Dynamic Flight Envelope Assessment with Flight Safety Applications

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

Rohit Pandita

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPY

Prof. Gary J. Balas, Adviser

December 2010

© Rohit Pandita 2010

Acknowledgments

I would like to thank my adviser Prof. Gary Balas for providing this opportunity and an outstanding research environment. It was my privilege to work with him for last five years. I was fortunate enough to see his extraordinary compassion and human touch.

I would also like to thank my committee members Prof. Demoz Gebre-Egziabher, Prof. Mihailo Jovanovic and Prof. Bernie Mettler for their excellent feedback and valuable suggestions. Prof. Gebre was particularly kind and welcoming to share his wisdom and advice whenever I sought it.

I cannot thank Dr. Pete Seiler enough for sharing his impressive insight in control theory and many fruitful technical discussions we've had. He played a pivotal role in helping me understand the nuances of sum-of-squares theory. Large part of this research would have been impossible without the SOSTOOLS toolbox that he co-authored.

I would like to thank Prof. Jozsef Bokor for the opportunity to work with him on the FDI problem. I admire his kindness and knowledge of FDI filter theory in equal measure.

Thank you Paw, Abhijit and Vijay for being awesome badminton buddies. It was great to let off some steam on the badminton courts. I would like to thank my lab mates and Arnar in particular for their friendship and company.

UAV group members, past and present played important role in my research. I would like to thank Zhefeng, David, Troy, Austin, Arion and Garrison for their support.

Greatest of all, I would like to thank my family for constant support, unconditional love and numerous sacrifices they've made to make this day happen. My mother has always been a piller of strength and it is to her that I dedicate this dissertation.

This dissertation was partially supported by NASA Langley NRA contract NNH077ZEA001N

entitled “Analytical Validation Tools for Safety Critical Systems” and the NASA Langley NNX08AC65A contract entitled ’Fault Diagnosis, Prognosis and Reliable Flight Envelope Assessment.”

Dedicated to my mother.
For I've not known god, but I've known you.

Abstract

Aircraft have a manufacturer prescribed operating flight envelope for safe operation. exceeding these limits can result in unrecoverable departures or even structural failure. Numerous commercial aircraft accidents in the past have been attributed to loss-of-control (LOC) resulting from exceeding the safe operating flight envelope. Hence, real-time knowledge of the safe operating flight envelope is essential for safe flight operation, a problem known as dynamic flight envelope assessment. This dissertation explores dynamic flight envelope assessment from a control theoretic perspective. Two notions of the flight envelope, namely, the reachable sets and the region-of-attraction analysis are investigated. The NASA generic transport model (GTM) aircraft dynamics is used as an application problem. Linear and nonlinear techniques for flight envelope assessment are formulated in the linear matrix inequality (LMI) and sum-of-squares (SOS) framework, respectively. LMI and SOS problems are computationally tractable convex optimization problems for which many semi-definite programming solvers are available.

This thesis also investigated fault detection and isolation strategies. Commercial jet transport aircrafts make extensive use of active controls. Faults or failures in the flight control system (FCS) elements like sensors or control effectors can lead to catastrophic failure. Model-based fault detection and isolation (FDI) filters can provide analytical redundancy by reliably detecting such faults in the system. Practical application of model-based FDI filters is limited so far due to poor performance, false alarms and missed detection arising out of uncertain dynamics of the aircraft, effect of nonlinearities in the system and the influence of closed-loop controllers. An application of closed-loop metrics to assess worst case FDI filter performance in the presence of a controller and uncertain dynamics is presented. Longitudinal GTM dynamics are considered. An $\mathcal{H}_\infty$ FDI filter and a geometric filter design are compared using the metrics and the results validated through simulation.

This research was expanded to include synthesis, real-time implementation and flight validation of robust FDI filters for a small uninhabited aerial vehicle. The influence of different closed-loop controllers on FDI filter performance is investigated. A comparison is presented between simulation of the predicted FDI filter performance and flight experiment results.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Commercial aircraft safety has come a long way since the days of first jet commercial passenger flight in the 1950s. Today air-transport is considered the safest mode of transport. This remarkable result is unquestionably due an excellent partnership that exists between industry, government, various regulatory instutions, academia and others towards a relentless quest of continuous safety improvement. In the 1960s the accident rate among scheduled passenger flights was around 15 accidents per million departures to currently around 1 per million departures, [5]. Nevertheless, aviation community formed an inter-organization action group called Commericial Aviation Safety Team (CAST) to further improve commercial aircraft safety. CAST is a collaboration of major organizations sharing a common aviation safety mission, to reduce the commercial aviation accident rate 80% over a ten-year period ending 2007. CAST includes the Federal Aviation Administration (FAA), the National Aeronautics and Space Administration (NASA), and the Department of Defense (DOD), representing government, and many organizations representing the aviation industry. Among those organizations are airplane, engine, and avionics manufacturers; Part 121 certificate holders (airlines); their trade organizations, such as Air Transport Association (ATA), Regional Airline Association (RAA), and National Air Carrier Association (NACA); and pilots associations such as the Air Line Pilots Association (ALPA) and the Allied Pilots Association (APA). The backbone of the CAST approach was statistical data-driven analysis of aviation safety records so that problem areas can be positively identified and subject to improvement.

By 2007, CAST had successfully achieved its objective. In this ten year-time period the industry saw new technology being introduced like the ground proximity warning sys-

tem (GPWS), traffic collision avoidance system (TCAS), etc. This period was also known for rapid adoption of newer aircrafts with automated flight-decks and glass cockpit which helped improve flight-crew situational awareness (SA) and reduce pilot workload.

A major source of fatal accidents in the past was what's known as controlled flight into terrain (CFIT), or when a perfectly flyable aircraft is unwittingly flown into terrain. With the introduction of GPWS, CFIT incidents all but dissapeared. Today over 95% of commercial *jet* aircrafts carry GPWS equipment and none have been involved in CFIT type of accidents. To get a perspective on the current scenario, consider this, In 2008, out of 15 commercial jet accidents none were related to CFIT, 5 were due to loss of control (LOC). This figure was 4 for 2007. At the same time in 2008, commercial non-jet transport (turboprops) suffered 17 fatal accidents out of which 6 were due to CFIT [1]. Incidently, GPWS like system is not mandatory on turboprops as of now. In conclusion, while commercial jet transport safety has improved considerably over the years, but now LOC has taken over as the single most important source of fatal accidents. In fact according to the UK civil aviation authority, [16], LOC is the only type of accident which is not reducing in world-wide statistics. Rightfully, the focus of all stake holders has shifted to reducing LOC incidents.

Loss of control (LOC) accident sub-type refers to incidents involving inability to maintain positive control of aircraft states/flight parameters. LOC are further divided into inflight or on ground LOC, but in this thesis LOC exclusively refers to LOC infight.

Fig. 1.1 shows the statistics for all commercial jet transport accidents during 1999-2008. It shows that out of the 16 accident catagories identified by CAST, LOC was the leading cause, responsible for 24% of total number of accidents and astounding 41% of inflight fatalities.

Loss of control accidents can be attributed to the following causes,

- System fault/failure: When a fault or failure in aircrafts flight control system or airframe lead to LOC. Examples include sensor faults, primary control surface damage, autopilot system failure, etc.
- Environmental factors: Weather phenomenon like turbulence, microburst, wake turbulence, icing, etc. Environmental factors often lead to what's known as *upset-condition* wherein an aircraft in trimmed flight is disturbed from its equilibrium condition and may require corrective control inputs to restore flight state.

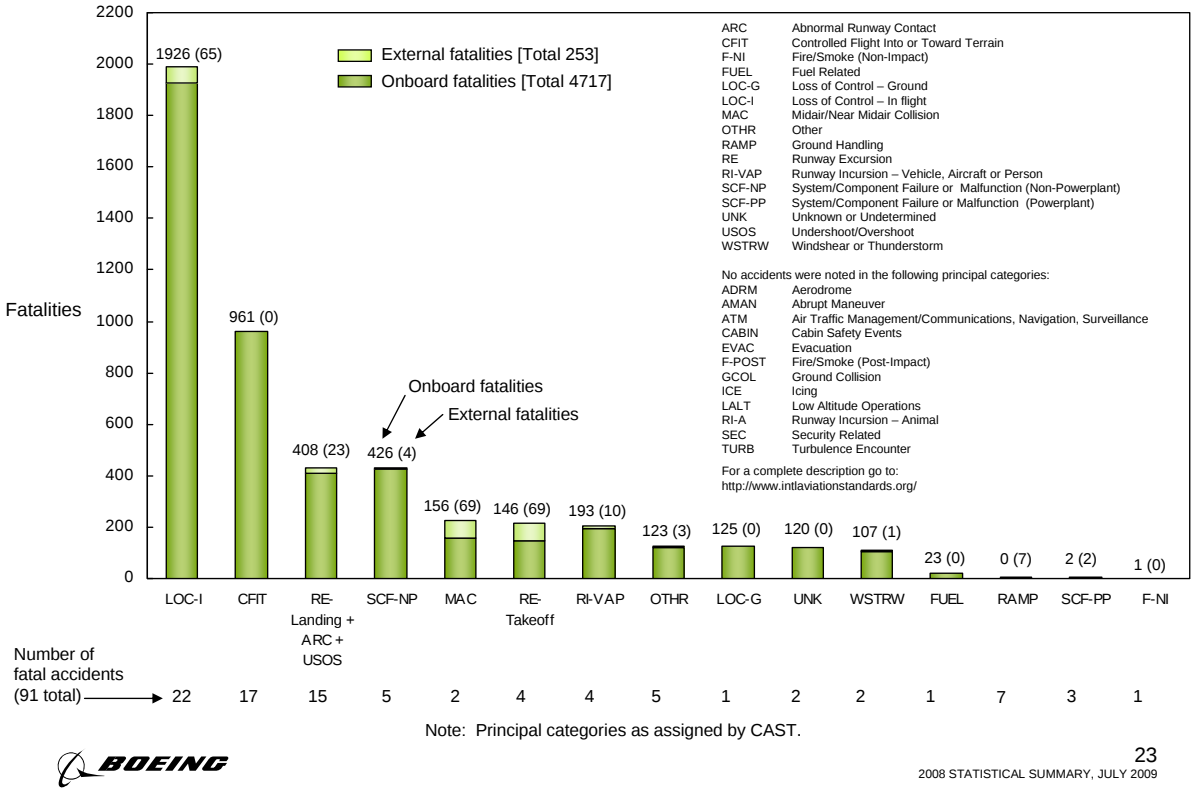


Figure 1.1: Source:http://www.boeing.com/news/techissues/statsum.pdf

- Human error: Human factors are the ones where human pilot by action or inaction can lead to unrecoverable adverse flight conditions, possibly in response to the first two causes of LOC.

In 2000, CAST chartered the Loss of Control (LOC) Joint Safety Implementation Team (JSIT) to develop a practical implementation agenda. The LOC JSITs mission would be to prioritize, select, and implement the interventions recommended by the LOC JSAT that would have the highest possible safety benefits. The LOC JSIT would take into account all major practical constraints, such as technical feasibility, time, and cost. The JSIT report [3] made recommendations based on two factors - potential safety impact and feasibility of the proposed technology in future aircraft. Below are excerpts of some high safety impact technologies recommended in the report [3]:

1. To ensure that the display of conflicting attitude information does not confuse or mis-

lead the flight crew, the attitude sensing and display systems should be designed so that invalid information is detected and clearly annunciated to the flight crew and/or removed from display.

2. To help avoid loss of control, manufacturers should develop and implement flight envelope protection.

3. To prevent loss of control, there should be redundancy and failure tolerance features for all flight critical components, such as dual path design, fail operational redundant systems, with fault annunciation.

4. To warn of impending loss of control with the autoflight system require annunciation of an airplane flight condition which significantly differs from that being commanded by the selected autoflight system.

5. Because some partial system failures may not be reliably observed, manufacturers should develop warning systems that alert the pilot to those partial/total failures of flight critical systems and flight instruments.

This wishlist throws a light on the technologies industry is seeking to address the LOC problem. This in part motivates the research work undertaken in this thesis.

Another important aspect of the LOC equation, the human factors, has gained lot of attention in the recent years in the wake of some very recent fatal accidents. Especially, pilot response to upset conditions are being looked at very carefully. Some of the accident case studies shall be discussed shortly in this light. Human factors in an increasingly automated flight-deck environment has been focus of attention for some time. According to an article published in 1999, [6], flight deck automation and flight control technology, coupled with excellent systems reliability and redundancy, allowed flight crews to easily control their aircraft from takeoff to landing regardless of outside visibility. However, if an anomaly occurs, the complex systems that automate, control and display information in modern flight decks can produce erroneous or insufficient information. When faced with the resulting uncertainties, flight crews must determine what information is reliable and what information should not be used in order to make the proper decisions. The article further stated that, unfortunately, safety data shows that not all flight crews have satisfactorily handled situations caused by erroneous flight instrument information. From October 1988 to 1999, more that 300 accidents and incidents had been reported as a result of erroneous flight deck

information, including problems with pitot-static probes and air data computers. Several fatal accidents that involved erroneous flight instruments information and six occurrences resulting from lost or erratic air data occurred in 1996 alone. Investigations of those occurrences indicated that, with proper preparation, the flight crews involved probably could have prevented them. Unfortunately, some of the very recent accidents still bear uncanny similarities.

Upset recovery is key to successful mitigation of LOC accidents. There are two aspects to upset recovery. One relates to corrective pilot actions which boils down to pilot training. The airline industry, FAA and NASA have recognised this problem and in 2009 have embarked on an ambitious project to study the current piloting skills of line pilots and to formalize upset recovery training (URT) as a part of standard practice, [7]. The other aspect relates to the systems side, which is of interest to this research work. There is a need to correctly identify and annunciate in realtime when the aircraft is approaching the limits of its flight envelope, which means *knowing* the flight envelope of an aircraft. Current generation aircraft envelope systems impose hard limits on permissible the aerodynamic angles, and aircraft attitude, airspeed, etc based on *design* capabilities of an aircraft. Such system are incapable to taking into account any degradation that might occur due to ageing, lifting surface/ control surface damage, etc and how that might affect the *safe flight envelope*. This is the second motivation for this thesis.

NASA has been very active to address the aviation safety related challenges through the Aviation Safety Program (AvSP) [72] in the past and currently through the Integrated Resilient Aircraft Control (IRAC) program which has partially supported this thesis.

1.1 Loss of control (LOC) incidents: Some case-studies

It has become clear that in the last decade that the main source of fatal aircraft accidents has shifted to the LOC sub-type. Below are some recent examples that hightlight the case and also bring out potential areas of improvement.

- *Boeing 777, Air Malasyia, near Perth, Australia, 2005*: The airdata and inertial reference unit (ADIRU) suffered an inflight fault in the accelerometer . Despite sensor hardware redudancy, software passed spurious data to control laws leading to rapid uncommanded pitch up at 38000ft. The stick shaker and stall warning activated. autopilot (A/P) was manually disengaged. Both A/P and autothrottle (A/T) functions

had failed. Aircraft landed safely, some injuries were reported.

Synopsis: Fault detection and isolation (FDI) algorithms could have potentially helped in detecting the sensor failure.

- *Tu-154, Pulkovo Aviation Enterprise, Ukraine, 2006*: The Tu-154 entered an area of severe turbulence, pushing up the airplane from 11,961 m to 12,794 m within just 10 seconds. The angle of attack (AOA) increased to 46 degrees and the airspeed dropped to zero. It entered a deep stall from which the crew could not recover.

  Synopsis: Recovery is not guaranteed but flight crew could have been made aware of rapidly approaching limits of the flight envelope.

- *Airbus A320, XL Airways, Perpignan, France, 2008*: The aircraft was on a check flight before handover to the owner airline. The pilots decided to check AOA protection system at low altitude. AOA vane 1 and 2 were frozen/stuck due to water penetration and AOA protection was unavailable as a result (unknown to the pilots). Aircraft stalled, application of full power to regain control increased pitch up tendency due to moments from underslung engines, pilots were unable to recover the aircraft and it crashed in the sea killing all onboard.

  Synopsis: Flight control system (FCS) and the flight crew failed to detect the sensor fault. An FDI system could have annunciated the failure, and made flight crew aware of the unavailability of envelope protection system.

- *Airbus A330-300 Qantas, near Learmonth, Australia, 2008*: ADIRU fault, spurious AOA spikes A/P disengaged at 37000ft. Abrupt pitch down motion ensued. AOA faults were not caught by the software.

  Synopsis: FDI problem.

- *Boeing 737-800 Turkish Airlines, Amsterdam, The Netherlands, 2009*: On final approach, one of the radar altimeter failed and read close zero altitude, A/T system proceeded to retard the throttles assuming touchdown. Flight crew failed to pay attention. Aircraft stalled and crash landed in a field.

  Synopsis: The second radar altimeter and barometric altimeter were functional. An FDI system could have been helpful in isolating the fault and inform flight crew.

1.2 Control theory perspective

Based on the brief synopsis of the LOC accidents in commercial jet transport sector we find that some promising tools exist in the control literature to help mitigate some of the problems discussed above. Sum-of-squares (SOS) tools allow compuational capability to analyize nonlinear systems. Region-of-attraction (ROA) analysis using SOS methods can be used to ascertain the *safe-set* for an aircraft at an equilibrium. This could be used to not only inform flight crew of the aircraft's operating point in realtime, but also recompute the safe-set following a damage or failure of a dynamical element of the system. Flight crew can be made aware of any dangerous proximity to safe-set boundaries so that preventive action could be taken instead of reactionary recovery tactics, which fail often times.

Similarly, the notion of *Reachable-sets* offers important insight into the controllable subspace and how it changes following damage or failure in the system. This can be useful to the pilots or the A/P system to be aware of limits of the available control authority.

Finally, model-based FDI filters offer a powerful means to exploit knowledge of the system dynamics to detect and even estimate faults in sensors and actuators/ controls used for primary control of the aircraft. This knowledge can be used to warn pilots of any serious failures that might otherwise go unnoticed and lead to potential disaster, as brought out by some of the case studies given above.

1.3 Contributions

The contributions of this thesis include:

1. Developed polynomial and LPV models for the generic transport model (GTM) aircraft for use in analysis.
2. Used SOS methods to compute the region-of-attraction (ROA) or safe-set for the nonlinear GTM longitudinal dynamics model.
3. Formulate and computationally solve reachable-sets problem for linear as well as nonlinear GTM model for input-energy and peak-input constraints. Develop methods to compute reliable upper and lower bounds for reachable-sets.
4. Enhanced the fault detection metrics to estimate the FDI filter performance in closed-loop systems first propsed in [63]. Used it to successfully predict worst-case filter performance of an $\mathcal{H}_\infty$ and a geometric FDI filter design for the GTM model.

5. Developed an infrastructure for low-cost flight research using small UAVs. Specific contributions include developement of multithreaded realtime software framework for the onboard flight program (OFP), software-in-the-loop (SIL) and hardware-in-the-loop (HIL) verification and validation (V&V) framework for algorithms. This work is not discussed in this thesis, but it is believed to be a very useful new resource to the current and future researchers in the AEM systems group conduct low-cost flight experiments.

6. Synthesized and flight validated an FDI filter to detect sensor and actuator fault using the small UAV infrastructure described above.

1.4 Organization and Scope

The first chapter preents basic mathematical notions and definitions that are used extensively in the later chapters. Chapter 2 outlines the developement of a polynomial longitudinal model to approximate full nonlinear longitudinal dynamics of the generic transport model (GTM) aircraft for use in analysis. Linear parameter varying (LPV) model for the GTM is also developed. Chapter 3 hightlights the importance of reachable-sets in dynamic flight envelope estimation. Linear and nonlinear analysis methods are developed and their LMI and SOS formulations presented, respectively. Numerial examples with the GTM dynamics are presented. Chapter 4 discusses developement of region-of-attraction (ROA) analysis techniques using SOS methods for nonlinear systems with applications. Chapters 5 and 6 focus of the fault diagnosis problem. In Chapter 5, closed-loop FDI filter performance metrics are presented with GTM example. An $\mathcal{H}_\infty$ FDI filter design and flight validation using small UAV infrastructure is presented in the Chapter 6. Thesis concludes with the summary and recommendations for future directions.

Chapter 2

Mathematical preliminaries

Various mathematical notions and definitions used in this thesis are briefly discussed in this chapter.

2.1 Convex functions

A function $f(x) \in \mathcal{C}^1$ is said to be convex on interval $[a, b]$ if for any two points x_1 and x_2 in $[a, b]$ and for any t s.t. $0 < t < 1$,

$$f[tx_1 + (1-t)x_2] \leq tf(x_1) + (1-t)f(x_2) \tag{2.1}$$

$\mathcal{C}^1$ is a set of functions with continuous first order derivatives.

2.2 Linear Matrix Inequalities (LMI)

Linear matrix inequalities (LMIs) are inequalities of the form,

$$G(x) = G_0 + G_1x_1 + G_2x_2 + \ldots + G_nx_n > 0 \tag{2.2}$$

where, $x_i \in \mathbb{R}^m, i = 1, ..., n$ are the variables and $G_i = G_i^T \in \mathbb{R}^{m\times m}, i = 1, ..., n$ are symmetric matrices.

Some key properties of LMIs,

1. The set $\{x \mid G(x) > 0\}$ is convex.

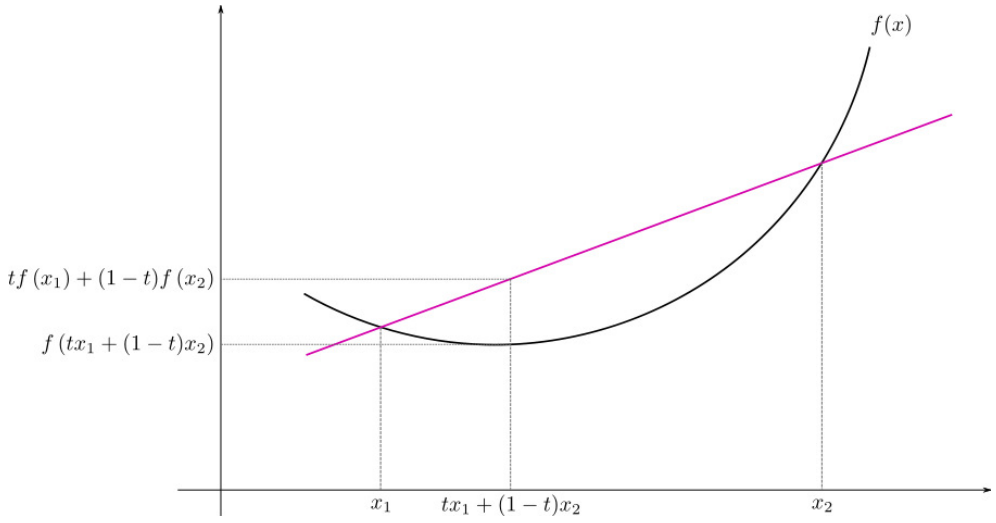


Figure 2.1: A convex function

2. Nonlinear convex inequalities like Algebraic Riccati equations can be converted to LMI by using Schur complement lemma,

$$\begin{aligned} V(x) > 0, U(x) - S(x)V(x)^{-1}S(x)^T > 0 \\ \Leftrightarrow \begin{bmatrix} U(x) & S(x) \\ S(x)^T & V(x) \end{bmatrix} > 0 \end{aligned} \tag{2.3}$$

   where, $U(x) = U(x)^T, V(x) = V(x)^T, S(x) = S(x)^T$ are affine in x.

3. Can be solved efficiently in *polynomial time* using semidefinite programming (SDP) solvers.

LMIs can be classified into three general categories:

1. **LMI feasibility problem (LMIP)**: For an LMI $G(x) > 0$, finding whether the inequality holds for some $x = x_{feas} \in \mathcal{X}$ is known as the LMI feasibility problem or LMIP for short. Many control problems can be reduced to LMIP, for example Lyapunov stability question for an linear time-invariant (LTI) system is an LMIP, $A^T + PA < 0, P > 0$. It may be noted that there is no cost function associated with LMIP, a unique solution isn't being sought.

2. **Eigenvalue problem (EVP)**: Problems of the form,

$$\begin{aligned} \min \quad & \lambda \\ \text{s.t.} \quad & \lambda I - G(x) > 0, H(x) > 0 \end{aligned}$$

which essentially entails minimizing the maximum eigenvalue of the matrix $A(x)$ affine in x, subject to an LMI constraint $H(x) > 0$.

3. **Generalized Eigenvalue problem (GEVP)**: A problem of the form,

$$\begin{aligned} \min \quad & \lambda \\ \text{s.t.} \quad & \lambda B(x) - A(x) > 0, B(x) > 0, C(x) > 0 \end{aligned}$$

where $A(x) = A(x)^T, B(x) = B(x)^T, C(x) = C(x)^T$ are all affine in x. The GEVP is equivalent to minimizing the maximum generalized eigenvalue of a pair of matrices linearly dependent on x, and an LMI feasibility constraint $C(x) > 0$.

The types of LMI problems discussed above can be solved efficiently by SDP solvers. Efficient algorithms are available solving convex optimization problems of this type like the interior point methods, [13].

2.3 Hyper-ellipsoids: Optimal approximation

One LMI problem that will be useful in this thesis is the optimization over a set of feasible ellipsoids for "maximum size". Reachable sets ($\mathcal{R}$) for dynamical systems can be represented as hyper-ellipsoids in $\mathbb{R}^n$ in the n-dimensional state-space,

$$\mathcal{R} := \{z \in \mathbb{R}^n \mid z^T P z < 1, P = P^T > 0\}$$

Symmetric positive definite matrix $P > 0$ defines the shape of the ellipsoid. It can be shown that axial dimensions of the ellipsoid are scaled by the eigenvalues of the matrix P. The *size* of an ellipsoid is a slightly ambiguous term. In the context of reachable sets, since we are trying to find the *largest* set for a given system, maximizing the volume of an ellipsoid or maximizing the sum of magnitudes of the semi-principal axes are two meaningful metrics. The most important property of these two metrics is that they are convex functions of the decision variables as will be explained further,

- **Maximal ellipsoid volume**: The volume (V) of an ellipsoid($\mathcal{R}$),

$$V \propto \frac{1}{\sqrt{\det P}}$$

or equivalently, the following LMI problem can be formulated to find the maximal reachable set, with some additional LMI feasibility constraint, $B(x) > 0$.

$$\begin{aligned} \max \quad & \log \det P^{-1} \\ \text{s.t.} \quad & P(x) > 0, B(x) > 0 \end{aligned}$$

It can be shown that $\log \det P^{-1}$ is a convex function. This problem cannot be directly cast into any of the three standard LMI problems described before. A special class of LMI solvers are available to solve such problems , [90, 52]. This type of problem cannot be solved directly in tools like MATLAB LMIlab, [31].

- **Maximal sum of semi-principal axes**: A simpler approach than the one described above is to maximize the sum of magnitudes of the semi-principal axes of an ellipsoid. It may be noted that volume (V) of an n-dimensional ellipsoid, $V = \frac{4}{3}\pi\sigma_1\sigma_2\ldots\sigma_n$, where $\sigma_i, i = 1, \ldots, n$ are magnitudes of the semi-principal axes.

  It can be shown that in fact $\sigma_i = \lambda_i, i = 1, n$, where λ_i are eigenvalues of the matrix P. Using the trace property of a matrix, i.e. $\sum_{i=1}^{n} \lambda_i = \text{Tr}P$, where $\text{Tr}P = P_{11} + P_{22} + \ldots + P_{nn}$ is the trace of matrix P.

  Leading to the following LMI problem which is equivalent to LMI EVP. This problem can be convenient solved using any LMI solver including the MATLAB LMI toolbox. Although the two metrics for reachable set optimization described in this section are not identical, they provide a reasonable measure for reachable set computation.

$$\begin{aligned} \max \quad & \text{Tr}P \\ \text{s.t.} \quad & P(x) > 0, B(x) > 0 \end{aligned}$$

2.4 The $\mathcal{S}$-procedure

The $\mathcal{S}$-procedure is a useful mathematical trick to prove two or more inequalities through a single inequality. To understand the principle, consider the case of two quadratic forms, $z^T Q_1 z$, $z^T Q_2 z$, $z \in \mathbb{R}^n, Q_1 = Q_1^T, Q_2 = Q_2^T$. When does the following inequality hold simultaneously?

$$z^T Q_1 z \geq 0, z^T Q_2 z \geq 0$$

A simple test is, if $\exists\ s \in \mathbb{R}, s \geq 0$ such that $Q_1 \geq sQ_2 \Rightarrow z^T Q_1 z \geq 0$ when $z^T Q_2 z \geq 0$.
Proof: If $Q_1 \geq sQ_2$ then $z^T Q_1 z \geq z^T Q_2 z$ is also true. Thus we have $z^T Q_2 z \geq 0 \Rightarrow z^T Q_1 z \geq 0 \quad \square$.

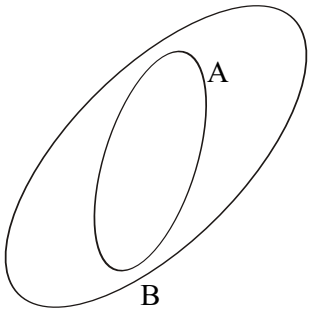


Figure 2.2: Set-containment

The procedure can be generalized for any two inequalities. Let $f : \mathbb{R}^n \mapsto \mathbb{R}$, $f = \{f_A, f_B\}$.

Lemma 2.4.1 ($\mathcal{S}$-procedure)**.** *If* $\exists\ s(x) : \mathbb{R}^n \mapsto \mathbb{R}, x \in \mathbb{R}^n$*, s.t.* $s(x) \geq 0\ \forall\ x$ *and,*

$$f_B(x) - f_A(x)s(x) \geq 0,\ \forall x$$

then,

$$f_A(x) \geq 0 \Rightarrow f_B(x) \geq 0$$

2.5 Set containment lemma

One property we will have to prove often in SOS formulation of ROA or reachability problems is the set-containment, or to prove one set is a subset of another. This problem can be posed as an SOS problem using modified $\mathcal{S}$-procedure.

Consider two mutually exclusive sets, A and B defined as,

$$A := x \in \mathbb{R}^n \mid f_A(x) \geq 0 B := x \in \mathbb{R}^n \mid f_B(x) \geq 0$$

Lemma 2.5.1 (Set-containment lemma)**.** *If* $\exists\ s : \mathbb{R}^n \mapsto \mathbb{R}$*, a multiplier, s.t.*

- $s(x) \geq 0\ \forall x$
- $f_B(x) - f_A(x)s(x) \geq 0,\ \forall x$

Then, $A \subseteq B$.

Proof: Let $x \in A$, such that the above two positive semi-definiteness conditions hold, then the following is also true,

$$\begin{aligned} &\Rightarrow f_A(x) \geq 0 \\ \Rightarrow f_B(x) \geq f_A(x)s(x) &\geq 0 \\ &\Rightarrow x \in B \end{aligned}$$

□

2.6 Sum-of-Squares (SOS) polynomials

A polynomial is said to be sum-of-squares (SOS) if it can be represented as a sum of square of some monomials. This notion can be applied to prove sign definiteness of polynomial functions, which is very useful to applications like Lyapunov stability, etc. Before going into details, it is necessary to define quantities that will be referred to often in the subsequent material.

Definition *Field*: A field is an algebraic structure permitting notions of addition, multiplication and division, and satisfying the *field axioms*, namely, associativity, commutativity, distributivity, identity and inverses. Example: Complex numbers ($\mathbb{C}$), real numbers ($\mathbb{R}$), rational numbers ($\mathbb{Q}$).

Definition *Vector field*: A map $f : \mathbb{R}^n \mapsto \mathbb{R}^n$.

Definition *Polynomial*: A mathematical expression consisting of a sum of powers ($\in \mathbb{N}$) of one or more variables with constant coefficients known as *monomials*. The highest exponent-sum of a monomial is known as the *degree* of a polynomial. Example: $p(x, y) = 2x^3 + y^2x - 5xy$, degree ($\partial(p)$)$= 3$.

The basic SOS question to be asked is, given polynomial a $p(x)$, is $p(x) \geq 0 \; \forall \; x$? This question is the most fundamental SOS problem. Historically, it has not been an easy one to answer. Infact, it was notorious enough to feature on the famous list of *Hilbert's twenty-three problems* announced at the turn of 20th century. This was the 17th problem on the list

> *Given a multivariate polynomial that takes only non-negative values over the reals, can it be represented as a sum of squares of rational functions?*

The problem was finally solved in the affirmative by Austrian mathematician Emil Artin in 1927.

In summary, the *sufficient condition* for a polynomial $p(x)$ to be positive semidefinite (PSD) is,

Theorem 2.6.1 (Sum-of-Squares)**.** *If* $p(x) = \Sigma_{i=1}^{n} f_i^2 \Rightarrow p(x) \geq 0 \; \forall \; x$

The converse is not necessarily true. A counterexample is the *Motzkin polynomial*,

$$\begin{gathered} M(x, y, z) = x^4y^2 + x^2y^4 + z^6 - 3x^2y^2z^2 \\ M \in PSD, M \notin SOS \end{gathered} \tag{2.4}$$

2.6.1 Polynomials as vector field

Consider a polynomial

$$p = \sum_{\alpha \in \mathcal{A}} C_\alpha X^\alpha$$

where,

$$X^\alpha = X_1^{\alpha_1}, X_2^{\alpha_2}, \ldots, X_n^{\alpha_n}$$

$$\Rightarrow p = \begin{bmatrix} X_1^{\alpha_1} & X_2^{\alpha_2} & \ldots & X_n^{\alpha_n} \end{bmatrix} \begin{bmatrix} C_{\alpha_1} \\ C_{\alpha_2} \\ \vdots \\ C_{\alpha_n} \end{bmatrix}$$

Thus, $p(x)$ can be viewed as a vector space with $\boldsymbol{X}^\alpha$ as the basis. Some properties about the polynomial vector field basis representation. Given a polynomial of degree d in n variables:

- There are $l_z = {}^{n+d}C_d$ monomials of $\partial \leq d$ and n variables that form the complete basis to represent any polynomial of degree d and n variables.
- Set of polynomials of degree d and in n variables is a vector space $z \in \mathbb{R}^{l_z}$.
- Every polynomial of degree d and in n variables has the form $p = z^T c$, where $c \in \mathbb{R}^{l_z}$ is a vector of constants.

2.6.2 The Gram Matrix representation

Let p be a polynomial with degree $2d$. Let $z \in \mathbb{R}^{l_z}$ be a vector of all the monomials of n variables and degree $\leq d$. Then, $p = z^T Q z, Q = Q^T$. Q is not unique. Example,

$$p = 5x_1^2 - 6x_1x_2 + 3x_2^2$$

$$= \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 5 & -3 \\ -3 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Theorem 2.6.2 (SOS)**.**

$$\begin{gathered} p \in SOS \\ \textit{iff} \\ \exists\, Q = Q^T \geq 0 \textit{ , such that } p = z^T Q z \end{gathered} \tag{2.5}$$

Proof.

$$(\Leftarrow) p = z^T Q z, Q \geq 0$$

Since, $Q \geq 0$, Q has a Cholesky decomposition, $Q = L^T L$. Therefore,

$$p = (Lz)^T (Lz)$$

$$(\Rightarrow) p \in \Sigma[x] \exists \{f_i\} \text{s.t.} p = \sum_i f_i^2 = \sum_i (c_i z)^2.$$

where c_i is a coefficient vector and z is the vector of monomials of degree $\leq d$.

$$\Rightarrow p = (cz)^T (cz) = z^T \underbrace{c^T c}_{Q \geq 0} z$$

□

Since the Gram-Matrix decomposition is not unique, a polynomial $p = z^T Q z$ can be written as,
$p = z^T Q_0 z$, a particular solution, and $0 = z^T Q_i z$, the homogeneous solutions. So given

polynomial p, every Q satisfying $p = z^T Q z$

$$\Rightarrow Q = Q_0 + \sum_i \lambda_i Q_i$$

Theorem 2.6.3 (SOS LMI Primal)**.**

$$\begin{gathered} p \in SOS \\ \textit{iff} \\ \exists\, Q_i = Q_i^T \geq 0, \lambda_i \textit{ , s.t. } Q_0 + \Sigma \lambda_i Q_i \geq 0 \end{gathered} \tag{2.6}$$

This is an LMI feasibility problem.

2.6.3 Matrix representation of the SOS theorem

In this section an alternate representation of the SOS theorem is presented. This representation is amenable to semidefinite programming.

Let $p(x)$ be a polynomial of degree $2d$ and in n variables (notation $p_{n,2d}$ shall be used subsequently). Let $z(x) \in \mathbb{R}^{l_z}$ be a basis vector of all monomials of degree $\leq d$ and in n variables such that $p(x) = z(x)^T Q z(x)$. Defining an operator $L(Q)$,

$$L(Q) = z(x)^T Q z(x)$$

$$L : \mathbb{R}^{l_z \times l_z} \mapsto p_{n,2d}$$

$L(Q)$ is a linear operator, i.e. $L(aQ_1 + bQ_2) = aL(Q_1) + bL(Q_2)$.

2.6.3.1 A basis for $\mathbb{R}^{l_z \times l_z}$

Let $\{Q_k\}_{k=1}^{l_z}$ be a basis. such that, $Q \in \mathbb{R}^{l_z \times l_z}$ can be represented as $Q = \sum_{k=1}^{l_z^2} q_k Q_k$.

$$Q_1 = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & \ddots & 0 & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad Q_2 = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 1 & \ddots & 0 & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$Q_{l_z} = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & \ddots & 0 & 0 \\ \vdots & & \ddots & \vdots \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad Q_{l_z^2} = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & \ddots & 0 & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.6.3.2 Range space of $p_{n,2d}$

Let $w(x)$ be a vector of all monomials of degree $\leq 2d$ in n variables.

$$w(x) = \begin{bmatrix} 1 \mid & x_1 & \dots & x_n \mid & x_1^2 & \dots & x_n^2 \mid & \dots \end{bmatrix}^T \in \mathbb{R}^{l_w}, l_w =^{n+2d} C_{2d}.$$

- $w(x)$ forms a basis for $p_{n,2d}$

$$p(x) = \sum_{j=1}^{l_w} c_j w_j(x) = w(x)^T c$$

- Matrix representation of L.

$$L(Q) = z^T(x) Q z(x) \sum_{j=1}^{l_z^2} q_k Q_k$$

where,

$$Q = \sum_{k=1}^{l_z^2} q_k Q_k$$

Thus, making use of linearity property of operator L,

$$L(Q) = L(\sum_k q_k Q_k) = \sum_k q_k L(Q_k) \tag{2.7}$$

In Eq. 2.7, for each Q_k, $L(Q_k) = w(x)^T a_k$. Therefore,

$$\begin{aligned} L(Q) &= \sum_k q_k Q_k = \sum_k q_k [w(x)^T a_k] \\ &= w(x)^T [\sum_k q_k a_k] \\ &= w(x)^T c \end{aligned} \tag{2.8}$$

To see how this works.

$$\begin{aligned} p(x) = & \underbrace{w(x)^T}_{\text{monomials vector}} \underbrace{c}_{\text{coefficient vector}} \\ &= a_1 w_1 + a_2 w_2 + \ldots + a_{l_w} w_{l_w} \\ &= w(x)^T \underbrace{\begin{bmatrix} 1 & 0 & \ldots & 0 \\ 0 & \ddots & 0 & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{A \in \mathbb{R}^{l_w \times l_z^2}} \underbrace{\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{l_w} \end{bmatrix}}_{\text{vec(q)}} \end{aligned} \tag{2.9}$$

$$\begin{aligned} p(x) &= w(x)^T c \\ &= w(x)^T (Aq) \end{aligned}$$

or, $c = Aq$ represents coefficients of the monomials of $p(x)$. Thus an equivalent representation of Theorem 2.6.3 is

Theorem 2.6.4 (SOS LMI Dual)**.**

$$\begin{aligned} & p \in SOS \\ & \textit{iff} \\ & \exists\, Q \geq 0,\ Q = Q^T \\ & \quad \textit{s.t.}\ Aq = c \end{aligned} \tag{2.10}$$

To summarize the connections between positive SOS polynomials, positive semi-definite matrices and the transformation from one basis representation to another,

- polynomial, $p = L(Q)$. Where L is a linear operator.
- Basis $A \in \mathbb{R}^{l_w \times l_z^2}$, such that $c = Aq$. where, $q = vec(Q)$ is a vectorial representation of the matrix Q.
- Relates back to the original polynomial, $p = w(x)^T c$.
- $A \in \mathbb{R}^{l_w \times l_z^2}$ is full row-rank. Usually, $l_w \ll l_z^2$
- A is a sparse matrix with 1's as the only non-zero entries.
- Since A is usually non-square ($l_w \ll l_z^2$), it has a null-space s.t. $Aq = 0 \leftrightarrow z(x)^T Q z(x) = 0$

SOS optimization problems can easily be solved using freeware SOSOPT toolbox, [68], which is MATLAB®toolbox which supports various freely available SDP solvers like [77],[86] for backend processing.

Chapter 3

Polynomial and Linear Parameter Varying (LPV) modeling of the GTM nonlinear longitudinal dynamics

The nonlinear equations of motion (EOM) describing six degree-of-freedom (6-DOF) dynamics of a rigid-body aircraft are well known in the aerospace literature [74, 75]. The algebraic form of the EOMs itself is nonlinear. But also the fact that these nonlinear equations are forced by nonlinear aerodynamic and propulsive forces and moments usually in the form of multi-dimensional look up tables (LUT). The equations of motion also have nonlinearities in the form of some transcendental functions (trigonometric). Considering all these factors, the EOMs turn out to be highly nonlinear functions with non-analytical components such as LUTs embedded within.

The aim in this chapter is to derive decoupled but nonlinear EOMs in a polynomial form, where parameters of the polynomial are nothing but the states of the system. The reason why a polynomial model is required will become much clearer once the nonlinear sum-of-squares (SOS) analysis tools have been discussed in the subsequent chapters. For the moment it would suffice to say that polynomial models are essential to apply the nonlinear sum-of-squares techniques like the region-of-attraction (ROA) and the nonlinear reachability analysis on a nonlinear dynamical system. Also, simpler polynomial models are sometimes desirable for improved numerical performance and faster simulation as it doesn't require costly interpolation operations used in the look up tables (LUTs).

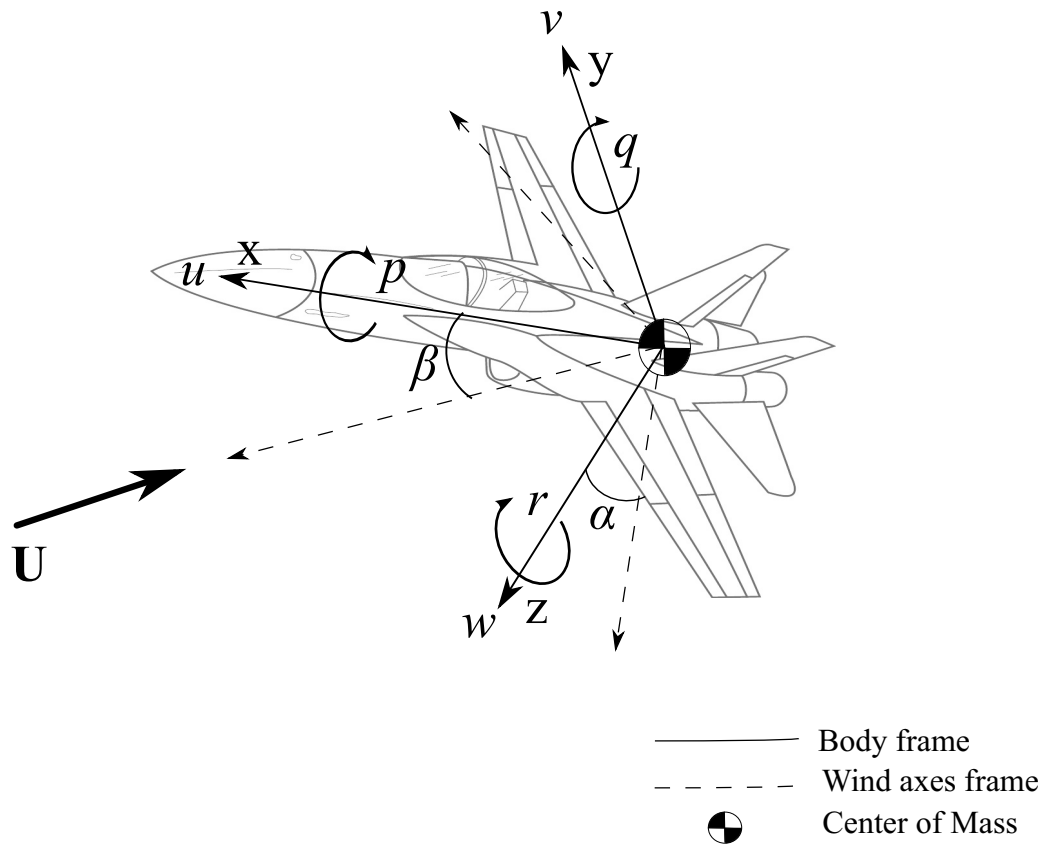


Figure 3.1: Aircraft body coordinate frame and wind-axes system

We consider decoupled dynamics in the way of traditional longitudinal and lateral-directional decoupling. Specifically, only the longitudinal nonlinear EOM shall be considered. This is because only the longitudinal dynamics is considered for the flight envelope assessment analysis in the thesis.

Before getting any further, it is important to define the basic aircraft coordinate frames and conventions that will be liberally used later. Fig. 3.1 shows the two coordinate systems commonly used in the equations of motion, namely, the body-axis and the wind-axis system. U is the wind-vector commonly refered to as the *relative-wind* vector. u, v, w are the linear aircraft velocity resolved in the body-axes, p, q, r are the angular velocities about the x, y, z body-axes respectively. α and β are the angle-of-attack and sideslip-angle, respectively. Fig.3.3 shows the position vectors of the location of the center-of-mass and the aerodynamic center, respectively. These are the focal points through which net inertial forces and aerodynamic forces are assumed to act, respectively. Its noteworthy that conventionally the aerodynamic data (forces and moments) of an aircraft is given about

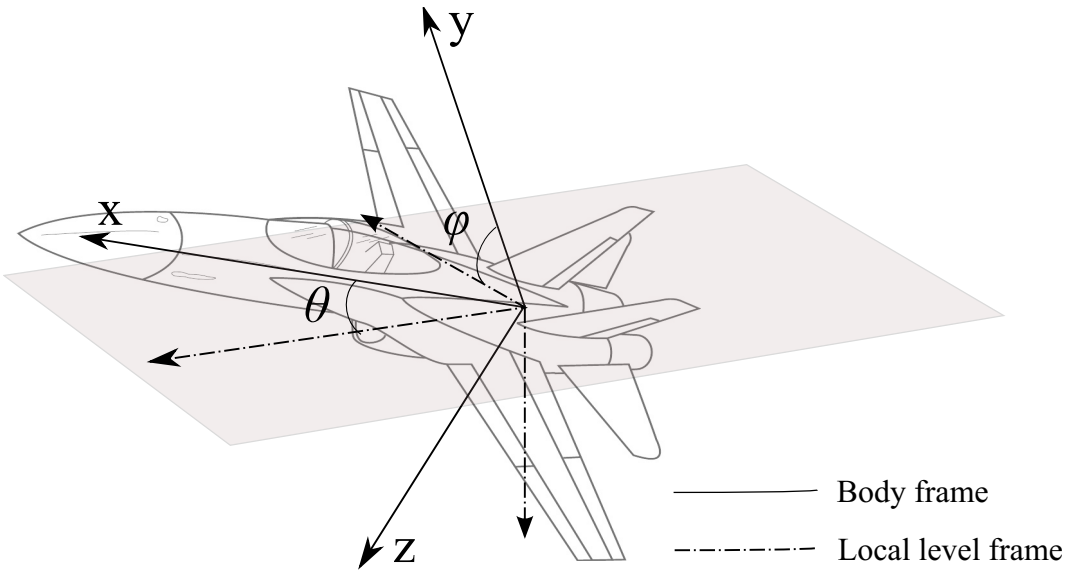


Figure 3.2: Aircraft orientation parameterization using Euler angles

the aerodynamic center (or quarter-chord location of the wing). Thus, the aerodynamic moments need to be transferred appropriately to the center-of-mass location for correct results. This is where knowledge of X_{AC} and X_{CG} becomes imporatant. This fact will be used later in the section.

3.1 The NASA Generic Transport Model (GTM) aircraft

The GTM is a 5.5% scale-model replica of a twin-jet airliner. It designed and developed by NASA Langley to conduct real-time flight tests in the area of flight safety improvement. The project was initially called AirSTAR and in the current form, consists of a fully instrumented airframe, mobile ground control station (GCS) with provision to control manually or autonomously under computer control.

An important aspect of the GTM is that it is dynamically scaled, which means all the aerodynamic/propulsive, inertial and gravitational are ratiometrically the same as found in a full-scale commercial airliner. This allows realistic insights and flight data to be obtained from flight tests as well as validation of advanced algorithms is expected to scale well to full-scale aircraft.

A high fidelity nonlinear simulation model was developed by NASA [20] for this airframe. Extensive wind-tunnel database exists of the aerodynamic data for the GTM. Moreover, it is arguably the only aircraft outside of high performance military jets for which aerodata

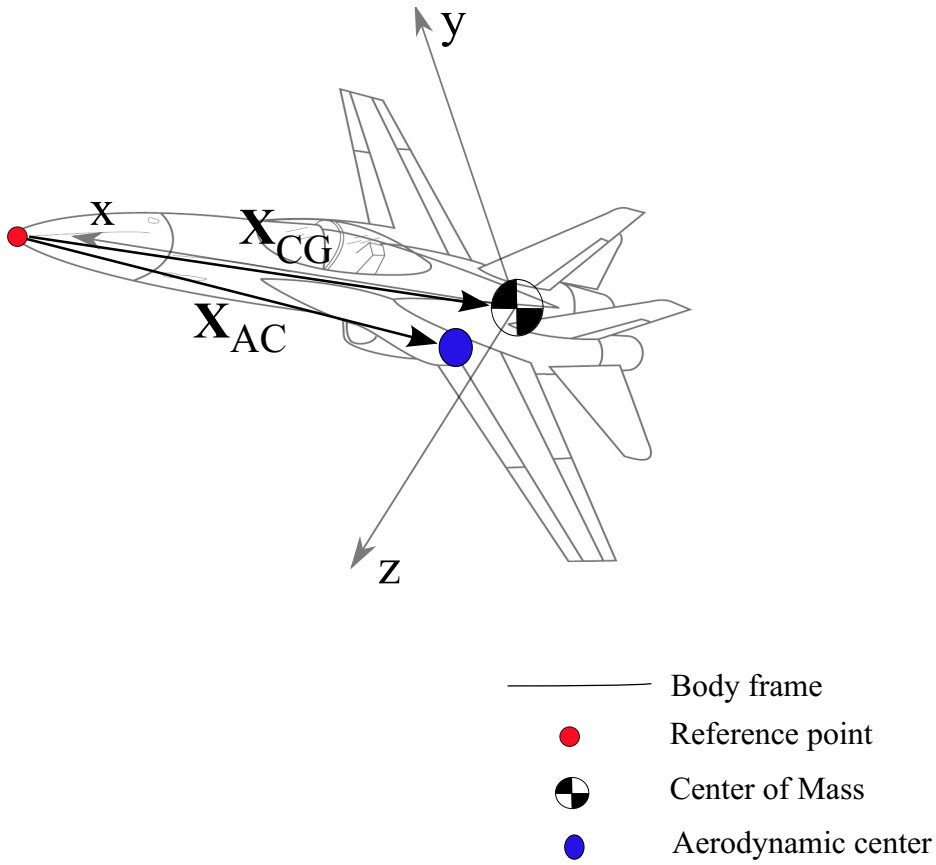


Figure 3.3: Aircraft center-of-mass and the aerodynamic center position vectors

exists and is freely available for a wide range of aerodynamic parameters like the angle-of-attack (AOA), side-slip, etc, well beyond the design flight envelope of the airframe. This allows researchers to study dynamic behavior of the aircraft in highly nonlinear flight regimes encountered in rare instances of extreme flight-upsets, airframe damage and failures, which often times lead to fatal accidents. Fig. 3.4 shows the GTM aircraft in flight.

3.2 Polynomial modeling of the equations of longitudinal motion

The polynomial fitting process to is restricted to a limited aerodynamic angles range so that sufficiently low order polynomial models can be fitted with good accuracy. The polynomial model described below is fitted for angle of attack range of $[0, 20]$ degrees, side-slip $[-10.+10]$ degrees, and $[-20, +20]$ for derivatives with respect to control surface deflection, pitch rate. Ideally, we would like to cover the entire aerodynamic envelope with polynomial fitting, but current state-of-the-art in computational SOS analysis tools does not scale well

Figure 3.4: The NASA GTM aircraft

computationally for complicated higher degree polynomial models.

$$\dot{U} = \frac{1}{m}(F_x cos\alpha + F_z sin\alpha) \tag{3.1a}$$

$$\dot{\alpha} = \frac{1}{mU}(-F_x sin\alpha + F_z cos\alpha) + q \tag{3.1b}$$

$$\dot{\theta} = q \tag{3.1c}$$

$$\dot{q} = \frac{M_y}{I_{yy}} \tag{3.1d}$$

In the following sections, polynomial approximation of each of the nonlinear elements of the EOMs in Eq. 3.1 is presented.

3.2.1 Trigonometric functions approximation

A convenient way to replace the trigonometric nonlinearities appearing in Eqs.3.1 is to use the Taylor series approximation. The Taylor series for sine and cosine functions are given

as follows [15].

$$sin(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1}, n \in \mathbb{Z}^+$$

$$cos(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n}, n \in \mathbb{Z}^+$$

By truncating the series upto third degree terms, one can still obtain approximation with good accuracy as shown in Fig. 3.5.

$$sin(x) = x - \frac{1}{6}x^3$$

$$cos(x) = 1 - \frac{1}{2}x^2$$

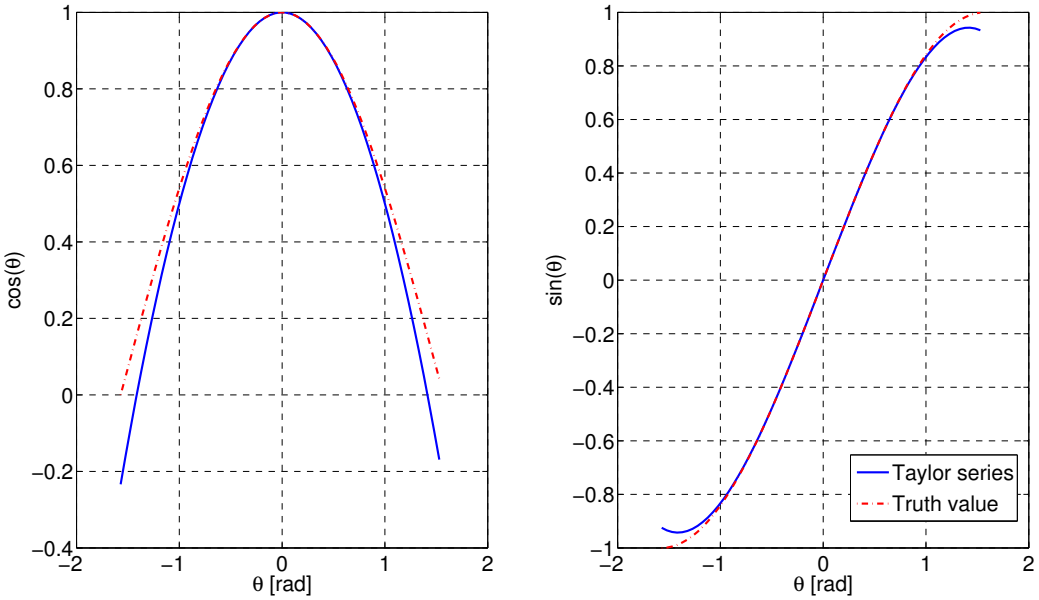


Figure 3.5: Taylor series approximation of sine and cosine function

3.2.2 Least-squares fitting: A primer

Polynomial approximation of the LUT table data is obtained using simple least squares method. The basic principle of simple least squares fitting is as follows. Consider a 1-D problem of finding $f : \mathbb{R} \rightarrow \mathbb{R}$.

