Fault-Tolerant Flight Control
Using One Aerodynamic Control Surface

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

Raghu Venkataraman

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

Peter J. Seiler, advisor

June, 2018

Acknowledgements

I thank my advisor Prof. Peter Seiler for providing a stimulating environment for conducting research and for guiding me through graduate school. I thank him for believing in the concept of flight using a single control surface and for helping us realize it. To put it simply, Prof. Seiler is an excellent advisor because he provides sound advice at the right place and at the right time. In addition to research opportunities, Prof. Seiler has provided me with many career development opportunities, which have been immensely helpful in my search for future opportunities.

We started working on flight using a single control surface because of a question that Prof. Gary Balas asked in 2013. I wish Prof. Balas were here to see the results.

I thank Prof. Demoz Gebre-Egziabher for his guidance and support throughout graduate school. I had the privilege of learning several courses on estimation and navigation from him. I also thank my committee members Prof. Maziar Hemati and Prof. Sairaj Dhople for reading this thesis and providing many constructive comments.

I thank Prof. Manas Kumar Laha of IIT Kharagpur for introducing me to flight dynamics and control. He is an outstanding instructor whose class notes I still use.

The flight tests and other experiments reported in this thesis were only possible because of the help and support of staff members at the UAV Lab. Specifically, all the flight tests used the `AuraUAS` software written and maintained by Curtis Olson. Curt also designed the autoland trajectory, created the head-up display in the flight videos, and was the test pilot for all the flights. Andrew Mahon, Chris Regan, and Brian Taylor helped with the component sizing and selection (Section 2.2.2) and with the wind tunnel tests for the thrust characterization (Section 2.3.4). Kathryn Kwiecinski and Chris helped with the bench tests for the servomotor characterization (Section 3.2.2). Brian helped with the initial flight tests using a single control surface, which eventually

evolved into this thesis. I also thank: T. Colten of Sentera, LLC for donating the first Vireo aircraft and several more since then; R. Condron, E. Day, D. Dessens, A. Krause, and P. Wagner for proving single surface flight using manual pilot control; N. Carter, R. Condron, D. Chryst, J. Habeck, K. Hamidun, L. Heide, A. Hrovat, C. Olson, C. Regan, and B. Taylor for helping with aircraft integration, maintenance, repair, and flight testing.

I thank Dr. Péter Bauer, Dr. Bin Hu, Dr. Sei Zhen Khong, Dr. Marcio Lacerda, Dr. Daniel Ossmann, Dr. Tamás Péni, Dr. Harald Pfifer, Dr. Zoltán Szabó, Dr. Bálint Vanek, and Dr. Shu Wang for stimulating discussions about integral quadratic constraints, robust synthesis, fault diagnosis, and unmanned aircraft reliability. I also thank Dr. Bálint Vanek and Dr. József Bokor for their hospitality during my visit to the Institute for Computer Science and Control (SZTAKI) in Budapest, Hungary.

I thank Dr. Vibhor Bageshwar for a very rewarding time at Honeywell Aerospace.

I thank my colleagues in graduate school: Abhineet, Adhika, Aditya, Albert, Andrea, Aniket, Béla, Bin, Bryce, Christian, Claudia, Daniel, David, Deb, Hamid, Harald, Harish, Inchara, Jen, Joel, John, Jordan, Julian, Jyot, Kathy, Kerry, Masanori, Paul, Praveen, Piyush, Rohit, Sally Ann, Sanjana, Shu, Supriya, Trevor, Vahhab, and any others whom I may have missed out. I thank the BME soccer team for an excellent time on the field.

I thank my sister Rasika, my brother-in-law Harysh, and my parents Mr. and Mrs. Venkataraman for supporting my professional endeavors.

Finally, I thank the financial sponsors of this research. This work was supported by the National Science Foundation under Grant No. NSF/CNS-1329390 entitled "CPS: Breakthrough: Collaborative Research: Managing Uncertainty in the Design of Safety-Critical Aviation Systems". In addition, the University of Minnesota financed my final year through the 2017-2018 Doctoral Dissertation Fellowship.

Dedication

To my sister Rasika, my brother-in-law Harysh,
and my parents Mr. and Mrs. Venkataraman.

Abstract

Small unmanned aircraft systems (UAS) have recently found increasing civilian and commercial applications. On-board fault management is one of several technical challenges facing their widespread use. The aerodynamic control surfaces of a fixed-wing UAS perform the safety-critical functions of stabilizing and controlling the aircraft. Failures in one or more of these surfaces, or the actuators controlling them, may be managed by repurposing the other control surfaces and/or propulsive devices. A natural question arises in this context: What is the minimum number of control surfaces required to adequately control a handicapped aircraft? The answer, in general, depends on the control surface layout of the aircraft under consideration. For some aircraft, however, the answer is one. If the UAS is equipped with only two control surfaces, such as the one considered in this thesis, then this limiting case is reached with a single control surface failure. This thesis demonstrates, via multiple flight tests, the autonomous landing of a UAS using only one aerodynamic control surface and the throttle.

In seeking to arrive at these demonstrations, this thesis makes advances in the areas of model-based fault diagnosis and fault-tolerant control. Specifically, a new convex method is developed for synthesizing robust output estimators for continuous-time, uncertain, gridded, linear parameter-varying systems. This method is subsequently used to design the fault diagnosis algorithm. The detection time requirement of this algorithm is established using concepts from loss-of-control. The fault-tolerant controller is designed to operate the single control surface for lateral control and the throttle for total energy control. The fault diagnosis algorithm and the fault-tolerant controller are both designed using a model of the aircraft. This model is first developed using physics-based first-principles and then updated using system identification experiments. Since this aircraft does not have a rudder, the identification of the lateral-directional dynamics requires some novelty.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Small unmanned aircraft systems (UAS) have recently found increasing civilian and commercial applications. Some examples include search and rescue, infrastructure inspection, precision agriculture, and package delivery. Aviation authorities in the United States (US) and the European Union (EU) are developing long-term frameworks for the safe integration of these UAS into their respective national airspaces. The European Aviation Safety Agency (EASA) has developed a concept of operations for UAS using a risk-based approach [1]. The US Federal Aviation Administration (FAA) regulates civil UAS via Part 107 of the Federal Aviation Regulations [2]. These regulations are expected to evolve to accommodate the increasing levels of autonomy of these UAS. Their widespread use, however, is subject to several technical challenges related to safety and reliability. These include obstacle detection, collision avoidance, path planning and routing, automated deconfliction, and on-board fault management.

This thesis focuses on the problem of on-board fault management. The traditional approach to this problem is to use sensors to detect faults and backup components to manage them. This approach, commonly called as hardware redundancy, is not well-suited for small UAS because they have constraints on their size, weight, and power. The modern approach is to use algorithms to detect faults and repurpose existing components to manage them. This approach, commonly called as analytical redundancy, does not require additional hardware and is thus a viable alternative for small UAS.

Each component of a small UAS can fail in a number of different ways. Some failure modes are more likely to occur and are more severe in consequence than others [3,4]. The aerodynamic control surfaces of a fixed-wing UAS perform the safety-critical functions of stabilizing and controlling the aircraft. Failures in one or more of these surfaces, or the actuators controlling them, pose acute safety risks to the UAS and, by extension, to the surrounding environment. This thesis specifically considers the problem of automatically detecting and managing control surface faults on a small, fixed-wing UAS. These faults are manageable as long as the other control surfaces and/or propulsive devices may be repurposed to provide adequate control authority to the handicapped aircraft.

Designing a system that automatically detects and manages control surface faults requires research into the areas of fault diagnosis and fault-tolerant control. Several researchers have investigated this problem on various UAS equipped with different numbers of control surfaces. What sets this thesis apart from the existing literature is the fact that it considers a UAS that is equipped with only two control surfaces (called *elevons*) and one puller-type electric motor. A fault in either of the elevons of this UAS must, therefore, be managed using the only other elevon and the throttle. This problem is interesting because the fault leads to an under-actuated control system. By following the modern approach of analytical redundancy, this thesis develops a fault diagnosis algorithm to detect the occurrence of the fault and a fault-tolerant controller to manage it. Both of these components are validated using flight tests. **The main highlight is a set of flight tests demonstrating the autonomous landing of this UAS using only one aerodynamic control surface and the throttle.**

1.1 Thesis Overview

The flight demonstrations provide a backdrop for the various investigations pursued in this thesis. The chapters of this thesis effectively layout the path to these demonstrations, as explained next. Each chapter includes a survey of the pertinent literature. Chapters 3, 4, 6, and 7 are supported by flight test data.

Chapter 2 introduces the small, fixed-wing UAS considered in this thesis and explains its various subsystems. The chapter then proceeds to develop a nonlinear flight dynamic model of the UAS using physics-based first-principles. To this end, the chapter models the geometric, inertial, propulsive, and aerodynamic properties of the aircraft. Some of these properties are obtained experimentally, e.g. wind tunnel tests are used to characterize the thrust. Others are obtained computationally, e.g. the vortex lattice method (VLM) is used to characterize the aerodynamic stability and control derivatives.

Chapter 3 updates the VLM-derived stability and control derivatives using flight tests. This is essentially an exercise in system identification. In particular, the aircraft dynamics are excited using specially designed control inputs. The limits of the experiments dictate which dynamic modes are excited. The principles of flight dynamics are invoked to identify only those stability derivatives that predominantly affect the excited modes. In addition, the chapter also presents the system identification of the servomotors that actuate the aerodynamic control surfaces. The updated nonlinear model is subsequently trimmed and linearized at multiple constant airspeed flight conditions. The resulting collection of linear models, which are parametrized on the airspeed, form a gridded, linear parameter-varying (LPV) representation of the aircraft dynamics.

Chapter 4 introduces the autopilot architecture of the UAS, including the guidance and control laws. The control law consists of a nominal controller and a fault-tolerant controller. When a fault occurs, the control law switches from the former to the latter. The chapter uses one of the linear models from Chapter 3 to develop the nominal controller, which consists of a total energy controller, a pitch attitude controller, and a roll attitude controller. The chapter concludes by assessing the robustness of the nominal controller and validating it using flight test data.

Chapter 5 presents some theoretical results that are used in designing the fault diagnosis algorithm. In particular, a method is presented for synthesizing output estimators and disturbance feedforward controllers for continuous-time, uncertain, gridded, LPV systems. Integral quadratic constraints (IQCs) are used to describe the uncertainty. Since gridded LPV systems do not have a valid frequency-domain interpretation, the time-domain, dissipation inequality approach is followed.

Chapter 6 introduces the fault detection and isolation (FDI) problem and reviews previous work in this area. The chapter then develops a requirement for the maximum allowable fault detection time by invoking the work of Wilborn and Foster [5] in the area of loss of control (LOC). This requirement is subsequently used to design two candidate FDI algorithms. The first candidate is parity-space based and simply uses the open-loop model of the aircraft dynamics. The second candidate is observer-based and is designed using the results of Chapter 5. The chapter concludes by validating and comparing the performances of both candidates using flight test data in an offline setting.

Chapter 7 reviews previous work in the area of fault-tolerant control and formulates the particular problem of controlling the aircraft in the presence of a stuck elevon. Given that the final goal is to land the UAS, the ability to change and maintain course is crucial. Thus the operable elevon is reserved for lateral control. Two candidate controllers are designed for this purpose. The first candidate is a repurposed version of the nominal roll attitude controller. The second candidate is designed using H_∞ synthesis and has favorable properties over the first, e.g. it explicitly avoids exciting the short period mode. In addition, the total energy controller of Chapter 4 is modified so as to increase the closed-loop damping ratio of the phugoid mode. The overall fault-tolerant controller meets the minimum desired robustness margins. The chapter concludes with flight tests demonstrating the autonomous landing of the UAS using only one elevon and throttle.

Chapter 8 presents the conclusions of this thesis and makes recommendations for future research. This chapter is followed by several appendices that provide material supplementary to the preceding chapters.

1.2 Thesis Contributions

In seeking to arrive at the flight demonstrations, this thesis makes several contributions.

1. **System identification:** The longitudinal modes (short period and phugoid) are weakly coupled for most rigid, fixed-wing aircraft. Consequently, the aerodynamic parameters affecting each mode may be identified independent of the other using separate reduced-order models. The lateral-directional modes, on other hand, are strongly coupled. Thus the aerodynamic parameters affecting these modes need to be identified simultaneously using aileron and rudder excitations. The aircraft considered in this thesis, however, does not have a rudder. Consequently, there are too many free aerodynamic parameters in the lateral-directional model that all of them cannot be identified using the aileron excitations. Chapter 3 resorts to two novel steps to navigate this roadblock. First, the chapter discerns the modes whose damping ratios and natural frequencies change significantly compared to their VLM-derived values. Second, the chapter updates only the stability and

control derivatives that predominantly affect the discerned modes, while retaining the remaining derivatives at their respective VLM-derived values. The main contribution of Chapter 3 is thus a procedure to identify the lateral-directional dynamics for a class of small, fixed-wing UAS that do not have a rudder.

2. **Robust synthesis:** Chapter 5 makes two contributions in the area of robust synthesis for gridded LPV systems. The first contribution is that a notion of duality is developed for the worst-case gain analysis of uncertain, gridded LPV systems. This includes notions of dual LPV systems and dual IQCs. Further, several technical results are developed to demonstrate that the sufficient conditions for bounding the worst-case gain of the primal and dual uncertain LPV systems are equivalent. The second contribution is that convex conditions are derived for the synthesis of robust output estimators for uncertain LPV systems. The estimator synthesis conditions, together with the duality results, enable the convex synthesis of robust disturbance feedforward controllers.

3. **Fault diagnosis:** Chapter 6 makes a contribution towards the detection and isolation of stuck control surface faults for fixed-wing aircraft. In particular, the chapter leverages the previous contribution (robust synthesis) to design a robust LPV observer of the control surface positions. The observer generates reliable estimates of the control surface positions even in the presence of a fault. This chapter also establishes some preliminary connections between the detection time requirement of FDI algorithms and the flight envelope of the aircraft.

4. **Fault-tolerant control:** Chapter 7 makes a contribution in the area of fault-tolerant control of fixed-wing aircraft. In particular, the designed fault-tolerant controller is capable of landing the aircraft using only one elevon and the throttle. The elevon is controlled using a H_∞ controller and the throttle is controlled using a modified total energy controller. Although both of these control design methods are standard, the flight tests demonstrating the autonomous landing are, to the best of the author's knowledge, among the first of their kind.

Finally, unless noted otherwise, this thesis uses SI units for all dimensional variables.

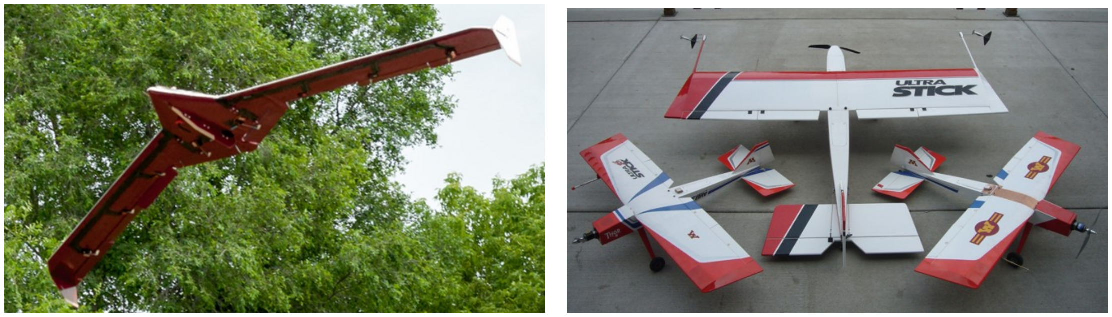
Chapter 2

UAS Platform and Aircraft Modeling

2.1 Introduction

The Uninhabited Aerial Vehicle Laboratory (UAV Lab) is the University of Minnesota's (UMN) primary facility for research involving UAS [6, 7]. The research that is conducted is application oriented, i.e. project objectives and requirements are typically drawn from the specific problem at hand. However, the process of meeting project requirements typically uncovers fundamental research questions in the areas of guidance, navigation, control, dynamics, modeling, and optimization. Application areas are diverse and interdisciplinary. Recent application areas include flutter prediction and control, fault diagnosis, fault-tolerance, spin susceptibility, synthetic airdata and wind estimation, GPS-denied navigation, aerial photography, and aerial entomology collection. The UAV Lab maintains and operates several UAS platforms (see Figure 2.1) to support these application areas as well as answer the fundamental research questions.

In addition to the UAS platforms, the UAV Lab maintains a high-fidelity nonlinear simulation environment that enables model-based algorithm development and validation [8]. The different UAS platforms maintained by the Lab are modeled within this simulation environment. Consequently, the algorithms may be tested in the software- and processor-in-the-loop simulations before being tested in real flight. This allows for



(a) The mini-MUTT flexible aircraft.

(b) The Ultra Stick family of aircraft.

Figure 2.1: Some of the aircraft maintained and operated by the Uninhabited Aerial Vehicle Laboratory at the University of Minnesota.

faster development times and enables the early detection and resolution of implementation issues. Overall, the simulation allows for a faster turnaround from model-based design to experimental validation.

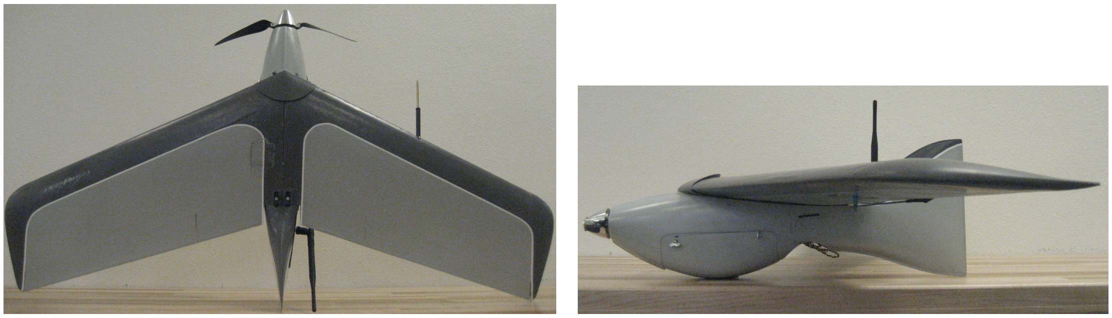
As mentioned previously, this thesis develops model-based algorithms for performing fault diagnosis and fault-tolerant control on a small UAS maintained by the UAV Lab. Any model-based algorithm is only as good as the underlying model. Hence, this chapter is devoted to modeling the UAS using physics-based first-principles. Section 2.2 describes the UAS and its various subsystems. Section 2.3 develops a first principles nonlinear model of the UAS, which is parametrized using the geometric (Section 2.3.2), inertial (Section 2.3.3), propulsive (Section 2.3.4), and aerodynamic (Section 2.3.5) properties of the aircraft. This model is trimmed (Section 2.4.1) and linearized (Section 2.4.2) at different flight conditions to obtain a collection of linear models. These linear models are subsequently used in designing the nominal controller (Chapter 4), the fault diagnosis algorithm (Chapter 6), and the fault-tolerant controller (Chapter 7).

2.2 UAS Platform

The fault diagnosis and fault-tolerant control algorithms developed in this thesis are centered around a particular small UAS that the UAV Lab maintains and operates. The UAS is comprised of three main subsystems: the aircraft, the avionics, and the ground station. The remainder of this section provides the details on each subsystem.

2.2.1 Aircraft

The aircraft is called the *Vireo* and is pictured in Figures 2.2a and 2.2b. As seen in the photographs, it is comprised only of a wing and a fuselage. The rear portion of the fuselage blends into the vertical tail. Since it does not have a horizontal tail, the camber line of the airfoil is reflexed to provide static stability. The wings and the fuselage are constructed out of foam and plastic, and disassemble for easy storage and transportation. This class of aircraft was originally designed and manufactured by Sentera, LLC [9] for commercial application in precision agriculture. In July 2015, Sentera donated a preproduction airframe of the Vireo to the UAV Lab. This is the airframe that is pictured in Figures 2.2a and 2.2b. Since then, Sentera has donated three more such airframes to the UAV Lab, thereby ensuring the continuity of the Vireo as a flight test platform.



(a) Top view.

(b) Port-side view.

Figure 2.2: The flight test platform is a small unmanned aircraft called the *Vireo*.

The fully integrated aircraft has a gross mass of 1.28 kg, a wing span of 0.97 m, and a fuselage length of 0.52 m. Its propulsion system consists of a nose-mounted puller-type electric motor that drives a fixed-pitch propeller. It is equipped with a pair of elevons, one on each wing. The elevons function as conventional elevators when deflected symmetrically and as conventional ailerons when deflected anti-symmetrically. Hence, the pair of elevons can simultaneously provide longitudinal and lateral control to the aircraft. Since the aircraft does not have a rudder, directional control can only be achieved indirectly via lateral control. For uniform convention, a trailing-edge down deflection of each elevon is considered positive. Each elevon is actuated by an independent servo motor. Moreover, each elevon is considered to be undeflected when it is flush with the airfoil and can achieve the deflection range $[-30, +20]\,^{\circ}$. Finally, owing to the low-cost nature of the UAS, the elevons are the only two aerodynamic control surfaces on the aircraft. This directly motivates the need for fault diagnosis and fault-tolerance capabilities on this low-cost UAS.

2.2.2 Avionics

The Vireo airframe that is acquired from Sentera is made airworthy by outfitting it with a battery, an electric motor, a propeller, actuators, sensors, a telemetry radio, a flight computer, and other supporting avionics. Figure 2.3 shows a schematic of the interfaces between the flight computer and the other avionics components. Table 2.1 provides a list of the final selected components. In particular, the battery is sized to meet a minimum endurance requirement of 10 min. The electric motor and propeller are sized to meet a maximum rate-of-climb requirement of 500 ft/min. The detailed performance analysis that forms the basis for sizing the battery, the electric motor, and the propeller is summarized in [10]. The servo motors controlling the elevons are selected such that their bandwidth is at least four times that of the fastest rigid body mode. Section 3.2 provides further details on the dynamics and the model of the servo motors.

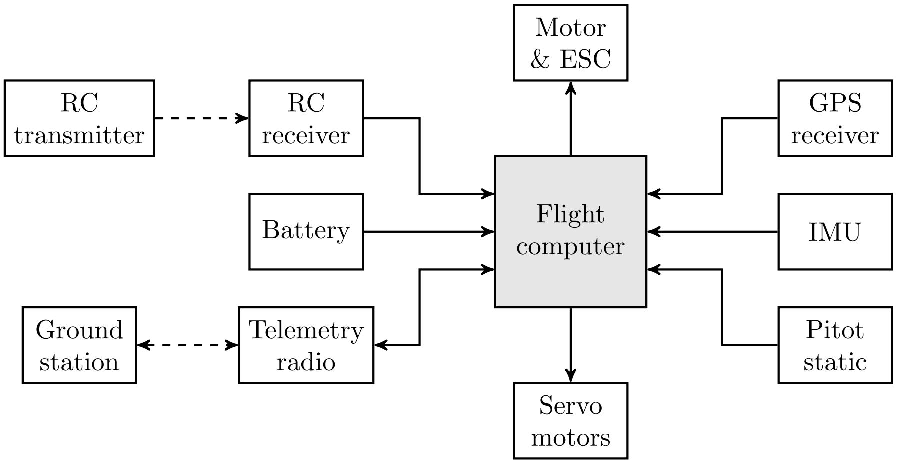


Figure 2.3: The flight computer interfaces with all the other avionics components.

The centerpiece of the avionics is the Beaglebone flight computer (Figure 2.3), which runs a version of Debian Linux and implements the software for performing sensing,

Table 2.1: The Vireo is outfitted with the following components.

Component	**Make/Model**	**Key specifications**
Battery (LiPo)	Thunder Power Elite	3S, 1800 mA h, 55C
Electric motor	Eflite Park 450 Outrunner	890 kV
Propeller	Aeronaut Cam 12” × 5”	Max RPM $= 13,000$
Servo motors	KST DS135MG	Bandwidth $\approx$ 9 Hz
Telemetry radio	3D Robotics	915 MHz
Electronic speed controller	Castle Creations Talon 25	
IMU	VectorNav VN-100	
GPS receiver	Gumstix U-Blox 7P	
Pitot-static system	Eagle Tree Prandtl	
Flight computer	Beaglebone and Goldy 2 board	

navigation, guidance, control, and data logging. The Beaglebone interfaces with a custom board called the Goldy 2, which is the result of a joint undertaking between UMN and Sentera. The Goldy 2 runs a bootloader that handles some of the machine-level tasks in the system. The flight software is documented in the `AuraUAS` codebase [11]. The core functions of the autopilot, such as the main loop, are implemented in C++ and Python. The main loop consists of data acquisition, navigation, guidance, feedback control, actuator command, data logging, and telemetry. The clock frequency of the Beaglebone is 1 GHz and its *programmed* sample rate is 100 Hz. However, the *actual* sample rate is ultimately limited by the packet rate of the Goldy 2. In practice, this packet rate is inconsistent and varies within any given flight between 85 Hz and 100 Hz, with a median value of 90 Hz. This variable sample rate introduces some challenges in implementing discrete-time feedback controllers, as explained further in Chapter 7.

The flight computer is supported with a set of sensors that provide it with real-time measurements. The inertial measurement unit (IMU) is comprised of accelerometer and gyroscope triads that measure translational accelerations (a_x, a_y, a_z) and angular velocities (p, q, r), respectively. The global positioning system (GPS) receiver provides positions in a geodetic reference frame and velocities in a local North-East-Down frame. The magnetometer measures the aircraft heading with respect to the magnetic North. The pitot-static system measures the static and the dynamic air pressures, from which the barometric airspeed V and the barometric altitude h_b are estimated.

The GPS provides position and velocity measurements to the flight computer at 1 Hz. This rate is not high enough to be used for feedback control of the aircraft. In addition, control laws require estimates of the attitude of the aircraft. Hence, an extended Kalman filter (EKF) [12] is employed to provide accurate and high bandwidth estimates of the aircraft navigation states, i.e. attitude, position, and velocity. The EKF also estimates the biases in the accelerometer and gyroscope measurements. For the purpose of control design (Chapters 4 and 7), the estimates of the navigation states are treated as measured outputs. However, since the computations performed by the EKF consume some fraction of the frame time, the navigation state estimates are a delayed version of their "true" counterparts. The control design accounts for this by ensuring that the time delay margin of the closed-loop is several frames in length.

Finally, the set of sensors deployed on the Vireo is considered typical for a small, low-cost UAS. Several other low-cost UAS platforms come equipped with a similar set of sensors, e.g. the Agribotix Hornet, the Trimble UX5, the senseFly eBee, and the RoboFlight RF1. Since these sensors do not have hardware redundancy, managing failures in them is an equally interesting problem in fault diagnosis and fault-tolerance, as reported in several recent works [13–17].

2.2.3 Ground Station

The ground component of the UAS consists of a RC transmitter, a telemetry radio, and a laptop computer. The RC transmitter gives the human pilot direct control over the aircraft during emergencies and critical flight phases. The transmitter is usually used in the stick-to-surface mode, although the flight computer may be programmed to allow for stick-to-attitude or stick-to-rate control. The ground station allows the UAS operator to view the real-time flight path of the aircraft and send commands to the aircraft, such as waypoints and routes, mission updates, controller gain updates, etc.

2.2.4 Standard Operating Procedure

All flight testing is conducted at the University of Minnesota Outreach, Research and Education (UMore) Park[1], at an airfield located at 44°43′32"N and 93°4′45"W. The

[1] `http://www.umorepark.umn.edu`

operations comply with the Part 107 of the Federal Aviation Regulations [2]. The tests involve a human pilot (at the RC transmitter), a UAS operator (at the ground station), and an external observer. The airfield is kept clear of non-participants during all operations. The aircraft is flown under a ceiling of 400 ft above ground level (AGL) and is kept within the visual line-of-sight of all personnel at all times.

Typical flight experiments involving the Vireo are divided into three segments: (1) hand-launched takeoff, (2) research experiments, and (3) landing. For safety reasons, the take-off is always under the manual control of the pilot. Once airborne, the pilot takes the aircraft up to the desired cruise altitude, which varies between 200 ft and 350 ft AGL. The pilot then turns on the autopilot using a toggle on the RC transmitter. Once under autopilot control, the aircraft by default tracks a circular holding pattern and awaits further instructions from the UAS operator. The UAS operator conducts the research experiments by sending commands from the ground station. After the completion of the experiments, the aircraft is landed either manually by the pilot or autonomously by the autopilot. All flights are documented and the logged flight data is publicly available on the UMN Digital Conservancy[2]. This thesis refers to specific flights using the nomenclature FLT**X**, where **X** denotes the flight number.

2.3 First Principles Nonlinear Model

The focus of this thesis is on *model-based* fault detection, isolation, and tolerance. The foundation for designing these algorithms is a good model of the aircraft dynamics. In addition to algorithm design, a good model allows one to simulate the various scenarios that may be encountered in implementation. This section considers a nonlinear model of the Vireo that is constructed using physics-based first principles. This nonlinear model is implemented in the simulation environment that the UAV Lab maintains [6]. For modeling purposes, the Vireo is assumed to be a rigid body with six degrees of freedom.

The literature on modeling rigid body aircraft dynamics is vast. Shortly after the first flights made by the Wright brothers in December 1903, many investigators started studying the problem of aircraft stability and control. However, the published works of Bryan [18] and Lanchester [19] are widely credited with laying the first foundations for

[2] `https://conservancy.umn.edu/handle/11299/163580`

the subject. Their modeling framework is the *de facto* standard used by every major textbook on flight dynamics. While the subject of aircraft stability and control has its roots in aerodynamics [19–23], modern treatments of the subject take an integrated approach by drawing from aerodynamics, mechanics, and control theory [24–28].

Bryan's description of aircraft dynamics is based on the general equations of motion of a rigid body with six degrees of freedom. The equations of motion of a rigid body aircraft are parametrized by its geometric, inertial, propulsive, and aerodynamic properties. The equations themselves are widely documented in the nonlinear state-space form [25,26,28] and are thus not reported in this thesis. Hence, the following subsections focus on modeling the geometric, inertial, propulsive, and aerodynamic properties of the Vireo using computational and experimental methods. An important modeling consideration is the level of fidelity required. In this regard, the assumption of six degrees of freedom automatically sets the number of states to twelve. In addition, the geometric and inertial parameters enter the equations in a highly structured form. Hence, the model fidelity is entirely controlled by the propulsion and aerodynamic models.

Finally, the parametrization of the equations of motion lends itself naturally to the construction of grey-box models, wherein the model structure is known but the model parameters are unknown. This is in contrast to black-box models, which make no assumptions about the model structure. Section 3.3, which deals with aircraft system identification, presents these grey-box models. The grey-box model parameters are initialized using the results of this section and then updated using flight test data.

2.3.1 Preliminaries

Some preliminaries are first presented on the reference frames and the model structure. The notation primarily follows [25, 26], except for some variables that are redefined. All drawings of the Vireo shown are reproduced with permission from Sentera, LLC.

Axes definitions

In the process of modeling the aircraft, this thesis uses Earth-fixed ($o_e x_e y_e z_e$), body ($o_b x_b y_b z_b$), and stability ($o_s x_s y_s z_s$) axes [25]. All three sets of axes are right-handed and orthogonal. For the Earth-fixed axes, the origin o_e is located at an arbitrary point on the surface of the Earth, $o_e x_e$ points North, $o_e y_e$ points East, and $o_e z_e$ points into the Earth along the gravity vector. The Earth-fixed axes are sometimes referred to as the North-East-Down (NED) reference frame. Since the $o_e z_e$ axis is tied to the gravity vector, the Earth is assumed to be locally flat. Moreover, the Earth's rotation is neglected and the $o_e x_e y_e z_e$ axes are assumed to represent the inertial reference frame.

Figure 2.4 shows the definitions of the body and stability axes. Both these axes are aircraft-fixed reference frames whose origins coincide with the center of mass. For practical purposes, the center of mass (CM) is assumed to coincide with the center of gravity (CG). The body axes are defined by selecting a physically identifiable feature on the aircraft. In this thesis, $o_b x_b$ is defined to be parallel to the shaft of the electric motor and points towards the nose. $o_b z_b$ is coplanar with the plane of symmetry of the aircraft and points towards the bottom of the fuselage. Consequently, $o_b y_b$ points towards the starboard wing.

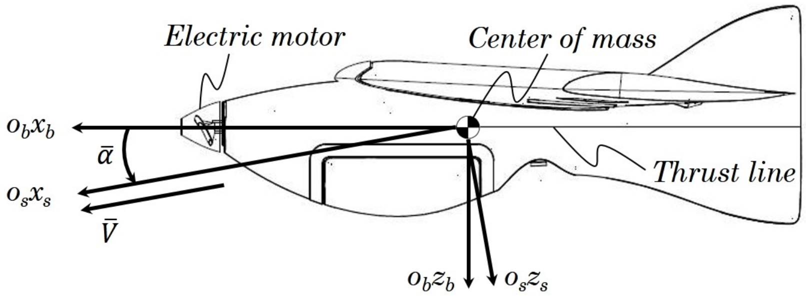


Figure 2.4: The body and stability axes definitions. The $o_b x_b$ axis is parallel to the shaft of the electric motor. The $o_s x_s$ axis is parallel to the trim airspeed vector.

The stability axes are defined after selecting a trim angle-of-attack $\bar{\alpha}$. As indicated in Figure 2.4, the stability axes are obtained by rotating the body axes about the $-o_b y_b$ axis by $\bar{\alpha}$. Thus, $o_s x_s$ is parallel to the trim airspeed vector, whose magnitude is $\bar{V}$. The $o_s y_s$ axis remains collinear with the $o_b y_b$ axis. The aerodynamic model presented in Section 2.3.5 utilizes the stability axes. The value of $\bar{\alpha}$, and hence the orientation of the stability axes with respect to the airframe, is unique for every flight condition. However, stability axes are used to study perturbations relative to a given flight condition. Hence, for any given flight condition, the stability axes are defined and fixed to the aircraft at the outset. Finally, a vector that is resolved in the body axes is transformed to the stability axes by multiplying it with the (unitary) direction cosine matrix (DCM):

$$C_{s\leftarrow b} = \begin{bmatrix} \cos\bar{\alpha} & 0 & \sin\bar{\alpha} \\ 0 & 1 & 0 \\ -\sin\bar{\alpha} & 0 & \cos\bar{\alpha} \end{bmatrix}. \tag{2.1}$$

The inverse transformation is performed using the DCM $C_{b\leftarrow s} = C_{s\leftarrow b}^{-1} = C_{s\leftarrow b}^{T}$.

Model structure

The aircraft equations of motion [25,26,28] in the nonlinear state-space form are:

$$\dot{x}\,(t) \;=\; f(x,u,t), \tag{2.2}$$

$$y\,(t) \;=\; h(x,u,t), \tag{2.3}$$

for some initial condition $x\,(0)$. Here, $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ is the input vector, $y \in \mathbb{R}^p$ is the output vector, and $t \in \mathbb{R}^+$ is time. In addition, $f : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^+ \to \mathbb{R}^n$ is the state function and $h : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^+ \to \mathbb{R}^p$ is the output function.

The state vector x contains the Euler angles (ϕ, θ, ψ), the angular velocity components expressed in the body axes (p, q, r), the airspeed components expressed in the body axes (u, v, w), and the position of the aircraft expressed in the Earth-fixed axes (X_e, Y_e, Z_e). This thesis employs a local East-North-Up coordinate system for the Earth-fixed axes. The input vector u contains the throttle setting δ_t, the left elevon deflection δ_l, and the right elevon deflection δ_r. Although not shown above, the corresponding commanded inputs are denoted by δ_{tc}, δ_{lc}, and δ_{rc}. The output vector y contains the barometric airspeed $V = \sqrt{u^2 + v^2 + w^2}$, the barometric altitude h_b, the geodetic altitude h, the Euler angles (ϕ, θ, ψ), the angular velocity components (p, q, r), and the translational acceleration components (a_x, a_y, a_z).

As described in Section 2.2.2, the components of y are either measured by the on-board sensors $(V, h_b, h, p, q, r, a_x, a_y, a_z)$ or obtained from the EKF (ϕ, θ, ψ). In addition, for simulation purposes, the angle-of-attack $\alpha = \tan^{-1}\left(\frac{w}{u}\right)$, the angle-of-sideslip $\beta = \sin^{-1}\left(\frac{v}{V}\right)$, and the flight path angle $\gamma = \theta - \alpha$ are computed. The simulation environment uses the full set of states, inputs, and outputs described above. However, the control designs presented in Chapter 7 will consider only a subset of these variables, depending on the quantities to be controlled. Unless noted otherwise, this thesis uses SI units for all the dimensional variables.

The state transition equation (2.2) can be decomposed into two parts: the forces and moments acting on the aircraft and the equations of motion describing the evolution of the states [25]. This is shown pictorially in Figure 2.5. There are three main sources of forces and moments: aerodynamic ($\boldsymbol{F_a}, \boldsymbol{M_a}$), gravitational ($\boldsymbol{F_g}.\boldsymbol{M_g}$), and propulsive ($\boldsymbol{F_p}, \boldsymbol{M_p}$). Each source has a corresponding reference point and a model that depends on x and u. The forces and moments are shown in boldface to emphasize the fact that they are vectors. The total force $\boldsymbol{F}$ and the total moment $\boldsymbol{M}$ are obtained via vector addition. For consistency, all the force and moment vectors are resolved in the body axes. The nonlinear equations of motion (EOM) use $\boldsymbol{F}$ and $\boldsymbol{M}$ to simulate the evolution of x, which is fed back to the force and moment models. Although not shown explicitly in Figure 2.5, the geometric and inertial parameters appear in the EOM. The EOM and the gravity model are well documented [25, 26, 28]. Hence, the next few subsections will focus on the geometry, inertia, propulsion, and aerodynamic models.

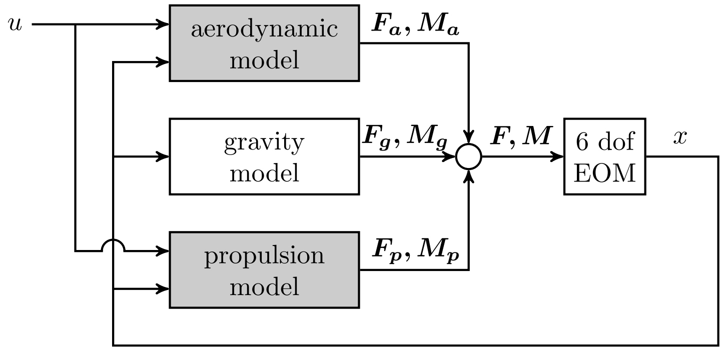

u
aerodynamic model
$\boldsymbol{F_a}, \boldsymbol{M_a}$
gravity model
$\boldsymbol{F_g}, \boldsymbol{M_g}$
$\boldsymbol{F}, \boldsymbol{M}$
6 dof EOM
x
propulsion model
$\boldsymbol{F_p}, \boldsymbol{M_p}$

Figure 2.5: The model structure, depicting the three main contributors to the forces and moments. The aerodynamic and propulsive models are described in detail.

Finally, the model structure shown in Figure 2.5 is simply one piece of the simulation environment, which is described in detail in Section 2.2 and Figure 2.6 of [8]. In this regard, it is worth noting that Figure 2.5 describes the contents of the block entitled “Nonlinear Aircraft Model” in Figure 2.6 of [8].

2.3.2 Geometry Model

The geometric layout of the aircraft is an essential part of the model. For flight dynamics analysis, the geometry of the aircraft is adequately described by a small number of dimensional reference parameters, which are listed in Table 2.2.

Table 2.2: The geometric reference parameters of the Vireo.

Parameter	**Notation**	**Value**
Wing span	b	0.97 m
Wing area	S	0.21 m^2
Aspect ratio	Æ	4.46
Mean aerodynamic chord	$\bar{c}$	0.22 m
Standard mean chord	c	0.22 m
Leading edge sweep angle	Λ	24.9°

In addition to these parameters, the propulsion and aerodynamic models have internal reference points and axes for modeling convenience. The aerodynamic forces and moments are modeled with respect to the aerodynamic reference point (AEROREF), which is defined as the leading edge of the root chord. (This is not to be confused with the classical aerodynamic center.) The propulsive forces and moments are modeled with respect to the thrust line, which is collinear with the shaft of the motor. The CG, AEROREF, and thrust line are themselves referenced to the FIREWALL, which is defined as the point where the motor mount intersects the bottom of the fuselage. Figure 2.6 shows the locations of these reference points relative to the outer mold line.

2.3.3 Inertia Model

For flight dynamics, the inertial properties of the aircraft are adequately described by the gross mass, the CG location, and the moments and product of inertia. Table 2.3 lists the measured or estimated values of these parameters for the fully integrated aircraft. The CG location is expressed as a vector in a reference frame that is parallel to $o_b x_b y_b z_b$, but whose origin is coincident with the FIREWALL. While the gross mass and CG location are measured, the moments and product of inertia are estimated using a computer-aided design (CAD) tool called OpenVSP. This tool is used to model discrete and continuous mass distributions, of different shapes and sizes, with user-specified density profiles.

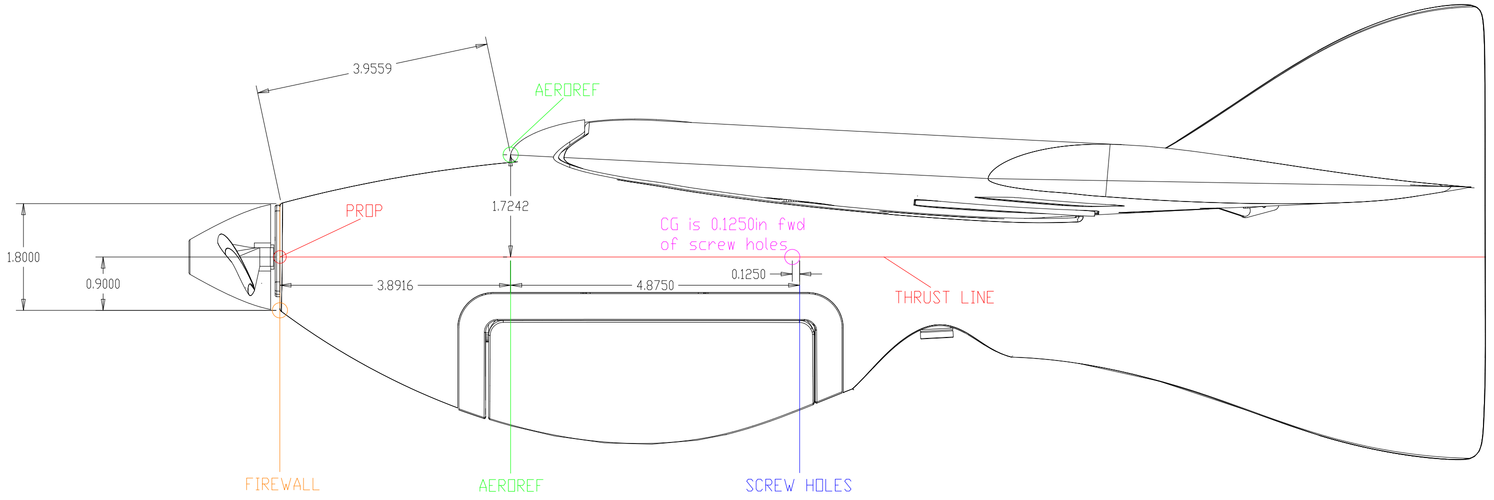
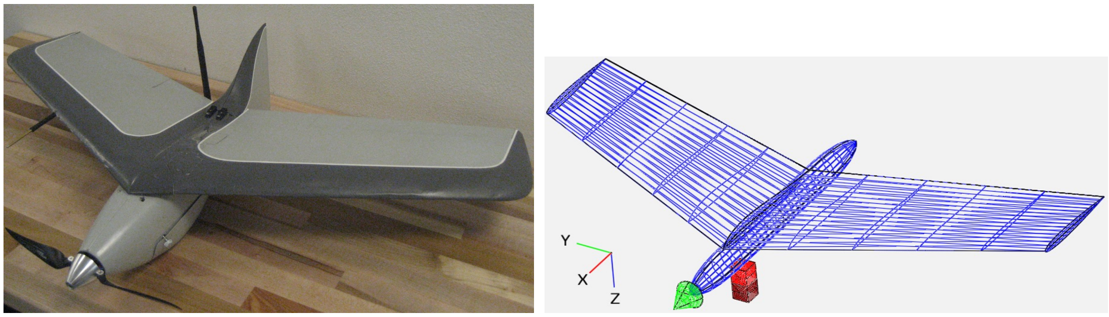


Figure 2.6: The port-side drawing of the Vireo, showing its outer mold line. The locations of the firewall, the thrust line, the aerodynamic reference point, and the center of gravity are indicated. All dimensions shown are in inches. The drawing is reproduced with permission from Sentera, LLC.

Figure 2.7 shows isometric views of the Vireo and the corresponding OpenVSP model. The model only includes the major contributors to the total inertia, i.e. the wing, fuselage, motor and propeller, flight computer, and battery. Each of these components is modeled as a continuous mass distribution with uniform density. The wing and the fuselage are modeled as thin-walled elements, while the other components are modeled as solid masses. Table 2.3 lists the estimated inertia components.

Table 2.3: The inertial parameters of the Vireo.

Parameter	**Notation**	**Value**
Gross mass	m	1.28 kg
CG location	–	$(-0.219, 0, -0.0229)$ m
Roll inertia	I_{xx}	0.0255 $\text{kg}\,\text{m}^2$
Pitch inertia	I_{yy}	0.0211 $\text{kg}\,\text{m}^2$
Yaw inertia	I_{zz}	0.0433 $\text{kg}\,\text{m}^2$
Product of inertia	I_{xz}	0.0020 $\text{kg}\,\text{m}^2$



(a) Photograph of the Vireo.

(b) Model used to estimate the inertia.

Figure 2.7: Port-side isometric views of the Vireo and the model used to estimate the inertia. The model assumes a continuous material with uniform density.

2.3.4 Propulsion Model

The propulsion system of the Vireo consists of a nose-mounted puller-type electric motor that drives a fixed-pitch propeller. The motor shaft is coplanar with the plane of symmetry of the aircraft. Table 2.1 lists the key component specifications. The propulsion system produces a net thrust $\boldsymbol{T}$ that is parallel to the motor shaft. From the definition of the body axes, it follows that $\boldsymbol{T}$ is also parallel to the $o_b x_b$ axis. Hence, $\boldsymbol{F_p}$ is resolved as $(T, 0, 0)$, where T is the magnitude of $\boldsymbol{T}$. The counter-torque exerted by the motor on the airframe is not modeled for simplicity. In addition, Figure 2.6 indicates that the distance between the CG and the thrust line is negligible. Hence, $\boldsymbol{M_p} \approx 0$.

