected into the static and total pressure channels to corrupt the measurements. The fault weight, Equation 4.9, is chosen such that the DC gain represents large faults (-20 dB). The weight is small for frequencies greater than 5 rad/s to

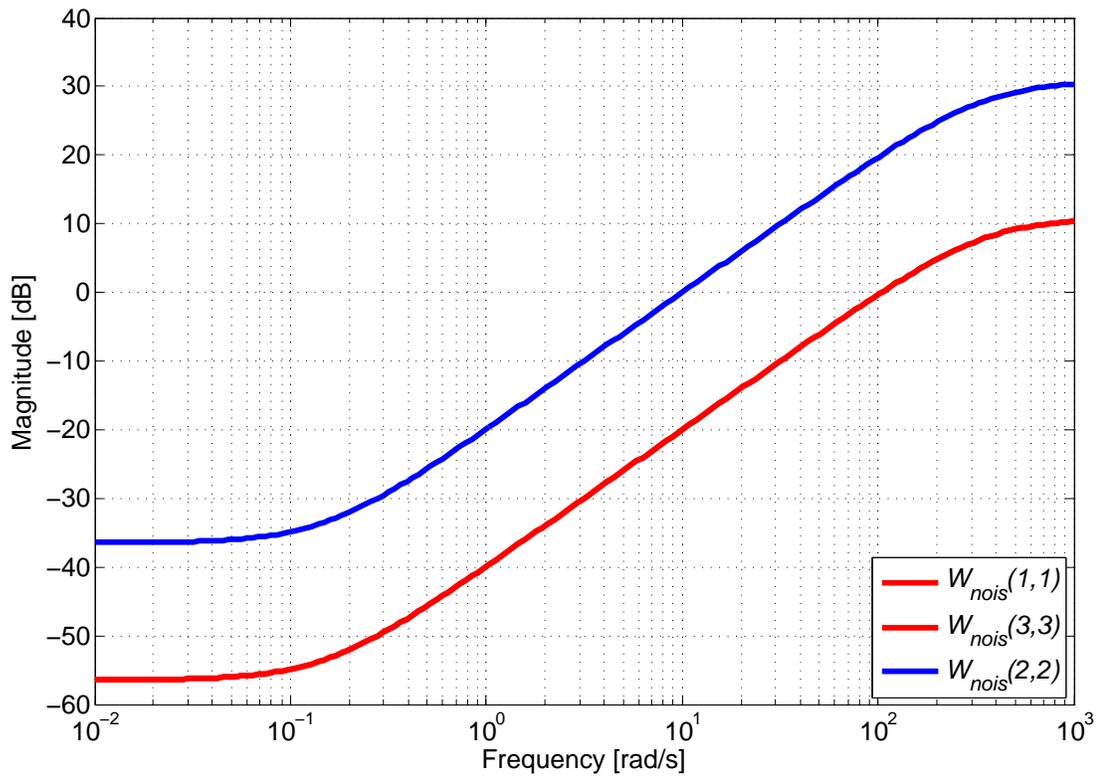


Figure 4.6:  $W_{nois}$ : magnitude frequency response

penalize tracking of high frequency faults. The aircraft dynamics roll off near this frequency; hence, the effect of higher frequency faults will not appear in the angle or rate measurements. This diminishes the tracking ability of  $F$  at high frequencies. The high frequency gain of  $W_{fault}$  is -60 dB. The magnitude frequency responses of the diagonal elements of  $W_{fault}$  are shown in Figure 4.7.

$$W_{fault} = \frac{0.001s + 3.562}{s + 35.27} I_2 \quad (4.9)$$

$W_{error}$

The error weight represents the inverse of the allowable tracking error at each frequency. Normally, the error weight would be large at low frequency to ensure close tracking. Tracking at high frequency is less desirable and error weightings will roll off to some small high frequency gain. The particular nature of this problem, however, is such that the usual error weighting methodology cannot be adopted for the generalized GTM filter synthesis. Equation 4.10 shows the DC gain of  $P_{\tilde{\mathbf{f}}\tilde{\mathbf{y}}}$ , the partition of *genGTM* from  $\tilde{\mathbf{f}}$  to  $\tilde{\mathbf{y}}$ .

$$P_{\tilde{\mathbf{f}}\tilde{\mathbf{y}}}|_{\omega=0} = \begin{bmatrix} -0.0370 & 0.0370 \\ -0.0370 & 0.0370 \\ -49.984 & 49.984 \\ 0 & 0 \\ -49.984 & 49.984 \end{bmatrix} \quad (4.10)$$

Note that the matrix representing the DC gain is rank deficient. Thus, faults in the direction  $\mathbf{f} = [1 \ 1]^T$  are indistinguishable from an unfaulted condition. The unobservability of this fault direction at DC has a simple physical explanation. As mentioned previously, a simultaneous and equal fault in both pitot probes has no effect on the airspeed measurement. A fault in the  $\mathbf{f} = [1 \ 1]^T$  direction only causes a bias in the altitude measurement. The model for the longitudinal dynamics is unaffected by a constant offset in altitude. Thus, a fault in the  $\mathbf{f} = [1 \ 1]^T$  direction will cause the closed loop system to adjust to a biased value of altitude but all measurements will appear, in steady state, to converge back to their original trim conditions.

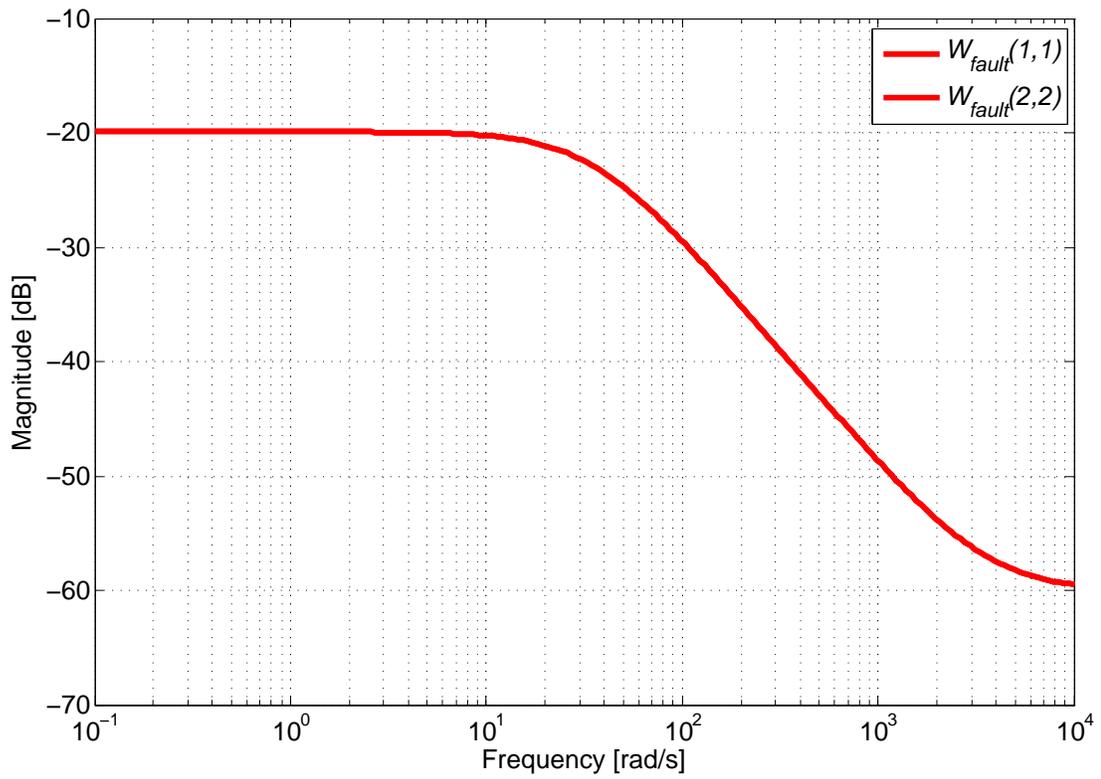


Figure 4.7:  $W_{fault}$ : magnitude frequency response

This rank deficiency places limits on the fault detection performance at low frequencies. For a filter  $F$  to ensure perfect fault tracking at low frequency,  $F$  must be a pseudoinverse of  $P_{\hat{\mathbf{f}}\mathbf{y}}$  over that frequency range. In particular, a filter  $F$  that would make the tracking error arbitrarily small at low frequency cannot be synthesized by `hinfsyn` [28] because the partition is rank-deficient and its pseudoinverse does not exist.

To circumvent this problem,  $W_{error}$  is chosen such that the DC gain is small (-40 dB) and begins to roll up at  $10^{-5}$  rad/s to -12 dB at  $10^{-2}$  rad/s. For frequencies greater than  $10^{-2}$ , the traditional approach of rolling off to a small high frequency gain (-60 dB) is applied. This error weighting has a small DC gain, rolls up at very low frequencies, and rolls down again at higher frequencies as shown in Figure 4.8; this allows for the best filter performance given the inherent system limitations.

$$W_{error} = \frac{0.0011s^2 + 0.1106s + 4.606 \times 10^{-5}}{s^2 + 0.4683s + 4.606 \times 10^{-3}} I_2 \quad (4.11)$$

### 4.3 FDI Filters

The weighted interconnection shown in Figure 4.3 is generalized into the weighted generalized plant  $P_{gen}$  [29]. The filter  $F$  is synthesized with a  $\gamma$ -value of 0.1045 using  $P_{gen}$  and `hinfsyn` to meet the objectives described in Sections 4.1-4.2. The `hinfsyn` algorithm synthesizes a filter at the low  $\gamma$ -value for a few reasons. First, the small weight choices scale  $\gamma$  to be small. Next, model uncertainty is not considered in this formulation, allowing for stronger filter attenuation of disturbances. Having achieved a fault detection filter, the filter performance will be analyzed in the subsequent chapter.