Let $\{y_1, y_2, \ldots, y_n\}$ be a given set of data-points to which a polynomial fit is desired. For the sake of example consider a problem of finding a quadratic function $f(x) = c_0 + c_1 x + c_2 x^2 = y$ that best fits the given data-set in the least squares sense.

$$
\begin{aligned}
c_0 + c_1 x_1 + c_2 x_1^2 &= y_1 \\
c_0 + c_1 x_2 + c_2 x_2^2 &= y_2 \\
\vdots \quad & \quad \vdots \quad \quad \vdots \\
c_0 + c_1 x_n + c_2 x_n^2 &= y_n \\
\Rightarrow \underbrace{\begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_1 & x_1^2 \\ \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix}}_{z} &= \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}}_{Y}
\end{aligned}
$$

Where A is a matrix of basis functions, z is a vector of coefficients. Then a unique optimal $z^* = \begin{bmatrix} c_0 & c_1 & c_2 \end{bmatrix}^T$ which minimizes $\|Az - Y\|_2$ or in the least-squares sense is given by,

$$
z^* = \underbrace{(A^T A)^{-1} A^T}_{\text{left pseudo-inverse of A}} Y \tag{3.2}
$$

This result can be used for any polynomial basis function of choice with little modification.

3.2.3 Polynomial approximation of the aerodynamic data

The forces and moment terms in Eqs.3.1 are given as follows, reader may refer to [74, 75] for further details on the aerodynamic and propulsive forces acting on an aircraft.

$$
F_x = \bar{q} S_{ref} [C_x(\alpha) + C_x(\alpha, \delta_e) + C_x(\alpha, \hat{q})] + 2T_X(\delta_{th}) - mgsin\theta \tag{3.3a}
$$

$$
F_z = \bar{q} S_{ref} [C_z(\alpha) + C_z(\alpha, \delta_e) + C_z(\alpha, \hat{q})] + 2T_Z(\delta_{th}) + mgcos\theta \tag{3.3b}
$$

$$
M_y = \bar{q} S_{ref} \bar{c} [C_m(\alpha) + C_m(\alpha, \delta_e) + C_m(\alpha, \hat{q})] + 2\Delta Z_{ENG} T_X(\delta_{th}) \tag{3.3c}
$$

where, the aerodynamic coefficients (functions of angle of attack(α), normalized pitch-rate($\hat{q}$) and elevator deflection (δ_e)) are computed from look up tables (LUTs). The aim here is to replace the LUTs with polynomial approximations.

The polynomial models of the aerodynamic data derived using least-square fit are plotted in Figs.3.8, 3.9 and 3.10. The polynomial aerodynamic models are given in Appendix- A.1.

3.2.4 Engine thrust polynomial model

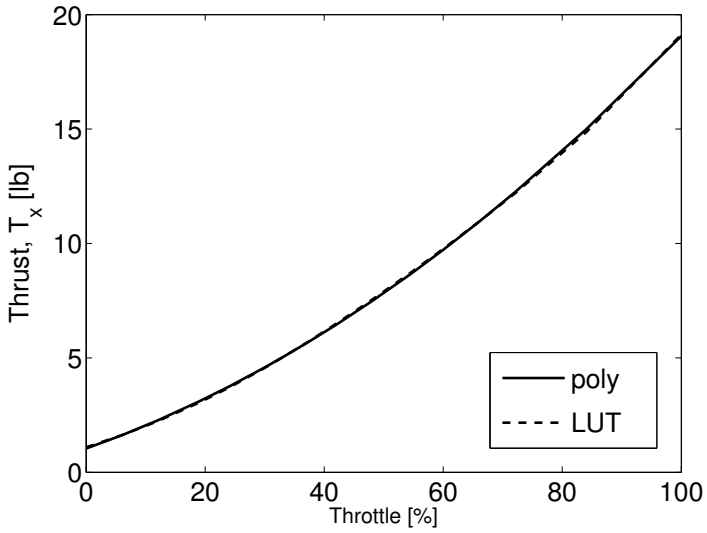


Figure 3.6: Polynomial fitting of the GTM engine thrust model

In Eqs.3.1 it may be noted that engine thrust terms (T_X, T_Z) are prefixed by multiple 2, this is to account for the fact that GTM has two engines, one on each side of the fuselage. Equal thrust setting for both engine is assumed. The least squares polynomial approximation of thrust as a function of throttle position is given in the table below. A comparative plot of the actual thrust LUT with polynomial approximation is shown in Fig.3.6, it can be seen that the an excellent fit is obtained with a relatively low order polynomial in this case.

$$T(\delta_{th}) = (1.04 + 8.871\delta_{th}^2 + 9.151\delta_{th})^1 \tag{3.4}$$

3.2.4.1 The direction cosine vector

It may be noted that engines are inclined with respect to the vehicle body axis, hence the thrust parameter given in the table must be transformed into body axis by a transformation

[1]For one P-70 engine only

vector, called direction cosine vector (DCV). Although the transformation vector is composed of trigonometric functions, the engine alignment is fixed with respect to time, hence DCV can be precomputed and used as a scaling vector on the engine axis thrust. Therefore no approximation of sines, cosines are necessary here. Reader is referred to the GTM Simulink model,[20] for more details on the transformation vector computation.

$$\begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix} = [DCV]_{3\times 1} T(\delta_{th}) \tag{3.5}$$

3.2.5 Equivalent Airspeed (EAS) inverse approximation

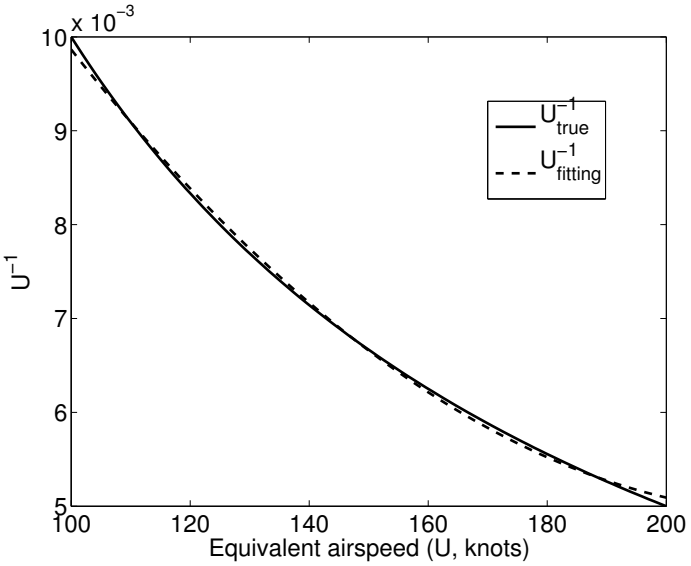
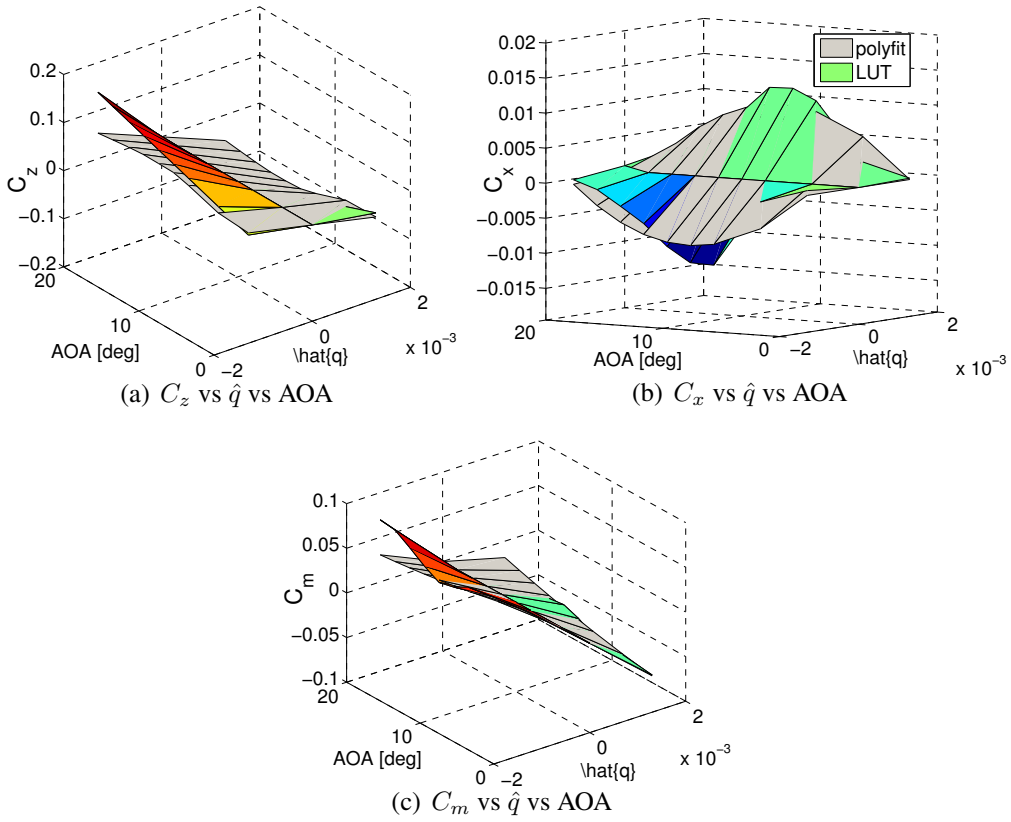


Figure 3.7: Equivalent airspeed inverse polynomial fitting

3.2.6 Aerodata/Engine data comparative plots: Lookup tables vs Polyfit

In Figs.3.8,3.9 and 3.10 the true aerodynamic lookup table (LUT) data for longitudinal forces and moment coefficients are compared with their polynomial approximations as described in the Appendix- A.1. Fig.3.8 shows comparison of normalized pitch-rate ($\hat{q}$) derivatives. Fig.3.9 shows the comparison of angle of attack (α) aero coefficients. Fig.3.10 shows comparison of elevator control aero coefficients. The aim is to show that the polynomial models obtained for aerodata are sufficiently accurate to capture aerodynamic characteristics of the airframe. In general it is observed that the fit obtained are good in the low aerodynamic angles range, but the fit tends to deteriorate at higher angles. This is a lim-

itation while using lower degree polynomial approximations. It is expected however that with improvement in computational SOS tools over time, sufficiently complex polymodels can be incorporated in the future. In Fig.3.10(b) it may be noted that the X force coefficient (C_X) due to elevator has been approximated to zero. This is because of two reasons, first, the magnitude of X-force coefficient is very small compared to Z-force coefficient, secondly, $C_{x_{\delta_e}}$ derivative is highly nonlinear and its not possible to capture the data with a low order polynomial fit.



(a) C_z vs $\hat{q}$ vs AOA

(b) C_x vs $\hat{q}$ vs AOA

(c) C_m vs $\hat{q}$ vs AOA

Figure 3.8: Pitch-rate aero-derivatives: polyfit vs LUT aerodata

3.2.7 Polynomial equations of motion

Combining polynomial approximations from the previous sections and neglecting any monomials of degree greater than five, pure polynomial longitudinal equations of motions are obtained. The longitudinal EOMs for the GTM are given in Appendix- A.2.

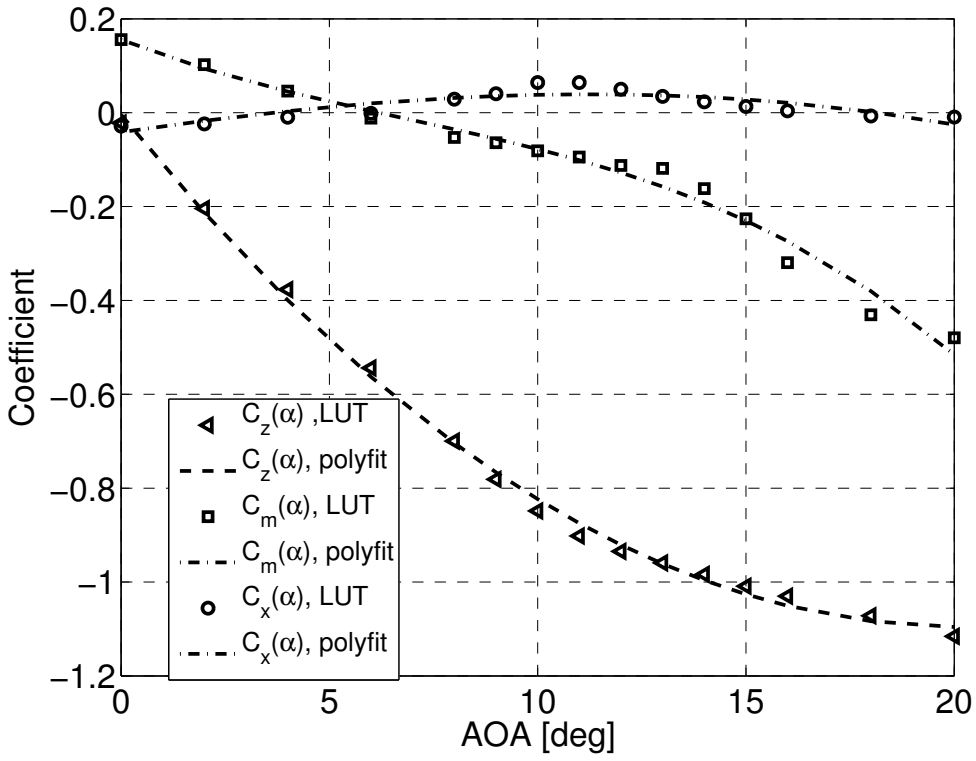
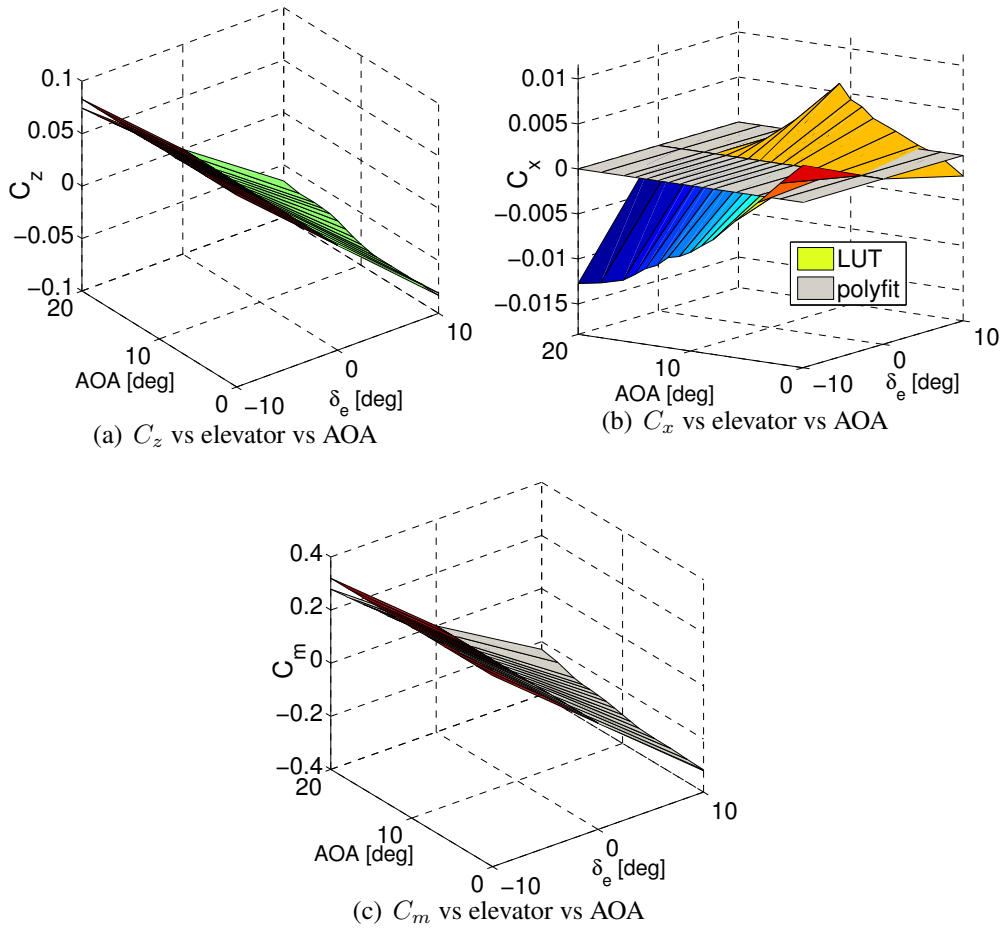


Figure 3.9: Polynomial fitting of AOA aeroderivatives

3.2.8 Comparison with full nonlinear simulation

Using polynomial equations derived in Eqs. A.1 the time response to elevator perturbation is compared with the full nonlinear GTM simulation model. Fig.3.11 shows the comparative plots. In this example, The two GTM models, full nonlinear and the polynomial model described in this report is first trimmed to a straight and level flight at EAS$=$ 90knots, then at time, t$=$ 1 second, the elevator channel is perturbed with a pulse input lasting two seconds. The response of the airplane is captured through longitudinal states, namely, equivalent airspeed (U/EAS), pitch-rate (q), angle of attack (AOA/α), pitch attitude (θ), also plotted in the figure. During the entire maneuver the throttle is kept constant at its trim setting. It can be observed that the response of polynomial model matches quite closely with the full nonlinear GTM model as desired.



(a) C_z vs elevator vs AOA

(b) C_x vs elevator vs AOA

(c) C_m vs elevator vs AOA

Figure 3.10: Polynomial fitting of control surface (elevator) aero-derivatives

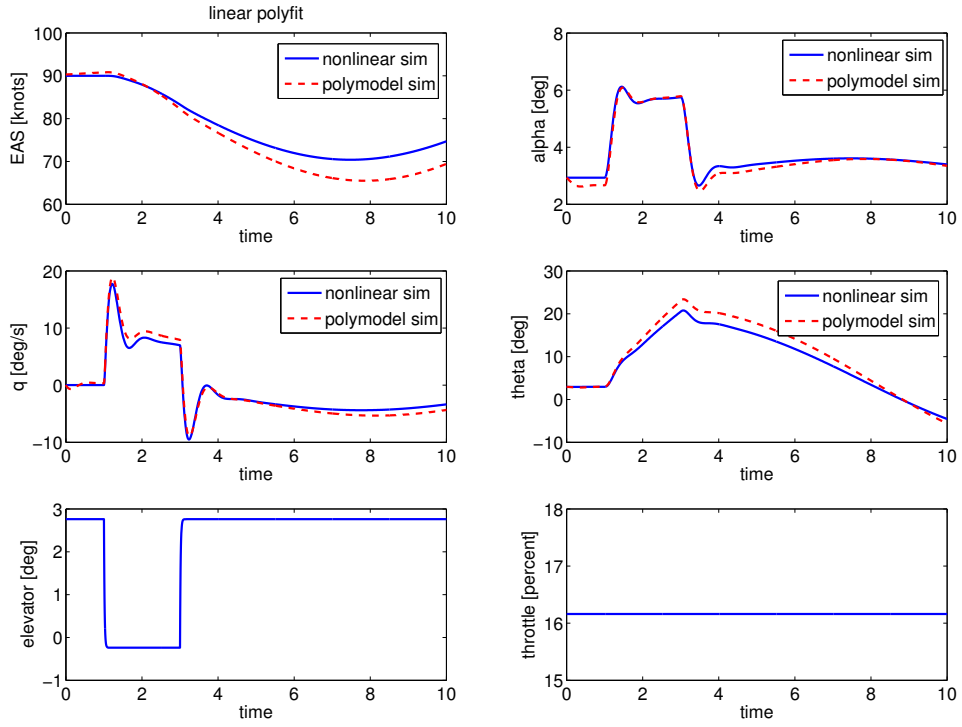


Figure 3.11: Time response comparison: Full nonlinear GTM model vs Polynomial EOMs

3.3 LPV modeling of GTM nonlinear dynamics

An n-dimensional state-space system whose dynamics ((A, B)) depend on certain known parameters $\rho(t)$, which could be a vector valued quantity, are known as parameter-varying systems [81]. Parameter-varying systems differ from time-varying system in that system dynamics are known only for past and upto the current-time, but not future. Whereas, for time-varying systems, system dynamics are known for all times, including the future. In parameter varying systems, we assume that the parameter $\rho(t)$ is known through some measurements on the system using sensors or states of the system itself, or a combination of the two.

It must be noted that although the development of the parameter-varying model of the GTM longitudinal dynamics is presented here, it is not made use of for any investigation or analysis later in the thesis. This is because of some changes in direction over the course of research that lead in the direction of nonlinear analysis instead. Nevertheless, the LPV models are presented here for the sake of completeness as well as made available to aid other researchers' efforts if and when need arises.

A linear parameter varying (LPV) system is one in which the dynamics are linear in x and u, akin to an LTI system. The general form of an linear parameter varying (LPV) system is,

$$\begin{bmatrix} \dot{x}(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} A(\rho(t)) & B(\rho(t)) \\ C(\rho(t)) & D(\rho(t)) \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} \tag{3.6}$$

where, $\rho(t)$ is a scheduling parameter vector. The scheduling vector parameters maybe exogenous variables or state some variables or both. Such systems are called Quasi-LPV systems. Let $x(t) = [\ z(t) \quad w(t)\ ]^T$ and $\rho(t) = [\ z(t) \quad p(t)\ ]^T$, where, $z(t)$,$w(t)$ and $p(t)$ are the scheduling states, non-scheduling states and exogenous scheduling parameters, respectively. Thus a quasi-LPV system may be written as,

$$\begin{bmatrix} \dot{z}(t) \\ \dot{w}(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} A_{11}(\rho(t)) & A_{12}(\rho(t)) & B_1(\rho(t)) \\ A_{21}(\rho(t)) & A_{22}(\rho(t)) & B_2(\rho(t)) \\ C_1(\rho(t)) & C_2(\rho(t)) & D(\rho(t)) \end{bmatrix} \begin{bmatrix} z(t) \\ w(t) \\ u(t) \end{bmatrix} \tag{3.7}$$

There are various ways of obtaining an LPV model for nonlinear systems [81, 26, 71]. For the GTM model two ways are presented - The Jacobian approach and the state transformation approach.

1. The Jacobian Approach:
   In this approach the nonlinear model is expanded using Taylor series around an equilibrium or trim condition. By truncating second and higher order terms a linear model of the system can be obtained. This is the basic idea behind Jacobian LPV model approach. Clearly, this model is only valid in the trimmable subspace of the state-space. Also, the nonlinear functions governing state evolution need to be smooth in the region of interest, else the Jacobian may be undefined in certain regions of state-space.

2. The State Transformation approach:
   In certain systems, it may be possible to mold the nonlinear equations into a suitable LPV form. The advantage of this approach is that since no truncation of higher order dynamics are made, the representation is an exact transformation of the original nonlinear system, hence most accurate.

3.4 Nonlinear dynamics of a rigid airplane

The full six degree of freedom (6-DOF) equations of motion (EOM) for any rigid body are given as follows,

$$\begin{aligned} \dot{\vec{v}}_b &= \frac{\vec{F}}{m} - \vec{\omega} \times \vec{v}_b \\ \dot{\vec{\omega}} &= \mathcal{I}^{-1}[\vec{M} - \vec{\omega} \times (\mathcal{I}\vec{\omega})] \end{aligned} \tag{3.8}$$

Eqs. 3.8 (a) and (b) are vectorial equations, where, v_b is the velocity vector measured in body axes, ω is the angular velocity of the body fixed orthogonal reference frame with reference (see Fig.3.1) to an inertial reference frame, $\mathcal{I}$ is the inertia tensor of the rigid body.

$$\mathcal{I} = \begin{bmatrix} I_{xx} & 0 & -I_{xz} \\ 0 & I_{yy} & 0 \\ -I_{xz} & 0 & I_{zz} \end{bmatrix}$$

$$\mathcal{I}^{-1} = \begin{bmatrix} \frac{1}{I_{xx} - \frac{I_{xz}^2}{I_{zz}}} & 0 & \frac{I_{xz}}{I_{xx}I_{zz} - I_{xz}^2} \\ 0 & \frac{1}{I_{yy}} & 0 \\ \frac{I_{xz}}{I_{xx}I_{zz} - I_{xz}^2} & 0 & \frac{1}{I_{zz} - \frac{I_{xz}^2}{I_{xx}}} \end{bmatrix}$$

As described in polynomial modeling section of this chapter, the forces(F) and moment(M)

in Eqs.3.8 act as the forcing functions on the equations of motion. These forces and moment comprise of the aerodynamic contribution of the complete airframe (i.e. including lifting surfaces), thrust and torque of the engine(s) and the gravitational force, given in Eqs. 3.9 (a), (b) below.

$$\vec{F} = [C_F]_{3\times1}\bar{q}S_{ref} + [m\vec{g}]_b + \vec{T} \tag{3.9a}$$

$$\vec{M} = diag([\ b \quad \bar{c} \quad b\ ])\vec{C}_M\bar{q}S_{ref} + (\vec{X}_{AC} - \vec{X}_{CG}) \times \vec{F} + (\vec{X}_{ENG} - \vec{X}_{CG}) \times \vec{T} \tag{3.9b}$$

In Eq. 3.9(b), X_{AC}, X_{CG} and X_{ENG} are the position vectors of the wing's aerodynamic center, the center of mass and the engines measured from origin of the body reference frame (see Fig. 3.3). Expanding Eq.3.9. (a), (b),

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \begin{bmatrix} C_x \\ C_y \\ C_z \end{bmatrix} \bar{q}S_{ref} + m \begin{bmatrix} -gsin\theta \\ gsin\phi cos\theta \\ gcos\phi cos\theta \end{bmatrix} + \begin{bmatrix} T_x \\ 0 \\ 0 \end{bmatrix} \tag{3.10a}$$

$$\begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} C_l b \\ C_m \bar{c} \\ C_n b \end{bmatrix} \bar{q}S_{ref} + \begin{bmatrix} -\Delta ZF_y + \Delta YF_z \\ \Delta ZF_x - \Delta XF_z \\ -\Delta YF_x + \Delta XF_y \end{bmatrix}_{AC-CG} + \begin{bmatrix} 0 \\ \Delta ZT_x \\ -\Delta Y\Delta T_x \end{bmatrix}_{ENG} \tag{3.10b}$$

The first term in each of the above two equation have the aerodynamic coefficients obtained for a given aerodynamic state of the airplane. Since aerodynamics are highly nonlinear over the flight envelope, its common to employ look up tables (LUT) or simplified polynomial approximations of the aerodata over the flight envelope of analysis and simulation. Aerodynamic contribution of the airframe may be divided into two parts, the contribution from main lifting surfaces (wings) and that from aerodynamic control surfaces. GTM airframe consists of multiple aerodynamic control surfaces listed below and grouped according to their primary force/moment contribution, the emphasis being that the aerodynamic coefficients seen in Eqs.3.10 are functions of the following airframe elements,

- Longitudinal
    1. Horizontal stabilizer

2. Elevators
3. Spoilers
4. Flaps

- Lateral-Directional
  1. Ailerons
  2. Rudder

For up and away flight, some of the control surfaces like spoilers and flaps are not used, and shall be omitted in our model subsequently. Also, horizontal stabilizer is primarily used as a trimming device and not in active control, so for longitudinal motion, only elevator control inputs shall be considered.

$$F_x = \bar{q}S_{ref}[C_x(\alpha,\beta) + C_x(\alpha,\beta,\delta_e,\delta_{st}) + C_x(\alpha,\beta,\delta_{sp}) + C_x(\alpha,\delta_f) + C_x(\alpha,\beta,\delta_r) + 2C_x(\alpha,\beta,\delta_a) + C_x(\alpha,\delta_{LG}) + C_x(\alpha,\hat{q})_{FO} + C_x(\alpha,\beta,\hat{\Omega})] + T_x + mg_x \tag{3.11a}$$

$$F_z = \bar{q}S_{ref}[C_z(\alpha,\beta) + C_z(\alpha,\beta,\delta_e,\delta_{st}) + C_z(\alpha,\beta,\delta_{sp}) + C_z(\alpha,\delta_f) + C_z(\alpha,\beta,\delta_r) + 2C_x(\alpha,\beta,\delta_a) + C_z(\alpha,\delta_{LG}) + C_z(\alpha,\hat{q})_{FO} + C_z(\alpha,\beta,\hat{\Omega})] + T_z + mg_z \tag{3.11b}$$

$$M_y = \bar{q}S_{ref}\bar{c}[C_m(\alpha,\beta) + C_m(\alpha,\beta,\delta_e,\delta_{st}) + C_m(\alpha,\beta,\delta_{sp}) + C_m(\alpha,\delta_f) + C_m(\alpha,\beta,\delta_r) + 2C_m(\alpha,\beta,\delta_a) + C_m(\alpha,\delta_{LG}) + C_m(\alpha,\hat{q})_{FO} + C_m(\alpha,\beta,\hat{\Omega})] + T_x Z_{eng} \tag{3.11c}$$

$$F_y = \bar{q}S_{ref}(C_y(\alpha,\beta) + C_y(\alpha,\hat{p}) + C_y(\alpha,\hat{r}) + C_y(\alpha,\beta,\delta_a) + C_y(\alpha,\beta,\delta_r)) \tag{3.11d}$$

$$M_x = \bar{q}S_{ref}b(C_l(\alpha,\beta) + C_l(\alpha,\hat{p}) + C_l(\alpha,\hat{r}) + C_l(\alpha,\beta,\delta_a) + C_l(\alpha,\beta,\delta_r)) \tag{3.11e}$$

$$M_z = \bar{q}S_{ref}b(C_n(\alpha,\beta) + C_n(\alpha,\hat{p}) + C_n(\alpha,\hat{r}) + C_n(\alpha,\beta,\delta_a) + C_n(\alpha,\beta,\delta_r)) \tag{3.11f}$$

where, $\hat{\Omega} = \Omega(\frac{b}{2U})$,$\hat{q} = q(\frac{\bar{c}}{2U})$. As is evident from equation Eqs. 3.11, the aerodynamic data is given in body axis system directly, which simplifies its integration into simulation and modeling tools, on the other side, the some aerodynamic characteristics of the airframe like stall angle of attack is hidden. It may also be noted that the dynamic aero coefficients,

which are a function of body angular rates, is computed using a technique called the *Hybrid Kalviste* method, [58], in which the aerodynamic forces and moments are resolved into two separate contributions based on body angular rates and orientation of the body axes with the wind axes into a steady state (the Ω dependent term) and an oscillatory component ($\hat{p}, \hat{q}, \hat{r}$ dependent term). Its called hybrid because it combines two different aerodynamic models, forced oscillations and rotary balance (steady state) tests through a switching logic for improved realism. This approach has been shown to successfully recreate airplane dynamics in highly nonlinear high angles of attack and side-slip regimes, like post stall, spin, departure, etc. Since LPV model is not designed for these specific regimes, only oscillatory contribution is considered, based on the reference.

3.5 LPV modeling

Some assumptions with respect to the airplane mass-inertia and aerodynamic characteristics,

1. The airframe is symmetric about XZ-plane, thus $I_{xz} = I_{zx} = 0$. Note, for the GTM model the product of inertia term is non-zero albeit very small, so is neglected anyway. Simulations support this assumption.

2. Longitudinal and Lateral-Directional axes are decoupled, so for longitudinal motion analysis lateral states may be treated as known constants, and vice-versa.

3.5.1 Longitudinal LPV model

The equations of motion described in Eqs. 3.8 are all in body axis reference frame. However, it is convenient to use a mixed axis representation because some of the scheduling states are measured in body and some in wind axis system. The equations given in Eqs. 3.8 can be transformed to into wind axis in a straight forward manner by using appropriate coordinate transformations. For details on derivation of the transformed equations, reader is referred to [75]. Restricting motion to longitudinal plane only,$\beta = 0, p = 0, r = 0$, one is left with four equations of motion for longitudinal dynamics, identical to that used for

polynomial model derivation, repeated here,

$$\dot{U} = \frac{1}{m}(F_x cos\alpha + F_z sin\alpha) \tag{3.12a}$$

$$\dot{\alpha} = \frac{1}{mU}(-F_x sin\alpha + F_z cos\alpha) + q \tag{3.12b}$$

$$\dot{\theta} = q \tag{3.12c}$$

$$\dot{q} = \frac{M_y}{I_{yy}} \tag{3.12d}$$

Aerodynamic forces and moments can also be simplified from Eq.3.11.(a),(b),(c) to,

$$F_x = \bar{q}S_{ref}[C_x(\alpha) + C_x(\alpha, \delta_e) + C_x(\alpha, \hat{q})_{FO}] + T_x + mg_x \tag{3.13a}$$

$$F_z = \bar{q}S_{ref}[C_z(\alpha) + C_z(\alpha, \delta_e) + C_z(\alpha, \hat{q})_{FO}] + T_z + mg_z \tag{3.13b}$$

$$M_y = \bar{q}S_{ref}\bar{c}[C_m(\alpha) + C_m(\alpha, \delta_e) + C_m(\alpha, \hat{q})_{FO}] + \Delta Z_{ENG}T_x \tag{3.13c}$$

And gravitational force components in the body reference frame are,

$$\begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix}_b = \begin{bmatrix} -gsin\theta \\ gsin\phi cos\theta \\ gcos\phi cos\theta \end{bmatrix} \tag{3.14}$$

where, ϕ and θ are the roll and pitch attitude, respectively.

3.5.1.1 Longitudinal LPV model by state transformation

As discussed in introduction, in the state transformation approach the nonlinear equations are directly cast into an appropriate linear parameter form as in Eq. 3.7, no approximations

like traditional Jacobian linearization are carried out. From Eqs. 3.11 and 3.12, we have,

$$\begin{aligned}\dot{U} = \frac{\bar{q}S_{ref}}{m}[cos\alpha(C_x(\alpha) + C_x(\alpha,\delta_e,\delta_{st}) + C_x(\alpha,\hat{q})_{FO} + C_x(\alpha,\hat{\Omega})) + sin\alpha(C_z(\alpha) \\ + C_z(\alpha,\delta_e,\delta_{st}) + C_z(\alpha,\hat{q})_{FO} + C_z(\alpha,\hat{\Omega}))] + \frac{cos\alpha T_x + sin\alpha T_z}{m} \\ + g(-sin\theta cos\alpha + cos\theta sin\alpha)\end{aligned} \tag{3.15a}$$

$$\begin{aligned}\dot{\alpha} = \frac{\bar{q}S_{ref}}{mU}[-sin\alpha(C_x(\alpha) + C_x(\alpha,\delta_e,\delta_{st}) + C_x(\alpha,\hat{q})_{FO} + C_x(\alpha,\hat{\Omega})) + cos\alpha(C_z(\alpha) \\ + C_z(\alpha,\delta_e,\delta_{st}) + C_z(\alpha,\hat{q})_{FO} + C_z(\alpha,\hat{\Omega}))] + \frac{-sin\alpha T_x + cos\alpha T_z}{mU} \\ + \frac{g(sin\theta sin\alpha + cos\theta cos\alpha)}{U} + q\end{aligned} \tag{3.15b}$$

$$\dot{q} = \frac{\bar{q}S_{ref}\bar{c}}{I_{yy}}[C_m(\alpha) + C_m(\alpha,\delta_e,\delta_{st}) + C_m(\alpha,\hat{q})_{FO} + C_m(\alpha,\hat{\Omega})] + \frac{\Delta Z_{ENG}T_x}{I_{yy}} \tag{3.15c}$$

$$\dot{\theta} = q \tag{3.15d}$$

Eqs. 3.15a, can be written in the standard state-space matrix form as shown on the following page, which is the LPV model as desired. The longitudinal quasi-LPV model is scheduled on the states α, q and θ, all of which are available for measurement. Fig. 3.12 shows a comparative plot of full nonlinear versus LPV simulation. An elevator pulse input is used as an excitation. The nonlinear model is initialized at a trimmed straight-and-level flight condition. It can be seen that good matching is obtained for the states corresponding to the faster short-period mode, namely, angle-of-attack, α and pitch-rate, q. The Pitch and airspeed do not match as well. The LPV model equations derived used the state transformation approach are given in Eq. 3.16b.

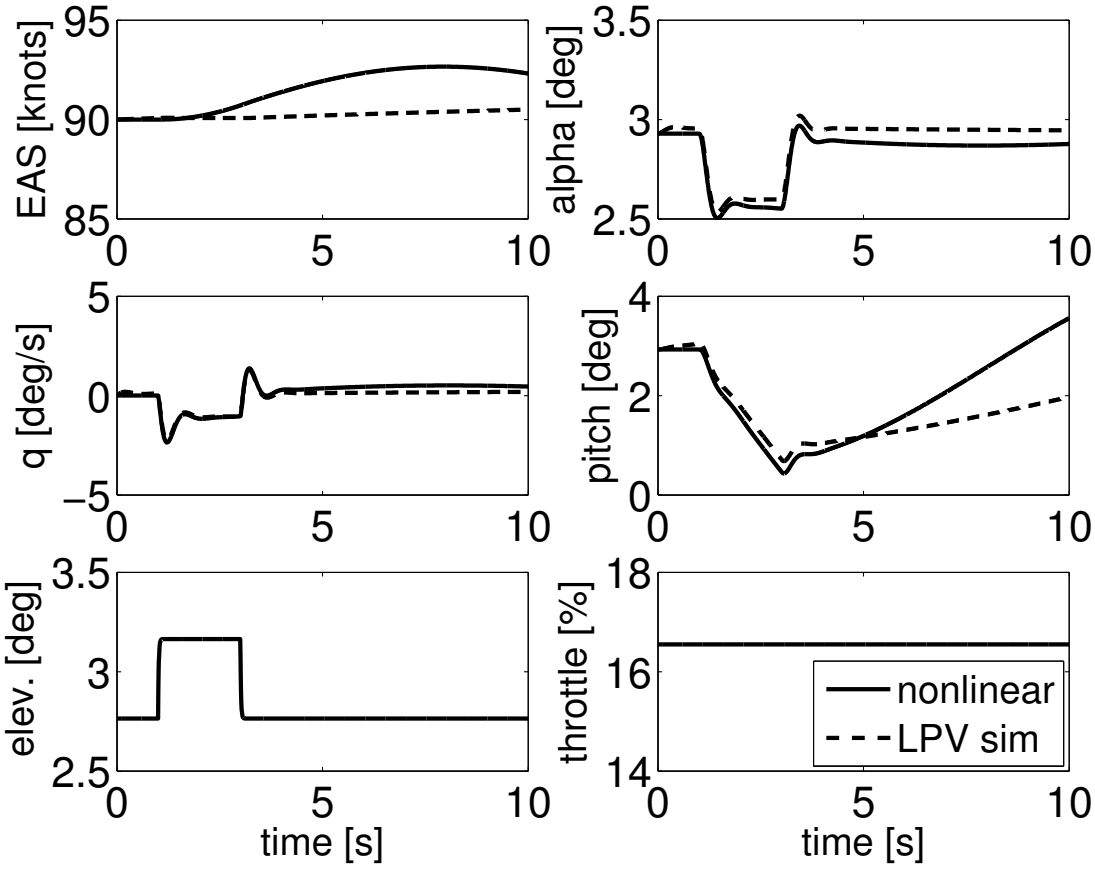


Figure 3.12: Comparison of the state-transformation approach based LPV GTM model with 6-DOF nonlinear simulation (longitudinal)

$$A = \begin{bmatrix} 0 & a_1(cos\alpha\frac{C_x(\alpha)}{\alpha} + sin\alpha\frac{C_z(\alpha)}{\alpha}) & a_1(cos\alpha\frac{C_x(\alpha,\hat{q})}{q} + sin\alpha\frac{C_z(\alpha,\hat{q})}{q}) & g(\frac{-sin\theta cos\alpha + cos\theta sin\alpha}{\theta}) \\ 0 & \frac{a_1}{U}(-sin\alpha\frac{C_x(\alpha)}{\alpha} + cos\alpha\frac{C_z(\alpha)}{\alpha}) & \frac{a_1}{U}(-sin\alpha\frac{C_x(\alpha,\hat{q})}{q} + cos\alpha\frac{C_z(\alpha,\hat{q})}{q}) + 1 & \frac{g}{U}(\frac{sin\theta sin\alpha + cos\theta cos\alpha}{\theta}) \\ 0 & a_2\frac{C_m(\alpha)}{\alpha} & a_2\frac{C_m(\alpha,\hat{q})}{q} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{3.16a}$$

$$B = \begin{bmatrix} a_1(cos\alpha\frac{C_x(\alpha,\delta_e)}{\delta_e} + sin\alpha\frac{C_z(\alpha,\delta_e)}{\delta_e}) & \frac{1}{m}(cos\alpha\frac{T_x}{\delta_{th}} + sin\alpha\frac{T_z}{\delta_{th}}) \\ \frac{a_1}{U}(-sin\alpha\frac{C_x(\alpha,\delta_e)}{\delta_e} + cos\alpha\frac{C_z(\alpha,\delta_e)}{\delta_e}) & \frac{1}{mU}(-sin\alpha\frac{T_x}{\delta_{th}} + cos\alpha\frac{T_z}{\delta_{th}}) \\ a_2\frac{C_m(\alpha,\delta_e)}{\delta_e} & \frac{\Delta Z_{ENG}}{I_{yy}}\frac{T_x}{\delta_{th}} \\ 0 & \end{bmatrix} \tag{3.16b}$$

$$a_1 = \frac{\bar{q}S_{ref}}{m}$$
$$a_2 = \frac{\bar{q}S_{ref}\bar{c}}{I_{yy}}$$

GTM longitudinal LPV model (State Transformation)

3.5.1.2 Longitudinal LPV model by Jacobian linearization

Let the nonlinear dynamics of the scheduling states, z(t) and non-scheduling states, w(t) be given by,

$$\dot{z}(t) = f_1(z(t), w(t), u(t), p(t)) \tag{3.17a}$$

$$\dot{w}(t) = f_2(z(t), w(t), u(t), p(t)) \tag{3.17b}$$

Using the Taylor series expansion around an equilibrium point (z_{eq}, w_{eq}) keeping only first order terms, we have,

$$\begin{aligned}\dot{z}(t) &= f_1(z_{eq}, w_{eq}, u_{eq}, p_{eq}) + \frac{\partial f_1}{\partial z}|_{eq}.(z - z_{eq}) + \frac{\partial f_1}{\partial w}|_{eq}.(w - w_{eq}) \\ &+ \frac{\partial f_1}{\partial u}|_{eq}.(u - u_{eq}) + \frac{\partial f_1}{\partial p}|_{eq}.(p - p_{eq}) \\ &= A_{11}.\delta_z + A_{12}.\delta_w + B_1.\delta_u + E_1.\delta_p\end{aligned} \tag{3.18a}$$

$$\begin{aligned}\dot{w}(t) &= f_2(z_{eq}, w_{eq}, u_{eq}, p_{eq}) + \frac{\partial f_2}{\partial z}|_{eq}.(z - z_{eq}) + \frac{\partial f_2}{\partial w}|_{eq}.(w - w_{eq}) \\ &+ \frac{\partial f_2}{\partial u}|_{eq}.(u - u_{eq}) + \frac{\partial f_2}{\partial p}|_{eq}.(p - p_{eq}) \\ &= A_{21}.\delta_z + A_{22}.\delta_w + B_2.\delta_u + E_2.\delta_p\end{aligned} \tag{3.18b}$$

Note, since there are not external scheduling parameters in the GTM model, matrix E term does not exist. In the familiar state-space matrix formalism,

$$\begin{pmatrix}\dot{z} \\ \dot{w}\end{pmatrix} = \begin{bmatrix}A_{11} & A_{12} \\ A_{21} & A_{22}\end{bmatrix}\begin{pmatrix}\delta_z \\ \delta_w\end{pmatrix} + \begin{bmatrix}B_1 \\ B_2\end{bmatrix}\delta_u + \begin{bmatrix}E_1 \\ E_2\end{bmatrix}\delta_p \tag{3.19}$$

From Eqs.3.15a, assuming first derivatives exist throughout the state space and are continuous at every equilibrium point, we can compute analytic relations for the Jacobian matrices of the GTM longitudinal dynamics. Fig. 3.13 shows a comparative time-simulation plot of the Jacobian LPV model with the nonlinear GTM model. It can be seen that very good matching for identical elevator pulse is obtained in this case compared to the state-transformation approach. The LPV model dervived using the Jacobian approach is given in Eqs. 3.20b.

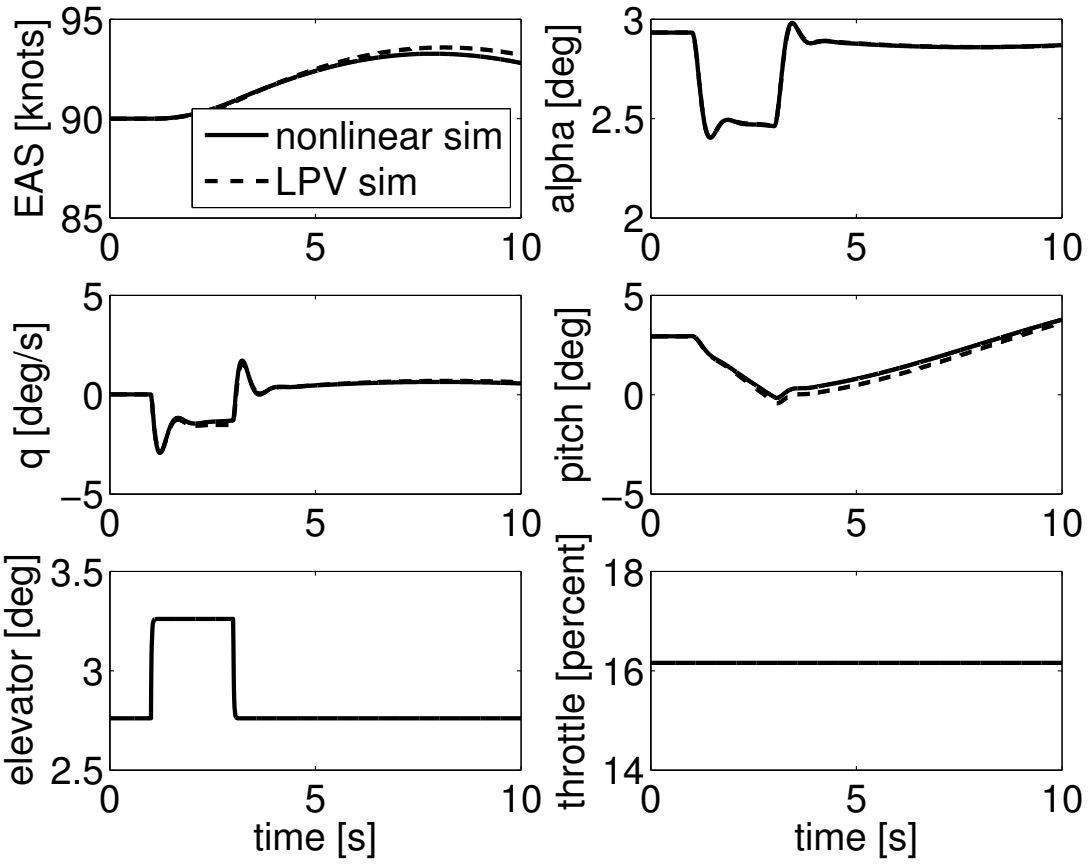


Figure 3.13: Comparison of the Jacobian LPV GTM model with 6-DOF nonlinear simulation (longitudinal)

$$A = \begin{bmatrix} \frac{\rho V_T S_{ref}}{m}[cos\alpha(C_x(\alpha)+C_x(\delta_e)+C_x(q)) & \frac{\bar{q}S_{ref}}{m}[-sin\alpha(C_x(\alpha)+C_x(\delta_e)+C_x(q))+cos\alpha\frac{\partial C_x(\alpha)}{\partial\alpha} & \frac{\bar{q}S_{ref}}{m}[cos\alpha\frac{\partial C_x(q)}{\partial q} & -g(cos\theta cos\alpha+sin\theta sin\alpha) \\ +sin\alpha(C_z(\alpha)+C_z(\delta_e)+C_z(q))] & +cos\alpha(C_z(\alpha)+C_z(\delta_e)+C_z(q))-sin\alpha\frac{\partial C_z(\alpha)}{\partial\alpha}]+ & +sin\alpha\frac{\partial C_z(q)}{\partial q}] & \\ & \frac{1}{m}(cos\alpha T_z - sin\alpha T_x)+g(sin\theta sin\alpha+cos\theta cos\alpha) & & \\ \frac{\rho S_{ref}}{2m}[-sin\alpha(C_x(\alpha)+C_x(\delta_e)+C_x(q)) & \frac{-\bar{q}S_{ref}}{mV_T}[cos\alpha(C_x(\alpha)+C_x(\delta_e)+C_x(q)) & \frac{\bar{q}S_{ref}}{mV_T}[-sin\alpha\frac{\partial C_x(q)}{\partial q} & \frac{g}{V_T}(cos\theta sin\alpha-sin\theta cos\alpha) \\ +cos\alpha(C_z(\alpha)+C_z(\delta_e)+C_z(q))] & +sin\alpha(C_z(\alpha)+C_z(\delta_e)+C_z(q)) & +cos\alpha\frac{\partial C_z(q)}{\partial q}]+1 & \\ -\frac{1}{mV_T^2}(-sin\alpha T_x+cos\alpha T_z) & +sin\alpha\frac{\partial C_x(\alpha)}{\partial\alpha}-cos\alpha\frac{\partial C_z(\alpha)}{\partial\alpha}] & & \\ -\frac{g}{V_T^2}(sin\theta sin\alpha+cos\theta cos\alpha) & -\frac{1}{mV_T}(cos\alpha T_x+sin\alpha T_z) & & \\ & +\frac{g}{V_T}(sin\theta cos\alpha-cos\theta sin\alpha) & & \\ \frac{\rho V_T S_{ref}\bar{c}}{I_{yy}}(C_m(\alpha)+C_m(\delta_e)+C_m(q)) & \frac{\bar{q}S_{ref}\bar{c}}{I_{yy}}\frac{\partial C_m(\alpha)}{\partial\alpha} & \frac{\bar{q}S_{ref}\bar{c}}{I_{yy}}\frac{\partial C_m(q)}{\partial q} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{3.20a}$$

$$B = \begin{bmatrix} \frac{\bar{q}S_{ref}}{m}(cos\alpha\frac{\partial C_x(\delta_e)}{\partial\delta_e}+sin\alpha\frac{\partial C_z(\delta_e)}{\partial\delta_e}) \\ \frac{\bar{q}S_{ref}}{mV_T}(-sin\alpha\frac{\partial C_x(\delta_e)}{\partial\delta_e}+cos\alpha\frac{\partial C_z(\delta_e)}{\partial\delta_e}) \\ \frac{\bar{q}S_{ref}\bar{c}}{I_{yy}}\frac{\partial C_m(\delta_e)}{\partial\delta_e} \\ 0 \end{bmatrix} \tag{3.20b}$$

GTM longitudinal LPV model (Jacobian)

3.6 Summary

In this chapter polynomial nonlinear model and linear parameter varying (LPV) models for GTM longitudinal dynamics were derived. Comparison of each with high-fidelity nonlinear simulation was presented. The polynomial model is used for nonlinear SOS analysis presented in later chapters.

Chapter 4

Reachable-sets

An important aspect of flight envelope assessment is to obtain a measure of the available control authority. Its importance stems from the fact that flight envelope boundaries are intimately tied with the available control authority in an aircraft. For example, the amount of maximum thrust available from the engine defines both maximum altitude and maximum airspeed boundaries. Similarly, elevator control authority governs the pitching moment available for controlling the pitch attitude of an aircraft. The notion of *reachable-sets* is a useful tool to answer the questions of this nature, as will be shown in this chapter.

In simple terms, the reachable set can be thought as the set of all points in the state-space that can be reached using available controls from an initial equilibrium condition. A more rigorous definition is put forth in subsequent section.

First thing to consider is that the question of reachable set computation is only meaningful in the context of well defined input constraints. If there are no input constraints, i.e. any large amount of input energy or magnitude is permissible, then the whole of the state-space is reachable. Second, the notion of reachability only applies to stable (at least locally) systems, system can be pushed to infinity with arbitrarily small input.

Various approaches have been proposed to compute the reachable sets. One approach is fundamentally based in the much celebrated Pontryagin's maximum principle and the subsequent development of the Hamilton-Jacobi Partial Differential Equations (HJPDE). Cross and Mitchell, [21] have applied HJPDE solution to differential algebraic equations. Bayen, Mitchell, et al. [11] applied the same principle to a hybrid autolander safety problem.

Another approach relies on Lyapunov theory and uses the concept of storage functions and level sets of a system. This approach is used in this thesis because it leads to computationally efficient LMI forms for LTI and LPV systems, and generalizes easily to a class of nonlinear systems, i.e. polynomial differential equations which can be solved using sum-of-squares optimization tools. Pandita et al., [62] have applied LMI based approached to an aircraft example, Shin, [73] has applied LMI formulation to reachable sets for linear parameter varying (LPV) systems using the Boeing B747 airplane example.

Reachable sets of a dynamical system is closely related to the notion of controllability and reachability. The two terms are sometimes incorrectly used interchangeably. To make distinction between the two clear,

Controllability Without loss of generality, let the origin be an equilibrium state of a dynamical system. The system is said to be *controllable* on $[t_0, t_1]$ if $\forall$ initial states $x_0 = x(t_0)$, $\exists\, u(.) \in \mathcal{U}$ such that, the final state $x(t_1) = 0$. $\mathcal{U}$ is known as the set of permissible control inputs.

Reachability The dynamical system is said to be *reachable* on $[t_0, t_1]$ if $\forall$ final states x_1 and initial state $x(t_0) = 0$, $\exists\, u(.) \in \mathcal{U}$ such that, the final state $x(t_1) = x_1$.

Controllability and reachability are equivalent for most LTI systems, but not all. In general, reachability implies controllability but vice-versa is not true. Kailath, [48] presents an example of one such case. A linear time-invariant systems is said to be *controllable*, iff the controllability matrix, $\mathcal{C}$ is full-rank.

$$\mathcal{C} = \begin{bmatrix} B & AB & \dots & A^{n-1}B \end{bmatrix}$$

Since the problem in this chapter deals with the ability to drive the system *from* an equilibrium state to another arbitrary state, the correct paradigm to discuss this is reachability and reachable sets.