Figure 2.5 shows that $\boldsymbol{F_p}$, and thus T, is a function of x and u. The only input that affects the thrust is the throttle setting δ_t, which is normalized to the range $[0, 1]$. Due to the electronic speed controller, the electric motor responds only if δ_t is greater than some cut-in value $\delta_{t,cut}$. Further, thrust is affected by changes in the dynamic pressure, and is thus a function of the airspeed V and the air density [29]. The effect of the air density manifests via the barometric altitude. However, since UAS are restricted to fly under 400 ft AGL [2], air density variations due to altitude changes are assumed to be negligible. Thus, this section simply expresses T as a function of V and δ_t [3].

For throttle settings above the cut-in value, the form of this function is derived by making three simplifying assumptions. First, for a fixed airspeed, the thrust is assumed to vary linearly with the throttle setting. This assumption is valid if the electronic speed controller is programmed correctly. Second, for a fixed throttle setting, the thrust is assumed to vary quadratically with the airspeed. This follows by noting that the thrust produced by the propeller disk varies linearly with the dynamic pressure. Third, V and δ_t are assumed to enter the function independently of each other. This assumption is made for simplicity. These three assumptions lead to the following function form:

$$T(\delta_t, V) = \begin{cases} 0 & \text{if } 0 \leq \delta_t < \delta_{t,cut} \\ \frac{(\delta_t - \delta_{t,cut})}{(1 - \delta_{t,cut})} \left(c_2 V^2 + c_1 V + c_0 \right) & \text{if } \delta_{t,cut} \leq \delta_t \leq 1 \end{cases} \tag{2.4}$$

where $\{c_i\}_0^2$ are coefficients to be estimated.

In order to estimate these coefficients, an experiment is conducted in the closed return wind tunnel maintained by UMN's Department of Aerospace Engineering and Mechanics. This wind tunnel has a rectangular cross section of size 1 m × 1.25 m and can achieve a maximum flow speed of 38 $\mathrm{m\,s^{-1}}$ in its test section. The tunnel is

[3] Propulsion models are typically based on blade element theory, and are thus parametrized using the non-dimensional coefficient of thrust [29]. This allows the same model structure to be used with different propeller sizes. This section eschews such an approach because the propeller size is fixed throughout the development cycle, spanning modeling and system identification, control design, and flight testing.

equipped with a six degrees of freedom sting that measures the forces and moments along three mutually orthogonal axes. In addition, the air temperature, static pressure, and differential pressure in the test section are collected by a data acquisition system.

In this experiment, the motor and propeller of the Vireo are mounted on a custom test stand that was built by the UAV Lab for the purpose of characterizing dynamic thrust. Before placing the test stand inside the wind tunnel, the throttle is gradually increased from zero until the motor starts rotating. From several trials, $\delta_{t,cut}$ is estimated as 0.306. Although the motor likely exhibits hysteresis, in the form of stiction, this is not modeled for simplicity. Next, the test stand is placed inside the wind tunnel. Subsequently, the wind tunnel is run at the test section airspeeds of zero, 10.79 $\mathrm{m\,s^{-1}}$, and 15.25 $\mathrm{m\,s^{-1}}$. At each airspeed, the test is repeated for the zero and maximum throttle settings. Test section airspeeds that are faster than 15.25 $\mathrm{m\,s^{-1}}$ are not conducted since the cruise airspeed of the Vireo is around 15 $\mathrm{m\,s^{-1}}$ (see Section 2.4.1).

Figure 2.8 shows the results of the wind tunnel experiment. The left-side graph plots the measured thrust versus the measured test section airspeed. The markers represent the measured thrust and the lines represent the function described in Equation (2.4). The data point corresponding to $\delta_t = 1$ and zero airspeed indicates that the maximum achievable static thrust is 9.08 N. As expected, the dynamic thrusts achieved at 10.79 $\mathrm{m\,s^{-1}}$ and 15.25 $\mathrm{m\,s^{-1}}$ are lower than this value. Using the data points corresponding to $\delta_t = 1$, the coefficients are estimated as $c_2 = -0.011$, $c_1 = -0.039$, and $c_0 = 9.084$. Since only three points are used, there is no fitting error with the quadratic function. It is acknowledged that the experiment should have been conducted at a fourth airspeed in order to validate the quadratic dependence of the thrust on the airspeed. Further, no thrust is produced for $\delta_t = 0$ at all airspeeds. However, for nonzero airspeeds, a net drag is produced due to the profile drag of the motor and propeller. All the data points shown in Figure 2.8 are corrected for this net drag at the corresponding airspeed. The right-side graph plots the measured thrust versus the throttle setting. As indicated, a net positive thrust is measured only for $\delta_t > \delta_{t,cut} = 0.306$. The measured thrust for $\delta_t = 1$ for the three airspeeds are plotted using different marker styles. The lines indicate the simple linear relationship that is assumed for $0.306 \leq \delta_t \leq 1$.

Finally, the thrust model described in Equation (2.4) is integrated into the simulation environment. Cumulative experience at the UAV Lab suggests that the electric motor

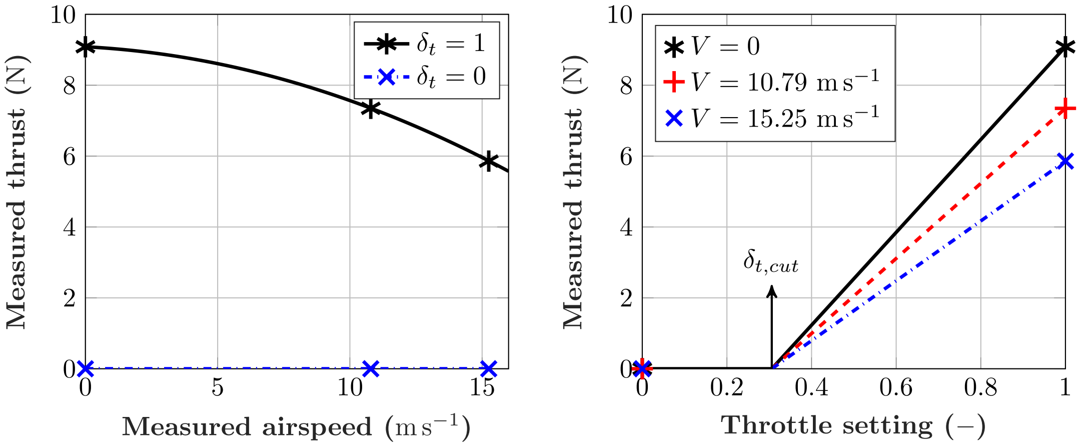


Figure 2.8: Left: The measured thrust varies quadratically with the airspeed for a fixed throttle setting. Right: The measured thrust varies linearly with the throttle setting for a fixed airspeed. The models apply for throttle settings greater than 0.306. Note that the measured thrust has been corrected for the drag.

and the electronic speed controller together have a bandwidth of approximately 1 Hz. These dynamics are modeled using the first-order transfer function $G_t^{\tau_f}(s) = \frac{2\pi}{s+2\pi}e^{-\tau_f s}$, where τ_f is the as-of-yet unknown time delay in the system. For simplicity, the dynamics are localized at the throttle input, i.e. $G_t^{\tau_f}$ is assumed to relate the commanded throttle setting δ_{tc} to the actual throttle setting δ_t. Section 3.3.8 will estimate the time delay τ_f using flight test data and update the simulation environment with that estimate.

2.3.5 Aerodynamic Model

Background

The aerodynamic model is the final piece required to obtain a complete model of the Vireo. Along with the propulsion model, it controls the model fidelity. Since the final goal of the modeling exercise is to develop linear controllers for the aircraft, the aerodynamic forces and moments should be modeled, at a minimum, to the first degree of approximation. This thesis uses the vortex lattice method (VLM) to obtain such first degree approximations. The concepts underlying VLM date back to the late 1930s and early 1940s, e.g. the work of Falkner [30, 31]. However, due to its numerical approach, the method was widely adopted only after the advent of computers. Computer code developed at the NASA Langley Research Center set a near-universal standard for the development of software tools intended for stability and control analysis [32].

VLM is an extension of Prandtl's classical lifting line theory for finite span wings [33]. Instead of using only one horseshoe vortex per lifting surface, as in lifting line theory, the VLM uses a lattice of horseshoe vortices. VLM is based on potential flow theory and makes several assumptions. First, the flow field is assumed to be incompressible, inviscid, and irrotational. Second, unlike general panel methods, the lifting surfaces are assumed to be thin. Third, α and β are assumed to be small angles, i.e. $\sin(\alpha) \approx \alpha$ and $\sin(\beta) \approx \beta$. Because of the inviscid flow assumption, VLM cannot estimate the parasitic drag. Hence, VLM provides good first approximations for most stability and control derivatives, except for those associated with the drag. A comprehensive overview of VLM is available in several textbooks [34, 35].

The VLM modeling and analysis is conducted using Athena Vortex Lattice (AVL), which is a software tool based on legacy code from NASA Langley [32]. In order to use AVL, the user defines the vortex lattice structure for each lifting surface on the aircraft. In addition, a parasitic drag coefficient is specified a priori, if it is known from some other modeling technique. For a given flow condition, the software computes the span-wise lift distribution, the coefficients of lift and drag, and the stability and control derivatives. If the inertial properties of the aircraft are also specified, the software conducts trim and eigenmode analysis. The following subsections elaborate on the VLM modeling and analysis conducted for the Vireo.

Wing airfoil

The Vireo uses the Eppler E330 flying wing airfoil. Figure 2.9 shows the airfoil geometry in normalized Cartesian coordinates, along with the chord line and the mean camber line. The mean camber line is located below the chord line from 60% chord to the trailing edge. The camber reflex is a feature since the airfoil was designed for flying wing aircraft. In particular, the camber reflex favorably alters the moment coefficient of the airfoil and enables a flying wing to be trimmed at a small positive angle-of-attack. The maximum airfoil thickness is 11% and occurs at 30% chord. The maximum mean camber is 2.2% and occurs at 18% chord. Since the airfoil thickness cannot be modeled using VLM, panels are placed along the mean camber line shown in Figure 2.9. This captures the general shape of the airfoil and the flow field around it.

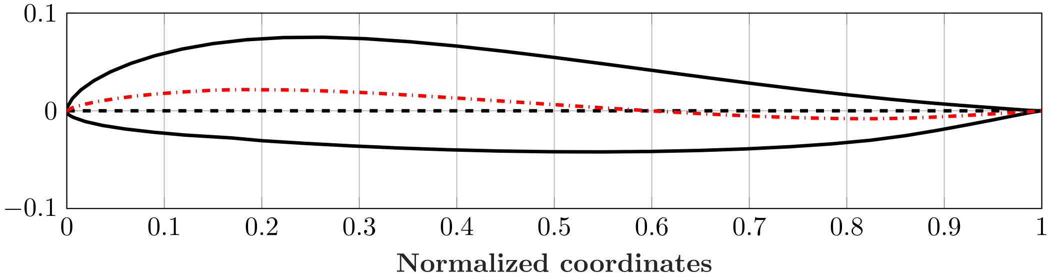


Figure 2.9: The Eppler E330 airfoil, the chord line, and the mean camber line.

Wing and vertical tail lattice structures

In order to create the vortex lattice structure, approximate geometries of the wing and the vertical tail are considered. The left-side of Figure 2.10 shows the approximate geometry of the starboard wing. The shaded quadrilateral indicates the starboard elevon. The wing is divided into three span wise sections, as shown by the dashed lines. The stations separating the three sections are marked on the leading-edge of the wing as S1, S2, S3, and S4. The sections are defined such that stations S2 and S3 straddle the span wise extremities of the elevon. In addition, stations S1 and S4 correspond to the leading-edges of the root chord and tip chord, respectively. The local axes are centered at S1 and are oriented along the span wise and chord wise directions. The arrows represent the free stream velocity V_∞.

In addition to being tapered (Figure 2.10), each wing also features washout, i.e. the local incidence decreases from the wing root (S1) to the wing tip (S4). The local incidence is defined as the angle between the local airfoil chord line and the $o_b x_b$ axis, projected onto the plane of symmetry of the aircraft. Table 2.4 lists the local airfoil chord length (obtained from Figure 2.10) and incidence at each of the four wing stations. These properties are linearly interpolated between adjacent stations. Sections S1-S2 and S2-S3 use the Eppler E330 airfoil, while the wing tip (station S4) uses the symmetric NACA 0010 airfoil. The chord wise panels at a given span wise section are placed along the camber line defined by the local airfoil. Table 2.4 indicates that the wing has a total washout of 4°, i.e. the difference between the airfoil incidences at S1 and S4.

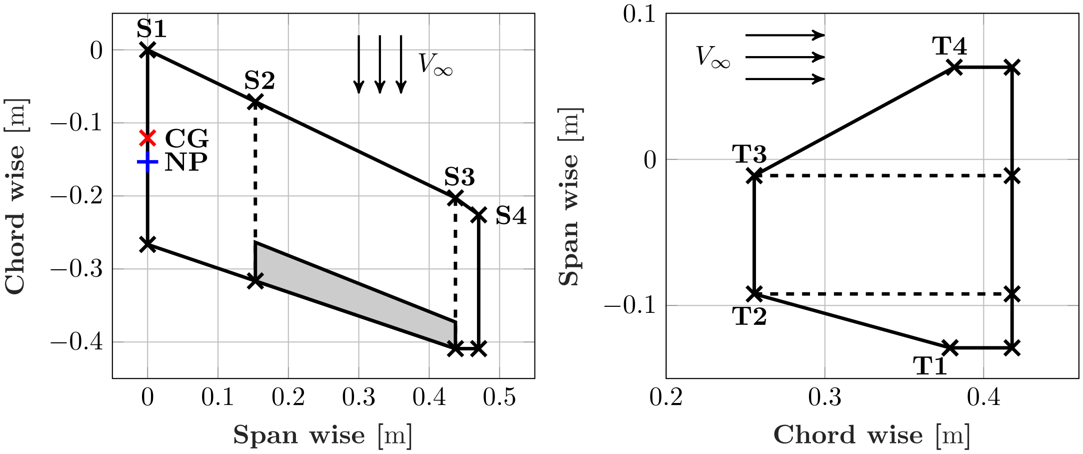


Figure 2.10: The approximate geometries of the starboard wing (shown on the left) and the vertical tail (shown on the right). The arrows represent the freestream.

Table 2.4: Airfoil properties along the starboard wing stations.

Station	**Airfoil**	**Chord length**	**Incidence**
S1	Eppler E330	0.2664 m	5°
S2	Eppler E330	0.2454 m	3.6°
S3	Eppler E330	0.2064 m	1°
S4	NACA 0010	0.1832 m	1°

Figure 2.10 also shows the locations of the center of gravity (CG) and the neutral point (NP) on the root chord of the wing. While the CG location is specified based on the inertia model (Section 2.3.3), the NP location is computed using VLM. The resulting static margin is approximately 15% of $\bar{c}$, indicating a stable short period mode.

The right-side of Figure 2.10 shows the approximate geometry of the vertical tail, as viewed from the port-side of the aircraft. The tail is also divided into three span wise sections, as shown by the dashed lines. The stations separating the three sections are marked on the leading-edge of the vertical tail as T1, T2, T3, and T4. The sections are defined such that stations T2 and T3 correspond to the leading-edge taper breaks. In addition, stations T1 and T4 correspond to the leading-edges of the bottommost and topmost chords, respectively. The entire tail is modeled as a thin, flat plate that is coplanar with the plane of symmetry of the aircraft. As before, the chord lengths are obtained from Figure 2.10 and are linearly interpolated between adjacent stations.

The starboard wing is duplicated across the plane of symmetry to yield the port wing. The two wings and the vertical tail geometries are aggregated into a single vortex lattice model, as shown in Figure 2.11. The wing lattice consists of 16 bound vortices distributed chord wise and 64 trailing vortices distributed span wise. Both the chord wise and span wise distributions map to the cosine function between 0 and π. This ensures dense chord wise distributions near the leading and trailing edges and dense span wise distributions near the wing tips. These are regions that experience large changes in the circulation and hence require tight vortex spacings. The tail lattice uses the same number and distribution of bound and trailing vortices as the wing lattice. Figure 2.11 depicts the bound vortices using solid pink lines and the trailing vortices using black dotted lines. Finally, Section 2.3.2 defined the aerodynamic reference point (AEROREF) as the leading-edge of the root chord of the wing, i.e. station S1. In the later sections, the aerodynamic forces and moments are resolved in axes that are centered at AEROREF and parallel to the stability axes $o_s x_s y_s z_s$.

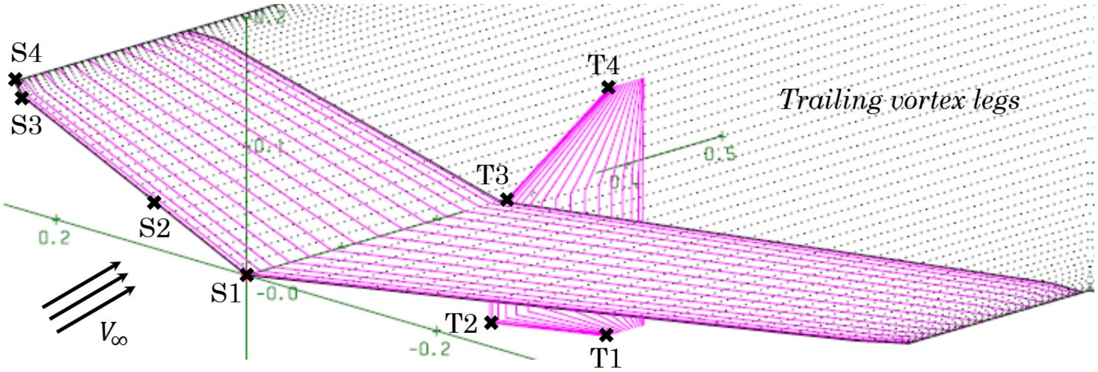


Figure 2.11: Port-side isometric view of the aggregated vortex lattice model showing the bound (solid lines) and the trailing (dotted lines) vortex legs. The arrows represent the freestream.

Drag model

The total drag on an airplane is equal to the sum of the parasitic drag and the lift-induced drag. The parasitic drag is itself composed of many drag components, such as skin friction drag, form drag, interference drag, and trim drag [23]. Obtaining highly accurate drag models is not critical to meeting the objectives of this thesis. Hence, the skin friction drag and the form drag are assumed to be the only two contributors to the parasitic drag. The remainder of this subsection develops simple models for the skin friction drag and the lift-induced drag using empirical formulas. For simplicity, the form drag is assumed to be of the same order as the skin friction drag.

The total drag D is equal to $\frac{1}{2}\rho V^2 S C_D$, where ρ is the air density and C_D is the coefficient of drag. C_D is equal to the sum of the parasitic drag coefficient C_{Dp} and the lift-induced drag coefficient C_{Di} [23]. The parasitic drag coefficient C_{Dp} is expressed as:

$$C_{Dp} = C_f \frac{S_w}{S} + C_{form}, \tag{2.5}$$

where C_f is the skin friction drag coefficient, C_{form} is the form drag coefficient, and S_w is the total wetted area, i.e. the total surface area that is in contact with the flow. Since VLM cannot estimate C_f, it is instead estimated using the Schlichting skin friction formula [23]. This formula empirically relates C_f to the Reynolds number Re for laminar and turbulent flows. The Reynolds number is given by the expression:

$$Re = \frac{\rho V \bar{c}}{\mu}, \tag{2.6}$$

where $\bar{c}$ serves as the characteristic length and μ is the dynamic viscosity. At mean sea level and 15 °C, the density and dynamic viscosity of air are $\rho = 1.225\ \mathrm{kg\,m^{-3}}$ and $\mu = 1.81 \times 10^{-5}\ \mathrm{kg\,m^{-1}\,s^{-1}}$, respectively. Further, at a cruise airspeed of $15\ \mathrm{m\,s^{-1}}$, the Reynolds number is approximately equal to 2.21×10^5. For the sake of conservatism, this Reynolds number is assumed to imply turbulent flow. Thus, the corresponding Schlichting skin friction formula is given by the expression [23]:

$$C_f = \frac{0.455}{\left(\log_{10}(Re)\right)^{2.58}}. \tag{2.7}$$

Using the Reynolds number computed above, C_f is obtained as 6×10^{-3}.

The total wetted area is expressed in terms of the reference area S. The top and bottom wing surfaces together contribute 200% of S and the curvature of the airfoil is assumed to add 4% of S. In addition, the fuselage and the tail are assumed to add 50% of the total wetted area of the wing. Hence, the total wetted area is $S_w = (2 + 0.04)\,(1.5)\,S$.

Assuming that $C_{form} \approx C_f \frac{S_w}{S}$, the net parasitic drag coefficient is $C_{Dp} = 3.69 \times 10^{-2}$.

As the name suggests, the lift-induced drag coefficient C_{Di} depends on the coefficient of lift C_L. It is empirically modeled as $C_{Di} = KC_L^2$, where K is the induced drag parameter [36]. This parameter is empirically computed as:

$$K = \frac{1}{\pi \, \text{Æ} \, e_o}, \tag{2.8}$$

where Æ is the aspect ratio of the wing and e_o is the Oswald efficiency factor. While VLM cannot estimate the Oswald efficiency factor, it can estimate the span efficiency factor e. VLM returns a span efficiency factor of $e = 0.8$ for the Vireo. Under the assumption that C_{Dp} is independent of C_L, e may be used instead of e_o in Equation (2.8). For simplicity, this assumption is made about the Vireo and the induced drag parameter is computed as $K = 8.93 \times 10^{-2}$. In order to complete the drag model, the lift coefficient and the drag polar are considered next.

Coefficient of lift and drag polar

The left-side of Figure 2.12 shows the variation of the lift coefficient C_L as a function of the angle-of-attack α. The dashed line represents the C_L variation expected for a wing of infinite aspect ratio that uses the Eppler E330 airfoil. It is obtained using a software called XFOIL, which implements a viscous formulation of high-order panel methods. Since this formulation captures the boundary layer and the wake of the airfoil, it predicts that the Eppler E330 stalls at $\alpha \approx 11°$, as seen in the plot. On the other hand, the solid line represents the C_L variation computed for the Vireo using VLM. The markers denote the specific cases that are run using AVL. Since VLM assumes the flow to be inviscid, the slope of the lift curve is constant and stall is not predicted. In addition, the slope of the lift curve of the Vireo is lower compared to the two-dimensional airfoil. This is consistent with the expected lifting characteristics of a wing that has finite aspect ratio [23]. The remainder of the aerodynamic model uses the lift curve of the Vireo (solid line). The lift curve of the Eppler E330 (dashed line) is primarily used for stall protection. This is done by setting a lower bound on the airspeed of the Vireo, as explained further in Section 2.4.1. Finally, the lift coefficient at zero angle-of-attack is $C_{L0} = 0.16$ and is shown by the red asterisk in Figure 2.12.

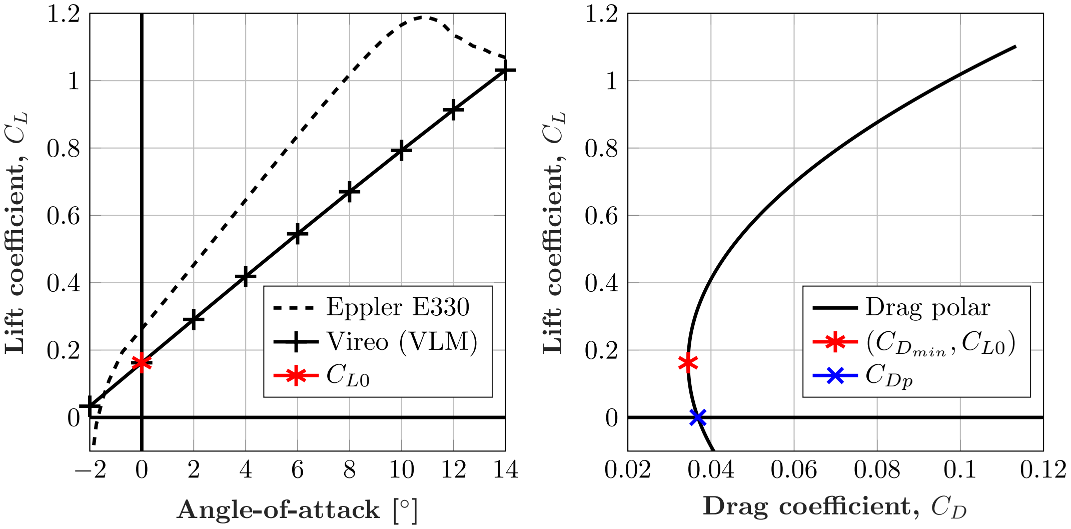


Figure 2.12: Left: The lift coefficient as a function of the angle-of-attack for the Eppler E330 airfoil and the Vireo. Right: The drag polar of the Vireo.

The relation $C_{Di} = KC_L^2$ implies that the total drag coefficient ($C_D = C_{Dp} + C_{Di}$) reaches a minimum when $C_L = 0$. In general, this is true only for symmetric airfoils. Since the Eppler E330 is a cambered, a more appropriate representation for C_D is [36]:

$$C_D = C_{D_{min}} + K\left(C_L - C_{L_{D,min}}\right)^2, \tag{2.9}$$

where $C_{D_{min}}$ is the minimum drag coefficient and $C_{L_{D,min}}$ is the lift coefficient at minimum drag. Since $C_{L_{D,min}}$ is unknown, it is estimated using XFOIL. In particular, XFOIL predicts that the Eppler E330 attains minimum drag around an angle-of-attack of zero. Hence, C_{L0} is used instead of $C_{L_{D,min}}$ in Equation (2.9). Next, $C_{D_{min}}$ is computed with the knowledge that $C_D = C_{Dp}$ when there is no lift-induced drag, i.e. $C_L = 0$. Applying this relation to Equation (2.9) yields the expression $C_{D_{min}} = C_{Dp} - KC_{L0}^2$. The resulting value for the Vireo is $C_{D_{min}} = 3.45 \times 10^{-2}$. Equation (2.9) is plotted as a drag polar on the right-side of Figure 2.12. The quadratic variation of the lift-induced drag is clearly seen. The red asterisk marks the point where minimum drag is attained, i.e. at *zero angle-of-attack*. The blue cross marks the point where the parasitic drag is the only contributor to the total drag, i.e. at the *zero-lift angle-of-attack*.

Stability and control derivatives

For modeling convenience, consider new axes that are denoted by $o_a x_a y_a z_a$. These axes are centered at AEROREF and are parallel to the stability axes $o_s x_s y_s z_s$. Let $\boldsymbol{F_{a,s}}$ and $\boldsymbol{M_{a,s}}$ denote the aerodynamic forces and moments, respectively, when resolved into $o_a x_a y_a z_a$. $\boldsymbol{F_{a,s}}$ consists of the drag force D, the side force Y, and the lift force L. $\boldsymbol{M_{a,s}}$ consists of the rolling moment l, the pitching moment m, and the yawing moment n. These forces and moments are further expressed using non-dimensional coefficients as:

$$\boldsymbol{F_{a,s}} = \begin{bmatrix} -D \\ Y \\ -L \end{bmatrix} = QS \begin{bmatrix} -C_D \\ C_Y \\ -C_L \end{bmatrix} \quad \text{and} \quad \boldsymbol{M_{a,s}} = \begin{bmatrix} l \\ m \\ n \end{bmatrix} = QS \begin{bmatrix} b\,C_l \\ \bar{c}\,C_m \\ b\,C_n \end{bmatrix}, \tag{2.10}$$

where $Q = \frac{1}{2}\rho V^2$ is the freestream dynamic pressure.

In general, the force and moment coefficients are nonlinear functions of 8 variables: the airflow angles (α, β), the angular velocity components (p, q, r), and the control inputs $(\delta_t, \delta_l, \delta_r)$. However, this thesis assumes that the aerodynamics are locally linear, and hence replaces the functions with their respective linear approximations. The reference condition for the linearization is defined by the 8 variables described above. Further, this thesis assumes that the longitudinal and lateral-directional dynamics are decoupled from each other [25]. Under these assumptions, the force and moment coefficients are expressed in terms of their respective stability and control derivatives as follows.

$$\begin{aligned}
C_D &= C_{D_{min}} + K\left(C_L - C_{L_{D,min}}\right)^2 + \left|C_{D_{\delta_l}}\delta_l\right| + \left|C_{D_{\delta_r}}\delta_r\right|, && (2.11)\\
C_Y &= C_{Y0} + C_{Y_\beta}\beta + C_{Y_{\hat{p}}}\hat{p} + C_{Y_{\hat{r}}}\hat{r} + C_{Y_{\delta_l}}\delta_l + C_{Y_{\delta_r}}\delta_r, && (2.12)\\
C_L &= C_{L0} + C_{L_\alpha}\alpha + C_{L_{\hat{q}}}\hat{q} + C_{L_{\delta_l}}\delta_l + C_{L_{\delta_r}}\delta_r, && (2.13)\\
C_l &= C_{l0} + C_{l_\beta}\beta + C_{l_{\hat{p}}}\hat{p} + C_{l_{\hat{r}}}\hat{r} + C_{l_{\delta_l}}\delta_l + C_{l_{\delta_r}}\delta_r, && (2.14)\\
C_m &= C_{m0} + C_{m_\alpha}\alpha + C_{m_{\hat{q}}}\hat{q} + C_{m_{\delta_l}}\delta_l + C_{m_{\delta_r}}\delta_r, && (2.15)\\
C_n &= C_{n0} + C_{n_\beta}\beta + C_{n_{\hat{p}}}\hat{p} + C_{n_{\hat{r}}}\hat{r} + C_{n_{\delta_l}}\delta_l + C_{n_{\delta_r}}\delta_r. && (2.16)
\end{aligned}$$

The expression given for C_D in Equation (2.11) is different from the expressions given for the other coefficients. The drag associated with α is implicitly included in C_L, thereby removing the need for a corresponding stability derivative. In addition, the drag associated with q is assumed to be negligible because of the absence of a horizontal tail [25]. Therefore, Equation (2.11) is essentially the same as Equation (2.9), except for the inclusion of the control derivatives associated with the left and right elevons. The values returned by AVL for $C_{D_{\delta_l}}$ and $C_{D_{\delta_r}}$ capture the lift-induced drag contributions due to the deflections of the elevons, as observed in the Trefftz plane [37]. In general, the sign of a term such as $C_{D_{\delta_l}}\delta_l$ depends on α as well as the direction of deflection of the elevon. However, for simplicity, it is assumed that an elevon deflection, by any amount and in either direction, results in an increase in C_D. Hence, the absolute values of $C_{D_{\delta_l}}\delta_l$ and $C_{D_{\delta_r}}\delta_r$ are considered. In addition, the C_L shown in Equation (2.11) refers to the value that is computed in Equation (2.13).

Equations (2.12) through (2.16) are the Taylor series expansions of the general nonlinear functions representing the coefficients, expressed up to the first degree. For example, the *zero-offset* C_{Y0} denotes the side force coefficient evaluated at the reference condition chosen for linearization. In addition, C_{Y_β} denotes the *stability derivative* $\frac{\partial C_Y}{\partial \beta}$ and $C_{Y_{\delta_l}}$ denotes the *control derivative* $\frac{\partial C_Y}{\partial \delta_l}$. A similar interpretation applies to all the other terms. The terms $(\hat{p}, \hat{q}, \hat{r})$ denote the non-dimensional angular velocity components, which are related to their respective dimensional counterparts as:

$$\hat{p} = \frac{pb}{2V}, \quad \hat{q} = \frac{q\bar{c}}{2V}, \quad \text{and} \quad \hat{r} = \frac{rb}{2V}. \tag{2.17}$$

Hence, all the stability and control derivatives have the units of rad^{-1}.

In general, the functions representing the force and moment coefficients are linearized at multiple reference conditions. This results in a schedule of zero-offsets, stability derivatives, and control derivatives that are collectively implemented as a lookup table. For aircraft such as the Vireo, variations in the stability and control derivatives, if any, will most likely arise from variations in the airflow angles. For simplicity, α and β are varied independently to create the set of reference conditions. First, nine equally-spaced values of α are chosen in the range $[-2, 14]\,^\circ$, while fixing the remaining seven variables at zero. Then, 11 equally-spaced values of β are chosen in the range $[-10, 10]\,^\circ$, while fixing the remaining seven variables at zero.

AVL is then used to compute the flow solution at each of the 20 reference conditions. Figure 2.13 shows the resulting stability and control derivatives. The top row shows the longitudinal stability and control derivatives as functions of α. The bottom row shows the lateral-directional stability and control derivatives as functions of β. In each of the six plots shown, the y-axis label uses $\star$ to denote all the applicable independent variables. Hence, the legend and the y-axis label are read in conjunction to interpret the derivative. In addition, the plots only show the control derivatives associated with δ_l, since those associated with δ_r have the same magnitude. The control derivatives of the left and right elevons have the same sign for the longitudinal coefficients and opposite signs for the lateral-directional coefficients. Finally, α and β are shown in degrees, while all the stability and control derivatives have the units of rad^{-1}.

The major observation from Figure 2.13 is that, apart from a few exceptions, most of the stability and control derivatives are largely invariant to changes in α and β. The coefficient of variation, which is defined as the ratio of the sample standard deviation to the sample mean, is used to quantify and compare the relative variability in the different stability and control derivatives. Table 2.5 lists the four highest coefficients of variation seen in Figure 2.13. There are two main observations from this table. First, $C_{D_{\delta_l}}$ has the highest coefficient of variation (a value of 71%) due to the sensitivity of the induced drag to variations in α. However, at a nominal flight condition wherein $\alpha \approx 4^\circ$ and $\delta_l \approx 6^\circ$, $C_{D_{\delta_l}}\delta_l$ is less than 8% of C_{Dp}. Since $C_{D_{\delta_l}}$ does not affect the total drag very much, its variability is assumed to be negligible. Second, despite having a coefficient of variation of 14%, $C_{l_{\delta_l}}$ is roughly equal to one-third of the mean value of $C_{l_{\hat{p}}}$. Since a non-zero angle-of-sideslip will induce a roll rate, it is assumed that the variation in $C_{l_{\delta_l}}$ is negligible for practical purposes. Finally, all the stability and control derivatives that are not listed in Table 2.5 have coefficients of variation that are less than 5%.

Table 2.5: Coefficients of variation of the different stability and control derivatives.

Stability or control derivative	$C_{D_{\delta_l}}$	$C_{l_{\delta_l}}$	$C_{n_{\delta_l}}$	$C_{Y_{\delta_l}}$	All others
Coefficient of variation	71%	14%	7.4%	6.5%	$< 5\%$

Since the majority of the stability and control derivatives are invariant to changes in the airflow angles, a lookup table is deemed unnecessary. Hence, the aerodynamic model only considers the reference condition $\alpha = \beta = 0$, which is shown in Figure 2.13 by the red-colored markers. As mentioned earlier, the remaining six independent variables are

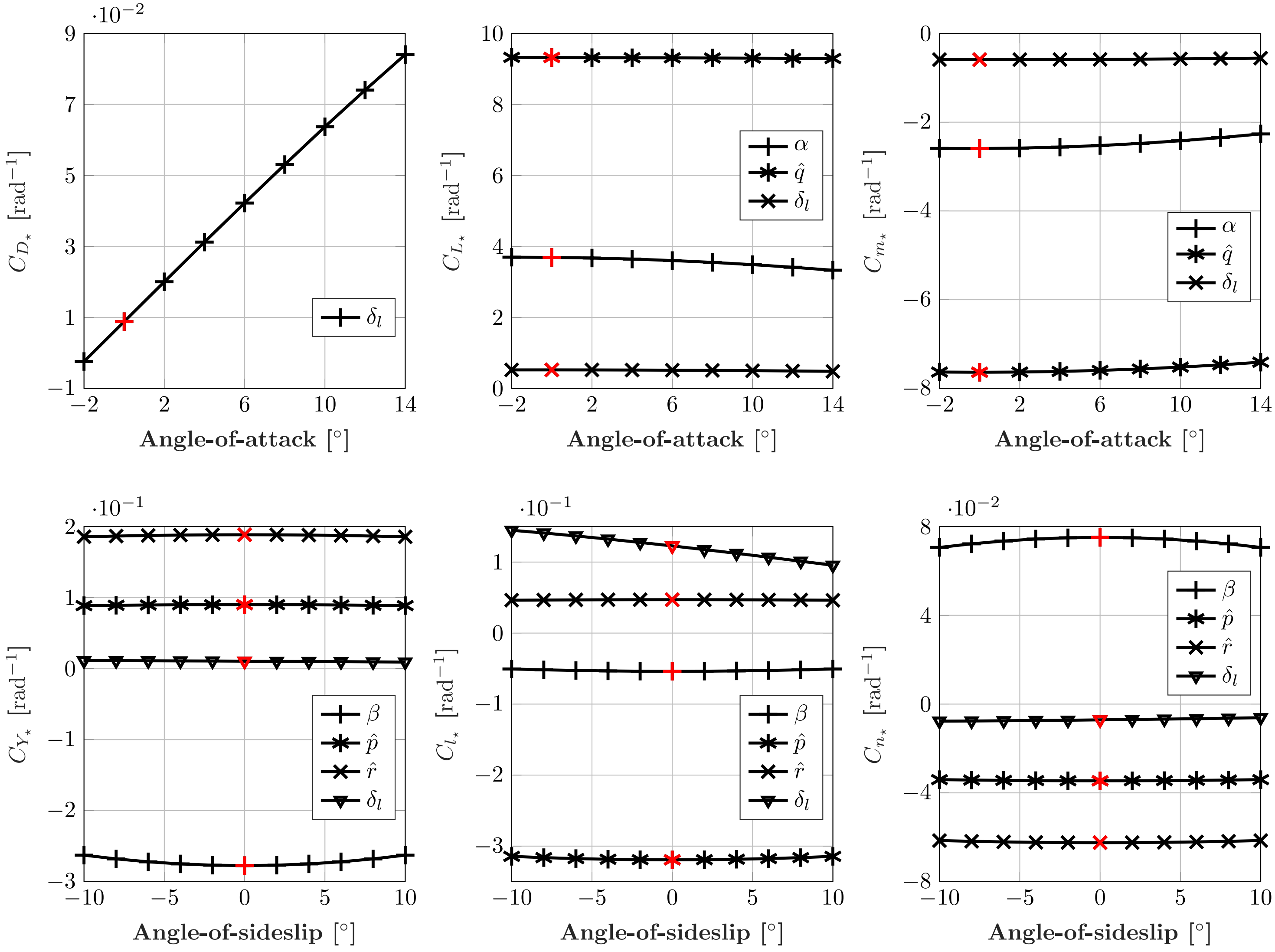


Figure 2.13: Top: The longitudinal stability and control derivatives as a function of the angle-of-attack. Bottom: The lateral-directional stability and control derivatives as a function of the angle-of-sideslip.

also equal to zero at this reference condition. In addition, the zero-offsets C_{Y0}, C_{l0}, and C_{n0} are all equal to zero, while $C_{L0} = 0.16$ and $C_{m0} = -5.4 \times 10^{-2}$. The resulting aerodynamic model is integrated into the model structure shown in Figure 2.5. As shown in the figure, the aerodynamic model uses x and u to compute the forces $\boldsymbol{F_a}$ and the moments $\boldsymbol{M_a}$ in the body axes. This happens over several steps. First, x is used to compute $(\alpha, \beta, \hat{p}, \hat{q}, \hat{r})$. Then, Equations (2.11) through (2.16) are used to compute the force and moment coefficients, which are then used in Equation (2.10) to compute $\boldsymbol{F_{a,s}}$ and $\boldsymbol{M_{a,s}}$. Next, $\boldsymbol{F_{a,s}}$ and $\boldsymbol{M_{a,s}}$ are translated from the AEROREF to the CG. Finally, they are rotated from the stability axes to the body axes using the DCM $C_{b \leftarrow s}$ (see the text following Equation (2.1)) to yield $\boldsymbol{F_a}$ and $\boldsymbol{M_a}$.

The low-fidelity aerodynamic model developed here captures the key flight dynamic characteristics of the aircraft. However, since VLM is based on potential flow theory, the current estimates of the stability and control derivatives are inaccurate. For example, VLM often underestimates the effectiveness of the control surfaces due to the inviscid flow assumption. Section 3.3 presents the aircraft system identification results, wherein flight test data is used to update the estimates of the stability and control derivatives.

2.4 Trim and Linearization

The geometry, inertia, propulsion, and aerodynamic models that are presented in Section 2.3 are integrated into a complete model, as shown in Figure 2.5. This model is *nonlinear* because the equations of motion contain trigonometric relations involving the Euler angles and terms such as V^2. This section uses the simulation environment to numerically trim and linearize this nonlinear model at various user-defined flight conditions. Although the system identification results are yet to be presented in Chapter 3, this section considers the final model, i.e. it includes a transfer function description of the elevon actuators (Section 3.2) as well as the updated stability and control derivatives (Section 3.3). In the following, the overline (e.g. $\bar{V}$) denotes a flight condition and the tilde (e.g. $\tilde{V}$) denotes the perturbation from a flight condition.

2.4.1 Trim Analysis

For the trim analysis, nine equally-spaced values of $\bar{\alpha}$ are chosen in the range $[-2, 14]\,^\circ$. This is the same discretization that is used in Figure 2.13. For each value of $\bar{\alpha}$ in this grid, the corresponding steady, wings-level, constant altitude, and constant airspeed flight condition is calculated using the `findop` function in Matlab. The left-side of Figure 2.14 shows the angle-of-attack versus the airspeed at each flight condition. As expected, low airspeeds correspond to high angles-of-attack and high airspeeds correspond to low angles-of-attack. The flight condition corresponding to the grid point $\bar{\alpha} = -2^\circ$ is infeasible and is thus not shown in Figure 2.14. This indicates that the high speed limit of the Vireo is around 20.5 $\mathrm{m\,s^{-1}}$ and corresponds to zero angle-of-attack. Further, reconsider the C_L-versus-α plot shown in Figure 2.12. This plot predicts that the Eppler E330 stalls at $\alpha \approx 11^\circ$. In order to be conservative, the stall angle-of-attack of the Vireo is considered to be approximately 10°. From Figure 2.14, the corresponding stall airspeed is around 12 $\mathrm{m\,s^{-1}}$. The flight control system uses the high speed and the stall airspeeds to implement a rudimentary flight envelope protection system.

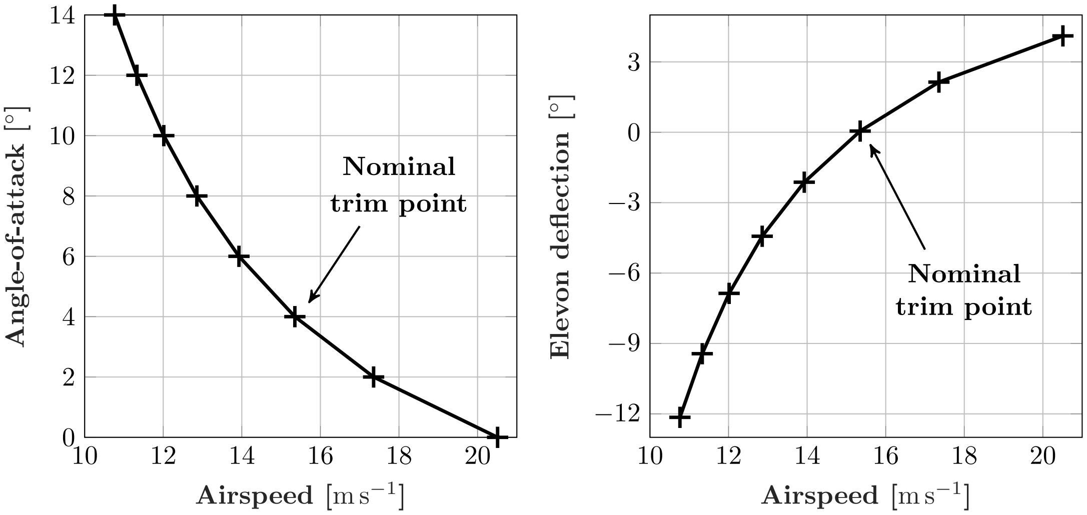


Figure 2.14: The steady, wings-level, constant altitude, and constant airspeed flight conditions of the Vireo. Left: The angle-of-attack versus the airspeed. Right: The elevon deflection versus the airspeed. The nominal cruise airspeed is 15.4 $\mathrm{m\,s^{-1}}$.

The right-side of Figure 2.14 shows the elevon deflection versus the airspeed at each flight condition. Note that the elevons are symmetrically deflected at the steady and wings-level flight conditions. As expected, low airspeeds correspond to large elevon deflections (trailing-edge up) and high airspeeds correspond to small elevon deflections. Extensive flight tests have yielded the nominal cruise airspeed of the Vireo to be around 15.4 $\mathrm{m\,s^{-1}}$. From Figure 2.14, this airspeed corresponds to $\bar{\alpha} = 4°$ and $\bar{\delta}_l = \bar{\delta}_r = 0.05°$. Although the nonlinear model predicts these values for the trim angle-of-attack and the trim elevon deflections, the flight tests yield slightly different numbers because of variations in the mass and the CG location of the aircraft. The remainder of this thesis refers to the airspeed of 15.4 $\mathrm{m\,s^{-1}}$ as the *nominal trim point*.

2.4.2 Linearization

The nonlinear model is linearized at each of the flight conditions shown in Figure 2.14. Figure 2.15 plots the poles of the linearized models in the s-plane. The polar grid indicates the isolines of constant damping ratios and natural frequencies. The poles are labeled as per the nomenclature of the classical longitudinal ($+$) and lateral-directional ($\times$) modes. Each pole is colored based on the airspeed at which the model is linearized. All the poles are located in the open left-half of the s-plane, thus implying that all the linearized models are stable. The following observations are made regarding the locus of the poles in Figure 2.15. As the airspeed increases, the natural frequency of the short period mode increases, while its damping ratio stays nearly the same. On the other hand, as the airspeed increases, the natural frequency of the phugoid mode decreases, while its damping ratio slightly increases. This observation is consistent with that predicted by Lanchester's phugoid model [25]. The dutch roll mode behaves similarly to the short period mode. The roll subsidence mode, which has no imaginary component, becomes more stable as the airspeed increases. Finally, the marginally stable spiral mode becomes slightly more stable as the airspeed increases.