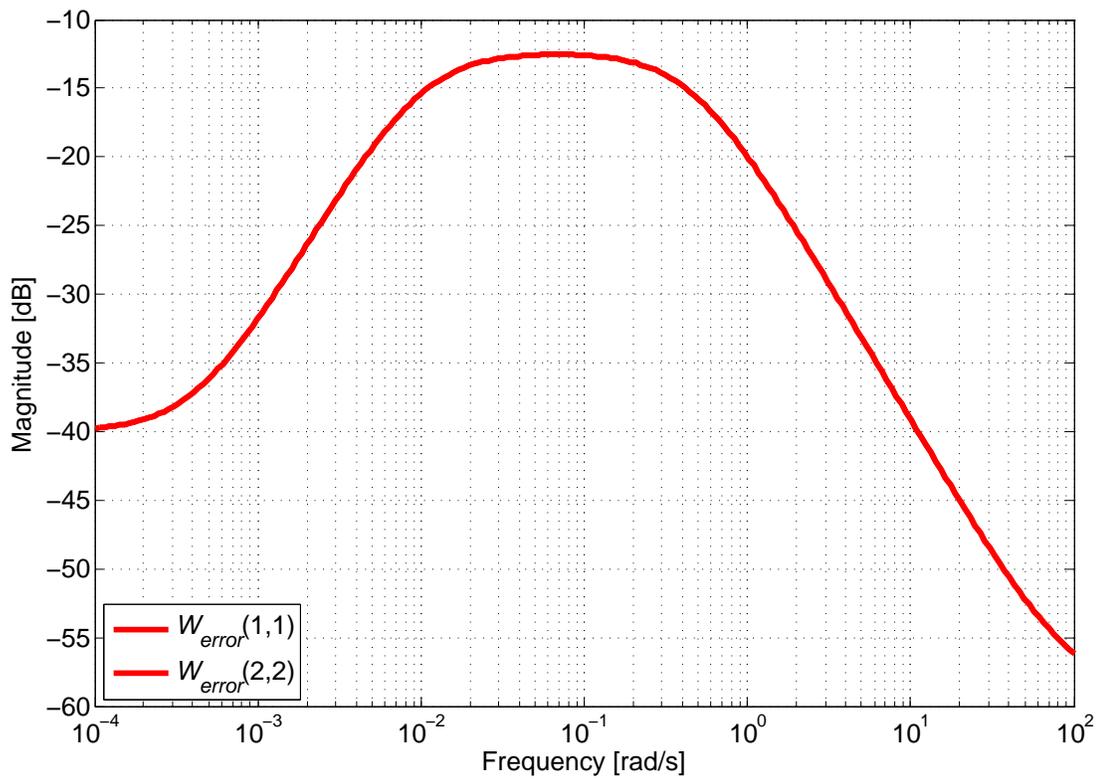


Figure 4.8:  $W_{error}$ : magnitude frequency response

# Chapter 5

## $H_\infty$ Fault Detection Results

The fault detection performance of the filter  $F$  synthesized in Chapter 4 is examined for three fault scenarios. First, fault detection performance is analyzed for a small step fault in the static pressure measurement of the closed-loop linear and nonlinear GTM models. The total pressure measurement is unfaulted. Similarly, fault detection performance is also evaluated for a step fault in the total pressure measurement of the linear and nonlinear GTM models while the static pressure measurement remains unfaulted. Finally, the performance of the synthesized filter is examined for a simultaneous fault in  $p_s$  and  $p_t$  in the closed-loop nonlinear GTM. Detection performance is analyzed in the presence of inertial sensor noise for all simulations.

### 5.1 Chosen Faults and Desired Filter Performance

A simple representation of certain air data blockages can be obtained by injecting step faults into the static and total pressure measurements. The performance of the synthesized filter under these scenarios is examined.

The filter should yield estimates that track the generalized fault inputs reasonably quickly with minimal steady-state error. False positives are very undesirable. Any fault detection system implemented with the goal of control reconfiguration must be sufficiently fast as to allow for reconfiguration before undesired aircraft maneuvers become unsafe.

## 5.2 Step Fault Detection

The fault tracking performance of  $F$  for a small step fault in the static pressure measurement is shown in Figure 5.1. At time  $t = 1$  second, a step fault of magnitude 0.01 psi is injected into the static pressure measurement signal for the linear and nonlinear generalized plant using the filter  $F$ . The simulation has a duration of 4 seconds and inertial and pitot sensor noise is included. The filter outputs  $\hat{\mathbf{f}}$  should track the fault in the faulted static pressure channel and additionally show no fault in the unfaulted total pressure channel.

The filter detects the static pressure fault in the linear model rapidly, rising to correctly estimate the fault magnitude within 1.5 seconds. Additionally, the filter fault estimate effectively tracks the static pressure measurement fault after detection. The filter does not yield errorless fault tracking, however, due to the *genGTM* rank deficiency described in Section 4.2. The slowest pole of  $F$  has a frequency on the order of  $10^{-5}$  rad/s, so the fault estimation error will grow quite slowly. The fault estimate will eventually decay to zero in the faulted channel and drift away from zero in the unfaulted channel. Measures to combat this drift must be designed into any algorithm that can be implemented on an operational system.

Because noise on the inertial sensors is fed into the filter, the resulting fault estimates in both channels will be noisy to some extent. In these simulations, however, the fault estimate in the total pressure fault channel will generally exhibit slightly higher noise levels relative to the static pressure fault estimate. As stated in Section 4.2, inertial sensor noise couples to  $\hat{f}_{pt}$  more strongly than  $\hat{f}_{ps}$ , accounting for the higher noise levels in the total pressure estimate. Since the inertial measurements are fed into the airspeed-hold autopilot, the filter relies on these measurements to detect the presence of a fault in the airspeed measurement more than it does for the altitude measurement. Consequently, the noise levels in  $\hat{f}_{pt}$  are larger. While the  $H_\infty$  filter is designed to minimize the effect of sensor noise on the fault estimates by penalizing high frequency filter outputs, noise in the estimates cannot be entirely eliminated.

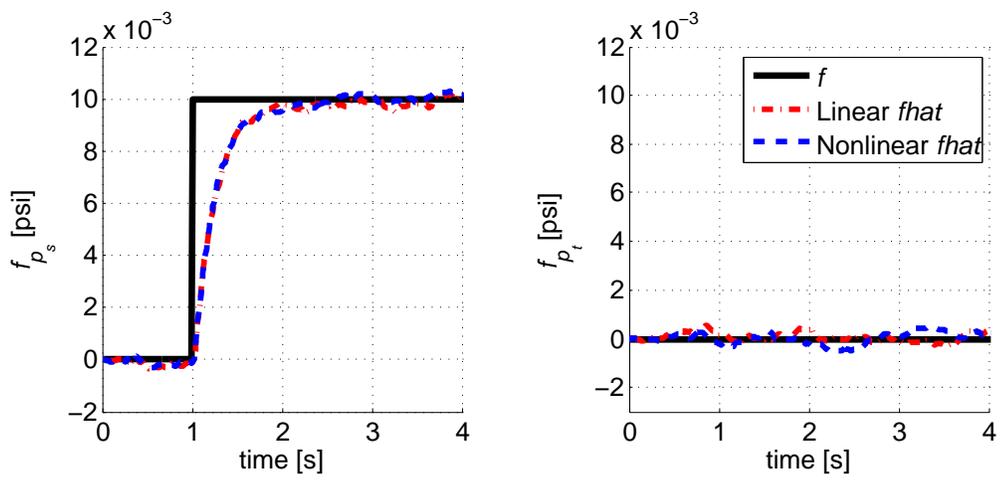


Figure 5.1: Fault estimation:  $p_s$  step fault, linear and nonlinear GTM

Next, the same simulations are conducted with a total pressure fault rather than a static pressure fault. Figure 5.2 shows that the filter detects the total pressure fault in the linear model and tracks the fault with some residual noise for the reasons explained above. The filter correctly estimates no fault on the static pressure measurement. Similar to the static pressure fault case, the estimates will eventually drift during long simulations.

Next, a fault of equal magnitude injected simultaneously into the static and total pressure channels will not have an impact on the airspeed measurement. Because such a compound fault has limited observability, it is interesting to examine the ability of  $F$  to detect such a condition. Again, 0.01 psi step faults are injected into both the static and total pressure measurements at time  $t = 1$ .

Figure 5.3 shows the fault estimates for the simultaneous fault in the nonlinear GTM. Figure 5.4 shows the control input and aircraft state responses of the nonlinear GTM simulation for the simultaneous fault condition. The simultaneous fault results in a bias in the altitude measurement while the indicated airspeed is correct. Note that all of the control inputs and aircraft states— except the altitude measurement—converge back to the original trim condition. The only effect of the simultaneous fault is that the aircraft converges to an offset altitude. Despite the simultaneous fault that does not appear in the airspeed measurement at zero frequency, the filter is able to detect the initial step in both measurements as faults. The filter uses the inertial state measurements to track the fault by compensating for the dynamic response of the aircraft to the step changes in the measurements. The fault estimates eventually decay to zero due to the unobservability of this fault in steady state. Because this decay is so slow, however, a fault identification and reconfiguration algorithm would have sufficient time to respond to the fault occurring in this simulation.