Recall, the notion of reachable sets is meaningful only in the context of suitable input constraints. Reachable sets for two input constraints cases are presented in this thesis. One is the case of systems with input energy constraints. Although, flight control application isn't typically constrained by input energy, but input energy constraints are pertinent to certain

other application areas, such as space-craft control, where amount of thruster-propellant is limited in quantity. Maximal reachable sets for constrained inputs is intimately tied with the theory of optimal control studied extensively through the 1960s with LQR/LQG and $\mathcal{H}_2$ control theory. For notation, $\mathcal{R}_{ue}$ shall be used to describe reachable sets with input-energy constraints.

Second input constraint case considered are systems with *peak-input* constraints. Any physical system is bounded by the magnitude of control input that can be applied. This is extremely relevant for most mechanical systems including aircrafts. For instance, aerodynamic control surface deflections used to control an aircraft can only be deflected through certain angular ranges due to aerodynamic as well as mechanical limits of the servo-actuators. $\mathcal{R}_{up}$ shall be used to describe reachable sets with peak-input constraints.

The chapter is organized in the following manner. For each input constraint case, we begin with derivation of the key LMI results for an LTI system followed by extension to nonlinear systems, specifically SOS formulation of the LMI results. The chapter concludes with examples of application of reachable set computation for the GTM airplane model and discussion of the results.

4.1 Reachable sets with input-energy constraints

4.1.1 LMI conditions for linear time-invariant systems

Let a linear time invariant (LTI) dynamical system be given by,

$$\dot{x}(t) = Ax(t) + Bu(t)$$

$$y(t) = Cx(t) + Du(t) \tag{4.1}$$

$x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$ and $y \in \mathbb{R}^p$ are the states, inputs and outputs of the LTI system. A, B, C and D are the state-space matrices of appropriate dimensions. Define reachable set for the system given in Eq. 4.1 with input energy bounded by a positive scalar γ, i.e. $\int_0^{t_1} u^T u dt \leq \gamma$, [13],

$$\mathcal{R}_{ue} \triangleq \left\{ x(t_1) \mid \int_0^{t_1} u^T u dt \leq \gamma, t_1 \geq 0, x(0) = 0. \right\} \tag{4.2}$$

The aim is to bound reachable set ($\mathcal{R}_{ue}$) with an ellipsoid (ζ) of the form

$$\zeta = \{\xi \mid \xi^T P \xi \leq \gamma \, , P > 0\}$$

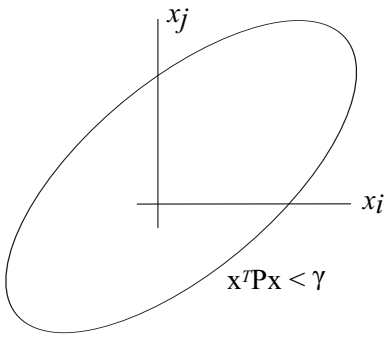


Figure 4.1: Reachable set

Define a storage function $V(x) = x^T P x, P > 0$, such that, $V(x) > 0, \forall x \neq 0, V(0) = 0$ and $\frac{dV(x)}{dt} \leq u^T u, \forall x, u$. Integrating both sides of the inequality,

$$\int_0^T \frac{dV(x)}{dt} dt \leq \int_0^T u^T u dt$$

$$\Rightarrow V(x(T) - V(x(0)) \leq \int_0^T u^T u dt$$

since $V(x(0)) = V(0) = 0$, we have

$$V(x(T)) \leq \gamma \tag{4.3}$$

Comparing with the definition of the bounding ellipsoid (ζ), ellipsoid ζ contains the reachable set $\mathcal{R}_{ue}$.
□

Theorem 4.1.1 ($\mathcal{R}_{ue}$)**.** *From [13], $V(x) \leq \gamma$ is a reachable set for a system with bounded input energy if*

$$\left\{ \begin{array}{c} V(x) \leq \gamma \mid V(x) > 0, \forall x \neq 0, V(0) = 0, \dfrac{dV(x)}{dt} \leq u^T u, \\ \displaystyle\int_0^T u^T u dt \leq \gamma, T \geq 0, x(0) = 0. \end{array} \right\}$$

To arrive at necessary LMI conditions for the ellipsoid (henceforth referred to by the *shape* parameter P) to contain the reachable set, consider the inequality introduced previously,

$$\dot{V}(x) - u^T u \leq 0$$

time-derivative of $V(x)$ can be expressed as,

$$\begin{aligned} \frac{dV(x)}{dt} = \dot{V}(x) &= \dot{x}^T P x + x^T P \dot{x} \\ &= (Ax + Bu)^T P x + x^T P (Ax + Bu) \\ &= x^T (A^T P + PA) x + u^T B^T P x + x^T P B u \end{aligned} \tag{4.4}$$

Thus we have,

$$\begin{aligned} \dot{V}(x) - u^T u \leq 0 \\ \Rightarrow x^T (A^T P + PA) x + u^T B^T P x + x^T P B u - u^T u \leq 0 \\ \Rightarrow \begin{bmatrix} x \\ u \end{bmatrix}^T \begin{bmatrix} A^T P + PA & PB \\ B^T P & -I \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} \leq 0 \end{aligned} \tag{4.5}$$

This is equivalent to an LMI feasibility problem,

$$\begin{bmatrix} A^T P + PA & PB \\ B^T P & -I \end{bmatrix} \leq 0, P > 0 \tag{4.6}$$

Theorem 4.1.2 (LMI conditions for $\mathcal{R}_{ue}$, [ref.[13]]**.** *Given an LTI system $\dot{x} = Ax + Bu$, if $\exists\, V(x) = x^T P x \leq \gamma$, s.t.*

$$\begin{bmatrix} A^T P + PA & PB \\ B^T P & -I \end{bmatrix} \leq 0, P > 0$$

then $\mathcal{R}_{ue} \subseteq V_\gamma(x)$

4.1.2 Sum-of-squares conditions for computing $\mathcal{R}_{ue}$ of polynomial systems

The given constraints on a feasible $V(x)$ to be a reachable set given in Theorem 4.1.1 are *global*. However, for nonlinear systems in general only local results can be obtained.

To be able to find reachable set for nonlinear systems, we present a *local* version of the theorem. It may be noted that derivation of theorem 4.1.1 didn't make any use of linearity property of the system, hence is equally applicable to nonlinear systems. Making use of the $\mathcal{S}$-procedure technique described previously, equivalent set-containment conditions are obtained as follows,

Theorem 4.1.3 (SOS conditions for $\mathcal{R}_{ue}$)**.** *For a nonlinear system governed by polynomial dynamics* $\dot{x} = k(x,u),\ k : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^n$*, if* $\exists\, V(x) : \mathbb{R}^n \mapsto \mathbb{R}$ *s.t.*

$$
\begin{gathered}
V(x) - l_1 \in \Sigma[x] \\
u^T u - \dot{V}(x) - s_1(\gamma - V(x)) \in \Sigma[x] \\
(\beta - p) - s_2(\gamma - V(x)) \in \Sigma[x] \\
s_1, s_2, l_1 \in \Sigma[x] \\
\beta > 0, \gamma > 0
\end{gathered}
\tag{4.7}
$$

then, $\mathcal{R}_{ue} \subseteq V_\gamma(x) \subseteq p_\beta(x)$.

where $p(x) < \beta$ is an n-dimensional ellipsoid of user defined shape. $V_\gamma(x) \triangleq V(x) \leq \gamma$ and $p_\beta(x) \triangleq p(x) \leq \beta$ denote the sub-level sets of $V(x)$ and $p(x)$.It may be noted that unlike LTI case, $V(x)$ may be not necessarily be quadratic. In fact depending on nonlinearities of the system, a quadratic storage function may not be sufficient to prove the prerequisite conditions given in Eqs. 4.7.

Eq. 4.7(a) is an SOS set containment condition that guarantees set $V(x) \leq \gamma$ to be a subset of region where $\dot{V}(x) \leq u^T u$, thus ensuring for every point inside $V(x) \leq \gamma$, $\dot{V}(x) \leq u^T u$ also holds true.

As stated before, $V(x)$ may not be a quadratic storage function for a nonlinear system, but can be a more complex geometrical shape. Our aim is to find simple ellipsoidal bounds of this reachable set. This is achieved by the second SOS condition, given in Eq. 4.7(b). This set containment condition ensures that ellipsoid $p(x) \leq \beta$ is an outer-bound of the reachable set $V(x) \leq \gamma$.

Some important aspects to note from the SOS conditions for reachable set given in Eq. 4.7,

- Eq.4.7(b) imposes stability requirement, since the constraint must also hold true for

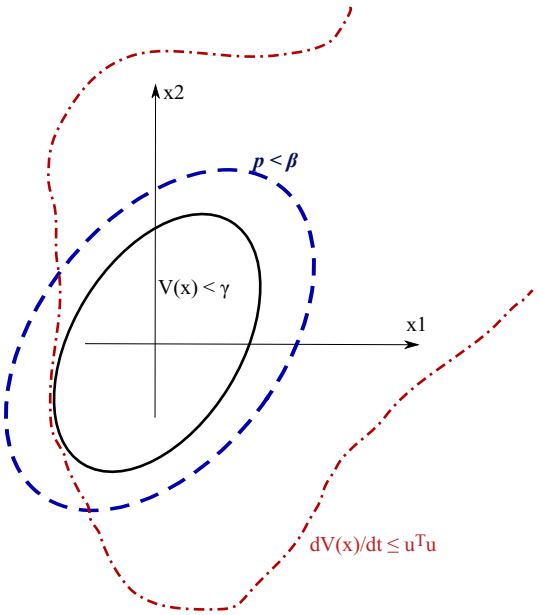


Figure 4.2: Reachable set $\mathcal{R}_{ue}$, nonlinear system

$u^T u = 0$, i.e. when no input is present, which implies $\dot{V}(x) \leq 0$. This is quite intuitive, as we know an unstable system's state trajectories will diverge to infinity for arbitrarily small inputs. This violates the notion of reachability. It also underlines the fact why global reachability results are hard to come by for most nonlinear systems since they have only a finite stable manifold [94, 50].

- SOS constraints Eq.4.7(a) and (b) are infact a *bilinear* matrix inequalities, since both the multiplier s_1, s_2 and scalar γ are unknown. SDP solvers for LMIs cannot be directly applied to such problems. One way to solve it use the so called *bisection method*, which simply involves assuming an upper-bound and lower-bound for feasible γ, and solving the LMI for value of γ starting at the upper-bound, until a feasible value is found.

4.1.2.1 Set-containment of the storage-function

A quadratic storage function $V(x)$may not be enough to computationally prove a reachable set (or a region-of-attraction as will be discussed in later chapters) as in the case of linear systems, for some nonlinear systems. Higher degree storage function of quartic or sometimes even higher complexity may be required depending on the nonlinearity of the system in question. Higher degree storage functions in n-dimensional state-space can have con-

voluted geometric shapes and difficult to visualize. An alternative is to outer-bound these higher degree storage functions with a simple n-dimensional ellipsoid (see Fig. 4.3) of the form,

$$p(x) = x^T P x \leq \beta, \text{where } P \text{ is a user defined shape-matrix}$$

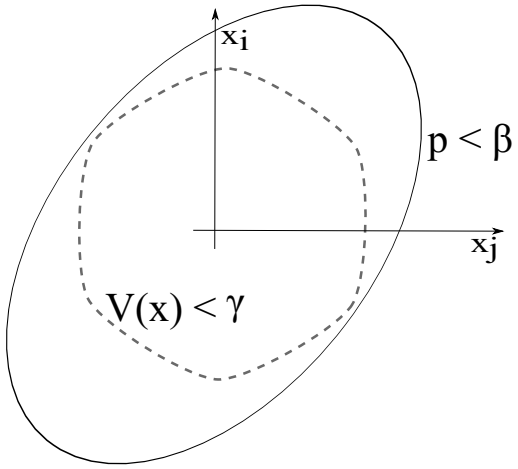
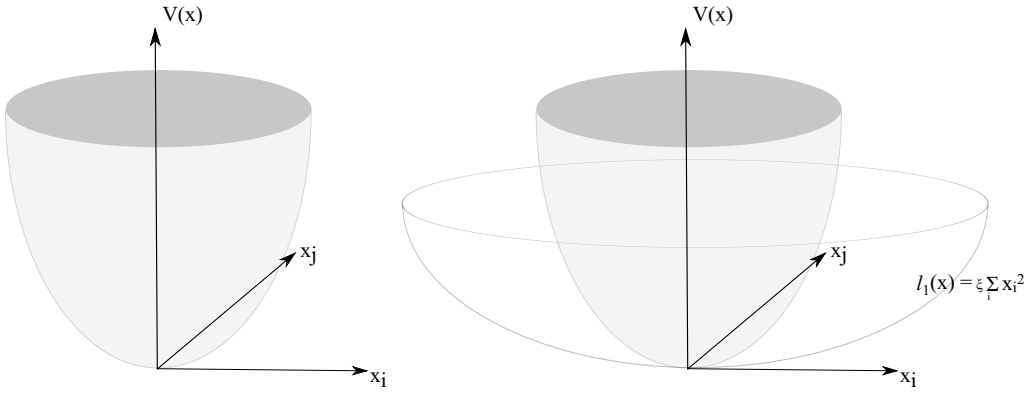


Figure 4.3: SOS set-containment of $V(x)$

4.1.2.2 SOS shaping of the storage-function $V(x)$



(a) Storage function $V(x)$ (b) Role of $l_1(x)$

Figure 4.4: SOS shaping of storage function $V(x)$

The main requirements for a valid storage function are,

$$V(x) > 0, \ \forall \, x \neq 0, V(0) = 0$$

some key mathematical properties of a storage function:

- Positive semidefinite function with closed and compact level-sets.
- Monotonically increasing.
- Radially unbounded, $x \to \pm\infty, V(x) \to \infty$

For further details on the properties of a storage function refer to Vidyasagar [94] and Khalil [50].

The requirement $V(0) = 0$ or a unique global minimum is met by enforcing a structure on $V(x)$ to have no zero degree or constant terms. This is easily enforced in the SOS optimization problem by restricting the choice of monomials used in the optimization process.

Additional requirements on the storage function $V(x)$ can be imposed as SOS constraints in the following manner,

$$V(x) - l_1(x) \in \Sigma[x],\ l_1(x) = \xi \sum_{i=1}^{n} x_i^2 \tag{4.8}$$

where, ξ is a small positive scalar. The SOS constraint implies that $V(x) > 0 \,\forall\, x \neq 0$. The term $l_1(x)$ also ensures that $V(x)$ is monotonically increasing since its a parabolic bowl shaped function which lowerbounds $V(x)$ according the SOS condition given in Eq. 4.8.

4.1.2.3 SOS algorithms for $\mathcal{R}_{ue}$ reachable-set computation

The SOS problem as described in Eqs.4.7 can be posed as SOS optimization problem in two different ways. In this section, we'll discuss properties of the two and why one method is preferable over the other.

In the context of the Fig. 4.5, our aim is to trace the γ vs β curve for a nonlinear system. The problem can be posed in the following two manners -

- Given a $V(x)$, $p(x)$ and γ find smallest value of β, such that constraints in Eq. 4.7 are satisfied. We call this the *climb method* since β is on the vertical axis as depicted in Fig. 4.5.
- Or one may ask, given a $V(x)$, $p(x)$ and β, find the largest value of γ for which the SOS constraints in Eq. 4.7 are satisfied. Call it the *drop method*.

The finer details of each of the two methods is described next section.

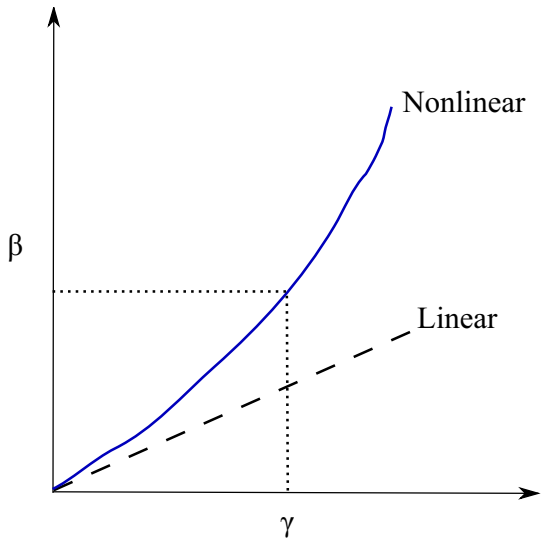


Figure 4.5: SOS iteration to compute $\mathcal{R}_{ue}$

4.1.2.4 The climb method

In this method, the problem is posed in the following way - Given a value of γ, the input energy bound, s.t. $\int_0^T u^T u dt < \gamma$, find β such that level set $V(x) < \gamma$ is contained within $p < \beta$. This is an upper bound for $\mathcal{R}_{ue}$. From Eq.4.7(a),(b) one can see that the problem formulation is linear in decision variables β and the multiplier s_2. This simplifies the problem in that no bisection is required unlike the case of a bilinear matrix inequality (BMI). The problem can be solved easily with SOSOPT function in the SOSTOOLS toolbox [67, 4]. This is the primary advantage the climb method to find the reachable set.

One drawback of this method is that for sufficiently large γ, the initial $V_{LIN}(x)$ used to bootstrap the V-s algorithm may not be feasible specifically on the constraint $\dot{V}(x) \leq u^T u$. This is because $V_{LIN}(x)$ is obtained for the linear model approximation of the actual nonlinear model around an equilibrium. The model is valid for only a small region around the equilibrium. Because of this limitation, it may not be possible to prove a valid $\mathcal{R}_{ue}$ for larger values of γ.

Another major drawback of the climb method is that for a large enough γ or input energy, a feasible $\mathcal{R}_{ue}$ may not even exist. This is because as previously stated, reachable set for a nonlinear set must lie within a stable manifold or *region of attraction (ROA)*. It is possible that for a sufficiently large input energy, the system may be driven out of its ROA, hence has no finite reachable set, $\mathcal{R}_{ue}$. This situation is also depicted in the graphic in Fig. 4.5, for large values of γ on the X-axis, the corresponding β may not exist due to the fact that

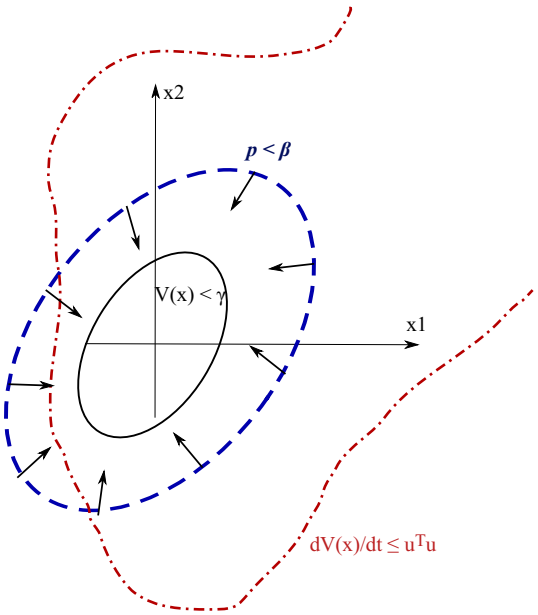


Figure 4.6: SOS climb method for computing the $\mathcal{R}_{ue}$

the nonlinear $\mathcal{R}_{ue}$ curve grows unbounded for certain values of γ.

An alternative to avoid these problems is to pose the problem differently, the so called drop method, which is presented next.

4.1.2.5 The drop method

In the drop method the problem posed is to find a feasible γ for a given β and $V(x)$. The main benefit of this approach is seen by closer examination of the SOS constraints in Eq. 4.7(a),(b). Its easy to see that both the SOS constraints are trivially feasible for $\gamma = 0$. An optimal γ value can be found by increasing it from small values $\gamma \sim 0$. And since $V_{LIN}(x)$ is valid for small region around the equilibrium, a feasible solution can always be found at the bootstrap step.

Also, looking at the nonlinear reachable set curve in Fig. 4.5 one can see that for every finite $\beta > 0$ a feasible γ or the input energy exists for which $p(x) < \beta$ is the reachable set. Hence, the problem of climb method is avoided.

Only drawback of the drop method is that SOS constraints that are needed to be solved are bilinear in decision variables, hence require some additional steps like the bisection algorithm to solve using SDP solvers like *SeDuMi* [78].

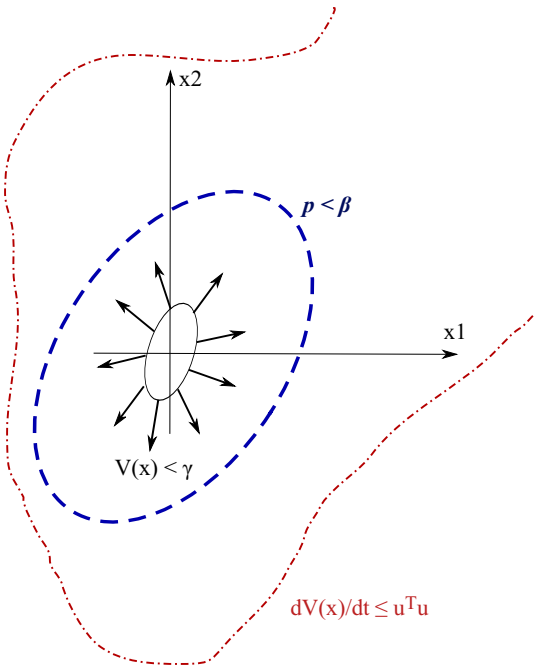


Figure 4.7: SOS drop method for computing the $\mathcal{R}_{ue}$

The algorithm to solve set-containment conditions given in Eq.4.7 for computing the $\mathcal{R}_{ue}$ upper bound using the drop-method is given below,

4.1.2.6 The $\mathcal{R}_{ue}$ SOS algorithm: V-s iteration

1. (*Bootstrap*) Given a β. Linearize nonlinear model to get $\dot{x} = Ax + Bu$.
2. Compute $\mathcal{R}_{ue}$ (linear) based on LMI formulation for LTI systems given in section...call it $V_{LIN}(x)$.
3. (*s-step*) Solve the bilinear SOS set containment conditions in Eq. 4.7(a),(b) independently for the multipliers s_1, s_2 and γ_1, γ_2, respectively.
4. Choose $min(\gamma_1, \gamma_2) = \gamma^*$. This way it is ensured that for a given $V(x)$, both the SOS constraints are satisfied by γ^*.
5. (*V-step*) Fix s_1, s_2, γ^*, solve for a new feasible $V(x)$ based on user specification. The $V(x)$ can be chosen to be a higher degree polynomial.
6. Iterate between V and s step, increasing γ value at each step (hopefully). Terminate the iteration based on a user specified tolerance criteria, etc.

This completes the discussion on computing an upper-bound of the reachable-set $\mathcal{R}_{ue}$ for nonlinear (polynomial) system with bounded input energy constraint. In the following section problem of computing the lower-bound of $\mathcal{R}_{ue}$ is discussed and an efficient algorithm is proposed.

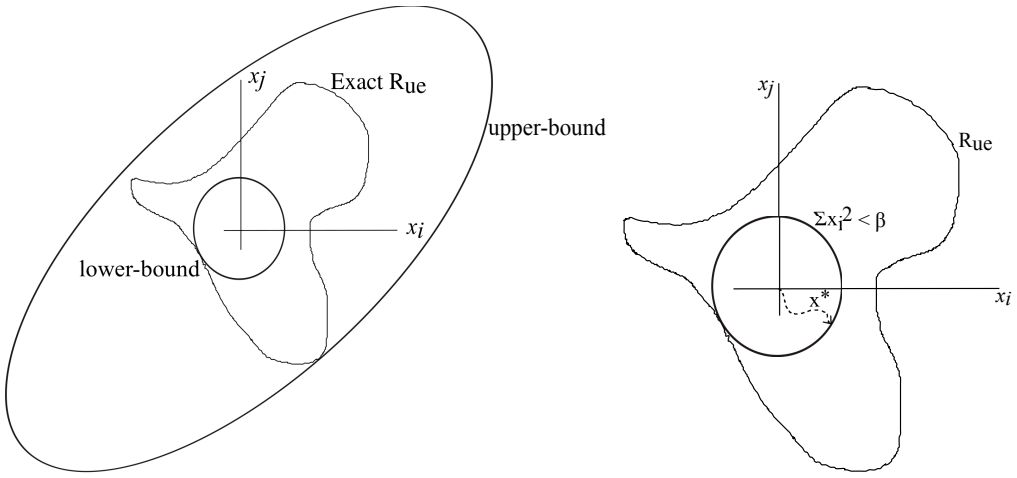
4.1.3 Lower bound for the reachable sets $\mathcal{R}_{ue}$

Different methods for the lower bound computation of the reachable set have been proposed in the literature. Brute force search using Monte-carlo simulations with random inputs is one method, another method proposed in Tierno et al., [85], an algorithm based on computing the solution of a two point boundary value problem with a cost function associated with the worst-case performance of a robust trajectory tracking problem is presented. Tan et al., [82], applied two-point boundary value method to compute lower-bound of the reachable set for aircraft short-period dynamics problem. We propose a method to compute the lower bound in a computationally efficient manner by exploiting linear optimal control theory. An example based on NASA Generic Transport Model (GTM) aircraft nonlinear (polynomial) longitudinal dynamics, [62], is presented. The proposed algorithm is compared with an existing state-of-the-art method given by Tierno and Murray, [85, 82].

4.1.3.1 Problem statement and analysis

Reachable set for a dynamical system is defined as the set of all states which can be reached from an initial state in a given time $T > 0$. Or more precisely, a dynamical system is said to be *reachable* on $[t_0, t_1]$ if $\forall$ final state $x_1 = x(t_1)$, $\exists u(.) \in \mathcal{U}$ so that $x(t_1) = x_1$. In general for a locally stable nonlinear system it can be expected that the reachable set for would be a closed, connected set of some irregular geometric shape in the n-dimensional state-space. Fig. 4.8(a) attempts to show such a typical scenario. Clearly, the best one could do is to come up with upper and lower bounds of the reachable set as shown in the figure.

The approach presented in this work is to construct an optimal control input for the linear system which minimizes the control energy. The idea hinges on the assumption that the least effort *direction* of the linear system state-space is closely aligned with the direction of the least-energy control direction of the nonlinear system (Fig. 4.8(b)). So while this principle yields optimal lower bound in the region of state-space close to the equilibrium, but as we reach further out with larger input energy bounds, the lower bound may not be optimal. However, as will be shown later through the example, the approach actually works



(a) Bounding the reachable set ($\mathcal{R}_{ue}$)

(b) $\mathcal{R}_{ue}$ lower-bound with optimalinput

Figure 4.8: Reachable set bounds for a nonlinear system with input energy constraint

surprisingly well for the example application, and is seen to perform better than the other algorithms available in the literature, [85, 82]. Moreover, since the computation is based on the linear model, its very efficient as well.

Consider the nonlinear dynamics model trimmed at an arbitrary equilibrium. A linear model for small perturbations can be easily obtained for the plant at this equilibrium. Consider a linear time varying system be defined as,

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \tag{4.9}$$

$x \in \mathbb{R}^n$, $u \in \mathbb{R}^p$. The solution to the state trajectory for this system on time-interval $[t0, t1]$ is given by,

$$x(t_1) = \Phi(t_1, t_0) + L_r(u) \tag{4.10}$$

where, $L_{r,[t0,t1]}(u(.)) = \int_{t_0}^{t_1} \Phi(t_1, \tau)B(\tau)d\tau$ is known as the reachability map on $[t0, t1]$, and $W_{r,[t0,t1]} = L_r L_r^T$ is defined as the reachability gramian.

$$\mathcal{W}_{r,[t0,t1]} = \int_{t0}^{t1} \Phi(t_1, \tau)B(\tau)B(\tau)^T\Phi(t_1, \tau)^T d\tau \tag{4.11}$$

Facts,

1. The system is said to be controllable on $[t_0, t_1]$, if and only if the map $L_{r,[t_0,t_1]}$ is

surjective.

2. Equivalently, the system is controllable on $[t_0, t_1]$ if and only if the reachability gramian is full-rank.

3. $W_r \in \mathcal{R}^{n \times n}$ is a positive semi-definite matrix.

Without loss of generality, consider for simplification, $t_0 = 0$, $x(t_0) = 0$ and $t_1 = t_f$, $x(t_1) = x_f$. Define a cost function penalizing control energy used as, $J(u) = \frac{1}{2}\int_0^{t_f} u^T u dt$. The well known result of the optimal control problem that gives a control input strategy that minimizes the control energy cost function while steering the system from $x(t_0) = 0$ to $x(t_f) = x_f$ is given by,

$$u^* = L^T_{r,[0,tf]} W^{-1}_{r,[0,tf]} x_f \tag{4.12}$$

and the optimal cost is,

$$J^* = J(u^*) = x_f^T W^{-1}_{r,[0,tf]} x_f \tag{4.13}$$

Claim: Lowest control input energy cost is incurred if the final state, x_f lies in a direction parallel to the eigenvector corresponding to the largest eigenvalue of the reachability gramian, $W_{r,[0,t_f]}$.

Proof: The eigen-decomposition of W_r can be written as,

$$W_r = V \Lambda V^T, VV^T = I,$$

$$\Lambda = diag(\begin{bmatrix} \lambda_1 & \lambda_2 & \cdots & \lambda_n \end{bmatrix})$$

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

$v_1, v_2, ..., v_n$ are the eigenvectors (column vectors), which form an orthonormal basis for the range-space of W_r.

Let, $x_f = \gamma \bar{v}$, where γ is an arbitrary scaling factor and $\bar{v}$ is the eigenvector corresponding to the largest eigenvalue of $W_{r,[0,tf]}$. Thus from Eq. 4.13,

$$
\begin{aligned}
J^* &= x_f^T W_r^{-1} x_f = x_f^T (V \Lambda V^T)^{-1} x_f \\
&= (\gamma \bar{v})^T V \Lambda^{-1} V^T (\gamma \bar{v}) \\
&= \gamma^2 \lambda_{max, W_r}^{-1} \qquad (4.14)
\end{aligned}
$$

is the lowest possible control energy to steer the system from $x = 0$ to $x = x_f$.
□

Remarks:

1. Computing Reachability Gramian, W_r for $t_f \to \infty$ is trivial, since W_r satisfies the Lyapunov equation, $W_r A^T + A W_r + B B^T = 0$, see Boyd et al., [13] for details.

2. For the finite time horizon case, W_r can be computed from the Eq. 4.11. This is non-trivial for a time-varying linear system in general. But for linear time invariant (LTI) case, computing the state-transition map $\Phi(.,.)$ is simple. This case is considered in the proposed algorithm.

To simplify computation of the reachability gramian, $W_{r,[0,t_f]}$, we propose using the discrete-time formulation, $W_{r,[0,k_1]} = L_{r,[0,k_1]} L_{r,[0,k_1]}^T$, where,

$$
L_{r,[0,k_1]}(u(.)) = \sum_{k=0}^{k=k_1-1} \Phi(k_1, k+1) B(k) u(k) = \mathbf{S}(0, k_1) \mathbf{U} \qquad (4.15)
$$

where, $\mathbf{S} \in \mathcal{R}^{n \times k_1 p}$, $\mathbf{U} = \begin{bmatrix} u(0) & u(1) & \cdots & u(k_1 - 1) \end{bmatrix}^T$.

4.1.3.2 The algorithm for $\mathcal{R}_{ue}$ lower bound computation

- Step 1: Given, system matrices, (A, B), control input-energy bound, γ and a final-time, t_f. Create a linearly spaced discrete time-series, $T = \begin{bmatrix} 0, & \cdots, & t_f \end{bmatrix} = \begin{bmatrix} 0, & t_1, & \cdots, & t_{k_1} \end{bmatrix}$.
- Step 2: Compute, $W_r = L_r L_r^T$ using Eq. 4.15.
- Step 3: Compute the eigen-decomposition, $W_r = V \Lambda V^T$. Let, $\bar{\lambda}$ and $\bar{v}$ be the largest eigenvalue and the corresponding eigenvector of W_r, respectively.

- Step 4: Compute an optimal control input sequence using Eq. 4.12, $u^*(.) = L_r^T W_r^{-1} \bar{v}$. Normalize input energy, $\hat{u}^* = \frac{u^*}{\|u^*\|}$
- Step 5: Let, $u_\gamma = \sqrt{\gamma}\hat{u}^*$. Numerically solve the nonlinear system, $\dot{x}(t) = f(x, u_\gamma, t), t = [0, t_f]$. Let x_f be the final state of the nonlinear system and $\beta = x_f^T x_f$. Then, hypersphere $x_1^2 + x_2^2 + \cdots + x_n^2 \leq \beta$ is the lower bound for reachable set for input energy bound, $\int_0^t u^T u dt \leq \gamma$ as desired.

4.2 Reachable sets with peak-input constraints

Reachable sets for bounded energy input signals was discussed in the previous section. In this section the case of peak magnitude bounded input signals is considered. As mentioned previously, permissible control inputs for any physical dynamical system are bounded in absolute magnitude due to physical limitations. Aircrafts are no different, for example the aerodynamic control effectors have limits on permitted angular deflection. Thus it is important to account for reachable sets for system with this type of input constraint.

First, LMI conditions for LTI systems shall be derived, followed by extension to polynomial systems using SOS formulation.

4.2.1 LMI conditions for LTI systems

Consider the LTI system given in Eq.4.1, the reachable set with bounded peak inputs is defined as,

$$\mathcal{R}_{up} \triangleq \left\{ x(T) \mid u^T u \leq \gamma, \dot{x} = Ax + Bu, T \geq 0, x(0) = 0 \right\} \tag{4.16}$$

Theorem 4.2.1. *Let there exist a quadratic function, $V(x) = x^T P x$, satisfying*

$$P > 0, \dot{x} = Ax + Bu$$

$$\frac{dV(x)}{dt} \leq 0 \; \forall \; V(x) \geq \gamma, u^T u \leq \gamma \tag{4.17}$$

then, $\mathcal{R}_{up} \subseteq V_\gamma(x)$. Or, $V(x) \leq \gamma$ contains the reachable set $\mathcal{R}_{up}$ for the LTI system $\dot{x} = Ax + Bu$, s.t. $u^T u < \gamma$. [Ref. [13]]

Proof: A constructive proof is presented. A representation of the reachable set of a dynamical system is shown in Fig. 4.2.1. The inward pointing arrows show the direction of state

trajectories on the boundary of this set. $P > 0 \Rightarrow V(x) = x^T P x > 0, \forall x, V(0) = 0$, hence $V(x)$ is a bowl shaped monotonically increasing function of x and radially unbounded. $V(x) = k$ is a closed, compact level-set of $V(x)$. Since, $\dot{V}(x) \leq 0$, all the state trajectories at the boundary of the level set $V(x) = \gamma$ point inward.

□

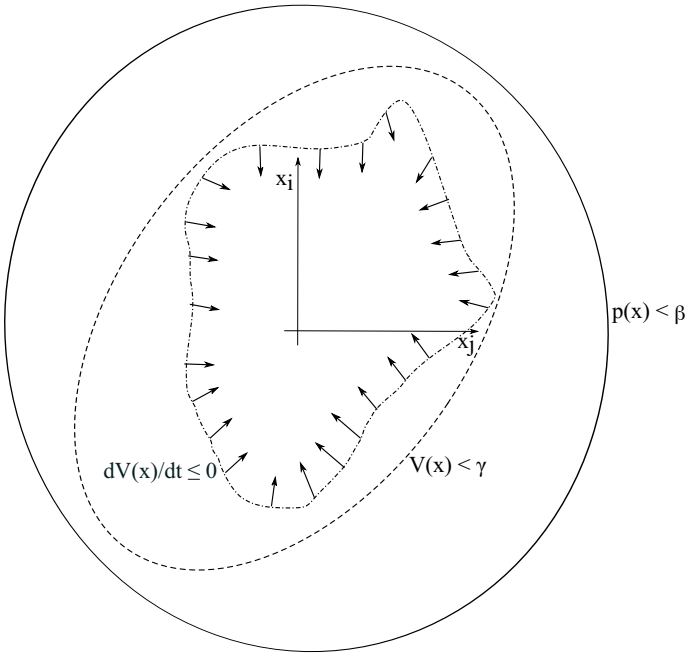


Figure 4.9: Reachable set for peak-input bounds $\mathcal{R}_{up}$

$$\begin{aligned} \dot{V}(x) &= \dot{x}^T P x + x^T P \dot{x} \\ &= (Ax + Bw)^T P x + x^T P (Ax + Bw) \\ &= x^T (A^T P + PA) x + u^T B^T P x + x^T P B w \end{aligned} \tag{4.18}$$

using $\mathcal{S}$-procedure, Eq.4.17 is equivalent to,

$$\dot{V} + \beta(\gamma - u^T u) + \alpha(V(x) - \gamma) \leq 0, \alpha > 0, \beta > 0 \tag{4.19}$$

Simplification yields,

$$
\begin{aligned}
\dot{V} + \beta(\gamma - u^T u) + \alpha(V(x) - \gamma) &\leq 0 \\
\Rightarrow x^T(A^T P + PA)x + u^T B^T P x + x^T P B w & \\
+\beta(\gamma - u^T u) + \alpha(x^T P x - \gamma) &\leq 0 \\
\Rightarrow \begin{bmatrix} x \\ u \end{bmatrix}^T \begin{bmatrix} A^T P + PA + \alpha P & PB \\ B^T P & -\beta I \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} + \gamma(\beta - \alpha) &\leq 0 \qquad (4.20) \\
\Leftrightarrow \begin{bmatrix} A^T P + PA + \alpha P & PB & 0 \\ B^T P & -\beta I & 0 \\ 0 & 0 & \gamma(\beta - \alpha) \end{bmatrix} &\leq 0
\end{aligned}
$$

From Eq. 4.20 its clear that for the inequality to hold true, $\beta \leq \alpha$. If (α_0, β_0) is a feasible pair, then the inequality also holds true for any β s.t. $\beta_0 \leq \beta \leq \alpha_0$. So without loss of generality we can assume $\alpha = \beta$ to simplify Eq. 4.20 as,

$$
\begin{bmatrix} A^T P + PA + \alpha P & PB \\ B^T P & -\alpha I \end{bmatrix} \leq 0, \alpha \geq 0, P > 0 \qquad (4.21)
$$

Its important to recognize that Eq. 4.21 is infact a bilinear matrix inequality (BMI), but for a fixed α it is an LMI.

Theorem 4.2.2 (LMI conditions for $\mathcal{R}_{up}$)**.** *Given an LTI system $\dot{x} = Ax + Bu$, $u^T u \leq \gamma$. If $\exists P > 0$ s.t. $V(x) = x^T P x \geq 0$, $V(x) \leq \gamma$ and*

$$
\begin{bmatrix} A^T P + PA + \alpha P & PB \\ B^T P & -\alpha I \end{bmatrix} \leq 0, \alpha \geq 0
$$

then, $\mathcal{R}_{up} \subseteq V_\gamma(x)$

4.2.2 Iterative LMI algorithm to compute $\mathcal{R}_{up}$

The constraint inequality for $\mathcal{R}_{up}$ as given in Theorem 4.2.2 is a bilinear matrix inequality and hence cannot be solved as such by various SDP solvers. In order to successfully find a feasible $\mathcal{R}_{up}$ or equivalently a feasible matrix $P, P = P^T > 0$ one must convert the BMI into an LMI by selecting a feasible α to initialize the algorithm. This presents a problem as there's no intuition on how to select a feasible α. One approach is by hit and trial, to randomly select α until one returns a feasible solution. Clearly this is not the most attractive

option. Another approach proposed by Zheng et al.,[27] founded on much sound footing is presented in this section.

Consider an LTI system $\dot{x} = Ax + Bu$ as before. One can rewrite inequality in Eq. 4.21 by replacing $Q = P^{-1}$ as,

$$\begin{bmatrix} QA^T + AQ + \alpha Q & B \\ B^T & -\alpha I \end{bmatrix} \leq 0, \alpha \geq 0, Q > 0 \tag{4.22}$$

Using Schur complement lemma,

$$QA^T + AQ + \alpha Q + \frac{1}{\alpha} BB^T \leq 0$$

Let $\Phi = \alpha Q + \frac{1}{\alpha} BB^T$.

- In Eq. 4.22 only Φ depends on α.
- $Q > 0$, $BB^T \geq 0 \Rightarrow \Phi \geq 0$.

That means an optimal α must exist such that value of the positive semidefinite (matrix) cost function $\Phi(\alpha)$ is minimized.

Φ being a simple function can be minimized analytically. One must remember that Φ is not a scalar cost function but a matrix. Hence, a suitable matrix norm must be chosen to be minimized. One such matrix norm is the *Frobenius norm*, $\|.\|_F$ defined as,

$$\|A\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2}$$

where a_{ij} are the elements of the matrix A. A useful property of the Frobenius norm is that it is equal to square root of the matrix trace, i.e. $\|A\|_F = \sqrt{\text{Tr}(AA^T)}$. Frobenius norm is chosen for this problem for the ease of computation.

Since $Q > 0, BB^T \geq 0$, the both can be factored as $Q = \Gamma_1\Gamma_1, BB^T = \Gamma_2\Gamma_2$, respectively, where Γ_1, Γ_2 are positive definite and positive semidefinite matrices respectively. Thus,

$$\Phi = \left(\sqrt{\alpha}\Gamma_1 - \sqrt{\frac{1}{\alpha}}\Gamma_2)(\sqrt{\alpha}\Gamma_1 - \sqrt{\frac{1}{\alpha}}\Gamma_2\right) + (\Gamma_1\Gamma_2 + \Gamma_2\Gamma_1)$$

Minimization of Φ is achieved by minimizing $\|\sqrt{\alpha}\Gamma_1 - \sqrt{\frac{1}{\alpha}}\Gamma_2\|_F$ (the second term in the above expression for Φ is independent of α). Or equivalently using the trace property of Frobenius norm, the objective is to minimize,

$$\text{Tr}\Big(\sqrt{\alpha}\Gamma_1 - \sqrt{\frac{1}{\alpha}}\Gamma_2\Big)\Big(\sqrt{\alpha}\Gamma_1 - \sqrt{\frac{1}{\alpha}}\Gamma_2\Big)$$

$$= \text{Tr}(\alpha Q + \frac{1}{\alpha}BB^T) - \text{Tr}(\Gamma_1\Gamma_2 + \Gamma_2\Gamma_1)$$

using first order conditions for optimality ($\frac{d\Phi}{d\alpha} = 0$) we have,

$$\alpha = \sqrt{\frac{\text{Tr}(BB^T)}{\text{Tr}(Q)}} \tag{4.23}$$

Thus an optimal α can be found using Eq. 4.23 if Q is known. But Q itself is dependent on α. To overcome this difficulty an iterative LMI algorithm is proposed in [27]. A slightly modified version of the algorithm is presented below to compute the reachable.set $\mathcal{R}_{up}$.

4.2.2.1 Iterative LMI algorithm

1. Given: LTI system matrices (A, B) and a positive scalar tolerance ϵ.
2. Choose an arbitrary initial matrix $Q_0 > 0$. Solve Lyapunov equation for Q.

$$QA^T + AQ + Q_0 = 0, Q > 0$$

3. Calculate α,

$$\alpha = \sqrt{\frac{\text{Tr}(BB^T)}{\text{Tr}(Q)}}$$

4. $(OP1)$: Solve the following LMI generalized eigenvalue problem (GEVP),

$$\begin{aligned} \tau^* := \min_{\tau,Q} \quad & \tau \\ \text{s.t.} \quad & \begin{bmatrix} QA^T + AQ + \alpha Q - \tau Q & B \\ B^T & -\alpha I \end{bmatrix} < 0 \\ & Q > 0 \end{aligned}$$

5. If $\tau^* < 0$, α, Q is a feasible pair, goto $OP2$.
   Else, α, Q pair is infeasible. GOTO 2.

6. $(OP2)$: Solve the following LMI eigenvalue problem (EVP) to minimize the size of the feasible ellipsoid, $Q : x^T Q^{-1} x \leq 1$.

$$\begin{aligned} Q^* := \min_{Q} \quad & \text{Tr}(Q) \\ \text{s.t.} \quad & \begin{bmatrix} QA^T + AQ + \alpha Q & B \\ B^T & -\alpha I \end{bmatrix} < 0 \\ & Q > 0 \end{aligned}$$

7. Recompute α_{new} using step 3.

8. Convergence check: If $\|\alpha_{new} - \alpha\|_2 < \epsilon$.

$$Q^* = Q, \alpha^* = \alpha.$$

   STOP.
   Else, set $\alpha = \alpha_{new}$. GOTO 4.

Notes:

- (*Step 2*) This step guarantees that Q obtained satisfies $QA^T + AQ < 0$. That the LTI system is stable.

- (*Step 4,5*) If $\tau < 0$, then $QA^T + AQ + \alpha Q < \tau Q < 0$. This is a verification step that $\exists Q$ such that the pair α, Q is feasible.

- (*Step 6*) Once it is known that at least one feasible pair α, Q exits, then our aim is to find the smallest outer approximation (outer-bound) of the $\mathcal{R}_{up}$. As discussed in Chapter 2. ideally we would like to minimize the volume of the ellipsoid characterized by Q, which is convex optimization problem related to determinant minimization (also known as *MAXDET* class of LMI problems). An easier approximation is to minimize the *trace* of matrix Q, which is equivalent to minimizing the sum of eigenvalues of matrix Q. The sum of the eigenvalues is directly proportional to the sum of *semi-principal axes* of the ellipsoid $x^T Q^{-1} x \leq 1$.

4.2.3 SOS conditions for polynomial systems

LMI conditions and an iterative algorithm to compute reachable set for bounded peak-input case was presented in the previous section. In this section, the results are extended

to polynomial SOS conditions for nonlinear (polynomial) systems. As in the case of SOS conditions for $\mathcal{R}_{ue}$, the reachable set with input-energy constraint assume system dynamics are governed by

$$\dot{x} = f(x,u) = k(x,u), x \in \mathbb{R}^n, u \in \mathbb{R}^m \tag{4.24}$$

where, $k(x,u)$ is a polynomial approximation of the exact nonlinear dynamics $\dot{x} = f(x,u)$. Using Eq. 4.19 we can derive an equivalent SOS theorem for $\mathcal{R}_{up}$

Theorem 4.2.3 (SOS conditions for $\mathcal{R}_{up}$)**.** *For a nonlinear system governed by polynomial dynamics* $\dot{x} = k(x,u),\ k : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^n$*, if* $\exists V(x) : \mathbb{R}^n \mapsto \mathbb{R}$ *s.t.*

$$V(x) - l_1 \in \Sigma[x]$$

$$-\dot{V}(x) + s_1(\delta - V(x)) + s_2(u^T u - \gamma) \in \Sigma[x]$$

$$s_1, s_2 \in \Sigma[x], \gamma > 0, \delta > 0$$

then $\mathcal{R}_{up} \subseteq V_\delta(x) \subseteq p_\beta(x)$.

Proof: Let a point x lie outside $V(x) = \delta$. If the SOS condition in the theorem is holds true and the peak input is bounded by $u^T u \leq \gamma$, then we have,

$$-\dot{V}(x) + \underbrace{s_1(\delta - V(x))}_{<0} + \underbrace{s_2(u^T u - \gamma)}_{<0} > 0$$

$$\Rightarrow \dot{V}(x) < 0$$

This means all the state-trajectories lying outside $V(x) \geq \delta$ point inwards of the level-set $V(x) = \delta$. Thus no trajectory starting within the domain can escape outside, hence $V(x) \leq \delta$ is invariant for $u^T u \leq \gamma$.
□

$V(x)$ can be any degree polynomial function as long as it satisfies the criteria for storage functions, namely - a radially unbounded function with closed, compact level-sets. For LTI systems a quadratic storage function is enough but for a nonlinear system higher degree storage function may be necessary, at the cost of computational complexity.

It should be noted that Theorem 4.2.3 is difficult to prove computationally. The problem lies in the fact that the SOS condition implies global negative-definiteness of $\dot{V}(x)$ is required outside the set $V(x) \leq \delta$, we know this is difficult as most nonlinear systems do not admit

global stability solutions. A solution is proposed to overcome this problem by relaxing the SOS conditions for $\mathcal{R}_{up}$ given in Theorem 4.2.3.

Corollary 4.2.4 (Relaxed SOS conditions for $\mathcal{R}_{up}$, Type-I)**.** *If* $\exists V(x) : \mathbb{R}^n \mapsto \mathbb{R}$*, s.t.*

$$
\begin{aligned}
&V(x) - l_1(x) \in \Sigma[x] \\
&-\dot{V}(x) + s_1(\delta - V(x)) + s_2(u^T u - \gamma) + s_3(V(x) - [\delta + \epsilon]) \in \Sigma[x] \\
&s_1, s_2, s_3 \in \Sigma[x] \\
&\gamma > 0, \delta > 0, \epsilon > 0
\end{aligned}
\tag{4.25}
$$

then $\mathcal{R}_{up} \subseteq V(x)$.

Corollary 4.2.5 (Relaxed SOS conditions for $\mathcal{R}_{up}$, Type-II)**.** *If* $\exists V(x) : \mathbb{R}^n \mapsto \mathbb{R}$ *s.t.*

$$
\begin{aligned}
&V(x) - l_1(x) \in \Sigma[x] \\
&-\dot{V}(x) + s(u^T u - \gamma) + h(V(x) - \delta) - \epsilon \in \Sigma[x] \\
&s \in \Sigma[x], h \in \mathbb{K}[x] \\
&\gamma > 0, \delta > 0
\end{aligned}
\tag{4.26}
$$

then $\mathcal{R}_{up} \subseteq V(x)$.

4.2.3.1 The effect of SOS constraint relaxation

The effect of relaxation in Corollary 4.2.4 and 4.2.5 is that $\dot{V}(x) \leq 0$ is required to be true only in a band around the level set $V(x) \leq \delta$ instead of $\mathbb{R}^n \setminus V_\delta(x)$ as shown in Fig. 4.10. This constraint is easier to satisfy since negative semi-definiteness of $\dot{V}(x)$ is not required globally outside of $V_\delta(x)$. Upon closer inspection, it can be seen how Corollary 4.2.4 and Corollary 4.2.5 achieve this differently.

Consider Corollary 4.2.4, in SOS constraint 4.25, when $V(x) < \delta$, term two is positive and with multiplier s_1 allows relaxation in the constraint so that $\dot{V}(x)$ can be positive in this domain. Similarly, when $V(x) > \delta + \epsilon$, with $s_3 \in \Sigma[x]$, the term four in the SOS constraint is positive, thus offers relaxation in the constraint for to $\dot{V}(x)$ to be positive in the domain $V(x) > \delta + \epsilon$. Within the region $\delta \leq V(x) \leq \delta + \epsilon$, both terms two and four of the SOS constraint, namely, $s_1(\delta - V(x))$ and $s_3(V(x) - [\delta + \epsilon])$ are negative, hence the SOS constraint 4.25 is true if and only if $\dot{V}(x) \leq 0$ ($u^T u \leq \gamma$ given).

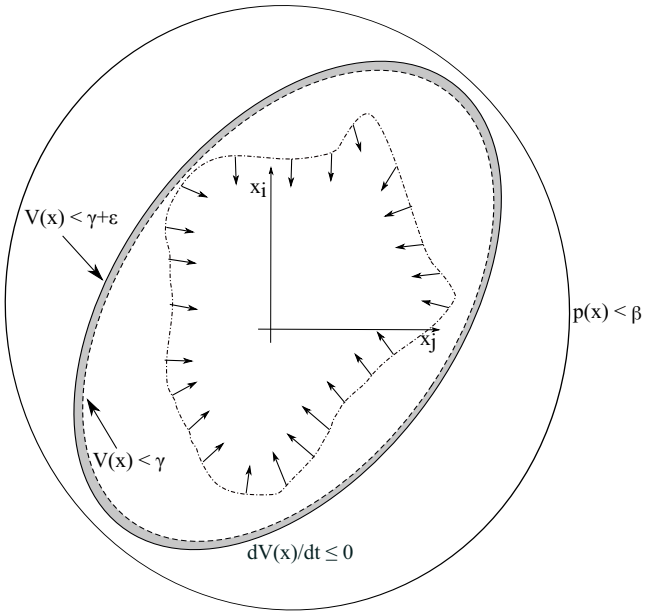


Figure 4.10: Relaxation of SOS conditions for $\mathcal{R}_{up}$ reachable set

Likewise analyzing SOS constraint 4.26 in Corollary 4.2.5, the multiplier h is not constrained to be PSD function, but just a polynomial (i.e. $\in \mathbb{K}[x]$). Thus term three of the constraint, $h(V(x) - \delta)$ is sign indefinite and hence does not force sign definiteness on $\dot{V}(x)$ when $V(x) \neq \delta$.

Theoretically speaking, Corollary 4.2.4 and 4.2.5 are very similar, but computationally Corollary 4.2.4 is more burdensome since it requires finding more SOS multipliers, and also due to the larger number of monomials in constraint 4.25, the SDP optimization problem is expected to be of a higher dimension.

4.2.3.2 SOS algorithm for the computing reachable set $\mathcal{R}_{up}$

The algorithm proposed to compute $\mathcal{R}_{up}$ is similar to the *drop* algorithm used to compute $\mathcal{R}_{ue}$ described in section 4.1.2.5. The algorithm involves the now familiar V-s iteration. The main difference lies in the SOS constraints given in Theorem 4.17.

4.2.3.3 The $\mathcal{R}_{up}$ SOS algorithm: V-s iteration

1. (*Bootstrap*) Given a $\beta, p(x)$ such that $p(x) \leq \beta$, the goal is find γ such that $\mathcal{R}_{up} \subseteq p_\beta(x)\ \forall u^T u \leq \gamma$. Linearize the nonlinear $\dot{x} = k(x, u)$ model to get $\dot{x} = Ax + Bu$.

2. Compute $\mathcal{R}_{up}$ (linear) based on LMI formulation for LTI systems given in section

4.2.2, call it $V_{LIN}(x)$.

3. (*s-step*) Find $V_\delta(x) \subseteq p_\beta(x)$, using $\mathcal{S}$-procedure.

$$\begin{aligned} \delta^* = \quad & \max \quad \delta \\ & \text{s.t.} \quad (\beta - p) - s_1(\delta - V) \in \Sigma[x] \\ & \qquad s_1 \in \Sigma[x] \end{aligned} \tag{4.27}$$

This problem is a bilinear matrix inequality since s_1 and δ are both decision variables. Simplest approach is to use a bisection based algorithm as discussed previously.

4. Using $V_{LIN}(x)$ obtained in the previous step and δ^*, solve the following SOS optimization. Note, $V_{LIN}(x)$ used only in the first iteration to bootstrap the algorithm, in subsequent iterations $V(x)$ obtained from V-step shall be used.

$$\begin{aligned} \gamma^* = \quad & \max \quad \gamma \\ & \text{s.t} \quad -\dot{V} + h(V - \delta) + s_2(u^T u - \gamma) - \epsilon \in \Sigma[x] \\ & \qquad s_2 \in \Sigma[x] \\ & \qquad h \in \mathbb{K}[x] \end{aligned} \tag{4.28}$$

This SOS problem is also a bilinear inequality requiring bisection an represents the type-II relaxation as proposed in Corollary 4.2.5. The only difference to this SOS algorithm for the case type-I relaxation to replace the SOS problem in Eq. 4.28 with that in Corollary 4.25(b) and the corresponding multipliers. If no feasible γ is found, reduce δ using a simple rule like, $\delta_{new} = 0.1\delta^*$ and repeat the SOS optimization. It is not unexpected that $V_{LIN}(x)$ may only be able to prove the above SOS feasibility for a small δ, this is because $V_{LIN}(x)$ is only valid reachable set for the linear approximation of the nonlinear model. However, this is not a handicap since the role of *bootstrap* step is just to find an initial feasible solution, in the V-step that follows. The optimization would try to find a higher degree $V(x)$ which may be able to prove $\mathcal{R}_{up}$ SOS constraints for a larger δ.