At a given airspeed, the dynamics of the aircraft are locally described by an LTI model. However, the variation of the poles indicates that the corresponding LTI models also vary with the airspeed. Thus, the entire collection of linear models, which are parametrized on the airspeed, can be analyzed as a gridded linear parameter-varying (LPV) system [38]. The theory underpinning the analysis and control of LPV systems originated in the 1980's and 1990's [39–41]. The LPV description captures the variation

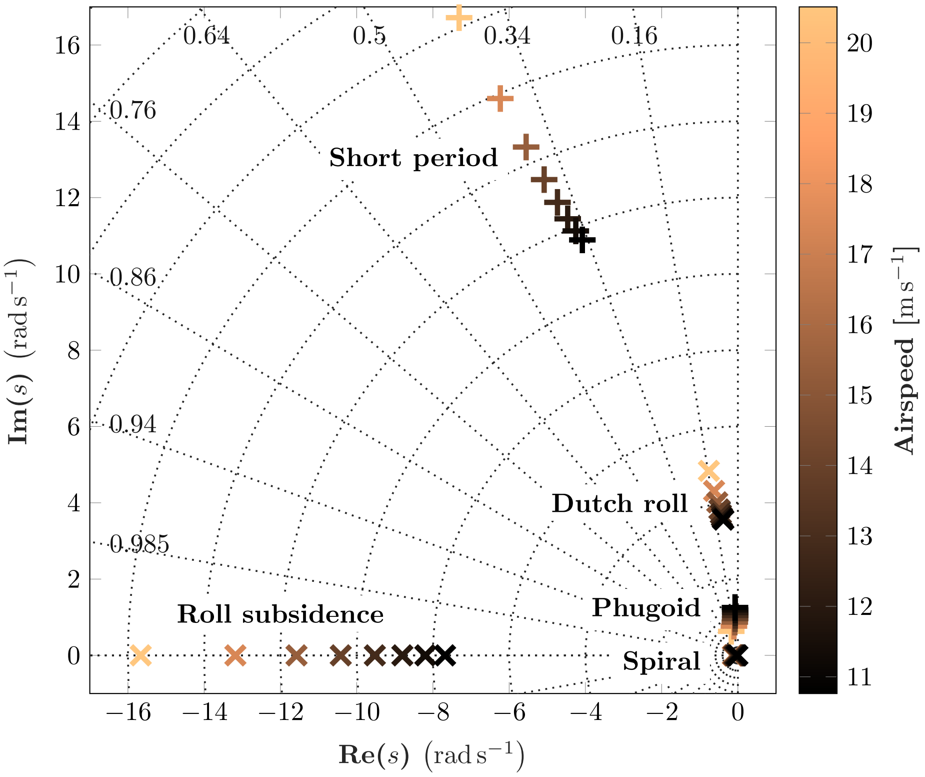


Figure 2.15: The poles of the linearized models in the s-plane. Each pole is colored based on the airspeed at which the model is linearized.

in the aircraft dynamics across flight conditions, while allowing for the use of well-studied analysis and design tools. LPVTools (a Matlab toolbox) implements algorithms for the modeling, analysis, and synthesis of LPV systems [42, 43]. The LTI model at the nominal trim is used in the design of the nominal controller (Chapter 4) and the fault-tolerant controller (Chapter 7). The LPV model is used in the design of the fault diagnosis algorithm (Chapter 6).

Chapter 3

Actuator and Aircraft System Identification

3.1 Theoretical Background

The main focus of this chapter is on the system identification of the actuator and the aircraft dynamics. Hence, this section reviews the relevant theoretical background on system identification, such as: frequency response and coherency (Section 3.1.2), model parametrization (Section 3.1.3), and prediction error minimization (Section 3.1.4). The theory and notation presented here is borrowed from the work of Ljung [44]. Some notation is modified to suit this thesis and is redefined where appropriate. Unless noted otherwise, this thesis uses SI units for all the dimensional variables.

3.1.1 Data Acquisition and Preparation

The experimental data is collected by a digital acquisition system and hence is recorded in discrete time. The input u and output y are observed at the sampling instants $t_k = kT_s$ for $k = 1, 2, \ldots, N$. Here, N denotes the total number of data samples in the experiment, k denotes the enumeration of the sequence, and t_k denotes the time (measured in s) at the k^{th} data sample. Section 2.2.2 mentioned that the flight computer on-board the Vireo experiences an inconsistent sample rate. To ease the system identification, the logged data is interpolated to a time base that uses a uniform sample time

T_s. The input and output sequences are denoted by $\{u(t_k)\}_1^N$ and $\{y(t_k)\}_1^N$, respectively, where $u(t_k) \in \mathbb{R}^{n_u}$ and $y(t_k) \in \mathbb{R}^{n_y}$. For brevity, these sequences will sometimes be simply denoted by u and y. Before being used for system identification, the data is prepared by trimming out extraneous time segments. Moreover, when identifying linear models, the sample mean is subtracted from u and y.

Although the experimental data is recorded in discrete-time, the actual input is a continuous-time signal $u(t)$ defined for all real t in the interval $[T_s, NT_s]$. Moreover, the model identification is done in continuous-time for the ease of simulation and control design. Hence, it is necessary to reconstruct $u(t)$ from $\{u(t_k)\}_1^N$ by assuming an intersample behavior (see Section 13.3 of [44]). The software drivers (Section 2.2.2) that control the actuators hold the command constant between frames. Hence, $u(t)$ is reconstructed using the zero order hold (ZOH) method, i.e. $u(t) = u(t_k)$ for $kT_s \leq t < (k+1)T_s$.

Finally, some of the model identification techniques require the data to be expressed in the frequency-domain. The Fast Fourier Transformation (FFT) is used to compute the spectral content $\{U(i\omega_k)\}_1^N$ and $\{Y(i\omega_k)\}_1^N$ of $\{u(t_k)\}_1^N$ and $\{y(t_k)\}_1^N$, respectively. Here, $i = \sqrt{-1}$ and $\omega \in [-\pi, \pi]$ denotes the frequency, measured in $\mathrm{rad\,s^{-1}}$.

3.1.2 Frequency Response and Coherency

System identification relies on several assumptions that are made about the underlying process, and hence the relationship between u and y. This thesis focuses on identifying linear models for the actuator and aircraft dynamics. Hence, the assumption of linearity is used to construct a frequency response function. The frequency response function describes the steady-state response of a system to sinusoidal input sequences. If u and y are related by some discrete-time transfer function $G(z)$, then the frequency response function is obtained by evaluating $G(z)$ on the unit circle described by $z = e^{i\omega}$. Hence, $G\left(e^{i\omega}\right)$ is complex-valued at any given frequency and can thus be decomposed into gain and phase components.

However, since $G(z)$ is not known a priori, the frequency response function is estimated using the Blackman-Tukey approach [44]. This approach uses the power and cross spectral densities of the input and output. The estimates of the power spectral densities of u and y are denoted by $\hat{\Phi}_u^N(\omega)$ and $\hat{\Phi}_y^N(\omega)$, respectively. The estimate of the cross-spectral density between u and y is denoted by $\hat{\Phi}_{yu}^N(\omega)$. The discrete-time *estimate* of the frequency response function $\hat{G}_N\left(e^{i\omega}\right)$ is then computed as:

$$\hat{G}_N\left(e^{i\omega}\right) = \frac{\hat{\Phi}_{yu}^N\left(\omega\right)}{\hat{\Phi}_u^N\left(\omega\right)}. \tag{3.1}$$

The spectral estimates $\hat{\Phi}_{yu}^N\left(\omega\right)$ and $\hat{\Phi}_u^N\left(\omega\right)$ are often non-smooth, especially at high frequencies. However, the true frequency response function is a smooth function of ω. Hence, frequency windowing is used to smooth the estimates of the power and cross-spectral densities (see Section 6.4 of [44]). At any given ω_k, the smoothing is done by considering the weighted sum of the estimates in a window around ω_k. The window size is selected by the user and controls the trade-off between bias and variance.

In addition to the frequency response function, the coherency spectrum $\hat{\kappa}_{yu}^N\left(\omega\right)$ between y and u is estimated. The coherency spectrum assumes values in the range $[0, 1]$ and can be viewed as a measure of the correlation between u and y [44]. For example, if $\hat{\kappa}_{yu}^N\left(\omega\right)$ is equal to 1 at a certain frequency, then there is perfect correlation between u and y at that frequency. The coherency spectrum is given by the expression:

$$\hat{\kappa}_{yu}^N\left(\omega\right) = \sqrt{\frac{\left|\hat{\Phi}_{yu}^N\left(\omega\right)\right|^2}{\hat{\Phi}_y^N\left(\omega\right)\hat{\Phi}_u^N\left(\omega\right)}}. \tag{3.2}$$

Note that $\hat{\Phi}_u^N\left(\omega\right)$ and $\hat{\Phi}_y^N\left(\omega\right)$ are always real-valued and $\hat{\Phi}_{yu}^N\left(\omega\right)$ is complex-valued.

3.1.3 Model Parametrization

The only assumption that is made in the construction of the frequency response function is the linearity of the underlying system. However, additional knowledge about the system is usually available. Such knowledge comes either from physics-based first principles or from past experience in modeling similar systems. In such cases, the additional knowledge is used to specify a model structure that is expressed in terms of one or more unknown parameters. The parameters are then estimated from the experimental input-output data.

Following [44], the model structure is denoted by $\mathcal{M}$ and the parameter vector is denoted by Θ. Each element of Θ varies within a user-specified range. If there are d

parameters, then Θ ranges over a subset $D_{\mathcal{M}}$ of $\mathbb{R}^d$. The set of models is thus:

$$\mathcal{M}^* = \{\mathcal{M}(\Theta) | \Theta \in D_{\mathcal{M}}\}, \tag{3.3}$$

where $\mathcal{M}(\Theta)$ denotes the particular model corresponding to Θ. Hence, $\mathcal{M}$ denotes a mapping from the parameter set $D_{\mathcal{M}}$ to the model set $\mathcal{M}^*$. The system identification of the actuator and aircraft dynamics utilize two types of model structures: black-box and grey-box. Since the models are identified in continuous-time, the description of the model structures given below is also in continuous-time.

Black-box model structure

Black-box models are flexible and accommodate a variety of systems, without looking into their internal details. A common black-box model structure is the transfer function. As compared to frequency response functions, which merely capture the gain and phase at each frequency, transfer functions are a good first step in obtaining a useful model. This chapter uses the generic, continuous-time transfer function form:

$$G_c(s, \Theta) = e^{-\tau s} \cdot \frac{b_0 s^n + b_1 s^{n-1} + \cdots + b_n}{s^n + a_1 s^{n-1} + \cdots + a_n}, \tag{3.4}$$

where $\{b_k\}_0^n$ and $\{a_k\}_1^n$ denote the numerator and denominator coefficients, respectively. In addition, τ denotes the pure time delay introduced by computational processes. Hence, $G_c(s, \Theta)$ is parametrized by $\Theta = [\tau, b_0, \ldots, b_n, a_1, \ldots, a_n]^T$.

The parameter set $D_{\mathcal{M}}$ is constrained by relying on prior knowledge about the system. For example, the order n of the transfer function is fixed a priori. In addition, the number of zeros of $G_c(s, \Theta)$ is sometimes known from first principles, thereby allowing some of the numerator coefficients to be set equal to zero.

Grey-box model structure

Often, additional knowledge about the model structure can be used to reduce the number of parameters down to what is actually unknown about the system. Such knowledge usually arises from physics-based first principles, and results in so-called grey-box models. Grey-box models are often parametrized by physically meaningful parameters, e.g.

mass, inertia, and aerodynamic coefficients. Since most laws of physics are expressed in continuous-time, using differential equations, grey-box models are commonly formulated as continuous-time state-space equations.

This chapter uses the generic, linear, continuous-time state-space model:

$$\dot{x} = A(\Theta)\, x + B(\Theta)\, u, \tag{3.5}$$

$$y = C(\Theta)\, x + D(\Theta)\, u, \tag{3.6}$$

where A, B, C, and D are parametrized by Θ. Moreover, the states contained in x are usually physically meaningful, e.g. positions, velocities, etc. Hence, $A(\Theta)$, $B(\Theta)$, $C(\Theta)$, and $D(\Theta)$ are highly structured. This structure is exploited to ensure that the total number of unknown parameters is far fewer than the total number of elements in the four matrices. In addition, since the parameters have physical meaning, reasonably good estimates of their range of variation are usually available. These range estimates are used to constrain the parameter set $D_{\mathcal{M}}$.

After formulating either the black-box or the grey-box model structure $\mathcal{M}$, the goal changes to selecting the parameter Θ that best explains the input-output data. Effectively, the information contained in $\{u(t_k)\}_1^N$ and $\{y(t_k)\}_1^N$ is used to select a proper member $\hat{\Theta}$ from the parameter set $D_{\mathcal{M}}$, and hence a proper member $\mathcal{M}(\hat{\Theta})$ from the model set $\mathcal{M}^*$. This problem can be solved by several different parameter estimation methods (see Chapter 7 in [44]). This chapter, however, will focus on *prediction error minimization* (PEM) methods.

3.1.4 Prediction Error Minimization

At its core, any PEM method works by minimizing the error between the measured output and the so-called predicted output. The difference is called the *prediction error*. In general, prediction methods forecast the model output k steps ahead into the future using the current and past values of u and y (see Chapter 3 of [44]).

This chapter will exclusively deal with the 1-step ahead prediction. For a given parameter Θ, the particular model $\mathcal{M}(\Theta)$ is used to predict the model output $\hat{y}(t_j|\Theta)$ at time $t_j = jT_s$ using the sequences $\{u(t_k)\}_1^{j-1}$ and $\{y(t_k)\}_1^{j-1}$. The predicted output is denoted by $\hat{y}(t_j|\Theta)$ to emphasize its dependence on Θ. If $y(t_j) \in \mathbb{R}^{n_y}$ denotes the measured output at time t_j, the prediction error is defined as:

$$e\left(t_j|\Theta\right) = y\left(t_j\right) - \hat{y}\left(t_j|\Theta\right). \tag{3.7}$$

If the experiment contains a total of N data samples, the prediction error is computed at each sampling instant t_j, thereby resulting in the error sequence $\left\{e\left(t_j|\Theta\right)\right\}_1^N$. The prediction error sequence is thus a measure of how well a particular model can predict future outputs. Some problems may prioritize small prediction errors over specific frequency ranges. For example, in modeling an actuator, it may be more important to minimize prediction errors within its bandwidth than outside. In general, $\left\{e\left(t_j|\Theta\right)\right\}_1^N$ is filtered through a user-specified, stable, linear filter. Filtering the prediction errors is the same as filtering the recorded input-output data. Hence, the filter is included when defining the model structure $\mathcal{M}$.

The goal is then to minimize the filtered prediction error sequence $\left\{\bar{e}\left(t_j|\Theta\right)\right\}_1^N$, where the overline distinguishes it from the unfiltered sequence. Each element of the sequence is a vector in $\mathbb{R}^{n_y}$. The size of this sequence can, in general, be measured using any vector norm. This thesis uses the quadratic norm and results in the cost function:

$$V\left(\Theta\right) = \frac{1}{N}\sum_{j=1}^{N} \bar{e}\left(t_j|\Theta\right)^T \bar{e}\left(t_j|\Theta\right). \tag{3.8}$$

For a given model structure $\mathcal{M}$ and input-output sequences $\left\{u\left(t_k\right)\right\}_1^N$ and $\left\{y\left(t_k\right)\right\}_1^N$, the cost function $V\left(\Theta\right)$ is a well-defined scalar-valued function of the parameter Θ. The common objective of the PEM algorithms is to minimize $V\left(\Theta\right)$ over the set of parameters. The optimal estimate of the parameter is then defined as:

$$\hat{\Theta} = \underset{\Theta \in D_{\mathcal{M}}}{\arg\min}\, V\left(\Theta\right). \tag{3.9}$$

In general, the cost function $V\left(\Theta\right)$ cannot be minimized by analytical methods. Thus, PEM algorithms solve the problem numerically via iterative search methods. Such search methods first initialize the parameters, and then update them. The parameter update is conducted by Newton algorithms, which determine the search direction from the values of the function, its gradient, and its Hessian. However, Newton algorithms only guarantee the convergence of the solution to a local minimum. The final solution is thus highly dependent on the initial parameter values. Hence, it is important to choose good initial parameter values. For grey-box models that are parametrized by physically meaningful parameters, physical insight or alternative modeling techniques may help in choosing the initial parameter values. For black-box models, several initialization methods exist. These are summarized in Section 10.5 of [44].

After estimating the optimal parameter $\hat{\Theta}$, the model is identified as $\mathcal{M}\big(\hat{\Theta}\big)$. Various model quality metrics exist in the literature to evaluate the quality of the identified model, or compare different model structures (see Section 16.4 in [44]). This thesis uses the *normalized root mean squared error* (NRMSE) as the main metric for evaluating how well the identified model $\mathcal{M}\big(\hat{\Theta}\big)$ matches the input-output data. The NRMSE is denoted by η and is defined as:

$$\eta = 100 \cdot \left(1 - \frac{\left\|E\big(\hat{\Theta}\big)\right\|_F}{\left\|Y - \bar{Y}\right\|_F}\right), \tag{3.10}$$

where, $E\big(\hat{\Theta}\big) = \big[\bar{e}\big(t_1|\hat{\Theta}\big), \ldots, \bar{e}\big(t_N|\hat{\Theta}\big)\big]$ and $Y = [y\,(t_1)\,, \ldots, y\,(t_N)]$ are matrices in $\mathbb{R}^{n_y \times N}$ formed from the filtered prediction error $\{\bar{e}\big(t_j|\hat{\Theta}\big)\}_1^N$ and output $\{y\,(t_j)\}_1^N$ sequences. Moreover, $\bar{Y}$ is a vector in $\mathbb{R}^{n_y}$ that is formed by computing the mean of each row of Y and $\|\cdot\|_F$ denotes the Frobenius norm in Equation (3.10). The value of η varies between $-\infty$ (bad fit) and 100 (perfect fit). If $\eta = 0$, then the model fits the data no better than the sample mean of the output.

Finally, new sets of input-output data are collected. Model validation is performed by simulating $\mathcal{M}\big(\hat{\Theta}\big)$ using the new input data and comparing the simulated outputs against the measured outputs (Chapter 16 in [44]). The System Identification Toolbox of Matlab [45] implements the algorithms described in this section. In the next section, the theoretical background presented here is applied to identify an actuator model.

3.2 Actuator System Identification

This thesis specifically considers the detection and tolerance of control surface faults. Since the control surfaces are actuated by servo motors, it is important to model their dynamics. To model the servos, system identification experiments are conducted to record their response to known input commands. The following subsections describe the apparatus, test procedure, test results, and system identification.

3.2.1 Apparatus

The apparatus that is used to characterize the servo is shown in Figure 3.1. It consists of a long metallic shaft that is supported on bearings. On one end of the shaft is an optical encoder with 5000 slots per rotation. This yields a resolution of 0.072° per slot. On the other end of the shaft is a mechanism for mounting the servo. The servo is mounted such that its arm is aligned and attached to the metallic shaft. The position of the metallic shaft is thus an accurate substitute for the position of the servo arm. The particular servo that is used on the Vireo (KST DS135MG) is pictured in the inset, shown at the top-right corner of Figure 3.1. This servo is controlled via pulse-width modulation (PWM) and the reference command is the pulse-width (measured in µs). The test is conducted by sending a time-series of known pulse widths to the servo and measuring the servo position using the shaft encoder. A data acquisition system collects and records the data generated by the shaft encoder.

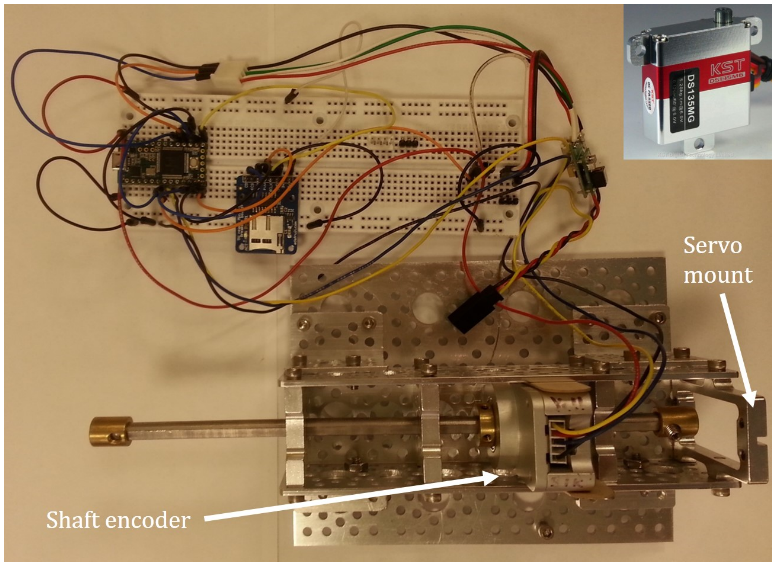


Figure 3.1: The apparatus used to characterize the servo, which is pictured in the inset.

3.2.2 Test Procedure

The test is divided into three components, each of which has a specific objective and is designed to uncover a specific characteristic of the servo.

1. **Hysteresis test:** The primary test objective is characterizing the hysteresis in the servo. The test involves sending pulse-widths that are linear and piecewise monotonic, with respect to time, to the servo and measuring the shaft position. First, the servo is initialized at the center of its deflection range. Then, the servo is commanded to deflect to one end of its physical limit. Next, the servo is commanded to deflect to the other end of its physical limit. Finally, the servo is commanded back to its initial position at the center of its deflection range.

   All through the test, the servo is commanded at a constant angular rate. The test yields a graph of the measured servo position δ_m versus the commanded pulse-width. This graph is used to compute the free-play in the servo as well as the mapping between the commanded pulse-width and the commanded servo position δ_c. This mapping effectively functions as a calibration and allows the subsequent tests to be conducted by directly commanding a servo position.

2. **Step response test:** The primary test objective is characterizing the angular rate limit of the servo. The test involves sending a step reference command to the servo, which causes its angular velocity $\dot{\delta}_m$ to saturate. The rate limit is estimated via the slope of the graph of the angular displacement versus time.

3. **Chirp response test:** The primary test objective is characterizing the frequency response of the servo. The test involves sending a continuous swept-sine, or chirp, command to the servo and measuring its response. The continuous swept-sine command is expressed as a function of time as:

   $$\delta_c\left(t\right) = A\cos\left(\omega_i\left(t\right)t\right), \tag{3.11}$$

   where A and $\omega_i\left(t\right)$ denote the amplitude and the instantaneous frequency, respectively. The instantaneous frequency of the sine wave is swept linearly over time as: $\omega_i\left(t\right) = \omega_1 + \frac{\left(\omega_2-\omega_1\right)}{2T_{ch}}t$, where ω_1, ω_2, and T_{ch} denote the start frequency, the end frequency, and the sweep duration, respectively.

3.2.3 Test Results

In all the tests, the servo is powered at 5 V and at a pulse repetition rate of 50 Hz.

Hysteresis test

For the hysteresis test, the angular position output of the shaft encoder is logged at a rate of 100 Hz. The test begins by initially commanding the servo at a pulse-width of 1520 µs. At this pulse-width, the servo position is defined as zero. All other servo positions are then referenced to this baseline. The servo is then commanded using the pulse-width profile shown on the left in Figure 3.2. The profile consists of three piecewise linear segments and lasts for a total duration of 11.4 s. First, the pulse-width is increased from 1520 µs to 1806 µs. Then, the pulse-width is decreased to 1240 µs. Finally, the pulse-width is increased back to its starting value of 1520 µs. This profile causes the servo to first deflect to 30°, then to −30°, and finally back to zero.

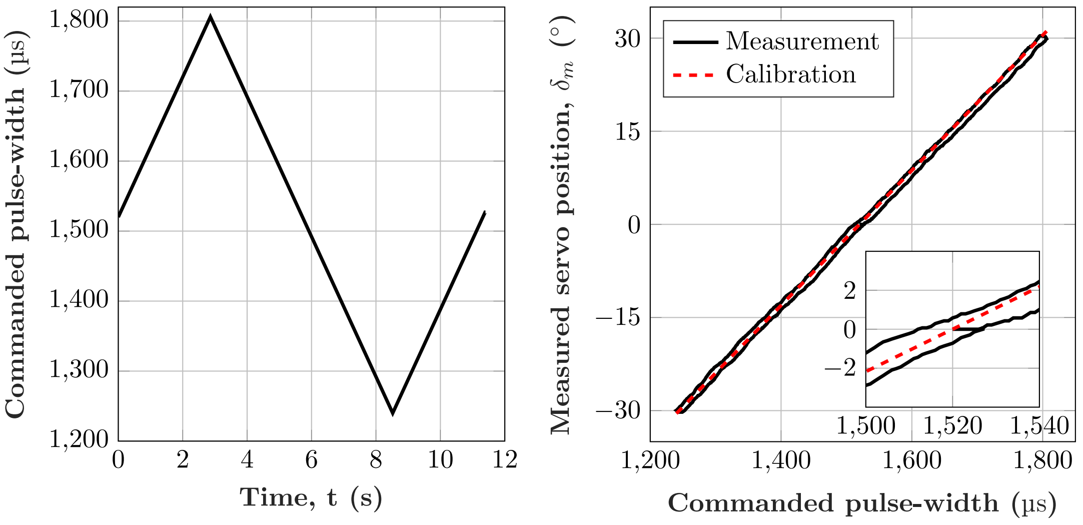


Figure 3.2: Left: The time history of the commanded pulse-width for the hysteresis test. Right: The graph of the measured servo position versus the commanded pulse-width, showing the experimental data and the linear calibration.

A plot of the measured servo position versus the commanded pulse-width is shown on the right in Figure 3.2. The inset shows a close-up view of the center of servo deflection range, from 1500 µs to 1540 µs. The solid line shows the experimental data. As seen from this plot, the measured servo position during the upward and downward sweeps of the pulse-widths are separated by a few degrees. This separation is due to the free-play in the internal mechanisms of the servo. By computing the average vertical separation between the two solid lines, the free-play is estimated to be 1.45°.

In addition to free-play, the graph of the measured servo position versus the commanded pulse-width is used to compute a linear mapping between these two quantities. The horizontal intercept is computed such that it passes through 1520 µs at the zero position of the servo. The slope is computed using the extremities of the servo deflection at $-30°$ and $30°$. The final linear mapping has the form $\delta_m = 0.11\,(x - 1520)$, where x is the commanded pulse-width. It is shown by the dashed line in the right-side plot in Figure 3.2. This linear mapping is used as a calibration between the commanded pulse-width and the commanded servo position. Hence, in all subsequent tests, the inverse mapping is used to directly command a servo position in degrees.

Step response test

For the step response test, the angular position output of the shaft encoder is logged at a rate of 2000 Hz. Figure 3.3 shows the time history of the command sent to the servo (solid line) and its response (dashed line). In particular, the test begins by commanding the servo to $-30°$. At 0.1 s, the servo is instantaneously commanded to a position of 30°, as shown by the solid line. The servo takes approximately 0.23 s to traverse the 60° step. At 1.1 s, the servo is instantaneously commanded back to $-30°$. From the slope of the plot of the servo response versus time, the rate limit is estimated as $338°\,\mathrm{s}^{-1}$.

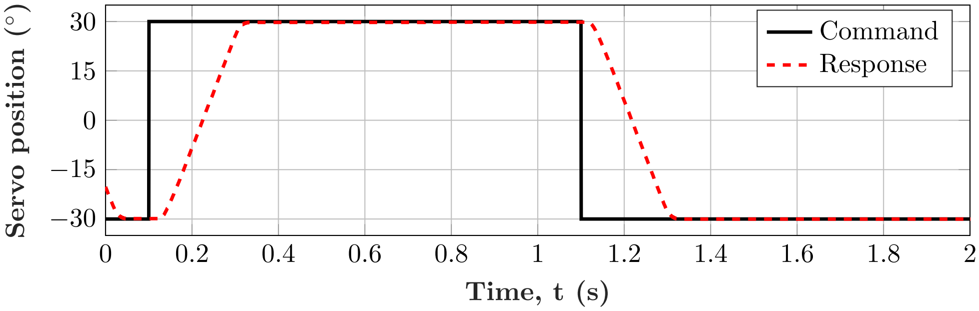


Figure 3.3: The time history of the step command and the servo response.

Chirp response test

For the chirp response test, the angular position output of the shaft encoder is logged at a rate of 200 Hz. The chirp command is designed with an amplitude of 5° and spans the frequency range $[1, 20]$ Hz over a period of 50 s. The total number of samples is $10,000$. The servo response, as measured by the shaft encoder, is shown in Figure 3.4. Note that the amplitude of the servo response rolls off as the instantaneous frequency of the chirp command increases. The results of the chirp response test are used to identify a transfer function model for the servo, as described next.

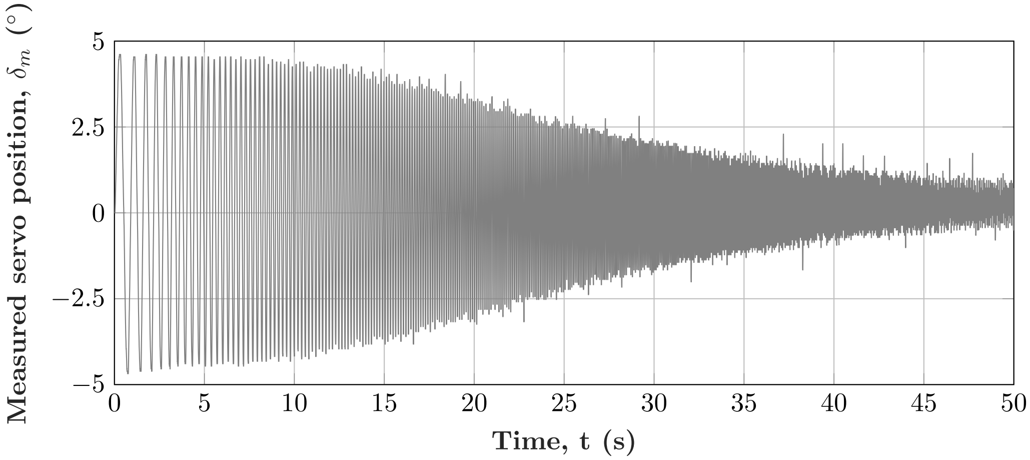


Figure 3.4: The servo response to a 5° amplitude chirp command spanning $[1, 20]$ Hz.

3.2.4 System Identification

The primary purpose of modeling the actuators is to incorporate the models during control design. Since a black-box transfer function model serves this purpose, a physics-based grey-box model is not pursued. The theoretical concepts introduced in Section 3.1 are applied to the chirp response data to obtain such a model. The input u corresponds to the commanded servo position δ_c and the output y corresponds to the measured servo position δ_m. First, the power and cross spectral densities of δ_c and δ_m are computed, and used in Equations (3.1) and (3.2) to construct the frequency response function and coherency spectrum, respectively.

Next, a transfer function is fitted to the frequency response function. The fitting is done using the prediction error minimization method described in Section 3.1.4. To aid the fitting, key transfer function properties are fixed a priori. In general, the order of the transfer function need not be any higher than that required to capture the key characteristics of the frequency response function. From past experience with modeling servo dynamics, it is expected that a transfer function with 2 poles, no zeros, and unit DC gain will offer a good match. Unit DC gain is used because the servo is already calibrated from the commanded pulse-width to the commanded servo position as part of the hysteresis test. In addition, computational processes during the experiment will add some time delay. Hence, the final form of the model is:

$$G_a\left(s, \Theta\right) = e^{-\tau_a s} \cdot \frac{\omega_a^2}{s^2 + 2\zeta_a \omega_a s + \omega_a^2}, \tag{3.12}$$

where τ_a denotes the time delay and the denominator coefficients are expressed in terms of the damping ratio ζ_a and the natural frequency ω_a. The prediction error is filtered using a fifth-order Butterworth filter with a bandpass range of $[1, 20]$ Hz. This prioritizes the actual frequency range spanned by the chirp command.

Estimating the time delay and the transfer function coefficients simultaneously is a difficult problem. Hence, the time delay is identified independently by comparing ARX models with different fixed delays. The Matlab function `delayest` is used to estimate the time delay as $\tau_a = 0.025$ s. This is equivalent to 5 missed frames in the chirp response test. With τ_a fixed, the remaining unknown parameters are ζ_a and ω_a. The prediction error minimization, implemented using the Matlab function `tfest`, results in: $\zeta_a = 0.77$ and $\omega_a = 62.8$ rad s^{-1}. The corresponding NRMSE is 81.23%.

Figure 3.5 shows the frequency response function and coherency spectrum of the chirp response test (solid line). The magnitude of the frequency response is 0 dB for low frequencies and rolls off above 50 rad s^{-1}. The phase is close to zero for low frequencies, but rapidly rolls off at high frequencies due to the presence of the time delay. The coherence is greater than 0.99 in the range $[6, 90]$ rad s^{-1}, indicating a high level of correlation between u and y. The Bode plot of the identified transfer function G_a is shown on the same plot using the dashed line. The phases of the transfer function and the frequency response function agree very well over the frequency range of interest. On the other hand, the magnitudes agree only below 50 $\mathrm{rad\,s^{-1}}$. In particular, the frequency response has lower gain than the transfer function at high frequencies. This is because the chirp command starts exceeding the rate limit of $338^\circ\,\mathrm{s^{-1}}$ for frequencies greater than 10.76 Hz $\left(= 67.6\ \mathrm{rad\,s^{-1}}\right)$. In order to verify this, note from Equation (3.11) that the maximum commanded rate of deflection at any instant during the chirp is $-A\omega_i\,(t)$. From Figure 3.5, it is evident that the bandwidth of the actuator is 57 $\mathrm{rad\,s^{-1}}$, where the bandwidth is defined as the frequency at which the magnitude first crosses −3 dB. Since the rate limit is exceeded only above the actuator bandwidth, the identified transfer function is acceptable for the purpose of control design.

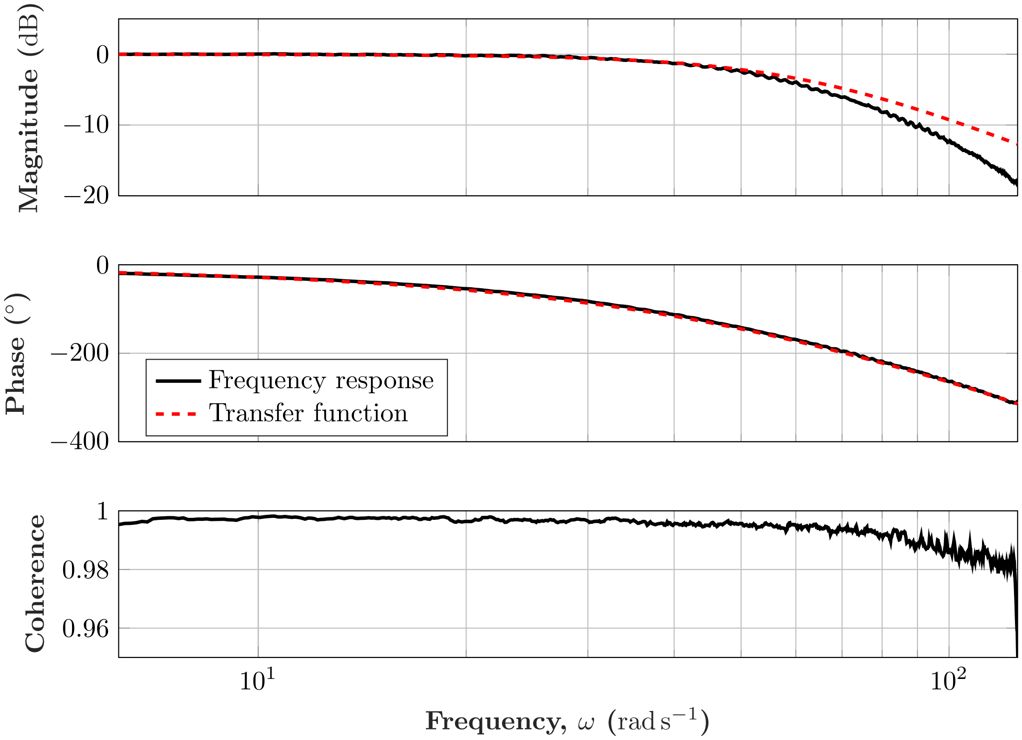


Figure 3.5: The frequency response function and coherency spectrum obtained from the chirp response test. The identified transfer function is shown using the dashed line.

Finally, the experimentally identified servo characteristics are integrated into the simulation environment. The servo is modeled as the second-order transfer function shown in Equation (3.12), subject to the position limits $[-30, +20]^\circ$ and the rate limits

$\pm 338^\circ\,\mathrm{s}^{-1}$. The time delay $\tau_a = 0.025$ s was based on the test apparatus described in Section 3.2.1. Since it is difficult to separate the delay contributions of the servo and the test apparatus, Section 3.3.8 revisits the problem of time delay estimation using flight test data. The revised estimate encompasses delays in the actuators, flight computer, and sensors, and is thus suitable for use in the control design.

3.3 Aircraft System Identification

The first principles nonlinear model that is developed in Section 2.3 is a good starting point for understanding the flight dynamics of the aircraft. However, the model parameters are uncertain due to the modeling approximations. In particular, the actuator, geometry, and propulsion models are experimentally-derived, and are hence considered to be accurate. However, the inertia and aerodynamic models are developed using approximate models and computational tools. The aerodynamic model, in particular, is prone to model uncertainty due to the assumptions made by VLM. This section considers an experimental approach to obtaining more accurate estimates for some of the model parameters. In particular, flight experiments are conducted to excite the dynamics of the aircraft. The concepts introduced in Section 3.1 are then invoked to identify black-box models and grey-box models for the key input-output relationships. In order to provide context to this exercise, some related work is introduced next.

3.3.1 Introduction

The literature on system identification is vast. Several textbooks provide the mathematical background underpinning system identification theory [44, 46]. Section 3.1 used the work of [44] to review the system identification concepts that are relevant to this thesis. In addition, several textbooks apply system identification concepts specifically to fixed-wing and rotary-wing aircraft [47]. These include both time-domain and frequency-domain methods [48, 49]. Recent literature has focused on system identification for small unmanned aircraft, including helicopter [50, 51], fixed-wing [52, 53], and multicopter [54] platforms. Software tools are also available to automate the aircraft system identification process [44, 47].

This thesis makes two main assumptions for the aircraft system identification. First, at a constant airspeed, the dynamics of the aircraft are assumed to be linear time-invariant (LTI). Second, the longitudinal and lateral-directional dynamics are assumed to be decoupled from each other. The system identification process largely follows [52]. Parameterized LTI models are constructed (Section 3.3.2) and evaluated at the nominal trim point (Section 3.3.3). This yields information on the key flight dynamic modes and the parameters that impact them. Flight experiments are designed within specific frequency ranges in order to excite these key modes (Section 3.3.4). The flight test data is used to identify the longitudinal (Section 3.3.5) and the lateral-directional (Section 3.3.6) models and validate them (Section 3.3.7). The data is also used to estimate the time-delay (Section 3.3.8) and model it using a low-order system (Section 3.3.9).

3.3.2 Parametrized LTI Models

The first assumption (that the aircraft dynamics are LTI) requires the linearized equations of motion. Although the simulation environment can trim and linearize the first principles nonlinear model, this process is numerical and does not shed light on the parametric dependencies of the linearized model. Hence, this section presents parametrized models that are developed by trimming and linearizing the nonlinear equations of motion (see Equations (2.2) and (2.3)) at some steady, wings-level, constant altitude, and constant airspeed flight condition [26, 48]. As a matter of semantics, the transfer function and the state-space forms of these LTI models are referred to as the black-box and the grey-box models, respectively. As before, the overline denotes a flight condition and the tilde denotes the perturbation from a flight condition. As such, the states of the linear models represent perturbations and should be marked by tildes. However, for brevity, the tilde is dropped in the remainder of this thesis. The second assumption (that the longitudinal and lateral-directional dynamics are decoupled) requires a corresponding notion of decoupled inputs. This is achieved by reverting to the traditional notion of the elevator and the aileron, as described next.

Input decoupling

Section 2.3.1 defined the input vector u to include the throttle setting δ_t, the left elevon deflection δ_l, and the right elevon deflection δ_r. The assumptions made in Section 2.3.4 imply that the throttle only affects the longitudinal dynamics. However, as seen from

Figure 2.13, the left and right elevons affect all the force and moment coefficients. Since an individual elevon excites both the longitudinal and lateral-directional dynamics, the system identification process reverts to the traditional notion of the elevator δ_e and the aileron δ_a. The elevons are related to the elevator and the aileron via the transformation:

$$\begin{bmatrix} \delta_e \\ \delta_a \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \delta_l \\ \delta_r \end{bmatrix}. \tag{3.13}$$

Equation 3.13 essentially decomposes the left and right elevon deflections into a symmetric elevator component δ_e and an anti-symmetric aileron component δ_a. The elevator and the aileron are *virtual* inputs since they do not correspond to any physical control surface on the Vireo. However, they enable the decoupling of the models because δ_e only affects the longitudinal dynamics and δ_a only affects the lateral-directional dynamics. The remainder of this thesis uses $T_{ea \leftarrow lr}$ to denote the 2-by-2 matrix in Equation (3.13). The inverse of this matrix is denoted by $T_{lr \leftarrow ea}$ and is used to transform the elevator and aileron deflections to equivalent elevon deflections.

Moreover, the transformations $T_{ea \leftarrow lr}$ and $T_{lr \leftarrow ea}$ also relate the elevon deflection commands (δ_{lc} and δ_{rc}) to the elevator command δ_{ec} and the aileron command δ_{ac}. This is useful during system identification since excitation commands are designed in terms of δ_{ec} and δ_{ac}. $T_{lr \leftarrow ea}$ is used to transform these excitation commands to δ_{lc} and δ_{rc}, which are sent to the servo motors controlling the elevons.

Finally, the actuator model G_a that was developed in Section 3.2.4 relates the elevon deflection commands (δ_{lc} and δ_{rc}) to the corresponding actual elevon deflections (δ_l and δ_r). Since Equation (3.13) is a linear transformation, G_a also relates the elevator command to its response ($\delta_{ec} \rightarrow \delta_e$) and the aileron command to its response ($\delta_{ac} \rightarrow \delta_a$). The usefulness of these relationships will become evident shortly.

Dimensional stability and control derivatives

The grey-box LTI models are derived by linearizing the nonlinear equations of motion shown in Figure 2.5. Section 2.3.1 defined the state vector x using the angular velocity components (p, q, r) and the airspeed components (u, v, w) expressed in the *body axes.* Since the linearized equations of motion use the same state vector, the grey-box LTI models are parametrized by *dimensional* stability and control derivatives that are also expressed in the body axes [26, 48]. In order to define these derivatives, the aerodynamic forces and moments are resolved in the body axes as $\boldsymbol{F_a} = (X, Y, Z)$ and $\boldsymbol{M_a} = (L, M, N)$, respectively[1]. Using generic variables, the dimensional stability and control derivatives are defined as $A_b = \frac{\partial A}{\partial b}$ where $A \in \{X, Y, Z, L, M, N\}$ and $b \in \{u, v, w, p, q, r, \delta_t, \delta_e, \delta_a\}$. The assumption that the longitudinal and lateral-directional dynamics are decoupled restricts the allowable combinations of A and b.

Finally, Table 5.1 in [26] provides expressions that relate the dimensional stability and control derivatives to their non-dimensional counterparts, shown in Equations (2.11) through (2.16). However, since the dimensional derivatives are expressed in the $o_b x_b y_b z_b$ axes and the non-dimensional derivatives are expressed in the $o_a x_a y_a z_a$ axes, transforming from one type to the other also involves a translation and a rotation.

Longitudinal dynamics

The longitudinal dynamics are affected by the inputs δ_t and δ_e. Since Section 2.3.4 already developed a propulsion model using experimental data, the system identification focuses on the pitch rate response to the elevator input. The principles of flight dynamics inform the parametrization of the black-box and the grey-box LTI models. In particular, the longitudinal dynamics are composed of the short-period and the phugoid modes, which contribute a total of four poles. In addition, the pitch rate output is associated with three zeros, including a zero on the imaginary axis [25]. Thus, the elevator-to-pitch rate response is described by the black-box model [25]:

$$G_{lon}(s, \Theta) = \frac{q(s)}{\delta_e(s)} = \frac{k_q s \left(s + z_{\theta_1}\right)\left(s + z_{\theta_2}\right)}{\left(s^2 + 2\zeta_p \omega_p s + \omega_p^2\right)\left(s^2 + 2\zeta_s \omega_s s + \omega_s^2\right)}, \tag{3.14}$$

where Θ includes all the parameters to be estimated, as per the notation of Section 3.1.3. The denominator is expressed in terms of the damping ratio ζ and the natural frequency ω of the $p$hugoid and the $s$hort period modes. The numerator is expressed in terms of the gain k_q and the zeros z_{θ_1} and z_{θ_2}.

The equivalent grey-box LTI model in the time-domain is given by [26]:

$$\dot{x}_{lon} = A_{lon}(\Theta)\, x_{lon} + B_{lon}(\Theta)\, \delta_e, \tag{3.15}$$

[1] Note that L here denotes the rolling moment and not the lift force.

where the state is $x_{lon} = [u, w, q, \theta]^T$ and the matrices are:

$$A_{lon}(\Theta) = \begin{bmatrix} X_u & X_w & X_q - \bar{w} & -g\cos\bar{\theta} \\ Z_u & Z_w & Z_q + \bar{u} & -g\sin\bar{\theta} \\ M_u & M_w & M_q & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \text{and} \quad B_{lon}(\Theta) = \begin{bmatrix} X_{\delta_e} \\ Z_{\delta_e} \\ M_{\delta_e} \\ 0 \end{bmatrix}. \tag{3.16}$$

The A_{lon} and B_{lon} matrices are populated with the dimensional stability and control derivatives, the trim variables $\bar{u}$, $\bar{w}$, and $\bar{\theta}$, and the acceleration due to gravity g. The output equation, which is not shown, simply picks q from the state x_{lon}.

Lateral-directional dynamics

The system identification for the lateral-directional dynamics focuses on the roll rate response to aileron input. The principles of flight dynamics inform the parametrization of the black-box and the grey-box LTI models. In particular, the lateral-directional dynamics are composed of the roll subsidence, the spiral, and the dutch roll modes, which contribute a total of four poles. In addition, the roll rate output is associated with three zeros, including a zero on the imaginary axis [25]. Thus, the aileron-to-roll rate response is described by the black-box model [25]:

$$G_{lat}(s, \Theta) = \frac{p(s)}{\delta_a(s)} = \frac{k_p s \left(s^2 + 2\zeta_\phi \omega_\phi s + \omega_\phi^2\right)}{\left(s + T_s^{-1}\right)\left(s + T_r^{-1}\right)\left(s^2 + 2\zeta_d \omega_d s + \omega_d^2\right)}, \tag{3.17}$$

where Θ includes all the parameters to be estimated, as per the notation of Section 3.1.3. The denominator is expressed in terms of the damping ratio ζ_d and the natural frequency ω_d of the dutch roll mode and the time constants of the spiral and the roll subsidence modes. The numerator is expressed in terms of the gain k_p and parameters ζ_ϕ and ω_ϕ.