### 5.3 Time-Varying Fault Detection

While none of the faults described in Chapter 2 would necessarily exhibit a clear periodic behavior, it is possible that loose, partial port blockages or fluid in the pneumatic lines could cause time-varying fluctuations in the air data measurements. Therefore, a full analysis must include investigations into the ability of the fault detection algorithm to succeed in the event of a fault with frequency content.

Because the closed-loop aircraft dynamics roll off at moderate frequencies, high fre-

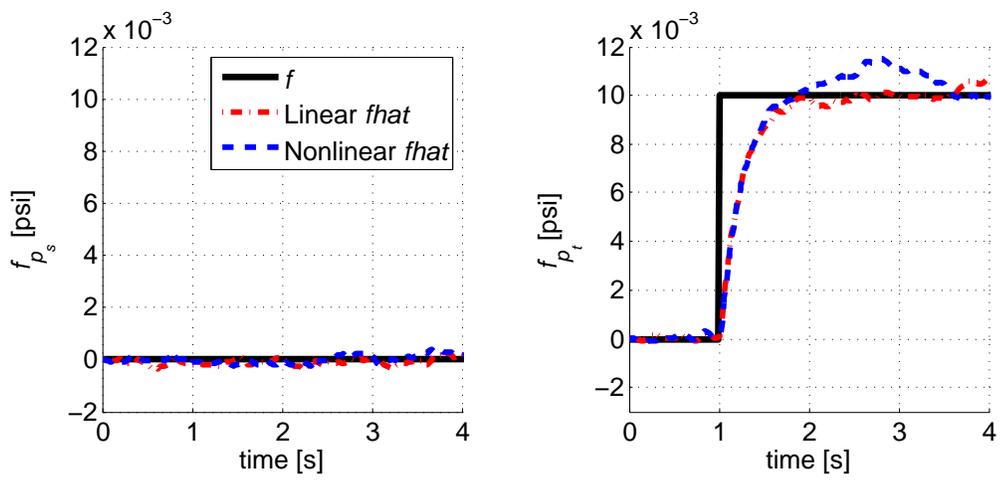


Figure 5.2: Fault estimation:  $p_t$  step, linear and nonlinear GTM

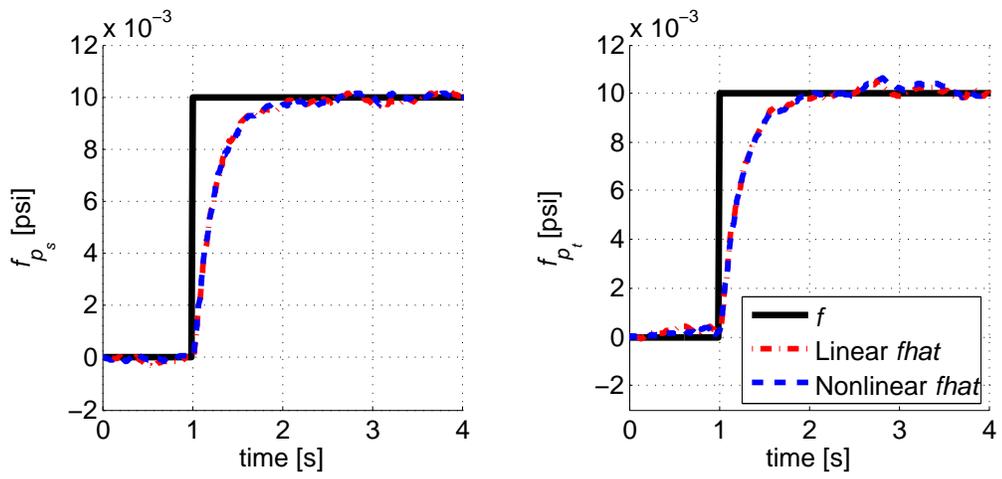


Figure 5.3: Fault estimation: simultaneous  $p_s$  and  $p_t$  step faults, linear and nonlinear GTM

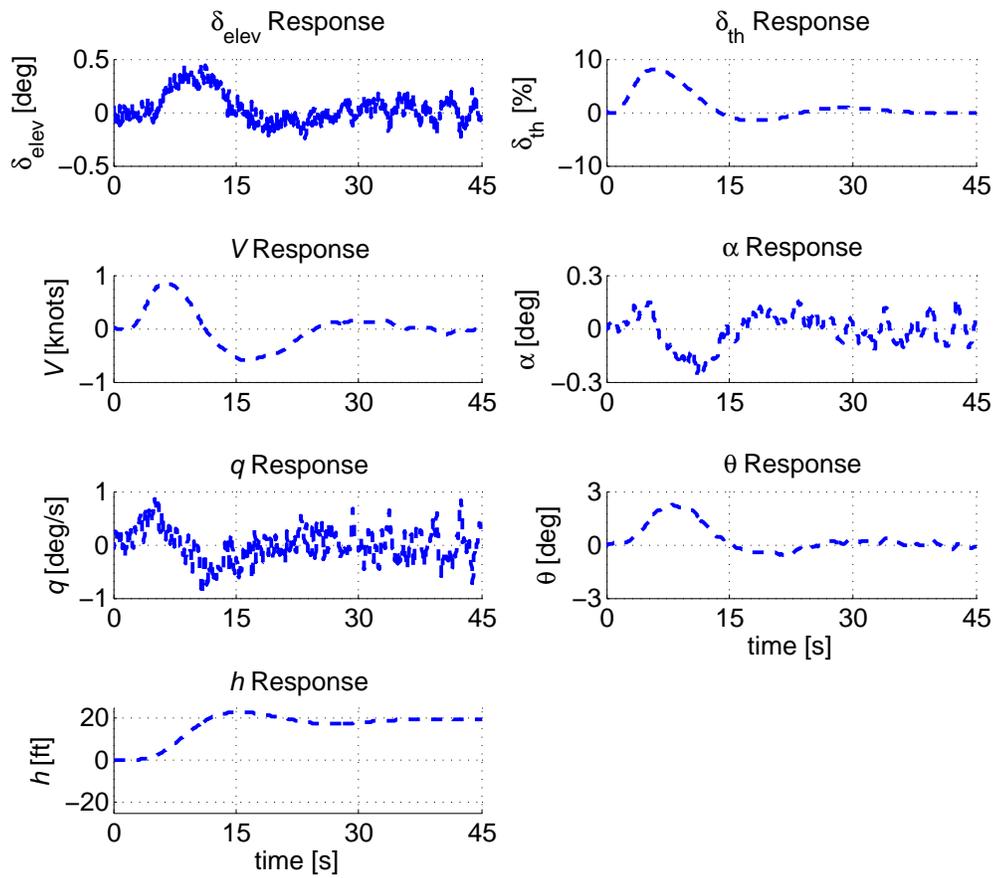


Figure 5.4: Nonlinear GTM control input and state responses: simultaneous  $p_s$  and  $p_t$  step faults

quency variations in the pressure measurements will not adversely impact the GTM aircraft in controlled flight. Fluctuations with lower frequency content, however, could pose a problem.

To examine the ability of the filter to detect time-varying faults, sinusoidal faults replace the step faults in the previously detailed simulations. The amplitude of the injected sinusoids is 0.01 psi – as with the step faults – and the first frequency considered is 0.03 radians per second. The faults are initiated with the simulation at  $t = 0$  and run for 45 seconds in order to capture data over a larger fraction of the fault period. Figure 5.5 shows the fault estimate tracking for the linear and nonlinear GTM models with a static pressure measurement fault. The filter tracks the slow sinusoidal fault well and properly indicates nominal operation for the unfaulted total pressure measurement channel.

The filter has shown positive results for steady-state faults and for slowly time-varying faults, so an examination of a higher frequency fault is considered. The previous simulation is run with a fault frequency of 0.3 radians per second – an order of magnitude faster – and all other parameters left the same. Figure 5.6 shows the fault estimates tracking well in the linear model, but the estimates in the nonlinear model begin to drift away from the true fault. With time, the estimates become sufficiently inaccurate to suggest a false positive fault identification in the total pressure channel. This example illustrates the degraded performance of the filter for time-varying faults of increasing frequencies. Moreover, it shows some limitations of this fault detection approach and filter design for a nonlinear system.

## 5.4 Filter Disturbance Rejection

Pilot reference commands in the  $H_\infty$  filtering problem are the analog of disturbances in the  $H_\infty$  control problem. When a pilot supplies a reference command for the aircraft to change airspeed or altitude, it is desirable that the fault detection filter performs such that false positives are not triggered due to the aircraft's motion. Furthermore, it is important that a fault that exists can continue to be detected during aircraft maneuvers.

To determine whether the filter can avoid false positive detections when the aircraft is changing airspeed or altitude, simulations of the nonlinear GTM are conducted with no faults and various changes in the altitude or airspeed reference commands.