5. (*V-step*) Fix $s_2, h, \gamma^*, \delta^*$, solve the SOS problem given in 4.28 for a new feasible $V(x)$ based on user specification. The $V(x)$ can be chosen to be a higher degree polynomial. Note, ϵ is a user defined arbitrary small positive scalar.

6. Iterate between V and s step, increasing γ value at each step. Stop iterating when γ^* stops improving, i.e. the iteration may be stopped based on a user specified tolerance criteria.

4.2.4 Multiplier selection guidelines

The SOS relaxation for $\mathcal{R}_{up}$ reachable set constraints section did not talk about composition of the multipliers used in the $\mathcal{S}$-procedure. By making use of empirical knowledge about the various terms in the SOS constraints in expressions 4.25 and 4.26, the optimization process can be improved significantly.

4.2.4.1 Multipliers for relaxed SOS constraint, type-I

$$(\beta - p) - s_1(\delta - V) \in \Sigma[x] \tag{4.29a}$$
$$-\dot{V} + s_2(\delta - V) + s_3(u^T u - \gamma) + s_4(V - [\delta + \epsilon]) \in \Sigma[x] \tag{4.29b}$$
$$s_1, s_2, s_3, s_4 \in \Sigma[x] \tag{4.29c}$$
$$\delta > 0, \epsilon > 0, \gamma > 0 \tag{4.29d}$$

4.2.4.2 Multipliers for relaxed SOS constraint, type-II

Consider the SOS problem given in expression 4.26,

$$(\beta - p) - s_1(\delta - V) \in \Sigma[x] \tag{4.30a}$$
$$-\dot{V} + h(V - \delta) + s_2(u^T u - \gamma) - \epsilon \in \Sigma[x] \tag{4.30b}$$
$$s_1, s_2 \in \Sigma[x] \tag{4.30c}$$
$$h \in \mathbb{K}[x] \tag{4.30d}$$
$$\delta > 0, \gamma > 0, \epsilon > 0 \tag{4.30e}$$

- **Properties of s_1:**
  Since the SOS problem depends only on x, s_1 should admit monomials of x, or $s_1 := s_1(x)$.

- **Properties of s_2:**
  Claim: s_2 is a function of x, u, i.e. $s_2 := s_2(x, u)$.

Argument: If s_2 is not a polynomial function of x, u then the second term of the SOS $s_2(u^T u - \gamma) = 0, \forall\ u$, at $x = 0$, i.e. control has no effect on $\dot{V}(x)$ and hence the states of the system at the origin, which is clearly untrue.

- **Properties of h:**

  1. *Question:* Is $h := h(x)$ or $h(x, u)$?
     Argument: If $h := h(x)$, then at the origin, $\dot{V} = 0$, for $x, u = 0$. But when $u \neq 0, x = 0$, the first term remains zero and second term is less than zero for all u s.t. $u^T u < \gamma$, or $\dot{V}(0) < 0, \forall\ u^T u < \gamma, x = 0$. That does not make sense. With $h := h(x, u)$, the first term has some positive contribution, hence is a suitable choice.
  2. Since, $x \to \infty, V(x) \to \infty$ (*radial unboundedness*). If h is positive (or SOS), implies $x \to \infty, \dot{V}(x) << 0$.
     Claim: SOS constraint 4.30b would be much easily satisfied if $h < 0$ as $x \to \infty$, because then $\dot{V}(x) < \infty$ is always true.
  3. Degree of $h(x, u)$ should be even such that $h(x, u)$ is symmetric about the y-axis, that is, $h \to -\infty$ when $x \to \pm\infty$ as required by the previous statement.
  4. *Value of h at the origin:*. So that for $V(0) = 0, \dot{V}(0) = h(0, 0)(\delta - 0) + s_2(0 - \gamma) = 0$.
     Claim: $s_2(0) \neq 0$, otherwise, $\dot{V}(x)$ cannot be affected by control u, then the system is stuck at the origin, this cannot be true. So if $s_2(0) \neq 0$ is true, then $\dot{V}(0) = 0 \implies h(0, 0) > 0$.
     Conclusion: (1) Both $s_2(x, u)$ and $h(x, w)$ should permit constant terms (degree zero monomials). (2) $h \geq s_2 > 0$. and,

$$\begin{cases} h = s_2, & x, u = 0 \\ h > s_2, & \text{elsewhere} \end{cases}$$

- **Role of ϵ:** Consider the SOS constraint 4.30b, when the system is being pushed with maximum permissible control authority, $u^T u = \gamma$,

$$\dot{V} \leq h(\delta - V) + s_2(\underbrace{u^T u - \gamma}_{=0}) - \epsilon \tag{4.31}$$

From Eq. 4.31 it is evident that at $V(x) = \delta$, the constraint forces $\dot{V}(x) \leq -\epsilon < 0$. And $\dot{V}(x) \leq 0$ at $V(x) = (\delta - \frac{\epsilon}{h})$. This implies that there is strip of width $\frac{\epsilon}{h}$ at

$V(x) = \delta$ where $\dot{V}(x) \leq 0$. Thus system trajectories can never escape the level set $V(x) = \delta \; \forall \; t$ as required by the definition of $\mathcal{R}_{up}$ reachable set.

4.3 Examples

For examples presented in this section the GTM longitudinal polynomial model and an LTI longitudinal model obtained at trim flight condition of EAS=90 knots, at an altitude of 200m. Details on the polynomial nonlinear model is given in chapter 3, the model itself is given in Appendix-A.2.

4.3.1 Applications of linear reachable-sets analysis

Fig.4.11 shows 2-dimensional projections of the input-energy constrained reachable-set ellipsoid, $\mathcal{R}_{ue}$ for the LTI GTM model obtained by solving the LMI problem given in theorem 4.2.2. A second order elevator actuator dynamics model of bandwidth 7Hz and damping ratio of 0.7 is included in the plant dynamics. Elevator input energy limit of 1 rad^2 is is considered in this example. The $\mathcal{R}_{ue}$ for a simulated 60% failure of the elevator is also plotted in the figure. As expected it can be seen that elevator effectiveness reduction results in a significant change in the reachable-set.

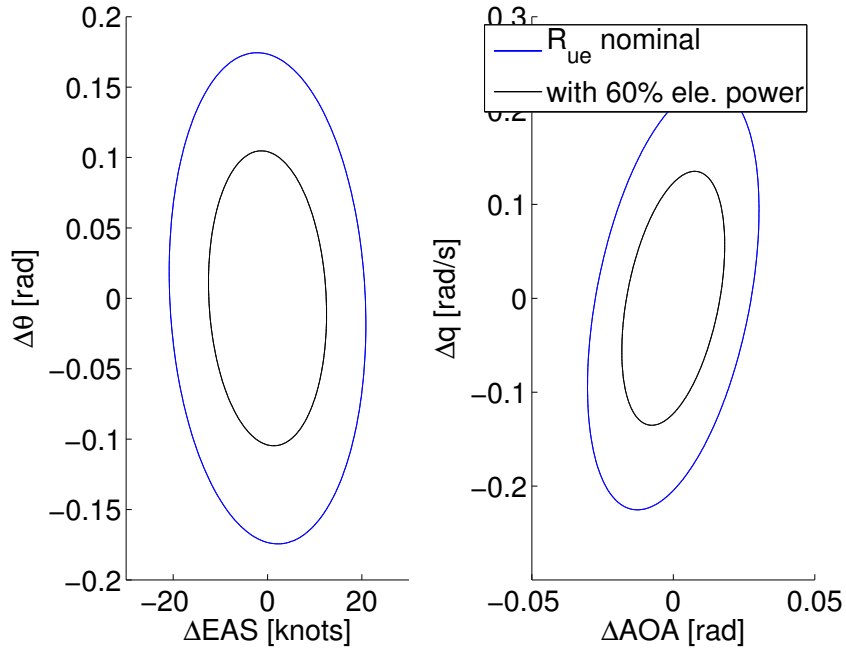


Figure 4.11: Effect of elevator power reduction on $\mathcal{R}_{ue}$

To verify the $\mathcal{R}_{ue}$ ellipsoidal approximation obtained, monte carlo linear simulations using random inputs with energy bound of $1rad^2$ are performed. The results are shown in Fig. 4.12. It can be seen that $\mathcal{R}_{ue}$ approximation is precise in the θ-EAS plane, whereas it is quite conservative in the $\alpha-$ plane. This behavior is indicative of the fact that the in convex optimization the optimal solution is one where atleast one of the constraint boundary is reached. It may be possible to obtain a better solution by using a different cost function, maximizing the volume of ellipsoid instead of the matrix trace.

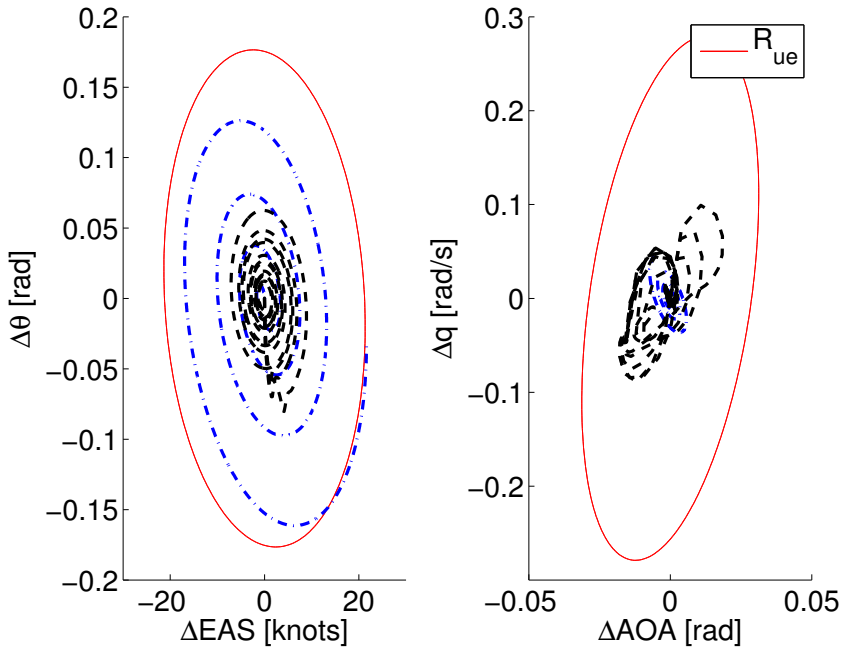


Figure 4.12: $\mathcal{R}_{ue}$ validation with Monte Carlo linear simulations

Reachable set with peak input constraint of ± 20 deg elevator is computed using the LMI theorem in 4.2.2 and the ILMI algorithm. Fig. 4.13 shows the $\mathcal{R}_{up}$ ellipsoid projections for the nomial and with 60% elevator loss. As expected we find that elevator authority reduction leads to a reduction in the size of reachable set of the system. Monte-carlo simulations for verification of $\mathcal{R}_{up}$ reachable ellipsoid is carried out using random elevator input signals within ± 20 deg limits. As shown in Fig. 4.14, the $\mathcal{R}_{up}$ ellipsoid obtained is an upper-bound on the true reachable-set.

The $\mathcal{R}_{ue}$ and $\mathcal{R}_{up}$ for LTI GTM model are shown together in Fig.4.15 for relative size comparison. It is observed that the $\mathcal{R}_{up}$ reachable-set is larger in size compared to the $\mathcal{R}_{ue}$. This is expected because with peak-input constraint there is no limit on the amount of energy that can be pumped into the system, hence it may be possible to drive the states

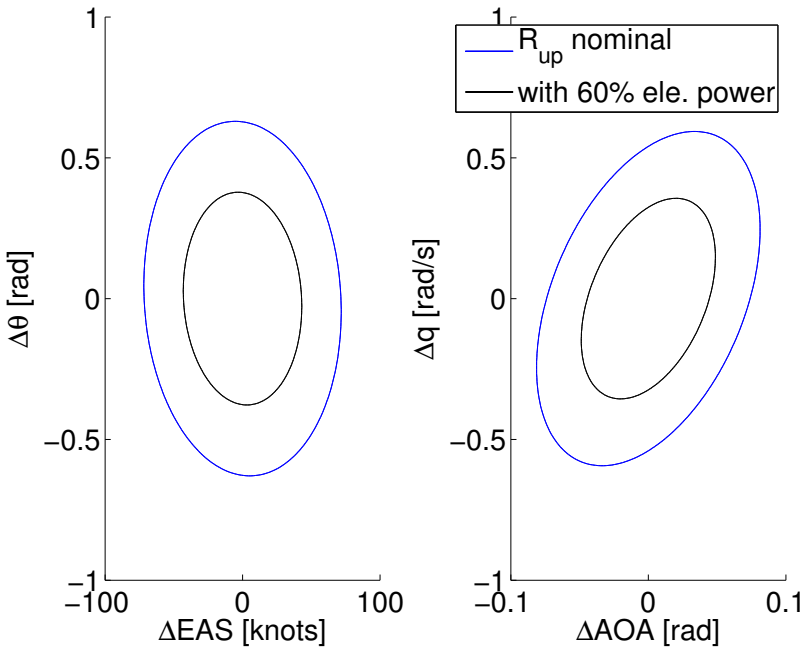


Figure 4.13: Effect of elevator power reduction on $\mathcal{R}_{up}$

further away.

The linear LMI approach is versatile enough to analyze effects of simple changes in system dynamics. For example, Figs. 4.16 show the effect of actuator bandwidth reduction. A scenario like this is possible in reality if say hydraulic pressure driving the servos is reduced due to engine malfunction. In this example, actuator bandwidth is reduced from 7Hz to 2Hz. The solid black line ellipsoid is the ellipsoid with reduced bandwidth. The detrimental effect is immidiately obvious from the figure, that size of the reachbale set is reduced considerably. From dynamic flight envelope assessment perspective this is benefecial because LMI computations can be carried out very quickly on avionics grade computing hardware and the effect of actuator bandwidth reduction can be quickly translated into more meaningful changes in flight envelope limits and presented to the flight crew or the autopilot system. Colored bands in the Fig. 4.16 illustrate how safe and unsafe regions of the state-space be presented to pilots in an easy to understand manner.

The examples so far use linear tools which are computationally efficient can be used for real-time computation. The linear tools allow simple failures to be included in the analysis like control authority reduction, actuator model changes, etc. Damage to the airframe can also be considered in analysis if an equivalent system dynamics matrix A can be obtained. The main drawback of this approach is that the results are valid only locally where the

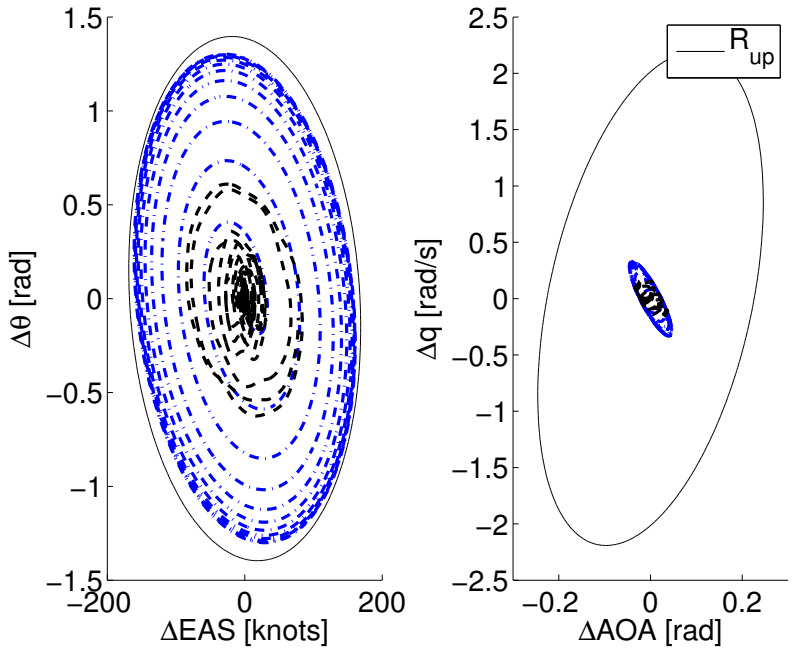


Figure 4.14: $\mathcal{R}_{up}$ validation with Monte Carlo linear simulations

linear model is valid.

4.3.2 Applications of nonlinear reachable-sets analysis

Applications of sum-of-squares based nonlinear reachable-set analysis is presented in this sub-section. Polynomial GTM longitudinal dynamics model is considered. SOS formulation and associated algorithms presented previously in this chapter are used to compute upper and lower bounds of reachable sets with energy and magnitude constraints.

4.3.2.1 $\mathcal{R}_{ue}$ upper-bound computation example

In this example, the effect of complexity of the chosen storage function is analyzed. As mentioned in course of the chapter, higher degree storage function allows for more complex geometry and hence can approximate the true irregular nonlinear reachable set more accurately. The potential benefit is at the cost of increased computational complexity. It is known that the associated SDP problem (see chapter 2) scales poorly with increase in complexity of polynomials in the SOS problem.

Nonlinear $\mathcal{R}_{ue}$ upper-bound approximation for polynomial GTM model is shown in Fig. 4.17. Reachable sets ($p(x) < \beta$) are plotted for various discrete values of input-energy

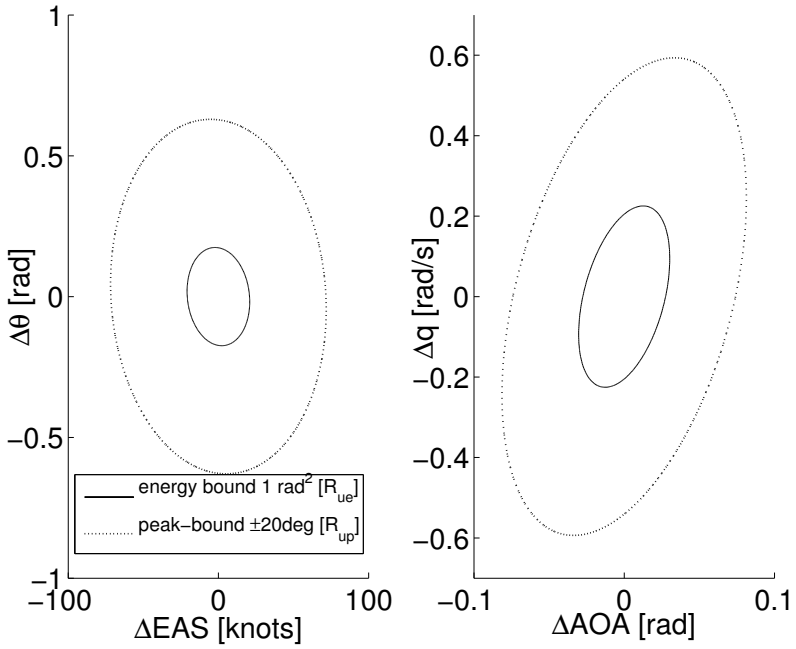
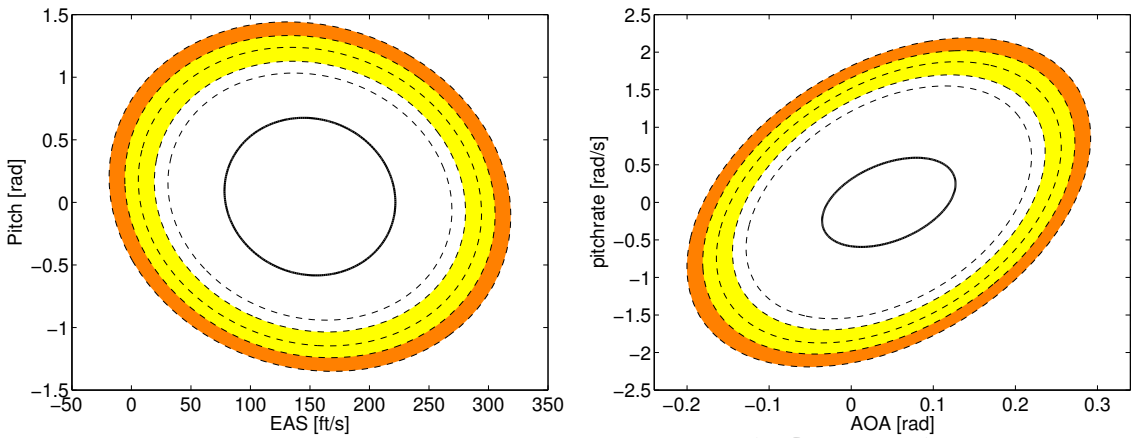


Figure 4.15: Comparison of linear $\mathcal{R}_{ue}$ and $\mathcal{R}_{up}$ for the GTM

constraints (γ). It can be seen that quadratic storage function ($V(x)$) quickly blows up and is insufficient to provide tighter upper bound. For comparison linear $\mathcal{R}_{ue}$ ellipsoid is also plotted. Using a quartic storage function, i.e. a storage function which contains upto degree four monomials, much tighter $\mathcal{R}_{ue}$ bounds are obtained as shown. Unfortunately, only three data points could be computed as the problem is computationally very expensive taking upto an hour for each data point on high performance desktop PC. This example highlights the importance of how the storage functions are selected. One may start with a quadratic $V(x)$, but if bounds prove to vague, then complexity of $V(x)$ may be increased to achieve tighter bounds.

4.3.2.2 Example: Computing the lower-bound

Computation of the lower bound of the reachable set is considered in this section. Polynomial model is considered in this work for two reasons: one, the upper bound computation is based on Sum-of-squares (SOS) optimization which is based on polynomial systems, and two, the algorithm for lower bound computation described in Tierno and Murray, [85] and used in Tan et al., [82] is also based on polynomial dynamics models. To make a meaningful comparison, it only makes sense to use identical models are used in all the test cases.



(a) $\mathcal{R}_{up}$ EAS-θ plane (b) $\mathcal{R}_{up}$ $q - \alpha$ plane

Figure 4.16: Effect of actuator bandwidth reduction on $\mathcal{R}_{up}$ of GTM

The GTM longitudinal dynamics model described in chapter 3 consists of four states - equivalent airspeed ($U, ft/s$), angle of attack (α, rad), pitch-rate ($q, rad/s$) and pitch-attitude (θ, rad). Two control inputs are available, the engine throttle position (δ_{th}) and the elevator deflection (δ_e, rad). The polynomial model is trimmed for straight and level cruise flight at $U = 150 ft/s$, the states and control inputs in the polynomial model are *trim-shifted*, so that the origin ($x = 0$) is the equilibrium. This simplifies computation. For reachability analysis only elevator control input (δ_e) is considered, the throttle (δ_{th}) is held fixed at its trim value.

Using the algorithm described in Section 4.3.2.2, lower bounds on the reachable set, $\mathcal{R}_{ue}$ are computed for different values of maximum input energy bound (γ) for the polynomial GTM dynamics model. A finite time horizon of $T = 200$ seconds is chosen. For comparison, the lower bound is also computed using the algorithm described in [85], as well as $\mathcal{R}_{ue}$ for linear model is computed using the linear matrix inequality (LMI) formulation presented in [13]. The lower bound of the $\mathcal{R}_{ue}$ for the GTM model is shown in Fig. 4.18.

The time taken for $\mathcal{R}_{ue}$ computation for each input energy bound (γ) on a Quad-core CPU clocked at 2.67GHz in MATLAB environment is given in Table. I. It is evident that the algorithm proposed in this thesis is orders of magnitude faster than the other method for computing lower bound for a nonlinear system.

The reachable set bounds for input energy constraint (γ) ranging from 0.01 to 1.5 is plotted in Fig. 4.18. Only two data points for the two-point boundary value power algorithm, [85], based algorithm are plotted. It is noteworthy that the proposed algorithm based on the com-

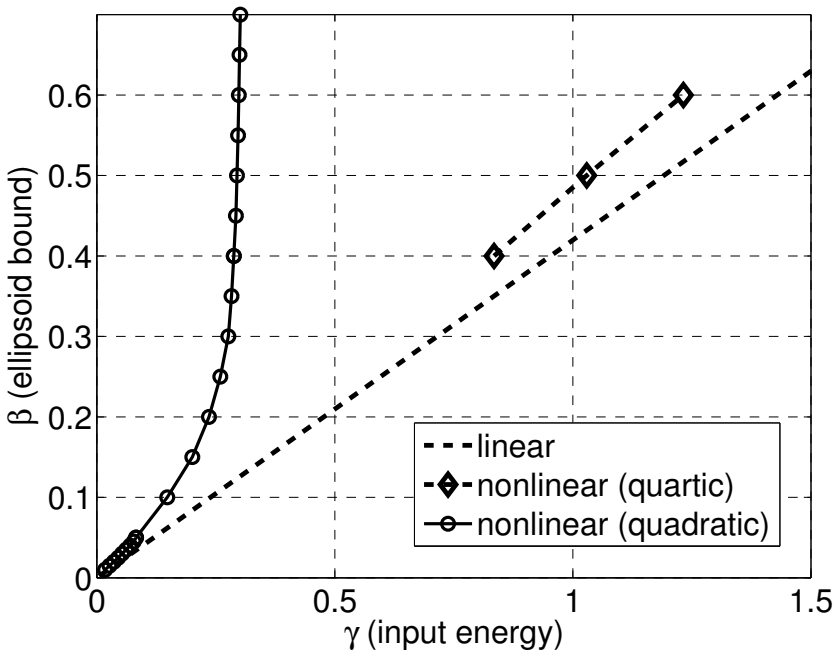


Figure 4.17: $\mathcal{R}_{ue}$ for nonlinear GTM model: Effect of quartic vs quadratic storage function

Table 4.1: CPU time (seconds) for computing the $\mathcal{R}_{ue}$ lower bound

Optimal input based algorithm	Two-point boundary value problem based algorithm	Linear model LMI solution
10.4	7280	2.5

puted optimal control inputs always returns a better lower bound for the polynomial GTM model compared to the algorithm presented in [85]. Although, it is not clear at this point whether the same comparative performance can be expected for all systems in general. Based on the example presented in this section, it is quite clear that the proposed algorithm for computing lower bounds could be a useful, computationally efficient alternative.

4.3.2.3 Compuation of $\mathcal{R}_{up}$ reachable-sets for nonlinear GTM dynamics

Example of reachable set with peak-input constraint for polynomial GTM dynamics is presented. SOS feasibility conditions given in Corollary 4.2.5 are used. The resulting SOS problem more complex than the $\mathcal{R}_{ue}$ problem due to additional number of multipliers as well as polynomial terms in each SOS constraint. A feasible solution couldn not be found for the four the four state degree five polynomial GTM model used so far in the examples. As a workaround a reduced short-period GTM longitudinal model with two states, angle-of-attack and pitch-rate and degree five was attempted successfully. The short-period model

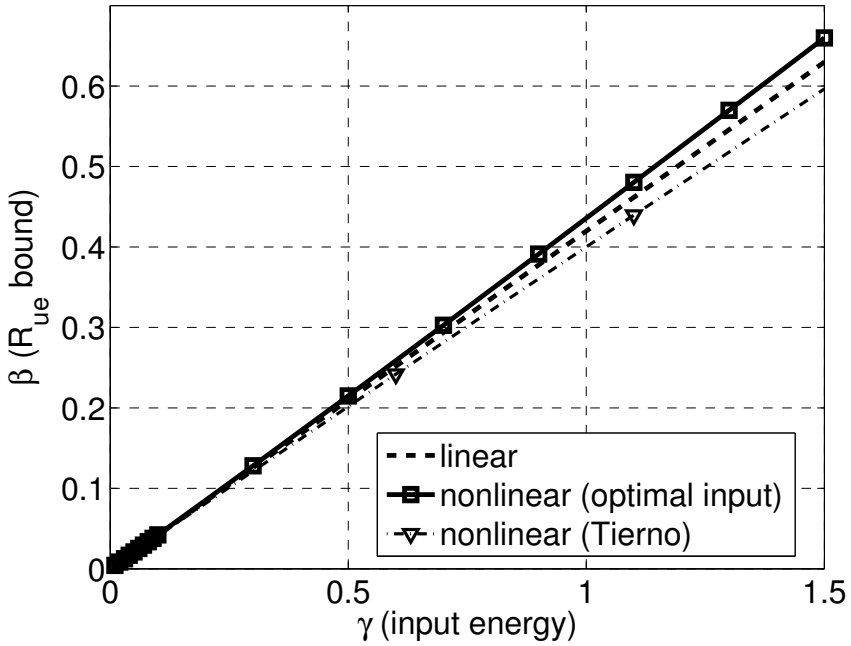


Figure 4.18: GTM Reachable set lower-bound comparison, T=200 seconds

of the GTM is obtained by plugging in the trim values of EAS and θ in the four state GTM poly model, i.e. assuming EAS and θ to be constant. The resulting model captures the dominant fast mode of the aircraft dynamics, the short-period mode. Fig. 4.19 shows plot of feasible reachable sets obtained for different values of β ranging from 0.5 to 2.5. For verification, one value of $\beta = 1.5$ is chosen and monte-carlo simulation of the polynomial GTM short-period model is carried out using random bounded peak elevator input signals ($u^u \leq \gamma$, $\gamma = 0.0051$ found from solving the SOS problem for $\beta = 1.5$) as shown in Fig. 4.20.

The lower-bound is computed using a trivial approach making use of basic linear systems theory. It is well known that for an LTI system, the maximum "distance" that the states can be pushed is related to the maximum gain of the system or the peak in magnitude bode plot. This corresponds to the resonant frequency of the lightest damped mode. Using this idea we simulate the polynomial GTM model while exciting it with a sinusoidal frequency corresponding to the only mode, the short-period frequency. Magnitude of sinusoid is the γ value found from the upper-bound computation. The maximum magnitude of states (x_i^2) found from this single time-simulation is the lower-bound of the $\mathcal{R}_{up}$ reachable set. Fig. 4.20 shows both the bounds for GTM short-period model and the monte-carlo time simulations used for verification indicate that the bounds are indeed reasonable.

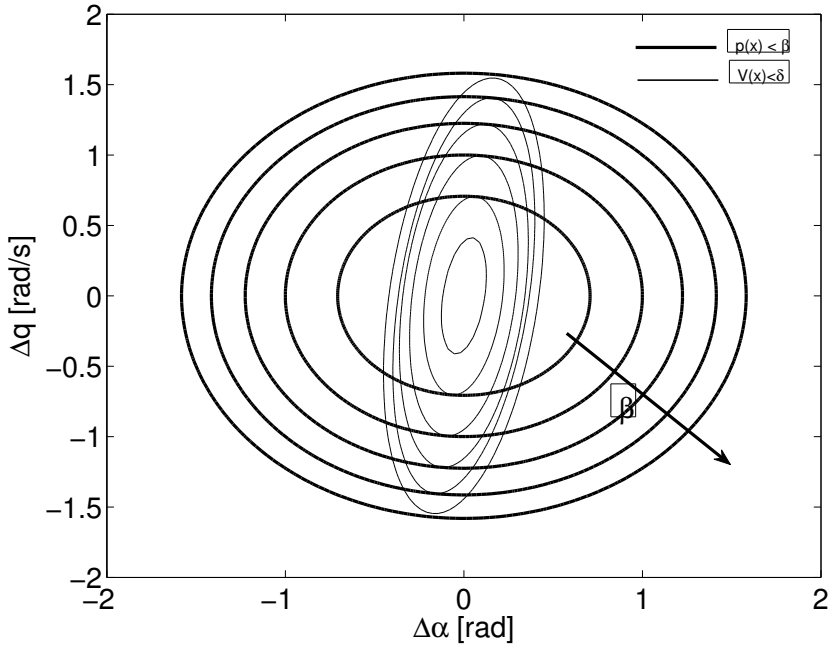


Figure 4.19: $\mathcal{R}_{up}$ for nonlinear GTM model

4.4 Summary

The examples presented in this chapter for computation of reachable sets provide methods to compute reachable sets for linear as well as nonlinear dynamical systems. Linear analayis is computationally efficient and can quickly provide reachable sets for various damage/actuation failure scenarios discussed. The analysis maybe only very locally valid, but it can still offer valuable insights about which "directions" of the state-space are more controllable or vice-versa. Nonlinear SOS based methods work with nonlinear models hence provide more relevant bounds but are computationally expensive and out of the realm of realtime implementation at least at this point in time. Future advances in SDP solvers as well as computational hardware can solve these problems.

From a practical standpoint of flight safety, the knowledge of reachable sets gives the pilots and indication of the availble control authority. This can be particularly useful in the event of any failure associated with the actuators, control surfaces or the airframe itself. Assuming that a online parameter estimator is available to estimate the changes in aerodynamic parameters that affect aircraft dynamics. Reachable sets computation can be used to annunciate any changes (reduction) in the flight envelope that might occur as a result of a sub-system failure (for example, partial loss of hydraulic pressure) or damage. This new

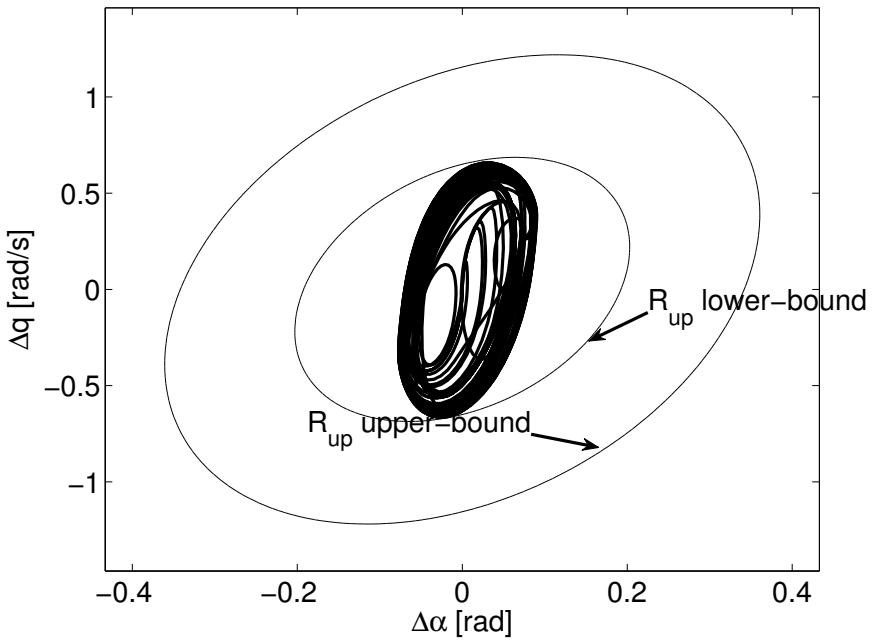


Figure 4.20: $\mathcal{R}_{up}$ for nonlinear GTM model: Monte-carlo verification of upper and lower bounds

information can also be used to reset flight-envelope protection system limits to new safe values.

Chapter 5

Region-of-Attraction analysis

Linear analysis is a local analysis which is only valid near an equilibrium point. Many systems, including the GTM aircraft, are more accurately described by nonlinear dynamical systems. For linear systems asymptotic stability of an equilibrium point is a global property. In other words, if the equilibrium point is asymptotically stable then the state trajectory will converge back to the equilibrium when starting from any initial condition. A key difference with nonlinear systems is that equilibrium points may only be locally asymptotically stable. Khalil [51] and Vidyasagar [95] provide good introductory discussions of this issue. The region-of-attraction (ROA) of an asymptotically stable equilibrium point is the set of initial conditions whose state trajectories converge back to the equilibrium [51]. If the ROA is small, then a disturbance can easily drive the system out of the ROA and the system will then fail to come back to the stable equilibrium point. Thus the size of the ROA is a measure of the stability properties of a nonlinear system around an equilibrium point. This provides the motivation to estimate the ROA for an equilibrium point of a nonlinear system. In this section the technical approach to estimating the ROA is presented.

5.1 Region-of-attraction analysis: Theory

Consider autonomous nonlinear dynamical systems of the form:

$$\dot{x}(t) = f(x(t)),\ x(0) = x_0 \tag{5.1}$$

where $x(t) \in \mathbb{R}^n$ is the state vector at time t. Without loss of generality, we assume that $x = 0$ is a locally asymptotically stable equilibrium point. Formally, the ROA is defined as

the set,

$$R_0 = \left\{ x_0 \in \mathbb{R}^n \mid \forall\, x(0) = x_0, \dot{x}(t) = f(x(t)), \text{then} \lim_{t\to\infty} x(t) = 0 \right\} \tag{5.2}$$

Computing the exact ROA for nonlinear dynamical systems is, in general, a very difficult problem. There has been significant research devoted to estimating invariant subsets of the ROA [18, 22, 30, 33, 32, 64, 84, 83, 91]. As in the case of computing reachable sets, our approach is to restrict the search to ellipsoidal approximations of the ROA. The ellipsoid is specified by $\{x_0^T N x_0 \leq \beta\}$ where $N = N^T > 0$ is a user-specified matrix which determines the shape of the ellipsoid. Given N, the problem is to find the largest ellipsoid contained in the ROA:

$$\begin{aligned} \beta^* = &\max \beta \\ &\text{subject to: } \{x_0^T N x_0 \leq \beta\} \subset R_0 \end{aligned} \tag{5.3}$$

Determining the best ellipsoidal approximation to the ROA is still a challenging computational problem. Instead we will attempt to solve for upper and lower bounds satisfying $\underline{\beta} \leq \beta^* \leq \bar{\beta}$. If these upper and lower bounds are close then we have approximately solved the best ellipsoidal approximation problem given in Equation 5.3. However, it may be noted that in case that the true ROA of a system is a skewed geometrical shape, the upper and lower bound may not converge to the same value after all.

Fig. 5.1 is an illustration of the ROA problem for a nonlinear system. True ROA may be an irregular shape shown by the shaded area ($dV(x)/dt < 0$). In such cases, the upper and lower bound depicted by the ellipsoids in solid and dashed lines, respectively may not converge.

The upper bounds are computed via a search for initial conditions leading to divergent trajectories. If $\lim_{t\to\infty} x(t) = +\infty$ when starting from $x(0) = x_{0,div}$ then $x_{0,div} \notin R_0$. If we define $\bar{\beta}_{div} := x_{0,div}^T N x_{0,div}$ then $\{x_0^T N x_0 \leq \bar{\beta}_{div}\} \not\subset R_0$. Thus $\bar{\beta}_{div}$ is a true upper bound on β^* and $\{x_0^T N x_0 \leq \bar{\beta}_{div}\}$ is an outer approximation of the best ellipsoidal approximation to the ROA. An exhaustive Monte Carlo search is used to find the tightest possible upper bound on β^*. Specifically, initial conditions are randomly chosen starting on the boundary of a large ellipsoid: Choose x_0 satisfying $x_0^T N x_0 = g$ where g is sufficiently large that $g \gg \beta^*$. If a divergent trajectory is found, then an upper bound $\bar{\beta}_{div}$ on β^* is computed by finding the minimum value of p along the divergent trajectory. The state vector x_{div} that

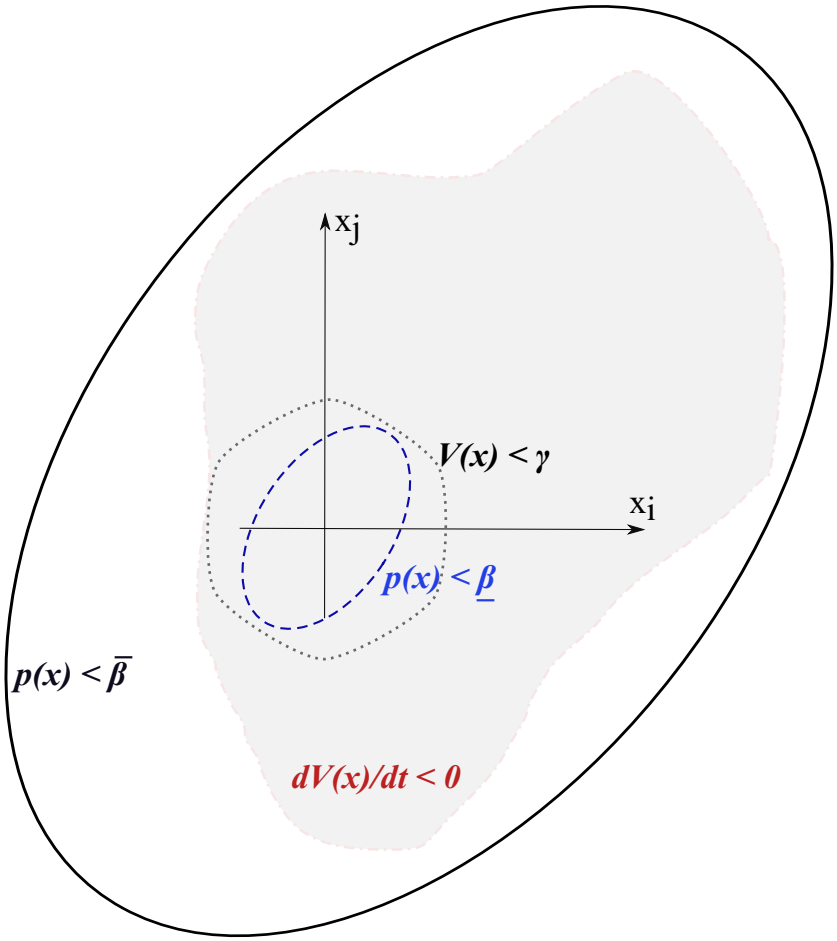


Figure 5.1: Region-of-attraction illustration of an equilibrium point of a nonlinear system

achieves $p(x_{div}) = \bar{\beta}_{div}$ is stored and g is decreased by a small factor, e.g. $g = 0.995\bar{\beta}_{div}$. The search continues until a maximum number of simulations is reached. $\bar{\beta}_{MC}$ will denote the smallest upper bound computed with this Monte Carlo search. This upper bound is illustrated in Fig. 5.1 by the ellipsoid in solid line.

The lower bounds are computed using an SOS formulation of the ROA problem. To compute these bounds we need to further assume that the vector field $f(x)$ in the system dynamics (Eq. 5.1) is a polynomial function. The computational algorithm is briefly described in the following paragraphs. Additional details are provided in literature references [40, 41, 80, 42, 79, 89, 88, 87]. Lemma 5.1.1 is the main Lyapunov theorem used to compute lower bounds on β^*. This specific lemma is proved by Tan [79] but very similar results are given in textbooks, e.g. by Vidyasagar [95].

Lemma 5.1.1. *If there exists a continuously differentiable function $V : \mathbb{R}^n \to \mathbb{R}$ such that:*

- $V(0) = 0$ *and* $V(x) > 0 \; \forall \; x \neq 0$
- *Sub-level set* $\Omega_\gamma := \{x \in \mathbb{R}^n \; : \; V(x) \leq \gamma\}$ *is bounded.*
- $\Omega_\gamma \subset \{x \in \mathbb{R}^n \; : \; \nabla V(x) f(x) < 0\}$

then for all $x \in \Omega_\gamma$*, the solution of Eq. 5.1 exists, satisfies* $x(t) \in \Omega_\gamma$ *for all* $t \geq 0$*, and* $\Omega_\gamma \subset R_0$.

A function V satisfying the conditions in Lemma 5.1.1 is a Lyapunov function and Ω_γ provides an estimate of the region of attraction. Any subset of Ω_γ is also inside the ROA. In principle we can compute a lower bound on β^* by solving the maximization:

$$\begin{aligned} \underline{\beta} := & \max \beta \\ & \text{subject to: } \{x_0^T N x_0 \leq \beta\} \subset \Omega_\gamma \end{aligned} \tag{5.4}$$

The ROA lower bound is illustrated by the ellipsoid in dashed line in Fig. 5.1. Our computational algorithm replaces the set containment constraint with a positive-definiteness condition using the $\mathcal{S}$-procedure (see Sec.2.4), which provides a sufficient condition for the set-containment problem:

$$\begin{aligned} \underline{\beta} := & \max \beta \\ & \text{subject to: } s(x) \geq 0 \; \forall x \\ & \qquad -(\beta - x^T N x) s(x) + (\gamma - V(x)) \geq 0 \; \forall x \end{aligned} \tag{5.5}$$

The function $s(x)$, known as a *multiplier function*, is a decision variable of the optimization, i.e. it will be found as part of the optimization. It is straight-forward to show that the two non-negativity conditions in the optimization in Eq. 5.5 are a sufficient condition for the set containment condition in Optimization 5.4. If both $V(x)$ and $s(x)$ are restricted to be polynomials then both constraints involve the non-negativity of polynomial functions. A sufficient condition for a generic multi-variate polynomial $p(x)$ to be non-negative is the existence of polynomials $\{g_1, \ldots, g_n\}$ such that $p = g_1^2 + \cdots + g_n^2$. A polynomial which can be decomposed in this way is appropriately called a sum-of-squares (SOS). Finally, if we replace the non-negativity conditions in the optimization in Eq. 5.5 with SOS constraints, then we arrive at an SOS optimization problem:

$$\begin{aligned} \underline{\beta} := & \max \beta \\ & \text{subject to: } s(x) \text{ is } \Sigma[x] \\ & \quad -(\beta - x^T N x)s(x) + (\gamma - V(x)) \text{ is } \Sigma[x] \end{aligned} \tag{5.6}$$

The choice of the Lyapunov function which satisfies the conditions of Lemma 5.1.1 has a significant impact on the quality of the lower bound, $\underline{\beta}$. The simplest method is to find $P > 0$ which solves the Lyapunov equation $A^T P + PA = -I$. $A := \frac{\partial f}{\partial x}\big|_{x=0}$ is the linearization of the dynamics about the origin. $V_{LIN}(x) := x^T P x$ is a quadratic Lyapunov function since $x = 0$ is assumed to be a locally asymptotically stable equilibrium point. Thus we can solve for the largest value of γ satisfying the set containment condition in Lemma 5.1.1: $\Omega_\gamma \subset \{x \in \mathbb{R}^n : \nabla V_{LIN}(x) f(x) < 0\}$. This problem can also be turned into an SOS optimization with "multiplier" functions as decision variables. $\underline{\beta}_{LIN}$ will denote the lower bound obtained using the quadratic Lyapunov function obtained from linearized analysis.

Unfortunately, $\underline{\beta}_{LIN}$ is typically orders of magnitude smaller than the upper bound $\bar{\beta}_{MC}$. Several methods to compute better Lyapunov functions exist, including V-s iterations [40, 41, 80, 42], bilinear optimization [79], and use of simulation data [89, 88, 87]. We briefly describe the V-s iteration starting from V_{LIN}. In the first step of the iteration, the multiplier functions and $\underline{\beta}_{LIN}$ are computed. Then the multiplier functions are held fixed and the Lyapunov function candidate becomes the decision variable. The SOS constraints of this new problem are those which arise from the two set containment conditions: $\Omega_\gamma \subset \{x \in \mathbb{R}^n : \nabla V_{LIN}(x) f(x) < 0\}$ and $\{x_0^T N x_0 \leq \beta\} \subset \Omega_\gamma$. In the next iteration, the multiplier functions are again decision variables and a lower bound is computed using the new Lyapunov function computed in the previous iteration. The V-s iteration continues as long as the lower bound continues to increase. In this iteration, Lyapunov functions are allowed to be of higher polynomial degree. Increasing the degree of the Lyapunov function will improve the lower bound at the expense of computational complexity. The computational time grows very rapidly with the degree of the Lyapunov function and so degree 4 candidates are about the maximum which can be used for problems like the GTM analysis. $\underline{\beta}_2$ and $\underline{\beta}_4$ will denote the best lower bounds computed with the V-s iteration for quadratic and quartic Lyapunov functions. Nonlinear analysis code based on SOS optimization (including demos of the V-s iteration) can be downloaded from the URL [4]

Theorem 5.1.2 (SOS conditions for ROA lower bound)**.** *For a nonlinear system governed by polynomial dynamics* $\dot{x} = k(x,u),\ k : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^n$*, if* $\exists\, V(x) : \mathbb{R}^n \mapsto \mathbb{R}$ *such that,*

$$
\begin{aligned}
V(x) - l_1 \in & \Sigma[x] && \textit{(a)} \\
-\dot{V}(x) - s_1(\gamma - V(x)) \in & \Sigma[x] && \textit{(b)} \\
(\gamma - V(x)) - s_2(\beta - p) \in & \Sigma[x] && \textit{(c)} \\
s_1, s_2, l_1 \in & \Sigma[x] && \textit{(d)}
\end{aligned}
$$

$$
\beta > 0, \gamma > 0 \quad \textit{(e)} \tag{5.7}
$$

then, $p_\beta(x) \subset V_\gamma(x) \subset R_0$.

The ROA computation presented above is equally applicable open-loop and closed-loop systems. For application example presented in this thesis, the GTM longitudinal dyanmics is considered. A simple pitch-controller is described in the next section. This will enable us to compute ROA for the GTM example for both open-loop and the closed-loop case. It will interesting to compare how a closed-loop controller might affect the ROA of a nonlinear system.

5.1.1 The ROA SOS algorithm: V-s iteration

1. Given polynomial dynamics $\dot{x} = k(x,u)$, equilibrium states, x_e and control inputs, u_e.

2. (*Trim-shift*) By letting $x = x + x_e, u = u + u_e$ in $k(x,u)$, the equilibrium is shifted to x=0, u=0, such that $k(0,0) = 0$.

3. (*Bootstrap*) Linearize the nonlinear model $\dot{x} = k(x,u)$ to get $\dot{x} = Ax + Bu$. Compute a linear Lyapunov function $V_{LIN}(x) = x^T P x, P = P^T > 0$ by solving $A^T + PA = -Q$, where $Q = Q^T > 0, Q \in \mathbb{R}^{n \times n}$ is an arbitrarily chosen matrix. Use $V = V_{LIN}(x)$ in the first iteration to solve the SOS problem given below, subsequently use $V(x)$ obtained from V-step.

4. (*s-step*) Solve the bilinear SOS set-containment condition in Eq. 5.7(b) for s_1 and γ. Then solve the SOS problem in Eq. 5.7(c) for β and s_2 using γ obtained from previous SOS solution.

5. (*V-step*) Fix s_1, s_2, γ, β, solve SOS problems Eq. 5.7(b),(c) for a new feasible $V(x)$ of user specified degree. The $V(x)$ can be a degree two or higher polynomial.

6. Iterate between V and s step, increasing γ and β value at each step. Terminate the iteration based on a user specified tolerance criteria to indicate when the iteration stops making substantial progress in γ, β increment. The β thus obtained is a lower bound for the region-of-attraction R_0. Call it $\underline{\beta}$.

5.1.2 Closed-loop GTM model: Pitch command tracking

A simple Proportional-Integral (PI) controller for pitch attitude tracking is presented and the polynomial model for the closed-loop system is derived. The controller block diagram is shown in Fig.5.2. The input to the controller is the pitchattitude tracking error and controller output is elevator demand. Throttle control is not used in the closed loop, hence is kept fixed at its trim value for all analysis and simulation, unless specified.

The presence of integrator in the controller introduces an additional state into the system in place of elevator input. Assuming constant pitch reference command, the elevator state derivative can be obtained using control law described in Fig.5.2 as,

$$\dot{\delta_e} = -20\varepsilon(5(\theta_{ref} - \theta) - \dot{\theta}) \tag{5.8}$$

where, $\dot{\theta}$ is obtained from GTM polynomial dynamics, pitch state derivative (see Appendix-A.2). Substituting for the elevator state derivative, a new closed-loop state-space in $\mathbb{R}^5$ is obtained. To avoid repetition, the full closed-loop polynomial equations are not presented, but the same may be obtained easily by combining Eq. 5.8 with the polynomial GTM dynamics (Appendix-A.2). In Fig.5.3, the closed loop response to a step input is presented, with initial conditions at straight and level flight at EAS$= 150$ ft/s or 90 knots.

5.2 Example

The region of attraction of the open-loop GTM polynomial model is estimated about the identical trim condition as before, EAS = 150 ft/sec and steady/level flight (flight path angle

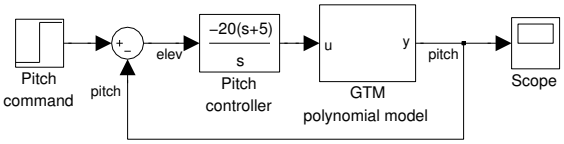


Figure 5.2: GTM closed loop interconnection block diagram with pitch tracking controller

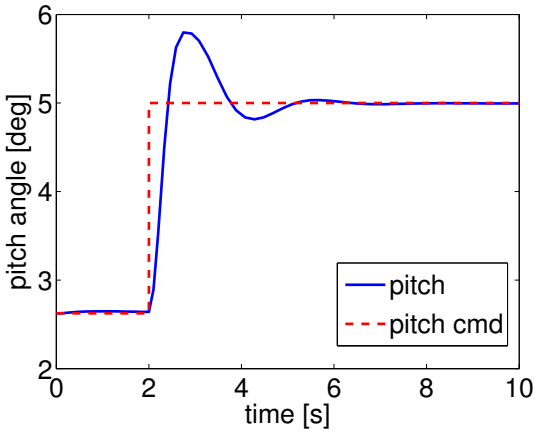


Figure 5.3: Nominal step response of the closed loop GTM polynomial model

= 0 deg). The state and input for this trim condition are:

$$
\begin{aligned}
U &= 150 \text{ ft/sec} \\
\alpha &= 0.0458 \text{ rads} = 2.62 \text{ degs} \\
q &= 0 \text{ rads/sec} = 0 \text{ degs/sec} \\
\theta &= 0.0458 \text{ rads} = 2.62 \text{ degs} \\
\delta_{thr} &= 0.0859 \text{ (Normalized)} \\
\delta_e &= 0.0463 \text{ rads} = 2.66 \text{ degs}
\end{aligned}
$$

To improve the numerical conditioning, the states are shifted and scaled using the scaling matrix D: $z := D^{-1}(x - x_{trim})$ where:

$$D := diag(50\text{ft/sec}, 20\text{deg}, 50\text{deg/sec}, 20\text{deg})$$

The shape function is defined to be $p(z) := z^T z$. This is an ellipsoid in the original coordinates with radii along each coordinate given by the entries of D. All results were generated on a 2.66 GHz Intel Core 2 Quad processor PC.

First, a Monte Carlo search is run to find the smallest value of $p(z)$ resulting in a divergent trajectory. After 10000 simulations, the smallest value of $p(z)$ was $\bar{\beta}_{MC} = 3.13$. $\bar{\beta}_{MC}$ is an upper bound on the largest ellipsoid in the region of attraction. It took 1721 sec $\approx$ 29min

to run these 10000 simulations. The initial condition leading to a divergent trajectory was:

$$
\begin{aligned}
U &= 186.12 \text{ ft/sec} \\
\alpha &= -0.41 \text{ rads} = -23.48 \text{ degs} \\
q &= 0.008 \text{ rads/sec} = 0.45 \text{ degs/sec} \\
\theta &= 0.38 \text{ rads} = 21.66 \text{ degs}
\end{aligned}
$$

The divergent trajectory starting from this initial condition is shown in Figure 5.4. The EAS goes to zero around $t = 6$ sec and so the polynomial model is not valid for the entire simulation. The GTM aircraft is basically fully stalled by t=6s with zero airspeed, hence cannot sustain flight.