The equivalent grey-box LTI model in the time-domain is given by [26]:

$$M_{lat}\dot{x}_{lat} = \bar{A}_{lat}(\Theta) x_{lat} + \bar{B}_{lat}(\Theta) \delta_a, \tag{3.18}$$

where the state is $x_{lat} = [v, p, r, \phi]^T$ and the matrices are:

$$M_{lat} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -\frac{I_{xz}}{I_{xx}} & 0 \\ 0 & -\frac{I_{xz}}{I_{zz}} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.19}$$

$$\bar{A}_{lat}(\Theta) = \begin{bmatrix} Y_v & Y_p + \bar{w} & Y_r - \bar{u} & g\cos\bar{\theta} \\ L_v & L_p & L_r & 0 \\ N_v & N_p & N_r & 0 \\ 0 & 1 & \tan\bar{\theta} & 0 \end{bmatrix}, \quad \text{and} \quad \bar{B}_{lat}(\Theta) = \begin{bmatrix} Y_{\delta_a} \\ L_{\delta_a} \\ N_{\delta_a} \\ 0 \end{bmatrix}. \tag{3.20}$$

The $\bar{A}_{lat}$ and $\bar{B}_{lat}$ matrices are populated with the dimensional stability and control derivatives, the trim variables $\bar{u}$, $\bar{w}$, and $\bar{\theta}$, and the acceleration due to gravity g. The output equation, which is not shown, simply picks p from the state x_{lat}.

The system identification exploits the structure that is evident in the black-box and the grey-box parametrizations of the longitudinal and the lateral-directional dynamics. In addition, identification considers the actuator dynamics G_a.

3.3.3 Initial Linear Models

This section presents the *initial* linear models obtained by linearizing the first principles nonlinear model (Section 2.3) at the nominal trim airspeed of 15.4 m s^{-1} (Section 2.4.1). These initial linear models use the VLM-derived stability and control derivatives.

Longitudinal dynamics

At the nominal trim point, the elevator-to-pitch rate black-box model is:

$$G_{lon}(s) = \frac{q(s)}{\delta_e(s)} = \frac{-186s(s + 0.354)(s + 3.79)}{(s^2 + 0.163s + 0.757)(s^2 + 11.2s + 210)}. \tag{3.21}$$

The phugoid mode has a damping ratio of 0.094 and a natural frequency of 0.87 rad s^{-1}. In addition, the short period mode has a damping ratio of 0.39 and a natural frequency of 14.5 rad s^{-1}. The populated matrices of the corresponding grey-box LTI model are:

$$A_{lon} = \begin{bmatrix} -0.151 & 0.753 & -1.02 & -9.78 \\ -0.883 & -5.69 & 13.9 & -0.668 \\ 0.878 & -12.9 & -5.49 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \text{and} \quad B_{lon} = \begin{bmatrix} 0.146 \\ -24.5 \\ -186 \\ 0 \end{bmatrix}. \tag{3.22}$$

Figure 3.6 shows the Bode diagram of $G_{lon}G_a$, i.e. the response from the elevator command δ_{ec} to the pitch rate q. The markers indicate the natural frequencies of the phugoid mode (+), the short period mode (∗), and the actuator (×). The highlighted portion indicates the frequency range covered by the input excitation (Section 3.3.4).

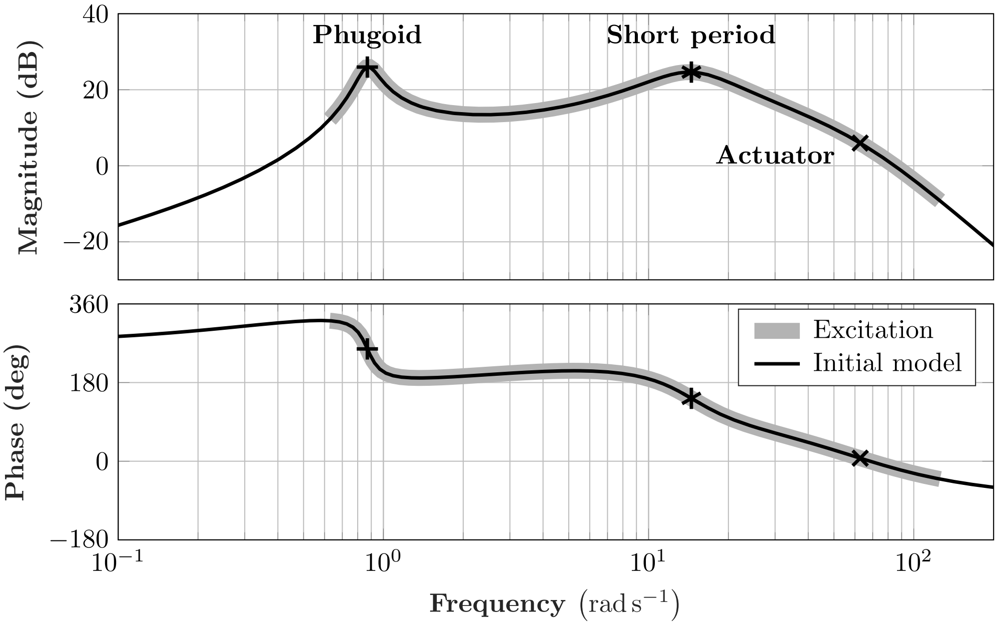


Figure 3.6: The Bode diagram of the initial model from the elevator command δ_{ec} to the pitch rate q. The frequency range of the excitation is highlighted.

Lateral-directional dynamics

At the nominal trim point, the aileron-to-roll rate black-box model is:

$$G_{lat}(s) = \frac{p(s)}{\delta_a(s)} = \frac{-283\,(s-0.0431)\left(s^2+1.23s+25.8\right)}{(s-0.0247)\,(s+11.8)\,(s^2+0.66s+34.6)}, \tag{3.23}$$

The dutch roll mode has a damping ratio of 0.056 and a natural frequency of 5.88 $\mathrm{rad\,s^{-1}}$. The roll subsidence mode has a time constant of 0.085 s. The spiral mode is marginally unstable and has a time constant of -40.5 s. The unstable spiral mode is also associated with a marginal right-half plane zero at 0.0431 $\mathrm{rad\,s^{-1}}$. The inertia values in M_{lat} are populated from Table 2.3 and the $\bar{A}_{lat}$ and $\bar{B}_{lat}$ matrices are populated as:

$$\bar{A}_{lat} = \begin{bmatrix} -0.42 & 1.12 & -15.3 & 9.78 \\ -5.27 & -11.2 & 2.14 & 0 \\ 1.6 & -0.953 & -0.755 & 0 \\ 0 & 1 & 0.0682 & 0 \end{bmatrix} \quad \text{and} \quad \bar{B}_{lat} = \begin{bmatrix} -0.488 \\ -281 \\ -3.44 \\ 0 \end{bmatrix}. \tag{3.24}$$

Figure 3.7 shows the Bode diagram of $G_{lat}G_a$, i.e. the response from the aileron command δ_{ac} to the roll rate p. The markers indicate the natural frequencies of the dutch roll mode ($+$), the roll subsidence mode ($*$), and the actuator ($\times$). The spiral mode lies outside the axis limits. The highlighted portion of the diagram indicates the frequency range covered by the input excitation, as explained next.

3.3.4 Design of Flight Experiments

The system identification process aims to experimentally estimate the parameters of the black-box and the grey-box LTI models presented in Section 3.3.2. In order to satisfy the assumption of local linearity, it is important that the flight condition of the aircraft stays close to the trim point throughout the experiment. The pilot achieves this by setting up the aircraft to fly at the nominal trim airspeed, at constant altitude, and along a constant course. Further, the system identification experiments are conducted open-loop. This simplifies the analysis since the dynamics of the controller are absent in the recorded responses. However, the pilot compensates for the disturbing effects of wind gusts and turbulence and, thus, functions as a low-bandwidth controller.

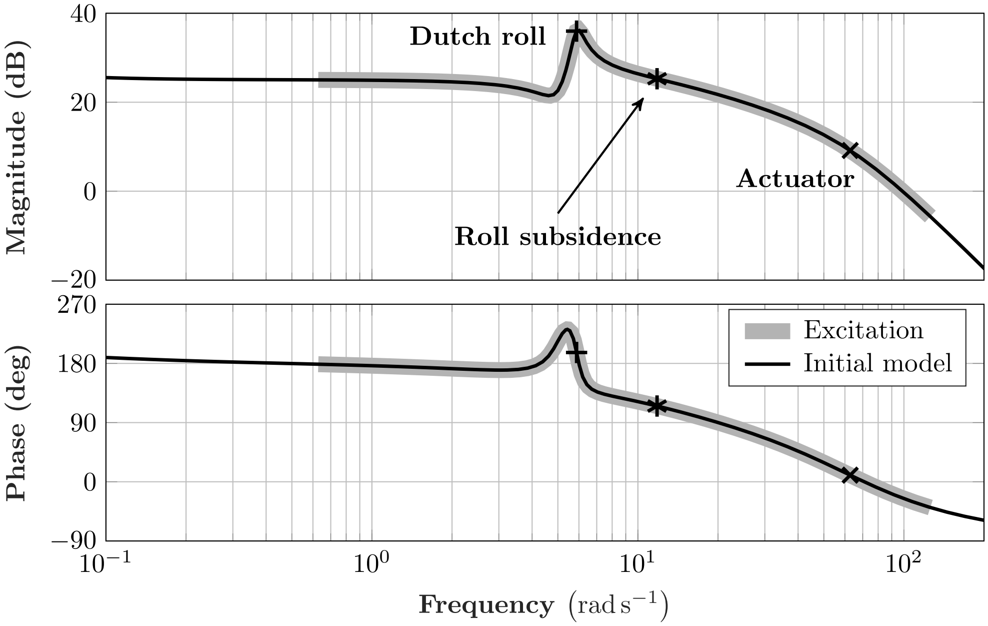


Figure 3.7: The Bode diagram of the initial model from the aileron command δ_{ac} to the roll rate p. The frequency range of the excitation is highlighted.

The system identification focuses on experimentally characterizing the elevator-to-pitch rate and the aileron-to-roll rate aircraft responses. The Bode diagrams shown in Figures 3.6 and 3.7 guide the design of the experiments. The elevator and the aileron commands are specified using the chirp function $A\cos\left(\omega_i\left(t\right)t\right)$, where A and $\omega_i\left(t\right)$ denote the amplitude and the instantaneous frequency, respectively. The instantaneous frequency is swept linearly over time as $\omega_i\left(t\right)=\omega_1+\frac{(\omega_2-\omega_1)}{2T_{ch}}t$, where ω_1, ω_2, and T_{ch} denote the start frequency, the end frequency, and the sweep duration, respectively.

Practical considerations

Chirps commands can be designed to cover a wide frequency range and have yielded excellent results in the past for the system identification of small UAS [52,53]. The spectral content of δ_{ec} and δ_{ac} are designed to span the longitudinal and lateral-directional modes, respectively. However, the frequency range of the chirp function is dictated by practical considerations. For instance, 20 s is the longest duration that the aircraft can fly along a constant course and remain within the airspace available at UMore Park.

This is limitation of the UMore Park test range and not one that applies to all UASs. Thus, 0.05 Hz is a lower bound on the chirp frequencies. In addition, the median sample rate of the flight computer is 90 Hz (Section 2.2.2). Thus, the Nyquist frequency of 45 Hz is an upper bound on the chirp frequencies for achieving alias-free sampling.

Table 3.1 summarizes the natural frequencies of the aircraft dynamic modes, the actuator bandwidth, and the bounds on the frequency of the chirp function. The actuator bandwidth is obtained from Section 3.2.4. The time delay is computed as 0.05 s using flight data in Section 3.3.8. Given these considerations, the final chirp commands are designed to span the frequency range $[0.1, 20]$ Hz. This range is highlighted in the Bode diagrams show in Figures 3.6 and 3.7, and includes the natural frequencies of all the longitudinal and the lateral-directional modes, as well as the actuator bandwidth.

Table 3.1: Key flight dynamic modes and limiting frequencies.

Mode/bound	**Frequency** (Hz)	**Source**
Chirp lower bound	0.05	Longest maneuver lasts 20 s
Phugoid mode	0.14	Initial LTI model
Dutch roll mode	0.94	Initial LTI model
Roll subsidence mode	1.9	Initial LTI model
Short period mode	2.3	Initial LTI model
Time delay	3.18	Computed as 0.05 s (Section 3.3.8)
Actuator bandwidth	9	Benchtop experiment (Section 3.2.4)
Nyquist frequency	45	One-half of the sample rate
Median sample rate	90	Goldy 2 firmware

Experiment design

Since the $[0.1, 20]$ Hz frequency range is too wide to be completed within 20 s, it is divided into three smaller segments, as shown in Table 3.2. In particular, the low, the medium, and the high segments use overlapping frequency ranges and correspond to 30, 20, and 10-fold increases between the start and end frequencies, respectively. Each segment is assigned a chirp ID, as shown in Table 3.2. Two flights are conducted for the system identification. The first flight commands elevator chirps over a duration of 20 s and uses the pitch rate response in order to identify the longitudinal dynamics. The second flight commands aileron chirps over a duration of 15 s and uses the roll rate response in order to identify the lateral-directional dynamics.

Table 3.2: The chirp parameters used in the flight experiments.

Range	**Chirp ID**	**Amplitude**	**Frequency**	**Increase**	**Trials**
Low	Ch1	2°	$[0.1, 3]$ Hz	30-fold	3
Medium	Ch2	2°	$[0.5, 10]$ Hz	20-fold	3
High	Ch3	2°	$[2, 20]$ Hz	10-fold	1

In both the flights, the low and medium frequency chirps are executed thrice and the high frequency chirp is executed once. Multiple trials are conducted to obtain a large sample size. In addition, the amplitude of the chirp command is set equal to 2° across all the experiments. Prior flight tests helped determine that this amplitude yields a sufficiently high signal-to-noise ratio (SNR) in the pitch and the roll rate signals. While larger chirp amplitudes would result in higher SNRs, they have the undesirable effect of causing the aircraft to deviate from its trim point. Even with the 2° chirp amplitude, the aircraft deviates from its trim point due to wind gusts and turbulence. The pilot corrects for these deviations using the RC transmitter. The flight computer superimposes the pilot stick inputs over the chirp commands. Thus, the following analyses consider the input to be the total elevator or aileron command, as seen by the aircraft.

3.3.5 Longitudinal Model Identification

Summary of experiments

Experiments for the longitudinal model identification are performed during FLT5 on 2017-04-21. Seven experiments are performed by commanding elevator chirps using the parameters shown in Table 3.2. Each chirp lasts for 20 s. The experiments are labeled using the nomenclature SE-Ch**X**-T**Y**, where **X** denotes the chirp ID number and **Y** denotes the trial number. For illustrative purposes, Figure 3.8 shows the elevator chirp command and the corresponding pitch rate response observed during the experiment SE-Ch2-T3. In this experiment, the chirp spans the medium frequency range $[0.5, 10]$ Hz and thus excites only the short period mode. This is visible in the plot of the pitch rate, whose magnitude rolls off midway through the chirp. In addition, the low frequency pilot stick inputs, and the elevator trim setting of $-1°$, are visible in the plot of the elevator command. The time history plots of the other elevator chirp experiments appear similar to Figure 3.8, and are not shown here for brevity.

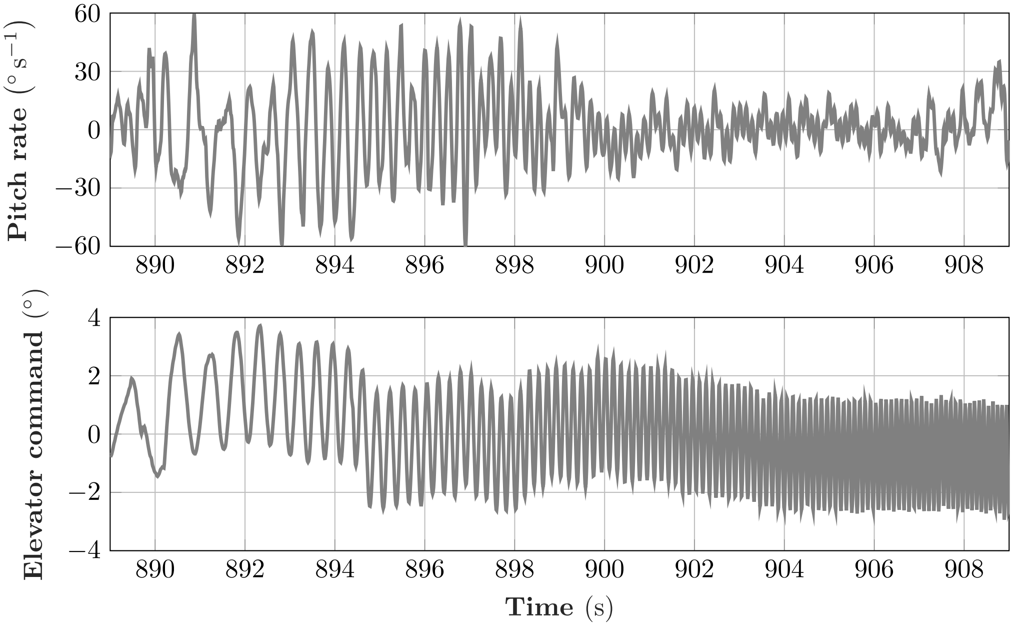


Figure 3.8: The elevator chirp command and the corresponding pitch rate response as observed during the experiment SE-Ch2-T3.

Although the target airspeed is 15.4 $\mathrm{m\,s^{-1}}$, the measured airspeeds during each of the seven experiments are closer to 19 $\mathrm{m\,s^{-1}}$. This is because the pilot is flying the aircraft open-loop and does not have feedback of the airspeed. The mean measured airspeed is computed over the duration of each experiment. This mean value varies between a low of 18.35 $\mathrm{m\,s^{-1}}$ in SE-Ch1-T3 and a high of 21 $\mathrm{m\,s^{-1}}$ in SE-Ch2-T2. Consequently, a trim airspeed of 19 $\mathrm{m\,s^{-1}}$ is considered to be representative of all the seven experiments, and is used to regenerate the initial LTI models for the system identification. At 19 $\mathrm{m\,s^{-1}}$, the phugoid mode has the parameters $\zeta_p = 0.19$ and $\omega_p = 0.7$ $\mathrm{rad\,s^{-1}}$ and the short period mode has the parameters $\zeta_s = 0.4$ and $\omega_s = 17$ $\mathrm{rad\,s^{-1}}$.

In this regard, note that the phugoid frequency at 19 $\mathrm{m\,s^{-1}}$ is 0.11 Hz. Since this is very close to the lowest chirp start frequency - a value of 0.1 Hz that is attained during the Ch1 chirps - the phugoid mode is not observed during any of the experiments. On the other hand, the short period frequency at 19 $\mathrm{m\,s^{-1}}$ is 2.7 Hz. This frequency is close to the edges of the Ch1 and Ch3 frequency ranges, but is well within the span of the Ch2 frequency range. Hence, the identification focuses on the short period mode.

For each of the seven chirp experiments, the coherency spectrum between q and δ_{ec} is computed using Equation (3.2). Figure 3.9 displays all the seven coherency spectra, labeled according to the nomenclature described earlier. These spectra guide the selection of the experiments for the system identification by highlighting the frequency ranges where the coherency is high. Based on the coherencies attained at the short period frequency of 2.7 Hz, the experiments SE-Ch1-T1, SE-Ch1-T2, SE-Ch2-T3, and SE-Ch3-T1 are selected for the system identification. Note that the selection includes at least one trial from the low, the medium, and the high frequency ranges. The selected experiments are detrended, by removing the sample mean from each experiment, and finally merged to create a single input-output data set. (See Chapter 14 in [44] for the details of detrending and merging experiments.)

Finally, the total time delay τ_f observed during the experiments is estimated to be equal to 0.05 s. The details of this estimation are relegated to Section 3.3.8. Hence, for the purposes of system identification, τ_f is considered to be a known parameter.

Frequency response function and black-box model

From the merged experiments, the frequency response function relating q and δ_{ec} is computed using Equation (3.1). This function not only characterizes the dynamics of the aircraft G_{lon} and the actuator G_a, but also captures the time delay τ_f. Section 3.2.4 models G_a and Section 3.3.8 estimates τ_f. Thus the system identification problem reduces to characterizing G_{lon}. The prediction error minimization method (Section 3.1.4), implemented using the `tfest` function, is used to estimate the parameters of the black-box model structure shown in Equation (3.14). Since the phugoid mode is not observed, only the short period mode is characterized from the merged experiments. While the parameters of the short period mode may be initialized using the VLM-derived model, they are initialized using a method that is built into the `tfest` function for simplicity. The prediction error minimization results in $\zeta_s = 0.47$ and $\omega_s = 22 \text{ rad}\,\text{s}^{-1}$. The corresponding NRMSE is 43%.

Figure 3.10 plots the Bode diagrams of the frequency response function, the black-box model, and the initial model from the elevator command δ_{ec} to the pitch rate q, at an

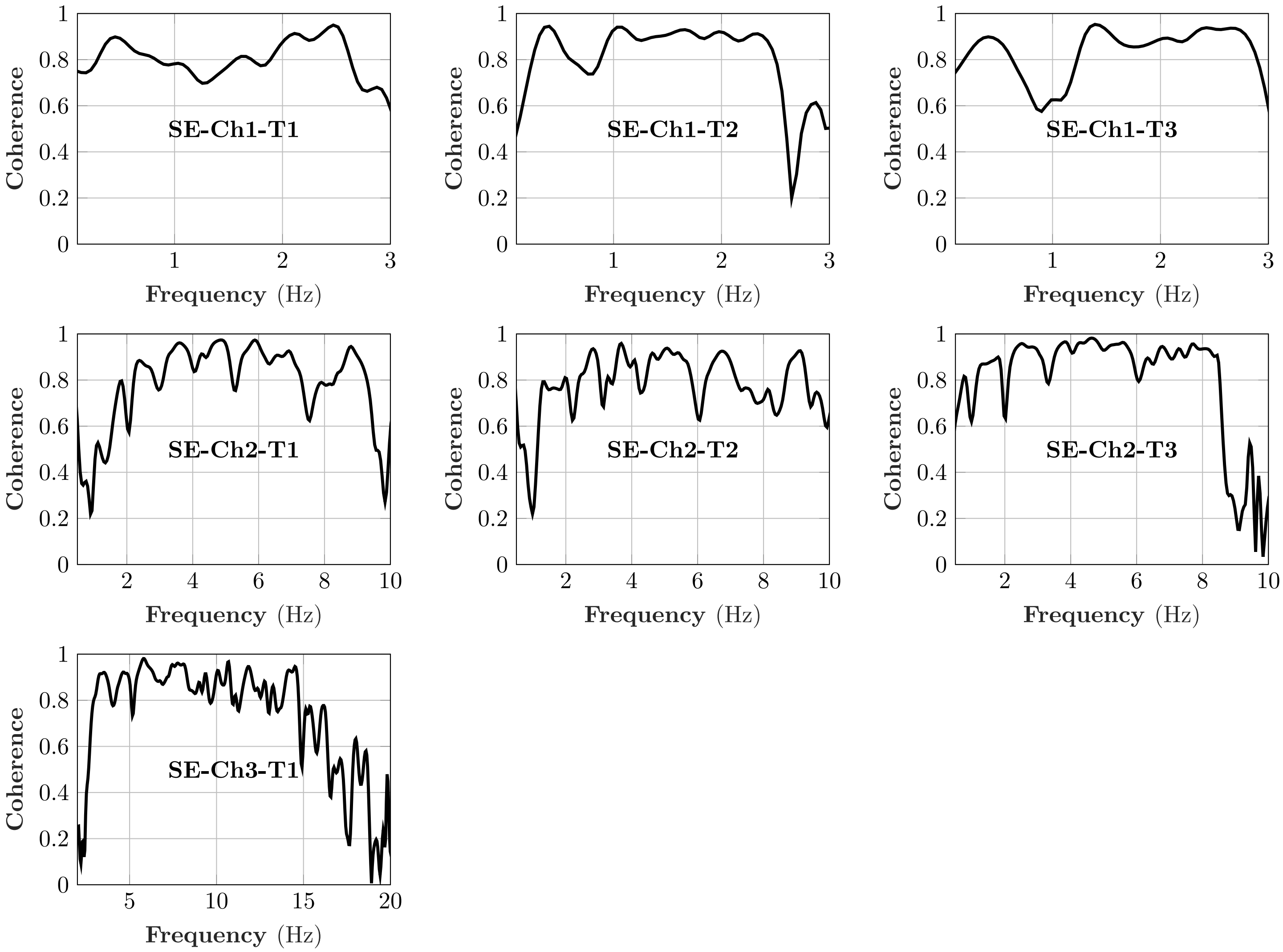


Figure 3.9: The coherency spectra obtained for the seven elevator chirp experiments.

airspeed of 19 $\mathrm{m\,s^{-1}}$. Note that all the Bode diagrams capture the short period mode and the actuator dynamics. Additionally, the initial model captures the unobserved phugoid mode, as evident in the magnitude peak near 0.7 $\mathrm{rad\,s^{-1}}$. The rapid phase loss that is seen in all of the plots is attributed to the time delay of 0.05 s. Although the NRMSE of 43% of relatively low, Figure 3.10 indicates that the initial model largely captures the frequency response function in the range that is relevant for feedback control, i.e. $[2, 50]$ $\mathrm{rad\,s^{-1}}$. Thus the grey-box parameter estimation is not pursued.

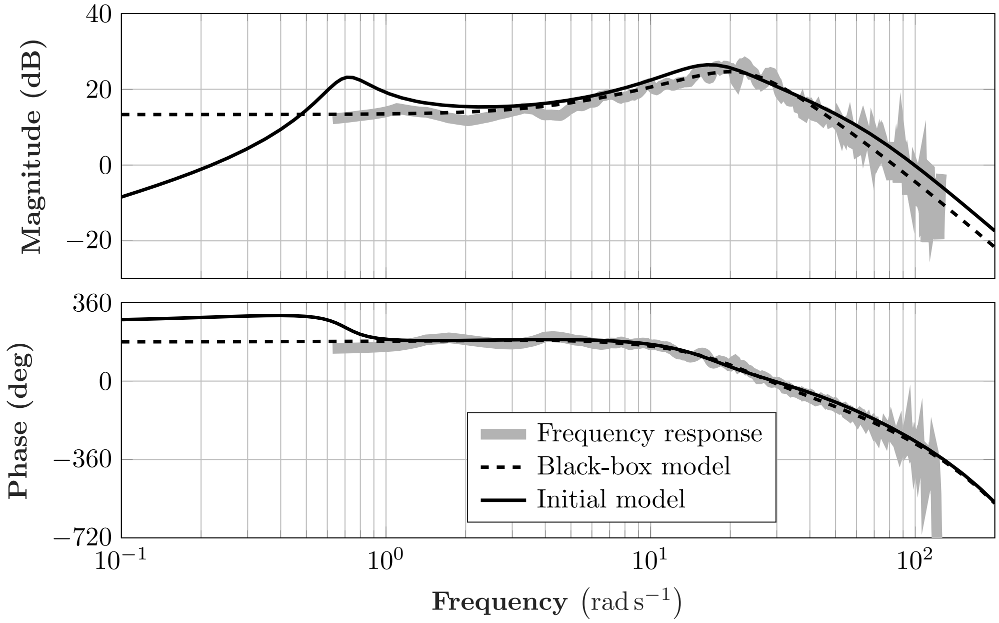


Figure 3.10: The frequency response function, the black-box model, and the initial model from the elevator command δ_{ec} to the pitch rate q, at an airspeed of 19 $\mathrm{m\,s^{-1}}$.

3.3.6 Lateral-Directional Model Identification

Summary of experiments

Experiments for the lateral-directional model identification are performed during FLT6 on 2017-04-21. Seven experiments are performed by commanding aileron chirps using the parameters shown in Table 3.2. Each chirp lasts for 15 s. The experiments are labeled using the nomenclature SA-Ch**X**-**TY**, where **X** denotes the chirp ID number and **Y** denotes the trial number. For illustrative purposes, Figure 3.11 shows the aileron chirp command and the corresponding roll rate response observed during the experiment SA-Ch2-T1. In this experiment, the chirp spans the medium frequency range $[0.5, 10]$ Hz and thus excites the dutch roll and the roll subsidence modes. This is visible in the plot of the roll rate, whose magnitude rolls off midway through the chirp. In addition, the low frequency pilot stick inputs, and the small aileron trim setting, are visible in the plot of the aileron command. The time history plots of the other aileron chirp experiments appear similar to Figure 3.11, and are not shown here for brevity.

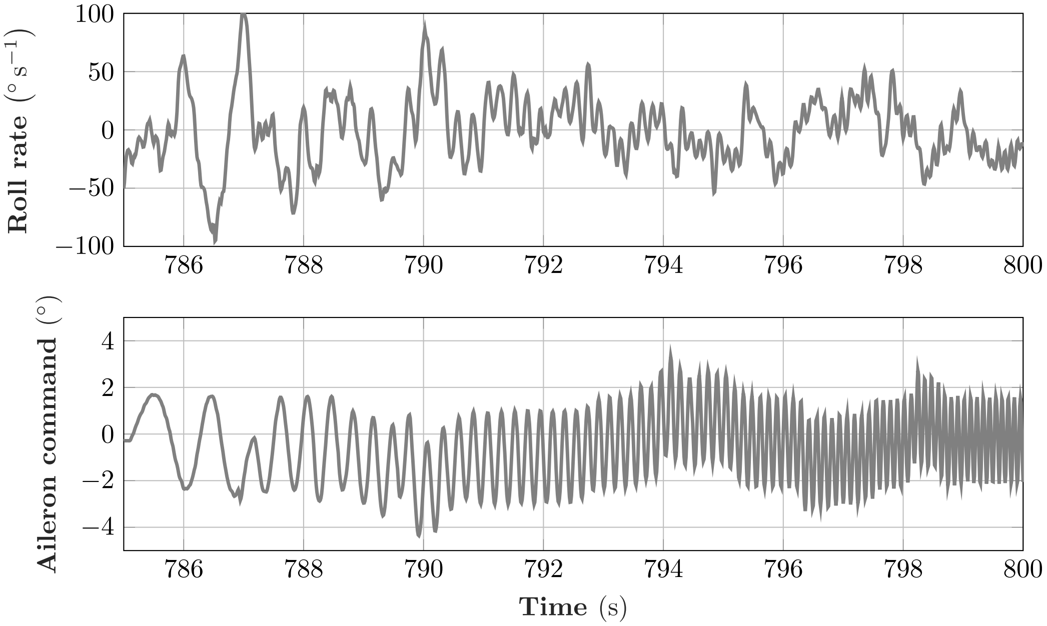


Figure 3.11: The aileron chirp command and the corresponding roll rate response as observed during the experiment SA-Ch2-T1.

Although the target airspeed is 15.4 $\mathrm{m\,s^{-1}}$, the measured airspeeds during each of the seven experiments are closer to 18 $\mathrm{m\,s^{-1}}$. This is because the pilot is flying the aircraft open-loop and does not have feedback of the airspeed. The mean measured airspeed is computed over the duration of each experiment. This mean value varies between a low of 16.7 $\mathrm{m\,s^{-1}}$ in SA-Ch1-T1 and a high of 18.1 $\mathrm{m\,s^{-1}}$ in SA-Ch2-T3. Consequently, a trim airspeed of 18 $\mathrm{m\,s^{-1}}$ is considered to be representative of all the seven experiments, and is used to regenerate the initial LTI models for the system identification. At 18 $\mathrm{m\,s^{-1}}$, the spiral mode has a time constant of 242 s and the roll subsidence mode has a time constant of 0.07 s. In addition, the dutch roll mode has a damping ratio of 0.06 and a natural frequency of 7 $\mathrm{rad\,s^{-1}}$.

The spiral mode lies outside the frequency range of all the chirp commands and is not observed during any of the experiments. On the other hand, the roll subsidence mode frequency of 2.3 Hz lies within the span of all the chirp commands and the dutch roll mode frequency of 1.1 Hz lies within the spans of the Ch1 and Ch2 chirp commands. Ideally, the identification would only focus on the dutch roll and the roll subsidence modes. However, unlike the longitudinal case, a clean separation between the lateral-directional modes is difficult to achieve. Reduced order models for the lateral-directional modal dynamics are generally insufficiently accurate to be of any real use [25]. Hence, the black-box model identification necessarily considers all the lateral-directional modes. However, as will be seen shortly, the grey-box model structure offers some latitude in targeting specific modes during the identification.

For each of the seven chirp experiments, the coherency spectrum between p and δ_{ac} is computed using Equation (3.2). Figure 3.12 displays all the seven coherency spectra, labeled according to the nomenclature described earlier. These spectra guide the selection of the experiments for the system identification by highlighting the frequency ranges where the coherency is high. Based on the coherencies attained at the natural frequencies of the dutch roll and the roll subsidence modes, the experiment SA-Ch2-T1 is selected for the system identification. Note that, unlike the longitudinal case, the selection only includes the first trial of the medium frequency range chirp. The experiment SA-Ch3-T1 is not included because the dutch roll mode lies outside its frequency range. The selected experiment is detrended, by removing the sample mean. Finally, as with the longitudinal case, the time delay τ_f is considered to be a known parameter.

Frequency response function and black-box model

The frequency response function relating p and δ_{ac} is computed using Equation (3.1). This function not only characterizes the dynamics of the aircraft G_{lat} and the actuator G_a, but also captures the time delay τ_f. Section 3.2.4 models G_a and Section 3.3.8 estimates τ_f. Thus the system identification problem reduces to characterizing G_{lat}. The prediction error minimization method (Section 3.1.4), implemented using the `tfest`

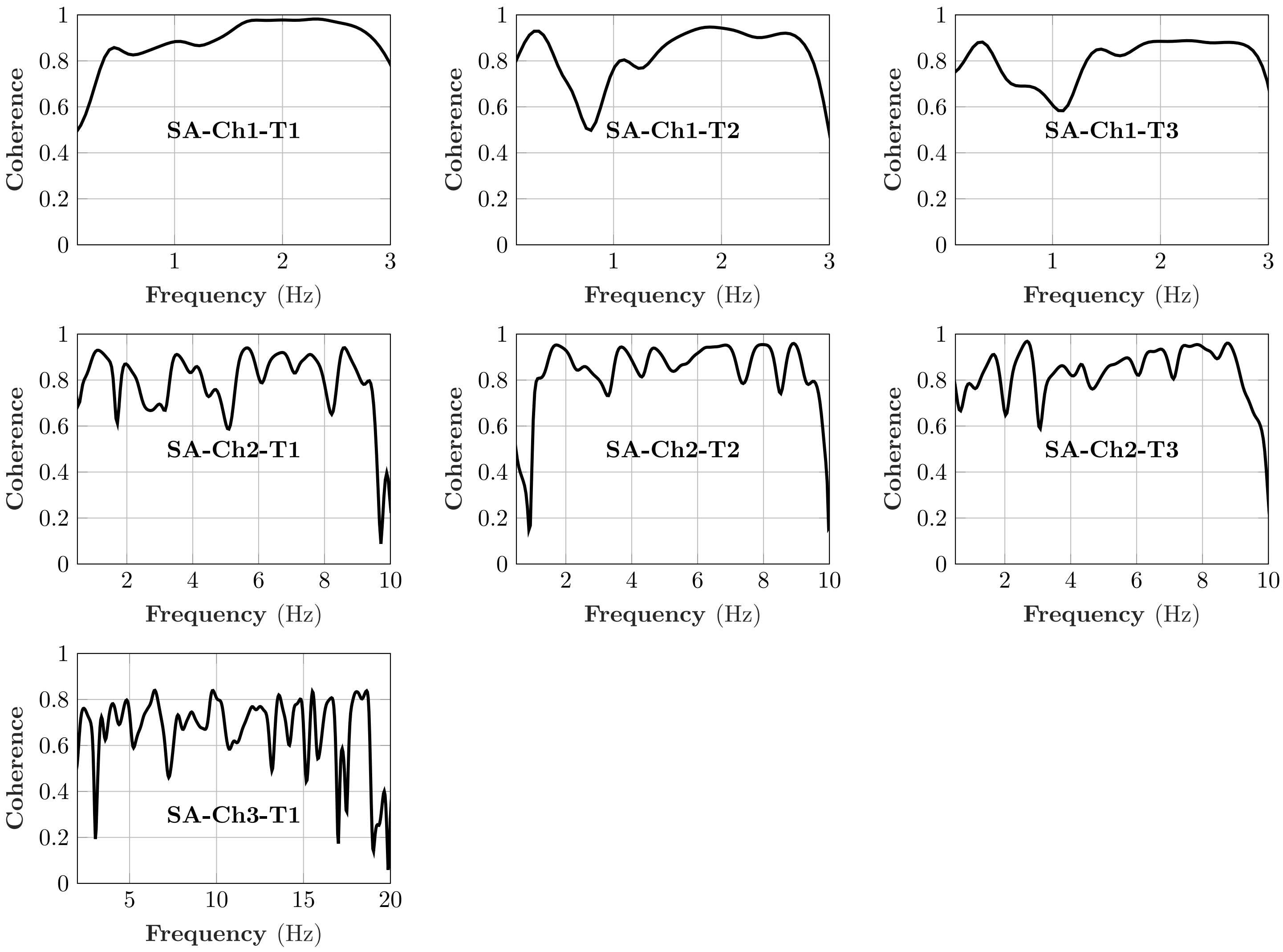


Figure 3.12: The coherency spectra obtained for the seven aileron chirp experiments.

function, is used to estimate the parameters of the black-box model structure shown in Equation (3.17). Since the lateral-directional modes cannot be separated, all the parameters shown in Equation (3.17) are estimated from the selected experiment. While these parameters may be initialized using the VLM-derived model, they are initialized using a method that is built into the `tfest` function for simplicity. The prediction error minimization results in a spiral mode time constant of 2.5 s, a roll subsidence mode time constant of 0.078 s, and a dutch roll mode damping ratio of 0.12 and natural frequency of 4.8 $\mathrm{rad\,s^{-1}}$. The corresponding NRMSE is 80%.

The modal parameters of the black-box model are different from those of the initial model. In particular, the dutch roll mode damping ratio increases from 0.06 to 0.12 and the natural frequency decreases from 7 $\mathrm{rad\,s^{-1}}$ to 4.8 $\mathrm{rad\,s^{-1}}$. The time constant of the roll subsidence mode slightly increases from 0.07 s to 0.078 s. The estimated spiral mode time constant of 2.5 s is two orders of magnitude lower than the initial value of 242 s. Since the spiral mode is not excited during any of the experiments, this estimate is inaccurate and does not correspond to any physical mode. It exists merely to ensure that G_{lat} has a total of four poles.

Figure 3.13 plots the Bode diagrams of the frequency response function, the black-box model, and the initial model from the aileron command δ_{ac} to the roll rate p, at an airspeed of 18 $\mathrm{m\,s^{-1}}$. Note that all the Bode diagrams capture the dutch roll mode and the actuator dynamics. The roll subsidence mode is also present, but is difficult to identify from the plot. The spiral mode is located outside the axis limits. The rapid phase loss that is seen in all of the plots is attributed to the time delay of 0.05 s. The NRMSE of 80% implies that the identified black-box model accurately describes the frequency response function, and hence the actual lateral-directional aircraft dynamics, in the frequency range $[0.5, 10]$ Hz. However, as seen in Figure 3.13, the initial model poorly describes the frequency response function at all frequencies and is insufficient for the purpose of control design. While the black-box model correctly describes the lateral-directional dynamics, it cannot be used to correct the initial model parameters. Hence, the next section presents the grey-box parameter estimation approach.

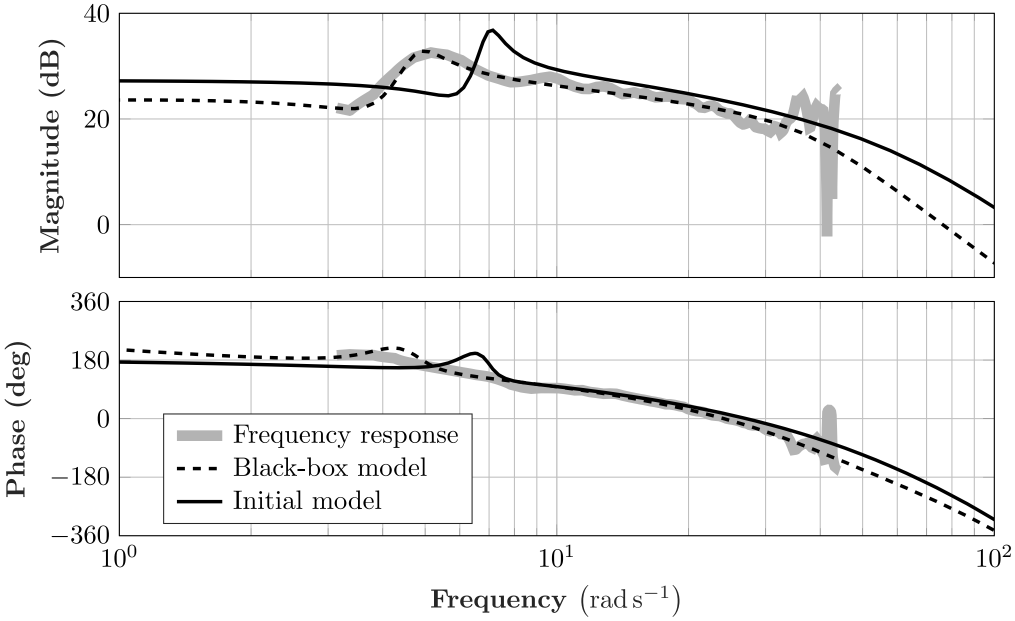


Figure 3.13: The frequency response function, the black-box model, and the initial model from the aileron command δ_{ac} to the roll rate p, at an airspeed of 18 $\mathrm{m\,s^{-1}}$.

Grey-box model

Equations (3.18), (3.19), and (3.20) describe the grey-box LTI model for the lateral-directional aircraft dynamics. Since this model is parametrized using the dimensional stability and control derivatives, it relates back to the aerodynamic model and, hence, the nonlinear aircraft model. Additionally, the grey-box model structure offers greater latitude in targeting specific modes, via the dimensional stability and control derivatives, as compared to the black-box model structure. In particular, each lateral-directional mode is predominantly affected by a subset of the dimensional stability derivatives [25], as shown in Table 3.3. The derivative Y_p, which captures the side force produced due to the roll rate, is small for the Vireo and thus does not appear in Table 3.3.

The goal of the grey-box parameter estimation is to update some of these derivatives, such that the updated model adequately describes the frequency response function shown in Figure 3.13. Table 3.3 thus acts as a guide to selecting the parameters to be updated. In this regard, reconsider the results of the black-box parameter estimation. The time constant of the roll subsidence mode increased only slightly from 0.07 s to

Table 3.3: Each lateral-directional mode is predominantly affected by a subset of the dimensional stability derivatives.

Mode	**Stability derivative**
Spiral	L_v, L_r, N_v, N_r
Dutch roll	Y_v, Y_r, N_v, N_r
Roll subsidence	L_p, N_p

0.078 s. Since this mode is already accurately described by the initial model, L_p and N_p are *not* updated. On the other hand, the dutch roll mode damping ratio doubled from 0.06 to 0.12 and the natural frequency decreased from 7 $\mathrm{rad\,s^{-1}}$ to 4.8 $\mathrm{rad\,s^{-1}}$. Since these are significant changes, Y_v, Y_r, N_v and N_r are updated in the grey-box model.

In addition, although the spiral mode is not excited during any of the experiments, Table 3.3 indicates that the spiral and the dutch roll modes share the N_v and N_r derivatives in common. Updating only some of the derivatives of the spiral mode, and not others, would lead to an over constrained model. Hence, L_v and L_r are also updated. Further, although the derivative Y_p does not predominantly affect any of the modes, it is updated in order to find the best value that explains the input-output data. Finally, the control derivative L_{δ_a}, which captures the aileron control effectiveness, is also updated since it directly affects the aileron-to-roll rate response.

As per the notation introduced in Section 3.1.3, the eight parameters to be updated are denoted by $\Theta = [Y_v, Y_p, Y_r, L_v, L_r, N_v, N_r, L_{\delta_a}]^T$ and are initialized using the initial model. In order to constrain the optimization, each positive parameter in Θ is lower bounded by one-half of its initial value and upper bounded by twice its initial value. This bounding strategy is reversed for negative parameters. These bounds effectively constrain the parameter vector to a hyper-rectangle $D_{\mathcal{M}} \subset \mathbb{R}^8$. The remaining parameters, the actuator dynamics G_a, and the time delay τ_f are already known, and hence are not updated during the grey-box estimation. The prediction error minimization method, described in Section 3.1.4, is used to estimate the optimal parameter vector $\hat{\Theta}$ from the experiment SA-Ch2-T1. Table 3.4 lists the initial and the final values, as well as the lower and the upper bounds, of the parameters of the grey-box model.

Table 3.4 shows that, among the parameters that are updated, only Y_p and Y_r reach their respective lower bounds at $\hat{\Theta}$. In addition, no parameter reaches its upper bound.

Table 3.4: The initial and updated parameters of the grey-box lateral-directional model.

Derivative	**Initial value**	**Lower bound**	**Upper bound**	**Final value**
L_{δ_a}	-467.9	-935.9	-234.0	-331.7
L_p	-14.79	not updated		-14.79
L_r	2.604	1.302	5.209	3.160
L_v	-8.960	-17.92	-4.480	-4.849
N_{δ_a}	12.99	not updated		12.99
N_p	-0.646	not updated		-0.646
N_r	-1.079	-2.157	-0.5393	-1.510
N_v	1.387	0.6935	2.774	0.7884
Y_{δ_a}	-0.8474	not updated		-0.8474
Y_p	0.0904	0.0452	0.1807	0.0452
Y_r	0.1892	0.0946	0.3784	0.0946
Y_v	-0.5729	-1.146	-0.2865	-0.7817

By considering the change in the absolute value of each parameter, it is evident that VLM overestimates L_{δ_a}, L_v, N_v, Y_p, and Y_r and underestimates L_r, N_r, and Y_v. The final values of these parameters correspond to a spiral mode time constant of 14 s, a roll subsidence mode time constant of 0.065 s, and a dutch roll mode damping ratio of 0.16 and natural frequency of 4.8 $\mathrm{rad\,s^{-1}}$. The grey-box parameter estimation achieves a NRMSE of 70%, which is 10 percentage points lower than that achieved with the black-box model. This drop in accuracy is expected because the grey-box model structure imposes a greater number of constraints, via the parameters that are not updated, as compared to the black-box model structure.