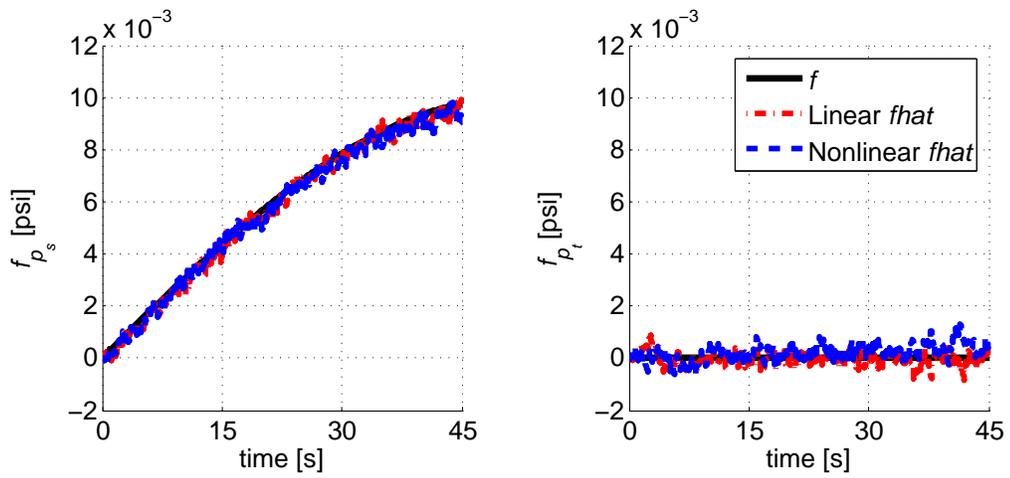


Figure 5.5: Fault estimation:  $p_s$  0.03 rad/s sinusoidal fault, linear and nonlinear GTM

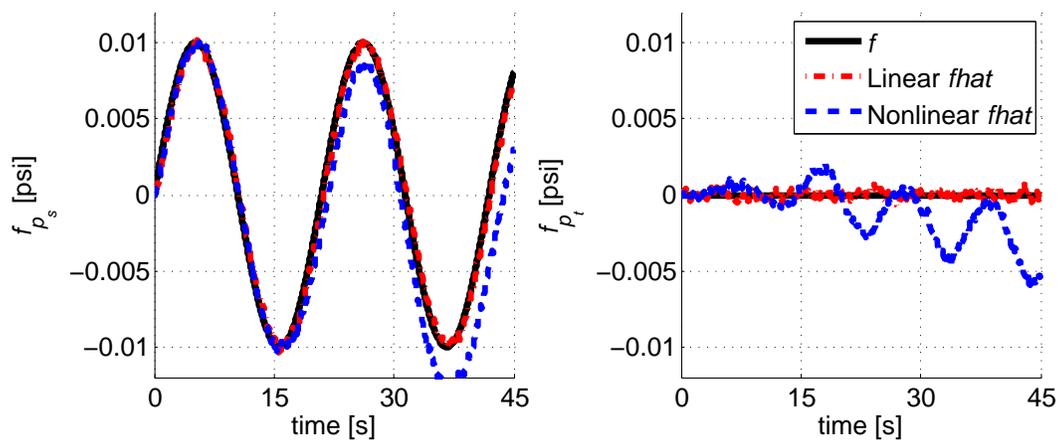


Figure 5.6: Fault estimation:  $p_s$  0.3 rad/s sinusoidal fault, linear and nonlinear GTM

Figure 5.7 shows the filter disturbance rejection for two different steps in the airspeed reference command at  $t = 5$  seconds: 3 knots and 10 knots. The smaller airspeed step of 3 knots roughly corresponds to the 0.01 psi error in the dynamic pressure that was simulated previously. The static pressure measurement fault estimate is hardly impacted by this maneuver. The total pressure measurement fault estimate is influenced more, but the estimate is approximately 0.001 – an order of magnitude smaller than the 0.01 psi fault. By properly designing thresholds to declare faults, one could effectively ignore these small variations during a small airspeed change.

For more significant maneuvers – such as the 10 knot airspeed step depicted in Figure 5.7 – the filter outputs could indicate a false positive. This suggests, not surprisingly, that the fault detection filter is most robust to disturbances such as changing reference commands when the aircraft remains near the reference flight condition at which the filter was designed.

Figure 5.8 shows the filter disturbance rejection for two different steps in the altitude reference command: 20 feet and 100 feet. Similar to the step choices for the airspeed maneuvers, the smaller 20 foot altitude step corresponds roughly to the step in the altitude measurement when a 0.01 psi fault occurs in the static pressure measurement. For an altitude change of this magnitude, very little error in the fault estimates is observed. The error is, in fact, less than the error for the 3 knot airspeed step. This is due to the fact that the aircraft dynamics do not change for altitude variations nearly as significantly as they change when airspeed is altered. Even small airspeed variations can significantly alter aerodynamic forces and moments on the aircraft. Altitude variations must be large in order to experience total pressures and total temperatures that are sufficiently different to cause a similar change in the aerodynamic forces and moments on the aircraft.

## 5.5 Summary

These experiments demonstrate that this filter can work well at detecting small steady-state and slowly time-varying faults in regions of the flight envelope near the reference flight condition at which the aircraft is trimmed and for which the filter is designed. The next chapter will explore refinements to this fault detection algorithm in order to better understand how successfully faults can be detected throughout the flight envelope of the aircraft.

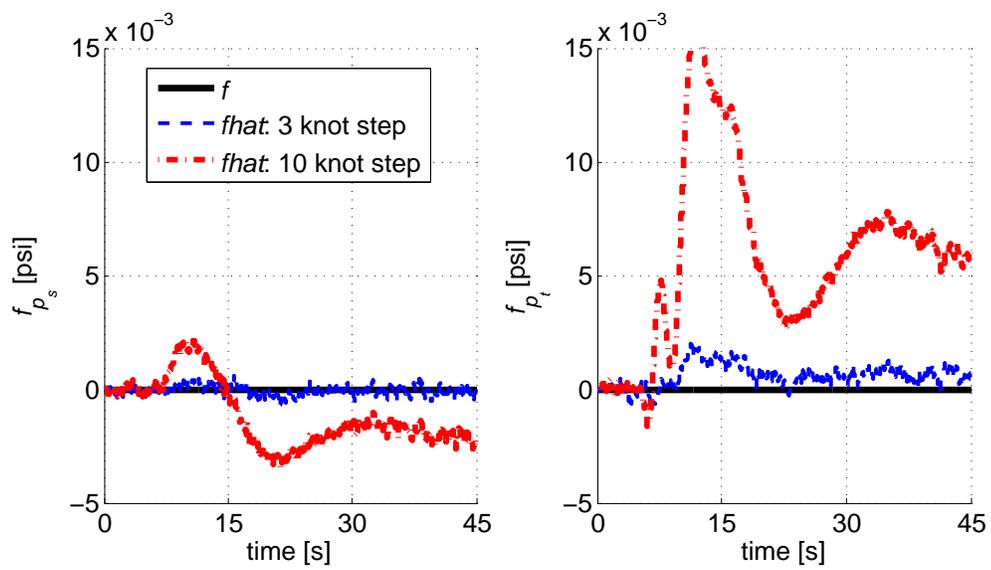


Figure 5.7: Disturbance rejection: airspeed reference command steps in nonlinear GTM simulation

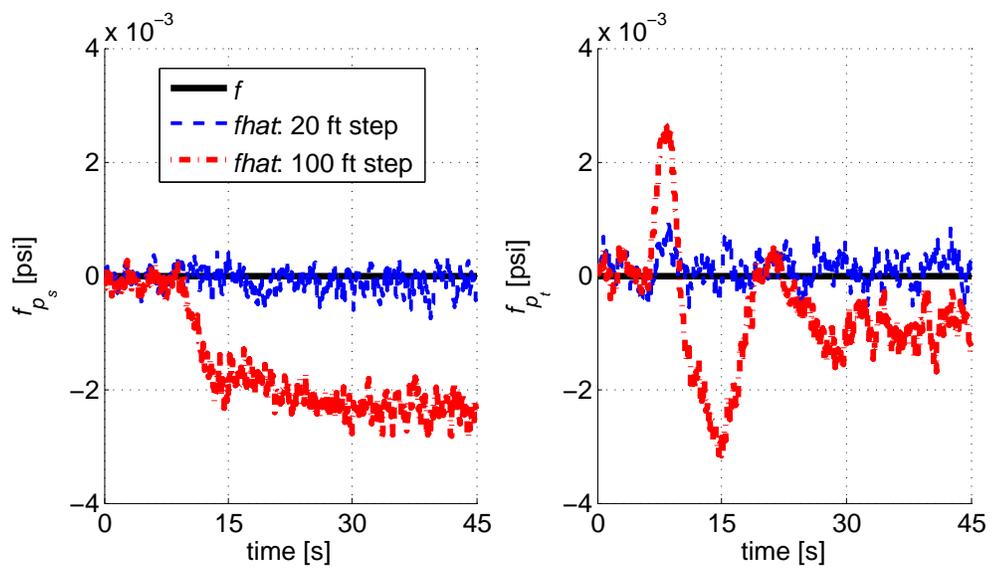


Figure 5.8: Disturbance rejection: altitude reference command steps in nonlinear GTM simulation

# Chapter 6

## Scheduling and Estimate Blending

Analysis of the fault detection results presented in Chapter 5 indicates that faults can be well-detected provided that aircraft is operating at or near the reference flight condition for which the filter is designed. While this is promising, any effective, implementable fault detection algorithm must perform consistently for most or all operating points within the flight envelope. The fault detection algorithm, therefore, must be refined in order to extend its performance capabilities beyond the limited cases shown in Chapter 5. This will allow for a more realistic assessment of the successes – and remaining challenges – associated with the model-based fault detection techniques presented in this thesis. Refinements to the fault detection filters examined in this chapter include: control scheduling, filter scheduling, DK-iteration filter synthesis, and filter output blending approaches.