For computing the lower-bound, SOS optimization is used to find the largest value of $p(z)$ that is provably in the region of attraction. The provable region using the quadratic Lyapunov function from linear analysis was only $\underline{\beta}_{LIN} = 1.89 \times 10^{-5}$. This took 20.9 sec. Running one hundred steps of the V-s iteration with quartic Lyapunov functions increased the provable region to $\underline{\beta}_4 = 1.87$. This took 7074.8 sec $\approx$ 117.9 min. The progress of the V-s iteration with quartic Lyapunov functions for the open loop case is shown in Figure 5.7. This figure also shows the progress of Monte Carlo search for divergent trajectory as a function of the number of simulations . β_{MC} decreased rapidly in the first 300 simulations and then made only small progress thereafter, as it gets harder and harder to find an divergent initial condition as one moves closer to the equilibrium states.

The inner/outer bounds on the region of attraction can be visualized by plotting slices of the ellipsoid $p(z) = \beta$. Figure 5.9 shows a slice in the AOA-pitchrate plane along with that of the closed loop ROA analysis. All curves are drawn in the original coordinates with respect to the trim condition. The region of attraction cannot contain all of this ellipsoid since we found a divergent trajectory on the surface of this ellipsoid. Every initial condition of these ellipsoids was proven, via the SOS methods, to be in the region of attraction. Increasing the degree of the Lyapunov function enables the SOS optimization to prove a larger estimate of the region of attraction. It is possible that a degree 6 Lyapunov function could prove an even larger region but this would require significantly more computation time. Figure 5.10 shows another slice of the ellipsoid in the EAS-pitch plane.

These ROA analysis results indicate that in the EAS/α and θ/α planes the open-loop aircraft will return to the trim solution: $U = 150$ ft/sec, $\alpha = 2.62^\circ$, $q = 0^\circ/sec$, and $\theta = 2.62^\circ$

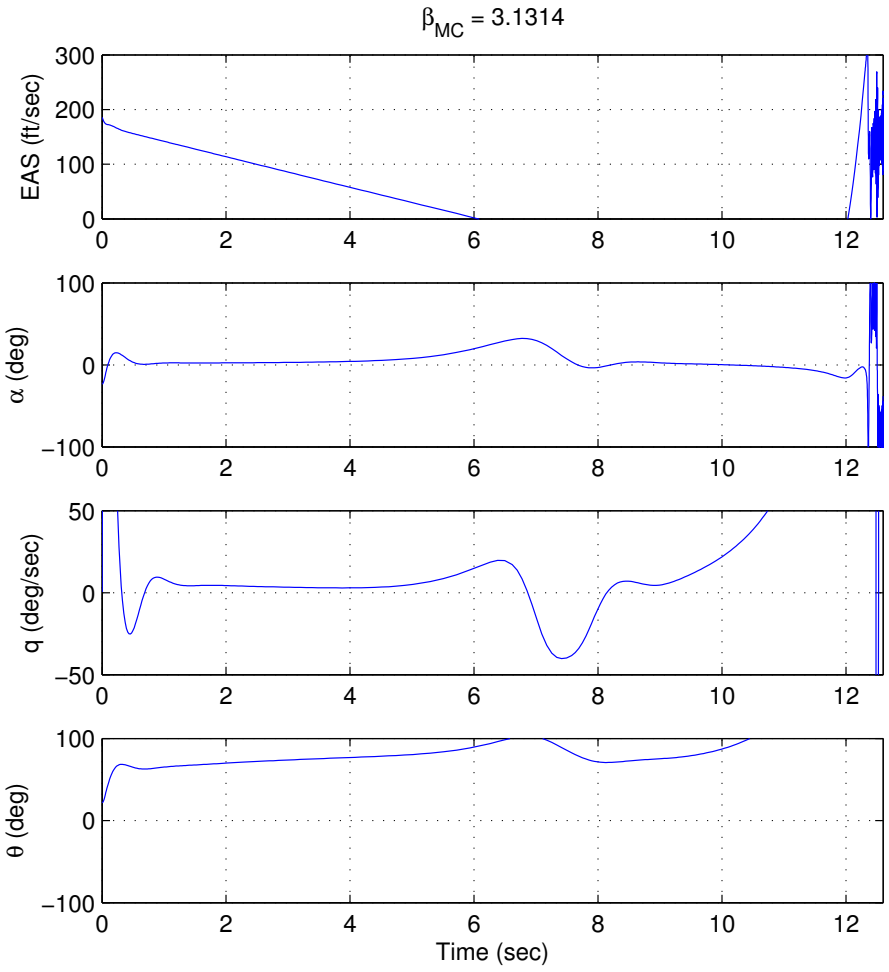


Figure 5.4: Divergent Trajectory from Monte Carlo Simulations

for initial perturbations in an ellipse defined by U between 80 and 220 ft/sec, θ and α perturbations between -24° and 30°.

Comparing ROA bounds for open loop GTM model with the closed loop model, we find from Fig.5.8 that provable stable region with SOS methods is larger for the closed loop case. β lower bound for the open loop case was found to be 1.87 while with the pitch controller wrapped around β was found to be 2.16. This indicates that presence of a well designed feedback controller actually enhances the ROA for a system like GTM. Corresponding to the β bound for the closed loop system, ROA ellipsoid projections are given alongwith that of the open loop case, in Figs.5.9,5.10. It may be noted that upper bound for β computed

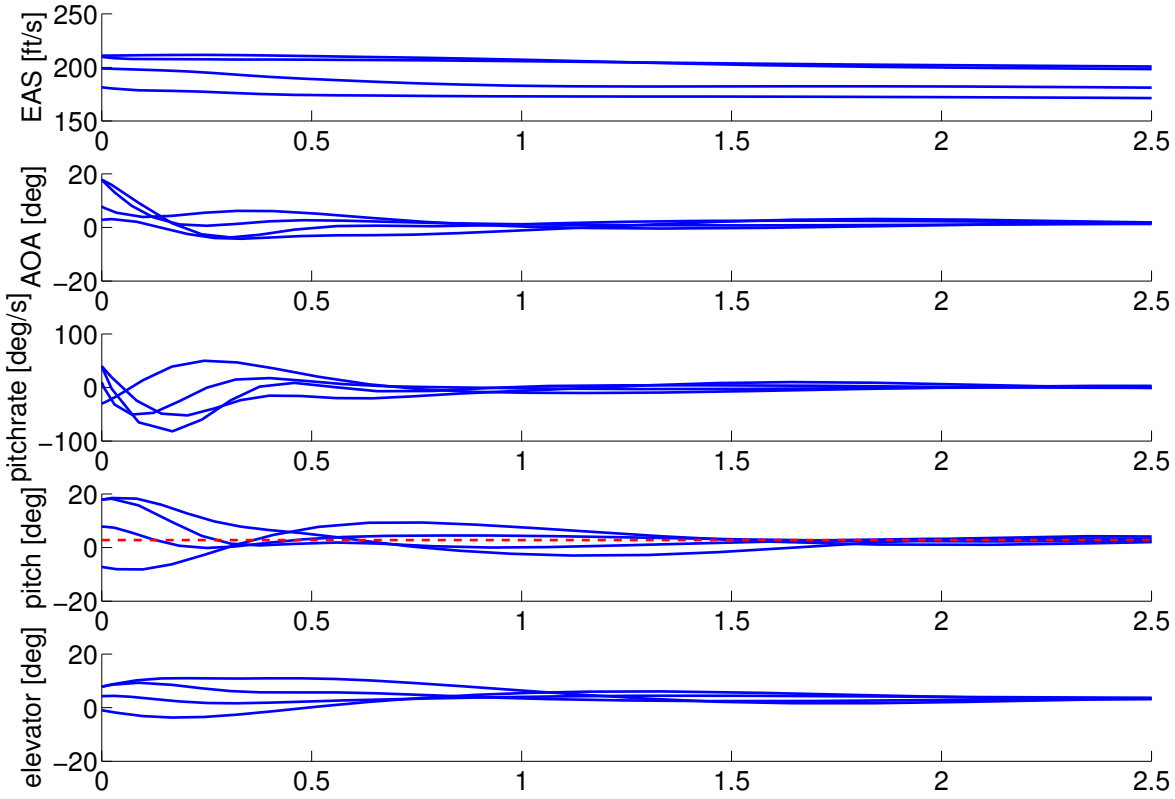


Figure 5.5: Stable convergent trajectories with initial conditions on the surface of the ellipsoid with proven stability, $\beta = 2.16$. Closed loop polynomial GTM model, elevator saturation included [± 20 deg].

using Monte Carlo simulation does not show the same trend. Its due to the stochastic nature of Monte Carlo sim and much larger number of iterations must be performed to arrive to a high confidence bound. Unfortunately, this is computationally intensive. Figs.5.9,5.10 also show rectangular boundaries to indicate the region of state-space for which the derived polynomial dynamics model is valid. This is to show that any results or claims lying outside the boundary are not valid strictly speaking. A solution maybe to find ways to extend the validity of the model, which in certain cases can be difficult while keeping degree of the polynomial model tractably small from numerical computation point of view. This is a main limitation of the SOS tools based ROA analysis at the moment.

In the Fig.5.5 simulation results for the closed loop GTM polynomial model is presented on

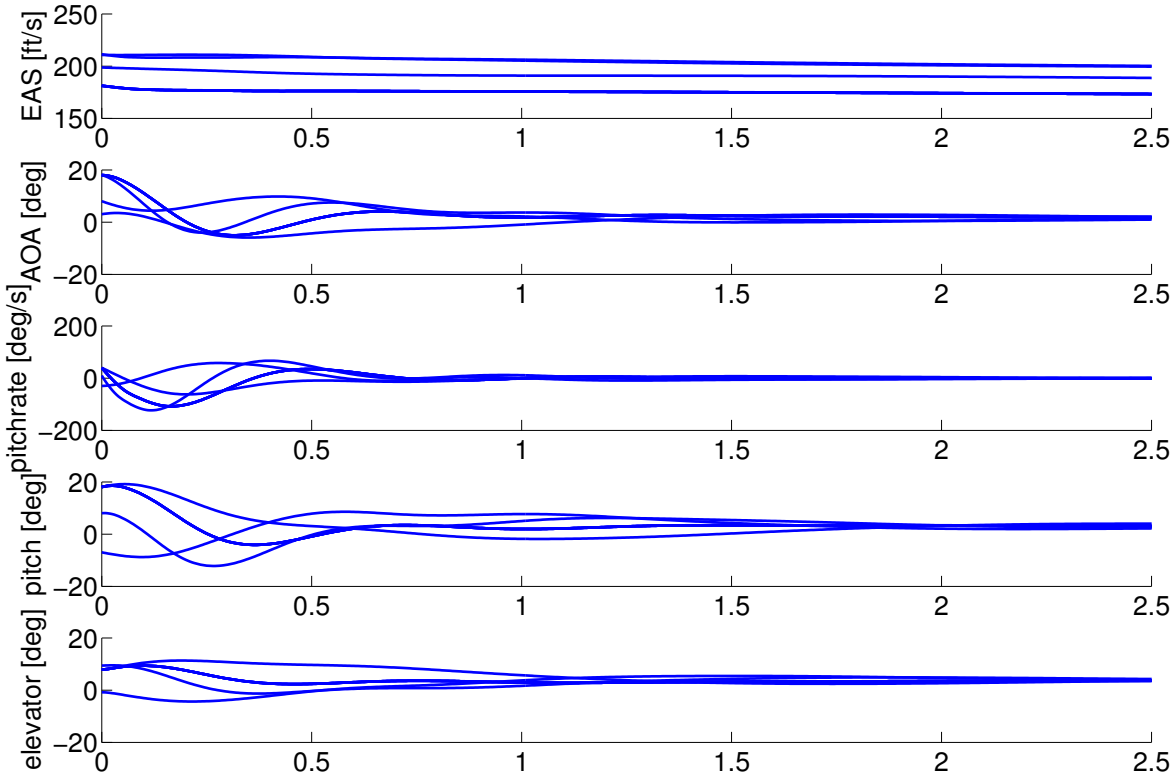


Figure 5.6: Full nonlinear closed loop GTM model simulation with initial conditions on proven ellipsoid boundary, $\beta = 2.16$.

which ROA analysis was carried out. elevator position constraints of± 20 degrees was also included. Initial conditions for the five states, including elevator state was chosen randomly on the surface for which guaranteed stability boundary was established as $\beta = 2.16$ in this section, also shown graphically in Figs.5.8, 5.9 and 5.10. It can be seen that the all initial conditions converge to the trim point as predicted by the analysis. In Fig.5.6 the same initial conditions are simulated in the full nonlinear GTM simulation model [46] with the same controller as described previously. Again, we find that the trajectories converge to the trim. Its worth noting that in the particular simulation instances the elevator limits were never exceeded so the effect of control constraints was not evident. It is entirely possible for any other dynamical system or a different control law that the control authority limits may be exceeded, the ROA analyis does not take that into account and may result in unstable trajectories despite of being proven otherwise. This may be seen as a limitation of the

current approach.

The other main limitation of SOS methods for ROA computation is the computational requirements and the fact that complexity of the problem scales badly with increase in dimensions of the system. At present research efforts are being to include actuator nonlinearities and hard limits in the ROA analysis. Integral quadratic constraints (IQC) based approaches [53] promise to be a step in that direction.

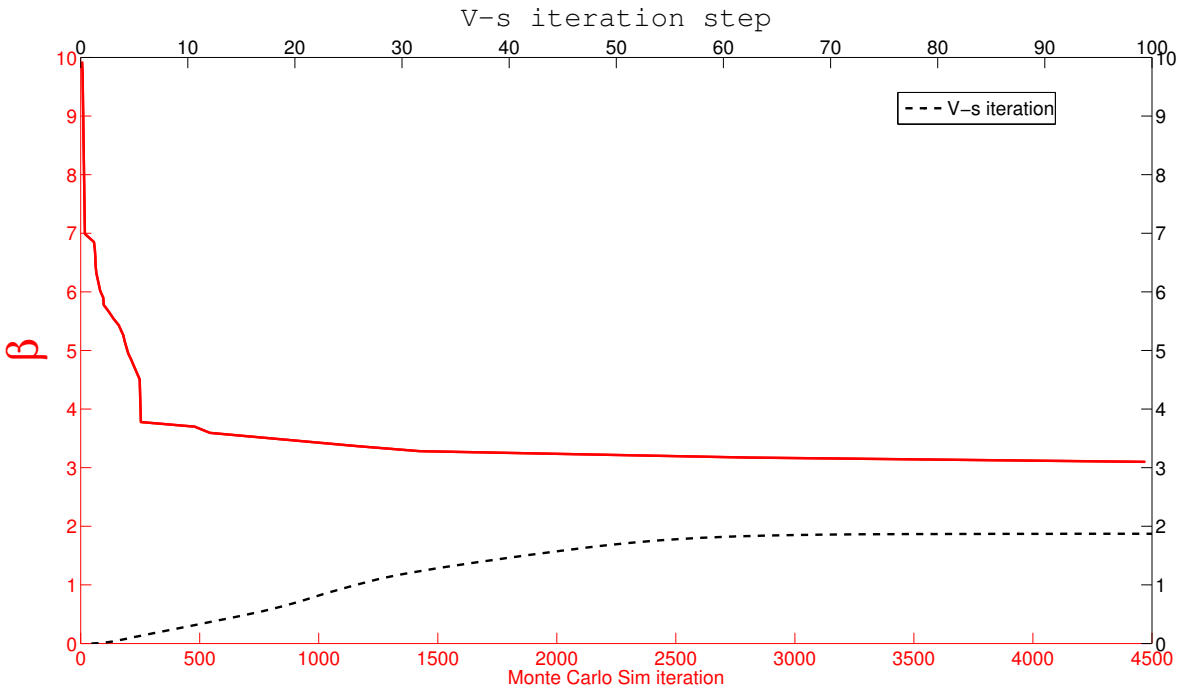


Figure 5.7: ROA bounds estimate, open loop GTM model

5.3 Summary

Region-of-attraction analysis is a very useful tool for safe flight envelope assessment. The knowledge of current operating point of an aircraft is available through various sensory measurements. By superimposing that information on computed ROA manifold of the nonlinear aircraft model it is possible to provide valuable advisory to flight crew as well as autopilot systems about impending departures. If an online parameter estimation system is available, then the ROA can be recomputed for an aircraft following a damage or acuator failure. Take the case of accident involving a Tu-154 in Ukraine in 2006 (see Chapter 1), due to extreme external wind disturbance, the aircraft control was lost in about 10 seconds.

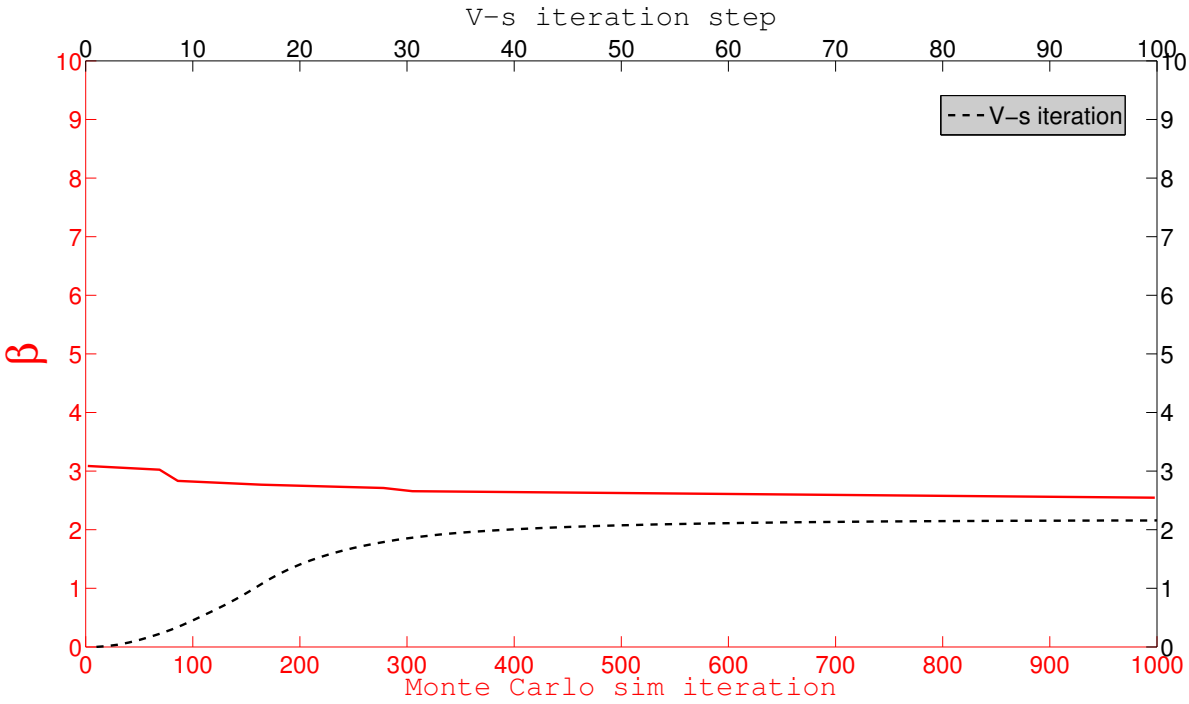


Figure 5.8: ROA bounds estimate, closed loop GTM model

Although the exact details of pilot reactions are unclear, a real-time display or annunciation of the aircrafts current states with respect to the ROA boundaries might have helped pilots understand urgency of the situation and take aggresive corrective actions.

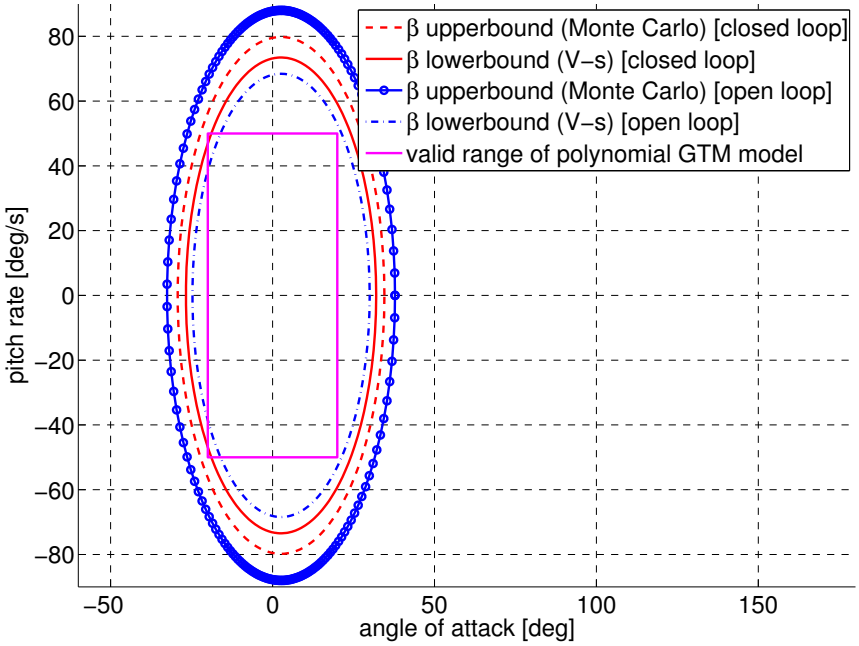


Figure 5.9: ROA ellipsoids comparison: open loop vs closed loop. Projected in AOA, pitch-rate plane

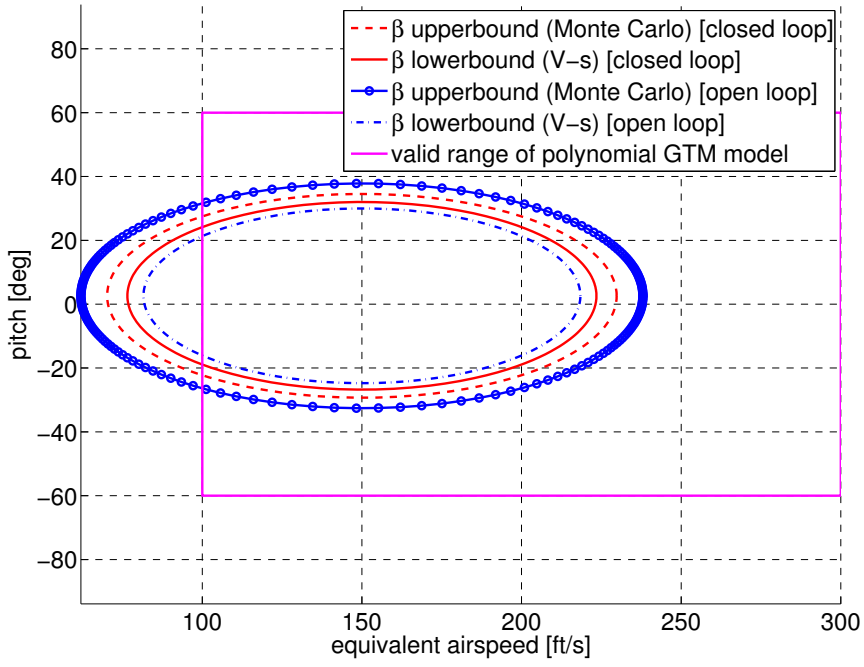


Figure 5.10: ROA ellipsoids comparison: open loop vs closed loop. Projected in EAS, pitch angle plane

Chapter 6

Closed-loop metrics for model-based fault diagnosis filters

Safety critical systems like flight control systems (FCS) require redundancy to guarantee fail-operational or fail-safe functionality. Traditionally this has been achieved through physical redundancy, requiring copies of sensors, computers, actuators and other important FCS sub-systems. This mechanism has been applied successfully in operational systems. However, the weight and cost penalty are significant. On the other hand, model-based fault diagnosis filters make use of a dynamic model of the system to identify and isolate faults. This approach has the potential to provide the same level of safety at a reduced cost and lower weight. A survey of the model-based fault diagnosis methods can be found in references, [17, 66, 36].

It is common practice for engineers to synthesize the feedback controllers independent of fault detection filters. Similarly, the model used for fault detection and isolation (FDI) filter design is often based on the open-loop system. This approach relies on knowledge of the system dynamics independent of the controller. The drawback of the independent filter/controller synthesis approach is the assumption that plant dynamics are precisely known, which is seldom the case in practical applications. FDI filters designed independently may suffer from poor performance, missed detections, false alarms, etc, due to coupling of the plant and controller and model errors in the system.

Stoustrup et al., [76] showed that in the presence of plant uncertainty the controller and FDI filter design are coupled, hence the design of the filter depends on the controller. An

optimal design cannot be obtained without considering the controller and FDI filter design problem simultaneously. This approach is known as the *integrated FDI filter design problem*. Uncertainty in the system model is inevitable in real-world control engineering problems including aircraft flight control. It is unknown how close to *optimal* the FDI filter can achieve when it is designed separately from the flight control system. Hence, it is imperative to better understand the interaction between the closed-loop system and the FDI filter in the presence of uncertainty.

Irrespective of the technique used to design an FDI filter, the basic requirements for FDI filters are typically the following:

- Capability to isolate faults that occur simultaneously.
- Sensitivity to a particular fault and insensitivity to other faults.
- Robustness to modeling uncertainty.
- Good attenuation of external disturbances and noise at the FDI filter output.
- Asymptotic convergence of the FDI filter output to a non-zero steady-state in response to a non-zero fault, zero output otherwise.

One popular FDI filter design technique is based on geometric techniques originally proposed by Massoumnia and Willsky, [54, 57], for the fault detection case with no disturbance or model uncertainty. As the name suggests, the geometric filter design technique exploits geometric properties of the system state-space. If the faults in the system have non-parallel signature directions, it is possible to design operators and output mixing maps that project these faults into disjoint, mutually orthogonal subspaces. The filter spectrum is designed to be stable, hence its output is non-zero only for non-zero faults. Similarly, the spectrum can be tuned to obtain the desired transient filter response. The geometric filter design paradigm was further extended and generalized for uncertain and time varying systems, [43, 25].

$\mathcal{H}_2$ and $\mathcal{H}_\infty$ model matching approaches to FDI filter design are also popular (see Varga, [92, 93], Marcos, [56], Ding, [23], Zolghadry and Henry, [8], Jaimoukha, [39].) A majority of the FDI techniques discussed make use of open-loop systems in their design, although there are frequency approaches that are applicable to the design of FDI filters for closed-loop situations (see Niemann et al., [59], Stoustrop et al., [76], Marcos et al., [55]).

The basic requirements for these analytical FDI filters are often in conflict and a particular design approach can typically satisfy only a subset of the requirements. The geometric design approach, for example, is known for its excellent fault isolation, fault reconstruction and sensitivity properties under small modeling uncertainty and noise, but it is assumed to be sensitive as the model uncertainty and noise increase. Hence development of a metric which provides a measure of interaction between the FDI filter and controller is important.

In this thesis closed-loop test metrics are derived for FDI filter performance analysis in the presence of model uncertainty. The role the closed-loop controller plays in the FDI problem is investigated as is the role of robustness in the controller design. The objective of the thesis is to develop metrics for fault detection to aid the designer.

The approach used in the thesis is to formulate the FDI filter and closed-loop controller synthesis problem as an $\mathcal{H}_\infty$ model matching problem. A similar problem exists in the literature and is based on the simultaneous design of a controller and a fault estimation filter. It is called the *four-parameter filter design problem*, [38, 76, 59]. The input-output properties of the FDI filter and controller are derived from the four-parameter filter design problem in the form of transfer matrices. The key component corresponding to the influence of the closed-loop controller on filter performance is the output sensitivity function (S). The output sensitivity function plays a key role in the FDI filter-controller performance metric. This fact results in design trade-offs between the filter and the controller. It is shown how the transfer function matrices obtained for the filter-controller synthesis problem can be reformulated as frequency domain test metrics to assess the performance of an FDI filter in the presence of an uncertain closed-loop system.

Separate metrics are obtained for actuator and sensor faults, respectively. These metrics can be used separately from the filter synthesis problem, to assess the worst-case performance of an FDI filter and controller for a given set of plant models.

The proposed FDI performance metrics are applied to a flight control example associated with the NASA GTM, [19] longitudinal axis dynamics. To demonstrate the versatility of the metrics, FDI filters are synthesized using two approaches, the $\mathcal{H}_\infty$ model matching and the geometric design technique. The $\mathcal{H}_\infty$ model matching design, [55, 56, 70, 92, 93, 23, 8, 39], is a closed-loop design problem in which knowledge of the tracking controller is exploited in filter synthesis. The geometric filter design technique, [54, 57, 12], is an open-loop filter design technique that exploits the invariant subspaces of the state-space. It is shown how the proposed metrics can be applied in comparing the fault detection performance

of different filters under identical uncertain plant dynamics and tracking controller. The effect different controllers have on the performance of an FDI filter and the effect dynamic uncertainty has on filter performance is investigated. The NASA GTM longitudinal axis application provides interesting insights about the role of a closed-loop controller and the output sensitivity function in the fault detection problem.

6.1 Problem statement and analysis

Input-output properties of the FDI filter and closed-loop system are derived in this section. The plant model is assumed to be a linear time-invariant (LTI) system and described by a state-space model,

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{6.1a}$$

$$y(t) = Cx(t) + Du(t) \tag{6.1b}$$

where, $x \in \mathbb{R}^n$, $u \in \mathbb{R}^p$, $y \in \mathbb{R}^m$ are the states, controls and measurements, respectively, and A, B, C and D are real matrices of the corresponding dimension. Let $G(s) = C(sI - A)^{-1}B + D$ be the transfer function representation of the linear plant model.

Model uncertainty is captured through a general class of uncertain plants, $G_\Delta(s)$, where $\Delta(s) := \{\mathbf{\Delta} : \Delta \in \mathbb{C}_{m\times m}, \bar{\sigma}(\Delta) \leq 1\}$, is a complex norm bounded uncertainty with arbitrary phase. This general description can include multiplicative, additive, linear fractional transformation (LFT) or other types of uncertain plant models, [24].

The fault estimation and tracking control problem is formulated as shown in Fig. 6.1(a). Sensor faults, f_s and actuator faults, f_a are assumed to enter in an additive manner. Weights, $W_a(s)$ and $W_s(s)$, describe the frequency content of actuator and sensor faults respectively and are in general block diagonal matrices for multi-input/multi-output (MIMO) systems. $W_a(s)$ and $W_s(s)$ are assumed to be known functions derived from fault modeling, and their development is not discussed in this thesis. See [49, 9] for information regarding the development of fault weighting functions. Weighting functions $W_{p_f}(s)$ and $W_{p_c}(s)$ are design parameters used to shape the desired controller and FDI filter performance objectives.

A standard model matching $\mathcal{H}_\infty$ synthesis technique is used to design a filter and controller. The design objective is to minimize the gain of the transfer function matrix from faults ($f = \begin{bmatrix} f_a & f_s \end{bmatrix}^\intercal$) and tracking reference (y_c) to weighted fault estimation error ($\tilde{e}_f$) and weighted reference tracking error ($\tilde{e}_c$), respectively, measured in terms of the $\mathcal{H}_\infty$ norm of

the closed-loop system. This synthesis problem is convex and well developed in the $\mathcal{H}_\infty$ control theory for which many standard algorithms are available in the literature, [44, 28].

The $\mathcal{H}_\infty$ problem formulation feeds the measurements ($m \times 1$) and tracking reference vector ($q \times 1$) separately to the controller-filter $K(s)$ which is to be designed. The outputs of $K(s)$ are the control inputs, u_c, and the fault estimation vector, $u_f = \begin{bmatrix} \hat{f}_a & \hat{f}_s \end{bmatrix}^\mathsf{T}$. $\hat{f}_a$ and $\hat{f}_s$ are estimates of actuator and sensor faults, respectively. The input-output topology allows $K(s)$ to be partitioned into four components as shown in Eq. 6.2. This is known as the *four-parameter controller-filter problem* in the literature, [38, 59, 76]. The problem formulation is motivated by the original setup described by Nett et al., [38] and Stoustrup et al., [76] with the difference that a tracking controller is considered instead of disturbance rejection and the problem is not tied down to the simultaneous FDI filter and controller synthesis. Note that the fault model affects the achievable performance of the controller, $f \rightarrow \tilde{e}_c$. In general, one would design the controller independent of the fault and investigate the fault effect a posteriori.

The controller-filter partitions are,

$$K(s) = \left[\begin{array}{c|c} K_{c_1}(s) & K_{c_2}(s) \\ \hline K_{f_1}(s) & K_{f_2}(s) \end{array}\right] \tag{6.2}$$

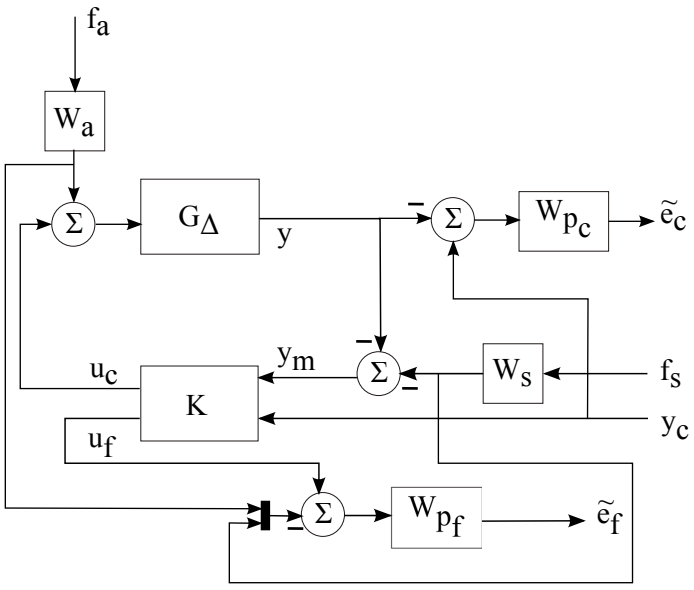
And the input-output signals for $K(s)$ are,

$$\begin{pmatrix} u_c \\ u_f \end{pmatrix} = K(s) \begin{pmatrix} y_m \\ y_c \end{pmatrix}$$

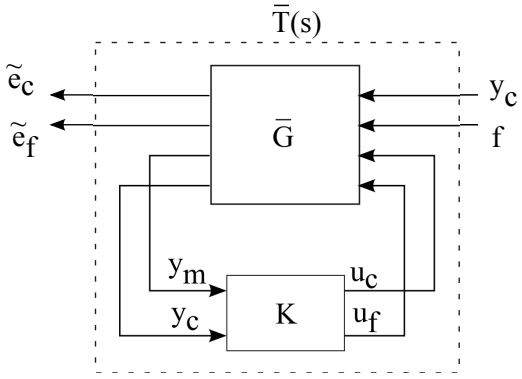
The elements $K_{c_1}(s)$ and $K_{f_1}(s)$ are the feedback components of the tracking controller and the filter, respectively, which act on the measurements, y_m. The elements $K_{c_2}(s)$ and $K_{f_2}(s)$ are the feed-forward elements, respectively.

Consider the open-loop equations for the interconnection shown in Fig. 6.1(a),

$$\begin{aligned} \tilde{e}_c &= W_{p_c}(s)[y_c - G_\Delta(s)(u_c + W_a(s)f_a) - W_s(s)f_s] && \text{(6.3a)} \\ \tilde{e}_f &= W_{p_f}(s)\left(u_f - \begin{bmatrix} W_a(s)f_a \\ W_s(s)f_s \end{bmatrix}\right) && \text{(6.3b)} \\ y_m &= -(G_\Delta(s)[u_c + W_a(s)f_a] + W_s(s)f_s) && \text{(6.3c)} \end{aligned}$$



(a) System interconnection



(b) The LFT structure

Figure 6.1: Fault estimation filter and tracking control problem

The system of equations in Eq. 6.3 (a)-(c) describing the open-loop input-output signals can be rewritten in a matrix form as,

$$\begin{bmatrix} \tilde{e}_c \\ \tilde{e}_f \\ y_m \\ y_c \end{bmatrix} = \underbrace{\left[\begin{array}{ccc|cc} W_{p_c} & -W_{p_c}\begin{bmatrix} G_\Delta W_a & W_s \end{bmatrix} & & -W_{p_c}G_\Delta & 0 \\ 0 & -W_{p_f}\begin{bmatrix} W_a & 0 \\ 0 & W_s \end{bmatrix} & & 0 & W_{p_f} \\ \hline 0 & -\begin{bmatrix} G_\Delta W_a & W_s \end{bmatrix} & & -G_\Delta & 0 \\ I & 0 & & 0 & 0 \end{array}\right]}_{\bar{G}(s)} \begin{bmatrix} y_c \\ f \\ u_c \\ u_f \end{bmatrix} \tag{6.4}$$

where, $f = \begin{bmatrix} f_a & f_s \end{bmatrix}^\intercal$, is the augmented fault vector. Introducing the controller-filter $K(s)$ to form the lower LFT structure as shown in Fig. 6.1(b) and simplifying system equations, the closed-loop transfer matrix, $\bar{T}(s)$, can be obtained,

$$\begin{bmatrix} \tilde{e}_c \\ \tilde{e}_f \end{bmatrix} = \bar{T}(s) \begin{bmatrix} y_c \\ f \end{bmatrix} \tag{6.5}$$

where $\bar{T}(s)$ is block partitioned as,

$$\bar{T}(s) = \begin{bmatrix} \bar{T}_{11} & \bar{T}_{12} \\ \bar{T}_{21} & \bar{T}_{22} \end{bmatrix} \tag{6.6}$$

and the elements of matrix $\bar{T}(s)$ are given as follows,

$$\bar{T}_{11} = W_{p_c}[I - S_\Delta G_\Delta K_{c_2}] \tag{6.7a}$$

$$\bar{T}_{12} = -W_{p_c}(I - T_\Delta)\begin{bmatrix} G_\Delta W_a & W_s \end{bmatrix} \tag{6.7b}$$

$$\bar{T}_{21} = -W_{p_f}(K_{f_1} S_\Delta G_\Delta K_{c_2} - K_{f_2}) \tag{6.7c}$$

$$\bar{T}_{22} = -W_{p_f}\left(K_{f_1} S_\Delta \begin{bmatrix} G_\Delta W_a & W_s \end{bmatrix} - \begin{pmatrix} W_a & 0 \\ 0 & W_s \end{pmatrix}\right) \tag{6.7d}$$

where,

$$T_\Delta = G_\Delta K_{c_1}(I + G_\Delta K_{c_1})^{-1} \qquad S_\Delta = (I + G_\Delta K_{c_1})^{-1} \tag{6.8}$$

are the output complementary sensitivity and the output sensitivity function, respectively.

The $\mathcal{H}_\infty$ controller and filter synthesis is carried out by minimizing the maximum singular value of the transfer function matrix $\bar{T}(s)$ over the set of permissible controllers/filters. The

synthesis of the filter and the controller can be carried out simultaneously or independently.

6.2 Metrics for FDI filter and controller interaction

The transfer function matrices obtained in Eq. 6.7 (a)-(d) offer valuable insights into the closed-loop behavior of FDI filters [70]. These transfer matrices can also be used to define performance metrics for FDI filters. Consider Eq. 6.7(d), based on problem setup shown in Fig. 6.1(a) and Eq. 6.5. The filter parameter $K_{f_1}(s)$ can be further block partitioned as,

$$K_{f_1}(s) = \begin{bmatrix} K^a_{f_1}(s) \\ K^s_{f_1}(s) \end{bmatrix} \tag{6.9}$$

Eq. 6.7(d) is the transfer function matrix from faults (f) to weighted fault estimation error ($\tilde{e}_f$),

$$\begin{bmatrix} \tilde{e}_{f_a} \\ \tilde{e}_{f_s} \end{bmatrix} = \bar{T}_{22}(s) \begin{bmatrix} f_a \\ f_s \end{bmatrix}$$

Using Eq. 6.7(d) and Eq. 6.9,

$$\bar{T}_{22}(s) = W_{p_f} \begin{bmatrix} K^a_{f_1} S_\Delta G_\Delta - I & K^a_{f_1} S_\Delta \\ K^s_{f_1} S_\Delta G_\Delta & K^s_{f_1} S_\Delta - I \end{bmatrix} \begin{bmatrix} W_a & 0 \\ 0 & W_s \end{bmatrix} \tag{6.10}$$

or independent of the weighting functions,

$$\begin{pmatrix} e_{f_a} \\ e_{f_s} \end{pmatrix} = \begin{bmatrix} K^a_{f_1} S_\Delta G_\Delta - I & K^a_{f_1} S_\Delta \\ K^s_{f_1} S_\Delta G_\Delta & K^s_{f_1} S_\Delta - I \end{bmatrix} \begin{pmatrix} f_a \\ f_s \end{pmatrix} \tag{6.11}$$

where, S_Δ is given by Eq. 6.8.

Thus, given the closed-loop controller, $K_c(s)$, the uncertain plant model, $G_\Delta(s)$ and the FDI filter, $K_f(s)$, the transfer matrix in Eq. 6.11 is completely known. A closer look at each term in this transfer matrix reveals

- The diagonal terms in Eq. 6.11 are the fault signals to fault estimation error transfer functions. The transfer function $(K^a_{f_1} S_\Delta G_\Delta - I)$ maps the actuator fault to the actuator fault estimation error. Similarly, $(K^s_{f_1} S_\Delta - I)$ maps the sensor fault to the senor

fault estimation error. For an ideal FDI filter, it is expected that the gain of these two transfer functions is zero, implying that the FDI filter can perfectly estimate fault signals across frequency.

- The off-diagonal terms in Eq. 6.11 represent fault cross-coupling, that is, an actuator fault showing up in sensor fault estimate signal/residual, and vice-versa. Ideally, the two transfer functions, $K^a_{f_1} S_\Delta$ and $K^s_{f_1} S_\Delta G_\Delta$ should be as close to zero as possible, a property referred to in the literature as *fault-isolation*.

Since these transfer functions map fault signals (f) to fault estimation error, ideally one would like the gain of these transfer functions to be zero across frequency, implying perfect fault estimation. From an analysis perspective, evaluating the worst-case gain of each of the four blocks of the transfer function matrix in Eq. 6.11 across frequency, one can find worst case fault estimation error of the FDI filter in the presence of $\Delta(s)$ uncertainty. These form the basis for the metrics to assess FDI filter performance. Packard et al., [61] describe a method to compute the worst case gain of any uncertain transfer function. The worst-case gain function, *wcgain*, in the MATLAB *Robust Control Toolbox*, [29] exploits this method and is used in the examples presented in later sections to compute the metrics. The actuator fault estimation metric can be computed as,

$$\max_{\Delta \in \mathbf{\Delta}} \quad \| K^a_{f_1} S_\Delta G_\Delta - I \|_\infty$$

and the corresponding sensor fault estimation metric as,

$$\max_{\Delta \in \mathbf{\Delta}} \quad \| K^s_{f_1} S_\Delta - I \|_\infty \tag{6.12}$$

$$\begin{pmatrix} e_{f_a} \\ e_{f_s} \end{pmatrix} = \begin{bmatrix} K^a_{f_1} S_\Delta G_\Delta K_{c_2} - K_{f_2} \\ K^s_{f_1} S_\Delta G_\Delta K_{c_2} - K_{f_2} \end{bmatrix} y_c \tag{6.13}$$

Similarly, from Eqs. 6.9 and 6.7(c), the transfer matrix from tracking reference to fault estimates (Eq. 6.13) is obtained. This is another cross-coupling term which represents misinterpretation of tracking reference signal as a sensor or actuator fault by the FDI filter. Clearly, good FDI filter performance requires that only actual faults in the system should be registered in the fault estimates of the FDI filter. Hence, the transfer functions

Table 6.1: FDI filter worst case performance metrics

Qualifier	Metric
Actuator fault metric	$\max_{\Delta\in\mathbf{\Delta}} \quad \|K^a_{f_1} S_\Delta G_\Delta - I\|_\infty$
Sensor fault metric	$\max_{\Delta\in\mathbf{\Delta}} \quad \|K^s_{f_1} S_\Delta - I\|_\infty$
Sensor-to-actuator fault cross-coupling	$\max_{\Delta\in\mathbf{\Delta}} \quad \|K^a_{f_1} S_\Delta\|_\infty$
Actuator-to-sensor fault cross-coupling	$\max_{\Delta\in\mathbf{\Delta}} \quad \|K^s_{f_1} S_\Delta G_\Delta\|_\infty$
Tracking reference-to-actuator fault cross-coupling	$\max_{\Delta\in\mathbf{\Delta}} \quad \|K^a_{f_1} S_\Delta G_\Delta K_{c_2} - K_{f_2}\|_\infty$
Tracking reference-to-sensor fault cross-coupling	$\max_{\Delta\in\mathbf{\Delta}} \quad \|K^s_{f_1} S_\Delta G_\Delta K_{c_2} - K_{f_2}\|_\infty$

$(K^a_{f_1} S_\Delta G_\Delta K_{c_2} - K_{f_2})$ and $(K^s_{f_1} S_\Delta G_\Delta K_{c_2} - K_{f_2})$ should also be as close to zero as possible. The FDI filter performance metrics derived in this section are summarized in Table. I. For good fault estimation performance of the FDI filter, the $\|.\|_\infty$ of all the metrics given in Table. I should be close zero.

6.3 Alternative approaches to the FDI filter problem: geometric filter design

The test metrics elaborated in the previous section can be used to evaluate various fault detection filters. These FDI filters can be designed applying e.g. parity space approach or observer based FDI filters designed by using geometric concepts.

A general form of open-loop detection problem is shown in Fig.6.2. The corresponding state equations can be written as,

$$\dot{x}(t) = Ax(t) + Bu(t) + L_a f_a(t) \tag{6.14a}$$

$$y(t) = Cx(t) + L_s f_s(t) \tag{6.14b}$$

The linear system described in Eqs. 6.14(a),(b) is similar to the system in Eqs. 6.1, but with explicit notation of fault and disturbance directions in the system state-space. In the closed-loop application of the FDI filters, noise and uncertainty properties will be determined by the controller via the sensitivity (S) and the complementary sensitivity functions (T). The geometric filter design concept is used to design filters for the open-loop system and applied to the closed-loop. The filters are be evaluated and compared with the $\mathcal{H}_\infty$ filter design

using the metrics given in Table. I.

The derivation of the FDI filters using geometric arguments is presented for LTI systems with no disturbance, no uncertainty and the detection and isolation of two faults. Given the state equations, the faults are modeled additively in the state dynamics (typical actuator faults) as:

$$\dot{x}(t) = Ax(t) + Bu(t) + L_1 f_1(t) + L_2 f_2(t) \tag{6.15a}$$
$$y(t) = Cx(t) \tag{6.15b}$$

where L_1, L_2 represent the faults directions in the state space and f_1, f_2 are the fault signals that are zero if there is no fault but nonzero if that particular fault occurs.

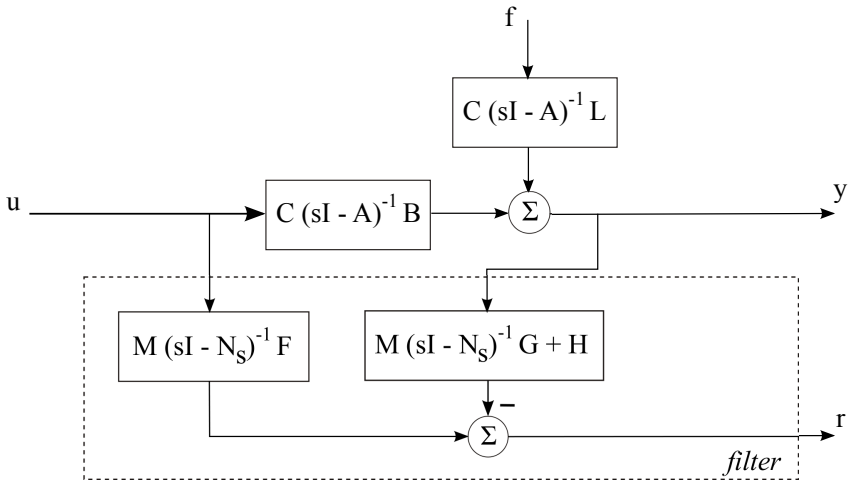


Figure 6.2: Geometric FDI filter block diagram

The problem of FDI filter design considered is the following. Synthesize a residual generator (filter) such that its output r_i is sensitive to f_i but insensitive to any $f_j, i \neq j$. More precisely if $f_i = 0$ then $\lim_{t\to\infty} r_i = 0$ and if $f_i \neq 0$ then $r_i \neq 0$.

If the residual generator has an observer (filter) structure, then it can be deduced that the fault effect modeled by L_2 will show up in the $\mathcal{W}_2 = \langle A+GC, L_2 \rangle$ invariant subspace of the state - error space. The goal is to design a filter gain G such that $\mathcal{W}_2$ is in the unobservability subspace of the filter designed to detect the failure showing up in the direction L_1.

Denote by $\mathcal{S}$ the unobservability subspace of a filter if there exists an output mixing map

H and filter gain G such that $\mathcal{S} = \langle KerHC|A + GC\rangle$. It is known that the family of controllability subspaces contained in a given subspace has a maximal element while the family of unobservability subspaces containing a given subspace has a minimal element. Denote these by $\mathcal{R}^*$ and $\mathcal{S}^*$, respectively.

Using the invariant subspaces defined above one can obtain the result cited frequently as the "fundamental problem of residual generation", see e.g. Massoumnia for LTI, [54],[57], and the extension for LPV systems in Bokor and Balas, [12].

If the smallest unobservability subspace $\mathcal{S}^*$ containing $\mathcal{L}_2$ satisfies $\mathcal{S}^* \bigcap \mathcal{L}_1 = 0$, then it is possible to obtain a (not necessarily stable) filter in the form

$$\dot{\xi}(t) = N\xi(t) - Gy(t) + Fu(t) \tag{6.16a}$$

$$r(t) = M\xi(t) - Hy(t) \tag{6.16b}$$

where G is such that $(A + GC)\mathcal{S}^* \subseteq \mathcal{S}^*$ holds, $\xi(t) = Px(t)$ where P is the projection $\mathcal{X} \to \mathcal{X}/\mathcal{S}^*$, $N = P(A + GC)P^T$, H is a solution to $MP = HC$ and $F = PB$.

To obtain quadratically stable filters one can set $N_s = N + G_sM$, where G_s is determined from the linear matrix inequality (LMI):

$$N^TX + XN + M^TK^T + KM < 0, X = X^T > 0, K = XG_s. \tag{6.17}$$

Using this approach there are as many filters as faults to detect, and their state dimensions are equal to the dimension of $\mathcal{X}/\mathcal{S}^*$. The filter poles can be tuned by imposing constraints in the LMI resulting in perfect reconstruction of fault signals f_i. Sensor faults can be converted to actuator fault problems by defining pseudo actuator directions.

The input-output form of the FDI filter is given by

$$r(s) = K_y y(s) + K_u u(s)$$

where

$$K_u(s) = M(sI - N_s)^{-1}F, \quad K_y(s) = -[M(sI - N_s)^{-1}G + H]$$

It is noteworthy that the performance metrics given in Table. I. are closed-loop metrics, meaning, the formulation assumes that the filter is implemented in closed-loop. As men-

tioned in the introduction section, most FDI filters, including the geometric filter discussed in this section, are designed for the open-loop system. So in order to apply these test metrics, its necessary to transform the open-loop filter equations into closed-loop. Inputs to the closed-loop filter are $\begin{bmatrix} y_m & y_c \end{bmatrix}^T$, as shown in Fig.6.1(a), whereas inputs to the open-loop filter are $\begin{bmatrix} y_m & u_c \end{bmatrix}^T$. Take for example, open-loop sensor fault estimation filter,

$$r_{OL}(s) = K_y(s)Y_m(s) + K_u(s)U_c(s) \tag{6.18}$$

Given, the closed-loop controller, $U_c(s) = K_{c_1}(s).Y_m(s) + K_{c_2}(s).Y_c(s)$, plugging in for $U_c(s)$ in Eq.6.18, leads to,

$$r_{CL}(s) = [K_y(s) - K_u(s)K_{c_1}]Y_m(s) + K_u(s)K_{c_2}(s)Y_c(s) \tag{6.19}$$

The filter equations in closed-loop form will be analyzed using the proposed detection filter metrics.

6.4 Application of fault detection metrics

Application of the metrics proposed in Sec. 6.2 to a flight control example problem is presented in this section. The NASA Generic Transport Model (GTM) plant model is considered. The following applications of the detection filter metrics are discussed:

- Comparison of the worst case performance of a set of FDI filters - Two FDI filters are designed, one based on the geometric technique, [54, 12] and one $\mathcal{H}_\infty$ FDI filter (described in Sec. 6.1) to provide a distinct pair of test-filters. An identical pitch command tracking controller, $K_c(s)$, is used with both the test filters as a benchmark.
- Effect of the controller on the performance of an FDI filter to detect faults - In this example, two $\mathcal{H}_\infty$ pitch tracking controllers are synthesized with different performance objectives. An FDI filter designed in the previous example is selected to investigate the influence of different controllers on fault detection performance.

6.4.1 NASA Generic Transport Model (GTM) longitudinal dynamics

The NASA Generic Transport Model (GTM) is a 5.5% scale model replica of a typical twin turbofan powered commercial airliner, which was designed as a platform for flight safety

related research and experiments, [45]. A high fidelity nonlinear model of the airplane provided by NASA, [19], is used. The GTM nonlinear model is trimmed a in straight-and-level flight condition at True Airspeed (TAS) = 90 knots using the available longitudinal controls, namely, the elevator deflection (δ_e) and the throttle position (δ_{th}) of the engines. A linearized longitudinal model is obtained at this equilibrium. The linear model has four states, namely, true airspeed (TAS, knots), angle of attack (α, rad), pitch rate (q, rad/s) and pitch attitude (θ, rad). Only one control input, the elevator deflection (δ_e, rad) is considered, the throttle (δ_{th}) is held fixed at the trim value. The Elevator is an aerodynamic control effector whose deflection generates pitching moment about the center of gravity of an aircraft, thereby modulating aircraft orientation in the pitching (longitudinal) plane.