The modal parameters of the grey-box model are similar to those of the black-box model. Although the derivatives associated with the roll subsidence mode are not updated, its time constant still changes slightly. This shows that, in general, each mode is affected by all the derivatives, and that Table 3.3 only provides an approximate separation. Further, the dutch roll mode parameters are very similar between the grey-box and the black-box models. The grey-box spiral mode time constant of 14 s is an order of magnitude lower than the initial value of 242 s. This estimate still cannot be trusted since the spiral mode is not excited during any of the experiments. However, a large uncertainty in the spiral mode time constant is usually not an issue, since the mode is stabilized even by low bandwidth controllers.

Figure 3.14 plots the Bode diagrams of the frequency response function, the grey-box model, and the initial model from the aileron command δ_{ac} to the roll rate p, at an airspeed of 18 $\mathrm{m\,s^{-1}}$. The most important observation is that the grey-box model accurately describes the frequency response function, and hence the actual lateral-directional aircraft dynamics, in the frequency range $[0.5, 10]$ Hz. In addition, the dutch roll mode and the actuator dynamics are clearly visible in the Bode diagrams. As before, the rapid phase loss is attributed to the time delay of 0.05 s.

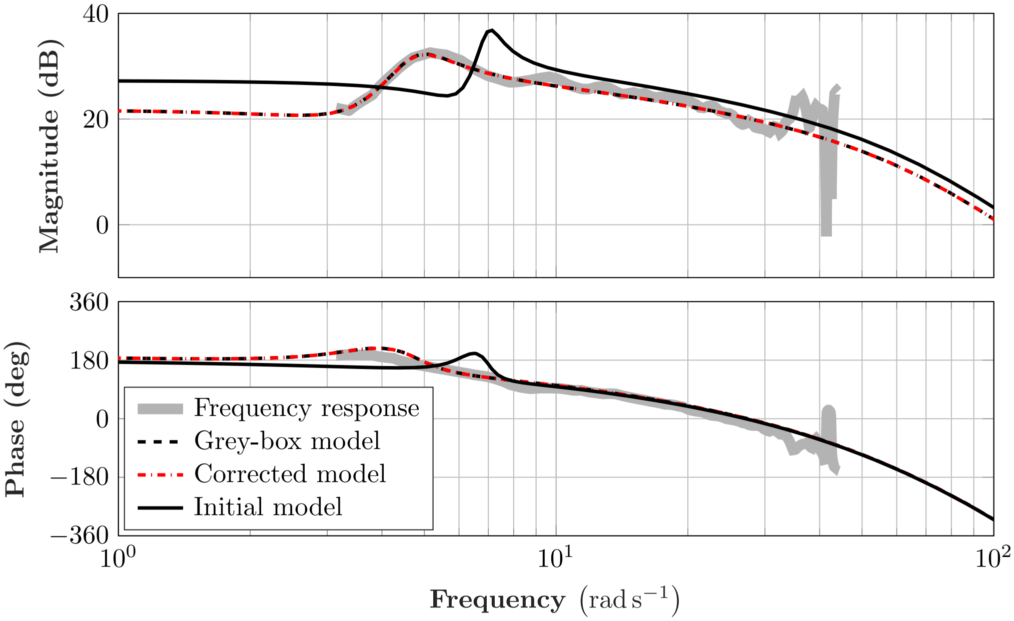


Figure 3.14: The frequency response function, the grey-box model, and the initial model from the aileron command δ_{ac} to the roll rate p, at an airspeed of 18 $\mathrm{m\,s^{-1}}$.

Figure 3.14 indicates that the grey-box model is sufficient for the purpose of control design. Hence, the final parameter values, shown in Table 3.4, are used to update the *non-dimensional* stability and control derivatives (see Equations (2.11) through (2.16)). Table 5.1 in [26] provides expressions that relate the two sets of stability and control derivatives. In addition to these expressions, note that transforming from one set to the other involves a translation and a rotation. Updating the non-dimensional stability and control derivatives is akin to updating the aerodynamic model of the aircraft. Hence, this process implicitly assumes that all of the mismatch between the initial model and the frequency response function arises due to an inaccurate aerodynamic model.

After all the updates are performed, the nonlinear aircraft model is again linearized at an airspeed of 18 $\mathrm{m\,s^{-1}}$. The Bode diagram of this so-called *corrected model* is also plotted in Figure 3.14. As seen in the figure, there is no distinguishable difference between the corrected model and the grey-box model. This implies that the identified parameters are properly integrated into the nonlinear aircraft model.

3.3.7 Model Validation

One of the final steps in the modeling process is validating the identified models. This is done using the experiments that are not selected during the model identification.

Longitudinal model validation

Recall, from Figure 3.10, that the initial model largely captures the frequency response function in the range $[2, 50]$ $\mathrm{rad\,s^{-1}}$. Since the phugoid mode is unobserved in the frequency response function, the order of the initial longitudinal model is reduced by truncating the states u and θ. This results in a model that only includes the short period mode (via the states q and w), the actuator dynamics, and the time delay. This reduced order model is termed as the *final model* for the purpose of model validation.

Time-domain simulations of this final model, at an airspeed of 19 $\mathrm{m\,s^{-1}}$, are compared against the pitch rate responses recorded during the experiments SE-Ch1-T3, SE-Ch2-T1, and SE-Ch3-T1. This selection includes one trial each from the low, the medium, and the high frequency ranges. None of these experiments, except for SE-Ch3-T1, are selected during the identification. Figure 3.15 shows the results of the validation over three subplots. The elevator chirp commands span the low (top subplot), the medium (middle subplot), and the high (bottom subplot) frequency ranges. In each subplot, the legend indicates the corresponding NRMSE achieved by the final model. The plots indicate that the NRMSEs progressively decrease from the low to the high frequency ranges. In addition, the NRMSE achieved with the experiment SE-Ch1-T3 is approximately 5 percentage points lower than the NRMSE achieved during the black-box model identification (see the text above Figure 3.10).

Although the NRMSEs achieved with the validation data set are only between 30% and 40%, it is more useful to evaluate the performance of the model over the frequency

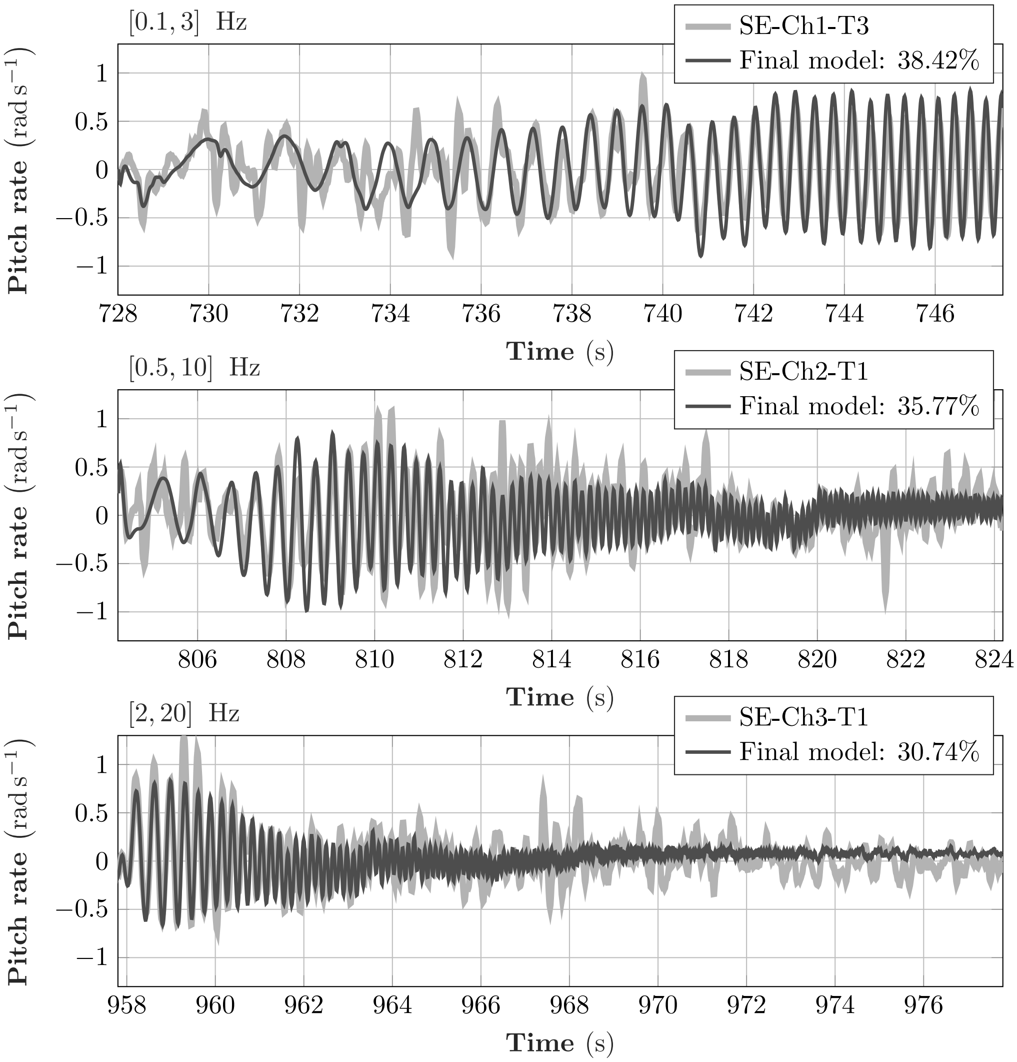


Figure 3.15: For the model validation, the pitch rate responses recorded during the experiments are compared to the simulations of the final model at an airspeed of 19 $\mathrm{m\,s^{-1}}$. The elevator chirp commands span the low (top subplot), the medium (middle subplot), and the high (bottom subplot) frequency ranges. In each subplot, the legend indicates the corresponding NRMSE achieved by the final model.

range of interest. In this regard, note that the simulations agree well with the experiments when the elevator chirp frequency is around the natural frequency of the short period mode, i.e. 2.7 Hz. This occurs in the top subplot over the interval $[740, 748]\,\mathrm{s}$, in the middle subplot over the interval $[807, 810]$ s, and in the bottom subplot over the interval $[958, 960]$ s. On the other hand, the performance of the model is poor at very low frequencies (e.g. in the top subplot over the interval $[728, 736]$ s) because the phugoid mode and other low frequency dynamics are not characterized from the experiments. Similarly, the performance of the model is poor at very high frequencies (e.g. in the bottom subplot over the interval $[966, 978]$ s) because the model does not account for the exogenous disturbances that affect the experiment, such as atmospheric turbulence and sensor noise. Since the performance of the model is adequate around the natural frequency of the short period mode, it is used in the control design (Chapters 4 and 7).

Lateral-directional model validation

The most important observation from Figure 3.14 is that the grey-box model accurately describes the frequency response function in the frequency range $[0.5, 10]$ Hz. Hence, the grey-box model is termed as the *final model* for the purpose of model validation.

Time-domain simulations of this final model, at an airspeed of 18 $\mathrm{m\,s^{-1}}$, are compared against the roll rate responses recorded during the experiments SA-Ch1-T1, SA-Ch2-T2, SA-Ch3-T1. This selection includes one trial each from the low, the medium, and the high frequency ranges. None of these experiments are selected during the identification. Figure 3.16 shows the results of the validation over three subplots. The aileron chirp commands span the low (top subplot), the medium (middle subplot), and the high (bottom subplot) frequency ranges. In each subplot, the legend indicates the corresponding NRMSE achieved by the final model. The plots indicate that the NRMSEs progressively decrease from the low to the high frequency ranges. In addition, the NRMSE achieved with the experiment SA-Ch1-T1 is approximately 8 percentage points lower than the NRMSE achieved during the grey-box model identification. Compared to the longitudinal model, the NRMSEs achieved with the lateral-directional model are higher for the low and the medium frequency range chirp commands.

As before, it is more useful to evaluate the performance of the model over the frequency range of interest. In this regard, note that the simulations agree well with the experiments when the aileron chirp frequency is around the natural frequency of the dutch roll mode, i.e. 0.76 Hz. This occurs in the top subplot over the interval $[625, 632]$ s and in the middle subplot over the interval $[812, 818]$ s. The dutch roll mode is not

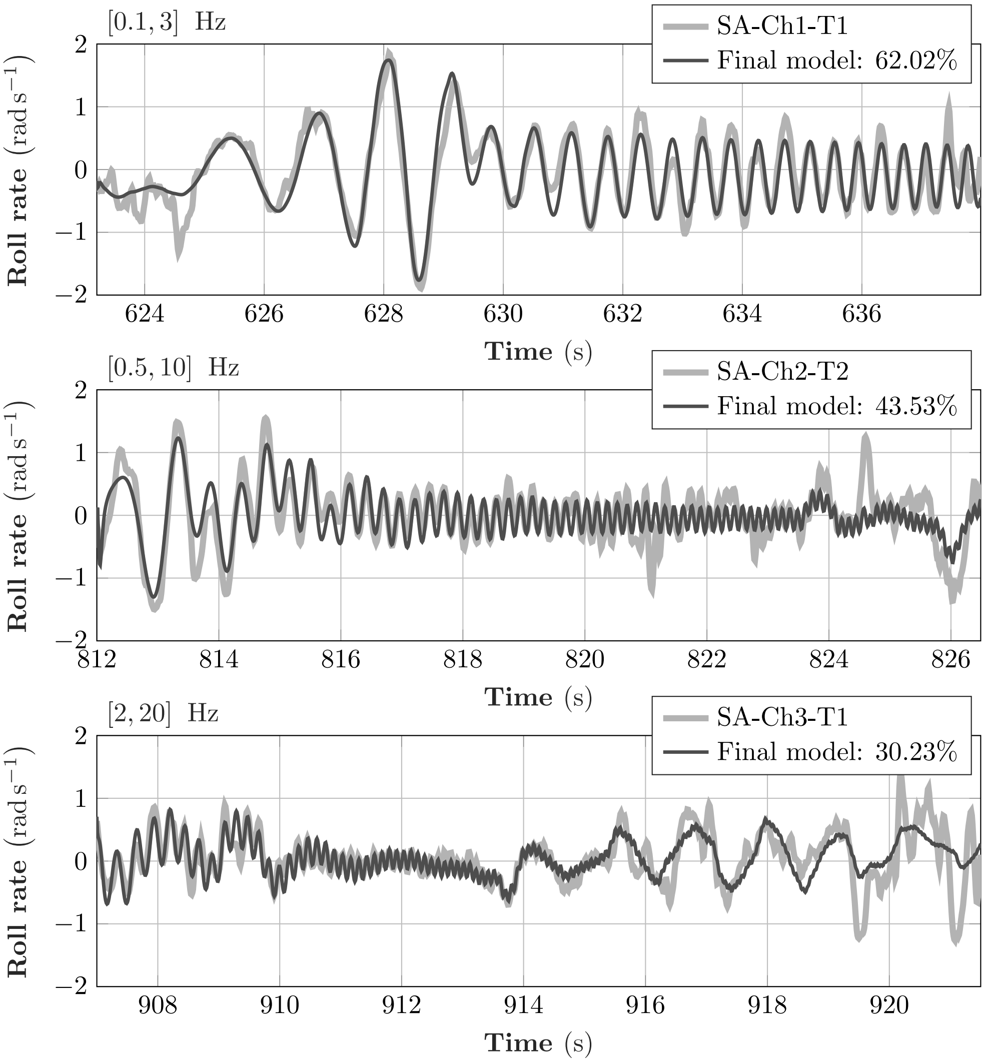


Figure 3.16: For the model validation, the roll rate responses recorded during the experiments are compared to the simulations of the final model at an airspeed of 18 m s^{-1}. The aileron chirp commands span the low (top subplot), the medium (middle subplot), and the high (bottom subplot) frequency ranges. In each subplot, the legend indicates the corresponding NRMSE achieved by the final model.

excited in the bottom subplot. The simulations also agree well with the experiments at low frequencies, especially when compared with the longitudinal model. On the other hand, the performance of the model is poor at very high frequencies (e.g. in the bottom subplot over the interval $[918, 922]\,\mathrm{s}$) because the model does not account for the exogenous disturbances that affect the experiment, such as atmospheric turbulence and sensor noise. Since the performance of the model is adequate around the natural frequency of the dutch roll mode, it is used in the control design (Chapters 4 and 7).

3.3.8 Time-Delay Estimation

This section provides the details on the estimation of the total time delay τ_f. The time delay is estimated by fitting black-box models with different, but known, fixed delays to the experimental data and observing the resulting NRMSE. A simple one-dimensional grid search then yields the time delay, via the corresponding model, that results in the largest NRMSE. However, the NRMSE is an effective metric only if it strongly depends on the time delay. Practically, this boils down to ensuring that the estimation includes the experiment(s) with the highest frequency content. This is because, for any given signal, the phase loss produced by the time delay is directly proportional to the frequency of the signal. For the experiments that are described in Section 3.3.4, this is simply achieved by including the high frequency chirp commands (see Table 3.2). In addition, the effect of the time delay is more pronounced on the highest frequency mode of the system. Since the short period mode ($2.3-2.7$ Hz) is the highest frequency mode of the Vireo, the time delay is estimated using the elevator chirp experiments.

The median sample time during the elevator chirp experiments is $T_s = 0.011$s. Nine different time delays $\tau_f = \{kT_s\}_{k=1}^{9}$ are selected to construct the one-dimensional grid. For each value of τ_f in this grid, a black-box model is identified from δ_{ec} to q in a manner similar to that described in Section 3.3.5. The identification is conducted for the experiments SE-Ch1-T1, SE-Ch1-T2, SE-Ch2-T3, and SE-Ch3-T1. Figure 3.17 shows the resulting NRMSEs obtained with each black-box model as a function of the time delay. It is evident from the figure that the NRMSEs corresponding to $\tau_f = 4T_s$ and $\tau_f = 5T_s$ are significantly larger than those obtained using the other time delays. However, since τ_f is the time delay associated with a continuous-time linear model, it is not restricted to be an integer multiple of T_s. If the black-box identification is repeated for $\tau_f = 4.5T_s$, the resulting NRMSEs are very close to those obtained using $\tau_f = 4T_s$. Thus, in order to be conservative, the final time delay is selected as $\tau_f = 0.05$ s.

On a related note, if the procedure described above is repeated using the aileron

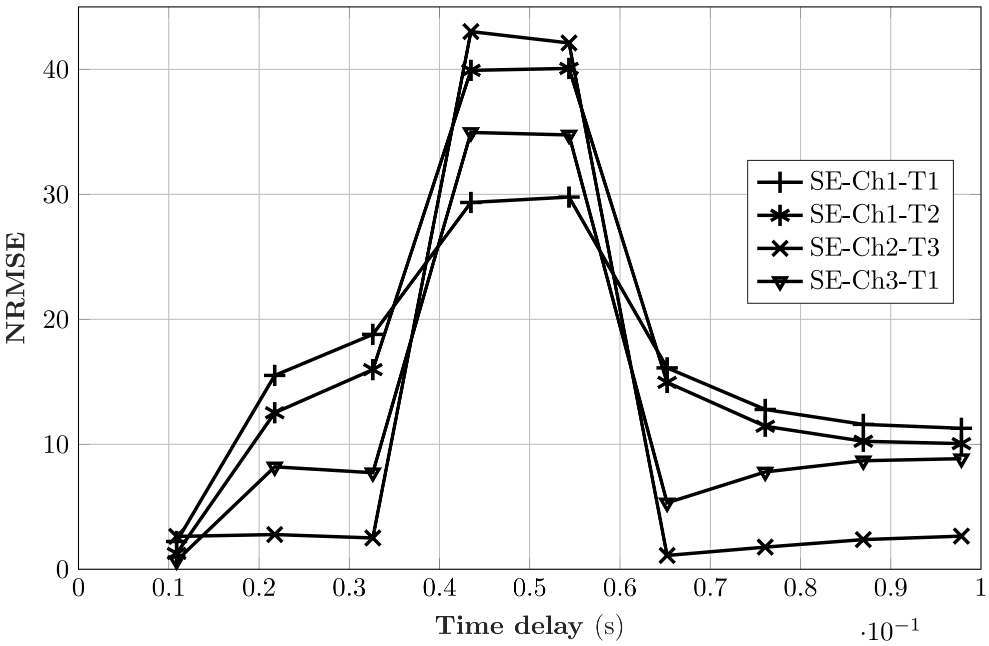


Figure 3.17: The time delay is estimated by fitting black-box models with different predetermined delays to the experimental data and observing the resulting NRMSE.

chirp experiments, the resulting NRMSEs are largely invariant to the time delay. In particular, for each of the SA-Ch**X**-T**Y** experiments, the NRMSE varies by less than 4 percentage points across the grid $\tau_f = \{kT_s\}_{k=1}^{6}$ and decreases for larger values of τ_f. This is because the dutch roll mode, with a natural frequency of 0.76 Hz, is relatively unaffected by the different amounts of the time delay. Finally, since τ_f is estimated directly from the experimental data, it encompasses delays in the actuators, the flight computer, and the sensors. For simplicity, all of this time delay is grouped at the input to the actuator, thus resulting in the new system $G_a^{\tau_f} = G_a e^{-\tau_f s}$. The nonlinear simulation environment is accordingly updated.

3.3.9 Low-Order Equivalent System

The model-based control design (Chapters 4 and 7) needs to account for the presence of the time delay. This is achieved by modeling the delayed actuator dynamics $G_a^{\tau_f}$ using a low-order equivalent system (LOES). Recall that $G_a^{\tau_f}$ is a second-order system with unit DC gain and a bandwidth of 57 rad s^{-1}. To obtain the LOES, $e^{-\tau_f s}$ is first replaced with a fifth-order Pade approximation, resulting in the seventh-order model G_a^P. Then, a balanced residualization [55] of G_a^P results in the following fifth-order LOES:

$$G_a^L \overset{s}{=} \left[\begin{array}{ccccc|c} -2.409 & 26.09 & 7.284 & -7.204 & 12.42 & -2.097 \\ -26.09 & -7.319 & -49.17 & 8.948 & -32.95 & -3.114 \\ 7.284 & 49.17 & -26.57 & 60.33 & -56.4 & 4.485 \\ 7.204 & 8.948 & -60.33 & -15.56 & 150.4 & 2.465 \\ 12.42 & 32.95 & -56.4 & -150.4 & -184.1 & 5.982 \\ \hline -2.097 & 3.114 & 4.485 & -2.465 & 5.982 & -0.06135 \end{array}\right]. \tag{3.25}$$

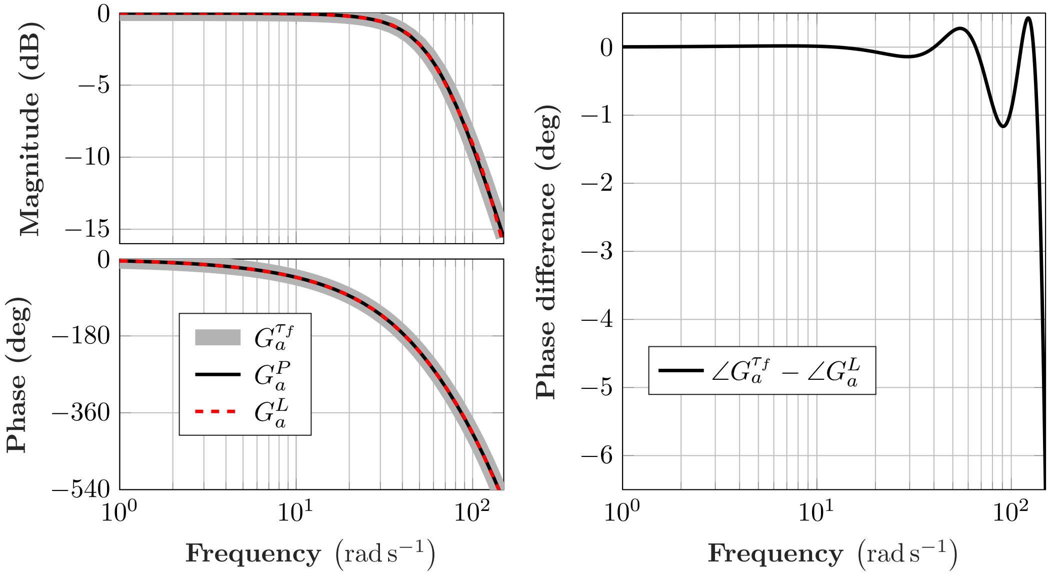


Figure 3.18: Left: The Bode diagrams of the delayed actuator dynamics $G_a^{\tau_f}$, its Pade approximation G_a^P, and the low-order equivalent system G_a^L. Right: The phase difference between $G_a^{\tau_f}$ and G_a^L is insignificant below the bandwidth of the actuator.

The peak error $\left\|G_a^P - G_a^L\right\|$ incurred in the residualization is approximately 7% of $\left\|G_a^P\right\| = 1$, where $\|\cdot\|$ denotes the H_∞ norm. This peak occurs at 393 rad s^{-1}, which is more than six times the bandwidth of G_a^P. The left-side of Figure 3.18 shows the Bode diagrams of the delayed actuator dynamics $G_a^{\tau_f}$, its Pade approximation G_a^P, and the low-order equivalent system G_a^L. As seen from the figure, there is no distinguishable difference between the three lines for frequencies up to and beyond the actuator bandwidth. For example, at 123 rad s^{-1}, the phase difference between $G_a^{\tau_f}$ and G_a^P is 1°. The right-side of Figure 3.18 shows the phase difference between the delayed actuator model $G_a^{\tau_f}$ and the LOES G_a^L. The phase difference is within 1° up to a frequency of approximately 85 rad s^{-1}. Since large phase differences only occur at frequencies that are significantly greater than the actuator bandwidth, the LOES is used in lieu of $G_a^{\tau_f}$ while designing the nominal (Chapter 4) and the fault-tolerant (Chapter 7) controllers.

It is acknowledged that the order of the LOES G_a^L may be reduced even further. In particular, the control design requires that the phase loss be matched primarily near the loop crossover frequency. Since the loop crossover frequencies for the control designs (Chapters 4 and 7) are far below the actuator bandwidth, the high accuracy of G_a^L may be unneeded. However, further simplifications of the LOES are not considered.

Finally, recall that Section 2.3.4 models the dynamics of the throttle using the first-order transfer function $G_t^{\tau_f}(s) = \frac{2\pi}{s+2\pi} e^{-\tau_f s}$, where $\tau_f = 0.05$ s, as per Section 3.3.8. In order to obtain a delay-free model of the throttle dynamics, $e^{-\tau_f s}$ is replaced with a second-order Pade approximation, resulting in the third-order model:

$$G_t^P \overset{s}{=} \left[\begin{array}{ccc|c} -6.28 & -30 & 0 & 2 \\ 0 & -120 & -75 & 16 \\ 0 & 64 & 0 & 0 \\ \hline 3.14 & 0 & 0 & 0 \end{array}\right]. \tag{3.26}$$

Since the bandwidth of $G_t^{\tau_f}$ is approximately one-ninth of the bandwidth of $G_a^{\tau_f}$, it is sufficient to use a second-order (and not higher) Pade approximation. Finally, the model G_t^P is used in lieu of $G_t^{\tau_f}$ during the control design (Chapters 4 and 7).

Chapter 4

Nominal Controller

4.1 Introduction

The previous chapters develop a model of the Vireo (Chapter 2) and update it using flight test data (Chapter 3). This chapter develops a nominal controller, which is intended to be active when there are no diagnosed faults in either of the elevons. The reader who is only interested in the fault detection and isolation algorithm and the fault-tolerant controller may skip ahead to Chapters 6 and 7, respectively. Before skipping ahead, however, it may help to read Section 4.2 to understand the autopilot architecture. The remainder of this chapter develops the nominal controller (Sections 4.3, 4.4, and 4.5), assesses its robustness (Section 4.6), and validates it using flight data (Section 4.7).

4.2 Autopilot Architecture

Figure 4.1 shows the autopilot architecture of the Vireo. The aircraft, the actuators, and the sensors are enclosed in a single block for compactness. The flight software implements the mission manager, the guidance law, the control law, and the navigation algorithm via the `AuraUAS` codebase [11]. The dashed line indicates the wireless link between the aircraft and the ground station. The UAS operator sends mission-level commands to the aircraft via this link. The mission manager interprets these commands and forms the desired trajectory, which the guidance law uses to compute the airspeed command V_{cmd}, the altitude command h_{cmd}, and the bank angle command ϕ_{cmd}. The

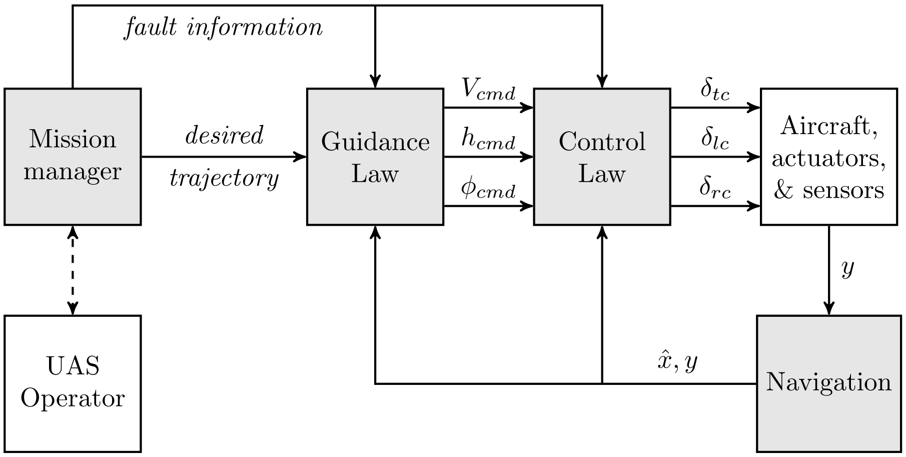


Figure 4.1: The autopilot architecture of the Vireo. The software components are shown with a grey background. The dashed line indicates the wireless link between the aircraft and the ground station.

UAS operator may also directly set V_{cmd} and h_{cmd}. The control law uses these three quantities to calculate the throttle command δ_{tc}, the left elevon deflection command δ_{lc}, and the right elevon deflection command δ_{rc}. An extended Kalman filter (EKF) [12, 56] uses the sensor measurements y to estimate the attitude, position, and velocity of the aircraft, collectively known as the navigation states $\hat{x}$. The guidance and control laws have access to both $\hat{x}$ and y, and are reconfigured based on the fault information.

4.2.1 Mission Manager

The mission manager that is implemented in the `AuraUAS` codebase (see Section 2.2.2) performs trajectory generation, task scheduling, event logging, etc. The mission manager forms the desired trajectory using either a list of waypoints or the parameters of a loiter circle, both of which are specified by the UAS operator. For example, Figure 4.2 depicts the aircraft tracking the waypoints WP1, WP2, and WP3, in that order. The waypoint tracking mode is underpinned by two key facts: (1) The desired trajectory is obtained by connecting consecutive waypoints using straight line segments and (2) A target waypoint is considered to be reached when the aircraft is within 50 m of it along the desired trajectory. With regard to Figure 4.2, when the aircraft is within 50 m of WP2, measured along the line segment connecting WP1 and WP2, the target changes to the next waypoint, i.e. WP3. The corresponding desired trajectory then changes to the line segment connecting WP2 and WP3. However, since the turn rate of the aircraft has a maximum limit, the actual trajectory follows a curved path, such as the one shown in the figure. Further, the turn radius depends on the prevailing winds, e.g. headwinds provide for sharper turns in comparison to tailwinds. Finally, in addition to the waypoint and the loiter circle trajectory modes, the mission manager includes a so-called autoland mode to perform the autonomous landing after the fault is detected. Section 7.9.1 provides more details about the autoland mode.

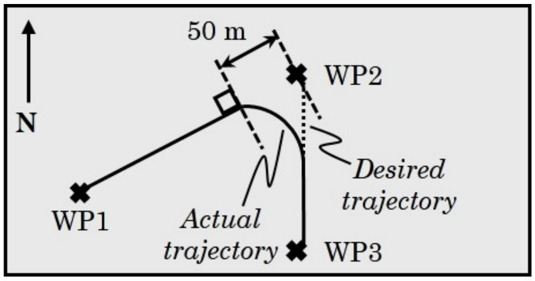


Figure 4.2: A schematic of the waypoint tracking mode of the mission manager.

4.2.2 Guidance Law

The guidance law computes V_{cmd} and h_{cmd} separately from ϕ_{cmd}. In particular, the UAS operator directly specifies V_{cmd} and h_{cmd} during the waypoint and the circle tracking modes. During the autoland mode, V_{cmd} is set to the approach speed and h_{cmd} varies along the glideslope. On the other hand, ϕ_{cmd} is computed using the nonlinear guidance law developed by Park et al. [57, 58]. In particular, the law uses the ground speed and the ground track of the aircraft to calculate the lateral acceleration command $a_{y,cmd}$. The reader is referred to Figure 1 and Equation (1) of [57] for the exact expression of $a_{y,cmd}$. The bank angle command is then computed as $\phi_{cmd} = \tan^{-1}(a_{y,cmd}/g)$.

The nonlinear guidance law is linearized for the ease of design and analysis. In particular, the closed-loop dynamics of the cross-track error is modeled as a damped second-order system. The particular guidance law of the Vireo is tuned using flight tests, such that the cross-track error response achieves a damping ratio of 1.7 and a natural frequency of 0.45 $\text{rad}\,\text{s}^{-1}$. In addition, the guidance law saturates ϕ_{cmd} in the interval

$[\phi_{lo}, \phi_{hi}]$ to prevent excess control effort. The reconfiguration of the guidance law only modifies the saturation bounds and does not modify the algorithm that computes ϕ_{cmd}.

When following a straight line path, the guidance law approximates a proportional-derivative controller on the cross-track error. In addition, the guidance law projects an imaginary point, located at some distance in front ot the aircraft, onto the upcoming flight path. This provides it with an element of anticipation for the tight tracking of curved paths. This guidance law is popular and is also used in the ArduPilot project[1]. The reader is referred to [57, 58] for the details of the guidance law, a Lyapunov-based proof of its asymptotic stability, and the flight test results demonstrating its efficacy.

4.2.3 Control Law

Figure 4.3 shows the internal structure of the control law, which consists of the nominal controller (NC) and the fault-tolerant controller (FTC). Both the NC and the FTC use the reference commands $(V_{cmd}, h_{cmd}, \phi_{cmd})$, the sensor measurements (y), and the navigation states $(\hat{x})$ to compute the control commands $(\delta_{tc}, \delta_{lc}, \delta_{rc})$. A switch selects the outputs of either the NC or the FTC depending on the fault information received by the control law. When a fault occurs, the control law switches instantaneously from the NC to the FTC. For simplicity, the states of the FTC are initialized at zero. The FTC also makes use of the fault information, e.g. in determining whether the fault has occurred in the left or the right elevon.

The control law described above is often referred to in the literature as the multiple model approach. This is because NC and FTC are designed using two different, but related, aircraft models. This approach requires an explicit fault diagnosis algorithm since there is a discrete switch in the operational controller. This is in contrast to adaptive control approaches, wherein the controller parameters are varied in response to the fault(s). Adaptive control approaches do not require a discrete switch and, thus, do not always require explicit fault diagnosis. The multiple model approach is chosen for this particular problem because NC and FTC have different feedback architectures. In particular, after one of the elevons fails, there are only two inputs available to control the six degrees of freedom of the aircraft. As Section 7.4 will demonstrate, changes to the feedback architecture are essential for controlling the most important motion variables.

[1] `http://ardupilot.org`

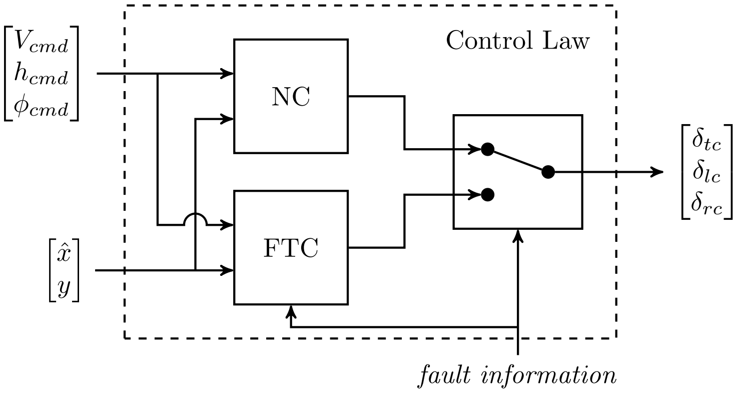

$\begin{bmatrix} V_{cmd} \\ h_{cmd} \\ \phi_{cmd} \end{bmatrix}$
Control Law
NC
FTC
$\begin{bmatrix} \hat{x} \\ y \end{bmatrix}$
$\begin{bmatrix} \delta_{tc} \\ \delta_{lc} \\ \delta_{rc} \end{bmatrix}$
fault information

Figure 4.3: The control law consist of the nominal controller (NC) and the fault-tolerant controller (FTC). When a fault occurs, the control law switches from NC to FTC.

Such architecture changes are difficult to implement without a discrete switch. Of course, this does not preclude the existence of a solution based on adaptive control.

4.3 Nominal Controller Architecture

Figure 4.4 shows the architecture of the nominal controller, i.e. the contents of the NC block in Figure 4.3. It consists of four components: a total energy controller K_{TECS}, a pitch attitude controller K_E, a roll attitude controller K_A, and a transformation block $T_{lr\leftarrow ea} = \left[\begin{smallmatrix} 1 & -1 \\ 1 & 1 \end{smallmatrix}\right]$ that converts the elevator and aileron commands to elevon commands (see the text following Equation 3.13). This structure enables the virtual elevator to be used for pitch control and the virtual aileron to be used for roll control.

Each component of NC is designed and analyzed with the aid of the linear model corresponding to the nominal trim airspeed of 15.4 m s^{-1} (Section 2.4.1). K_{TECS} and K_E are designed using G_{lon}, which has the state $x_{lon} = [u, w, q, \theta, Z_e]^T$, the input $u_{lon} = [\delta_t, \delta_e]^T$, the output $y_{lon} = [V, q, \theta, h]^T$, and the state-space representation:

$$A_{lon} = \begin{bmatrix} -0.151 & 0.753 & -1.02 & -9.78 & 0 \\ -0.883 & -5.69 & 13.9 & -0.668 & 0 \\ 0.878 & -12.9 & -5.49 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ -0.0681 & 0.998 & 0 & -15.4 & 0 \end{bmatrix},\ B_{lon} = \begin{bmatrix} 6.53 & 0.146 \\ 0 & -24.5 \\ 0 & -186 \\ 0 & 0 \\ 0 & 0 \end{bmatrix},$$

$$C_{lon} = \begin{bmatrix} 0.998 & 0.0681 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix},\ \text{and}\ D_{lon} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}. \tag{4.1}$$

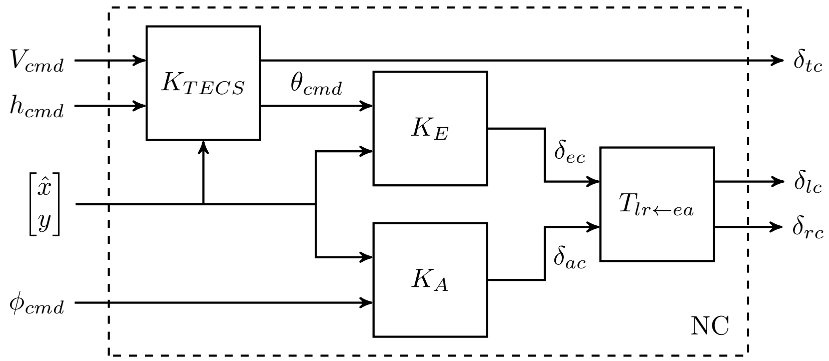


Figure 4.4: The architecture of the nominal controller (NC).

K_A is designed using G_{lat}, which has the state $x_{lat} = [v, p, r, \phi]^T$, the input $u_{lat} = \delta_a$, the output $y_{lat} = [\phi, p]^T$, and the state-space representation:

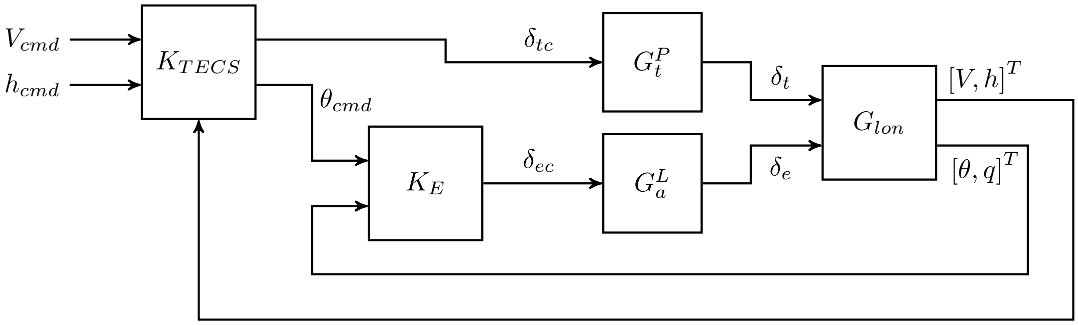
$$A_{lat} = \begin{bmatrix} -0.574 & 1.12 & -15.3 & 9.78 \\ -3.99 & -11.3 & 2.5 & 0 \\ 0.311 & -1.49 & -0.944 & 0 \\ 0 & 1 & 0.0683 & 0 \end{bmatrix},\ B_{lat} = \begin{bmatrix} -0.488 \\ -201 \\ -9.61 \\ 0 \end{bmatrix},\ C_{lat} = \left[\begin{smallmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{smallmatrix}\right],\ \text{and } D_{lat} = \left[\begin{smallmatrix} 0 \\ 0 \end{smallmatrix}\right]. \tag{4.2}$$

In addition to G_{lon} and G_{lat}, the control design incorporates the low-order equivalent systems (Section 3.3.9) that model the actuator dynamics and the time delay.

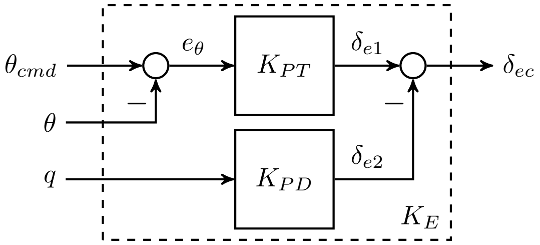
4.4 Pitch and Roll Attitude Controllers

Figure 4.5a shows the models describing the longitudinal aircraft dynamics G_{lon}, the throttle actuator G_t^P, and the elevator actuator G_a^L in feedback with K_{TECS} and K_E. This section focuses on the design of K_E and Section 4.5 focuses on the design of K_{TECS}. Figure 4.5b shows the architecture of the pitch attitude controller K_E, which comprises a pitch tracker K_{PT} and a pitch damper K_{PD}. The pitch tracker regulates the error in the pitch angle ($\theta_{cmd} - \theta$) using a PI controller. The pitch damper uses a proportional gain on the pitch rate. The output of K_E is the elevator command δ_{ec}.

Figure 4.6 shows the models describing the lateral-directional aircraft dynamics G_{lat} and the aileron actuator G_a^L in feedback with the roll attitude controller K_A, which comprises a roll tracker K_{RT} and a roll damper K_{RD}. The roll tracker regulates the error in the roll angle ($\phi_{cmd} - \phi$) using a PI controller. The roll damper uses a proportional gain on the roll rate. The output of K_A is the aileron command δ_{ac}.



(a) K_{TECS} and K_E are designed using the longitudinal aircraft model.



(b) The architecture of the pitch attitude controller K_E.

Figure 4.5: The longitudinal controller consists of K_{TECS} and K_E.

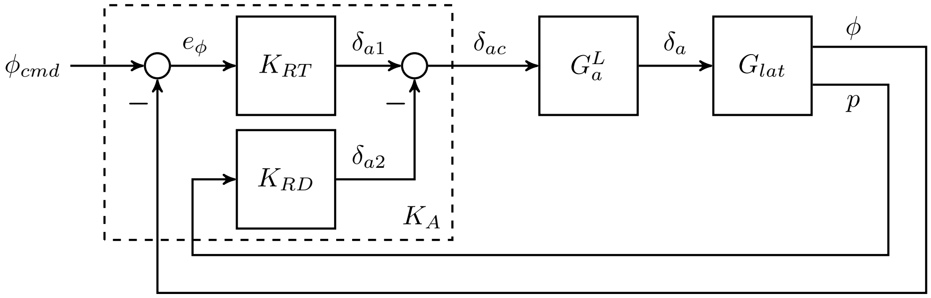


Figure 4.6: The architecture of the roll attitude controller K_A, which is designed using the lateral-directional aircraft model.

The performance and robustness targets specified in Table 4.1 guide the design of the pitch and the roll attitude controllers. In particular, the performance targets are divided into a primary target and a secondary target. The primary performance target is to achieve a desired bandwidth in the closed-loop complementary sensitivity function. The secondary performance target is to achieve a desired damping ratio in the dominant rigid body mode, such that Level-1 flying quality is guaranteed. The robustness targets are specified in terms of the minimum desired gain, phase, and delay margins.

Table 4.1: The performance and robustness targets that are used in tuning the pitch and the roll attitude controllers.

Target	**Pitch attitude controller K_E**	**Roll attitude controller K_A**
Performance (primary)	Bandwidth of 1.7 $\mathrm{rad\,s^{-1}}$ in the $\theta_{cmd} \rightarrow \theta$ channel, i.e. twice the natural frequency of the phugoid mode.	Bandwidth of 1.4 $\mathrm{rad\,s^{-1}}$ in the $\phi_{cmd} \rightarrow \phi$ channel, i.e. thrice the natural frequency of the desired cross-track error dynamics.
Performance (secondary)	Short period mode damping ratio of 0.5, i.e. Level-1 flying quality.	Dutch roll mode damping ratio of 0.38, i.e. Level-1 flying quality.
Robustness margins	Gain margin of ±6 dB, phase margin of ±45°, and delay margin of 0.22 s (approx. 20 frames).	Gain margin of ±6 dB, phase margin of ±45°, and delay margin of 0.22 s (approx. 20 frames).

The bandwidth target for K_E is sufficiently faster than that of the total energy controller, presented shortly. A short period mode damping ratio of 0.5 is roughly in the middle of the range required for Level-1 flying quality [26]. A dutch roll mode damping ratio of 0.38 is twice the minimum value required for Level-1 flying quality [26].