For a model-based approach, the characterization of the underlying dynamics of the model—and the ability to account for variations of those dynamics—is crucial. Before making any adjustments to the filter design, it is important to recall that the aircraft autopilot described in Section 3.4 was designed for a single flight condition, shown in Equation 3.14. Open-loop aircraft dynamics vary according to the aircraft operating point; using a controller designed for a single operating point for a set of plants can yield a set of closed-loop systems with significant dynamics variation across the set. A filter designed for just one of these conditions (as in Chapter 4) can produce erroneous fault estimates at off-design conditions due to the variation of the closed-loop system dynamics.

In order to demonstrate this behavior, fault detection performance for a set of GTM operating points is examined using the previously designed filter and control law. At the low altitudes flown by the GTM, variations in altitude between operating points have negligible effects on the associated aircraft dynamics. Changes in airspeed have a more significant effect on in-flight aerodynamic forces, and these variations will more adversely impact fault detection efficacy. The GTM model is trimmed and linearized about a set of steady, level reference flight conditions; airspeeds vary from 60 to 90 knots in 5 knot increments, and all altitudes are 500 feet. This results in seven distinct GTM models for simulation. The same control law from Chapter 3 and filter from Chapter 4 are used for each of the models. The model corresponding to the original trim condition from Chapter 3 (75 knots airspeed, 500 feet altitude), for the purposes of explanation, is considered the *design* model since it is used to develop the control law and filter designs. All other models are considered *off-design* models.

Figure 6.1 shows the fault estimate response to a 0.01 psi static pressure step fault for the GTM linearized about each of the seven trim conditions associated with the airspeed sweep. As shown in the previous chapter, the filter properly estimates the static pressure measurement fault and correctly indicates no fault in the total pressure measurement for the design model. For the set of off-design models, however, there are significantly larger estimation errors for the static pressure channel than for the design model. In the total pressure channel, fault estimates incorrectly indicate that the total pressure measurement is somehow faulted. These errors are sizeable, and similar unacceptable errors are seen when introducing a step fault in the total pressure channel or sinusoidal faults in either channel. Measures to reduce dynamics variations across the flight envelope are required for the filter to more consistently estimate faults.

## 6.1 Control and Filter Scheduling

Implementing a scheduled controller is the first step to improving fault detection performance and does not require modification of the filter. The control law was originally developed using a loop-at-a-time approach, and a scheduled controller can be achieved by scheduling each tracking loop in succession. While the complexity of gain scheduling can vary significantly depending on the particular application, designing the scheduled controllers for this thesis is a simple process. For each tracking loop, a new controller of the same type as described in Section 3.4 is designed for

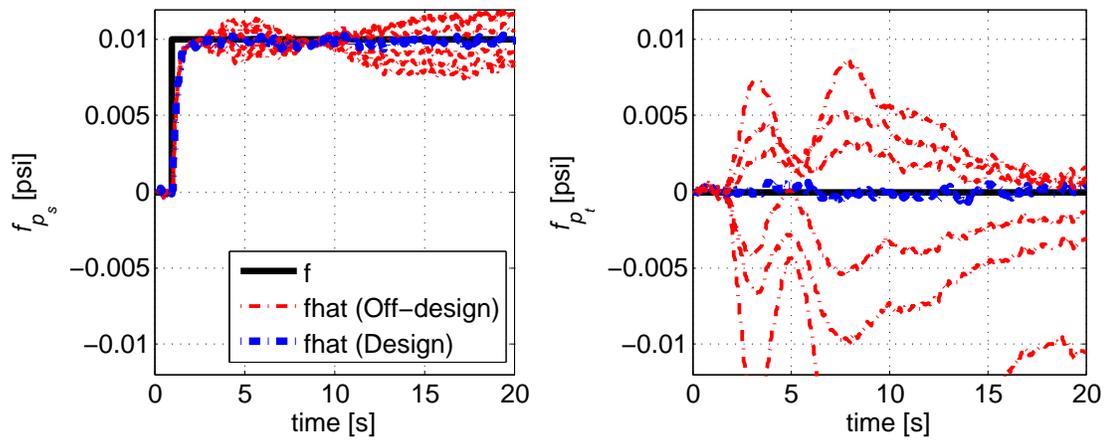


Figure 6.1: Fault detection: performance across airspeed swept models with  $p_s$  step fault

the linear model corresponding to the 90 knots reference flight condition. Constants are adjusted such that time and frequency response characteristics of the controlled aircraft model correspond closely to those of the original controlled GTM model from Chapter 3. Adjusting the controller gains in this manner removes significant variation of the full aircraft dynamics between those two conditions. After determining the appropriate constants for the 90 knots reference flight condition, an interpolation is performed for airspeeds between 75 and 90 knots. This process yields the scheduled set of controllers over the airspeed sweep.

Simulations with the scheduled control law are conducted in order to assess the fault detection performance across the airspeed interval. A linear GTM model is derived on 2.5 knot intervals over the entire airspeed sweep (i.e. [75, 77.5, ...90]). As with previous simulations, 0.01 psi faults in the static and total pressure channels are injected into each of these models. Figures 6.2-6.3 show the fault estimate responses without and with control scheduling, respectively. False positives continue to be a problem, and the estimates from the off-design models consistently exhibit a scaling behavior. Control scheduling is shown to insufficiently reduce model variation between reference flight conditions. Perhaps more advanced techniques in control design and scheduling could lead to marginal improvements, but filter refinement stands to have a greater impact upon fault detection performance.

While the filter performs well at the flight condition for which it was designed, it is sensitive to measurement changes at off-design flight conditions. One approach to counter the negative effects of the filter sensitivity is to deploy a set of fault detection filters that are designed for different operating points within the flight envelope. Scheduling filters in this way forms the basic principle behind more advanced linear, parameter-varying filtering designs, and it can significantly improve the utility of a fault detection algorithm. [30]

For each reference condition used in the control scheduling, a filter is synthesized in the manner outlined in Chapter 5. The design of the filters (i.e. their weighting functions) remains unchanged. As with the control scheduling simulations, static and total step faults are injected into the models—this time adding scheduled filters—across the airspeed sweep. Simulation results are shown in Figures 6.4-6.5. Fault tracking performance is quite strong since each filter is tuned for the GTM dynamics at each flight condition. Furthermore, no false positive fault estimates are provided

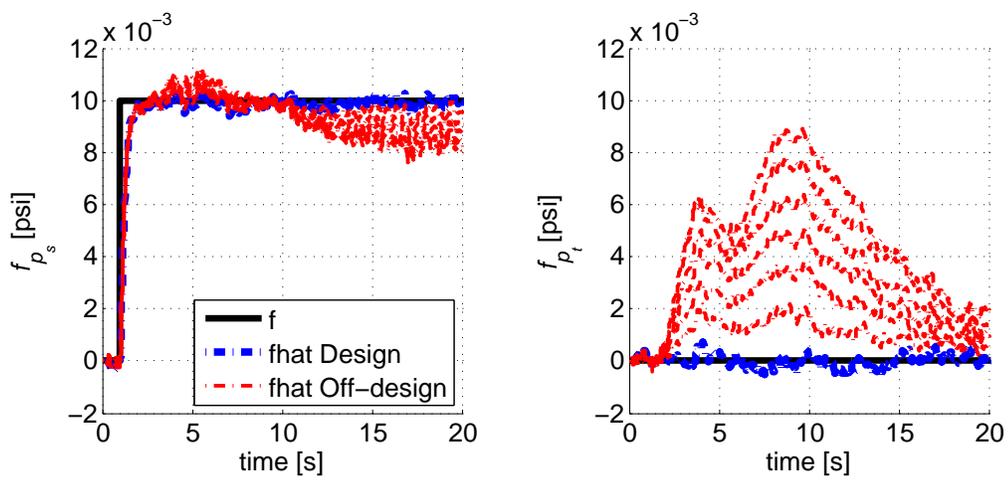


Figure 6.2: Fault detection:  $p_s$  step fault with control scheduling over [75 90] knots airspeed interval

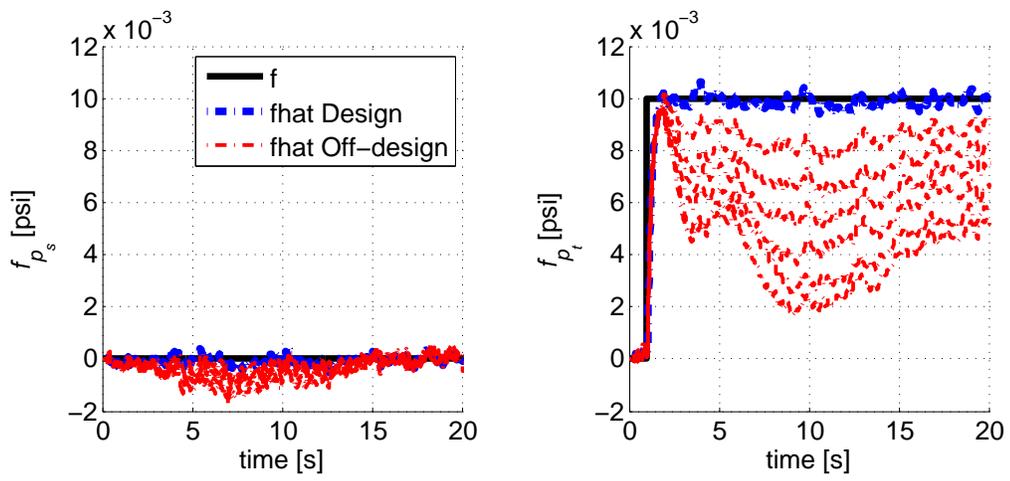


Figure 6.3: Fault detection:  $p_t$  step fault with control scheduling over [75 90] knots airspeed interval