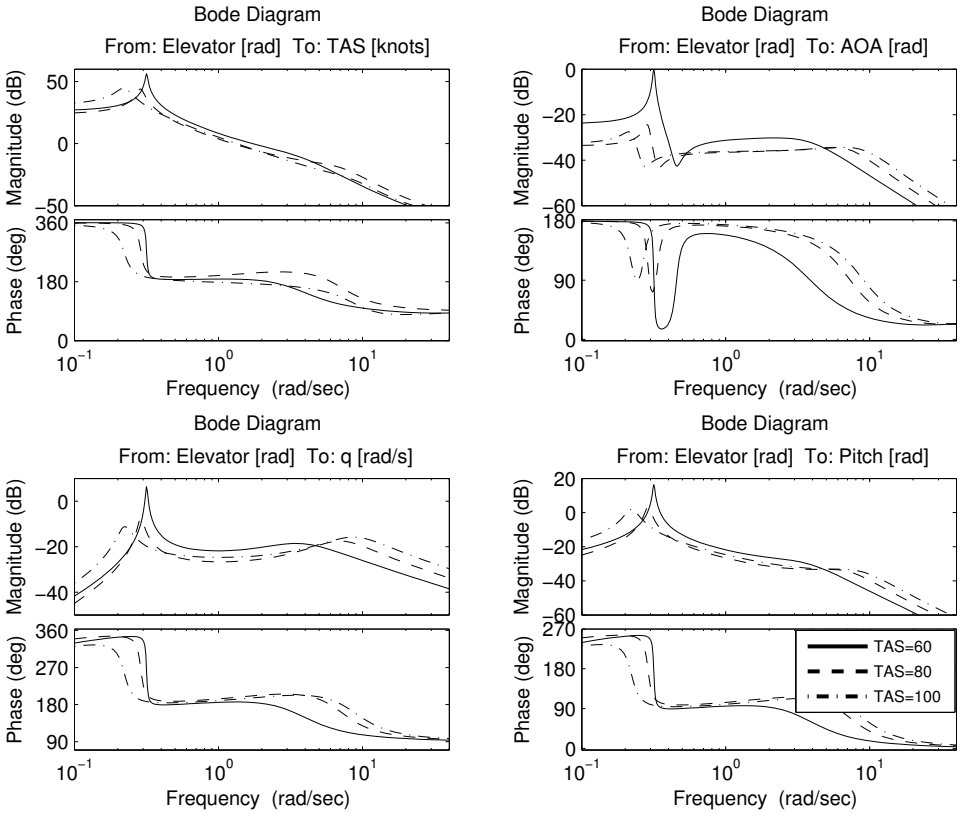


Figure 6.3: GTM longitudinal: Plant uncertainty range

Uncertainty Modeling:

The GTM is capable of trimmed flight at various airspeeds and altitudes, which defines the flight envelope of the aircraft. Linear models at each operating point can have varying dynamic characteristics which must be taken into account to guarantee robust performance. Fig. 6.3 shows frequency responses of the GTM aircraft to elevator input at different cruise

flight conditions at 60, 80 and 100 knots. An input multiplicative uncertainty model is proposed to over-bound this set of plant models. A model of the form $G_\Delta(s) = G(s)(I + \Delta(s)W_u(s))$, $\bar{\sigma}(\Delta(s)) \leq 1$ captures the entire straight and level flight envelope of the GTM. $W_u(s)$ is the uncertainty weighting function (see Appendix-B.1). $G_\Delta(s)$ is used to design a single $\mathcal{H}_\infty$ filter or controller that is optimized to perform well at all flight conditions. As described previously in Section 6.1, the $\mathcal{H}_\infty$ synthesis approach finds a filter/controller that *minimizes* the worst case performance across all plant models in the uncertain set.

6.4.2 $\mathcal{H}_\infty$ reference tracking controller: Pitch attitude

A pitch attitude command/reference tracking controller is designed using $\mathcal{H}_\infty$ model-matching techniques. The GTM linear plant used in the synthesis is presented in Appendix-B.1. A multiplicative uncertain plant model, $G_\Delta(s) = G(s)(I + \Delta(s)W_u(s))$, is used to describe the set of possible aircraft models being considered (see Appendix-B.1). The GTM plant outputs used for feedback are: pitch attitude (θ, rad), pitch-rate (q, rad/s), airspeed (TAS, knots) and angle-of-attack (α, rad). The control input is elevator deflection (δ_e, rad). The design performance and robustness objectives are - a rise time less than 4 seconds, steady-state error less than 5 percent, gain margin greater than 6 dB, phase margin greater than or equal to 45 degrees and peak-overshoot less than 10 percent.

The controller design problem interconnection is shown in Fig. 6.4. An $\mathcal{H}_\infty$ optimal controller $K_c(s)$ is obtained with five inputs (four measurements and the pitch reference), one output (elevator command) and 11 controller states. Fig. 6.5 shows the nominal response of the linear closed-loop system to a step pitch-command input. Note, the Δ prefix on the state-names in Fig. 6.5 indicates that only the perturbation around equilibrium is plotted, not to be confused with uncertainty descriptor $\Delta(s)$. It can be seen from Fig. 6.5 that the design specifications are achieved. On the nominal system, rise-time is less than 1 second, overshoot is less than 1 percent and the steady-state error is less than 0.2 percent.

6.4.3 Sensor and actuator fault description

Two faults are selected for the synthesis of FDI filter, one sensor and one actuator fault. The pitch-rate gyro is chosen as a faulted sensor. The GTM longitudinal model has only one input, the elevator, which is selected as the actuator fault channel.

In general, multiple failure modes exist for the pitch-rate gyro and the elevator, for example,

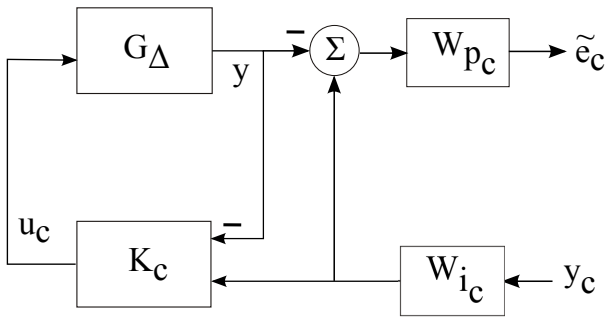


Figure 6.4: $\mathcal{H}_\infty$ controller synthesis: system interconnection

a fault in the pitch-rate gyro can manifest itself as a null-shift, zero-output (dead-sensor) or scale-factor error. Similarly, the elevator fault can manifest itself as a locked-elevator, a hard-over failure, floating elevator, etc. Two failure modes are considered in this work, the null-shift and the floating failure mode. The null-shift is classified as a change in zero position of the sensor/actuator. The floating failure implies that the output (sensor output or actuator position) floats (oscillates) between two arbitrary values.

6.4.4 The $\mathcal{H}_\infty$ fault estimation filter design

A filter is synthesized that minimizes the $\mathcal{H}_\infty$ norm from faults vector, f, to weighted fault estimation error vector, $\tilde{e}_f$, based on the problem interconnection shown in Fig. 6.1(a). The controller synthesized in Sec. 6.4.2 is used in the design. The resulting $\mathcal{H}_\infty$ filter is presented in Appendix-B.3. The nominal time-response of the linear closed-loop system with the FDI filter is shown in Fig. 6.6. An elevator fault of 1 deg is injected into the system at t=15s, a pitch rate gyro null-shift of -2 deg/s is injected at t=30s, and a pitch angle command reference of 10 deg is applied at t=20. The effect of these faults are shown in Fig. 6.6. It can be seen from Fig. 6.6 that the fault estimates have fast detection times, and there is minimal coupling between sensor and actuator fault estimates. Similarly, the fault estimates match the fault signals closely in the steady-state.

6.4.5 Application of the test metrics to filter performance assessment

6.4.5.1 Comparison of two FDI filters

In this example, two FDI filters are compared with respect to each other using the proposed filter metrics given in Table. I for sensor and actuator faults. The GTM longitudinal model with multiplicative input uncertainty and the $\mathcal{H}_\infty$ pitch controller described in Sec. 6.4.2

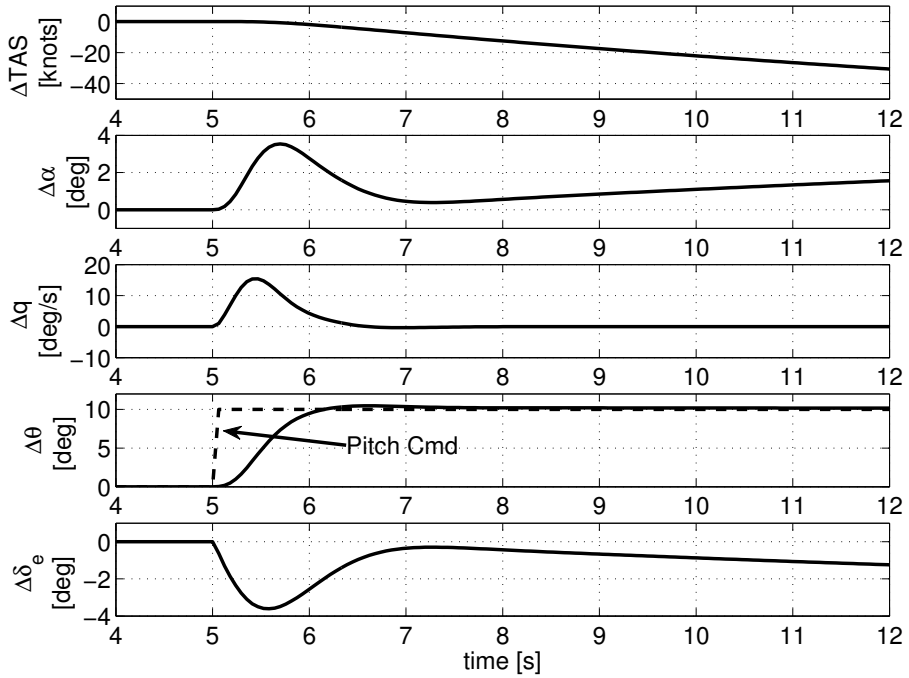


Figure 6.5: $\mathcal{H}_\infty$ controller: nominal closed-loop response

comprise the set of closed-loop systems. The two FDI filters considered are:

1. Filter 1: Geometric filter design, [54, 12] with five inputs, namely, the four measured states, airspeed, angle-of-attack, pitch-rate and pitch angle, and one plant input, the elevator deflection. The filter is designed to have stable residual dynamics, i.e., filter output (residual) asymptotically converges to zero when no fault is present (filter design data given in Appendix-B.2).

2. Filter 2: $\mathcal{H}_\infty$ fault estimation filter design presented in Sec. 6.4.4 with the five measurements as above.

The worst case actuator and sensor fault estimation metrics given in Table. I. are computed using *wcgain*. The function *wcgain* computes the worst-case gain of an uncertain transfer function (performance metrics in this case) at each frequency point as shown in Fig. 6.7 and also returns the corresponding $\Delta(s)$ uncertainty that results in the worst gain, at each frequency. The difference in the performance of each filter is evident from Fig. 6.7. The metrics predict the filters to have very good steady-state performance, indicated by a small value of the metric in the low frequency range. Also, at high frequency both

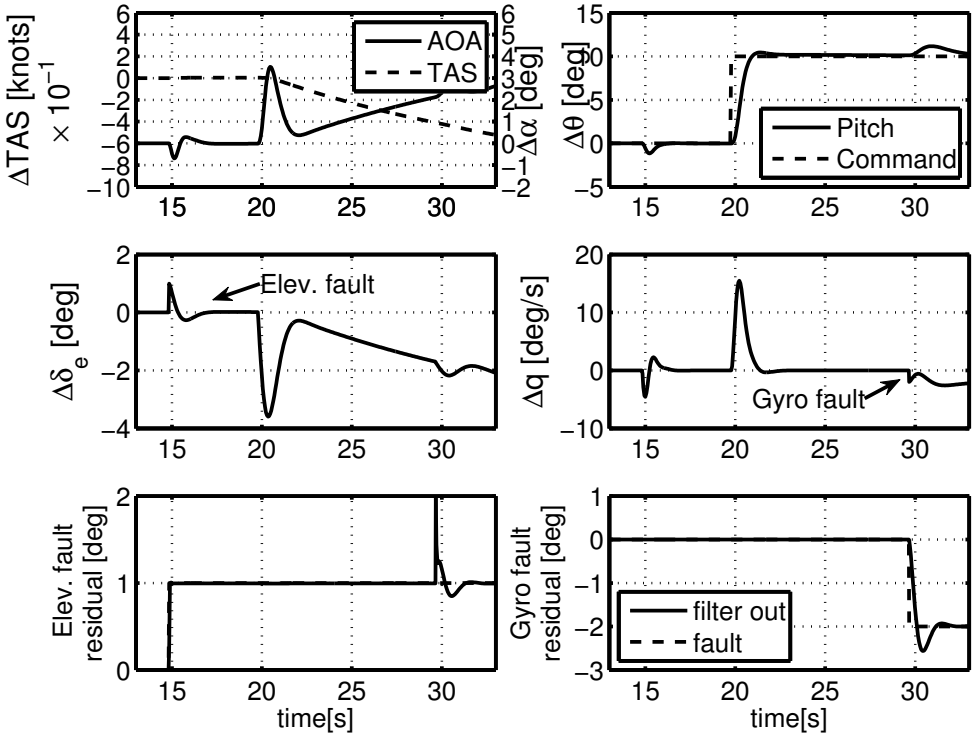
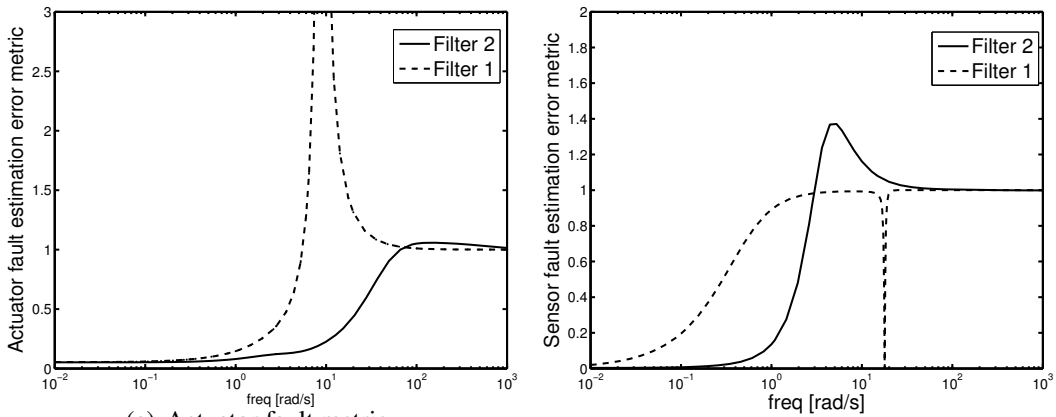


Figure 6.6: $\mathcal{H}_\infty$ fault estimation filter: Nominal response [linear closed loop]

sensor and actuator fault metrics settle at unity gain. As seen from the sensor and actuator fault metrics in Table. I, unity gain of the two metrics at high frequency is a consequence of $K^s_{f_1}(s)S_\Delta(s) \to 0$, $K^a_{f_1}(s)S_\Delta(s)G_\Delta(s) \to 0$, as $s \to \infty$, respectively. This is due to the filter *roll-off* which results in its inability to detect/estimate high frequency sensor and actuator faults.

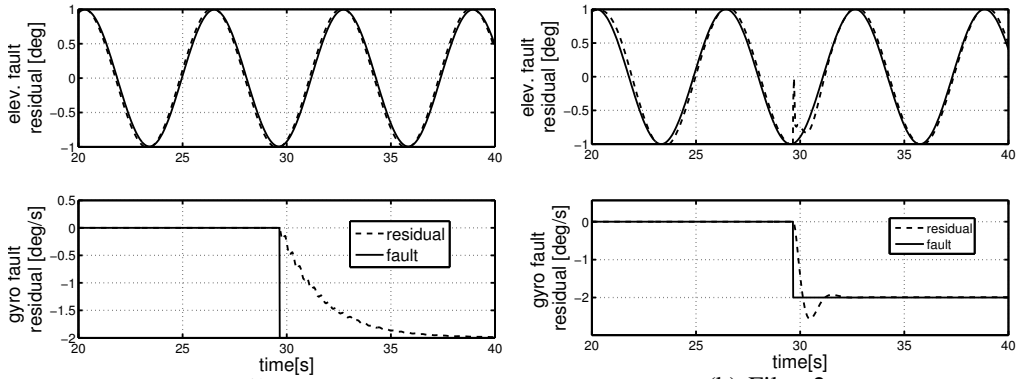
Consider the worst-case filter performance as predicted by the metrics in Fig. 6.7 (a),(b) at the frequency of 1 rad/s. The two filters are compared at that specific frequency point using the worst case linear time-domain simulation in Fig. 6.8 and 6.9. The time domain simulation is obtained by first substituting the worst-case $\Delta(s)$ uncertainty obtained from the worst-case gain computation of the two metrics, respectively, into the multiplicative uncertainty model of GTM longitudinal dynamics, $G_\Delta(s)$. Two test cases are investigated, one compares the sensor fault estimation performance in which 1 deg/s peak amplitude sinusoidal gyro fault is injected at 1 rad/s, Fig. 6.8. The other compares actuator fault estimation performance by injecting a sinusoidal elevator fault of 1 deg peak magnitude at 1 rad/s frequency as shown in Fig. 6.9.



(a) Actuator fault metric (b) Sensor fault metric

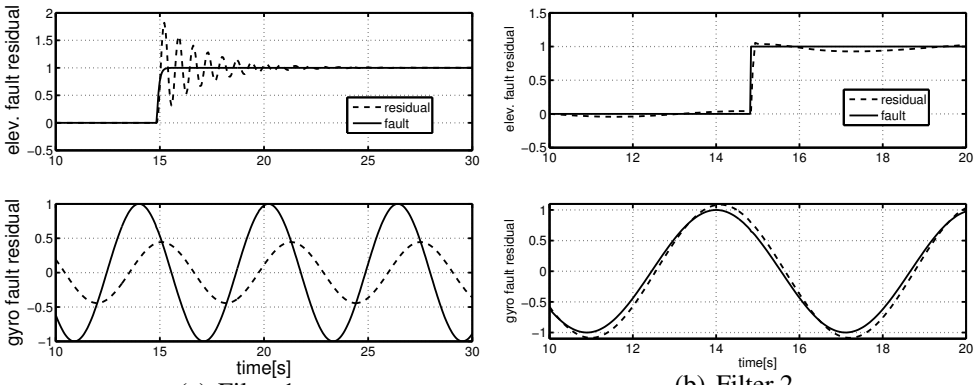
Figure 6.7: Comparison of fault estimation error metrics for two different FDI filters

The actuator fault metric at 1 rad/s for the two filters is close to zero in Fig. 6.7(a), indicating that both the filters should perform equally well for a 1 deg elevator fault. This is consistent with the time-domain simulation results seen in Fig. 6.8 where both the filters are able to track a 1 deg peak fault with very little error. Similarly, comparing the sensor fault metric at 1 rad/s for the two filters in Fig. 6.7(b) it may be noted that the value of the metric for the Filter 1 is much higher than that for Filter 2, hence Filter 2 is expected to perform better in this test. This assessment is also consistent with the worst-case time-domain simulation seen in Fig. 6.9, in which the worst-case performance for Filter 1 is seen to have much larger steady state error in gyro fault estimate compared to Filter 2.



(a) Filter 1 (b) Filter 2

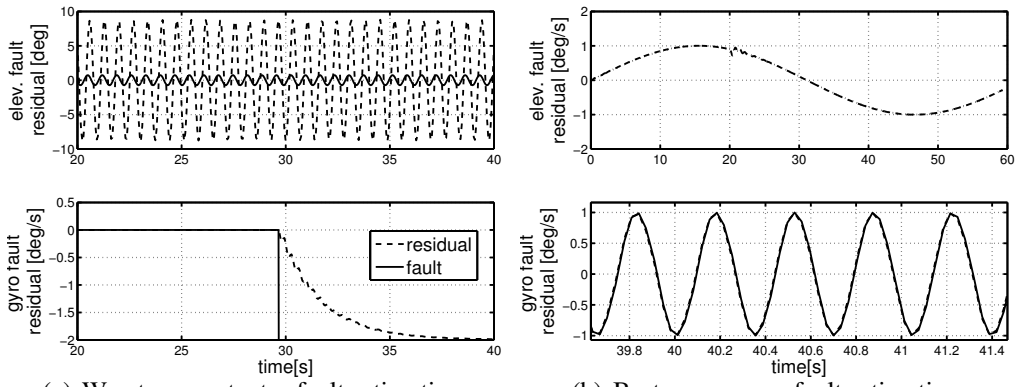
Figure 6.8: FDI Filter-comparison: Worst-case performance to 1 rad/s sinusoidal elevator fault



(a) Filter 1 (b) Filter 2

Figure 6.9: FDI Filter-comparison: Worst-case performance to 1 rad/s sinusoidal gyro fault

Another interesting feature to examine is the peculiar peak and notch seen in the actuator and sensor fault metrics for the geometric filter (Filter 1). This peak corresponds to a pair of lightly damped poles in the FDI filter at approximately 10 rad/s. The corresponding $\Delta(s)$ block obtained from the worst-case gain analysis is included in the linear simulation,with the actuator and sensor faults at 10 rad/s and 18 rad/s, respectively. it is interesting to see that the predicted behavior is clearly seen in the time-domain simulation. Thus, from the above example it is clear that the metrics are also able to identify some atypical characteristics of an FDI filter.



(a) Worst case actuator fault estimation (b) Best case sensor fault estimation

Figure 6.10: Worst-case linear simulation

6.4.5.2 Comparison of FDI performance for two different closed-loop controllers

Consider an example scenario in which an FDI filter and two controllers for the identical uncertain plant model are given. The objective is to compare the filter performance in the presence of different controllers. This example highlights the role played by a closed-loop controller in the FDI filter problem. The $\mathcal{H}_\infty$ filter and $\mathcal{H}_\infty$ controller synthesized in the previous sections are used in this example. For comparison, a second $\mathcal{H}_\infty$ controller is synthesized whose performance specifications are relaxed, i.e., rise time less than 10s, steady-state error less than 10 percent, while the rest of the design specifications are kept the same as for the controller synthesized in Sec. 6.4.2 (See the time-response for the two closed-loop systems in Fig. 6.13(a). Note, the Δ prefix on the state-names in the figure indicates that only the perturbation around equilibrium is plotted, not to be confused with uncertainty descriptor $\Delta(s)$).

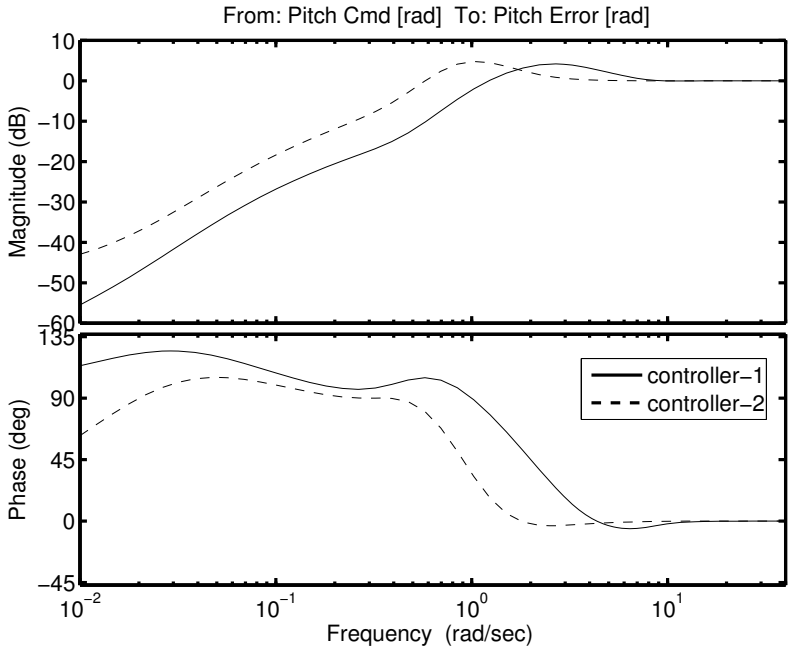


Figure 6.11: Comparison of the Sensitivity function of the two closed-loop systems

Fig. 6.11 shows the comparison of the sensitivity functions from pitch command to pitch attitude with the two $\mathcal{H}_\infty$ controllers. It can be seen that the second controller synthesized in this section has lower closed loop bandwidth (cross-over frequency) as well as greater steady-state tracking error (higher gain of the sensitivity function at low frequency).

The actuator and sensor fault estimation error metrics are computed for the two systems and shown in Fig. 6.12. The metrics indicate that the filter performance is nearly identical

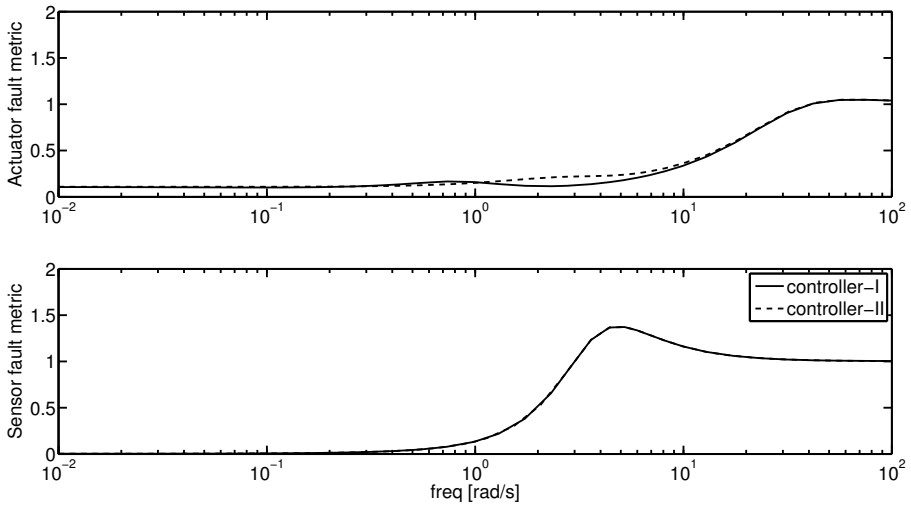


Figure 6.12: Sensor and Actuator fault estimation error metrics for a FDI filter with two different closed-loop controllers

under both tracking controllers' influence, despite the fact that their closed-loop tracking performance is quite different as seen in Fig. 6.13(a). To understand this behavior, consider the actuator fault and the actuator fault metric (from Table. I), the only term in the actuator fault metric that depends on the plant and the controller is $S_\Delta(s)G_\Delta(s)$. The Bode plot for a sample of the uncertain set of the transfer function, $S_\Delta(s)G_\Delta(s)$ for the two closed loop systems is shown in Fig. 6.14. It can be seen that although the transfer function itself is quite different for the two closed-loop systems, the *spread* of $S_\Delta(s)G_\Delta(s)$ magnitude and phase around the nominal model for both the Controller-I and Controller-II is nearly identical across frequency. Hence, both controllers are robust to the model uncertainty.

The controllers reduce the variation of the closed-loop system due to the uncertainty which results in the FDI filters being insensitive to the model error. If a controller was less robust, the increased variation in $S_\Delta(s)G_\Delta(s)$ would have resulted in poorer performance of the FDI filter. Hence, the FDI filter achieves almost identical performance as predicted by Fig. 6.12. Similarly, the sensor fault estimation performance of the FDI filter in question under the influence of Controller-I and Controller-II can also be explained by the same reasoning. The frequency domain results are validated in the time domain simulation shown in Fig. 6.13, where it can be seen that fault estimates are nearly identical for the two cases.

The role of closed-loop controller in FDI filter performance: The output sensitivity function
In the last two sections, it was shown how the proposed performance metrics can be used

to predict the worst-case performance of various FDI filters. The discussion is now taken a step further to study the important role a closed-loop controller plays in FDI filter performance. Consider the example described in Sec. 6.4.5.1 comparing multiple FDI filters.

The geometric filter (Filter 1), achieves a worst case actuator performance metric value close to zero, Fig. 6.7(a) at 0.5 rad/s, thus the actuator fault detection filter is expected to perform very well at this frequency. The time-domain linear simulation with the elevator fault entering the system at 0.5 rad/s and worst case uncertainty perturbation obtained from computation of the metric, is shown in Fig. 6.15. As expected, the filter is able to estimate elevator fault almost perfectly. Also notice in Fig. 6.15 that elevator demand is very close to steady-state and does not reflect the fault signal entering the system. This is due to the robustness property of the controller allowing it to compensate for the elevator fault and maintain tracking performance (see pitch command response in Fig. 6.15). For comparison, the identical filter and fault scenario is simulated, but this time in open-loop, yielding the time-response presented in Fig. 6.16. Immediately, it becomes apparent that

- The elevator demand reflects the sinusoidal fault entering the system. Unlike the closed-loop case, there is no controller to compensate for the actuator fault.
- The elevator fault estimate of the geometric filter in open-loop is slightly different (worse) than that seen in the closed-loop case (Fig. 6.15). This example shows that a closed-loop controller does not adversely affect FDI filter, but infact a *robust* controller may help the FDI filter perform better in the presence of uncertainty.

To appreciate why the FDI filter performance differs when implemented in the open-loop or the closed-loop, consider the FDI filter performance metrics given in Table. I. The corresponding actuator performance metric is,

$$\max_{\Delta \in \mathbf{\Delta}} \quad \|K^a_{f_1} S_\Delta G_\Delta - I\|_\infty$$

The difference between the open-loop and the closed-loop fault detection scenarios is reflected in the test metric in the output sensitivity function, $S_\Delta(s)$. The output sensitivity function is equal to identity (I) in the open-loop case (because $K_{c_1}(s) = 0$, Eq. 6.8). Hence, the term in the actuator test metric indicating the interaction between filter and controller, $K^a_{f_1}(s)S_\Delta(s)G_\Delta(s)$ reduces to $K^a_{f_1}(s)G_\Delta(s)$ in the open-loop case. Clearly, if the

gain perturbation of the transfer function matrix $S_\Delta(s)G_\Delta(s)$ under uncertainty $\Delta(s)$ is less than that of $G_\Delta(s)$, the positive influence of the robust controller becomes evident.

A magnitude plot of the upper and lower bound for the uncertain set, $\Delta(s)$ of transfer functions $S_\Delta(s)G_\Delta(s)$ and $G_\Delta(s)$ is presented in Fig. 6.17. Indeed this is exactly what was expected, the spread of gain at each frequency point for these transfer matrices is significantly less in the case of the closed-loop system. Thus, the difference in filter performance in open-loop and closed-loop becomes clear. Also note that in the absence of any uncertainty $\Delta(s)$, the term $S(s)G(s)$ as well as $G(s)$ have a unique magnitude plot, hence, the open-loop and closed-loop filter performance with the *nominal* plant, G(s), will be identical.

This analysis shows that a robust controller may actually help enhance FDI filter performance for an uncertain system by minimizing the plant perturbation around the nominal model, under the influence of plant uncertainty. Hence a closed-loop robust controller can be very beneficial to the performance of the FDI filter. This is an important attribute to stress since the integrated filter-controller design has be advocated by many in the literature, [76, 59], but not enough has been said about the importance of robustness property of the controller in the FDI filter problem.

6.5 Summary

Closed-loop worst-case performance metrics for fault detection and isolation (FDI) filters are presented to assess filter performance and controller-filter interaction in the presence of uncertain dynamics. The influence of a controller on FDI filter performance is discussed. It is shown how a robust controller may help enhance FDI filter performance in the presence of uncertain dynamics. Example applications of the proposed metrics to the NASA GTM sub-scale jet transport model were presented to highlight usefulness of the proposed metrics. Filter performance is compared using proposed metrics and validated using worst-case analysis and linear simulations.

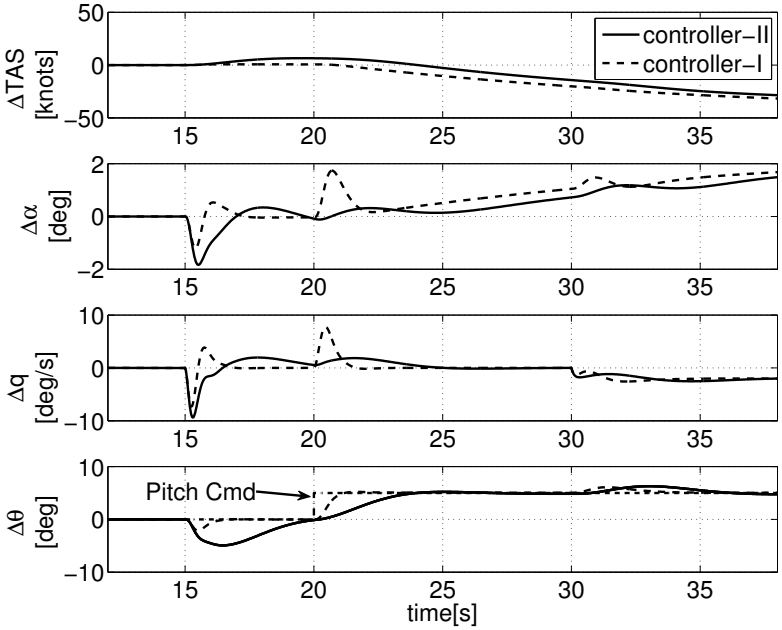
Key results of this research are:

1. FDI filter metrics are shown to be effective in comparing worst-case performance of differnt fault estimation filters.

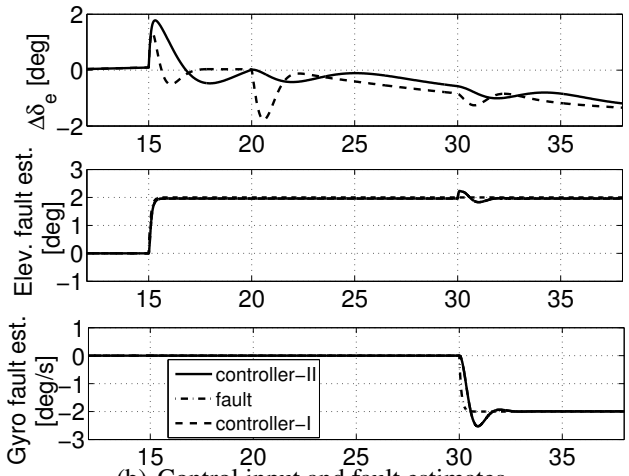
2. FDI metrics are agnostic to the FDI filter design technique employed.

3. Robustness of a controller to modeling uncertainity is beneficial from fault diagnosis perspective.

4. The output sensitivity function ($S_{\Delta}(s)$) was shown to improve worst-case FDI filter performance in the case of closed-loop.



(a) Output states and reference tracking



(b) Control input and fault estimates

Figure 6.13: FDI filter performance comparison with two different $\mathcal{H}_\infty$ pitch command tracking controllers. GTM longitudinal dynamics [Time simulation]

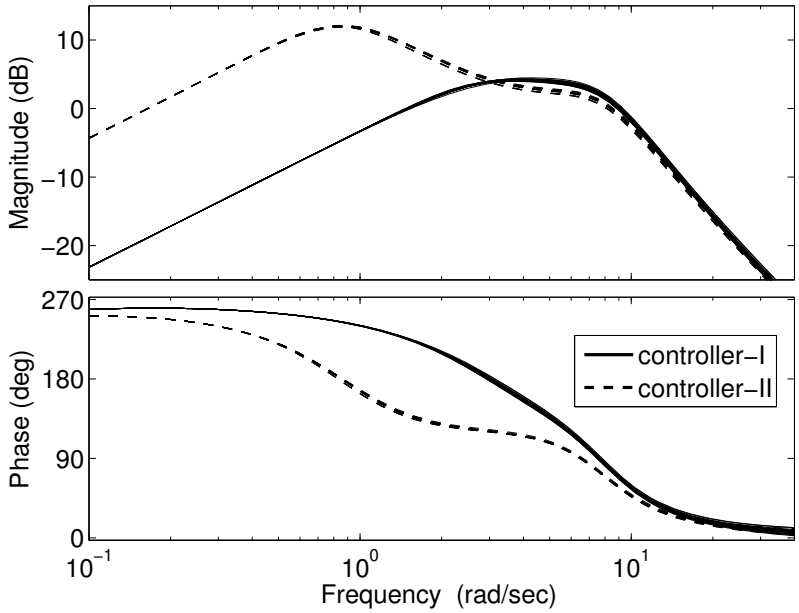


Figure 6.14: Comparison of $S_\Delta G_\Delta$ for the two controllers

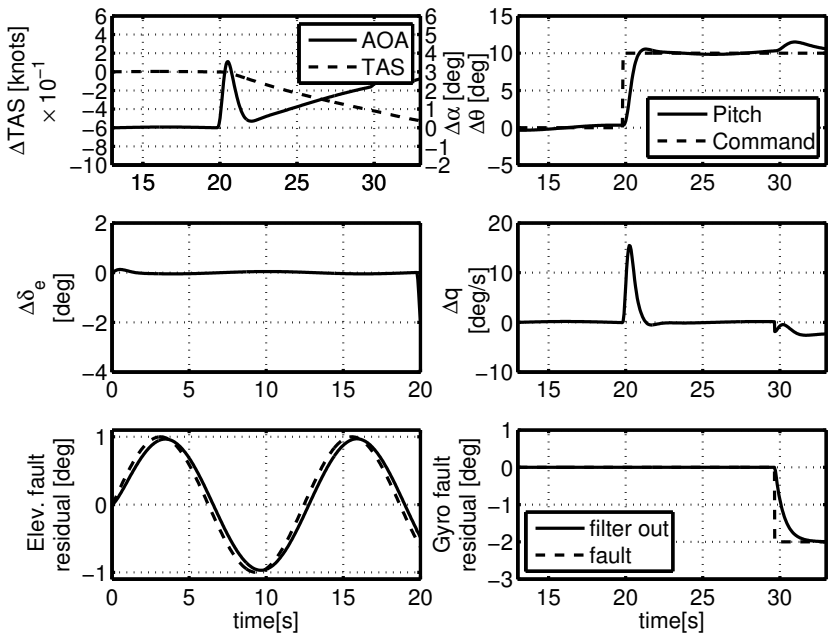


Figure 6.15: Filter 1: Closed-loop time response with 0.5 rad/s elevator fault

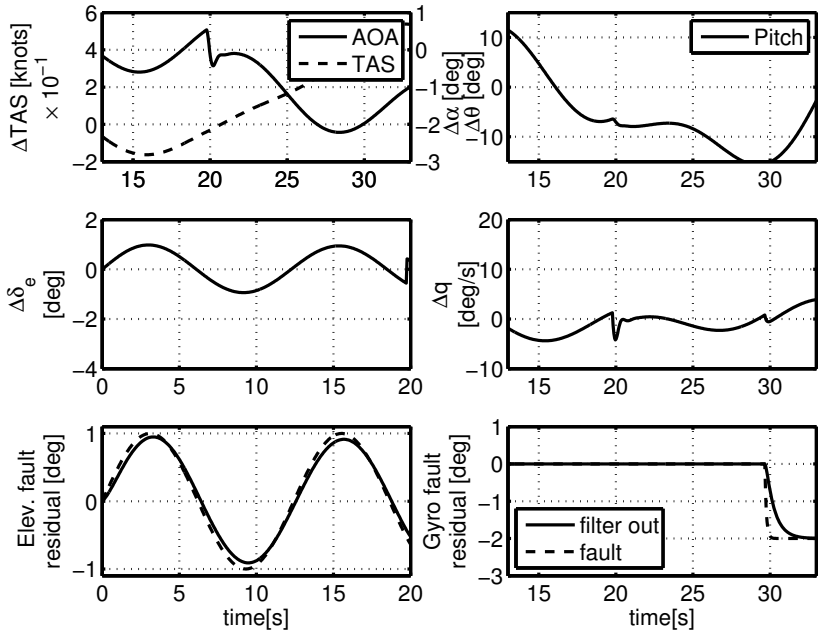
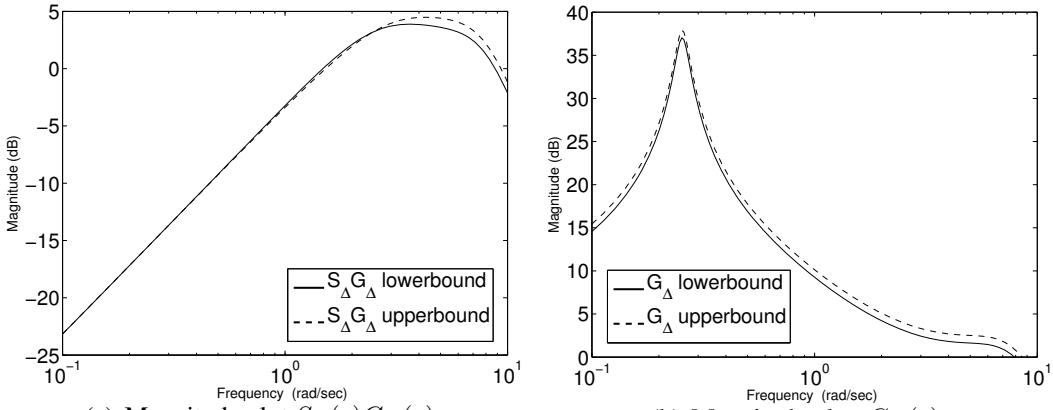


Figure 6.16: Filter 1: Open-loop time response with 0.5 rad/s elevator fault



(a) Magnitude plot,$S_\Delta(s)G_\Delta(s)$ (b) Magnitude plot, $G_\Delta(s)$

Figure 6.17: The magnitude plot comparison of open-loop and closed-loop filter influence

Chapter 7

Synthesis and flight validation of an $\mathcal{H}_\infty$ FDI filter for small UAVs

Small uninhibited aerial vehicles (UAV) are pervasive today in various military applications as well as with various civil organizations involved in public safety and rescue operations. One area that has attracted much research in the field of UAVs is the aspect of improving system reliability and safety. Traditionally, larger and more expensive UAVs have gone the way of established technology of hardware redundancy [97, 14]. However, a large number of small and medium size UAVs cannot afford the approach due to significant cost, and more significantly, weight and volume constraints. Analytical redundancy [23] is a promising technology to help alleviate these problems. Model-based fault diagnosis is one such approach to analytical redundancy which makes explicit use of the knowledge of system dynamics for fault diagnosis, [37].

Model-based techniques for fault-diagnosis have been applied to UAVs in the past. Bateman et al. [10] have applied a correlation based approach for UAV actuator fault diagnosis. Rago et al. [69] used a multiple Kalman filter approach for sensor and actuator fault detection and isolation in the Eagle-Eye UAV. The published literature in the area of aerospace fault diagnosis so far falls short of actually implementing and flight validation of FDI algorithms. A main problem in this respect is that real-world systems do not offer the luxury of perfect knowledge of system dynamics and the measured signals are affected by noise and external disturbances, all of which lead to poor FDI filter performance with false alarms and missed detections in practice, undermining utility of such filters.

Model-based FDI filter techniques use only open-loop plant dynamics for design and fail to take into account modeling uncertainties as well as controller influence. Robust $\mathcal{H}_\infty$ model-based techniques [43, 47] address model-uncertainty explicitly and offer open-loop as well as closed-loop approach to the filter design problem. In some cases, the sheer computation requirements imposed by high order filter dynamics have discouraged attempts towards online real-time implementation. The main contribution of this work is the synthesis of a robust model-based FDI filter and implementation in real-time software to demonstrate fault diagnosis capability with low-cost, low-quality sensors typically used on small UAVs. In addition, the effect of various closed-loop controllers on the FDI filter performance is investigated.

7.1 $\mathcal{H}_\infty$ FDI filter design

The standard $\mathcal{H}_\infty$ model-matching approach [44, 47] is used to design a FDI filter. The generalized plant interconnection for the problem is shown in Fig. 7.1. This design approach seeks to minimize the $\mathcal{H}_\infty$ norm of error signals of interest over the convex set of controllers (or an observer in this case), $K_f(s)$. Certain technical rank and eigenvalue conditions have to be met by the generalized plant in question to ensure a well-posed synthesis problem. The design objective in the FDI filter synthesis problem is to minimize the infinity norm (i.e. worst-case gain) of the transfer function matrix from faults (f) to weighted fault estimation error ($\tilde{e}_f = W_{p_f}(f - \hat{f})$), where, $\hat{f} = [\hat{f}_a \ \hat{f}_s]^T$ and, $\hat{f}_a$ and $\hat{f}_s$ are the estimates of rudder fault and yaw-rate gyro fault produced by the FDI filter $K_f(s)$, respectively. This synthesis problem is convex, and well developed in the $\mathcal{H}_\infty$ control theory, [44, 28] for which many standard algorithms are available in the literature.

Consider a linear plant described by a nominal state-space model,

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du \tag{7.1}$$

where, states, $x \in \mathbb{R}^n$, controls, $u \in \mathbb{R}^p$ and measurements, $y \in \mathbb{R}^m$, respectively. Let $G(s) = C(sI - A)^{-1}B + D$ be the transfer function representation of the linear plant model.

As shown in Fig.7.1, an input multiplicative uncertainty model, $G_\Delta(s) = G(s)(I+W_u(s)\Delta(s))$ is chosen to capture the uncertainty in the linear plant model given in Eq. 7.1. $\Delta(s) := \{\mathbf{\Delta} :$

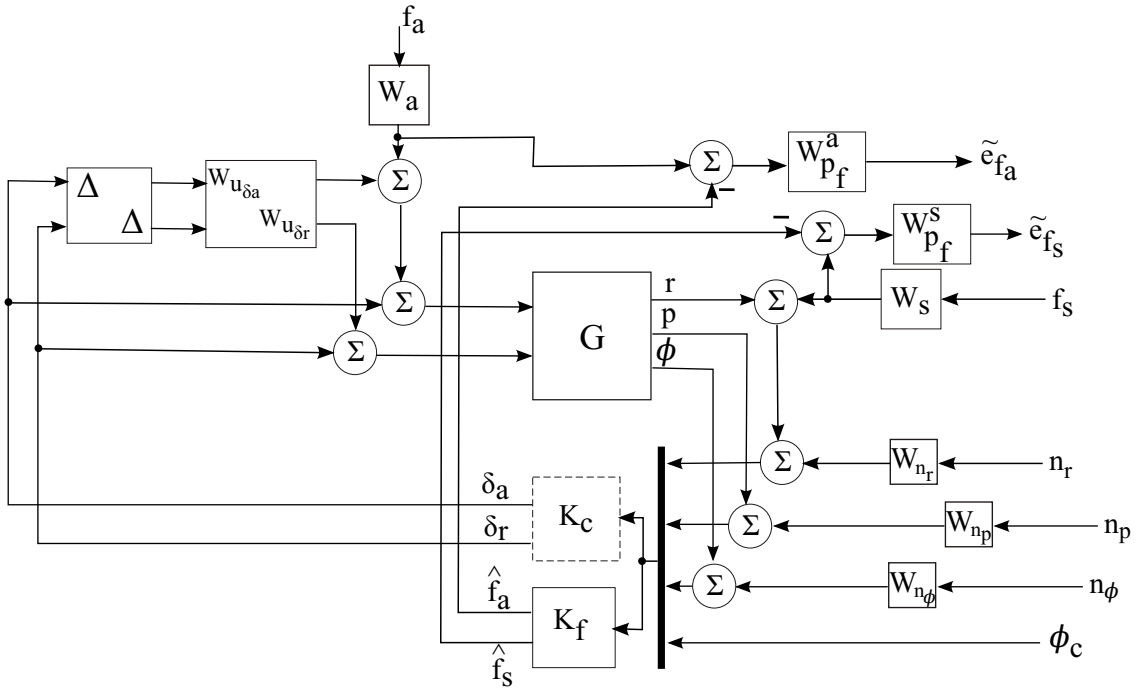


Figure 7.1: FDI problem interconnection

$\Delta \in \mathcal{C}_{m\times m}, \bar{\sigma}(\Delta) \leq 1\}$ is a complex norm bounded uncertainty with arbitrary phase. Uncertainty in the plant model such as aircraft dynamics arise due to factors like unmodeled dynamics, higher order flex-body modes, uncertainty in aerodynamic data, variability in equilibrium flight conditions, etc.

The fault detection application considered is associated with aircraft lateral-directional dynamics. This application is a multi-input-multi-output (MIMO) system with inputs aileron (δ_a, rad) and rudder (δ_r, rad) deflection, and outputs are yaw-rate (r, rad/s), roll-rate (p, rad/s) and roll attitude (ϕ, rad). The angular rates p and r are measured directly by rate gyros (sensors), whereas ϕ is estimated by an extended Kalman filter (EKF). The objective is to design the FDI filter to estimate fault in yaw-rate gyro, denoted by f_s and an actuator fault in the rudder channel, denoted by f_a. The faults enter the plant in an additive manner as shown in Fig. 7.1.

Weighting functions, $W_a(s)$ and $W_s(s)$, describe the frequency content of the actuator and sensor faults respectively. $W_a(s)$ and $W_s(s)$ are assumed to be known from fault modeling. Weighting function $W_{p_f}(s) = diag([W^a_{p_f}\ W^s_{p_f}])$ is the design parameter used to shape the desired FDI filter performance objective.

7.2 Modeling

7.2.1 UMN research aircraft model

The aircraft model used in the design and experiments is the UMN flight research platform (FRP). It is a 1.3m wingspan electric powered airplane with an all-up weight of around 1.7kg. The basic airframe is an off-the-shelf radio control (RC) airframe called the Ultrastick®25e. The airframe is retrofitted with UMN avionics hardware and software for sensing, control and datalogging.

Figure 7.2: UMN flight research platform

The UMN UAV research group has developed a nonlinear simulation model for this air vehicle (AV) and is extensively documented in [65]. The nonlinear simulation model is built in Matlab Simulink®environment. It consists of true mass-inertia properties of the instrumented airframe, along with models for the electric propulsion (electric motor and propeller data) and servo-actuators. The aerodynamic database is obtained from the NASA FASER air vehicle [60] which used a geometrically scaled variant of the Ultrastick®RC airframe with 1.93m wingspan. The *power-off* aerodynamic properties of the FASER and UMN FRP airframe is assumed to be nearly identical since the scaling ratio is not large enough to alter basic air-flow properties in the two airframes. In practice, there are some quantitative differences in the aerodynamic model and the actual flight dynamics properties of the UMN FRP. This results in some mismatch between expected and observed performance of the control laws and FDI filter as shall be shown later in the chapter. The UMN FRP simulation model also has environment simulation models builtin using the Mathworks®Aerospace Toolbox®. Environment parameters like local gravity, Earth's magnetic field strength as well as steady-winds, gusts and turbulence effects are included in the full nonlinear Ultrastick aircraft simulation.

The linear control and filter design techniques used in this work require development of a

linear model of the aircraft around an equilibrium condition. For the examples and flight testes presented here a straight-and-level flight condition was chosen. This trim condition is consistent with the flight condition seen as part of the experimental flight tests. A linear plant model is obtained by trimming the nonlinear simulation model in the chosen flight condition, i.e. indicated airspeed (IAS)$=$ 17m/s in straight-and-level flight. Nominal trim-states at this equilibrium condition are obtained as:

IAS	$=$	17.048 m/s
θ, α	$=$	0.0738 rad
ϕ	$=$	0 rad
p, r, q	$=$	0 rad/s
Elevator, δ_e	$=$	-0.0951 rad
Rudder, δ_r	$=$	0.0436 rad
Aileron, δ_a	$=$	-0.0063 rad
Throttle, δ_t	$=$	58.86%

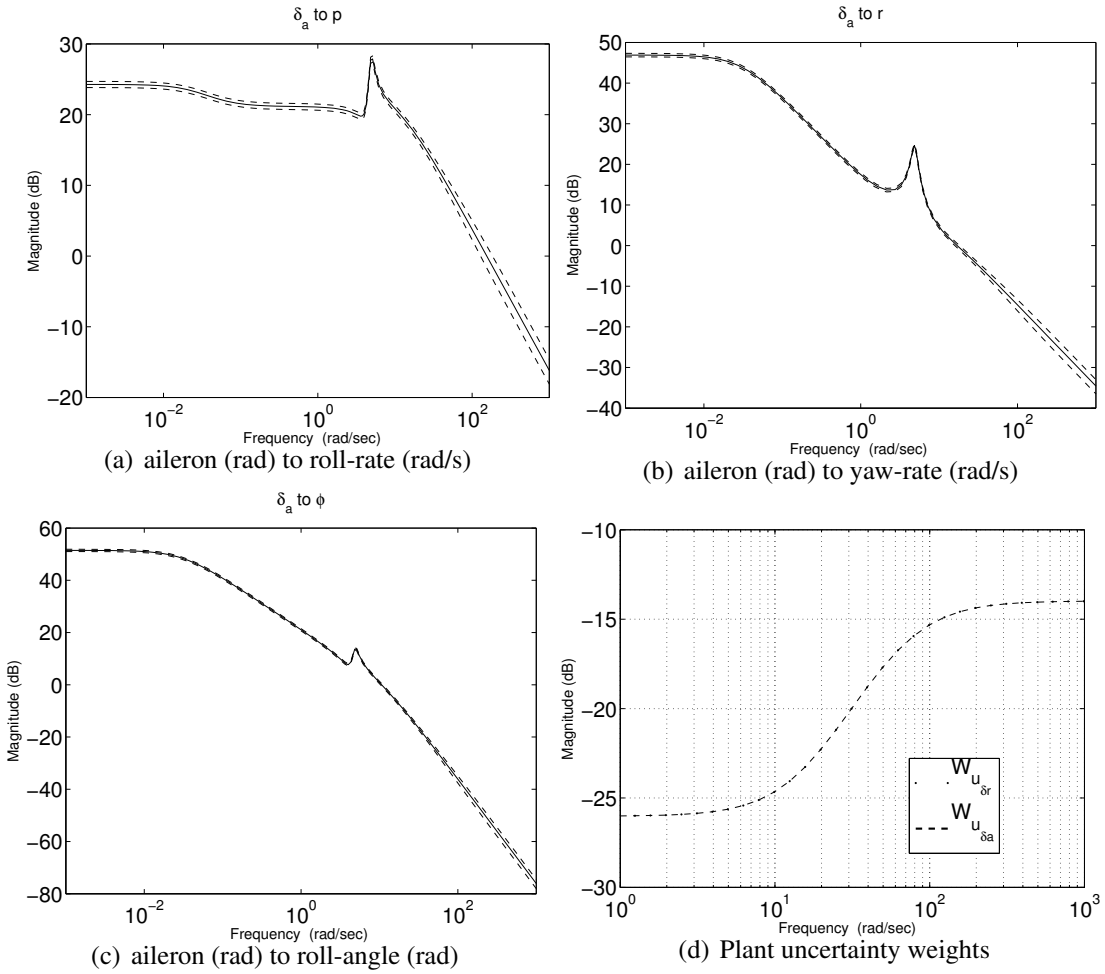
The airspeed chosen represents the nominal cruise airspeed of the UMN FRP aircraft. The trimmed nonlinear model is then linearized around this equilibrium condition to obtain the full linear model. The linear model is known to decouple into longitudinal and lateral-directional dynamics. Only the lateral-directional model is used in this work since that is the focus of FDI filter synthesis problem discussed here.

The lateral-directional linear model consists of four states namely, roll-rate (p, rad/s), yaw-rate (r, rad/s), sideslip angle (β, rad) and roll-angle (ϕ, rad), however not all of these states are available for measurement. The measured system outputs are roll-rate (p), yaw-rate (r) and roll-angle (ϕ). Control inputs for the lateral-directional model are aileron deflection (δ_a, rad) and rudder deflection (δ_r, rad). The state-space matrices for the linear lateral-directional model are given in Appendix-C.1.

7.2.2 Uncertainty modeling

An input multiplicative uncertainty model is used to overbound the dynamic uncertainty in the linear model at different equilibrium flight conditions ranging from IAS = 14 m/s to 22 m/s, with the nominal equilibrium at IAS=17 m/s. Weighting function $W_u(s) = diag([W_{u_{\delta_a}}\ W_{u_{\delta_r}}])$ depicted in the Fig. 7.1 represents the uncertainty in the input channels, ailerons and rudder. A suitable value to overbound the uncertain plant set was found to

be 5% at low frequencies and 20% at higher frequencies. This is typical in flight control problem where higher frequency structural modes are not modeled or ignored in control law synthesis, hence greater model uncertainty at higher frequency makes sense. Figs. 7.2.2 (a),(b) and (c) show frequency response upper and lower bound of the uncertain plant set (dashed lines) and the nominal model (solid line) from aileron (δ_a) input to roll-rate, yaw-rate and roll-angle, respectively.