K_E and K_A are initially tuned using classical design techniques, e.g. the root locus method, and then iteratively updated using flight tests. Table 4.2 lists the final gains of K_E and K_A, along with the bandwidths and the robustness margins of the associated loop transfer functions. The loops are broken one-at-a-time, i.e. the dampers are included when computing the loop transfer functions of the trackers. Table 4.2 suggests that the primary performance target is achieved exactly for K_E and approximately for K_A. In particular, the achieved bandwidth of the roll tracker is slightly higher than the desired value of 1.4 $\mathrm{rad\,s^{-1}}$. The bandwidth of the pitch damper is not listed since its loop transfer function is below 0 dB. The bandwidth of the roll damper is 8.1 $\mathrm{rad\,s^{-1}}$, which is approximately twice the open-loop natural frequency of the dutch roll mode. Further, all the loops listed in Table 4.2 achieve the robustness targets of Table 4.1.

Table 4.3 lists the open-loop and the closed-loop poles at an airspeed of 15.4 $\mathrm{m\,s^{-1}}$. As seen from the table, the secondary performance targets are achieved for both K_E and K_A. The damping ratio of the phugoid mode is also increased in the closed-loop to 0.5. In addition, the natural frequencies of all the rigid body modes, except for the

Table 4.2: The final gains of K_E and K_A, and the bandwidths and the margins of the associated loop transfer functions.

Controller	**Gains (P/PI)**	**Loop transfer func.** $(I \rightarrow O)$	**Bandwidth** $(\mathrm{rad\,s^{-1}})$	**Gain margin** (dB)	**Phase margin** (°)	**Delay margin** (s)
Pitch damper	$K_{PD} = -0.05$	$\delta_{ec} \rightarrow \delta_{e2}$	loop below 0 dB	±7.53	–	–
Pitch tracker	$K_{PT} = -0.4 - \frac{0.2}{s}$	$e_\theta \rightarrow \theta$	1.7	±8.7	±79	0.82
Roll damper	$K_{RD} = -0.06$	$\delta_{ac} \rightarrow \delta_{a2}$	8.1	±7.84	±109	0.24
Roll tracker	$K_{RT} = -0.34 - \frac{0.086}{s}$	$e_\phi \rightarrow \phi$	1.8	±11.5	±86.4	0.85

Table 4.3: The open-loop and the closed-loop poles of the Vireo model at 15.4 $\mathrm{m\,s^{-1}}$.

	Open-loop		**Closed-loop**	
Mode	ζ	ω_n $(\mathrm{rad\,s^{-1}})$	ζ	ω_n $(\mathrm{rad\,s^{-1}})$
Short period	0.39	14.5	0.56	17.3
Phugoid	0.094	0.87	0.5	1.1
Roll subsidence	1	12	1	20
Dutch roll	0.13	4.1	0.38	3.6
Spiral	1	0.12	1	0.28

dutch roll mode, are higher in the closed-loop as compared to the open-loop.

4.5 Total Energy Controller

The concept of total energy control was introduced by Lambregts in 1983 [59–62]. At the time it was introduced, the state-of-the-art in operational longitudinal flight control involved two single-input single-output loops: an autopilot and an auto-throttle. The autopilot used the elevator to control the pitch angle, which was in turn used to control the altitude. The auto-throttle used the throttle to control the airspeed. A problem with this approach was the lack of coordination between the elevator and the throttle commands, a task that human pilots performed well. Lambregts posed the problem of controlling the altitude and the airspeed into a problem of controlling the potential and the kinetic energies of the aircraft. In this approach, the controlled variables were the sum (also called the total) and the difference (also called the balance) of the two energies. Lambregts' solution was to control the total energy using the throttle and the balance energy using the elevator. This solution, called the total energy control system (TECS), affords better tracking performance as compared to the previous approach.

TECS controllers have been tested on several types of aircraft, including an experimental Boeing 737 [63, 64], fighter aircraft [65, 66], and a small tail-sitter UAV [67]. Moreover, TECS is compatible with several control design methods, e.g. optimal control [68], eigenstructure assignment [69], L_1 adaptive control [70], and adaptive nonlinear control [71]. However, the most common approach, including the one followed by this thesis, uses proportional-integral controllers to track the total and the balance energies.

Figure 4.7 shows the architecture of the nominal total energy controller K_{TECS}, which comprises an energy calculation block, a total energy tracker K_{TE}, and a balance energy tracker K_{BE}. The energy calculation block uses the commanded and the measured values of the airspeed V and the altitude h to compute the total energy error ΔE and the balance energy error ΔB. These calculations happen over several stages.

First, the commanded and the current potential energies are calculated as:

$$U_{cmd} = U_0 + mgh_{cmd} \quad \text{and} \quad U = U_0 + mgh, \tag{4.3}$$

where U_0 is the potential energy at any reference altitude. The potential energy error

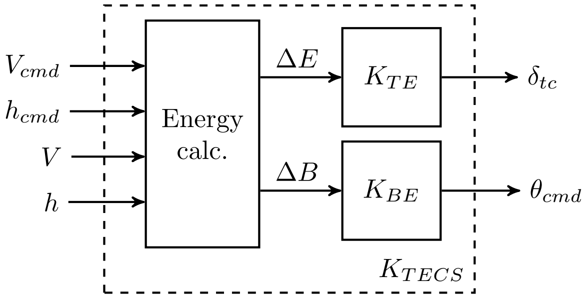


Figure 4.7: The architecture of the nominal total energy controller K_{TECS}.

is $\Delta U = U_{cmd} - U$. Next, the commanded and the current kinetic energies are:

$$K_{cmd} = \frac{1}{2} m V_{cmd}^2 \quad \text{and} \quad K = \frac{1}{2} m V^2. \tag{4.4}$$

The kinetic energy K is defined with respect to the airspeed and not the (inertial) ground speed. The kinetic energy error is $\Delta K = K_{cmd} - K$. Finally, defining the total energy as $E = K + U$ and the balance energy as $B = K - U$, their respective errors are $\Delta E = \Delta K + \Delta U$ and $\Delta B = \Delta K - \Delta U$. K_{TE} and K_{BE} each implement a PI controller to drive ΔE and ΔB to zero, respectively.

The total energy controller is nonlinear because of the inclusion of V^2 in the kinetic energy term. In order to apply linear control design techniques, the energy calculation block is linearized about the nominal trim airspeed of 15.4 $\mathrm{m\,s^{-1}}$, yielding:

$$\begin{bmatrix} \Delta \tilde{E} \\ \Delta \tilde{B} \end{bmatrix} = \begin{bmatrix} m\bar{V} & mg & -m\bar{V} & -mg \\ m\bar{V} & -mg & -m\bar{V} & mg \end{bmatrix} \begin{bmatrix} \tilde{V}_{cmd} & \tilde{h}_{cmd} & \tilde{V} & \tilde{h} \end{bmatrix}^T, \tag{4.5}$$

where the tildes represent the perturbations of the variables about the trim point.

Since the total energy controller introduces cross-coupling between the airspeed and the altitude tracking loops, the gains are initially tuned one-controller-at-a-time using the following method. The gains of K_{TE} are initially fixed at very low values and K_{BE} is tuned to achieve a favorable response in the $V_{cmd} \to V$ channel. This makes the assumption that the effect of θ_{cmd} is primarily to change the kinetic energy. Then, the gains are K_{BE} are fixed at their tuned values and K_{TE} is tuned to achieve a favorable response in the $h_{cmd} \to h$ channel. This makes the assumption that the effect of δ_{tc}

is primarily to change the potential energy. While these two assumptions are certainly not true, this method yields gains that work reasonably well in practice. After the initial tuning, the gains are iteratively updated using flight tests, finally resulting in $K_{TE} = 6 \times 10^{-4} + \frac{4\times10^{-5}}{s}$ and $K_{BE} = -1.2 \times 10^{-3} + \frac{-9.4\times10^{-5}}{s}$.

4.5.1 Saturations

The autopilot includes saturation blocks to bound various commands. Recall from Section 2.4.1 that the stall airspeed and the high speed limit of the Vireo are approximately 12 $\mathrm{m\,s^{-1}}$ and 20.5 $\mathrm{m\,s^{-1}}$, respectively. In order to prevent the aircraft from breaching these limits, the guidance law saturates V_{cmd} in the interval $[13, 18]$ $\mathrm{m\,s^{-1}}$ and ϕ_{cmd} in the interval $[-35, +35]\,^{\circ}$. The total energy controller saturates θ_{cmd} in the interval $[-10, +25]\,^{\circ}$. In addition, the throttle command is bounded within $[0, 1]$ and the left and the right elevon deflection commands are bounded within $[-30, +20]\,^{\circ}$.

4.6 Robustness Analysis

A disk margin analysis is conducted to assess the robustness of K_A, K_E, and K_{TECS} separately, as shown in Table 4.4. For simplicity, only the input-to-plant loop transfer function L_i associated with each controller is analyzed. A good rule of thumb is a disk margin of at least 0.4 on L_i, which corresponds to a minimum disk gain margin of $[0.71, 1.67]$, a minimum disk phase margin of $\pm 23°$, and a maximum H_∞ norm of 8 dB on the input-to-plant sensitivity function $S_i := (I + L_i)^{-1}$ [72]. The disk margins provide lower bounds on the classical gain and phase margins. The crossover frequency of L_i is a good measure of the bandwidth. The loops are broken individually as specified in the second column of Table 4.4. The bandwidths of the pitch and the roll attitude controllers are of the same order. Moreover, the bandwidth of K_E is sufficiently larger than that of K_{TECS}. The time delay margins of K_A, K_E, and K_{TECS} are 0.032 s, 0.024 s, and 11.6 s, respectively. All the loops listed in Table 4.4 achieve the minimum desired disk gain and phase margins and the maximum desired $\|S_i\|$.

Table 4.4: The disk margins of the loops associated with the components of the nominal controller.

Controller	**Input-to-plant loop transfer func.** L_i	**Loop crossover** $(\mathrm{rad\,s^{-1}})$	$\|S_i\|$ (dB)	**Disk gain margin** $(-)$	**Disk phase margin** (°)	**Critical frequency** $(\mathrm{rad\,s^{-1}})$
Roll attitude controller K_A	Break loop at δ_{ac} in Figure 4.6	2.5	5.87	$[0.5, 2]$	±37	20
Pitch attitude controller K_E	Break loop at δ_{ec} in Figure 4.5a	2	7.6	$[0.59, 1.7]$	±29	21
Total energy controller K_{TECS}	Break loop at δ_{tc} in Figure 4.5a	0.062	2.87	$[0.47, 2.15]$	±40	0.06

4.7 Flight Test Results

This section presents flight test data to validate the nominal controller. In all the time history plots, the dashed line denotes the autopilot command and the solid line denotes the aircraft response. Further, although each flight lasts for about 10 min, the plots only show the snippets of the flight data that are relevant to the validation. Appendix C lists short descriptions of all the flights conducted to date on the Vireo.

4.7.1 Lateral-Directional Controller Validation

After completing the gain tuning, the lateral-directional controller (K_A) is validated during FLT15 on 2017-08-11. FLT15 includes a bowtie maneuver that excites the roll tracking channel. Figure 4.8 shows the ground track of the bowtie maneuver in the local North-East reference frame. The origin of this frame coincides with the center of the bowtie. The maneuver starts at ×, ends at ∘, and lasts for around four minutes. The dimensions of the bowtie are defined by the four waypoints ($*$) shown on the outer edges of the plot. These waypoints are positioned at 300 m from the bowtie center and at azimuths of 70°, 110°, 290°, and 250° with respect to North. The Vireo traces the bowtie pattern by flying to each waypoint in sequence and then repeating the sequence. The desired trajectory is obtained by connecting consecutive waypoints using straight line segments, as explained in Section 4.2.1 and Figure 4.2.

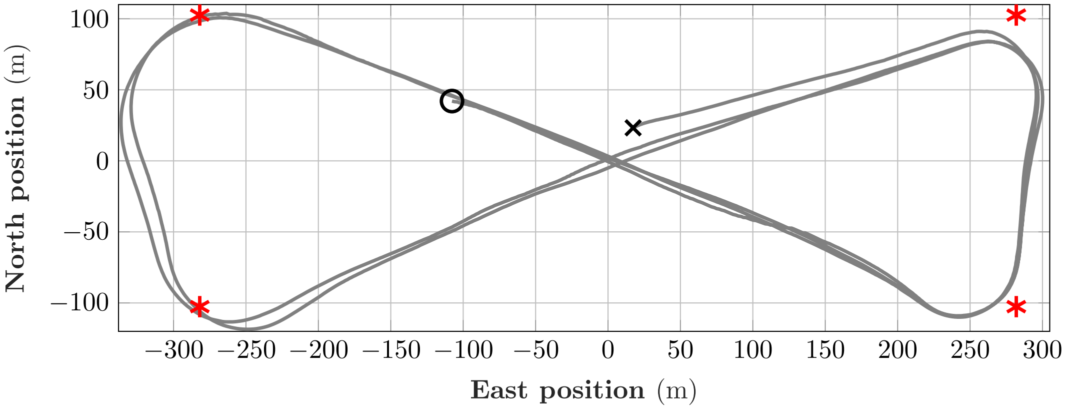


Figure 4.8: FLT15: The ground track of the Vireo depicting the bowtie maneuver.

Figure 4.9 shows the time histories of the roll and the course angles during the bowtie maneuver. Upon reaching each waypoint, the course command changes by approximately 110°, e.g. at 1043 s, in order to direct the Vireo to the next waypoint. The resulting course error saturates the roll angle command at 35°, thus producing a banked turn. The easterly end of the bowtie produces a pair of right banked turns and the westerly end of the bowtie produces a pair of left banked turns. The rise time and the overshoot of the roll angle, when averaged across all the banked turns, are approximately 1 s and 18%, respectively. The RMS roll tracking error over the time segment shown in Figure 4.9 is 5.3°. The RMS course tracking error, if computed over the same time segment, is on the order of 29°. This is because the desired trajectory is expressed using straight line segments that connect consecutive waypoints (Section 4.2.1). Since the turn rate of the aircraft has a maximum limit, the course error is large when the aircraft transitions from one waypoint to the next, e.g. whenever the roll angle command saturates in Figure 4.9. However, if the RMS course tracking error is computed only over the straight line segments of the bowtie, then it is on the order of 3.7°.

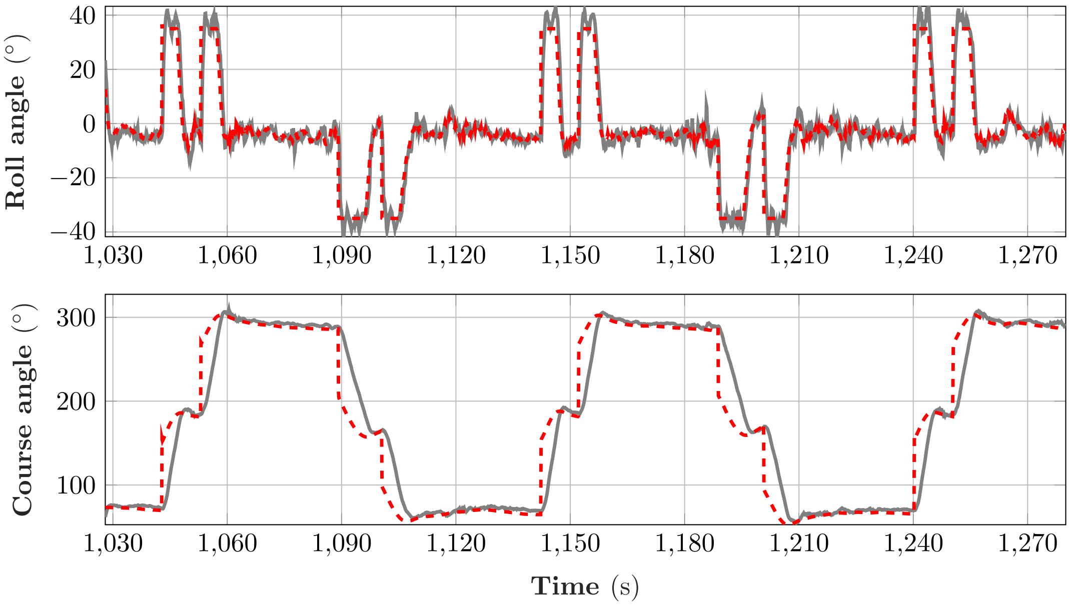


Figure 4.9: FLT15: The roll and the course angles during the bowtie maneuver.

4.7.2 Longitudinal Controller Validation

After completing the gain tuning, the longitudinal controller (K_{TECS} and K_E) is validated during FLT18 on 2017-08-23. FLT18 includes step commands in the altitude and the airspeed. Figure 4.10 shows the time histories of the airspeed, the altitude above sea level (ASL), the pitch angle, and the throttle during a two minute-long segment of this flight. At the beginning of this segment, the Vireo is in a left banked turn of 100 m radius. This is a large turning radius, given the dimensions of the aircraft, and allows the test points to be executed over several minutes. The Vireo is initially trimmed at a constant altitude of 350 m ASL and a constant airspeed of 15.4 $\mathrm{m\,s^{-1}}$. The corresponding trim pitch attitude and throttle are approximately 5° and 0.69, respectively.

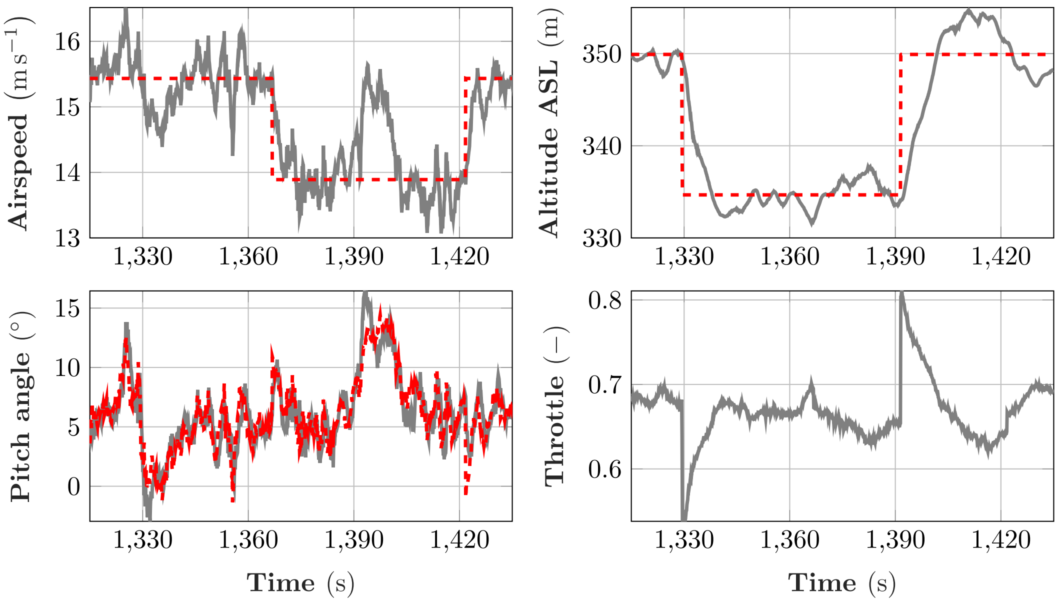


Figure 4.10: FLT18: Altitude and airspeed step commands are used to validate the longitudinal controller.

At 1329 s, the altitude command is decreased to 335 m ASL. The total energy controller responds by decreasing the throttle to 0.54, thus causing the aircraft to lose altitude. The altitude takes about 10 s to reach its target, but overshoots it by 18% and settles after an additional 10 s. The altitude loss produces transient responses in the airspeed and the pitch angle, both of which take about 10 s to settle. At 1367 s, the airspeed command is decreased to 13.9 m s^{-1}, which amounts to a new trim condition. The total energy controller responds by increasing the target pitch attitude, thus causing the aircraft to slow down. Although the target pitch attitude increases momentarily to 11°, its new trim value does not appear to increase much. The airspeed takes about 3 s to reach its target, but overshoots it by approximately 13%. The airspeed reduction produces small transients in the altitude and the throttle responses.

Subsequently, the altitude command is increased back to 350 m ASL at 1391 s. This produces a larger overshoot in the altitude, and larger transients in the airspeed and the pitch angle, than before. The airspeed command is increased back to 15.4 m s^{-1} at 1422 s, causing the pitch attitude to decrease back to its original trim value. The RMS tracking errors over the segment shown in Figure 4.10 are: 0.52 m s^{-1} in the airspeed, 3.8 m in the altitude, and 1.6° in the pitch angle. When recomputed over a two minute-long time segment that does not include reference command changes, the RMS values improve to: 0.33 m s^{-1} in the airspeed, 1.3 m in the altitude, and 1.4° in the pitch angle.

Figure 4.11 shows the roll angle, the course angle, and the elevon deflections during the altitude and the airspeed steps commands. As seen in the plot, the tracking of the roll and the course angles are unaffected by the altitude and the airspeed steps. The RMS tracking errors are: 1.5° in the roll angle and 3.6° in the course angle.

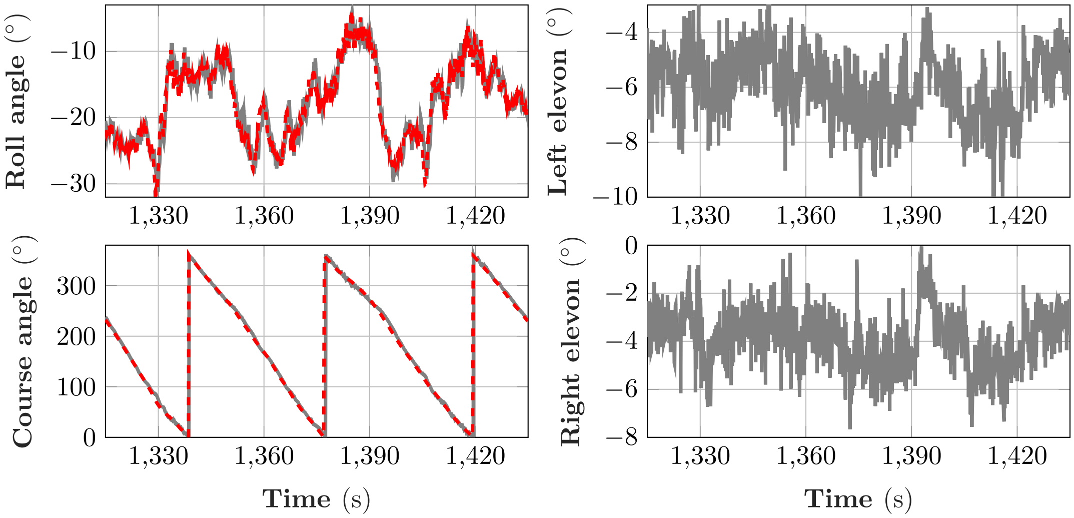


Figure 4.11: FLT18: The roll and course angles during the altitude and airspeed steps.

Chapter 5

Robust Estimation and Feedforward Control

5.1 Introduction

Section 2.4.2 of Chapter 2 describes the linearization of the nonlinear model of the Vireo at multiple constant airspeed flight conditions. The resulting collection of linear models, which are parametrized on the airspeed, can be analyzed as a gridded linear parameter-varying (LPV) system. This chapter presents some theoretical results for the synthesis of robust output estimators and disturbance feedforward controllers for uncertain, gridded LPV systems [73]. These results are then used during the design of the fault diagnosis algorithm. The reader who is only interested in the fault detection and isolation algorithm may skip ahead to Chapter 6.

Robust estimator and feedforward synthesis problems have been widely studied in the literature under various assumptions on the plant and uncertainty. For example, robust estimator synthesis results have been obtained for linear time-invariant (LTI) [74–83], linear time-varying (LTV) [84], and linear parameter-varying (LPV) [85, 86] plants. Previous work has also considered different classes of uncertainties including structured LTI [74, 75], single full block [84], norm-bounded time-varying [76–78], and polytopic [87–89] uncertainties. Moreover, robust estimator synthesis results have been obtained for uncertainties described by static [79, 80] and dynamic [81–83] integral quadratic constraints (IQC). In many of these previous works, convex formulations have been obtained for the synthesis. This is in contrast to the more general robust feedback synthesis problem which is non-convex, thus requiring heuristic approaches such as DK-synthesis [90] or IQC-synthesis [91–93]. The disturbance feedforward problem is structurally the dual of the output estimation problem [90]. As a result, many of the previous results summarized above have parallel results for the robust feedforward synthesis problem [81, 94–98]. For example, robust feedforward synthesis results have been obtained for LTI plants under structured LTI, LTV, or nonlinear uncertainties [94], LPV plants under mixed LTI and time-varying uncertainties [95], and LTI plants under uncertainties described by dynamic IQCs [81,96]. In addition, feedforward synthesis results have been obtained for LPV plants with polytopic parameter spaces [97], and LPV plants with bounded parameter rates-of-variation using parameter-dependent Lyapunov functions [98].

This chapter complements the existing literature by deriving convex conditions for the synthesis of output estimators and disturbance feedforward controllers for continuous-time, uncertain LPV systems. The uncertain system is an interconnection of a nominal gridded LPV system and a block structured perturbation that is described using dynamic IQCs. IQCs provide a general framework to characterize the input-output behavior of several different classes of perturbations [99], e.g. LTI uncertainties, static nonlinearities, time delays, etc. A frequency-domain stability theorem was formulated in [99] to analyze a feedback interconnection of a LTI plant and any perturbation that is characterizable using IQCs. However, gridded LPV systems are time-varying and hence they do not have a valid frequency response interpretation [41,100]. Consequently, a theorem was formulated in the time-domain using dissipativity theory for the input-output analysis of uncertain, gridded LPV systems [101, 102] by building on the work of [103]. This chapter utilizes the main result of [102] for the convex synthesis of estimators and feedforward controllers for uncertain, gridded LPV systems.

The most closely related works in the literature presented convex solutions for robust LTI synthesis using dynamic IQCs [82, 96]. In particular, convex synthesis conditions were derived for the robust output estimation problem [82]. A frequency-domain duality result was also developed to synthesize feedforward controllers using the (convex) conditions for the estimator synthesis [96]. It was later shown that the estimator and feedforward synthesis problems are special cases of a feedback structure that has no uncertainties in the control channel of the closed-loop [104]. For such feedback structures, [85] provided a general synthesis framework for robust, gain-scheduled controllers. This general synthesis framework is restricted to linear fractional transform (LFT) based LPV plants, whose state matrices are restricted to depend rationally on the scheduling parameters [105–107]. Note that frequency-domain arguments are applicable for LFT-based LPV systems since the nominal plant is LTI. Such frequency-domain arguments are not applicable for gridded LPV systems because, as noted above, these systems do not have a valid frequency response interpretation. This chapter instead develops a time-domain duality result for feedforward synthesis.

Before presenting the two main contributions, some background on IQCs and LPV systems is presented (Section 5.2). This discussion includes previous work on the stability and input-output analysis of uncertain LTI [103, 108] and LPV [101, 102] systems using dissipativity theory. The first main contribution is a time-domain notion of duality for uncertain, gridded LPV systems (Section 5.3). This is needed to exploit the duality between the estimation and feedforward problems [90]. The second main contribution is a rigorous convex solution for the robust output estimation problem (Section 5.4.1) for uncertain, gridded LPV systems. Finally, a convex solution is obtained for the robust disturbance feedforward problem by combining the two main contributions (Section 5.4.2). A numerical example is used to demonstrate feedforward synthesis for a gridded LPV plant that is affected by a sector-constrained nonlinearity (Section 5.5).

5.2 Background

5.2.1 Notation

Most notation used is from [90]. $\mathbb{R}$ and $\mathbb{C}$ denote the set of real and complex numbers. $\mathbb{RL}_\infty$ denotes the set of rational functions with real coefficients that are proper and have no poles on the imaginary axis. $\mathbb{RH}_\infty$ is the subset of functions in $\mathbb{RL}_\infty$ that are analytic in the closed right half of the complex plane. $\mathbb{R}^n$ denotes the set of $n \times 1$ vectors and $\mathbb{R}^{m \times n}$ denotes the set of $m \times n$ matrices whose elements are in $\mathbb{R}$. Similar notation is used for the sets $\mathbb{C}$, $\mathbb{RL}_\infty$, and $\mathbb{RH}_\infty$. $\mathbb{R}^+$ denotes the set of nonnegative real numbers. For a matrix $M \in \mathbb{C}^{m \times n}$, M^T denotes the transpose and M^* denotes the Hermitian adjoint. $\star$ denotes a symmetric block in matrices. $L_2^n [0, \infty)$ is the space of functions $v : [0, \infty) \to \mathbb{R}^n$

satisfying $\|v\| < \infty$, where $\|v\| := \sqrt{\int_0^\infty v(t)^T v(t)\,dt}$. For $v \in L_2^n[0,\infty)$, v_T is the truncated function: $v_T(t) = v(t)$ for $t \leq T$ and $v_T(t) = 0$ otherwise. The extended space, denoted L_{2e}, is the set of functions v such that $v_T \in L_2\ \forall T \geq 0$. The para-Hermitian conjugate of $H \in \mathbb{RL}_\infty^{m\times n}$ is defined as $H^\sim(s) := H(-s)^T$. Finally, $\mathcal{F}_u(G,\Delta)$ denotes the LFT of G and Δ, where Δ closes the upper channels of G.

5.2.2 Integral Quadratic Constraints

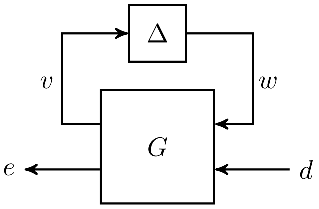


Figure 5.1: Interconnection of gridded LPV plant G and perturbation Δ.

Figure 5.1 shows the type of *uncertain* LPV systems considered in this chapter. G is a *nominal* grid-based LPV system, described further in Section 5.2.3. Δ is a block-structured perturbation [90] whose input-output behavior is described using IQCs. IQCs were introduced in [99] and are defined using frequency-domain multipliers $\Pi : j\mathbb{R} \to \mathbb{C}^{(n_v+n_w)\times(n_v+n_w)}$ that are measurable Hermitian-valued functions. The signals $v \in L_2^{n_v}[0,\infty)$ and $w \in L_2^{n_w}[0,\infty)$ satisfy the IQC defined by Π if:

$$\int_{-\infty}^{\infty} \begin{bmatrix} \hat{v}(j\omega) \\ \hat{w}(j\omega) \end{bmatrix}^* \Pi(j\omega) \begin{bmatrix} \hat{v}(j\omega) \\ \hat{w}(j\omega) \end{bmatrix} d\omega \geq 0, \tag{5.1}$$

where $\hat{v}(j\omega)$ and $\hat{w}(j\omega)$ are the Fourier transforms of v and w, respectively. A bounded, causal operator $\Delta : L_{2e}^{n_v}[0,\infty) \to L_{2e}^{n_w}[0,\infty)$ satisfies the IQC defined by Π if (5.1) holds for all $v \in L_2^{n_v}[0,\infty)$ and $w = \Delta(v)$. This concept can be formalized using graph theory. First, let $\mathrm{IQC}(\Pi)$ denote the set of signals $v \in L_2^{n_v}[0,\infty)$ and $w \in L_2^{n_w}[0,\infty)$ that satisfy inequality (5.1). Further, define the graph of Δ as $\mathcal{G}_\Delta := \{[v; \Delta(v)] : v \in L_2^{n_v}[0,\infty)\}$ [109]. Then Δ satisfies the IQC defined by Π if $\mathcal{G}_\Delta \subset \mathrm{IQC}(\Pi)$. This chapter will use the simpler (but equivalent) notation $\Delta \in \mathrm{IQC}(\Pi)$. As such, a *set of operators* $\mathbf{\Delta}_\Pi$ satisfy the IQC defined by Π, i.e. $\mathbf{\Delta}_\Pi := \{\Delta : \Delta \in \mathrm{IQC}(\Pi)\}$. Hence, the original uncertainty or uncertainty set is subsumed under the larger set $\mathbf{\Delta}_\Pi$.

Next, consider the following special class of multipliers.

Definition 1 ([110])**.** *Let* $\Pi = \Pi^\sim \in \mathbb{RL}_\infty^{(n_v+n_w)\times(n_v+n_w)}$ *be partitioned as* $\begin{bmatrix} \Pi_{11} & \Pi_{12} \\ \Pi_{12}^\sim & \Pi_{22} \end{bmatrix}$*, where* $\Pi_{11} \in \mathbb{RL}_\infty^{n_v \times n_v}$ *and* $\Pi_{22} \in \mathbb{RL}_\infty^{n_w \times n_w}$*.* Π *is said to be a strict positive-negative (PN) multiplier if* $\Pi_{11}(j\omega) > 0$ *and* $\Pi_{22}(j\omega) < 0$ $\forall \omega \in \mathbb{R} \cup \{\infty\}$*.*

Note that the strict PN conditions imply that $\Pi(j\omega)$ has n_v positive eigenvalues and n_w negative eigenvalues for all $\omega \in \mathbb{R} \cup \{\infty\}$. This follows from the Courant-Fischer minimax Theorem [111]. Thus strict PN multipliers have constant inertia on the extended imaginary axis. The remainder of this chapter will exclusively make use of strict PN multipliers.

Since this chapter employs IQCs for the robustness analysis of uncertain grid-based LPV systems, IQCs also need to be expressed in the time-domain. A multiplier $\Pi \in \mathbb{RL}_\infty^{(n_v+n_w)\times(n_v+n_w)}$ can be factorized as $\Pi = \Psi^\sim M \Psi$, where $M = M^T \in \mathbb{R}^{n_z \times n_z}$ and $\Psi \in \mathbb{RH}_\infty^{n_z \times (n_v+n_w)}$ [103]. The IQC given in (5.1) can be rewritten as:

$$\int_0^\infty z(t)^T M z(t) dt \geq 0, \tag{5.2}$$

where $z := \Psi \begin{bmatrix} v \\ w \end{bmatrix} \in \mathbb{R}^{n_z}$ is the output of the linear system Ψ driven by the input signals v and w starting from zero initial conditions [103]. Let Ψ have the state-space realization:

$$\begin{bmatrix} \dot{x}_\Psi \\ z \end{bmatrix} = \begin{bmatrix} A_\Psi & B_{\Psi v} & B_{\Psi w} \\ C_\Psi & D_{\Psi v} & D_{\Psi w} \end{bmatrix} \begin{bmatrix} x_\Psi \\ v \\ w \end{bmatrix}, \tag{5.3}$$

where $x_\Psi \in \mathbb{R}^{n_\Psi}$ and $x_\Psi(0) = 0$. While there are infinite ways to factorize Π, this chapter will use the following special class of factorizations.

Definition 2 ([112, 113])**.** $\left(\hat{\Psi}, J_{n_v,n_w}\right)$ *is called a* J_{n_v,n_w}*-spectral factor of* $\Pi = \Pi^\sim \in \mathbb{RL}_\infty^{(n_v+n_w)\times(n_v+n_w)}$ *if* $\Pi = \hat{\Psi}^\sim J_{n_v,n_w} \hat{\Psi}$*,* $J_{n_v,n_w} = \left[\begin{smallmatrix} I_{n_v} & 0 \\ 0 & -I_{n_w} \end{smallmatrix}\right]$*, and the systems* $\hat{\Psi}, \hat{\Psi}^{-1} \in \mathbb{RH}_\infty^{(n_v+n_w)\times(n_v+n_w)}$.

J-spectral factorizations are special because J is diagonal and $\hat{\Psi}$ is square, stable, and stably invertible. J-spectral factorizations exist for all strict PN multipliers (Lemma 4 in [103]). The remainder of this chapter will use (Ψ, M) to denote general factorizations of Π and $\left(\hat{\Psi}, J\right)$ to denote J-spectral factorizations of Π.

5.2.3 Input-Output Analysis of LPV Systems

LPV systems are a class of systems whose state-space matrices depend on a time-varying parameter $\rho : \mathbb{R}^+ \to \mathbb{R}^{n_\rho}$. The rate-of-variation of ρ is denoted by $\dot{\rho} : \mathbb{R}^+ \to \mathbb{R}^{n_\rho}$. Consider the following state-space realization for the LPV system G (Figure 5.1):

$$\begin{bmatrix} \dot{x}_G \\ y \end{bmatrix} = \begin{bmatrix} A_G(\rho) & B_G(\rho) \\ C_G(\rho) & D_G(\rho) \end{bmatrix} \begin{bmatrix} x_G \\ u \end{bmatrix}, \tag{5.4}$$

where $x_G \in \mathbb{R}^{n_G}$ is the state, $u = \begin{bmatrix} w \\ d \end{bmatrix} \in \mathbb{R}^{n_w + n_d}$ are the inputs, and $y = \begin{bmatrix} v \\ e \end{bmatrix} \in \mathbb{R}^{n_v + n_e}$ are the outputs. The matrices in (5.4) have dimensions compatible with these signals and are continuous functions of ρ. In the remainder of the chapter, the functional dependence of the state matrices on ρ is occasionally suppressed for brevity.

In general, the matrices in (5.4) are only assumed to be continuous functions of ρ. This chapter follows the grid-based approach, wherein the parameter space is discretized. While an advantage of this approach is that the state matrices can be arbitrary functions of ρ, a disadvantage is that any analysis is conducted over a finite number of grid points [41,100]. The other major modeling approach involves expressing the LPV system as a LFT of a LTI plant and a scheduling block. While an advantage of the LFT-based approach is that it does not require the parameter space to be discretized, a disadvantage is that the state matrices are restricted to depend rationally on ρ [105–107].

This chapter assumes that ρ: (i) is an exogenous signal that is independent of x_G, u, and y, (ii) is a continuous function of time, and (iii) is restricted to a known compact set $\mathcal{P} \subset \mathbb{R}^{n_\rho}$. Further, this chapter assumes no knowledge of bounds on $\dot{\rho}$. Consequently, the results stated assume that ρ can vary arbitrarily fast with time. However, with additional notation, the main results can be adapted for the case where $\dot{\rho}$ has known bounds, by using parameter-dependent Lyapunov matrices (e.g. [102]). Following the notation of [102], the set of admissible trajectories is defined as:

$$\mathcal{T} \coloneqq \left\{\rho : \mathbb{R}^{+} \to \mathbb{R}^{n_\rho} \,:\, \rho \in \mathcal{C}^{0} \,\text{and}\, \rho\left(t\right) \in \mathcal{P}\; \forall t \geq 0\right\}. \tag{5.5}$$

If ρ depends on x_G, u, or y then the LPV system is nonlinear from input to output. Such systems are often termed quasi-LPV. The remainder of the chapter assumes ρ has no dependence on x_G, u, and y (as per assumption (i) above) and hence the LPV system is linear for any parameter trajectory $\rho \in \mathcal{T}$. This chapter assumes that G is quadratically stable, as defined next.

Definition 3 ([41])**.** *G is quadratically stable if $\exists P > 0$ such that $A_G\left(\rho\right)^T P + PA_G\left(\rho\right) < 0\; \forall \rho \in \mathcal{P}$.*

As discussed in Section 1.2 of [41], quadratic stability is a form of internal state stability. In particular, if G is quadratically stable and autonomous, x_G exponentially decays to zero for any initial condition $x_G\left(0\right) \in \mathbb{R}^{n_G}$ and any admissible parameter trajectory $\rho \in \mathcal{T}$. This is proved after noting that $x_G^T P x_G$ is a Lyapunov function. In addition to internal state stability, this chapter requires a notion of input-output norm. In particular, define the induced L_2 norm of G as:

$$\|G\| \coloneqq \sup_{\substack{0 \neq u \in L_2^{nw+n_d} \\ \rho \in \mathcal{T},\, x_G(0)=0}} \frac{\|y\|}{\|u\|}. \tag{5.6}$$

The following lemma provides sufficient conditions for bounding $\|G\|$.

Lemma 1 ([41])**.** *G is quadratically stable and $\|G\| < \gamma$ for some $\gamma \in (0, \infty)$ if $\exists P > 0$ such that*

$$\begin{bmatrix} A_G^T\left(\rho\right)P + PA_G\left(\rho\right) & \star \\ B_G^T\left(\rho\right)P & -\gamma I \end{bmatrix} + \frac{1}{\gamma}\begin{bmatrix} C_G^T\left(\rho\right) \\ D_G^T\left(\rho\right) \end{bmatrix}\left(\star\right) < 0\; \forall \rho \in \mathcal{P}. \tag{5.7}$$

This lemma essentially generalizes the Bounded Real Lemma for LPV systems and follows from Theorem 3.3.1 of [41]. By applying the Schur complement lemma on the second term, inequality (5.7) can be written as a LMI involving the LPV plant G, a Lyapunov matrix P, and the gain upper bound γ. In order to find the least upper bound, a semidefinite program is formulated with γ as the linear cost function to be minimized subject to the LMI constraints $P > 0$ and (5.7). Henceforth, LMIs of the form (5.7) will be referred to using the short form $LMI_{BR}(G, P, \gamma) < 0$. The subscript BR indicates that it is a LMI associated with the Bounded Real Lemma and the arguments (G, P, γ) indicate that the system, Lyapunov matrix, and gain bound are involved.

On a related note, the plant G can be scaled at the inputs and/or outputs using γ in order to yield a gain bound of 1. For example, if B_G and C_G were each scaled by $\gamma^{-0.5}$ and D_G was scaled by γ^{-1}, then (5.7) can be rewritten with a gain bound of 1. Such normalizations will be used in Section 5.2.5.

5.2.4 Input-Output Analysis of Uncertain LPV Systems

The previous section considered the input-output analysis of *nominal* LPV systems. This section considers the input-output analysis of *uncertain* LPV systems, wherein the uncertainty is described using IQCs. The input-output analysis of $\mathcal{F}_u(G, \Delta)$, shown in Figure 5.1, was considered in [101, 102]. The induced L_2 norm from inputs d to outputs e is denoted by $\|\mathcal{F}_u(G, \Delta)\|$ and is defined in a manner similar to Equation (5.6). Since Δ contains nonlinearities and uncertainties that are hard to analyze, it is not always possible to compute $\|\mathcal{F}_u(G, \Delta)\|$ exactly. Instead the *set* of operators $\mathbf{\Delta}_\Pi$ containing Δ are used to define a worst-case gain:

$$\sup_{\Delta \in \mathbf{\Delta}_\Pi} \|\mathcal{F}_u(G, \Delta)\| . \tag{5.8}$$

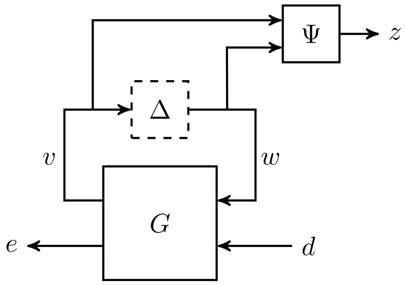


Figure 5.2: Interconnection of the gridded LPV system G, perturbation Δ, and IQC filter Ψ.

Next, consider Figure 5.2. In addition to the interconnection of G and Δ, the IQC factor Ψ is appended such that it is driven by signals v and w, and produces signal z. The extended LPV system, formed by the interconnection of G and Ψ, has the following state-space representation:

$$\begin{bmatrix} \dot{x}_e \\ z \\ e \end{bmatrix} = \begin{bmatrix} \mathcal{A}(\rho) & \mathcal{B}_w(\rho) & \mathcal{B}_d(\rho) \\ \mathcal{C}_z(\rho) & \mathcal{D}_{zw}(\rho) & \mathcal{D}_{zd}(\rho) \\ \mathcal{C}_e(\rho) & \mathcal{D}_{ew}(\rho) & \mathcal{D}_{ed}(\rho) \end{bmatrix} \begin{bmatrix} x_e \\ w \\ d \end{bmatrix}, \tag{5.9}$$

where $x_e = \left[x_G^T, x_\Psi^T\right]^T \in \mathbb{R}^{n_G+n_\Psi}$. Theorem 2 of [102] provided sufficient conditions for bounding the worst-case gain of $\mathcal{F}_u(G,\Delta)$ and is paraphrased next.

Theorem 1 ([102]). *Let G be a quadratically stable LPV system defined by (5.4) and Δ be a bounded, causal operator such that $\mathcal{F}_u(G,\Delta)$ is well-posed. Assume $\Delta \in \mathbf{\Delta}_\Pi$ and consider a factorization $\Pi = \Psi^\sim M \Psi$ with Ψ stable. If Π is a strict PN multiplier and $\exists P = P^T$ such that*

$$\begin{bmatrix} \mathcal{A}^T(\rho)P + P\mathcal{A}(\rho) & \star & \star \\ \mathcal{B}_w^T(\rho)P & 0 & \star \\ \mathcal{B}_d^T(\rho)P & 0 & -\gamma I \end{bmatrix} + \frac{1}{\gamma}\begin{bmatrix} \mathcal{C}_e^T(\rho) \\ \mathcal{D}_{ew}^T(\rho) \\ \mathcal{D}_{ed}^T(\rho) \end{bmatrix}(\star) + \begin{bmatrix} \mathcal{C}_z^T(\rho) \\ \mathcal{D}_{zw}^T(\rho) \\ \mathcal{D}_{zd}^T(\rho) \end{bmatrix} M(\star) < 0 \; \forall \rho \in \mathcal{P}, \tag{5.10}$$

for some $\gamma \in (0,\infty)$ then,

(1) $\lim_{T\to\infty} x_e(T) = 0 \; \forall x_e(0) \in \mathbb{R}^{n_G+n_\Psi}$, $\forall d \in L_2^{n_d}$, and $\forall \rho \in \mathcal{T}$, and

(2) $\sup_{\Delta \in \mathbf{\Delta}_\Pi} \|\mathcal{F}_u(G,\Delta)\| \leq \gamma$.

This theorem provides sufficient conditions for $\mathcal{F}_u(G,\Delta)$ to have bounded worst-case gain. By applying the Schur complement lemma on the second term, inequality (5.10) can be written as a LMI involving the LPV plant G, a Lyapunov matrix P, the gain upper bound γ, and the IQC factorization (Ψ, M). Henceforth, LMIs of the form (5.10) will be referred to using the short form $LMI_{WC}(G,P,\gamma,\Psi,M) < 0$. The subscript WC indicates that it is a LMI associated with a worst-case gain problem and the arguments (G,P,γ,Ψ,M) indicate that the system, Lyapunov matrix, gain bound, and IQC factorization are involved.

Finally, Theorem 1 does not require the matrix P to be positive definite. This is in contrast to Lemma 1 where $P > 0$ was required. The IQCs used in Theorem 1 contain hidden energy. Arguments from game theory can be used to define a new Lyapunov function that includes this hidden energy, and which is indeed positive definite. Section 3 of [102] provides the full proof.