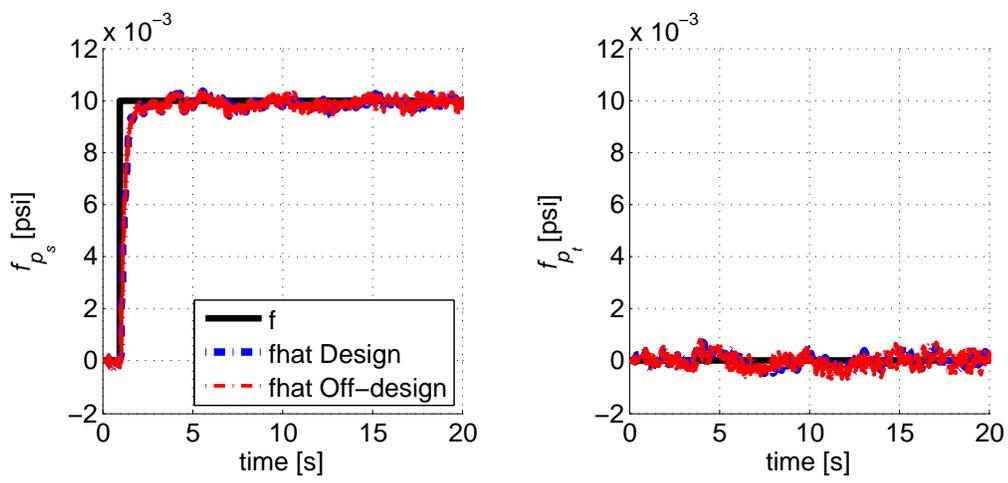


Figure 6.4: Fault detection:  $p_s$  step with control and filter scheduling over [75 90] airspeed interval

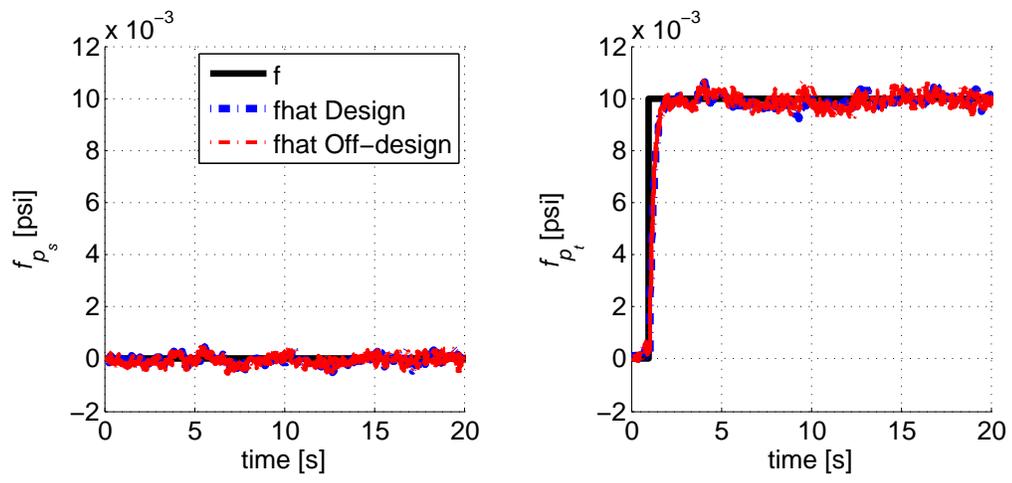


Figure 6.5: Fault detection:  $p_t$  step with control and filter scheduling over [75 90] airspeed interval

by the filters. Of course, it is not feasible to synthesize a unique fault detection filter for every possible point in the flight envelope in the manner used here. For an implementable fault detection system, it is necessary to accurately estimate faults at operating points in between the filter design points. Put another way, the robustness of the fault detection filter(s) must be further enhanced in order to ensure adequate performance for a general operating point in the flight envelope.

## 6.2 Uncertainty Modeling and *DK*-iteration

Recall from Chapter 4 that the  $H_\infty$  problem formulation does not include uncertainty modeling. By mathematically representing model uncertainty within a generalized plant prior to controller(filter) synthesis, a  $H_\infty$  controller can be chosen such that the closed-loop system is more robust to uncertainty. This improvement to the original GTM generalized plant yields a more realistic model for filter synthesis and fault detection simulations. The *DK*-iteration filter synthesis technique is employed in order to compare fault detection performance with the previously considered  $H_\infty$  approach.

### 6.2.1 Problem Formulation

The *DK*-iteration procedure is a numerical technique for approximating  $\mu$ -synthesis controller design. In  $\mu$ -synthesis design, the goal is to minimize the structured singular value  $\mu$  of the robust performance problem for an uncertain generalized plant. *D-K* iteration combines  $H_\infty$  synthesis and  $\mu$ -analysis techniques and can yield improved results.

Uncertain perturbations to the system of interest are represented by the diagonal matrix

$$\Delta = \text{diag}\{\Delta_i\} \tag{6.1}$$

where each  $\Delta_i$  represents a particular uncertainty source. The controller  $K$  and plant  $P$  are configured with  $\Delta$  to form the general control configuration for uncertain systems shown in Figure 6.6. The system contains three sets of inputs: perturbations  $w$ , disturbances  $d$ , and controls  $u$ . There are three sets of outputs: perturbation outputs  $z$ , errors  $e$  and measurements  $y$ .

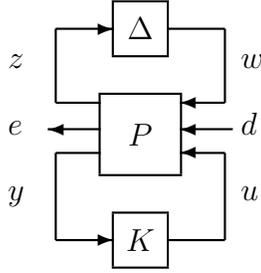


Figure 6.6: System configuration for uncertain plant control synthesis

The design objective is to choose a stabilizing controller  $K$  such that for all perturbations  $\Delta$ , belonging to the set of allowable perturbations, the system is stable and satisfies

$$\| F_L[F_U(P, \Delta), K] \|_\infty \leq 1 \quad (6.2)$$

The upper linear fractional transformation represents the perturbed plant in Equation 6.2. Define  $\mathcal{D}$  to be the set of matrices  $D$  which satisfy  $D\Delta = \Delta D$ . The parameter considered in the  $DK$ -iteration procedure is the upper bound on  $\mu$  (Equation 6.3) represented in terms of the scaled singular value:

$$\mu(N) \leq \min_{D \in \mathcal{D}} \bar{\sigma}(DND^{-1}) \quad (6.3)$$

For  $DK$ -iteration, the goal is to find the controller minimizing the peak value of the upper bound over all frequency

$$\min_K (\min_{D \in \mathcal{D}} \| DN(K)D^{-1} \|_\infty) \quad (6.4)$$

This computation is an iterative, two-step minimization process:

1.  $K$ -step: Synthesize an  $H_\infty$  controller for  $\min_K \| DN(K)D^{-1} \|_\infty$  with fixed  $D(s)$ .
2.  $D$ -step: Find  $D(j\omega)$  to minimize  $\bar{\sigma}(DND^{-1}(j\omega))$  with fixed  $N$ . The magnitude of each element of  $D(j\omega)$  is fit to a stable, minimum-phase transfer function  $D(s)$ .

This iteration is repeated until satisfactory performance is achieved, the  $H_\infty$  norm no longer decreases, or  $\|DND^{-1}\|_\infty < 1$  [20].

As with the  $H_\infty$  synthesis detailed in Chapter 4, the objective of the  $DK$ -iteration approach is to design a filter  $F$  to track fault signals in the static and total pressure measurements rather than other exogenous signals. To formulate the  $DK$ -iteration for the GTM, an uncertainty model must first be created. Input uncertainty is frequently modeled in control design due to its presence in any practical application. Perturbed throttle and elevator control inputs  $\mathbf{u}_p$  are given by

$$\mathbf{u}_p = (I + \Delta_I W_I) \mathbf{u} \quad (6.5)$$

where  $W_I$  is the input uncertainty weight and  $\Delta_I$  is the 2x2 block-diagonal, normalized input uncertainty. The uncertainty is norm-bounded to be less than 1. For simplicity, a constant  $W_I$  of 0.50 is chosen, representing a maximum of 50% uncertainty in either the elevator command or throttle command. This is a large level of uncertainty that will provide a conservative performance estimate. A block diagram of the input uncertainty architecture is shown in Figure 6.7. Using this architecture and the GTM model derived previously, the MATLAB `dk`syn function is used to synthesize the filter.

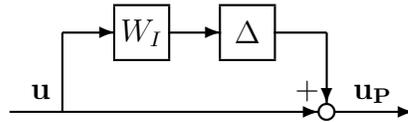


Figure 6.7: Multiplicative input uncertainty

### 6.2.2 Results

The GTM fault detection simulations are configured for control signal perturbations. The previous results show that the accuracy of fault estimates is sensitive to small changes in the aircraft's flight condition. This sensitivity motivates filter scheduling across many flight conditions. Before moving forward with implementing a scheduled filter throughout the flight envelope, however, it is useful to ascertain whether the uncertainty modeling and  $DK$ -iteration approach can improve filter accuracy at a given flight condition in the presence of input perturbations.