(a) aileron (rad) to roll-rate (rad/s)

(b) aileron (rad) to yaw-rate (rad/s)

(c) aileron (rad) to roll-angle (rad)

(d) Plant uncertainty weights

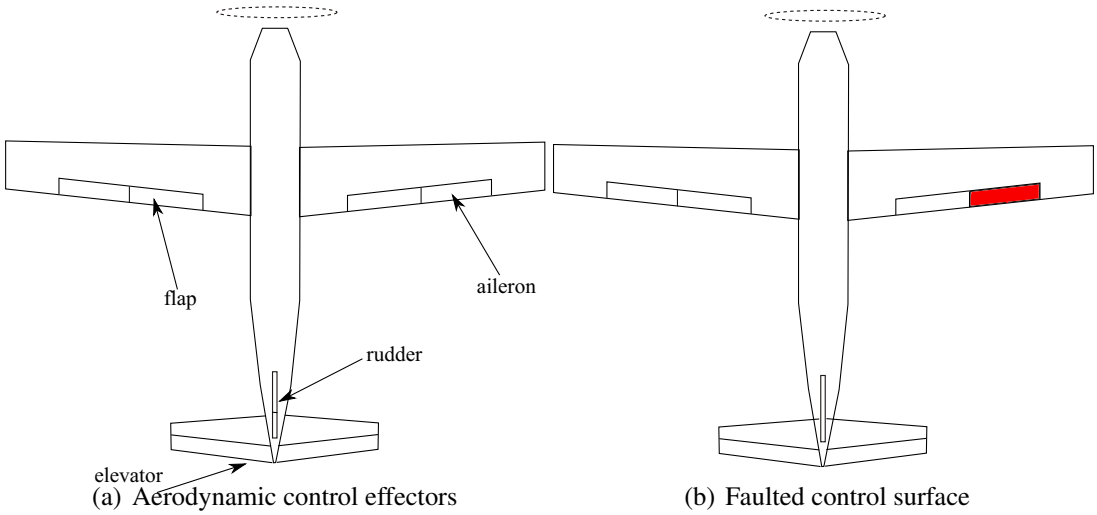
Figure 7.3: Uncertain plant set: Magnitude plots with uncertainty bounds

7.2.3 Fault model

The FDI filter design problem in this work concerns with developement of algorithm to identify aileron and yaw-rate gyro faults. Fig. 7.4(a) shows the aerodynamic control surfaces available on the UMN FRP airplane. The elevator and throttle (which controls pro-

peller thrust) is used to control the longitudinal motion of the airplane. The ailerons and rudder are used for lateral (roll) and directional (yaw) control, respectively. Together these four controls are known as the primary controls of an airplane. Secondary controls such as the flaps which are symmetric inboard surfaces as shown in Fig. 7.4(a) are used only for specific flight modes, like takeoff and landing in the case of flaps. Flaps help generate additional lift which is useful in reducing stall-speed of an aircraft.

The aileron and gyro faults can manifest themselves in multiple ways. Understanding the physics of the faults is very important for successful filter design. In $\mathcal{H}_\infty$ synthesis technique for instance, the frequency signature of various faults is heavily exploited. The knowledge about the failure modes of a particular sensor or actuator is embedded in the weighting functions $W_s(s)$ and $W_a(s)$, respectively. The model-based $\mathcal{H}_\infty$ algorithms use the aircraft model with the weighting functions to synthesize the optimal FDI filter.



(a) Aerodynamic control effectors (b) Faulted control surface

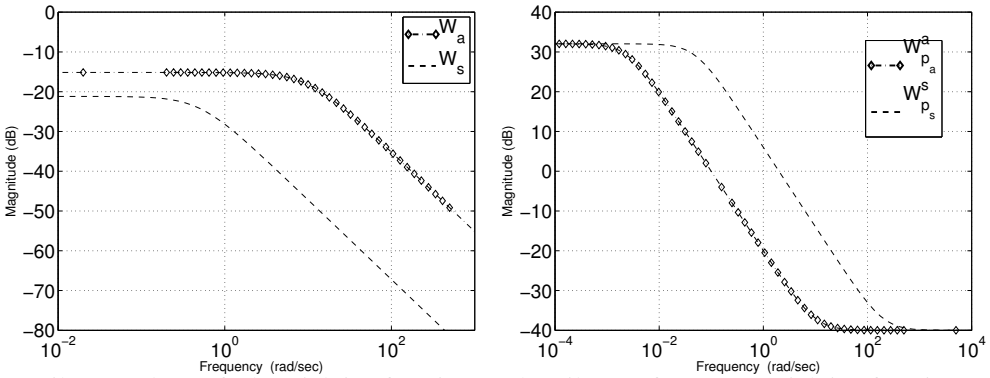
Common failure modes for aileron (and other aerodynamic control surfaces) area-

- Bias: Bias failure is a failure condition when the control surface reflects a non-zero difference or bias between actual and the commanded deflection. This can arise due to poor rigging, slippage of servo-actuator gears, or bent control linkages. This is a common fault mode for small UAVs. In the flight experiment, this failure mode shall be used as a test case.

- Floating surface: The torque transfer from the servo-actuator to the surface is disrupted if a control linkage connecting control surface to the servo-actuator is broken

or servo-drive shaft is broken. In this situation the control surface is free to float about its hinge based on aerodynamic hinge-moments acting on it and its inertia, this can cause uncommanded time-varying control force/moments on the system. Ideally, a surface is aerodynamically balanced to float to its neutral (zero) position.

- Hardover: This situation arises when the control surface gets locked at one of its extreme positions. This can be caused by a broken control linkage, broken servo gears, unbalanced control surface, etc. A hardover is the most extreme of all the failures because it imposes maximum uncommanded control forces/moments on the system.



(a) Aileron and gyro fault weighting functions (b) Filter performance weighting functions

Figure 7.4: Filter performance shaping

Small, low cost UAVs often use MEMS gyroscopes and accelerometers to measure vehicle angular rates and linear accelerations. Typically they are built to tolerate industrial shock and vibration environments, but not necessarily known for long mean time between failures (MTBFs). Common fault modes for MEMS gyroscopes are:

- Bias: Can occur due to physical damage to sensor element or poor quality manufacturing. MEMS gyros also exhibit slow time-varying bias due to temperature variation and a fixed turn-on bias, etc.
- Null or dead sensor: Unresponsive sensor, zero or an constant non-zero output.
- Floating-sensor: Often caused by poor electrical grounding or unsteady supply voltage in the avionics hardware.

7.3 The FDI filter synthesis for UMN-FRP

Standard model-matching $\mathcal{H}_\infty$ synthesis techniques are used to synthesize FDI filter for the UMN-FRP. The basic idea is to define input signal frequency domain characteristics in the generalized plant (see Fig.7.1) by the way of weighting functions (stable, proper, minimum-phase transfer functions). Performance objectives are characterized by performance weights that penalize error signals of interest (or the sensitivity function (S)). Filter synthesis solves the following convex optimization problem,

$$\min_{K,\omega} \max_{\Delta} \|\bar{G}(j\omega)\|_\infty$$

where, K is a convex-set of filters, $\bar{G}(j\omega)$ is the transfer function from all input signals to all output signals of the generalized plant shown in Fig. 7.1.

One key, but often less stressed, aspect of the FDI filters design and verification is that frequently the filters operate in parallel with closed-loop controllers in a dynamical system. Rarely are systems operated purely in open-loop mode where fault diagnosis capability is required. Industrial plants, chemical process control, automotive system and aircraft ,all have active control systems engaged. Effect of controller influence on FDI filter performance is a subject of active research [70]. Clearly there's a need to evaluate filter performance in the presence of controllers. Researchers have proposed a variety of approaches to synthesize FDI filters in the literature, though often these approaches consider an open-loop plant. Researchers have shown how controller and filter design is coupled in case plant dynamics have some uncertainty. Optimal performance requires simultaneous design of the controller and the filter [76]. This approach is known as integrated filter-controller design. In practice this is frequently not feasible since the control system is designed independently from the fault diagnosis system.

The FDI filter synthesis can be classified in the following different ways:

1. *Integrated design*: Both controller and the filter are designed simultaneously to meet tracking and FDI objectives. This approach often leads to good performance though a drawback is that it cannot be used to retrofit existing systems.

2. *Discrete design*: The controller and filter are designed separately. This approach may not offer best performance, but can be used to design filters for existing closed-loop systems. The controller behavior is implicitly embedded in the FDI filter.

3. *Discrete open-loop design*: Only system outputs and inputs are available to the filter. The FDI filter has no knowledge of the dynamic properties of the controller. As expected, the filter performance is degraded compared to the other approaches, but offers most flexible design framework so that the filter can be used with any controller for a given plant.

A discrete open-loop $\mathcal{H}_\infty$ based technique for the FDI filter design in this paper since it allows us to test a variety of controllers with the same FDI filter. This approach also helps to highlight limitations of an open-loop design for a practical real-world system with model-uncertainty, noise and external disturbances.

The aim is to identify and estimate low frequency and bias faults in both aileron and yaw-rate gyro for the FDI filter design problem under consideration. In model-based $\mathcal{H}_\infty$ synthesis this is achieved by *shaping* the signals of interest using proper, stable, minimum-phase transfer functions (usually low-pass filters) appropriately. As shown in Fig.7.1, this achieved by filters $W_a(s)$ and $W_s(s)$ which shape input fault signals f_a and f_s, respectively. The magnitude plot of $W_a(s)$ and $W_s(s)$ is shown in Fig.7.4(a). These two transfer functions are also known as fault models since they characterize the fault characteristics in the frequency domain. The steady-state gain of $W_a(s)$ the aileron fault model and $W_s(s)$, the yaw-rate gyro fault model are 0.1745 rad (or 10 deg) and 0.0873 rad/s (or 5 deg/s) , respectively. This relates to the maximum magnitude of the respective errors expected. The corner frequency of $W_a(s) = 10$ rad/s is chosen based on the closed-loop bandwidth of the roll-command tracking loop of 12.5 rad/s, we would like the FDI filter to be able to track aileron faults extending through the entire closed-loop bandwidth. In the case of yaw-rate gyro fault, the corner frequency of $W_s(s)$ is chosen at 0.5 rad/s, which means means an interest in low frequency faults like the bias fault.

7.3.1 Filter performance weights

Fig. 7.4(b) shows the performance weighting functions chosen to shape the closed loop filter sensitivity function (S). Transfer functions $W_{p_f}^a$ and $W_{p_f}^s$ ultimately control the steady-state error of the filter, the bandwidth as well as high frequency roll-off. The functions are designed to meet the desired performance specifications of less than 3% steady-state error (low frequency gain of $W_{p_f}^a$ and $W_{p_f}^s$ is 30dB) in fault estimation, rise-time (detection time) of less than 20 seconds.

7.3.2 Noise model

Sensor noise models used in the filter synthesis are shown in the magnitude plot in Fig. 7.3.2. The noise weighting functions are determined based on manufacturer specification sheets. Sensor noise has mostly high-frequency content, the yaw-rate gyro noise model (W_{n_r}) reflects high frequency noise of up to 3.5 deg/s, the roll-rate gyro noise model (W_{n_p}) is artificially kept high as a tuning parameter to force FDI filter's high frequency poles to lower acceptable levels.

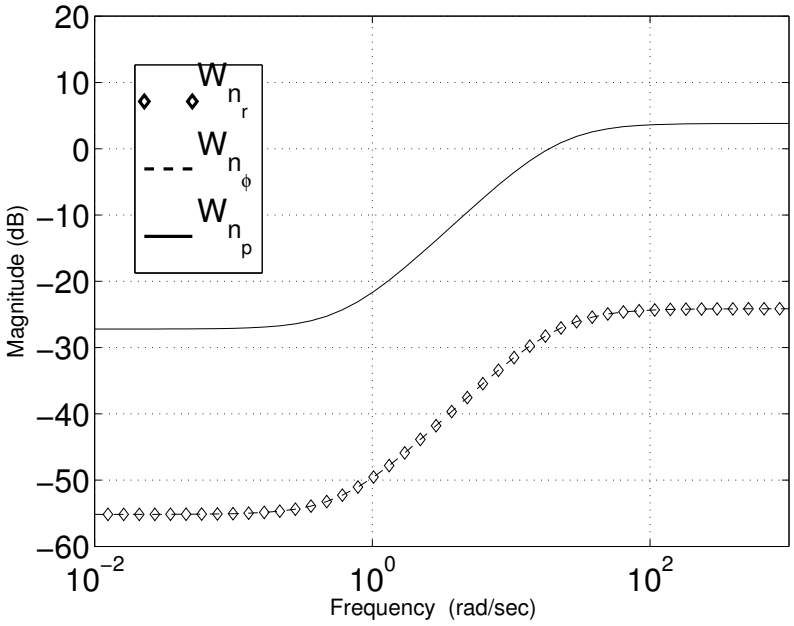


Figure 7.5: Measurement noise weighting functions

An $\mathcal{H}_\infty$ optimal FDI filter is obtained with 11 states. The filter state-space data and weighting functions are given in Appendix-C.3. This filter is discretized using Tustin's bilinear transform method at a sampling frequency of 50Hz for real-time implementation.

7.4 Tracking controllers

A common flight control tracking problem that is readily affected by aileron fault is the roll-angle command tracking problem. This is because ailerons are the primary control surfaces used for roll control. Three roll-angle command tracking controller designs are considered - a classical PID controller. an linear quadratic (LQ) optimal controller and an adaptive controller.

The performance objectives for the linear controllers are - steady-state tracking error less than 2%, rise-time less than 3 seconds, overshoot less than 20%. Robustness objectives used for design are gain-margin of at least 6dB and a phase-margin of atleast 45 degrees. For the adaptive controller, linear performance and robustness metrics are irrelevant, hence heuristic tuning is used, in conjunction with time-domain simulations to achieve the desired performance and robustness objectives.

7.4.1 Classical controller

The PID classical-control law is given in Eq. 7.2. The classical roll-controller consists of PID gains with integrator anti-windup scheme built-in. A yaw-rate damping loop with a washout filter is used to suppress the lightly damped dutch-roll mode of the lateral-directional dynamics.

$$\begin{aligned} \delta_a(s) &= (K_P + \frac{K_I}{s})(\phi_r - \phi) + K_D p \\ \delta_r(s) &= K_{YD}(s) r \end{aligned} \tag{7.2}$$

Controller gains and yaw-damper ($K_{YD}(s)$) transfer-function are given in Appendix-C.2.

7.4.2 Optimal controller

A linear quadratic optimal roll-controller is designed based on the optimal control theory. Some modifications are incorporated based on ideas presented in [34] in which the authors provide a framework for adding integral action in the LQ controller. It may be noted that canonical optimal controller is a *static state-feedback* controller. Incorporating integral action allows use of smaller controller gains to meet steady-state performance requirements, as well as provide zero steady-state tracking error.

$$\delta_a(s) = \frac{K_{11}^{LQ}}{s}(\phi - \phi_r) + K_{12}^{LQ} p + K_{13}^{LQ} r + K_{14}^{LQ} \phi \tag{7.3a}$$

$$\delta_r(s) = \frac{K_{21}^{LQ}}{s}(\phi - \phi_r) + K_{22}^{LQ} p + K_{23}^{LQ} r + K_{24}^{LQ} \phi \tag{7.3b}$$

Not all system states are available for measurement, hence static *output-feedback* based on projection based eigen-structure assignment [96] is used to retain the dominant eigen-structure of the ideal state-feedback optimal control closed-loop system. Controller gains

are given in Appendix-C.5.

7.4.3 Adaptive controller

Adaptive controllers offer a powerful framework to address system nonlinearities and variations. Various theoretical adaptive design frameworks are available in the literature, [35], which can be broadly classified into two branches - indirect and direct adaptive controllers. A direct model-reference adaptive controller (MRAC) works by directly modifying controller gains, this in contrast to the indirect MRAC technique in which adaptation is essentially in the system parameter identification part, the controller itself is parametrized in terms of the system parameters. The direct MRAC approach does not suffer from some theoretical limitations of the indirect design, which require subtle nuances like the persistence of excitation [35]. Direct MRAC augmentation approach has been successfully applied by Wise et al. [34] to design a lateral-directional controller for a UCAV. The adaptive law is given in Eq. 7.4.

$$\dot{\hat{K}} = -\Xi\,\Gamma x e^T P B \tag{7.4}$$

where, $x = \begin{bmatrix} \int(\phi - \phi_r)dt & v & p & r & \phi \end{bmatrix}^T$ and $e = x - x_m$ where x_m is the state vector of the reference model. $\hat{K}$ is the adaptive augmentation gain which acts in conjunction with the nominal LQ gain based on the law given in Eq. 7.4. Ξ is a projection operator that maps optimal control law from state-feedback to output-feedback space. The augmented law is,

$$u = -(K_{LQ} + \hat{K})y \tag{7.5}$$

A direct model reference adaptive controller (DMRAC) augmentation is applied to the LQ controller described in the previous subsection. The main idea is that if plant dynamics change due to physical damage or failure, the controller gains are modified so that the closed-loop behavior is unaffected as much as possible. This ideal system behavior is embedded in the DMRAC controller in the form of a reference model given in Appendix-C.4 along with other design parameters. The reader is refered to [34] for further details on the theory and design methodology. The main drawback of the adaptive design is the lack of robustness to time-delay. To overcome this limitation, a conservative adaptation rate (Γ, see Eq. 7.4) is used for this design. This ensures that adaptation rate $\dot{\hat{K}}$ is slow and less aggressive.

7.4.4 Controllers: Linear analysis

Performance and robustness analyis is presented for the two linear controllers designed in the previous section. Figs. 7.6 and 7.7 show the comparison of Bode and Nyquist plots for the roll-command (ϕ_c) to roll-angle (ϕ) loop of the classical and LQ optimal controllers, respectively. Both the controllers exceed robustness requirements of 6dB gain-margin and 45 degree phase-margin. Cross-over frequencies are 1.6 rad/s and 3.85 rad/s for the classical and the LQ closed-loop system respectively. Hence the PID controller falls short on the rise-time requirements, whereas LQ controller meets it. Zero steady-state error is achieved as seen from unity gain at low frequency in the closed-loop magintude Bode plots. Comparing Nyquist plots of the two designs, one can see that loop-gain is clear of (-1,j0) point and does not encircle it. The closed-loop system is stable as required.

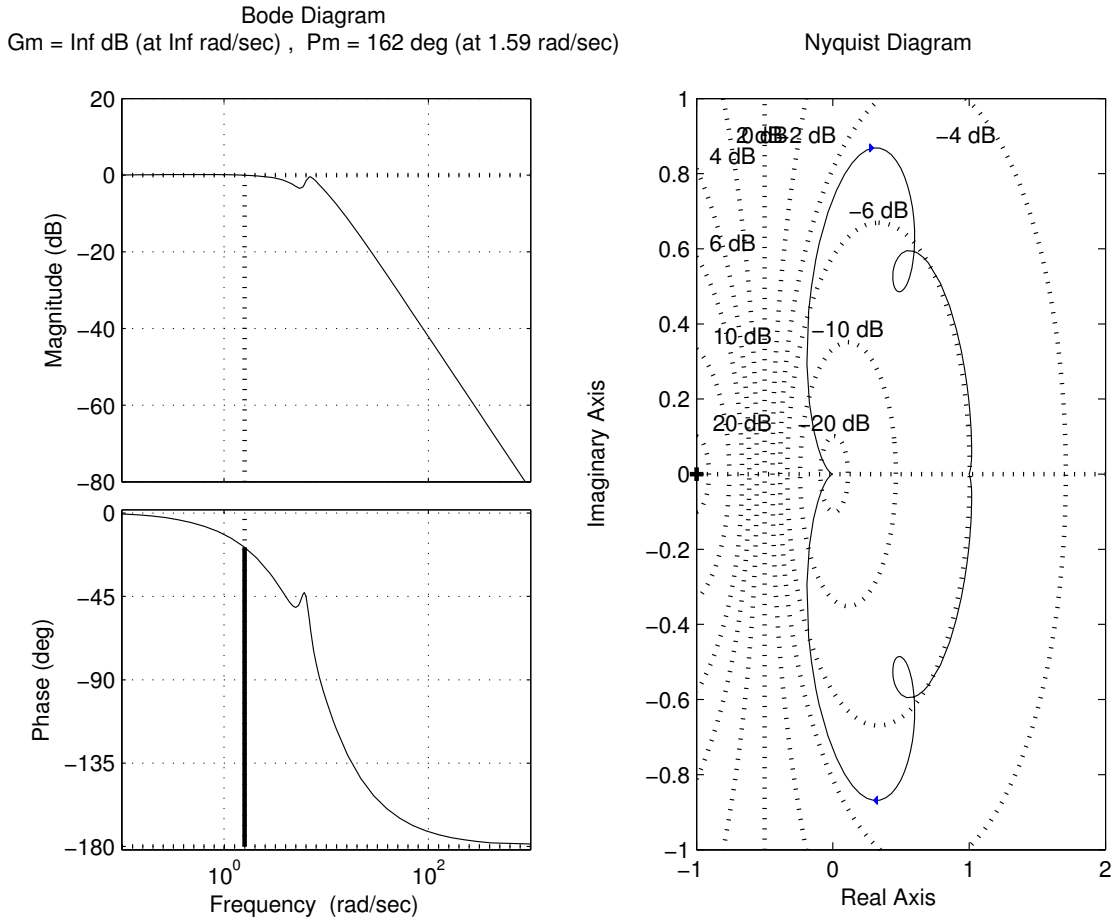


Figure 7.6: PID closed loop: Bode and Nyquist plots

An interesting insight is obtained comparing Nichols charts. In Fig. 7.8, the chart on left corrsponds to the LQ controller, the one on the right is for the classical controller. Although

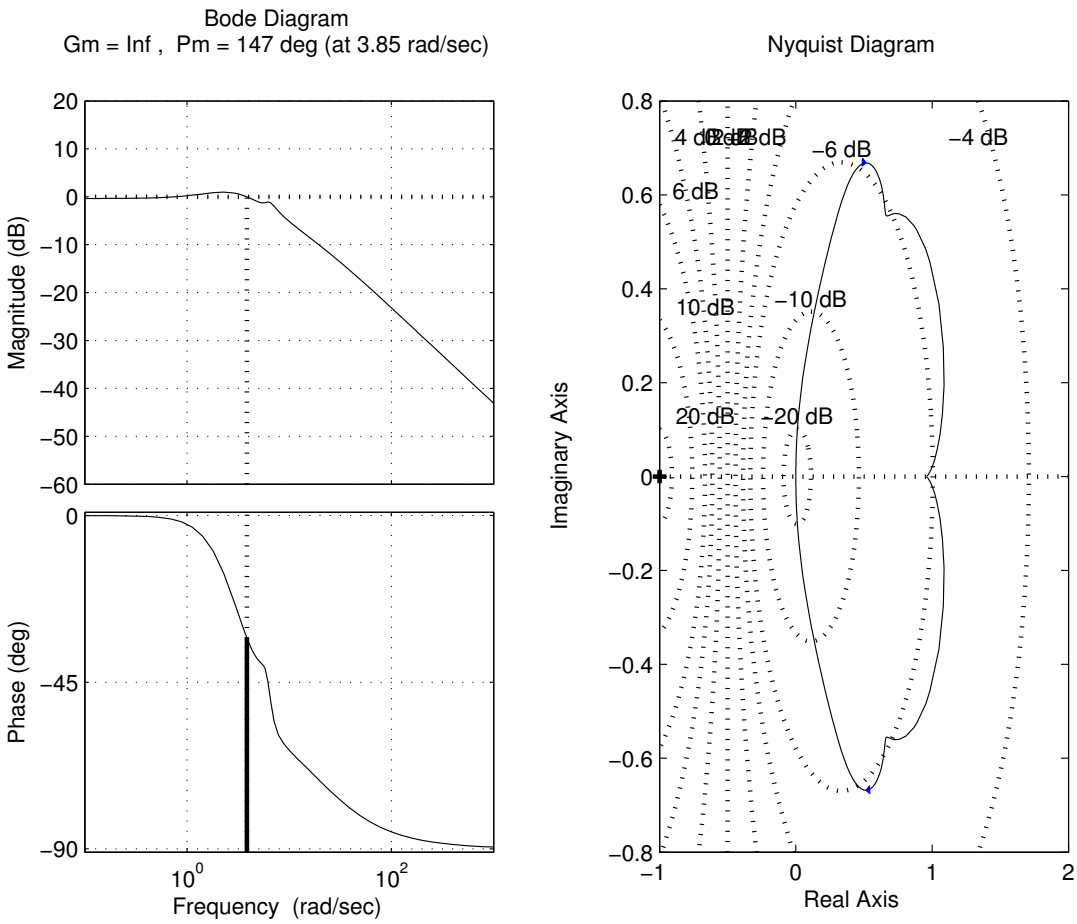


Figure 7.7: LQ closed loop: Bode and Nyquist plots

Bode plot suggests that the classical controller is more robust because it has PM=162deg, GM=Inf compared to PM=147deg,GM=Inf for the LQ controller. The Nichols charts indicate that the LQ controller has a much larger exculsion zone compared to the classical one, which means LQ controller is more robust to simultaneous gain and phase variation, which is a more realistic measure of system robustness.

7.5 Simulation results

Linear simulations comparing tracking performance of the two controllers as well as the FDI filter is presented. The test case for the simulation consists of a roll command doublet initiated at t=10 seconds lasting up to t=16 seconds. At t=6 seconds an aileron fault of 5 deg is injected as a step change. Fig. 7.9 and 7.10 shows the time-response of the nominal linear system. One can see that classial controller in Fig.7.9 is unable to handle the fault

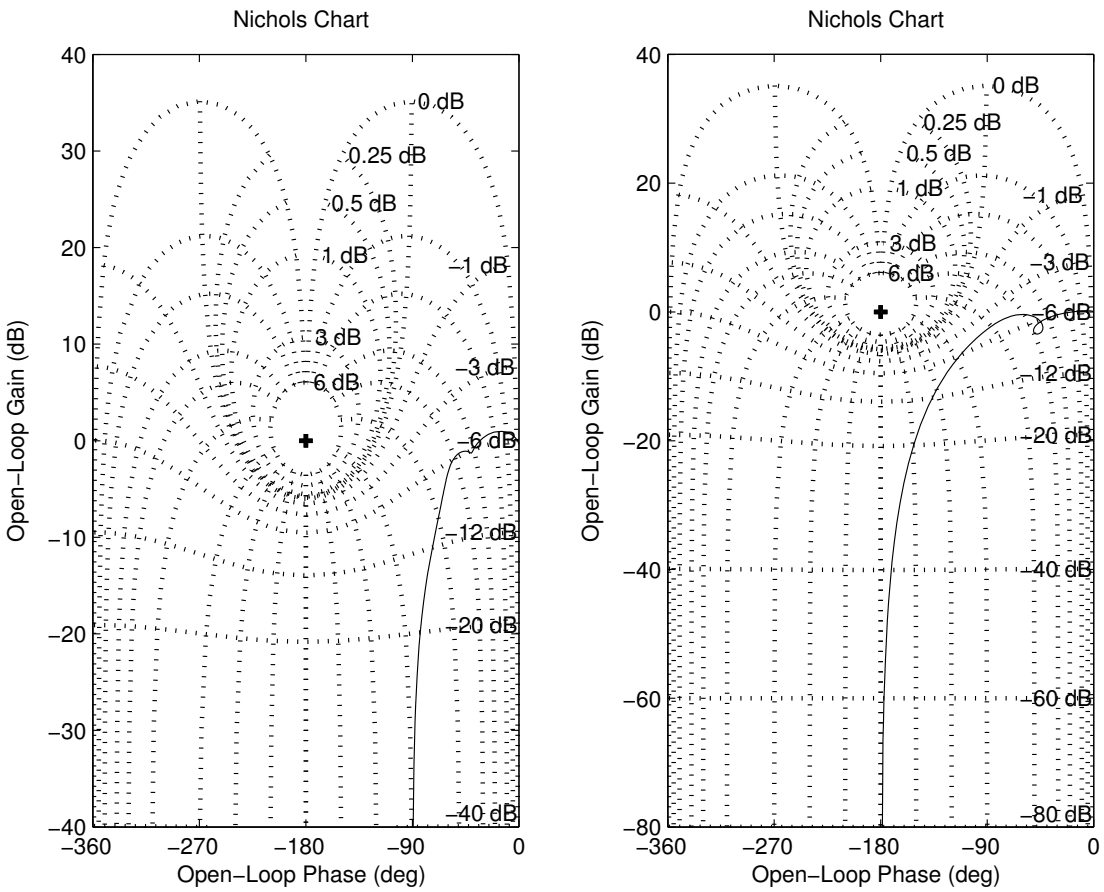


Figure 7.8: Nichols chart: PID and LQ closed-loop systems

well, and tracking performance suffers. In contrast, Fig.7.10 shows LQ controller is able to perform very well under identical fault. A transient roll perturbation is seen at t=6 seconds due to the fault being injected, this is natural. An interesting aspect is that the FDI filter performs quite differently in the two cases. One can see that while under LQ control, the FDI filter is able to identify and estimate the fault almost perfectly, with no cross-coupling from other channels. The FDI filter with classical controller suffers significant coupling from roll-command channel. An identical scenario is simulated at an off-design condition using linear model of UMN-FRP at IAS=22m/s to test the robustness of the two controller-filter setups to plant model variation. One can clearly see from Fig. 7.11 and Fig. 7.12 that both the systems suffer from poor FDI filter performance. Controller performance is unaffected by the change in plant model. It is interesting to note that while FDI filter under classical controller has larger spikes due to coupling with roll command, the steady-state error is still practically zero. On the other hand, with LQ controller, the coupling peaks are smaller but, convergence to zero steady-state error is markedly slower.

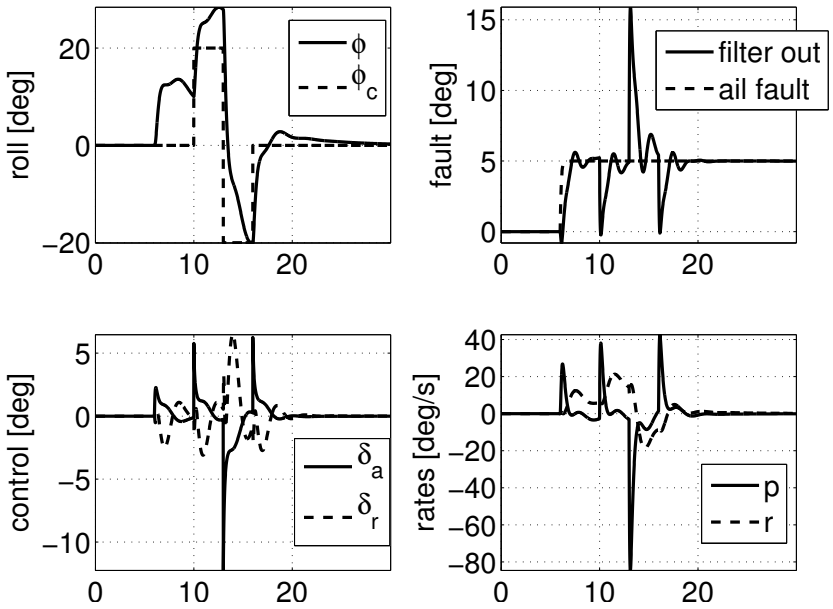


Figure 7.9: Linear simulation: PID controller with FDI filter, aileron fault, nominal flight condition

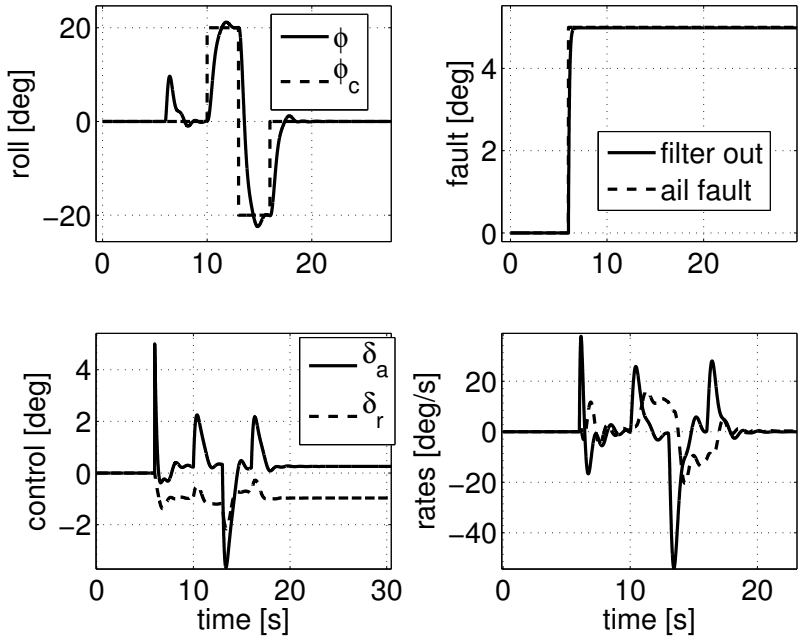


Figure 7.10: Linear simulation: LQ optimal controller with FDI filter, aileron fault, nominal flight condition

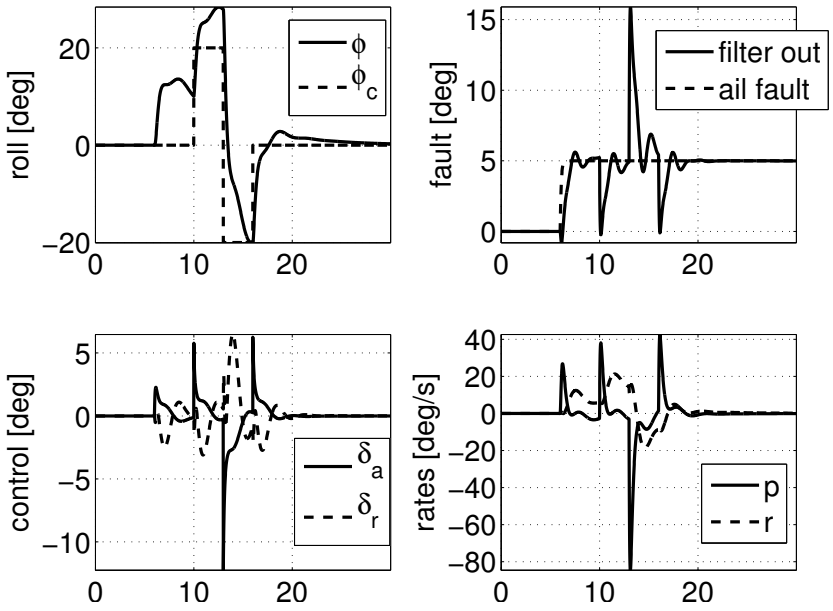


Figure 7.11: Linear simulation: PID controller with FDI filter, aileron fault, off-design flight condition

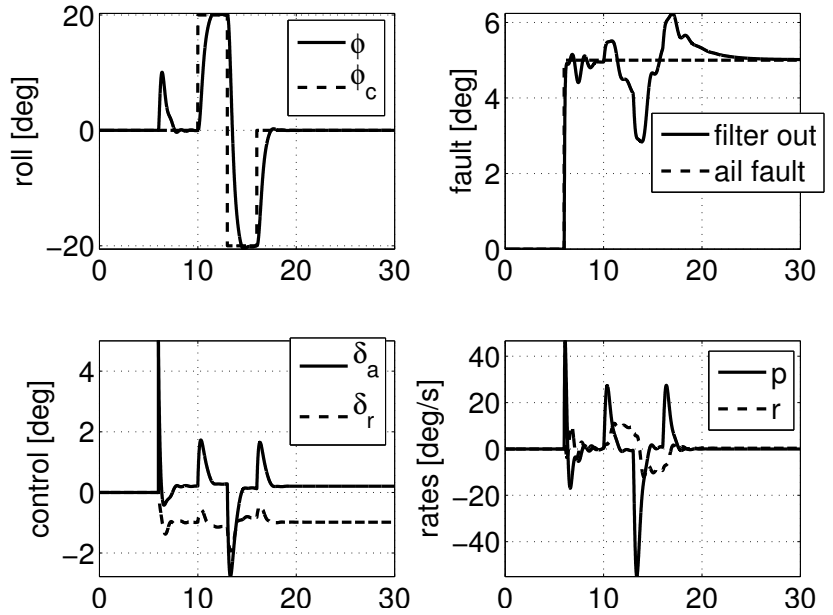


Figure 7.12: Linear simulation: LQ controller with FDI filter, aileron fault, off-design flight condition

7.6 Experiments

The $\mathcal{H}_\infty$ FDI filter design was test flown on the UMN FRP. Three roll-command tracking controllers presented in section-7.4 were tested. The main objectives of the tests were as follows:

1. Experiment 1: Compare the robustness of the three controllers to an aileron fault.
2. Experiment 2: Compare the performance of the FDI filter to detect and estimate the aileron fault under the influence of the three roll controllers.

7.6.1 Setup

UMN-FRP is a commercial-off-the-shelf (COTS) airframe with COTS sensors and avionics, these hardware components were integrated in the AEM UAV lab at University of Minnesota. The avionics hardware consists of the following components:

1. Analog Devices ADIS16405 inertial measurement unit (IMU): Measures three body axes rates and accelerations as well as components of Earth's magnetic field. Temperature calibrated.
2. Phytec MPC5200-Tiny System on Module (SoM): Credit-card sized Freescale® MPC5200B microcontroller based computer. 32-bit 400MHz CPU with 64-bit hardware floating-point unit (FPU).
3. SiRF Star-III 1Hz GPS chipset.
4. Honeywell static and differential pressure sensors.

The IMU provides angular rates, linear acceleration and the magnetic field strength in the three body axes. A GPS receiver provides position and velocity measurements, and an airdata system consisting of absolute and differential pressure sensors is used to measure pressure altitude and the airspeed of the vehicle. Data from all theses sensors are acquired by Freescale® MPC5200B microcontroller based flight control computer (FCC). The FCC operates in a prioritized multi-threaded environment provided by an open-sources real-time operating system (RTOS) called eCos [2]. A custom port of eCos was developed in-house at the UAV Lab for Phytec MPC5200B SoM.

The FCC records the relevant sensed parameters and control variables in real-time for post flight analysis. The vehicle takes-off and lands under pilot control. The research experiment is initiated by the pilot by flick of a switch from the ground on the R/C transmitter. This discrete signal is registered by the FCC which then enables research algorithms and sends control surface commands to the servos if needed.

7.6.2 Results

Flight experiments were carried out on UMN-FRP Ultrastick with UMN avionics suite. Three different controllers described in Section 7.4 were tested. The FDI filter operates in parallel with the controllers and monitors the closed-loop system for aileron faults. The controller and filter are initialized once a discrete command from ground R/C transmitter is received. This time is denoted by t_0 in the following discourse. A pre-coded fault signal time-series and roll-command doublet is executed with respect to t_0. A pitch controller is used to track a 5 deg pitch attitude command. The throttle is set to 70% to obtain a constant airspeed flight profile of 17m/s at approximately 100 meters altitude.

Two different experiments are carried out:

1. Three roll doublet commands are applied in succession. After the first doublet is completed, right aileron fault of +10deg locked aileron is applied. The purpose of this test is to establish baseline roll command performance without fault and with the first doublet command. Followed by two doublet commands in the presence of aileron fault, to assess fault tolerance capability of each controller and FDI filter fault estimation performance.

2. A single roll doublet is commanded in presence of the fault. The fault is removed at the end of doublet command. This experiment tests that the FDI filter output returns to zero when the fault in the system is removed.

To summarize, the main objectives of these experiments is:

- To evaluate if the FDI filter can detect and estimate the aileron fault.
- The FDI filter output must return to zero once the fault is removed.
- The effect of fault on closed-loop roll tracking performance vis-a-vis different controllers.

- Evaluate the relative performance of the FDI filter in presence of different controllers.
- Compare predicted simulation results with the flight tests.

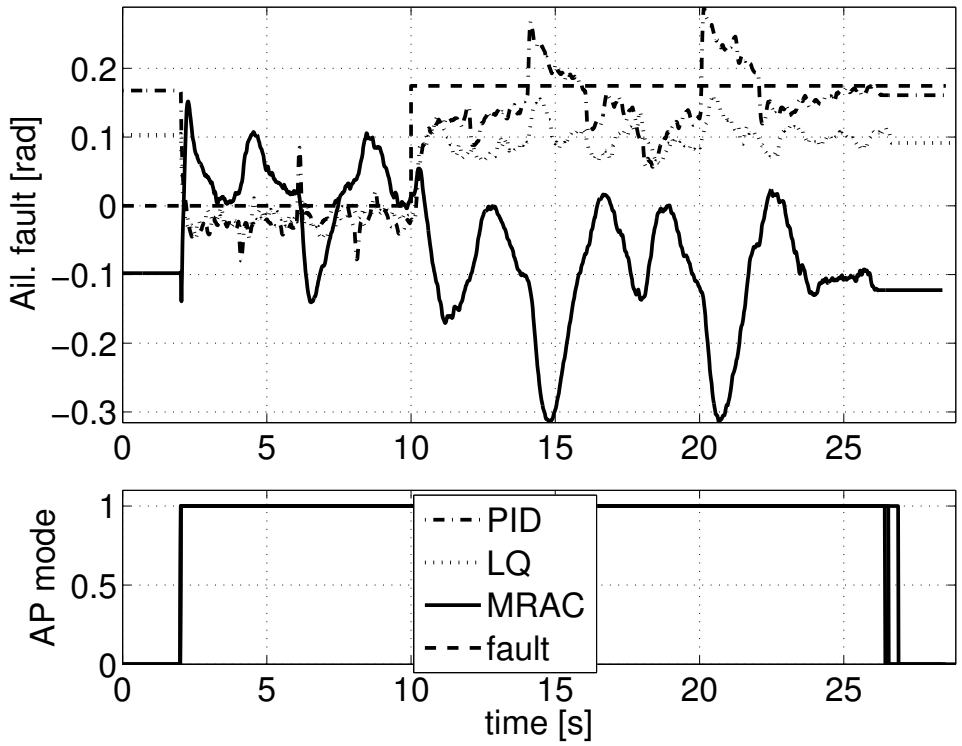
7.6.2.1 Experiment 1: Aileron fault 10 degrees, step fault, three doublets

Flight test results of this experiment are shown in Figs. 7.13. Classical PID, LQ optimal and the adaptive augmentation controllers were tested. Fig. 7.13(b) shows the tracking performance of the three controllers. Recall that the controllers being investigated are associated with the lateral-directional axes of the aircraft. An identical pitch controller is used in all experiments to track 5 deg pitch command reference. The parameter *'AP mode'* indicates the status of FCC, i.e. whether the pilot or onboard flight program (OFP) is in control, Fig. 7.13(a). One can see that at t=2 seconds autopilot or OFP is engaged. Fig. 7.13 (b) shows pitch (θ) and roll (ϕ) tracking response. The dashed line indicates the command reference. The three controllers are seen to track the first doublet (no fault case) quite well as expected from the linear analysis. Aileron fault is injected at t=10s, causing a transient in roll angle. The LQ and adaptive controller recover nominal performance quickly, but the PID controller is unable to track roll command as seen from the nex two doublet command sequences. This shows that although the nominal performance of the three controllers in very similar. The LQ controller and LQ with adaptive augmentation are able to handle the fault better than the PID controller.

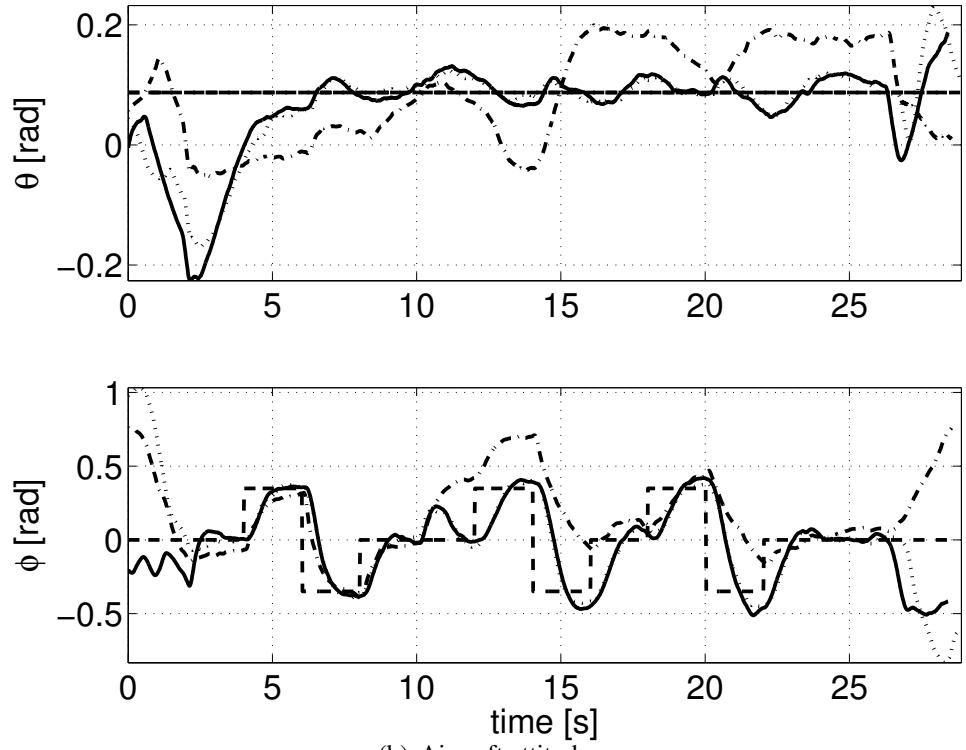
Fig. 7.13 (a) shows the response of the FDI filter. It is observed that filter fault detection performance under PID and LQ controlled dynamics is very good, the filter output (fault estimate) stay close to zero from t=2 to t=10s as one would expect. The three transients that occur during this period are due to the changes in roll command reference. This coupling was also observed in simulation at off-design operating condition, which is the case here (Fig. 7.13(d)). Another contributing factor is the model mismatch between what is used for filter synthesis versus actual aircraft dynamics. However it is interesting to note, despite the apparent difference in the model used for filter design and the true aircraft dynamics, the qualitative characteristics of the filter performance in off-design flight condition matches with the simulation data. For instance, comparing Fig. 7.13(a) with Figs. 7.12 and 7.11, it can be easily seen that under PID control, the filter exhibits more coupling from roll-command reference channel as shown by large peaks, compared to the LQ controlled plant. Also, it is seen that filter exhibits excellent steady-state fault estimation with PID controller, whereas with LQ controller fault estimation shows some bias in the steady-state.

The MRAC augmented system affects the filter performance in the worst possible manner as seen in Fig. 7.13 (a). This could be as a result of nonlinearities associated with the adaptive control, but not investigated further. It is also noteworthy that the MRAC augmentation doesn't offer any perceptible benefit in command tracking vis-a-vis unaugmented LQ controller. This could be attributed to conservative adaptation rate chosen for the design. It is interesting to note that the MRAC controller does degrade the fault diagnosis filter performance.

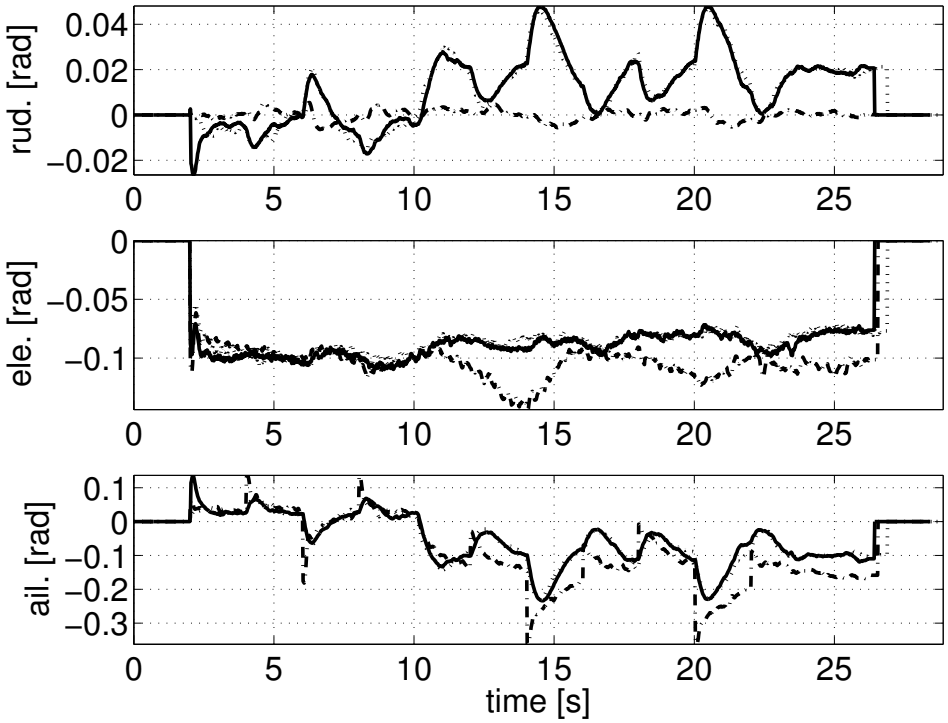
Control surface demands and altitude-airspeed for the experiment is plotted in Fig. 7.13 (c) and (d), respectively. MRAC controller exhibits larger activity (or gain) in the rudder channel indicating that it is trying to compensate for loss of aileron control power with rudder control.



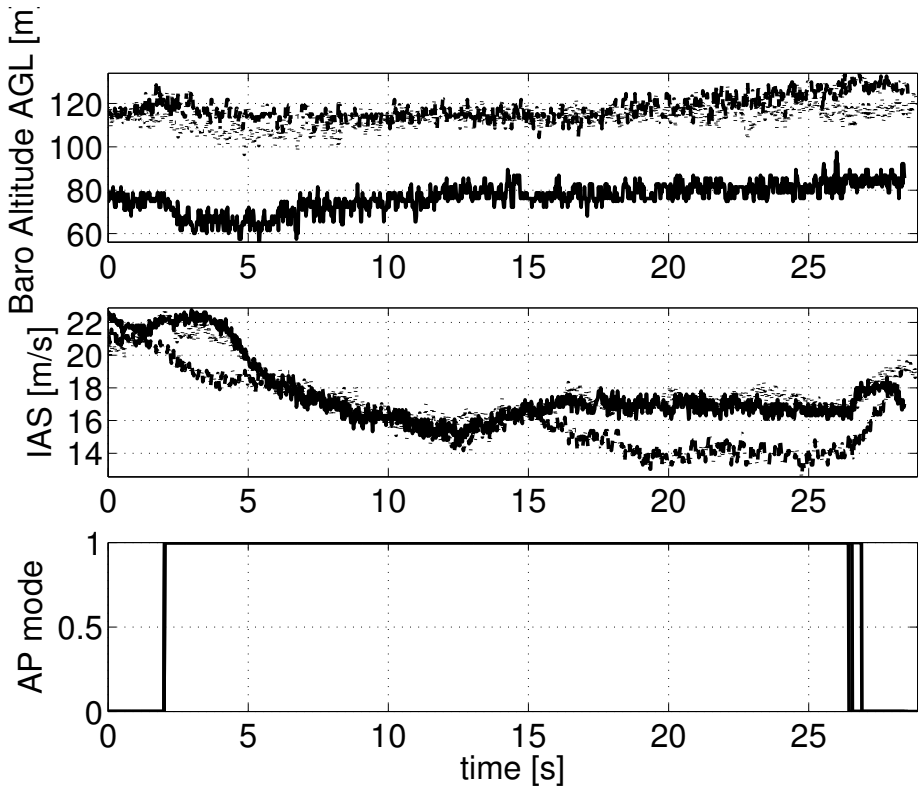
(a) FDI filter output



(b) Aircraft attitude



(c) Control surface commands



(d) Altitude and airspeed

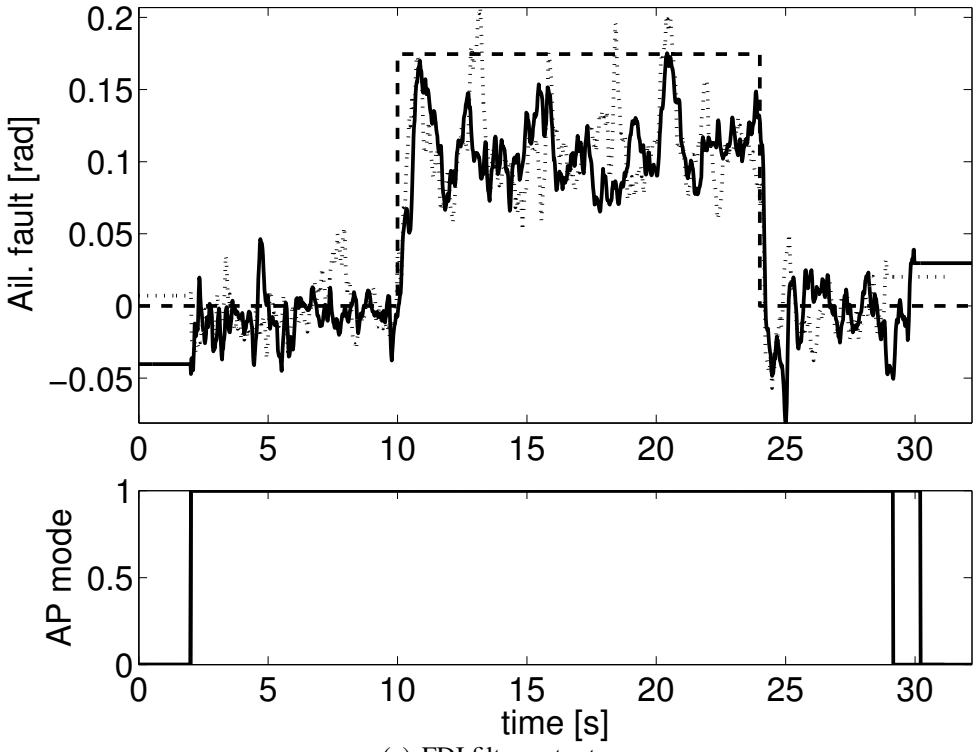
Figure 7.13: Experiment 1: flight data analysis

7.6.2.2 Experiment 2: Aileron fault 10 degrees, pulse fault, one doublet

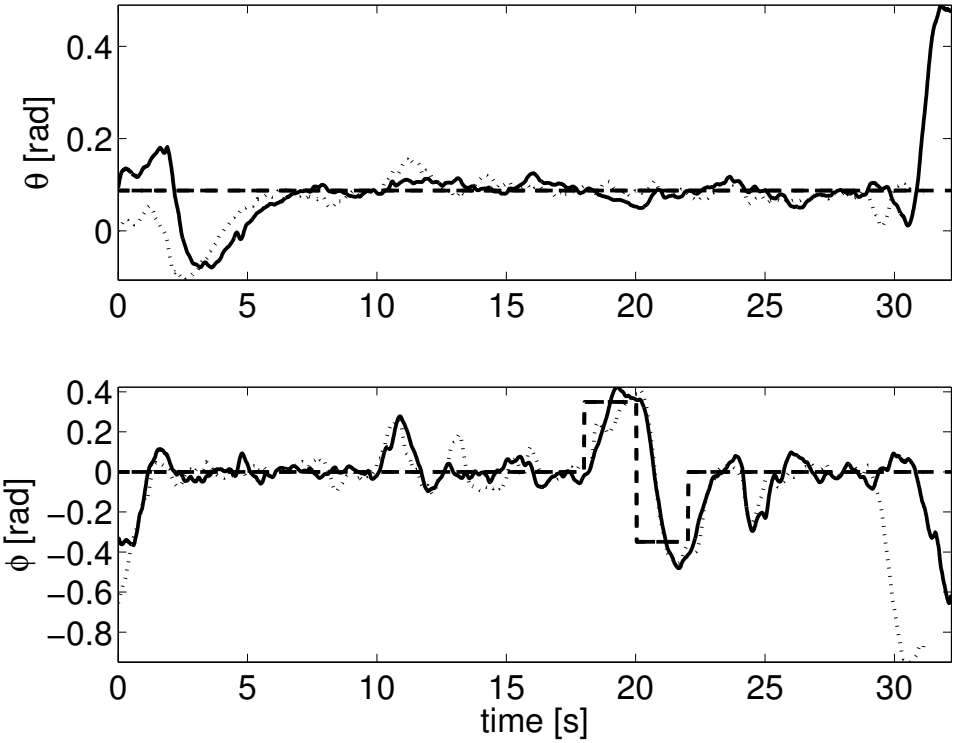
This experiment consists of a pitch command reference input of 5 deg at t=2 second and roll command reference of 0 deg (wing-level) is applied to the respective controllers. A partial hardover fault of 10 deg is injected in right aileron channel (Fig. 7.4(b)). This causes the right aileron to lock in 10 degree down deflection position for the duration of the fault. Roll-controller aileron demands are effective only through the left aileron. All other control surfaces, elevator, rudder and throttle operate normally. This aileron fault is injected at t=10 seconds. At t=18 seconds a roll-command doublet of 20 deg amplitude is applied lasting for 4 seconds. The fault is still in effect during the doublet sequence, hence only left aileron is available for control. At t=24s, aileron fault is removed and the right aileron resumes normal operation. Each experiment is repeated twice to verify the results are repeatable. Results from the two runs are overlayed in each plot in Fig. 7.14 and 7.15.