5.2.5 Relation Between Nominal/Uncertain Input-Output Analyses

Section 5.2.3 considered the input-output analysis of *nominal* LPV systems. Section 5.2.4 considered the input-output analysis of *uncertain* LPV systems. This section develops a relation between the nominal and uncertain input-output analyses, which will be used in Section 5.3. One of the main ways in which $LMI_{WC}\left(G, P, \gamma, \Psi, M\right) < 0$ is different from $LMI_{BR}\left(G, P, \gamma\right) < 0$ is its inclusion of the IQC factor $\left(\Psi, M\right)$. Hence, in order to develop a relation between $LMI_{WC}\left(G, P, \gamma, \Psi, M\right) < 0$ and $LMI_{BR}\left(G, P, \gamma\right) < 0$, one must understand the role played by the IQC factor.

At this point, it is easier to work with the J-spectral factor $\left(\hat{\Psi}, J\right)$ of Π. Since $\hat{\Psi}$ is square, its output can be partitioned as $z = \begin{bmatrix} \tilde{v} \\ \tilde{w} \end{bmatrix}$, where $\tilde{v}$ and $\tilde{w}$ have the same sizes as v and w, respectively. In addition, the state-space matrices associated with z (see Equation (5.9)) can be partitioned as:

$$\begin{bmatrix} \mathcal{C}_z\left(\rho\right) & \mathcal{D}_{zw}\left(\rho\right) & \mathcal{D}_{zd}\left(\rho\right) \end{bmatrix} = \begin{bmatrix} \mathcal{C}_{\tilde{v}}\left(\rho\right) & \mathcal{D}_{\tilde{v}w}\left(\rho\right) & \mathcal{D}_{\tilde{v}d}\left(\rho\right) \\ \mathcal{C}_{\tilde{w}}\left(\rho\right) & \mathcal{D}_{\tilde{w}w}\left(\rho\right) & \mathcal{D}_{\tilde{w}d}\left(\rho\right) \end{bmatrix}. \tag{5.11}$$

Using the above matrix partitions, the full form of $LMI_{WC}\left(G, P, \gamma, \hat{\Psi}, J\right) < 0$ is:

$$\begin{bmatrix} \mathcal{A}^T\left(\rho\right)P + \star & \star & \star \\ \mathcal{B}_w^T\left(\rho\right)P & 0 & \star \\ \mathcal{B}_d^T\left(\rho\right)P & 0 & -\gamma I \end{bmatrix} + \frac{1}{\gamma}\begin{bmatrix} \mathcal{C}_e^T\left(\rho\right) \\ \mathcal{D}_{ew}^T\left(\rho\right) \\ \mathcal{D}_{ed}^T\left(\rho\right) \end{bmatrix}\left(\star\right) + \begin{bmatrix} \mathcal{C}_{\tilde{v}}^T\left(\rho\right) & \mathcal{C}_{\tilde{w}}^T\left(\rho\right) \\ \mathcal{D}_{\tilde{v}w}^T\left(\rho\right) & \mathcal{D}_{\tilde{w}w}^T\left(\rho\right) \\ \mathcal{D}_{\tilde{v}d}^T\left(\rho\right) & \mathcal{D}_{\tilde{w}d}^T\left(\rho\right) \end{bmatrix} J\left(\star\right) < 0\ \forall \rho \in \mathcal{P}. \tag{5.12}$$

In this inequality, the third term is the IQC term involving $\hat{\Psi}$ and J. Common LMI manipulation techniques, such as the Schur complement lemma, cannot be applied to the entire IQC term because $J = \left[\begin{smallmatrix} I & 0 \\ 0 & -I \end{smallmatrix}\right]$ is sign indefinite. Rather, it can only be applied to the positive definite sub-block of the IQC term. Hence, an alternative approach is followed wherein inequality (5.12) is simplified. First, note that the $(2, 2)$ block of (5.12) is $\gamma^{-1}\mathcal{D}_{ew}^T\left(\rho\right)\mathcal{D}_{ew}\left(\rho\right) + \mathcal{D}_{\tilde{v}w}^T\left(\rho\right)\mathcal{D}_{\tilde{v}w}\left(\rho\right) - \mathcal{D}_{\tilde{w}w}^T\left(\rho\right)\mathcal{D}_{\tilde{w}w}\left(\rho\right) < 0$. This can be rearranged to show that $\mathcal{D}_{\tilde{w}w}^T(\rho)\,\mathcal{D}_{\tilde{w}w}(\rho) > 0$, i.e. $\mathcal{D}_{\tilde{w}w}(\rho)$ is nonsingular for all $\rho \in \mathcal{P}$. Next, define the following parameter-dependent congruence transformation matrix:

$$T(\rho) \coloneqq \begin{bmatrix} I & 0 & 0 \\ -\mathcal{D}_{\tilde{w}w}^{-1}(\rho)\,\mathcal{C}_{\tilde{w}}(\rho) & \mathcal{D}_{\tilde{w}w}^{-1}(\rho) & -\gamma^{-0.5}\mathcal{D}_{\tilde{w}w}^{-1}(\rho)\,\mathcal{D}_{\tilde{w}d}(\rho) \\ 0 & 0 & \gamma^{-0.5}I \end{bmatrix}. \tag{5.13}$$

Multiplying inequality (5.12) on the left and right by $T^T(\rho)$ and $T(\rho)$, respectively, results in:

$$\begin{bmatrix} \bar{A}^T(\rho)\,P + P\bar{A}(\rho) & \star \\ \bar{B}^T(\rho)\,P & -I \end{bmatrix} + \begin{bmatrix} \bar{C}^T(\rho) \\ \bar{D}^T(\rho) \end{bmatrix}(\star) < 0 \;\forall \rho \in \mathcal{P}, \tag{5.14}$$

where,

$$\begin{aligned}
\bar{A}(\rho) &\coloneqq \mathcal{A}(\rho) - \mathcal{B}_w(\rho)\,\mathcal{D}_{\tilde{w}w}^{-1}(\rho)\,\mathcal{C}_{\tilde{w}}(\rho), && (5.15)\\
\bar{B}(\rho) &\coloneqq \Big[\mathcal{B}_w(\rho)\,\mathcal{D}_{\tilde{w}w}^{-1}(\rho) \quad \gamma^{-0.5}\left(-\mathcal{B}_w(\rho)\,\mathcal{D}_{\tilde{w}w}^{-1}(\rho)\,\mathcal{D}_{\tilde{w}d}(\rho) + \mathcal{B}_d(\rho)\right)\Big], && (5.16)\\
\bar{C}(\rho) &\coloneqq \begin{bmatrix} \mathcal{C}_{\tilde{v}}(\rho) - \mathcal{D}_{\tilde{v}w}(\rho)\,\mathcal{D}_{\tilde{w}w}^{-1}(\rho)\,\mathcal{C}_{\tilde{w}}(\rho) \\ \gamma^{-0.5}\left(\mathcal{C}_e(\rho) - \mathcal{D}_{ew}(\rho)\,\mathcal{D}_{\tilde{w}w}^{-1}(\rho)\,\mathcal{C}_{\tilde{w}}(\rho)\right) \end{bmatrix}, && (5.17)\\
\bar{D}(\rho) &\coloneqq \begin{bmatrix} \mathcal{D}_{\tilde{v}w}(\rho)\mathcal{D}_{\tilde{w}w}^{-1}(\rho) & \gamma^{-0.5}\left(-\mathcal{D}_{\tilde{v}w}(\rho)\mathcal{D}_{\tilde{w}w}^{-1}(\rho)\mathcal{D}_{\tilde{w}d}(\rho)+\mathcal{D}_{\tilde{v}d}(\rho)\right) \\ \gamma^{-0.5}\left(\mathcal{D}_{ew}(\rho)\mathcal{D}_{\tilde{w}w}^{-1}(\rho)\right) & \gamma^{-1}\left(-\mathcal{D}_{ew}(\rho)\mathcal{D}_{\tilde{w}w}^{-1}(\rho)\mathcal{D}_{\tilde{w}d}(\rho)+\mathcal{D}_{ed}(\rho)\right) \end{bmatrix}. && (5.18)
\end{aligned}$$

Note that inequality (5.14) is similar to LMI (5.7) in the Bounded Real Lemma, except that it involves transformed state-space matrices. Consistent with the notation introduced earlier, inequality (5.14) can be shortened to $LMI_{BR}\left(\bar{G}, P, 1\right) < 0$, where $\bar{G} \overset{s}{=} \left[\begin{array}{c|c} \bar{A}(\rho) & \bar{B}(\rho) \\ \hline \bar{C}(\rho) & \bar{D}(\rho) \end{array}\right]$ depends on G, γ, and $\hat{\Psi}$. From Equations (5.15) through (5.18), it can be inferred that the second input and second output of $\bar{G}$ are scaled by $\gamma^{-0.5}$ each. This results in a gain bound of 1 in $LMI_{BR}\left(\bar{G}, P, 1\right) < 0$. The equivalence of $LMI_{WC}\left(G, P, \gamma, \hat{\Psi}, J\right) < 0$ and $LMI_{BR}\left(\bar{G}, P, 1\right) < 0$ was previously reported as Lemma 2 in [93] and is rephrased below.

Lemma 2 ([93]). *Let G be the LPV system defined in (5.4) and $\left(\hat{\Psi}, J\right)$ be a J-spectral factor of Π. $P = P^T$ satisfies $LMI_{WC}\left(G, P, \gamma, \hat{\Psi}, J\right) < 0$ if and only if it satisfies $LMI_{BR}\left(\bar{G}, P, 1\right) < 0$, where the state-space matrices of $\bar{G}$ are defined in Equations (5.15) through (5.18).*

As per Theorem 1, $LMI_{WC}\left(G, P, \gamma, \hat{\Psi}, J\right) < 0$ is a sufficient condition for the worst-case gain of $\mathcal{F}_u\left(G, \Delta\right)$ to be bounded by γ. As per the Bounded Real Lemma, $LMI_{BR}\left(\bar{G}, P, 1\right) < 0$ is a sufficient condition for $\left\|\bar{G}\right\|$ to be bounded by 1. Lemma 2 states that the sufficient condition for bounding the worst-case gain of $\mathcal{F}_u\left(G, \Delta\right)$ by γ is equivalent to the sufficient condition for bounding the induced L_2 norm of $\bar{G}$ by 1. Finally, note that Lemma 2 only requires P to be symmetric and makes no assumptions on its sign definiteness. However, since a J-spectral factorization of Π is used, any $P = P^T$ that satisfies $LMI_{WC}\left(G, P, \gamma, \hat{\Psi}, J\right) < 0$ is also necessarily positive definite. (This can be proved using arguments from game theory. See Section 3 of [102] for details.) Since the same P is used in $LMI_{BR}\left(\bar{G}, P, 1\right) < 0$, the quadratic stability of $\bar{G}$ is guaranteed by Lemma 1.

5.3 Dual Input-Output Analysis

A convex formulation for the disturbance feedforward problem requires certain duality results. The reader who is only interested in the convex synthesis results may skip ahead to Section 5.4. The duality results presented in this section consist of different components. Figure 5.3 provides an overview of these components and the relationships between them. The top row depicts the relation between the nominal and uncertain input-output analyses (Section 5.2.5). The upper-left figure shows the interconnection of G and $\hat{\Psi}$ used in the worst-case input-output analysis. The upper-right figure shows the plant $\bar{G}$ used in the equivalent nominal input-output analysis. Lemma 2 demonstrates the equivalence of the LMI conditions for these two analyses.

While the state-space matrices of $\bar{G}$ are listed in Equations (5.15) through (5.18), additional insight can be gained by understanding the composition of $\bar{G}$. In particular, it can be shown through linear algebra that $\bar{G}$ is the LFT of a scaled version of G and a new system $\hat{\Psi}^\dagger$, which is obtained after inverting the $\tilde{w}/w$ channels of $\hat{\Psi}$ and reordering the new inputs. Consequently, $\hat{\Psi}^\dagger$ has inputs $(\tilde{w}, v)$ and outputs $(\tilde{v}, w)$, as shown in Figure 5.3. If $\hat{\Psi}$ is partitioned conformably with the sizes of v and w as $\left[\begin{smallmatrix} \Psi_{11} & \Psi_{12} \\ \Psi_{21} & \Psi_{22} \end{smallmatrix}\right]$, then the steps described above result in $\hat{\Psi}^\dagger := \left[\begin{smallmatrix} \Psi_{12}\Psi_{22}^{-1} & \Psi_{11}-\Psi_{12}\Psi_{22}^{-1}\Psi_{21} \\ \Psi_{22}^{-1} & -\Psi_{22}^{-1}\Psi_{21} \end{smallmatrix}\right]$. Note that $\hat{\Psi}^\dagger$ is the Potapov-Ginsburg transform[1] of $\hat{\Psi}$ [114]. The Potapov-Ginsburg transform

[1] See Equation (2.1) on page 454 of [114] for the definition of the Potapov-Ginsburg transform. The

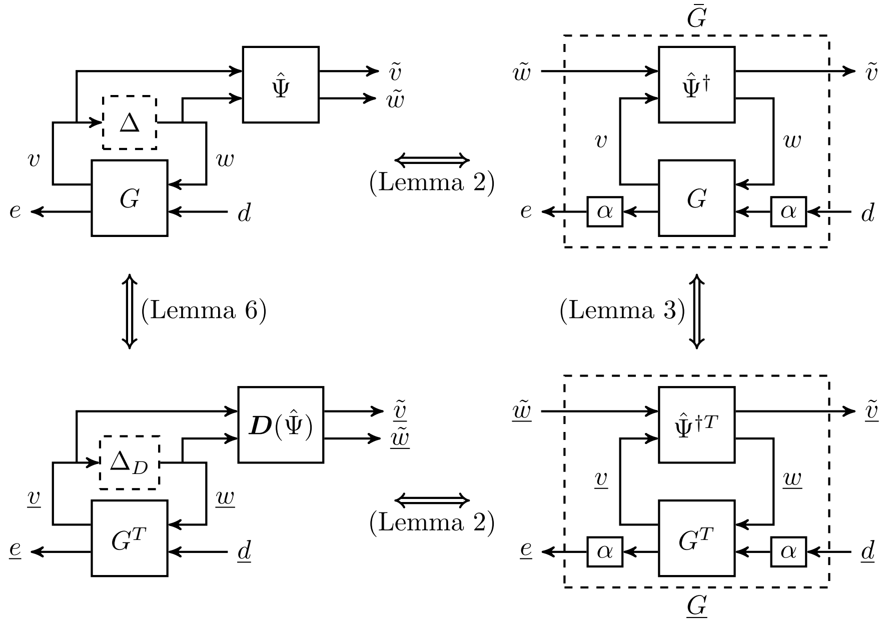


Figure 5.3: An overview of the systems and relations covered by Section 5.3. The upper-left and lower-left corners show the systems involved in the primal and dual worst-case gain analyses. The upper-right and lower-right corners show the systems involved in the corresponding nominal gain analyses with $\alpha = \gamma^{-0.5}$.

exists because Π is restricted to be a strict PN multiplier. In particular, note that $\Pi_{22} = \Psi_{12}^{\sim}\Psi_{12} - \Psi_{22}^{\sim}\Psi_{22} < 0$ on the extended imaginary axis. This can be rearranged to show that $\Psi_{22}^{\sim}\Psi_{22} > 0$ on the extended imaginary axis. Hence Ψ_{22} has a nonsingular feed-through matrix and has an inverse.

One interpretation of the relation between the upper-left and upper-right figures is through the equivalence of their respective LMIs. This is what Lemma 2 does. Another interpretation is through the relations between the signals. Recall that the input-output analysis of uncertain LPV systems (Theorem 1) replaces the precise behavior of Δ with an IQC. In this formulation (shown in the upper-left figure), w is treated as an input to the system that implicitly constrains the signals $(\tilde{v}, \tilde{w})$. The inversion of the $\tilde{w}/w$ channels in the Potapov-Ginsburg transform reveals that w is explicitly defined by the signals $(\tilde{w}, v)$ and the system $\hat{\Psi}^{\dagger}$. As a result the nominal analysis problem

precise form of $\hat{\Psi}^{\dagger}$ given in this chapter is slightly different because of a sign change and a reordering of the inputs.

(shown in the upper-right figure) has inputs $(\tilde{w}, d)$ and outputs $(\tilde{v}, e)$. Consequently, $LMI_{BR}\left(\bar{G}, P, 1\right) < 0$ is a sufficient condition for the L_2 gain from $(\tilde{w}, d)$ to $(\tilde{v}, e)$ to be bounded by 1. Further, the two problems may also be related through their respective dissipation inequalities [93].

The remaining systems and relations that are shown in Figure 5.3 are described in the following subsections. A notion of duality for nominal LPV systems is developed in Section 5.3.1 and is used to relate $\bar{G}$ (shown in the upper-right figure) with its dual (shown in the lower-right figure). This relation is formally presented as Lemma 3. Next, a notion of duality for IQCs is developed in Section 5.3.2. Dual LPV systems and dual IQCs are used together to develop a notion of duality for uncertain LPV systems in Section 5.3.4. Section 5.3 culminates in Lemma 6, which relates the uncertain LPV system $\mathcal{F}_u(G, \Delta)$ (shown in the upper-left figure) with its dual (shown in the lower-left figure). The following subsections will make occasional reference to Figure 5.3 and define the remaining notation.

5.3.1 Dual LPV Systems

The concept of duality is well developed for LTI systems, e.g. duality between the concepts of controllability and observability [90]. For LPV systems, duality is defined as follows.

Definition 4. *If* $G \overset{s}{=} \left[\begin{array}{c|c} A_G(\rho) & B_G(\rho) \\ \hline C_G(\rho) & D_G(\rho) \end{array}\right]$ *is a (primal) LPV system then* $G^T \overset{s}{=} \left[\begin{array}{c|c} A_G^T(\rho) & C_G^T(\rho) \\ \hline B_G^T(\rho) & D_G^T(\rho) \end{array}\right]$ *is the corresponding dual system.*

Lemma 1 proves that the existence of $P > 0$ such that $LMI_{BR}(G, P, \gamma) < 0$ is sufficient for $\|G\| < \gamma$. In a similar manner, the existence of $Q > 0$ such that $LMI_{BR}\left(G^T, Q, \gamma\right) < 0$ is sufficient for $\left\|G^T\right\| < \gamma$. The next lemma relates the two sets of sufficient conditions.

Lemma 3. *$LMI_{BR}(G, P, \gamma) < 0$ for $P > 0$ if and only if $LMI_{BR}\left(G^T, Q, \gamma\right) < 0$ for $Q := P^{-1} > 0$.*

Proof. It follows from linear algebra that $P > 0$ if and only if $Q := P^{-1} > 0$. Apply the

Schur complement lemma to show that $LMI_{BR}(G, P, \gamma) < 0$ is equivalent to

$$\begin{bmatrix} A_G^T(\rho) P + P A_G(\rho) & \star & \star \\ B_G^T(\rho) P & -\gamma I & \star \\ C_G(\rho) & D_G(\rho) & -\gamma I \end{bmatrix} < 0 \; \forall \rho \in \mathcal{P}. \tag{5.19}$$

Next, apply the congruence transformation diag $\left(P^{-1}, I, I\right)$ on the left and right of LMI (5.19). Finally, apply the Schur complement lemma to the $(2, 2)$ block of the resulting LMI to show that it is equivalent to $LMI_{BR}\left(G^T, Q, \gamma\right) < 0$. □

Therefore, Lemma 3 effectively shows that the sufficient conditions for bounding the induced L_2 norms of the primal and dual forms of a nominal LPV system are equivalent. With reference to Figure 5.3, the upper-right figure shows the *primal* nominal LPV system $\bar{G} := \mathcal{F}_u\left(\hat{G}, \hat{\Psi}^{\dagger}\right)$, where $\hat{G}$ is the scaled version of G shown in Figure 5.3. The lower-right figure shows the LPV system $\underline{G} := \mathcal{F}_u\left(\hat{G}^T, \hat{\Psi}^{\dagger T}\right)$. It can be verified using linear algebra that $\underline{G} = \bar{G}^T$. (See Section 12.2 in [90] for a discussion on algebraic duality and its application to LFT interconnections.) In other words, $\bar{G}$ and $\underline{G}$ are dual systems. Hence, Lemma 3 is used to relate their respective Bounded Real LMIs.

5.3.2 Dual IQCs

The notion of duality for uncertain LPV systems requires a specific notion of duality for IQCs. Dual IQCs were previously introduced in [96]. These dual IQCs were defined in the frequency-domain for the stability analysis and feedforward control of LTI systems [96]. The results in [96] are briefly summarized in this subsection as this will ultimately lead to a related time-domain definition for dual IQCs. To begin, consider the uncertain system shown in Figure 5.1, with G assumed to be LTI. The main IQC theorem in [99] roughly states that the following frequency-domain inequality is a sufficient condition for the stability of $\mathcal{F}_u(G, \Delta)$ for every $\Delta \in \mathbf{\Delta}_{\Pi}$:

$$\begin{bmatrix} G(j\omega) \\ I \end{bmatrix}^* \Pi(j\omega) \begin{bmatrix} G(j\omega) \\ I \end{bmatrix} < 0 \; \forall \omega \in \mathbb{R} \cup \{\infty\}. \tag{5.20}$$

This result can be interpreted as a test of strict separation between the graph of Δ and the inverse graph of G [109,115]. Inequality (5.20) is expressed in terms of G and Π and is denoted as the *primal* frequency-domain inequality. It is shown in Section 2.1 of [96] that inequality (5.20) is equivalent to

$$\begin{bmatrix} I \\ -G\left(j\omega\right)^{*} \end{bmatrix}^{*} \Pi\left(j\omega\right)^{-1} \begin{bmatrix} I \\ -G\left(j\omega\right)^{*} \end{bmatrix} > 0 \; \forall \omega \in \mathbb{R} \cup \left\{\infty\right\}. \tag{5.21}$$

Inequality (5.21) is expressed in terms of $G\left(j\omega\right)^{*}$ and Π^{-1} and is denoted as the *dual* frequency-domain inequality [96]. Moreover, the multiplier Π^{-1} is denoted as the *dual* IQC multiplier in [96]. Note that inequality (5.21) involves $G\left(j\omega\right)^{*}$, but Definition 4 defines the dual of G as G^{T}. In order to bring inequality (5.21) into a form that uses G^{T}, the following equivalent definition of the dual IQC multiplier is used in this chapter.

Definition 5. *Given the strict PN primal IQC multiplier* $\Pi = \Pi^{\sim} \in \mathbb{RL}_{\infty}^{(n_v+n_w)\times(n_v+n_w)}$, *the dual IQC multiplier is denoted by* $\boldsymbol{D}\left(\Pi\right) \in \mathbb{RL}_{\infty}^{(n_w+n_v)\times(n_w+n_v)}$ *and is defined as:*

$$\boldsymbol{D}\left(\Pi\right) := \begin{bmatrix} 0 & -I_{n_w} \\ I_{n_v} & 0 \end{bmatrix} \Pi^{-T} \begin{bmatrix} 0 & -I_{n_v} \\ I_{n_w} & 0 \end{bmatrix}. \tag{5.22}$$

In Equation (5.22), Π^{-T} is the transpose of the matrix inverse of Π evaluated pointwise across frequency. Since Π is assumed to be a strict PN multiplier, Definition 1 implies that $\Pi_{11}\left(j\omega\right) > 0$ and $\Pi_{22}\left(j\omega\right) < 0 \; \forall \omega \in \mathbb{R} \cup \left\{\infty\right\}$. Hence $\Pi^{-1}\left(j\omega\right)$ and, by extension, $\boldsymbol{D}\left(\Pi\left(j\omega\right)\right)$ exist $\forall \omega \in \mathbb{R} \cup \left\{\infty\right\}$ (see Lemma 4 in [103]). Using Definition 5, it is verified that (5.21) is equivalent to

$$\begin{bmatrix} G\left(j\omega\right)^{T} \\ I \end{bmatrix}^{*} \boldsymbol{D}\left(\Pi\left(j\omega\right)\right) \begin{bmatrix} G\left(j\omega\right)^{T} \\ I \end{bmatrix} < 0 \; \forall \omega \in \mathbb{R} \cup \left\{\infty\right\}. \tag{5.23}$$

G^{T} in inequality (5.23) is the dual of G in the sense of Definition 4 (albeit with no parameter dependence). Note that the form of inequality (5.23) is similar to that of (5.20), except it involves the dual LTI plant G^{T} and the dual IQC multiplier $\boldsymbol{D}\left(\Pi\right)$.

It is worth noting some subtle points about the dual IQC multiplier. In the standard IQC analysis problem, once the perturbation Δ is specified, the multiplier Π is chosen from a library [99] such that $\Delta \in \boldsymbol{\Delta}_\Pi$. The dual multiplier $\boldsymbol{D}(\Pi)$ is different because, rather than being chosen from a library, it is derived from the primal multiplier Π using Definition 5. As such, a set of operators $\boldsymbol{\Delta}_{\boldsymbol{D}(\Pi)}$ satisfy the IQC defined by $\boldsymbol{D}(\Pi)$. This set is defined as $\boldsymbol{\Delta}_{\boldsymbol{D}(\Pi)} \coloneqq \{\Delta_D : \Delta_D \in \mathrm{IQC}(\boldsymbol{D}(\Pi))\}$. Just as $\boldsymbol{D}(\Pi)$ is the dual of Π, $\boldsymbol{\Delta}_{\boldsymbol{D}(\Pi)}$ is the dual of $\boldsymbol{\Delta}_\Pi$. If $\Delta \in \boldsymbol{\Delta}_\Pi$ is a linear operator, then it has a well-defined adjoint operator Δ_D. However, if Δ is nonlinear then an adjoint need not exist. The notion of duality used in this chapter handles such situations. In particular, the set $\boldsymbol{\Delta}_\Pi$ yields a well defined dual set $\boldsymbol{\Delta}_{\boldsymbol{D}(\Pi)}$ using Definition 5.

To summarize, the main result in [99] is that the primal frequency domain inequality (5.20) is sufficient for the stability of $\mathcal{F}_u(G, \Delta)$ for $\Delta \in \boldsymbol{\Delta}_\Pi$. This is equivalent to stating that the dual frequency domain inequality (5.23) is sufficient for the stability of $\mathcal{F}_u\left(G^T, \Delta_D\right)$ for $\Delta_D \in \boldsymbol{\Delta}_{\boldsymbol{D}(\Pi)}$. Hence, Definition 5 allows $(G, \boldsymbol{\Delta}_\Pi)$ to be related to $\left(G^T, \boldsymbol{\Delta}_{\boldsymbol{D}(\Pi)}\right)$. This result enables certain problems, e.g. feedforward synthesis, to be convexified by converting from primal to dual form [96]. These results are known for LTI systems and LFT-based LPV systems.

In Section 5.3.4, these results are extended for uncertain, gridded LPV systems. Since these systems are time-varying and do not have a valid frequency response interpretation, a time-domain definition for dual IQCs is considered. In particular, dual J-spectral factorizations are considered. Let $\left(\hat{\Psi}, J\right)$ be a J-spectral factorization of Π. Using Definition 5, the dual IQC multiplier is expressed as

$$\boldsymbol{D}(\Pi) = \begin{bmatrix} 0 & -I_{n_w} \\ I_{n_v} & 0 \end{bmatrix} \hat{\Psi}^{-T\sim} J^{-1} \hat{\Psi}^{-T} \begin{bmatrix} 0 & -I_{n_v} \\ I_{n_w} & 0 \end{bmatrix}. \tag{5.24}$$

With a few more steps, it follows that

$$\boldsymbol{D}(\Pi) = \boldsymbol{D}\left(\hat{\Psi}\right)^{\sim} J \boldsymbol{D}\left(\hat{\Psi}\right), \text{ where } \boldsymbol{D}\left(\hat{\Psi}\right) \coloneqq \begin{bmatrix} 0 & -I_{n_w} \\ I_{n_v} & 0 \end{bmatrix} \hat{\Psi}^{-T} \begin{bmatrix} 0 & -I_{n_v} \\ I_{n_w} & 0 \end{bmatrix}. \tag{5.25}$$

Further, $\hat{\Psi}, \hat{\Psi}^{-1} \in \mathbb{RH}_\infty^{(n_v+n_w)\times(n_v+n_w)}$ if and only if the duals $\boldsymbol{D}\left(\hat{\Psi}\right), \boldsymbol{D}\left(\hat{\Psi}\right)^{-1} \in \mathbb{RH}_\infty^{(n_w+n_v)\times(n_w+n_v)}$. Therefore, $\left(\boldsymbol{D}\left(\hat{\Psi}\right), J\right)$ is a J-spectral factorization of $\boldsymbol{D}(\Pi)$. With reference to Figure 5.3, the upper-left figure shows $\mathcal{F}_u(G, \Delta)$ where $\Delta \in \boldsymbol{\Delta}_\Pi$ and $\left(\hat{\Psi}, J\right)$ is a J-spectral factorization of Π. The lower-left figure shows $\mathcal{F}_u\left(G^T, \Delta_D\right)$

where $\Delta_D \in \boldsymbol{\Delta}_{\boldsymbol{D}(\Pi)}$ and $\left(\boldsymbol{D}\left(\hat{\Psi}\right), J\right)$ is a J-spectral factorization of $\boldsymbol{D}(\Pi)$. Section 5.3.4 will develop a relation between these two systems.

5.3.3 Technical Results

Before extending duality to uncertain, gridded LPV systems, some technical results are presented. First, consider the sufficient condition $LMI_{WC}(G, P, \gamma, \Psi, M) < 0$ that was presented in Theorem 1 for bounding the worst-case gain of $\mathcal{F}_u(G, \Delta)$ over the set $\boldsymbol{\Delta}_\Pi$. Since $LMI_{WC}(G, P, \gamma, \Psi, M) < 0$ can be composed using any stable factorization of Π, it is important to understand how its feasibility depends on the factorization. The next lemma relates the state-space realizations of two stable factorizations of Π.

Lemma 4. *Let a frequency-domain IQC multiplier have the following two factorizations,*

$$\Pi(s) = \Pi^\sim(s) = \Psi_1^\sim(s) M_1 \Psi_1(s) = \Psi_2^\sim(s) M_2 \Psi_2(s), \tag{5.26}$$

where $\Psi_1(s) \overset{s}{=} \left[\begin{array}{c|c} A_1 & B_1 \\ \hline C_1 & D_1 \end{array}\right]$ *and* $\Psi_2(s) \overset{s}{=} \left[\begin{array}{c|c} A_2 & B_2 \\ \hline C_2 & D_2 \end{array}\right]$ *are stable, minimal realizations with state dimension n. Define* $\begin{bmatrix} Q_i & S_i \\ S_i^T & R_i \end{bmatrix} = \begin{bmatrix} C_i^T \\ D_i^T \end{bmatrix} M_i \begin{bmatrix} C_i & D_i \end{bmatrix}$ *for* $i = 1, 2$. *Then* $\exists T_1 \in \mathbb{R}^{n \times n}$ *such that:*

(1) $A_2 = T_1 A_1 T_1^{-1}$,

(2) $B_2 = T_1 B_1$, *and*

(3) $\begin{bmatrix} Q_2 & S_2 \\ S_2^T & R_2 \end{bmatrix} = \begin{bmatrix} T_1^{-T} & 0 \\ 0 & I \end{bmatrix} \left(\begin{bmatrix} Q_1 & S_1 \\ S_1^T & R_1 \end{bmatrix} - \begin{bmatrix} A_1^T \bar{X} + \bar{X} A_1 & \bar{X} B_1 \\ B_1^T \bar{X} & 0 \end{bmatrix} \right) \begin{bmatrix} T_1^{-1} & 0 \\ 0 & I \end{bmatrix}$, *where* $\bar{X} = \bar{X}^T$ *is the unique solution to the Lyapunov Equation* $A_1^T \bar{X} + \bar{X} A_1 = Q_1 - T_1^T Q_2 T_1$.

Proof. The proof mainly relies on standard facts regarding Lyapunov equalities. See Appendix A.1. □

The next lemma relates the feasibility of two worst-case gain LMIs using Lemma 4.

Lemma 5. *Consider two factorizations* (Ψ_1, M_1) *and* (Ψ_2, M_2) *of* Π *such that* Ψ_1 *and* Ψ_2 *are stable and have minimal state-space realizations. There exists* $P_1 = P_1^T$ *satisfying* $LMI_{WC}(G, P_1, \gamma, \Psi_1, M_1) < 0$ *if and only if there exists* $P_2 = P_2^T$ *satisfying* $LMI_{WC}(G, P_2, \gamma, \Psi_2, M_2) < 0$.

Proof. By the assumption made, the two factorizations of Π are stable and have minimal state-space realizations. Hence, there exist T_1 and $\bar{X}$ satisfying conclusions (1)-(3) in Lemma 4. Next, define $T_\beta := \operatorname{diag}\left(I, T_1^{-1}, I, I\right)$. To prove necessity, assume that there exists $P_2 = P_2^T$ satisfying $LMI_{WC}\left(G, P_2, \gamma, \Psi_2, M_2\right) < 0$. Then, multiply $LMI_{WC}\left(G, P_2, \gamma, \Psi_2, M_2\right) < 0$ on the left and right by T_β^{-T} and T_β^{-1}, respectively. Finally, use statements (1) and (2) of Lemma 4 to show that P_1 satisfies $LMI_{WC}\left(G, P_1, \gamma, \Psi_1, M_1\right) < 0$, where P_1 and P_2 are related as:

$$P_1 = \begin{bmatrix} I_{n_G} & 0 \\ 0 & T_1^T \end{bmatrix} P_2 \begin{bmatrix} I_{n_G} & 0 \\ 0 & T_1 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0 & \bar{X} \end{bmatrix}. \tag{5.27}$$

To prove sufficiency, reverse the algebraic steps. □

Lemma 5 essentially proves that the feasibility of $LMI_{WC}\left(G, P, \gamma, \Psi, M\right) < 0$ is independent of the factorization (Ψ, M), as long as Ψ is stable and has a minimal state-space realization. In other words, the feasibility of $LMI_{WC}\left(G, P, \gamma, \Psi, M\right) < 0$ only depends on G and Π.

5.3.4 Dual Uncertain LPV Systems

This subsection brings together all the concepts discussed thus far, including the relation between the nominal and uncertain input-output analyses (Section 5.2.5), dual LPV systems (Section 5.3.1), dual IQCs (Section 5.3.2), and technical results (Section 5.3.3).

To provide context, consider the roles played by Lemmas 1 and 3 in the input-output analysis of nominal LPV plants. The Bounded Real Lemma, as stated in Lemma 1, provides a sufficient LMI condition to bound the induced L_2 norm of a nominal LPV system. Lemma 3 demonstrates an equivalence between the primal and dual forms of this LMI condition. This subsection derives a similar set of results for uncertain LPV systems. Theorem 1 already establishes sufficient conditions to bound the worst-case gain of an uncertain LPV system. Hence, the missing piece in the puzzle is a lemma for uncertain LPV systems that demonstrates an equivalence between the primal and dual forms, analogous to the result in Lemma 3 for nominal LPV systems. The next lemma proves that the primal and dual forms of the sufficient conditions presented in Theorem 1 are equivalent.

Lemma 6. *Given G and Π, the following statements hold.*

(1) G is quadratically stable if and only if G^T is quadratically stable.

(2) Π is a strict PN multiplier if and only if $\boldsymbol{D}(\Pi)$ is a strict PN multiplier.

(3) Let (Ψ, M) be any stable factorization of Π and (Γ, N) be any stable factorization of $\boldsymbol{D}(\Pi)$. Then $\exists P = P^T$ satisfying $LMI_{WC}(G, P, \gamma, \Psi, M) < 0$ if and only if $\exists Q = Q^T$ satisfying $LMI_{WC}\left(G^T, Q, \gamma, \Gamma, N\right) < 0$.

Proof. Statement (1) follows from Lemmas 1 and 3. Statement (2) is proved as follows. For sufficiency, assume that Π is a strict PN multiplier. First, from Definition 1, $\Pi_{11}(j\omega) > 0$ and $\Pi_{22}(j\omega) < 0$ $\forall\omega \in \mathbb{R} \cup \{\infty\}$. This implies that $\Pi^{-1}(j\omega)$ exists $\forall\omega \in \mathbb{R} \cup \{\infty\}$ and can be partitioned as $\left[\begin{smallmatrix} W_{11}(j\omega) & W_{12}(j\omega) \\ W_{12}^{\sim}(j\omega) & W_{22}(jw) \end{smallmatrix}\right]$. In the next few steps, the argument $(j\omega)$ is dropped for brevity. From the matrix inversion lemma, it follows that $W_{11} = \left(\Pi_{11} - \Pi_{12}\Pi_{22}^{-1}\Pi_{21}\right)^{-1}$ and $W_{22} = \left(\Pi_{22} - \Pi_{21}\Pi_{11}^{-1}\Pi_{12}\right)^{-1}$. Note that $\Pi_{11} > 0$ and $\Pi_{22} < 0$ together imply $W_{11} > 0$ and $W_{22} < 0$. Next, partition $\boldsymbol{D}(\Pi)$ as $\left[\begin{smallmatrix} \Pi_{D11} & \Pi_{D12} \\ \Pi_{D12}^{\sim} & \Pi_{D22} \end{smallmatrix}\right]$, where $\Pi_{D11} \in \mathbb{RL}_{\infty}^{n_w \times n_w}$ and $\Pi_{D22} \in \mathbb{RL}_{\infty}^{n_v \times n_v}$. Finally, using Definition 5, it can be shown that $\Pi_{D11} = -W_{22}^T > 0$ and $\Pi_{D22} = -W_{11}^T < 0$. Hence, $\boldsymbol{D}(\Pi)$ is a strict PN multiplier. For necessity, note that $\boldsymbol{D}(\boldsymbol{D}(\Pi)) = \Pi$ and use similar arguments.

Statement (3) is proved by invoking multiple previous lemmas. First, consider the statement: $\exists P = P^T$ satisfying $LMI_{WC}(G, P, \gamma, \Psi, M) < 0$ for some stable factorization (Ψ, M) of Π. Let $\left(\hat{\Psi}, J\right)$ denote a J-spectral factorization of the multiplier Π. Since $\hat{\Psi}$ is stable by definition, it is inferred from Lemma 5 that $\exists \hat{P} = \hat{P}^T$ satisfying $LMI_{WC}\left(G, \hat{P}, \gamma, \hat{\Psi}, J\right) < 0$. Further, from Lemma 2, $\hat{P} = \hat{P}^T$ satisfies $LMI_{WC}\left(G, \hat{P}, \gamma, \hat{\Psi}, J\right) < 0$ if and only if it satisfies $LMI_{BR}\left(\bar{G}, \hat{P}, 1\right) < 0$, where $\bar{G}$ depends on G, γ, and $\hat{\Psi}$ through Equations (5.15) through (5.18).

Next, consider the statement: $\exists Q = Q^T$ satisfying $LMI_{WC}\left(G^T, Q, \gamma, \Gamma, N\right) < 0$ for some stable factorization (Γ, N) of $\boldsymbol{D}(\Pi)$. Let $\left(\boldsymbol{D}\left(\hat{\Psi}\right), J\right)$ denote a J-spectral factorization of $\boldsymbol{D}(\Pi)$. Since $\boldsymbol{D}\left(\hat{\Psi}\right)$ is stable by definition, it is inferred from Lemma 5 that $\exists \hat{Q} = \hat{Q}^T$ satisfying $LMI_{WC}\left(G^T, \hat{Q}, \gamma, \boldsymbol{D}\left(\hat{\Psi}\right), J\right) < 0$. Further, from Lemma 2, $\hat{Q} = \hat{Q}^T$ satisfies $LMI_{WC}\left(G^T, \hat{Q}, \gamma, \boldsymbol{D}\left(\hat{\Psi}\right), J\right) < 0$ if and only if it satisfies $LMI_{BR}\left(\underline{G}, \hat{Q}, 1\right) < 0$, where $\underline{G}$ depends on G^T, γ, and $\boldsymbol{D}\left(\hat{\Psi}\right)$ through equations that are similar to Equations (5.15) through (5.18).

It can be verified, with a significant amount of algebra, that $\underline{G} = \bar{G}^T$, i.e. $\bar{G}$ and

$\underline{G}$ are dual systems. Finally, from Lemma 3, $\hat{P} = \hat{P}^T$ satisfies $LMI_{BR}\left(\bar{G}, \hat{P}, 1\right) < 0$ if and only if $\hat{Q} := \hat{P}^{-1}$ satisfies $LMI_{BR}\left(\underline{G}, \hat{Q}, 1\right) < 0$. □

Lemma 6 can be better understood in the context of two related worst-case gain problems (see Figure 5.3). The upper-left figure shows the primal problem, which involves bounding the worst-case gain of $\mathcal{F}_u(G, \Delta)$ over the set $\mathbf{\Delta}_\Pi$. The lower-left figure shows the dual problem, which involves bounding the worst-case gain of $\mathcal{F}_u\left(G^T, \Delta_D\right)$ over the set $\mathbf{\Delta}_{\boldsymbol{D}(\Pi)}$. Both problems have separate (but similar) sets of sufficient conditions (Theorem 1) for bounding their respective worst-case gains. Lemma 6 essentially states that the two sets of sufficient conditions are equivalent. Statement (1) establishes equivalence between the primal and dual nominal LPV systems, in the sense of quadratic stability. Statement (2) establishes equivalence between the primal and dual IQC multipliers, in the sense of the strict PN property. Statement (3) establishes equivalence between the primal and dual worst-case gain LMI conditions.

Finally, note that the duality result for uncertain LPV systems (Lemma 6) requires a more complicated proof than the corresponding result for nominal LPV systems (Lemma 3). Lemma 3 is straightforward because the Schur complement lemma can be applied to blocks involving I or $-I$ in $LMI_{BR}(G, P, \gamma) < 0$. However, a similar procedure cannot be followed for uncertain LPV systems because of the presence of the additional IQC term in $LMI_{WC}(G, P, \gamma, \Psi, M) < 0$, as explained previously in Section 5.2.5. To circumvent this roadblock, Lemma 2 is invoked to convert the primal and dual worst-case gain LMI conditions to their respective Bounded Real LMI conditions. This is shown along the top and bottom rows of Figure 5.3. The two Bounded Real LMI conditions are then shown to be duals of each other using Lemma 3 (shown along the right-hand side column).

5.4 Convex Synthesis for Uncertain LPV Systems

5.4.1 Output Estimation

While the output estimation problem was previously considered in [86], the derivation of the synthesis conditions provided in this section is more rigorous in three specific ways. First, the state-space matrices of the estimator are completely eliminated from the LMI conditions given in the synthesis theorem. Second, a matrix dilation lemma is used to complete the sign indefinite Lyapunov matrix. Third, an explicit method is provided to reconstruct the state-space matrices of the estimator.

The output estimation problem is formulated using the interconnection shown in Figure 5.4. H is a nominal LPV plant with parameters $\rho \in \mathcal{P}$, states $x_H \in \mathbb{R}^{n_H}$, disturbance inputs $d \in \mathbb{R}^{n_d}$, measurable outputs $y \in \mathbb{R}^{n_y}$, and unmeasurable outputs $q \in \mathbb{R}^{n_q}$. The LPV plant H is connected with an uncertainty Δ via signals $v \in \mathbb{R}^{n_v}$ and $w \in \mathbb{R}^{n_w}$. This creates an uncertain LPV system $\mathcal{F}_u\left(H, \Delta\right)$ from the input disturbance d to the outputs y and q. The problem is to synthesize a estimator F that uses the measurements y to generate an estimate of q. Let $\hat{q}$ denote the estimate of q and $e := \hat{q} - q$ denote the estimation error. The synthesis objective is to bound the worst-case induced L_2 norm from d to e over the set of uncertainties that satisfy the IQC defined by Π.

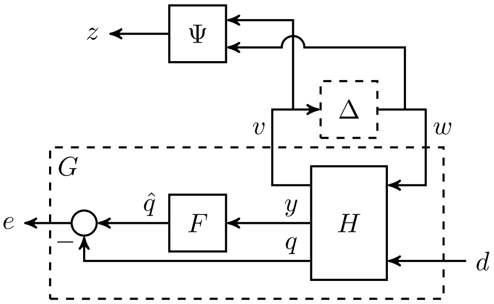


Figure 5.4: The output estimation problem.

In addition to the LFT of H and Δ, the IQC filter Ψ is appended such that it is driven by signals v and w, and produces signal z. The interconnection of H and Ψ has the state-space representation

$$\begin{bmatrix} \dot{x} \\ z \\ y \\ q \end{bmatrix} = \begin{bmatrix} A\left(\rho\right) & B_1\left(\rho\right) & B_2\left(\rho\right) \\ C_1\left(\rho\right) & D_{11}\left(\rho\right) & D_{12}\left(\rho\right) \\ C_2\left(\rho\right) & D_{21}\left(\rho\right) & D_{22}\left(\rho\right) \\ C_3\left(\rho\right) & D_{31}\left(\rho\right) & D_{32}\left(\rho\right) \end{bmatrix} \begin{bmatrix} x \\ w \\ d \end{bmatrix}, \tag{5.28}$$

where $x = \left[x_H^T, x_\Psi^T\right]^T \in \mathbb{R}^{n_H + n_\Psi}$ are the combined states of H and Ψ. It is assumed that $D_{22}\left(\rho\right) \in \mathbb{R}^{n_y \times n_d}$ has full row rank $\forall \rho \in \mathcal{P}$. This ensures that all components of the measurement y are affected by some component of the disturbance d.

The estimator F to be synthesized has the state-space representation:

$$\begin{bmatrix} \dot{x}_F \\ \hat{q} \end{bmatrix} = \begin{bmatrix} A_F(\rho) & B_F(\rho) \\ C_F(\rho) & D_F(\rho) \end{bmatrix} \begin{bmatrix} x_F \\ y \end{bmatrix}, \tag{5.29}$$

where $x_F \in \mathbb{R}^{n_F}$ are the estimator states, y is the input to the estimator, and $\hat{q}$ is the output from the estimator. As shown by the large dashed box in Figure 5.4, the closed-loop formed by the interconnection of H and F is denoted by G, with states $x_G = \left[x_H^T,\, x_F^T\right]^T$. In the remainder of this section, the notation $G(H,F)$ will be used in some cases to make explicit the dependence of G on H and F. Theorem 1 provides conditions to bound the worst-case gain of $\mathcal{F}_u(G(H,F),\Delta)$ over the set $\mathbf{\Delta}_\Pi$. The objective is to synthesize F which minimizes this bound.