The `hinfsyn` and `dksyn` filters are generated based on the GTM linear model associated with the 75 knot, 500 foot reference flight condition. Rather than simulating the filters with the model upon which they were designed, the GTM model is then linearized about the 77.5 knot, 500 foot reference flight condition, and the  $H_\infty$  and  $DK$ -iteration filters are simulated with that model in the presence of uncertainty. This allows for a comparison of filter performance—in the presence of input uncertainty—at an off-design operating point. A filter that is more effective at off-design flight conditions is more suitable for extension to the entire flight envelope as it provides an additional level of robustness.

For constant step faults such as those used previously, both filters perform similarly. The differences in filter performance, while small, become more apparent when examining time-varying faults. This is to be expected, as input uncertainty will have the greatest impact when the control inputs are varying. For time-varying faults, the controller commands more control surface actuation than for a step fault (which allows the aircraft to reach a new trim condition relatively quickly). The injected static pressure fault is a sinusoid with 0.01 psi amplitude and 0.03 rad/s frequency. Twenty random input uncertainty models are selected and the fault detection performance is repeatedly simulated with each uncertainty model. Figure 6.8 shows the overlaid fault estimate responses with the  $DK$ -iteration filter simulation results in the top two plots and the  $H_\infty$  filter simulation results on the bottom.

Comparing fault tracking in the static pressure measurement channel, both the  $DK$  and  $H_\infty$  track the sinusoidal fault well. For a filter simulated at an off-design flight condition, this is a promising sign. Additionally, there is little variance in the fault estimates among the twenty uncertain models; either filter can accommodate significant input uncertainty and track faults of this type. In the unfaulted total pressure measurement channel, both filters fail to track zero and show small sinusoidal cross-coupling effects. The cross-coupling is potentially small enough to avoid triggering a fault flag depending on the chosen thresholding scheme. The effects of uncertainty are also more apparent in this channel. The  $DK$  filter has a more consistent response with a smaller variance in the fault estimate between simulations. The  $H_\infty$  does not perform as well, and for particular uncertainty combinations the fault estimate response begins to diverge from the other simulated responses. These differences are small, but they show that the filter performance can be improved to be less sensitive to flight conditions and unmodeled dynamics—yet still provide useful fault estimates.

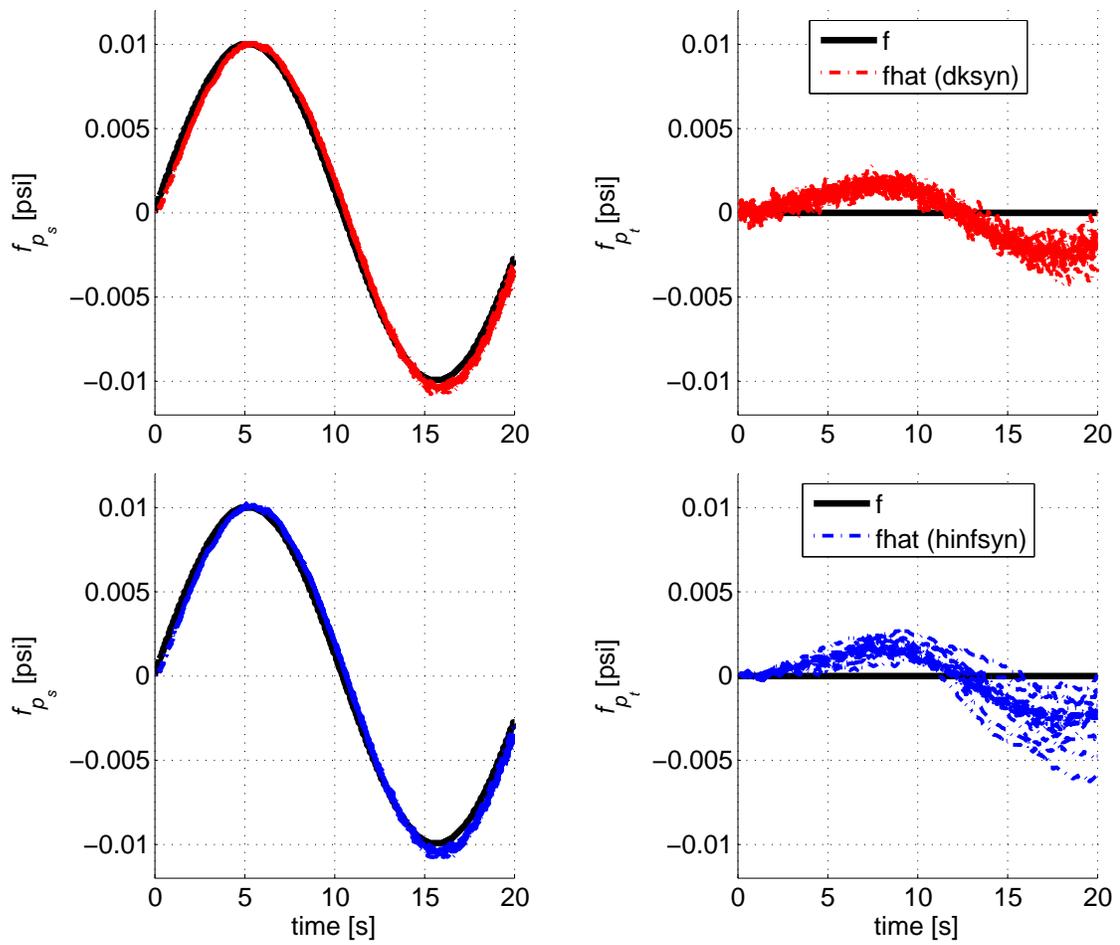


Figure 6.8: Fault detection performance comparison for  $DK$ -iteration and  $H_\infty$ -synthesis with multiplicative input uncertainty

### 6.3 Estimate Blending

Using the *DK*-iteration approach, a series of scheduled filters can be generated as in Section 6.1. For automated implementation, a process must be established to effectively stitch the filters together so that an aircraft moving about different flight conditions can continue to receive a smooth, accurate fault estimate. There are a variety of ways to approach this problem, including LPV techniques. For this thesis, not specifically focused on the estimate blending problem, a simpler approach will be explored in order to understand the initial fault detection abilities and shortcomings of model-based techniques.

A series of filters are synthesized over the airspeed sweep [75, 76, 77...90] knots and 500 feet altitude. These filters are scheduled based on that airspeed vector, with linear interpolation on the filter state space matrices between breakpoints.

It is desirable that the aircraft be able to move about the flight envelope without inducing a false positive fault indication in either the static or total pressure channel. To verify that this is possible, the aircraft is simulated at the 90 knot, 500 foot steady, level reference flight condition. At  $t = 5$ , the aircraft is provided a -20 knot airspeed step, causing the aircraft to slow through (and beyond) the entire airspeed sweep for which filters were designed. This is repeated 15 times with random input uncertainty, and Figure 6.9 shows the overlaid fault estimate responses.

The filter correctly indicates no fault in either channel and provides a consistent response in the presence of uncertainty. There is a larger variance on the total pressure fault estimate, yet it is not so large as to indicate a false positive. Due to changes in total pressure associated with airspeed changes, it is expected that the total pressure fault estimates may have a higher variance than the static pressure.

Next, the same simulation is performed with simultaneous static and total pressure step faults of size 0.01 psi. The faults occur at the same time the aircraft is commanded to reduce airspeed,  $t = 5$  seconds. With properly function fault detection filters, the aircraft should be able to reduce its airspeed yet still correctly estimate the faults. Figure 6.10 shows that the filter is able to track the faults well despite the moving aircraft. The estimate in the total pressure measurement channel is, again, less precise than the static fault measurement. Further refinements to this algorithm could be used to fine-tune filter performance to meet particular specifications.

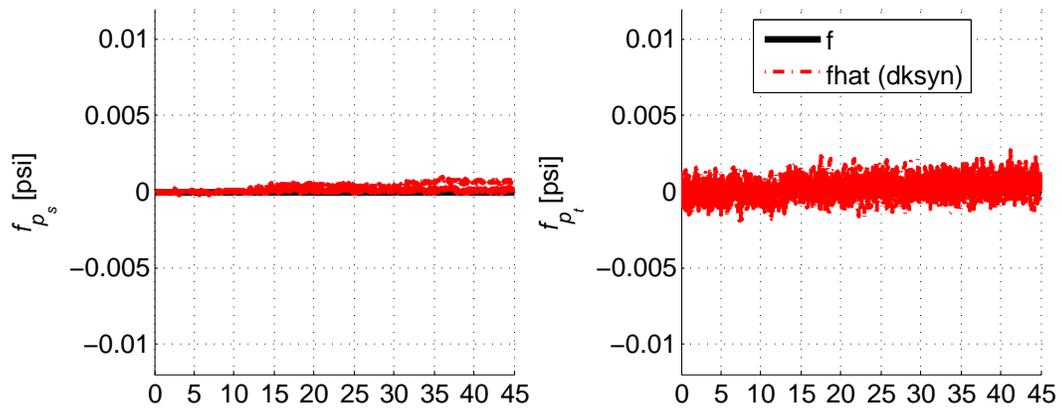


Figure 6.9: Disturbance rejection: -20 knot airspeed step with scheduled control and blended  $DK$ -iteration filters