The discrete variable *AP mode* plotted in Figs. 7.15 and 7.14 shows the state of autopilot or the FCC. A value of one indicates autopilot is engaged, while zero indicates ground pilot is in control. Fig. 7.14 shows the system response in the case of optimal LQ roll-controller and FDI filter, whereas Fig. 7.15 shows flight parameters for classical roll-controller and FDI filter experiment. In Fig. 7.15(b) and 7.14(b), the solid line and dotted lines represent actual aircraft attitude response, roll (ϕ) and pitch (θ). The broken lines indicate the reference commands to the respective controllers. Comparing the plots one can see effect of aileron fault as it kicks in at $t = 10$ seconds. The performance of the classical controller is affected significantly and in fact it cannot track the roll-command reference, optimal controller also suffers a transient due to the fault, but is quickly able to restore reference tracking. This is due to the fact that optimal controller is multivariable and uses both aileron and rudder for reference tracking. The classical controller only uses rudder for dutch-roll damping (see section 7.4). This is also evident from aileron and rudder commands in Fig. 7.14(c), that there is a significant jump in both the channels at $t_0 + 8$ seconds as fault kicks in. But in the case of the classical controller 7.15(c), one can see that only the aileron channel is trying to compensate for the fault.

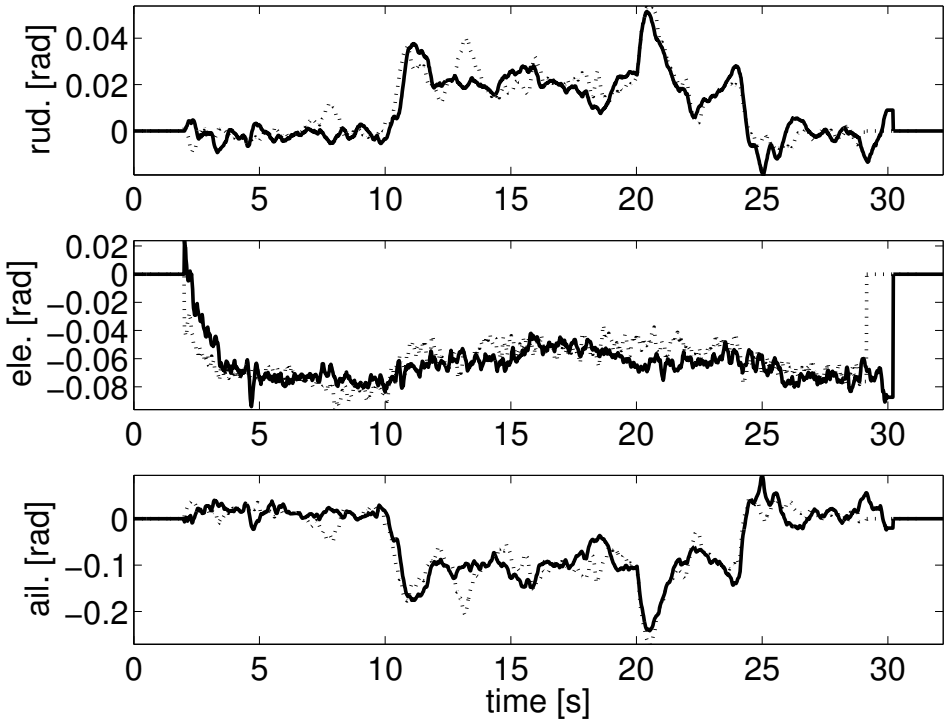
Comparing doublet roll-command tracking of the controllers at $t = 18$ seconds, one can observe that the optimal controller is able to perform the tracking task very well, but the classical controller is barely able to do so. In conclusion, its safe to say that the tracking performance of the optimal LQ controller is significantly better than that of the classical controller.



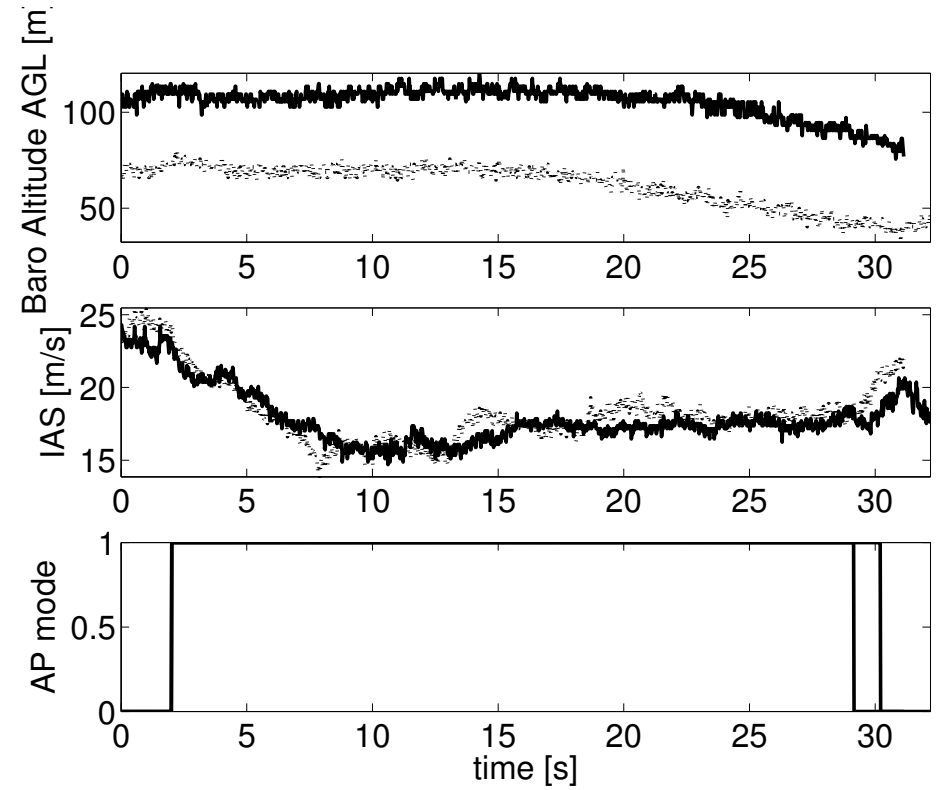
(a) FDI filter output



(b) Aircraft attitude

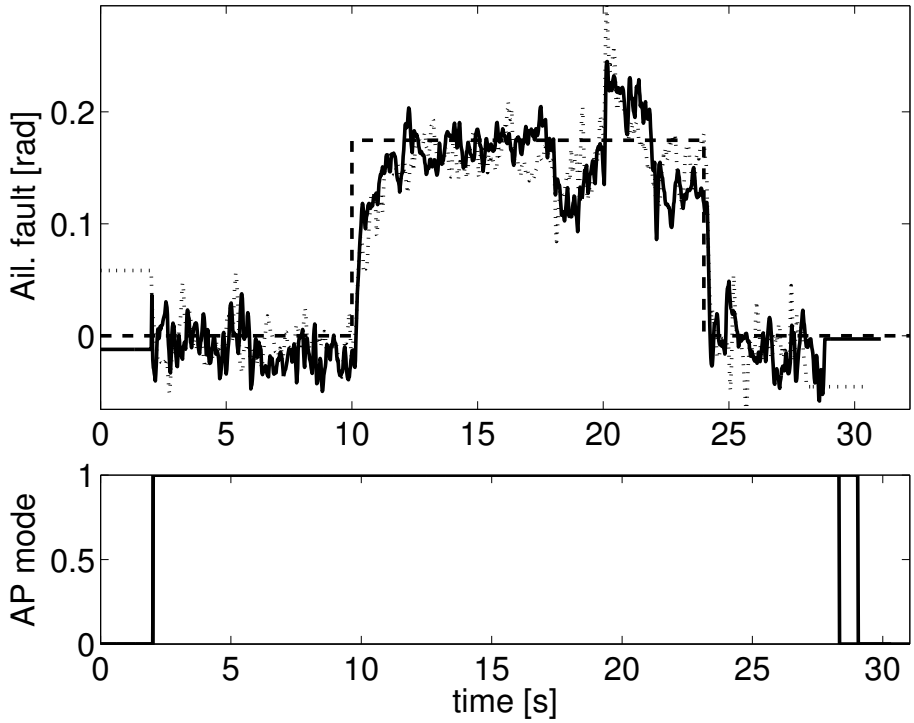


(c) Control surface commands

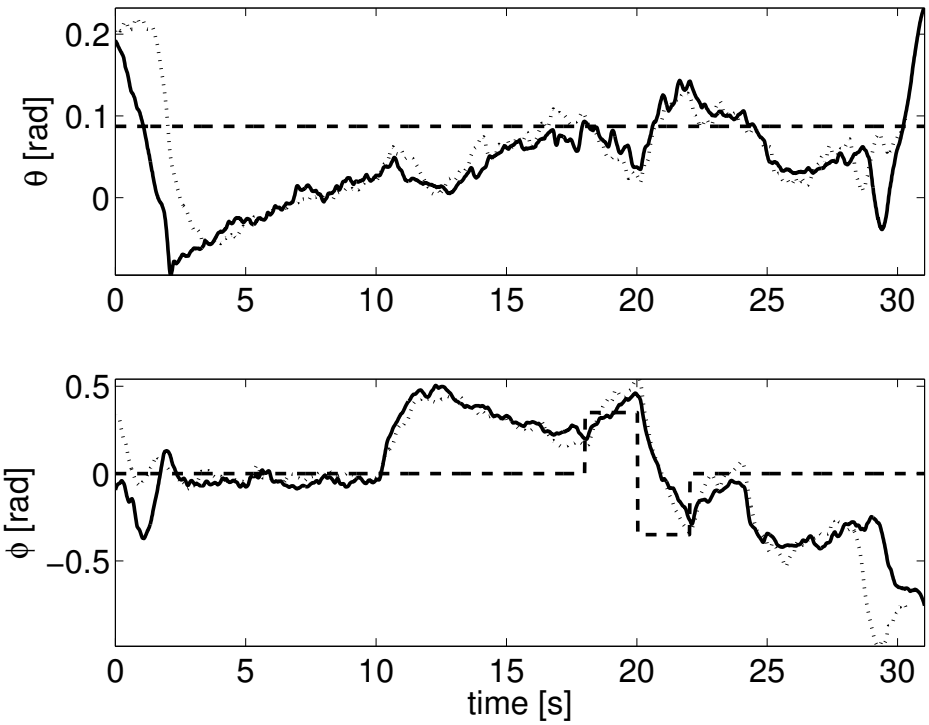


(d) Altitude and airspeed

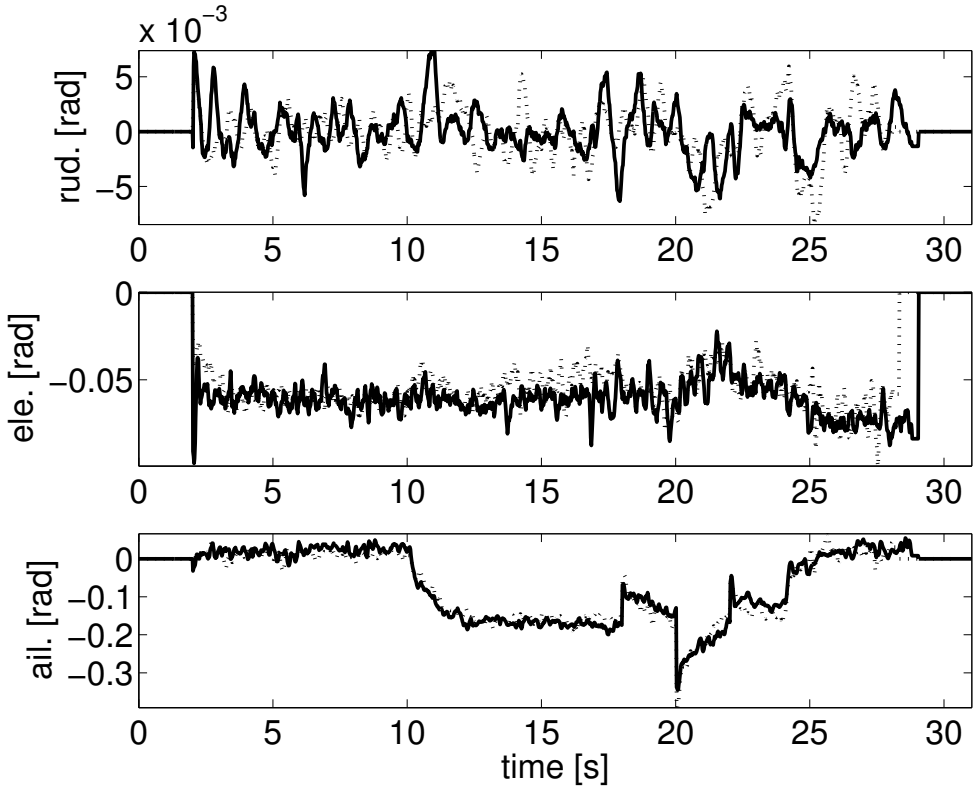
Figure 7.14: Experiment 2: FDI filter performance with roll-command tracking optimal controller LQ in the loop



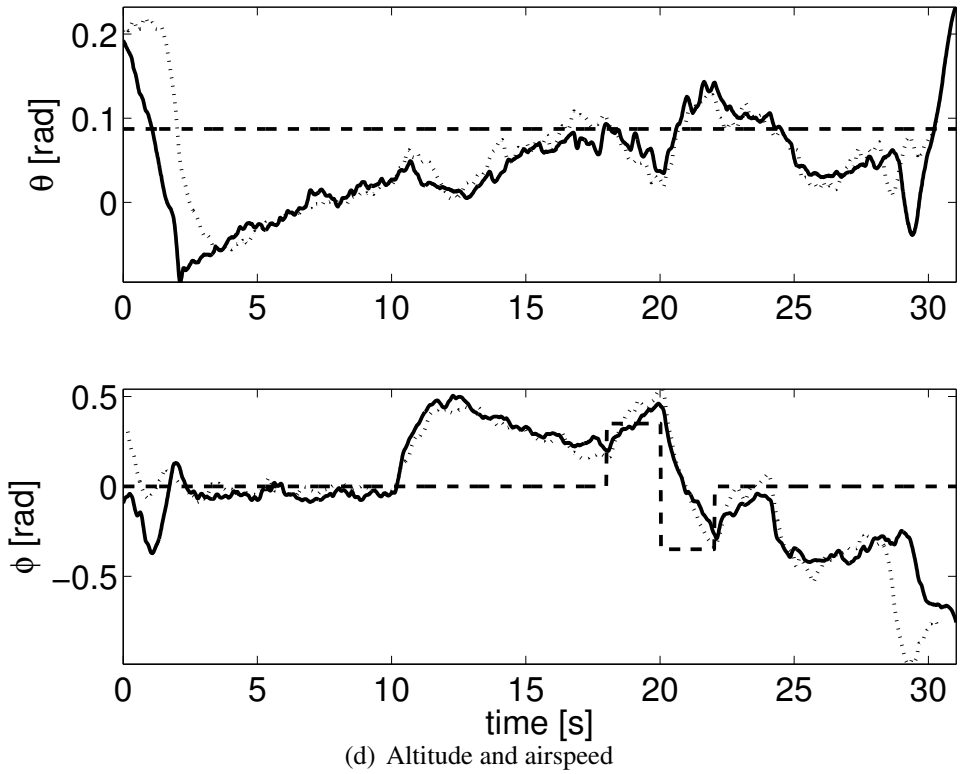
(a) FDI filter output



(b) Aircraft attitude



(c) Control surface commands



(d) Altitude and airspeed

Figure 7.15: Experiment 2: FDI filter performance with PID controller

The FDI filter performance in these two experiments can be evaluated from Figs. 7.14(a) and 7.15(a). The filter is found to be capable of detecting aileron fault with rise-time less than one second in both cases. The filter output remains close to zero in the absence of fault, which is an essential requirement. It is observed that there is a small amount of coupling between command reference to filter output at $t = 18$, which should ideally be zero. Similar behavior was observed in Experiment 1, and can be explained by the same factors.

Following conclusions can be drawn from the experiments:

- The FDI filter is able to detect and estimate aileron fault quite well with short detection time of around 1 second. This is a desirable feature, since earliest positive detection would allow safe control law reconfiguration (or adaptation) to take place before the aircraft has time to depart significantly from its operating condition.
- Filter output returns to zero when the fault is removed from the system. This is essential since one would like the filter output to be non-zero only when a fault is present, zero otherwise. This is important to help minimize the false alarm rate.
- Simulation results one would expect the controller to have no influence on FDI filter performance, which is true in the nominal case. But in the off-design case, controller affects FDI performance in a significant way. This was shown through linear simulation as well as flight experiments.
- There is a good qualitative correlation between flight results and off-design simulation results.
- Open-loop FDI filter design is not desirable for aerospace fault diagnosis problem because filter performance is very sensitive to uncertainties in the plant model, variability in operating flight condition, etc.

It would be interesting to see if the FDI performance could improve in the case integrated controller-filter design methodology was employed, integrated controller-filter explicitly takes into account the closed-loop dynamics as well as coupling arising due to uncertain dynamics [76]. This is beyond the scope of the current work.

7.7 Conclusion

An $\mathcal{H}_\infty$ model-matching FDI filter design was presented for aileron fault detection and estimation for a small UAV test-bed. The filter was implemented in real-time embedded software and flight tested. The flight test was used to validate successful real-time fault detection capability and we believe is a significant step towards realizing full benefits of robust model-based fault diagnosis techniques, especially for low-cost small UAVs. Flight experiments were conducted to compare the effect of different closed-loop controllers on the FDI filter performance.

Model-based fault diagnosis filters can prove valuable from flight safety perspective. Some of the aircraft accident case-studies presented in Chapter 1 could benefit from an FDI filter. For example in the accident involving XL Airways Airbus A320, failed state of frozen AOA sensors missed by the hardware redundancy elements. A fault diagnosis filter works differently by comparing "expected" input-output dynamical properties of the system, hence can detect even "common mode" failures in redundant sensors.

Chapter 8

Conclusion and future directions

Various complementary methods to address the challenge of improving flight safety specifically from the perspective of loss of control (LOC) were presented. Aircraft examples using linear and nonlinear GTM dynamics model were dicussesed. The positive role of region-of-attraction analysis and reachable-sets analysis in dynamic flight envelope assessment was brought out through theoretical developement and application examples. On fault diagnosis aspect, metrics for closed-loop FDI filter performance were discussed and successful application to dissimilar filter examples presented. Low cost flight research and realtime software infrastructure was developed with the aim of facilitating advanced control, guidance and estimation research. A model-based fault diagnosis filter was designed and flight tested successfully demonstrating the immense value of such filters in detection of critical faults that have in the past played decisive part in commercial aircraft accidents.

Some future directions for research:

1. Higher fidelity aerodynamic data polynomial fitting could yield benefits. As of now, low order fits restrict the validity of nonlinear results to relatively small aerodynamic angle range. To realize full benefit of nonlinear analysis tools described in this thesis, high fidelity multi-dimensinal fitting is desirable.

2. There is potential and a strong need to improve scalability of nonlinear sum-of-squares analysis methods. As was shown, the SDP problem dimensions grows rapidly with the complexity of the polynomials. In order to exploit full benefit of high fidelity polynomial models, the SOS tools as well as SDP solvers deserve attention.

3. Currently, SOS methods proposed in this thesis do not allow analysis of systems with hard nonlinearities like saturation and rate-limits. It is well known that all actuators including ones found of aircrafts have finite limits. There is some exciting research currently being done in the area of integral quadratic constraints (IQCs) that tries to address these challenges.

4. Need to examine optimal FDI filter synthesis strategies like, the integrated filter-controller synthesis, closed-loop filter synthesis, etc, to exploit full knowledge of the system and controller dynamics.

Bibliography

[1] http://www.flightglobal.com/articles/2008/10/30/318137/loss-of-control-becomes-top-killer-type-of-airliner.html.

[2] eCos: A free realtime operating system. http://ecos.sourceware.org/about.html.

[3] Loss of control joint safety implementation team report 2003. http://www.icao.int/fsix/cast/JSIT%20Loss%20Of%20Control.pdf.

[4] ROA estimation, SOS tools, etc. http://jagger.me.berkeley.edu/software/acc09/.

[5] Statistical summary of commercial jet airplane accidents. http://www.boeing.com/news/techissues/statsum.pdf.

[6] http://www.boeing.com/commercial/aeromagazine/aero_08/erroneous_story.html, 1999.

[7] NASA upset recovery research program. http://www.etcupsetrecovery.com/research.php, 2009.

[8] Zolghadri A., F. Castang, and D. Henry. Design of robust fault detection filters for multivariable feedback systems. *International journal of modelling and simulation, ACTA Press*, 26(1), 2006.

[9] Gary J. Balas, Andrew K. Packard, and Peter J. Seiler. Uncertain model set calculation from frequency domain data. In Paul M.J. Hof, Carsten Scherer, and Peter S.C. Heuberger, editors, *Model-Based Control:*, pages 89–105. Springer US, 2009.

[10] F. Bateman, N. Hassan, and M. Ouladsine. Actuators fault diagnosis and tolerant control for an unmanned aerial vehicle. In *16th International Conference of Control Applications*, pages 1061–1066, Singapore, 1-3 October, 2007.

[11] Alexandre M. Bayen, Ian M. Mitchell, Meeko M.K. Oishi, and Claire J. Tomlin. Aircraft autolander safety analysis through optimal control-based reach set computation. *Journal of Guidance, Control and Dynamics*, 30(1), 2007.

[12] J. Bokor and G. Balas. Detection filter design for LPV systems - a geometric approach. *Automatica*, 40:511–518, 2004.

[13] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*, volume 15 of *Studies in Applied Mathematics*. SIAM, Philadelphia, PA, June 1994.

[14] D. Briere and P. Traverse. AIRBUS A320/A330/A340 electrical flight controls-a family of fault-tolerant systems. In *The 23rd international symposium on fault-tolerant computing*, pages 616–623, Toulouse, France, 1993.

[15] J.W. Brown and R.V. Churchill. *Complex Variables and Applications*. McGraw-Hill, New York, NY, 8th edition, 2009.

[16] Civil Aviation Authority (CAA). Safety plan 2006/07 - 2010/11, 2006.

[17] J. Chen and R. J. Patton. *Robust model-based fault diagnosis for dynamic systems*. Kluwer Academic Publishers, 1999.

[18] H.-D. Chiang and J.S. Thorp. Stability regions of nonlinear dynamical systems: A constructive methodology. *IEEE Transactions on Automatic Control*, 34(12):1229–1241, 1989.

[19] D. Cox et al. NASA GTM DesignSim. Technical report, NASA Langley Research Center, Langley, VA, 2009.

[20] Dave Cox. The GTM DesignSim v0905. Technical report, NASA Langley, VA, 2008.

[21] Elizabeth Ann Cross and Ian M. Mitchell. Level set methods for computing reachable sets of systems with differential algebraic equation dynamics. *American Control Conference, Seattle, Washington, USA*, 2008.

[22] E.J. Davison and E.M. Kurak. A computational method for determining quadratic Lyapunov functions for nonlinear systems. *Automatica*, 7:627–636, 1971.

[23] Steven X. Ding. *Model-based Fault Diagnosis Techniques*. Springer, 2008.

[24] John Doyle, Bruce Francis, and Allen Tannenbaum. *Feedback Control Theory*. Macmillan Publishing Co., 1990.

[25] A. Edelmayer, J. Bokor, and Z. Szabo. Inversion-based residual generation for robust detection and isolation of faults by means of estimation of the inverse dynamics in linear dynamical systems. *International Journal of Control*, 82(8):1526–1538, 2009.

[26] Andrés Marcos Esteban. A linear parameter varying model of the Boeing 747-100/200 longitudinal motion. Master's thesis, University of Minnesota, Minneapolis, 2001.

[27] F.Zheng, Q-G Wang, and T.H.Lee. A parameter optimization approach to solving quasi-lmi problems. In *Proc. 40th IEEE Conf. on Decision and Control*, pages 3607–3612, 2001.

[28] P. Gahinet and P. Apkarian. A linear matrix inequality approach to $\mathcal{H}_\infty$ control. *Int. J. Robust Nonlinear Control*, 1994.

[29] G.Balas, R.Chiang, A.Packard, and M.Safonov. Robust Control Toolbox. In *The Mathworks, Inc.*, Natick, MA.

[30] R. Genesio, M. Tartaglia, and A. Vicino. On the estimation of asymptotic stability regions: State of the art and new proposals. *IEEE Transactions on Automatic Control*, 30(8):747–755, 1985.

[31] L. El Ghaoui, R. Nikoukhah, and F. Delebecque. LMITOOL: a package for lmi optimization. In *In Proceedings of the IEEE Conference on Decision and Control*, pages 3096–3101, 1995.

[32] O. Hachicho and B. Tibken. Estimating domains of attraction of a class of nonlinear dynamical systems with LMI methods based on the theory of moments. In *Proceedings of the IEEE Conference on Decision and Control*, pages 3150–3155, 2002.

[33] J. Hauser and M.C. Lai. Estimating quadratic stability domains by nonsmooth optimization. In *Proceedings of the American Control Conference*, pages 571–576, 1992.

[34] Yutaka Ikeda, James Ramsey, Eugene Lavretsky, and Patrick McCormick. Robust adaptive control of UCAVs: Final report. Technical Report BOEING-STL-2004P0037, The Boeing Company, September 2004.

[35] Petros A. Ioannou and Jing Sun. *Robust adaptive control*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1995.

[36] R. Isermann. Process fault detection based on modeling and estimation methodsa survey. *Automatica*, 20(4):387–404, 1984.

[37] Rolf Isermann. Model-based fault-detection and diagnosis-status and applications. *Annual Reviews in Control*, 29:71–85, 2005.

[38] C. A. Jacobson, C. N. Nett, and A. T. Miller. An integrated approach to controls and diagnostics: The 4-Parameter controller. *Proc. American Control Conference*, pages 824–835, 1988.

[39] Emmanuel Mazarsand Imad M. Jaimoukha and Zhenhai Li. Computation of a reference model for robust fault detection and isolation residual generation. *Journal of Control Science and Engineering*, 2008:1–12, 2008.

[40] Z. Jarvis-Wloszek. *Lyapunov Based Analysis and Controller Synthesis for Polynomial Systems using Sum-of-Squares Optimization*. PhD thesis, University of California, Berkeley, 2003.

[41] Z. Jarvis-Wloszek, R. Feeley, W. Tan, K. Sun, and A. Packard. Some controls applications of sum of squares programming. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, volume 5, pages 4676–4681, 2003.

[42] Z. Jarvis-Wloszek, R Feeley, W. Tan, K. Sun, and A. Packard. *Positive Polynomials in Control*, volume 312 of *Lecture Notes in Control and Information Sciences*, chapter Controls Applications of Sum of Squares Programming, pages 3–22. Springer-Verlag, 2005.

[43] J.Bokor, A.Edelmayer, and L.Keviczky. An $\mathcal{H}_\infty$ filtering approach to robust detection of failures in dynamical systems. *Proc. 33rd IEEE Conf. on Decision and Control*, 3:3037–3039, 1994.

[44] J.C.Doyle, K.Glover, P.P.Khargonekar, and B.A.Francis. State-space solutions to standard H_2 and H_∞ control problems. *IEEE Trans. Automatic Control*, 34(8), 1989.

[45] T. Jordan and R. Bailey. NASA langley's AirSTAR testbed: A subscale flight test capability for flight dynamics and control system experiments. *AIAA Guidance, Navigation and Control Conference and Exhibit, Honolulu, Hawaii*, Aug. 18-21, 2008.

[46] T.L. Jordan, W.M. Langford, and J.S. Hill. Airborne subscale transport aircraft research testbed aircraft model development. In *AIAA 2005-6432, AIAA Guidance, Navigation, and Control Conference and Exhibit, San Francisco, CA*, 2005.

[47] J.Stoustrup and H.Niemann. Fault estimation-a standard problem approach. *International Journal of Robust and Nonlinear Control*, 12(8), 2002.

[48] Thomas Kailath. *Linear Systems*. Prentice Hall Information and System Sciences Series, 1980.

[49] Tamas Keviczky, Ryan Ingvalson, Hector Rotstein, Oreste Riccardo Natale, Gary Balas, and Andrew Packard. An integrated multi-layer approach to software-enabled control: Mission planning to vehicle control. Technical report, Department of AEM, University of Minnesota, Minneapolis, MN, 2004.

[50] Hassan Khalil. *Nonlinear systems*. Prentice-Hall, Inc., 3nd edition, 2001.

[51] H.K. Khalil. *Nonlinear Systems*. Prentice Hall, 3rd edition, 2002.

[52] J. Lfberg. YALMIP : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

[53] A. Magretski and A. Rantzer. System analysis via integral quadratic constraints. *IEEE Trans. Automatic Control*, 42:819–830, 1997.

[54] M.A.Massoumnia. A geometric approach to the synthesis of failure detection filters. *IEEE Trans. Automatic Control*, AC-31(9):839–846, 1986.

[55] A. Marcos and G. J. Balas. A Boeing $747-100/200$ aircraft fault tolerant and fault diagnostic benchmark. Technical report, Department of AEM, University of Minnesota, Minneapolis, MN, 2003.

[56] A. Marcos, S. Ganguli, and G.J. Balas. An application of $\mathcal{H}_\infty$ fault detection and isolation to a transport aircraft. *Control Engineering Practice*, 13(1):105–119, 2005.

[57] M. A. Massoumnia, G. C. Verghese, and A. S. Willsky. Failure detection and identification. *IEEE Transactions on Automatic Control*, 34:316–321, 1989.

[58] Austin Murch and John Foster. Recent NASA research on aerodynamic modeling of post-stall and spin dynamics of large transport airplanes. In *45th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, Jan. 8-11*, 2007.

[59] H. Niemann and J. Stourstrup. Integration of control and fault detection: Nominal and robust design. *IFAC SAFEproceeESS, Hull, UK*, 1997.

[60] D. Bruce Owens, David E. Cox, and Eugene A. Morelli. Development of a low-cost sub-scale aircraft for flight research: The faser project.

[61] A. Packard, G. Balas, R. Liu, and J. Y. Shin. Results on worst-case performance assessment. In *Proc. American Control Conference*, pages 2425–2427, Piscataway, NJ, 2000.

[62] R. Pandita, A. Chakraborty, P.J. Seiler, and G.J. Balas. Reachability and region of attraction analysis applied to GTM dynamic flight envelope assessment. In *AIAA Guidance, Navigation and Control Conference, Chicago, Illinois*, 10-13 August, 2009.

[63] Rohit Pandita. Fault diagnosis filters and controller interaction for uncertain dynamical systems. Master's thesis, University of Minnesota, 2008.

[64] P. Parrilo. *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*. PhD thesis, California Institute of Technology, 2000.

[65] Yew Chai Paw. *Synthesis and validation of flight control for UAV*. PhD thesis, University of Minnesota, Minneapolis, 2009.

[66] P.M.Frank. Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy–a survey and some new results. *Automatica*, 26(3):459–474, 1990.

[67] S. Prajna, A. Papachristodoulou, P. Seiler, and P. A. Parrilo. *SOSTOOLS: Sum of squares optimization toolbox for MATLAB*, 2004.

[68] Stephen Prajna, Antonis Papachristodoulou, Peter Seiler, and Pablo A. Parrilo. SOSTOOLS: Sum of squares optimization toolbox for MATLAB, 2004.

[69] C. Rago, R. Prasanth, R.K. Mehra, and R. Fortenbaugh. Failure detection and identification and fault tolerant control using the IMM-KF with applications to the Eagle-Eye UAV. In *Proc. 37th IEEE Conf. on Decision and Control*, pages 4208–4213, Tampa, Florida, December, 1998.

[70] R.Pandita and G.J.Balas. A metric for FDI filter and controller interaction in uncertain dynamical systems. In *IEEE Multi-Systems Conference, San Antonio, TX*, San Antonio, TX, 2008. IEEE Multi-Systems Conference.

[71] J.S. Shamma and J.R. Cloutier. Gain-scheduled missile autopilot design using linear parameter varying transformations. *Journal of Guidance, Control and Dynamics*, 16(2):256–263, 1993.

[72] Jaiwon Shin. The NASA aviation safety program: Overview. Technical Report NASA-TM-2000-209810, NASA Glenn Research Center, March 2000.

[73] Jong-Yeob Shin. Analysis of linear parameter varying system models based on reachable sets. CR 211231, ICASE, NASA Langley Research Center, 2001.

[74] Robert F. Stengel. *Flight Dynamics*. Princeton University Press, Princeton, NJ, 2004.

[75] B.L. Stevens and F.L. Lewis. *Aircraft Control and Simulation*. Wiley-Interscience, 2nd edition, 2003.

[76] Jakob Stoustrup, M.J.Grimble, and Henrik Niemann. Design of integrated systems for the control and detection of actuator/sensor faults. *Sensor Review*, 17(2):138–149, 1997.

[77] Jos F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones, 1998.

[78] Jos F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, pages 625–653, 1999.

[79] W. Tan. *Nonlinear Control Analysis and Synthesis using Sum-of-Squares Programming*. PhD thesis, University of California, Berkeley, 2006.

[80] W. Tan and A. Packard. Searching for control Lyapunov functions using sums of squares programming. In *42nd Annual Allerton Conference on Communications, Control and Computing*, pages 210–219, 2004.

[81] Weehong Tan. Applications of linear parameter-varying control theory. Master's thesis, University of California, Berkeley, 1997.

[82] Weehong Tan, Ufuk Topcu, Peter Seiler, Gary Balas, and Andrew Packard. Simulation-aided reachability and local gain analysis for nonlinear dynamical systems. In *Proc. 47rd IEEE Conf. on Decision and Control*, Cancun, Mexico, 2008.

[83] B. Tibken. Estimation of the domain of attraction for polynomial systems via LMIs. In *Proceedings of the IEEE Conference on Decision and Control*, pages 3860–3864, 2000.

[84] B. Tibken and Y. Fan. Computing the domain of attraction for polynomial systems via BMI optimization methods. In *Proceedings of the American Control Conference*, pages 117–122, 2006.

[85] Jorge E. Tierno, Richard M. Murray, and John C. Doyle. An efficient algorithm for peformance analysis of nonlinear control systems. In *Proc. American Control Conference*, Seattle, WA, 1995.

[86] K. C. Toh, M.J. Todd, R. H. Ttnc, and R. H. Tutuncu. Sdpt3 - a MATLAB software package for semidefinite programming. *Optimization Methods and Software*, 11:545–581, 1999.

[87] U. Topcu. *Quantitative Local Analysis of Nonlinear Systems*. PhD thesis, University of California, Berkeley, 2008.

[88] U. Topcu, A. Packard, and P. Seiler. Local stability analysis using simulations and sum-of-squares programming. *Automatica*, 44(10):2669–2675, 2008.

[89] U. Topcu, A. Packard, P. Seiler, and T. Wheeler. Stability region analysis using simulations and sum-of-squares programming. In *Proceedings of the American Control Conference*, pages 6009–6014, 2007.

[90] Lieven Vandenberghe, Stephen Boyd, and Shao po Wu. Determinant maximization with linear matrix inequality constraints. *SIAM Journal on Matrix Analysis and Applications*, 19:499–533, 1998.

[91] A. Vannelli and M. Vidyasagar. Maximal lyapunov functions and domains of attraction for autonomous nonlinear systems. *Automatica*, 21(1):69–80, 1985.

[92] A. Varga. On computing least order fault detectors using rational nullspace bases. In *In Proceedings of the IFAC Symp. SAFEPROCESS'2003, Washington D.C.*, 2003.

[93] A. Varga. General computational approach for optimal fault detection. In *SAFEPROCESS '09, Barcelona, Spain*, 2009.

[94] M. Vidyasagar. *Nonlinear systems analysis*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2nd edition, 1992.

[95] M. Vidyasagar. *Nonlinear Systems Analysis*. Prentice Hall, 2nd edition, 1993.

[96] Kevin A. Wise and Farhad Deylami. Approximating a linear quadratic missile autopilot design using an output feedback projective control. In *AIAA Guidance, Navigation and Control Conference, New Orleans, LA*, pages 114–122, August 1991.

[97] Y. C. Yeh. Triple-triple redundant 777 primary flight computer. *IEEE*, 1996.

Appendix A

Polynomial modeling

A.1 Polynomial approximations of aerodynamic LUT data

Coefficient	Polynomial approximation
$C_x(\alpha)$	$a_{11}\alpha + a_{01}$
	$= 0.001302\varepsilon\alpha + 0.002515$
$C_z(\alpha)$	$a_{22}\alpha^2 + a_{12}\alpha + a_{02}$
	$= 0.002731\varepsilon^2\alpha^2 - 0.109\varepsilon\alpha - 0.005959$
$C_m(\alpha)$	$a_{13}\alpha + a_{03}$
	$= -0.02511\varepsilon\alpha + 0.1506$
$C_x(\alpha, \hat{q})$	$C_{x_q} q$
	$= 0.0q$
$C_z(\alpha, \hat{q})$	$C_{z_q} q$
	$= -24\frac{\bar{c}}{2U} q$
$C_m(\alpha, \hat{q})$	$C_{m_q} q$
	$= -45\frac{\bar{c}}{2U} q$
$C_x(\alpha, \delta_e)$	$C_{x_{\delta_e}} \delta_e$
	$= 0.0\varepsilon\delta_e$
$C_z(\alpha, \delta_e)$	$C_{z_{\delta_e}} \delta_e$
	$= -0.0083\varepsilon\delta_e$
$C_m(\alpha, \delta_e)$	$C_{m_{\delta_e}} \delta_e$
	$= -0.032\varepsilon\delta_e$

A.2 Polynomial longitudinal EOMs for the NASA GTM aircraft

$$
\begin{aligned}
\dot{\alpha} = {} & 0.040619\alpha^3\delta_{th}^2 - 2.8888\times10^{-4}\alpha^3\delta_{th}U + 3.4371\times10^{-4}\alpha^3U^2 + 7.8376\times10^{-4}\alpha^3U\theta \\
& + 2.2917\times10^{-5}\alpha^2\delta_eU^2 + 3.1481\times10^{-5}\alpha^2\delta_{th}^2U - 7.2895\times10^{-8}\alpha^2\delta_{th}U^2 \\
& - 5.8362\times10^{-6}\alpha^2U^3 + 3.4823\times10^{-6}\alpha^2U^2q - 0.0011756\alpha^2U\theta^2 \\
& - 3.7716\times10^{-6}\alpha\delta_{th}^2U^2 - 9.3472\times10^{-9}\alpha U^4 + 7.8376\times10^{-4}\alpha U\theta^3 \\
& - 7.093\times10^{-10}\delta_eU^4 - 1.0778\times10^{-10}U^4q + 0.041901\alpha^3\delta_{th} - 3.283\times10^{-5}\alpha^3U \\
& - 0.11368\alpha^3\theta - 0.0045663\alpha^2\delta_{th}^2 + 3.2475\times10^{-5}\alpha^2\delta_{th}U + 8.4125\times10^{-4}\alpha^2U^2 \\
& + 0.17053\alpha^2\theta^2 + 0.0016802\alpha\delta_{th}^2U - 3.8906\times10^{-6}\alpha\delta_{th}U^2 + 4.1641\times10^{-6}\alpha U^3 \\
& + 1.0556\times10^{-5}\alpha U^2\theta - 0.11368\alpha\theta^3 + 3.1599\times10^{-7}\delta_eU^3 + 1.4133\times10^{-7}\delta_{th}^2U^2 \\
& - 6.7118\times10^{-12}U^4 + 4.8015\times10^{-8}U^3q - 5.2779\times10^{-6}U^2\theta^2 + 0.004762\alpha^3 \\
& - 0.0047104\alpha^2\delta_{th} + 0.002355\alpha^2U - 0.24372\alpha\delta_{th}^2 + 0.0017333\alpha\delta_{th}U \\
& - 6.0445\times10^{-4}\alpha U^2 - 0.0047026\alpha U\theta - 4.5834\times10^{-5}\delta U^2 - 6.2962\times10^{-5}\delta_{th}^2U \\
& + 1.4579\times10^{-7}\delta_{th}U^2 + 2.9901\times10^{-9}U^3 - 6.9645\times10^{-6}U^2q + 0.0023513U\theta^2 \\
& + 0.6821\alpha\theta + 0.0091326\delta_{th}^2 - 0.34159\alpha^2 - 0.25141\alpha\delta_{th} + 1.9698\times10^{-4}\alpha U \\
& - 6.4949\delta_{th}U + 1.0139\times10^{-5}U^2 - 0.34105\theta^2 - 0.028572\alpha + 0.0094208\delta_{th} \\
& - 0.0047099U + q + 0.68317
\end{aligned}
$$

$$
\begin{aligned}
\dot{U} = {} & -0.071796\alpha^3\delta_{th}^2 + 0.032482\alpha^3U^2 + 2.6812\alpha^3\theta^2 - 2.6812\alpha^2\theta^3 - 0.074062\alpha^3\delta_{th} \\
& - 5.7479\alpha^2\delta_{th}^2 - 0.034453\alpha^2eas^2 - 0.002162\alpha\delta_eU^2 - 3.2851\times10^{-4}\alpha U^2q \\
& - 5.3708\alpha^3 - 5.9293\alpha^2\delta_{th} + 16.087\alpha^2\theta + 0.43077\alpha\delta_{th}^2 + 0.0033397\alpha U^2 \\
& - 16.087\alpha\theta^2 + 5.3623\theta^3 - 0.67386\alpha^2 + 0.44437\alpha\delta_{th} + 11.4958\delta_{th}^2 \\
& - 1.8821\times10^{-4}U^2 + 32.2246\alpha + 11.8586\delta_{th} - 32.174\theta + 1.3477
\end{aligned}
$$

$$
\begin{aligned}
\dot{q} = {} & -0.030927\alpha^3U^2 + 0.01089\alpha^2U^2 - 0.003\alpha U^2 - 0.002765\delta_eU^2 - 2.0431\times10^{-4}U^2q \\
& + 1.2398\delta_{th}^2 + 2.3418\times10^{-4}U^2 + 1.2789\delta_{th} + 0.14535 \\
\dot{\theta} = {} & q
\end{aligned}
\tag{A.1}
$$

Appendix B

FDI metrics applications: Design data

B.1 NASA GTM linear longitudinal model

$$A = \begin{bmatrix} -0.0433 & 13.3181 & -0.1955 & -32.1740 \\ -0.0028 & -3.2087 & 0.9509 & 0 \\ -0.0054 & -51.7656 & -4.0931 & 0 \\ 0 & 0 & 1.0000 & 0 \end{bmatrix}, B = \begin{bmatrix} -2.5772 \\ -0.3061 \\ -57.2326 \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, D = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$W_u(s) = 0.2\frac{(s+15.64)}{(s+62.57)}$$

B.2 Geometric FDI Filter

B.2.1 Pitch-rate gyro FDI filter

$$N_s = \begin{bmatrix} -0.5000 & 7.8184 & 0.0860 \\ -7.8184 & -0.5000 & 16.0705 \\ -0.0860 & -16.0705 & -0.5000 \end{bmatrix}, G_s = \begin{bmatrix} -7.8052 & 2.4754 & -0.6183 & -0.0860 \\ 0.4983 & 7.8024 & 0.0112 & -16.0705 \\ 16.0545 & -0.0039 & -1.7229 & -0.5000 \end{bmatrix}$$

$$F_s = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, H_s = \begin{bmatrix} -0.0053 & 1.0 & -0.0051 & 0.0 \\ -0.9990 & -0.0051 & 0.0450 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}, M_s = \begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$$

B.2.2 Elevator FDI filter

$$N_a = \begin{bmatrix} -0.5000 & 8.8880 \\ -8.8880 & -0.5000 \end{bmatrix}, G_a = \begin{bmatrix} -2.8928 & -21.3141 & 0.0 \\ 44.9149 & 8.1596 & 0.0 \end{bmatrix}, F_a = \begin{bmatrix} 2.5900 \\ 0.0655 \end{bmatrix}$$

$$H_a = \begin{bmatrix} -0.6823 & -0.5918 & 0.0 & 0.4293 \\ 0.7175 & -0.4293 & -0.0000 & 0.5485 \end{bmatrix}, M_a = \begin{bmatrix} 0.7316 & 0.6817 \\ -0.6817 & 0.7316 \end{bmatrix}$$

B.3 $\mathcal{H}_\infty$ FDI filter

The performance weights in this section and the $\mathcal{H}_\infty$ controller given in Appendix-B.4 are used for the filter synthesis. Filter state-space data is not included here because of its high state dimension. The reader may use this data to reproduce the $\mathcal{H}_\infty$ FDI filter.

$$W_a(s) = \frac{0.6981}{0.1s+1}, W_s(s) = \frac{0.08726}{2s+1}, W_{p_f}(s) = \begin{bmatrix} \frac{0.01s+0.1}{s+0.001} & 0 \\ 0 & \frac{0.01s+5}{s+0.05} \end{bmatrix}$$

B.4 $\mathcal{H}_\infty$ Pitch Controller

$$A = \begin{bmatrix} -0.609 & 14.25 & -23.73 & -31.29 & 0 & -1.208 & 6.834 & -0.0913 & 19.19 & -1.695 & -0.755 \\ -0.0084 & -3.125 & -2.741 & -0.187 & 0 & -0.1435 & 0.8118 & 0.0116 & 3.013 & 0.0373 & -0.0897 \\ -1.009 & -34.28 & -526.3 & -28.97 & 0 & -26.82 & 151.8 & 0.6958 & 425.8 & 2.06 & -16.77 \\ 0.0074 & -0.0195 & 0.0550 & -0.1153 & 0 & 0 & 0 & 0.0160 & 0.7737 & 0.0944 & 0 \\ 0 & 0 & 0 & 0 & -5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.016 & 0 & 0 & 0 & 0 & 0 \\ 0.2774 & 0.2317 & 152.6 & 0.0377 & 0 & 0 & -62.57 & -0.1907 & -124.9 & -0.0308 & 0 \\ -0.0272 & -25.72 & 0.1809 & 0.406 & 0 & 0 & 0 & -1.052 & -0.1481 & -0.3324 & 0 \\ 0.00747 & 0.295 & -1.463 & 1.069 & 0 & 0 & 0 & -0.2428 & -21.02 & -0.8749 & 0 \\ -0.0992 & 0.0974 & 0.0521 & -23.86 & 0 & 0 & 0 & -0.0802 & -0.0427 & -2.685 & 0 \\ 0.0158 & -5.111 & 0.6075 & 8.067 & 0 & 7.499 & 1.414 & -0.0119 & -0.0109 & -0.55 & -5.31 \end{bmatrix}$$

$$B = \begin{bmatrix} -1.363 & 0.2739 & -56.65 & 5.266 & 0 \\ -0.0130 & -0.0334 & -8.897 & -0.0790 & 0 \\ -2.289 & -1.919 & -1257 & -0.2729 & 0 \\ 0.01787 & -0.0470 & -2.285 & -0.2788 & 0 \\ 0 & 0 & 0 & 0 & 2.417 \\ 0 & 0 & 0 & -4.835 & -4.835 \\ 0.6706 & 0.56 & 368.8 & 0.0911 & 0 \\ -0.0659 & -62.18 & 0.4373 & 0.9815 & 0 \\ 0.0180 & 0.7132 & -3.537 & 2.583 & 0 \\ -0.2398 & 0.2355 & 0.1261 & -57.68 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, C = \begin{bmatrix} 0.0004 \\ -0.1321 \\ 0.01571 \\ 0.2086 \\ 0 \\ 0.1939 \\ 0.03655 \\ -0.0003 \\ -0.0003 \\ -0.0142 \\ 0.1212 \end{bmatrix}^{\mathsf{T}}, D = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}^{\mathsf{T}}$$

Appendix C

FDI filter design and flight validation

C.1 UMN FRP lateral-directional linear model

$$A = \begin{bmatrix} 0.4928 & 1.258 & -17.17 & 9.779 \\ -2.671 & -11.02 & 2.492 & 0 \\ 0.8668 & -1.867 & -1.524 & 0 \\ 0 & 1 & 0.07398 & 0 \end{bmatrix}, B = \begin{bmatrix} -0.7251 & 2.475 \\ 155.4 & 0.4087 \\ 18.7 & 17.92 \\ 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}, D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

C.2 Classical roll-controller

$$K_P = 0.3, K_I = 0.05, K_D = 0.05$$

Yaw-damper,

$$K_{YD}(s) = 0.765 \frac{0.25s}{0.25s + 1}$$

C.3 $\mathcal{H}_\infty$ FDI filter

Weighting functions used in the filter synthesis,

$$W_a(s) = \frac{0.1745}{0.1s+1}, W_s(s) = \frac{0.08726}{2s+1}$$

$$W_{n_\phi}(s) = W_{n_r}(s) = \frac{0.002792s + 0.001745}{0.045s+1}, W_{n_p} = \frac{0.002792s + 0.001745}{0.045s+1}$$

$$W^a_{p_f}(s) = \frac{0.01s+0.1}{s+0.002501}, W^s_{p_f}(s) = \frac{0.01s+2.001}{s+0.05001}$$

$$W_{u_{\delta_a}} = W_{u_{\delta_r}} = \frac{0.2(s+15.64)}{s+62.57}$$

$$A_F = \begin{bmatrix} 0.4928 & 2.3111 & -17.0652 & 14.2497 & -4.4118 & -0.1372 & -5.9914 & 00 & 1.9871 & 6.7833 \\ -2.6705 & -101.8465 & -20.1046 & -249.4664 & 380.3669 & 30.2821 & 334.32030 & 0 & -425.8851 & 1.1200 \\ 0.8668 & -12.1682 & -4.5550 & -20.4542 & 43.1410 & 4.0619 & 27.41150 & 0 & -51.2398 & 49.1086 \\ 0 & 0.2200 & -0.0247 & -13.2760 & 3.2666 & 0.1322 & 17.79170 & 0 & 0 & 0 \\ 0 & -1.1784 & 0.5030 & 66.2111 & -17.2873 & -0.6741 & -88.73220 & 0 & 0 & 0 \\ 0 & 0.0025 & -0.0309 & 0.0193 & -0.0104 & -22.1808 & -0.02590 & 0 & 0 & 0 \\ 0 & 0.0469 & -0.0150 & -7.5105 & -0.1962 & 0.0201 & -12.15710 & 0 & 0 & 0 \\ 0 & -0.1571 & -0.0385 & -0.6273 & 0.6577 & 0.0517 & 0.8407 & -10.00260 & 0 & 0 \\ 0 & 0.1828 & 0.0258 & 2.8715 & -0.7654 & -0.0346 & -3.8482 & 0 - 200.0525 & 0 & 0 \\ 0 & -0.1698 & -0.0458 & -0.1908 & 0.7110 & 0.0614 & 0.25570 & 0 & -62.5700 & 0 \\ 0 & -0.0061 & 0.1164 & 0.0003 & 0.0254 & -0.1559 & -0.0004 & 0 & 00 & -62.5700 \end{bmatrix}$$

$$B_F = \begin{bmatrix} 0.3496 & 0.0340 & 1.4837 & -0.2407 & -0.8215 \\ -30.1417 & -7.4989 & -82.7899 & 51.5790 & -0.1356 \\ -3.4187 & -1.0059 & -6.7881 & 6.2056 & -5.9475 \\ -0.2589 & -0.0327 & -4.4059 & 0 & 0 \\ -0.3911 & 0.1669 & 21.9733 & 0 & 0 \\ 0.0008 & -0.0103 & 0.0064 & 0 & 0 \\ 0.0156 & -0.0050 & -2.4925 & 0 & 0 \\ -0.0521 & -0.0128 & -0.2082 & 0 & 0 \\ 0.0607 & 0.0086 & 0.9530 & 0 & 0 \\ -0.0563 & -0.0152 & -0.0633 & 0 & 0 \\ -0.0020 & 0.0386 & 0.0001 & 0 & 0 \end{bmatrix}$$

$$C_F = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 120.53 & 0 & 0 \\ 0 & 0 & -3.0 & 0 & 04.023 & 0 & 0 & 301.33 & 0 & 0 \end{bmatrix}$$

$$D_F = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -0.9962 & 0 & 0 & 0 \end{bmatrix}$$

C.4 Direct MRAC controller data

$$P = \begin{bmatrix} 14.4305 & 1.2613 & -0.2545 & 1.8634 & 5.1158 \\ 1.2613 & 0.3584 & 0.0998 & -0.4846 & 0.9707 \\ -0.2545 & 0.0998 & 0.1753 & -0.7929 & 0.6155 \\ 1.8634 & -0.4846 & -0.7929 & 3.7451 & -2.4154 \\ 5.1158 & 0.9707 & 0.6155 & -2.4154 & 4.8933 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 \\ -0.7251 & 2.4755 \\ 155.4201 & 0.4087 \\ 18.6991 & 17.9214 \\ 0 & 0 \end{bmatrix}$$

Adaptation rate,

$$\Gamma = 1 \times 10^{-04}$$

Projection operator,

$$\Xi = \begin{bmatrix} 1.0000 & 0 & 0 & 0 \\ -8.0047 & -0.2226 & 0.8843 & -3.9260 \\ 0 & 1.0000 & 0 & 0 \\ 0 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

C.4.1 Reference model

$$A_m = \begin{bmatrix} 0 & 0 & 0 & 0 & 1.0000 \\ 0.1798 & 0.4966 & 1.2752 & -17.2231 & 9.8817 \\ -96.8484 & -1.7945 & -14.5969 & -2.2021 & -53.0171 \\ -13.6146 & 1.0296 & -2.2912 & -2.6471 & -7.4216 \\ 0 & 0 & 1.0000 & 0.0740 & 0.0000 \end{bmatrix}, B_m = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$C_m = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, D_m = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

C.5 LQ optimal controller data

$$K_{LQ} = \begin{bmatrix} 0.6679 & 0.0243 & 0.0251 & 0.3631 \\ 0.1355 & 0.0004 & 0.0284 & 0.0710 \end{bmatrix}$$