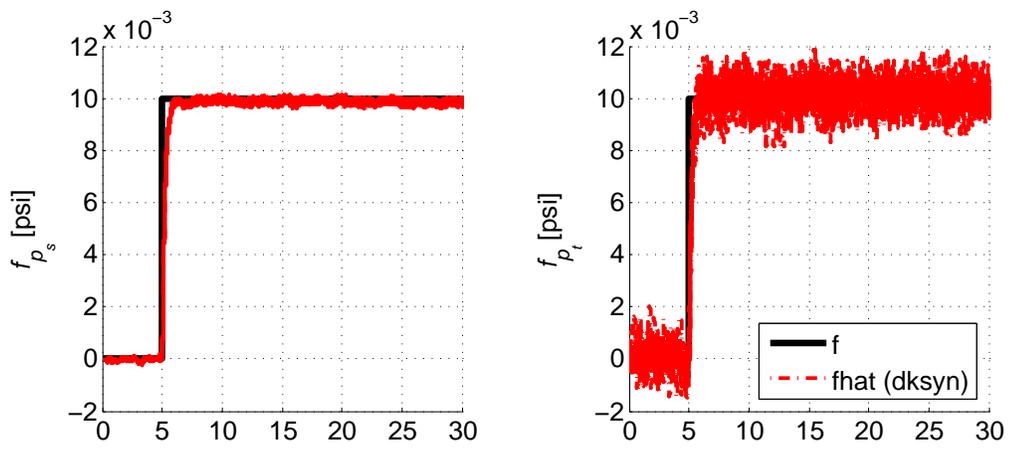


Figure 6.10: Disturbance rejection with static and total pressure step faults: -20 knot airspeed step with scheduled control and blended  $DK$ -iteration filters

## 6.4 Summary

This chapter has demonstrated a number of refinements to the fault detection technique in order to improve performance at conditions for which the filter and controller were not specifically designed. The approach was improved via scheduled control, input uncertainty modeling,  $DK$ -iteration filter synthesis, linear interpolation based filter scheduling. The improved filters were shown to be successful at detecting a simultaneous fault in a maneuvering aircraft as well as rejecting effects related to commanded aircraft maneuvers.

# Chapter 7

## Conclusion

This thesis developed a simple model of an air data system based upon knowledge of industry-standard devices. This model was combined with the a nonlinear longitudinal model of the NASA GTM aircraft in order to provide a fault detection research platform. Using this platform, this thesis developed a method to detect faults in a pitot-static probe using  $H_\infty$  synthesis of a robust fault detection filter. Signal weights were chosen to circumvent the unobservable fault at low frequency. The performance of the fault detection filter was analyzed on the linear and nonlinear longitudinal GTM dynamics for individual step faults, individual sinusoidal faults, and a simultaneous, equal magnitude fault in each pressure measurement. This approach was shown to be effective for detecting step faults and combinations of step faults. The filters were also effective in rejecting the influence of small pilot inputs upon the fault estimates. Large aircraft maneuvers were shown to cause sufficient inaccuracy in the fault estimates to incorrectly indicate the presence of a fault when it did not exist.

This motivated the need for a filter that could more accurately detect faults at off-design conditions in broader regions of the flight envelope. Refinements to the filtering technique were explored, starting with scheduling the controller in order to reduce dynamics variations at different reference flight conditions. This yielded improvements but was insufficiently effective. Filter scheduling showed that consistent fault estimates could be achieved at different reference flight conditions provided the aircraft was flying at a filter design condition. The need for performance between design conditions remained, and an alternative filter synthesis technique using  $DK$ -iteration with uncertainty modeling was examined. Extending this technique to an expanded

flight envelope, successful simultaneous step fault detection and disturbance rejection was demonstrated.

Going forward, enhancements in air data probe modeling must be made in order to improve the knowledge base necessary to better detect and identify air data system faults should they occur. This thesis considered relatively simple faults, but faults involving complex heating mechanisms and very slow-acting drift faults pose difficult challenges for air data system reliability. In order to provide true fault tolerance for air data systems, approaches to these problems must be explored. Perhaps the most important short-term goal is to enhance the fault detection algorithms so that estimates are more accurate at off-design conditions. This thesis has considered fault detection performance in a small piece of the flight envelope with relatively small aircraft maneuvers. Real aircraft undergo far more extreme maneuvers over a wide flight envelope, and extending these results to those scenarios is required. Using linear parameter-varying techniques has shown promise in other applications and would potentially provide a stronger understanding of fault tolerance implications at all parts of the flight regime.

# Bibliography

- [1] Bleeg, R., “Commercial jet transport fly-by-wire architecture considerations,” *AIAA/IEEE Digital Avionics Systems Conference*, 1988, pp. 399–406.
- [2] Collinson, R., *Introduction to Avionics Systems*, Kluwer, 2nd ed., 2003.
- [3] Yeh, Y., “Triple-triple redundant 777 primary flight computer,” *Proceedings of the 1996 IEEE Aerospace Applications Conference*, 1996, pp. 293–307.
- [4] Yeh, Y., “Design Considerations in Boeing 777 fly-by-wire computers,” *Third IEEE International High-Assurance Systems Engineering Symposium*, 1998, pp. 64–72.
- [5] Aviation Safety Network, “Accident Database Record,” <http://aviation-safety.net/database/record.php?id=19991017-0&lang=en>, 2011.
- [6] ACC News Service, “B-2 accident report released,” <http://www.af.mil/news/story.asp?id=123101727>, 2008.
- [7] Bureau d’Enquêtes et d’Analyses, “Accident to the Airbus A330-203 Flight AF 447 on 1st June 2009: Update on Investigation,” 2011.
- [8] Ambrose, R., Wilcox, B., Reed, B., Matthies, L., Lavery, D., and Korsmeyer, D., “Robotics, Tele-Robotics, and Autonomous Systems Roadmap: Technology Area 04,” Tech. rep., NASA, 2010.
- [9] Adler, M., Wright, M., Campbell, C., Clark, I., Engelund, W., and Rivellini, T., “Entry, Descent, and Landing Roadmap,” Tech. rep., NASA, 2010.
- [10] Gertler, J. J., *Fault detection and diagnosis in engineering systems*, Marcel Dekker, 1st ed., 1998.

- [11] Isermann, R., *Fault-diagnosis systems: an introduction from fault detection to fault tolerance*, Springer, 1st ed., 2005.
- [12] Ding, S. X., *Model-based Fault Diagnosis Techniques: Design Schemes, Algorithms, and Tools*, Springer, 1st ed., 2008.
- [13] Goodrich, *Air Data Handbook*, Goodrich Corporation – Sensors and Integrated Systems, 2002.
- [14] Administration, F. A., *Advisory Circular 91-85 (Authorization of Aircraft and Operators for Flight in Reduced Vertical Separation Minimum Airspace)*, U.S. Department of Transportation, 2009.
- [15] Murch, A. and Foster, J., “Recent NASA research on aerodynamic modeling of post-stall and spin dynamics of large transport airplanes,” *45th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada*, No. AIAA-2007-463, 2007.
- [16] Cox, D., *The GTM DesignSim v1006*, 2010.
- [17] Stevens, B. and Lewis, F., *Aircraft Control and Simulation*, John Wiley & Sons, 1992.
- [18] Dorobantu, A., Murch, A., and Balas, G., “H-Infinity Robust Control Design for the NASA AirSTAR Flight Test Vehicle,” *50th AIAA Aerospace Sciences Meeting, Nashville, TN*, 2012.
- [19] Simon, D., *Optimal State Estimation: Kalman, H-infinity, and Nonlinear Approaches*, John Wiley & Sons, 2006.
- [20] Skogestad, S. and Postlethwaite, I., *Multivariable Feedback Control*, John Wiley and Sons Ltd., 2007.
- [21] Doyle, J., Francis, B., and Tannenbaum, A., *Feedback Control Theory*, Macmillan Publishing Co., 1990.
- [22] Varga, A., “On computing least order fault detectors using rational nullspace bases,” *In Proceedings of the IFAC Symp. SAFEPROCESS’2003, Washington D.C.*, 2003.
- [23] Varga, A., “General computational approach for optimal fault detection,” *SAFE-PROCESS ’09, Barcelona, Spain*, 2009.

- [24] Marcos, A., Ganguli, S., and Balas, G., “An application of H-infinity fault detection and isolation to a transport aircraft,” *Control Engineering Practice*, Vol. 13, No. 1, 2005, pp. 105–119.
- [25] Ding, S. X., *Model-based Fault Diagnosis Techniques*, Springer, 2008.
- [26] Zolghadri, A., Castang, F., and Henry, D., “Design of robust fault detection filters for multivariable feedback systems,” *International journal of modelling and simulation*, ACTA Press, Vol. 26, No. 1, 2006.
- [27] Mazarsand, E., Jaimoukha, I., and Li, Z., “Computation of a Reference Model for Robust Fault Detection and Isolation Residual Generation,” *Journal of Control Science and Engineering*, Vol. 2008, 2008, pp. 1–12.
- [28] Balas, G., Chiang, R., Packard, A., and Safonov, M., *Robust Control Toolbox*, The MathWorks, 2010.
- [29] Doyle, J., Glover, K., Khargonekar, P., and Francis, B., “State-space solutions to standard H-2 and H-infinity control problems,” *IEEE Transactions on Automatic Control*, Vol. 34, No. 8, 1988, pp. 831–847.
- [30] Szaszi, I., Marcos, A., Balas, G., and Bokor, J., “LPV Detection Filter Design for a Boeing 747-100/200,” *Computer and Automation Research Institute*, 2002.