To formulate the synthesis theorem, consider the extended LPV system formed by the interconnection of $G(H,F)$ and Ψ. This extended system has the state-space realization:

$$\begin{bmatrix} \dot{x}_e \\ z \\ e \end{bmatrix} = \begin{bmatrix} \mathcal{A}(\rho) & \mathcal{B}_w(\rho) & \mathcal{B}_d(\rho) \\ \mathcal{C}_z(\rho) & \mathcal{D}_{zw}(\rho) & \mathcal{D}_{zd}(\rho) \\ \mathcal{C}_e(\rho) & \mathcal{D}_{ew}(\rho) & \mathcal{D}_{ed}(\rho) \end{bmatrix} \begin{bmatrix} x_e \\ w \\ d \end{bmatrix}, \tag{5.30}$$

where $x_e = \left[x_H^T, x_\Psi^T, x_F^T\right]^T \in \mathbb{R}^{n_H+n_\Psi+n_F}$ are the combined states of H, Ψ, and F. These state-space matrices are expressed in terms of the matrices appearing in Equations (5.28) and (5.29) as:

$$\begin{bmatrix} \mathcal{A} & \mathcal{B}_w & \mathcal{B}_d \\ \mathcal{C}_z & \mathcal{D}_{zw} & \mathcal{D}_{zd} \\ \mathcal{C}_e & \mathcal{D}_{ew} & \mathcal{D}_{ed} \end{bmatrix} = \left[\begin{array}{cc:c:c} A & 0 & B_1 & B_2 \\ 0 & 0 & 0 & 0 \\ \hdashline C_1 & 0 & D_{11} & D_{12} \\ \hdashline -C_3 & 0 & -D_{31} & -D_{32} \end{array}\right] + \left[\begin{array}{cc} 0 & 0 \\ I & 0 \\ \hdashline 0 & 0 \\ \hdashline 0 & I \end{array}\right] \begin{bmatrix} A_F & B_F \\ C_F & D_F \end{bmatrix} \left[\begin{array}{cc:c:c} 0 & I & 0 & 0 \\ C_2 & 0 & D_{21} & D_{22} \end{array}\right], \tag{5.31}$$

where the dependence of the matrices on ρ is suppressed for brevity.

According to Theorem 1, the worst-case gain of $\mathcal{F}_u(G(H,F),\Delta)$ is bounded by γ

if there exists $P = P^T$ satisfying

$$\begin{bmatrix} \mathcal{A}^T(\rho)P + P\mathcal{A}(\rho) & \star & \star \\ \mathcal{B}_w^T(\rho)P & 0 & \star \\ \mathcal{B}_d^T(\rho)P & 0 & -\gamma I \end{bmatrix} + \frac{1}{\gamma}\begin{bmatrix} \mathcal{C}_e^T(\rho) \\ \mathcal{D}_{ew}^T(\rho) \\ \mathcal{D}_{ed}^T(\rho) \end{bmatrix}(\star) + \begin{bmatrix} \mathcal{C}_z^T(\rho) \\ \mathcal{D}_{zw}^T(\rho) \\ \mathcal{D}_{zd}^T(\rho) \end{bmatrix} M(\star) < 0\ \forall \rho \in \mathcal{P}. \tag{5.32}$$

However, inequality (5.32) is not a LMI because of the presence of bilinear terms involving P and the state-space matrices of F. For example, the term $\mathcal{A}^T(\rho)P$ involves the product of $A_F(\rho)$ and P, both of which are variables to be selected. Since (5.32) is a bilinear matrix inequality (BMI), it will be referred to using the short form $BMI_{WC}(G(H,F), P, \gamma, \Psi, M) < 0$. The subscript WC indicates that it is a BMI associated with a worst-case gain problem and the arguments $(G(H,F), P, \gamma, \Psi, M)$ indicate that the plant H, estimator F, Lyapunov matrix P, gain bound γ, and IQC factorization (Ψ, M) are involved. $BMI_{WC}(G(H,F), P, \gamma, \Psi, M) < 0$ is a sufficient condition for $\mathcal{F}_u(G(H,F), \Delta)$ to have bounded worst-case gain. However, this formulation is non-convex due to the presence of the bilinear terms. The next theorem is adapted from [86] and provides convex LMI conditions that are equivalent to the non-convex $BMI_{WC}(G(H,F), P, \gamma, \Psi, M) < 0$.

Theorem 2. *Let H be a quadratically stable LPV system, Π be a strict PN multiplier, and (Ψ, M) be a stable factorization of Π. Let the interconnection of H and Ψ have the state-space realization given in (5.28). Let the columns of $N(\rho)$ form bases for the null space of $\begin{bmatrix} C_2(\rho) & D_{21}(\rho) & D_{22}(\rho) \end{bmatrix}$, where $D_{22}(\rho)$ has full row rank $\forall \rho \in \mathcal{P}$. Denote $\bar{N} := diag(N(\rho), I)$. There exists a quadratically stable estimator F of order n_F and some matrix $P = P^T$ satisfying $BMI_{WC}(G(H,F), P, \gamma, \Psi, M) < 0$ if and only if there exist symmetric matrices X and Z satisfying*

$$X - Z \geq 0,\ rank(X - Z) \leq n_F, \tag{5.33}$$

$$\begin{bmatrix} A^T(\rho)Z + ZA(\rho) & \star & \star \\ B_1^T(\rho)Z & 0 & \star \\ B_2^T(\rho)Z & 0 & -\gamma I \end{bmatrix} + \begin{bmatrix} C_1^T(\rho) \\ D_{11}^T(\rho) \\ D_{12}^T(\rho) \end{bmatrix} M(\star) < 0\ \forall \rho \in \mathcal{P},\ \textit{and} \tag{5.34}$$

$$\bar{N}^T \left(\begin{bmatrix} A^T(\rho)X + \star & \star & \star & \star \\ B_1^T(\rho)X & 0 & \star & \star \\ B_2^T(\rho)X & 0 & -\gamma I & \star \\ -C_3(\rho) & -D_{31}(\rho) & -D_{32}(\rho) & -\gamma I \end{bmatrix} + \begin{bmatrix} C_1^T(\rho) \\ D_{11}^T(\rho) \\ D_{12}^T(\rho) \\ 0 \end{bmatrix} M(\star) \right) \bar{N} < 0 \, \forall \rho \in \mathcal{P}. \tag{5.35}$$

Further, feasibility of conditions (5.33), (5.34), and (5.35) implies that $\mathcal{F}_u(G(H,F),\Delta)$ *satisfies*

(1) $\lim_{T\to\infty} x_e(T) = 0 \ \forall x_e(0) \in \mathbb{R}^{n_H+n_\Psi+n_F}$*,* $\forall d \in L_2^{n_d}$*,* $\forall \Delta \in \mathbf{\Delta}_\Pi$*, and* $\forall \rho \in \mathcal{T}$*, and*

(2) $\sup_{\Delta\in\mathbf{\Delta}_\Pi} \|\mathcal{F}_u(G(H,F),\Delta)\| \le \gamma$*.*

Proof. The proof of sufficiency adapts the proof of Lemma 3.1 in [116]. Assume there exists a quadratically stable estimator F of order n_F (where n_F is any positive integer) and some matrix $P = P^T$ satisfying $BMI_{WC}(G(H,F), P, \gamma, \Psi, M) < 0$. Apply the Schur complement lemma to show that $BMI_{WC}(G(H,F), P, \gamma, \Psi, M) < 0$ is equivalent to

$$\begin{bmatrix} \mathcal{A}^T(\rho)P + P\mathcal{A}(\rho) & \star & \star & \star \\ \mathcal{B}_w^T(\rho)P & 0 & \star & \star \\ \mathcal{B}_d^T(\rho)P & 0 & -\gamma I & \star \\ \mathcal{C}_e(\rho) & \mathcal{D}_{ew}(\rho) & \mathcal{D}_{ed}(\rho) & -\gamma I \end{bmatrix} + \begin{bmatrix} \mathcal{C}_z^T(\rho) \\ \mathcal{D}_{zw}^T(\rho) \\ \mathcal{D}_{zd}^T(\rho) \\ 0 \end{bmatrix} M(\star) < 0 \, \forall \rho \in \mathcal{P}. \tag{5.36}$$

Inequality (5.36) can be rewritten, using the matrix expressions given in (5.31), as

$$L(\rho) + Q^T\Theta(\rho)R(\rho) + R^T(\rho)\Theta^T(\rho)Q < 0 \, \forall \rho \in \mathcal{P}, \tag{5.37}$$

where $\Theta(\rho) := \left[\begin{smallmatrix} A_F(\rho) & B_F(\rho) \\ C_F(\rho) & D_F(\rho) \end{smallmatrix}\right]$,

$$L(\rho) := \begin{bmatrix} \begin{bmatrix} A(\rho) & 0 \\ 0 & 0 \end{bmatrix}^T P + \star & \star & \star & \star \\ \begin{bmatrix} B_1^T(\rho) & 0 \end{bmatrix} P & 0 & \star & \star \\ \begin{bmatrix} B_2^T(\rho) & 0 \end{bmatrix} P & 0 & -\gamma I & \star \\ \begin{bmatrix} -C_3(\rho) & 0 \end{bmatrix} & -D_{31}(\rho) & -D_{32}(\rho) & -\gamma I \end{bmatrix} + \begin{bmatrix} C_1^T(\rho) \\ 0 \\ D_{11}^T(\rho) \\ D_{12}^T(\rho) \\ 0 \end{bmatrix} M(\star), \quad (5.38)$$

$$Q := \begin{bmatrix} \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} P & \begin{matrix} 0 \\ 0 \end{matrix} & \begin{matrix} 0 \\ 0 \end{matrix} & \begin{matrix} 0 \\ I \end{matrix} \end{bmatrix}, \text{ and } R(\rho) := \begin{bmatrix} 0 & I & 0 & 0 & 0 \\ C_2(\rho) & 0 & D_{21}(\rho) & D_{22}(\rho) & 0 \end{bmatrix}. \quad (5.39)$$

Let the columns of $N(\rho)$ form bases for the null space of $\begin{bmatrix} C_2(\rho) & D_{21}(\rho) & D_{22}(\rho) \end{bmatrix}$. Define the matrices N_Q and $N_R(\rho)$ as:

$$N_Q := \begin{bmatrix} P^{-1}\begin{bmatrix} I \\ 0 \end{bmatrix} & \begin{matrix} 0 \\ 0 \end{matrix} & \begin{matrix} 0 \\ 0 \end{matrix} \\ 0 & I & 0 \\ 0 & 0 & I \\ 0 & 0 & 0 \end{bmatrix} \text{ and } N_R(\rho) := \begin{bmatrix} N_1(\rho) & 0 \\ 0 & 0 \\ N_2(\rho) & 0 \\ N_3(\rho) & 0 \\ 0 & I \end{bmatrix}, \quad (5.40)$$

where $N_1(\rho)$, $N_2(\rho)$, and $N_3(\rho)$ correspond to a block partition of the rows of $N(\rho)$ consistent with the dimensions of $C_2(\rho)$, $D_{21}(\rho)$, and $D_{22}(\rho)$, respectively. The columns of N_Q and $N_R(\rho)$ form bases for the null spaces of Q and $R(\rho)$, respectively. From the matrix elimination lemma (Lemma 3.1 in [116]), there exists a matrix $\Theta(\rho)$ of compatible dimensions satisfying inequality (5.37) if and only if

$$N_Q^T L(\rho) N_Q < 0 \;\forall \rho \in \mathcal{P} \text{ and} \quad (5.41)$$

$$N_R^T(\rho) L(\rho) N_R(\rho) < 0 \;\forall \rho \in \mathcal{P}. \quad (5.42)$$

Next, define $n := n_H + n_\Psi$ and partition P as $\begin{bmatrix} X & X_2 \\ X_2^T & X_3 \end{bmatrix}$, where $X = X^T \in$

$\mathbb{R}^{n\times n}$, $X_3 = X_3^T \in \mathbb{R}^{n_F\times n_F}$, and $X_2 \in \mathbb{R}^{n\times n_F}$. Further, partition P^{-1} as $\begin{bmatrix} Y & \bullet \\ \bullet & \bullet \end{bmatrix}$, where $Y = Y^T \in \mathbb{R}^{n\times n}$ and $\bullet$ denotes terms that are not relevant here. Using these Lyapunov matrix partitions, inequality (5.42) is shown to be equivalent to inequality (5.35). Further, inequality (5.41) is shown to be equivalent to

$$\begin{bmatrix} YA^T(\rho) + A(\rho)Y & \star & \star \\ B_1^T(\rho) & 0 & \star \\ B_2^T(\rho) & 0 & -\gamma I \end{bmatrix} + \begin{bmatrix} YC_1^T(\rho) \\ D_{11}^T(\rho) \\ D_{12}^T(\rho) \end{bmatrix} M(\star) < 0 \; \forall \rho \in \mathcal{P}. \tag{5.43}$$

Setting $Z \coloneqq Y^{-1}$ and multiplying inequality (5.43) on the left and right by $\operatorname{diag}(Z, I, I)$ yields (5.34).

Using the partition for P, the $(1,1)$ block of $BMI_{WC}(G(H,F), P, \gamma, \Psi, M) < 0$ yields $A_F^T(\rho) X_3 + X_3 A_F(\rho) < 0 \; \forall \rho \in \mathcal{P}$. Since F is assumed to be quadratically stable, Definition 3 implies that $X_3 > 0$. A variation of the matrix dilation lemma (Lemma 7.9 in [117]) is stated and proved as Lemma 7 in Appendix A.2. From Lemma 7, it is concluded that $X - Z \geq 0$ and $\operatorname{rank}(X - Z) \leq n_F$.

For necessity, assume that there exist symmetric matrices $X, Z \in \mathbb{R}^{n\times n}$ satisfying conditions (5.33), (5.34), and (5.35). By a variation of the matrix dilation lemma (Lemma 7 in Appendix A.2), there exist $X_2 \in \mathbb{R}^{n\times n_F}$ and $X_3 = X_3^T \in \mathbb{R}^{n_F\times n_F}$ such that $X_3 > 0$ and $\begin{bmatrix} X & X_2 \\ X_2^T & X_3 \end{bmatrix}^{-1} = \begin{bmatrix} Z^{-1} & \bullet \\ \bullet & \bullet \end{bmatrix}$. The algebraic steps used in the proof of sufficiency are now reversed. Specifically, from the matrix elimination lemma, X and Z satisfy LMIs (5.34) and (5.35) if and only if there exists a matrix $\Theta(\rho)$ of compatible dimensions satisfying inequality (5.37). Partition $\Theta(\rho)$ as given before, where $A_F \in \mathbb{R}^{n_F\times n_F}$. Note that $F \overset{s}{=} \left[\begin{array}{c|c} A_F(\rho) & B_F(\rho) \\ \hline C_F(\rho) & D_F(\rho) \end{array}\right]$ and $P \coloneqq \begin{bmatrix} X & X_2 \\ X_2^T & X_3 \end{bmatrix}$ satisfy $BMI_{WC}(G(H,F), P, \gamma, \Psi, M) < 0$. Using this partition for P, the $(1,1)$ block of $BMI_{WC}(G(H,F), P, \gamma, \Psi, M) < 0$ yields $A_F^T(\rho) X_3 + X_3 A_F(\rho) < 0 \; \forall \rho \in \mathcal{P}$. From Definition 3, $X_3 > 0$ implies that F is quadratically stable.

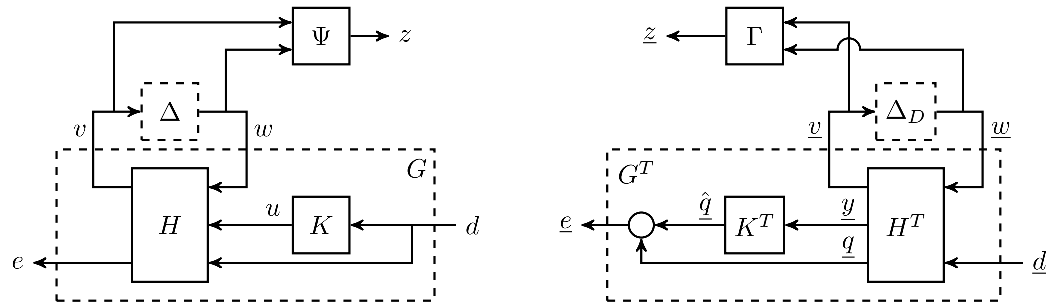
Finally, since H was already assumed to be quadratically stable, the quadratic stability of F implies the quadratic stability of $G(H,F)$. From Theorem 1, if there exists $P = P^T$ satisfying $BMI_{WC}(G(H,F), P, \gamma, \Psi, M) < 0$, then $\mathcal{F}_u(G(H,F), \Delta)$ satisfies statements (1) and (2). □

The conditions (5.33), (5.34), and (5.35) are LMIs in the variables X, Z, M, and γ. Hence, Theorem 2 circumvents the non-convexity of $BMI_{WC}\left(G\left(H,F\right),P,\gamma,\Psi,M\right)<0$ by providing equivalent LMI conditions. This is done purely through linear algebra, including the application of the Schur complement lemma, the matrix elimination lemma, and a variation of the the matrix dilation lemma. Theorem 2 results in no additional conservatism over the sufficient conditions of Theorem 1. Moreover, Theorem 2 is different from the existing results because it allows for grid-based LPV plants whose state matrices are arbitrary functions of the parameters. The rank constraint on $X-Z$ given in (5.33) is not convex when $n_F<n$. However, by choosing $n_F\geq n$ one can ensure that the rank constraint is automatically satisfied. In practice, it suffices to choose $n_F=n$, yielding an estimator whose order equals the combined order of H and Ψ.

The decision variables X and Z obtained from the semidefinite program are used to complete P as $\begin{bmatrix} X & Z-X \\ Z-X & X-Z \end{bmatrix}$. Finally, the estimator F is reconstructed via an explicit procedure that only relies on the state-space matrices of H and Ψ, and the values of X and Z. This entails deriving explicit expressions for N_Q and $N_R\left(\rho\right)$, and then forming a matrix T that spans the union of the null spaces of Q and $R\left(\rho\right)$. Upon observing that T is nonsingular, it is used in a congruence transformation of inequality (5.37). The remainder of the reconstruction procedure is omitted here, since it closely follows the proof of Lemma 3.1 given in Appendix A of [116].

5.4.2 Disturbance Feedforward

The disturbance feedforward problem is formulated using the interconnection shown in Figure 5.5a. H is a nominal LPV plant with parameters $\rho\in\mathcal{P}$, states $x_H\in\mathbb{R}^{n_H}$, control inputs $u\in\mathbb{R}^{n_u}$, disturbance inputs $d\in\mathbb{R}^{n_d}$, and generalized errors $e\in\mathbb{R}^{n_e}$. The LPV plant H is connected with an uncertainty Δ via signals $v\in\mathbb{R}^{n_v}$ and $w\in\mathbb{R}^{n_w}$. This creates an uncertain LPV system $\mathcal{F}_u\left(H,\Delta\right)$ from the inputs u and d to the output e. The problem is to synthesize a feedforward controller K that uses the disturbances d to generate control inputs u. The synthesis objective is to bound the worst-case induced L_2 norm from d to e over the set of uncertainties that satisfy IQC$\left(\Pi\right)$.



(a) The disturbance feedforward problem.

(b) The dual of the feedforward problem.

Figure 5.5: The dual of the disturbance feedforward problem is the output estimation problem.

In addition to the LFT of H and Δ, the IQC filter Ψ is appended such that it is driven by signals v and w, and produces signal z. The interconnection of H and Ψ has the state-space representation

$$\begin{bmatrix} \dot{x} \\ z \\ e \end{bmatrix} = \begin{bmatrix} A(\rho) & B_1(\rho) & B_2(\rho) & B_3(\rho) \\ C_1(\rho) & D_{11}(\rho) & D_{12}(\rho) & D_{13}(\rho) \\ C_2(\rho) & D_{21}(\rho) & D_{22}(\rho) & D_{23}(\rho) \end{bmatrix} \begin{bmatrix} x \\ w \\ u \\ d \end{bmatrix}, \tag{5.44}$$

where $x = \left[x_H^T, x_\Psi^T\right]^T \in \mathbb{R}^{n_H+n_\Psi}$ are the combined states of H and Ψ. It is assumed that $D_{22}(\rho) \in \mathbb{R}^{n_e \times n_u}$ has full column rank $\forall \rho \in \mathcal{P}$. This ensures that all components of the control input u affect some component of the generalized error e.

The feedforward controller K to be synthesized has the state-space representation:

$$\begin{bmatrix} \dot{x}_K \\ u \end{bmatrix} = \begin{bmatrix} A_K(\rho) & B_K(\rho) \\ C_K(\rho) & D_K(\rho) \end{bmatrix} \begin{bmatrix} x_K \\ d \end{bmatrix}, \tag{5.45}$$

where $x_K \in \mathbb{R}^{n_K}$ is the state, d is the input, and u is the output of the feedforward controller. As shown by the large dashed box in Figure 5.5a, the closed-loop formed by the interconnection of H and K is denoted by G, with states $x_G = \left[x_H^T, x_K^T\right]^T$. In the remainder of this section, the notation $G(H, K)$ will be used in some cases to make explicit the dependence of G on H and K. Theorem 1 provides conditions to bound the worst-case gain of $\mathcal{F}_u\left(G\left(H,K\right),\Delta\right)$ over the set $\mathbf{\Delta}_\Pi$. The objective is to synthesize K which minimizes this bound.

Next consider the extended LPV system formed by the interconnection of $G\left(H,K\right)$ and Ψ. This extended system has the state-space realization:

$$\begin{bmatrix} \dot{x}_e \\ z \\ e \end{bmatrix} = \begin{bmatrix} \mathcal{A}\left(\rho\right) & \mathcal{B}_w\left(\rho\right) & \mathcal{B}_d\left(\rho\right) \\ \mathcal{C}_z\left(\rho\right) & \mathcal{D}_{zw}\left(\rho\right) & \mathcal{D}_{zd}\left(\rho\right) \\ \mathcal{C}_e\left(\rho\right) & \mathcal{D}_{ew}\left(\rho\right) & \mathcal{D}_{ed}\left(\rho\right) \end{bmatrix} \begin{bmatrix} x_e \\ w \\ d \end{bmatrix}, \tag{5.46}$$

where $x_e = \left[x_H^T, x_\Psi^T, x_K^T\right]^T \in \mathbb{R}^{n_H+n_\Psi+n_K}$ are the combined states of H, Ψ, and K. These state-space matrices are expressed in terms of the matrices appearing in Equations (5.44) and (5.45) as:

$$\begin{bmatrix} \mathcal{A} & \mathcal{B}_w & \mathcal{B}_d \\ \mathcal{C}_z & \mathcal{D}_{zw} & \mathcal{D}_{zd} \\ \mathcal{C}_e & \mathcal{D}_{ew} & \mathcal{D}_{ed} \end{bmatrix} = \left[\begin{array}{cc|c|c} A & 0 & B_1 & B_3 \\ 0 & 0 & 0 & 0 \\ \hline C_1 & 0 & D_{11} & D_{13} \\ \hline C_2 & 0 & D_{21} & D_{23} \end{array}\right] + \left[\begin{array}{cc} 0 & B_2 \\ I & 0 \\ \hline 0 & D_{12} \\ \hline 0 & D_{22} \end{array}\right] \begin{bmatrix} A_K & B_K \\ C_K & D_K \end{bmatrix} \left[\begin{array}{cc|c|c} 0 & I & 0 & 0 \\ 0 & 0 & 0 & I \end{array}\right], \tag{5.47}$$

where the dependence of the matrices on ρ is suppressed for brevity.

According to Theorem 1, the worst-case gain of $\mathcal{F}_u\left(G\left(H,K\right),\Delta\right)$ is bounded by γ if there exists $P = P^T$ satisfying $BMI_{WC}\left(G\left(H,K\right),P,\gamma,\Psi,M\right) < 0$. Applying the Schur complement lemma, $BMI_{WC}\left(G\left(H,K\right),P,\gamma,\Psi,M\right) < 0$ is equivalent to the inequality

$$\begin{bmatrix} \mathcal{A}^T\left(\rho\right)P + P\mathcal{A}\left(\rho\right) & \star & \star & \star \\ \mathcal{B}_w^T\left(\rho\right)P & 0 & \star & \star \\ \mathcal{B}_d^T\left(\rho\right)P & 0 & -\gamma I & \star \\ \mathcal{C}_e\left(\rho\right) & \mathcal{D}_{ew}\left(\rho\right) & \mathcal{D}_{ed}\left(\rho\right) & -\gamma I \end{bmatrix} + \begin{bmatrix} \mathcal{C}_z^T\left(\rho\right) \\ \mathcal{D}_{zw}^T\left(\rho\right) \\ \mathcal{D}_{zd}^T\left(\rho\right) \\ 0 \end{bmatrix} M\left(\star\right) < 0 \;\forall \rho \in \mathcal{P}. \tag{5.48}$$

As before, inequality (5.48) is not a LMI because of the presence of bilinear terms involving P and the state-space matrices of K. For example, the term $\mathcal{A}^T\left(\rho\right)P$ involves the product of $A_K\left(\rho\right)$ and P, similar to that appearing in the output estimation problem. However, the disturbance feedforward problem has the additional complication that $\mathcal{C}_z\left(\rho\right)$, $\mathcal{D}_{zw}\left(\rho\right)$, and $\mathcal{D}_{zd}\left(\rho\right)$ depend on the state-space matrices of the feedforward controller to be synthesized. Thus, the second term in inequality (5.48) involves quadratic products of the state-space matrices of K. As a consequence, it does not seem possible to convert inequality (5.48) into equivalent LMI conditions via the matrix elimination lemma. Hence an alternative approach is followed wherein the dual of the disturbance feedforward problem is considered. This alternative approach provides convex (LMI) synthesis conditions for K.

Before considering the dual problem, recall the key implication of Lemma 6. Lemma 6 proved that the sufficient conditions for bounding the worst-case gain of $\mathcal{F}_u\left(G, \Delta\right)$ over the uncertainty set $\mathbf{\Delta}_{\Pi}$ are equivalent to the sufficient conditions for bounding the worst-case gain of $\mathcal{F}_u\left(G^T, \Delta_D\right)$ over the set $\mathbf{\Delta}_{\boldsymbol{D}(\Pi)}$. Now, denote $\mathcal{F}_u\left(G\left(H, K\right), \Delta\right)$ shown in Figure 5.5a as the *primal* uncertain LPV system. The corresponding *dual* uncertain LPV system is $\mathcal{F}_u\left(G^T\left(H^T, K^T\right), \Delta_D\right)$ as shown in Figure 5.5b. Here, G^T is the dual of G in the sense of Definition 4 and $\Delta_D \in \mathbf{\Delta}_{\boldsymbol{D}(\Pi)}$. It is verified from algebra that G^T is the interconnection of H^T and K^T, shown in Figure 5.5b. As before, the notation $G^T\left(H^T, K^T\right)$ is used to make explicit the dependence of G^T on H^T and K^T.

H^T is the dual of H with state $\underline{x}_H \in \mathbb{R}^{n_H}$, inputs $\underline{w} \in \mathbb{R}^{n_v}$ and $\underline{d} \in \mathbb{R}^{n_e}$, and outputs $\underline{v} \in \mathbb{R}^{n_w}$, $\underline{y} \in \mathbb{R}^{n_u}$, and $\underline{q} \in \mathbb{R}^{n_d}$. The inputs of H^T are partitioned conformably with the outputs of H. For example, in the preceding discussion, the outputs of H were partitioned as $v \in \mathbb{R}^{n_v}$ and $e \in \mathbb{R}^{n_e}$. Consequently, the inputs of H^T are partitioned as $\underline{w} \in \mathbb{R}^{n_v}$ and $\underline{d} \in \mathbb{R}^{n_e}$. Similarly, the outputs of H^T are partitioned conformably with the inputs of H. K^T is the dual of K with state $\underline{x}_K \in \mathbb{R}^{n_K}$, input $\underline{y} \in \mathbb{R}^{n_u}$, and output $\underline{\hat{q}} \in \mathbb{R}^{n_d}$. On comparing Figure 5.5b with Figure 5.4, it is inferred that K^T is effectively an output estimator for H^T. The output $\underline{\hat{q}}$ of K^T is effectively an estimate of the output $\underline{q}$ of H^T. Because of the way the feedforward problem is formulated in Figure 5.5a, note that the estimation error $\underline{e}$ equals $\underline{\hat{q}} + \underline{q}$ rather than $\underline{\hat{q}} - \underline{q}$ as was the case in Section 5.4.1. The next theorem proves that synthesizing an output estimator K^T to bound the worst-case gain of $\mathcal{F}_u\left(G^T\left(H^T, K^T\right), \Delta_D\right)$ over the set $\mathbf{\Delta}_{\boldsymbol{D}(\Pi)}$ is equivalent to synthesizing a disturbance feedforward controller to bound the worst-case gain of $\mathcal{F}_u\left(G\left(H, K\right), \Delta\right)$ over the set $\mathbf{\Delta}_{\Pi}$.

Theorem 3. *Let H be a quadratically stable LPV system and Π be a strict PN multiplier. Let $\left(\Psi, M\right)$ be any stable factorization of Π and $\left(\Gamma, N\right)$ be any stable factorization of $\boldsymbol{D}\left(\Pi\right)$. Let $G\left(H, K\right)$ and $G^T\left(H^T, K^T\right)$ denote the closed-loop primal (Figure 5.5a) and dual (Figure 5.5b) systems for a given K, respectively. Then K is a quadratically stable feedforward controller that satisfies $BMI_{WC}\left(G\left(H, K\right), P, \gamma, \Psi, M\right) < 0$ for some* *symmetric matrix P if and only if K^T is a quadratically stable estimator that satisfies $BMI_{WC}\left(G^T\left(H^T, K^T\right), Q, \gamma, \Gamma, N\right) < 0$ for some symmetric matrix Q. Further, the feasibility of $BMI_{WC}\left(G\left(H,K\right), P, \gamma, \Psi, M\right) < 0$ implies that $\mathcal{F}_u\left(G\left(H,K\right), \Delta\right)$ satisfies:*

(1) $\lim_{T\to\infty} x_e\left(T\right) = 0$ $\forall x_e\left(0\right) \in \mathbb{R}^{n_H+n_\Psi+n_K}$, $\forall d \in L_2^{n_d}$, $\forall \Delta \in \boldsymbol{\Delta}_\Pi$, *and* $\forall \rho \in \mathcal{T}$, *and*

(2) $\sup_{\Delta\in\boldsymbol{\Delta}_\Pi} \left\|\mathcal{F}_u\left(G\left(H,K\right), \Delta\right)\right\| \leq \gamma$.

Proof. From statement (1) of Lemma 6, H is quadratically stable if and only if H^T is quadratically stable. Similarly, K is quadratically stable if and only if K^T is quadratically stable. From statement (2) of Lemma 6, Π is a strict PN multiplier if and only if $\boldsymbol{D}\left(\Pi\right)$ is a strict PN multiplier. For sufficiency, assume K is a quadratically stable feedforward controller that satisfies $BMI_{WC}\left(G\left(H,K\right), P, \gamma, \Psi, M\right) < 0$ for some $P = P^T$. It is verified that $G\left(H,K\right)$ is quadratically stable because H and K are both quadratically stable. From statement (1) of Lemma 6, it follows that $G\left(H,K\right)^T = G^T\left(H^T, K^T\right)$ is also quadratically stable. From statement (3) of Lemma 6, $\exists Q = Q^T$ satisfying $BMI_{WC}\left(G^T\left(H^T, K^T\right), Q, \gamma, \Gamma, N\right) < 0$. For necessity, use similar arguments. Finally, from Theorem 1, if there exists $P = P^T$ satisfying $BMI_{WC}\left(G\left(H,K\right), P, \gamma, \Psi, M\right) < 0$, then $\mathcal{F}_u\left(G\left(H,K\right), \Delta\right)$ satisfies statements (1) and (2). □

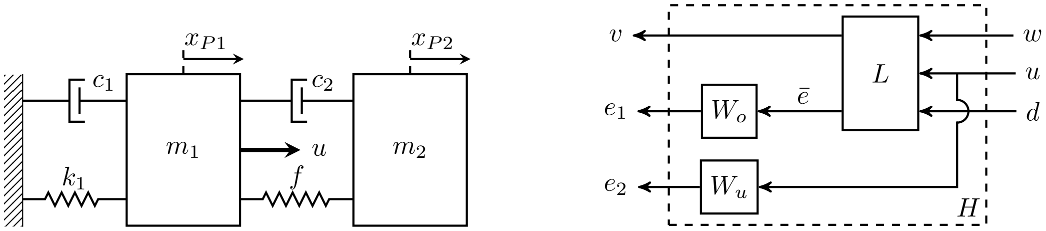
Theorem 3 shows that K^T is an output estimator that satisfies the sufficient conditions for bounding the worst-case gain of $\mathcal{F}_u\left(G^T\left(H^T, K^T\right), \Delta_D\right)$ by γ if and only if K is a disturbance feedforward controller that satisfies the sufficient conditions for bounding the worst-case gain of $\mathcal{F}_u\left(G\left(H,K\right), \Delta\right)$ by γ. Hence, the disturbance feedforward problem is solved by implementing its corresponding dual form. In particular, when a feedforward synthesis problem is specified using H and Π, Theorem 2 is invoked on H^T and $\boldsymbol{D}\left(\Pi\right)$ so that the estimator K^T is synthesized instead. Theorem 2 is implemented as a semidefinite program as explained in Section 5.5.

One technical issue is that the solution of the disturbance feedforward problem by the dual semidefinite program requires an appropriate parametrization of the IQC multiplier. For example, if Δ is defined by multiplication in the time-domain with a norm-bounded, time-varying real scalar, it satisfies all IQCs defined by multipliers Π of the form $\begin{bmatrix} X & Y \\ Y^T & -X \end{bmatrix}$, where $X = X^T \geq 0$ and $Y = -Y^T$ [99]. In this example, Π is parametrized by the real symmetric matrix X and the real skew-symmetric matrix Y. In general, Π is parametrized by several variables. While parametrizations aid in the enlargement of the set of feasible IQC multipliers, only those that preserve the linearity of the matrix inequalities can be implemented in semidefinite programs. For several perturbations, suitable parametrizations of Π are available in Section 4.2 of [92].

However, the dual multipliers (Definition 5) involve matrix inversion. As a result, even if the primal multiplier has a convex parametrization, the dual multiplier may not. Hence, suitable parametrizations of the dual multiplier should be found independently and on a case-by-case basis. However, for linear perturbations, if Δ satisfies the IQCs defined by several primal multipliers Π_i, then Δ_D satisfies the IQCs defined by every one of the corresponding dual multipliers $\boldsymbol{D}(\Pi_i)$ [96]. Consequently, affine parametrizations of Π_i and $\boldsymbol{D}(\Pi_i)$ can be used for the primal and dual worst-case gain problems, respectively. More details can be found in Section 2.1 of [96].

5.5 Numerical Example

The following numerical example illustrates convex feedforward synthesis for a grid-based LPV plant that is affected by a sector-constrained nonlinearity. Figure 5.6a depicts a spring-mass-damper system consisting of two springs, two masses, and two dampers. The masses are $m_1 = 1$ kg and $m_2 = 0.5$ kg. The spring connecting the wall and mass m_1 is linear and has a spring constant $k_1 = 1\ \mathrm{N\,m^{-1}}$. The spring connecting the two masses is nonlinear, where $f : \mathbb{R} \to \mathbb{R}$ denotes the nonlinear function mapping the spring deformation to the spring force. For a spring deformation $v \in \mathbb{R}$, the spring force is $f(v) := k_2 v + \Delta(v)$. Here, $k_2 = 1\ \mathrm{N\,m^{-1}}$ denotes the linear spring constant and $\Delta : \mathbb{R} \to \mathbb{R}$ denotes a sector-constrained nonlinear function. The damping coefficient c_1 is certain, but depends on a time-varying scheduling parameter $\rho(t)$ as $c_1 = |\sin(\rho(t))|$. Admissible parameter trajectories satisfy $\rho(t) \in \mathcal{P} = \left[0, \frac{\pi}{3}\right]\ \forall t \geq 0$. Since c_1 is a transcendental function of ρ, this problem is not directly solvable by the LFT-LPV approach [85]. Following the grid-based LPV approach, the parameter space is discretized into three grid points $\left\{0, \frac{\pi}{6}, \frac{\pi}{3}\right\}$. These three points are simply chosen for the purpose of demonstration and the grid may be made as dense as needed [41]. The damping coefficient $c_2 = 2\ \mathrm{N\,s\,m^{-1}}$ is certain and time-invariant.



(a) The damper c_1 is parameter-varying and the spring force f is a nonlinear function of the spring deformation.

(b) The weights penalize the tracking error at low frequencies and the control effort at high frequencies.

Figure 5.6: Left: The spring-mass-damper system. Right: The interconnection showing the frequency-dependent weights W_o and W_u.

A command tracking problem is formulated as follows. Mass m_1 is externally forced through the control input u. The positions of m_1 and m_2 relative to their respective equilibrium positions are denoted by x_{P1} and x_{P2}. The commanded position of mass m_2 relative to its equilibrium position is denoted by d. The objective is to design a feedforward controller K that uses d to generate u such that x_{P2} tracks the reference command. The design should ensure that large tracking errors are avoided at low frequencies and large control inputs are avoided at high frequencies. The feedforward controller should be scheduled with the parameter $\rho(t)$ and should be robust to the sector-constrained nonlinearity Δ. In order to describe the equations of motion of the spring-mass-damper system, consider a LPV system L with the state-space representation

$$\begin{bmatrix} \dot{x}_{P1} \\ \dot{x}_{P2} \\ \dot{x}_{P3} \\ \dot{x}_{P4} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{-(k_1+k_2)}{m_1} & \frac{k_2}{m_1} & \frac{-(c_1+c_2)}{m_1} & \frac{c_2}{m_1} \\ \frac{k_2}{m_2} & \frac{-k_2}{m_2} & \frac{c_2}{m_2} & \frac{-c_2}{m_2} \end{bmatrix} \begin{bmatrix} x_{P1} \\ x_{P2} \\ x_{P3} \\ x_{P4} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{m_1} & \frac{1}{m_1} & 0 \\ \frac{-1}{m_2} & 0 & 0 \end{bmatrix} \begin{bmatrix} w \\ u \\ d \end{bmatrix}, \tag{5.49}$$

$$v = x_{P2} - x_{P1}, \text{ and } \bar{e} = d - x_{P2}. \tag{5.50}$$

where $x_{P3} = \dot{x}_{P1}$ and $x_{P4} = \dot{x}_{P2}$ are the velocities of the two masses and $w = \Delta(v)$.

The output v captures the net deformation of the spring connecting the two masses.

As per the definition of the function f given previously, the resulting spring force is $f\left(v\right) = k_2 x_{P2} - k_2 x_{P1} + w$, where $w = \Delta\left(v\right)$ is the nonlinear component of the spring force. The output $\bar{e}$ captures the error between the commanded and actual positions of mass m_2. The equations of motion for the entire spring-mass-damper system, including the nonlinearity, are given by $\mathcal{F}_u\left(L, \Delta\right)$. Figure 5.6b shows the interconnection of L and two weighting functions W_o and W_u. The weight $W_o = \frac{0.1}{(s+0.1)(s+0.01)}$ penalizes the tracking error $\bar{e}$ at low frequencies. The weighting function $W_u = \frac{100(s+0.1)}{s+1000}$ penalizes the control effort u at high frequencies. The generalized error vector is denoted by e and has two components: $e_1 := W_o\bar{e}$ and $e_2 := W_u u$. As shown in Figure 5.6b, the interconnection of L, W_o, and W_u is denoted by H in order to relate back to the notation used Figure 5.5a.

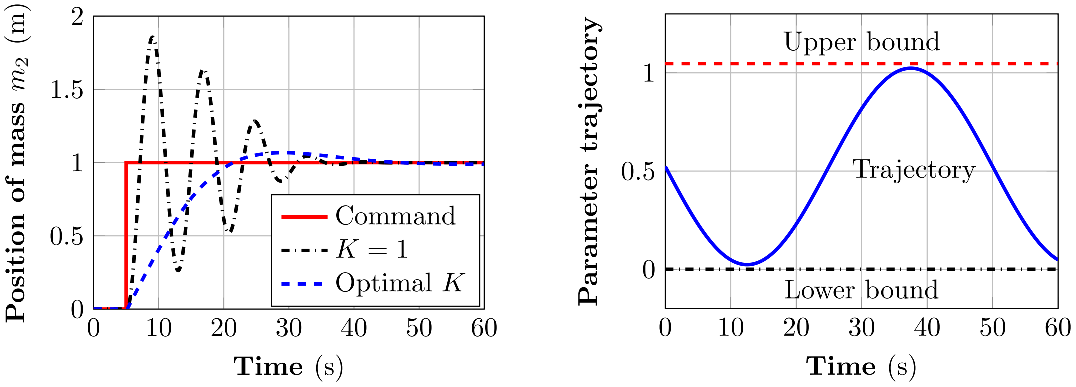
Since Δ is a sector-constrained nonlinearity, it satisfies all IQCs defined by multipliers Π of the form $\begin{bmatrix} -2\alpha\beta & \alpha+\beta \\ \alpha+\beta & -2 \end{bmatrix}$, where α and β define the slopes of the sector [99]. In this example, the sector is defined by lines of slope $\alpha = -0.9$ and $\beta = 1.5$. These values of α and β ensure that Π is a strict PN multiplier. The choice of the sector is important in ensuring the applicability of Theorem 3. For example, f is a nonlinear function in a rotated sector defined by lines of slope 0.1 and 2.5. However, the multiplier for this sector is not strict PN and cannot be used in Theorem 3. The dual multiplier $\boldsymbol{D}\left(\Pi\right) = \left(\alpha - \beta\right)^{-2}\Pi$ is simply a scaled version of the primal.

First, an analysis problem is considered by setting $K = 1$. This feedforward controller is static, parameter-invariant, and non-optimal. Let γ denote an upper bound on the worst-case gain from d to e. To find the least upper bound, a semidefinite program is formulated with γ as the linear cost function. The LMIs given in Theorem 1 are enforced at each grid point and share a common Lyapunov matrix, making this approach significantly different from a pointwise analysis. Further, from Lemma 6, any bound γ on the worst-case gain of $\mathcal{F}_u\left(G, \Delta\right)$ over the set $\boldsymbol{\Delta}_{\Pi}$ is also a bound on the worst-case gain of $\mathcal{F}_u\left(G^T, \Delta_D\right)$ over the set $\boldsymbol{\Delta}_{\boldsymbol{D}(\Pi)}$. Hence, Theorem 1 may be applied either using G and a J-spectral factorization of Π or using G^T and a J-spectral factorization of $\boldsymbol{D}\left(\Pi\right)$. The least upper bound on the worst-case gain from d to e is 1109.

Next, Theorem 3 is applied to synthesize the optimal feedforward controller K. As explained previously, specifying a feedforward synthesis problem in terms of H and Π

is equivalent to specifying an estimator synthesis problem in terms of H^T and $\boldsymbol{D}(\Pi)$. Hence, Theorem 2 is implemented with its LMIs formulated using H^T and a J-spectral factorization of $\boldsymbol{D}(\Pi)$. The least upper bound on the worst-case gain from d to e is 5.48, demonstrating that the command tracking performance is significantly better with an optimally designed LPV feedforward controller.

Finally, a simulation is performed to evaluate the command tracking performances of the closed-loops formed using $K = 1$ and the optimal K. For the simulation, the nominal plant is considered and the equations of motion are given by $\mathcal{F}_u(L, 0)$. The reference command d is specified as the unit step function with a step time of 5 s. The LPV plant and feedforward controller are both scheduled using the trajectory $\rho(t) = \frac{1}{2}\sin\left(\frac{2\pi}{50}t + \pi\right) + \frac{\pi}{6}$. All the states are initialized at zero and the simulation is performed for 60 s. Figure 5.7a shows the closed-loop responses of the position of mass m_2, given by the state x_{P2}, using $K = 1$ and the optimal K. Since $K = 1$ is static, parameter-invariant, and non-optimal, its command tracking performance is poor. In particular, it results in 85% overshoot of the reference command. The optimal K, synthesized using Theorem 3, exhibits good command tracking performance. In particular, it results in an overshoot of 6.8%, a rise time of 12 s (in rising from 10% to



(a) The closed-loop responses of mass m_2 to a unit step reference command using $K = 1$ and the optimal K.

(b) The parameter trajectory is an exogenous sinusoidal signal between the lower bound 0 and the upper bound $\pi/3$.

Figure 5.7: The command tracking performance of the feedforward controller is assessed in the time-domain.

90% of its steady-state value), and a settling time of 36 s (in settling to within 2% of its steady-state value). Figure 5.7b shows the parameter trajectory used in the simulation.

5.6 Conclusions

This chapter considered the twin problems of synthesizing output estimators and disturbance feedforward controllers for continuous-time, uncertain, gridded, linear parameter-varying (LPV) systems. Integral quadratic constraints (IQCs) were used to describe the uncertainty. While convex conditions are readily obtained for the output estimation problem, it does not seem possible to directly obtain convex conditions for the disturbance feedforward problem. However, notions of duality were developed for LPV systems and IQCs in the time-domain and used to demonstrate that the two problems are duals of each other. Further, a technical result was developed to prove that the sufficient conditions for bounding the worst-case gain of a primal uncertain LPV system are equivalent to those of its dual. Consequently, a convex synthesis of feedforward controllers is possible by solving the dual output estimation problem. The duality result has no effect on conservatism. A numerical example illustrated convex feedforward synthesis for a gridded LPV plant that was affected by a sector-constrained nonlinearity.

The next chapter applies the convex synthesis results for the design of the fault diagnosis algorithm.

Chapter 6

Fault Detection and Isolation Algorithm

6.1 Introduction

After the onset of an elevon fault, the control law switches from the nominal controller (Chapter 4) to the fault-tolerant controller (Chapter 7). This chapter describes the algorithm that performs the intermediate steps of detecting the fault and isolating its source. This chapter is organized as follows. Section 6.2 presents related work on fault detection and isolation (FDI). Section 6.3 provides the motivation behind and the scope of the particular problem considered here. Section 6.4 presents the requirements for the FDI algorithm. Sections 6.5, 6.6, and 6.7 present the architecture and the details of two designs of the algorithm. Finally, Section 6.8 validates the FDI algorithms using flight test data in an offline setting.

6.2 Background

In a general setting, the title of this chapter refers to two distinct, but complementary, tasks. *Fault detection* is the determination of one or more faults present in a system and the time of such detection. *Fault isolation* is the determination of the types and the locations of the faults. The interested reader may refer to Section 2 of [118] for additional terminology and their distinctions, e.g. fault diagnosis, identification, etc.

Most fault detection algorithms make use of the so-called residual to draw inferences regarding the presence of a fault. A typical algorithm consists of a residual generation step and a residual evaluation step [119]. The residuals may be generated using either model-based[1] [119] or data-driven [120] approaches. Popular textbooks on this subject broadly classify different residual generation methods as being either observer-based or parity space-based [119–123]. Some of these textbooks make finer classifications within each category, e.g. whether it is the state or the output that is being observed or whether the input is known or unknown [122, 123]. Regardless of the method used, it is desirable to generate residuals that are insensitive to noise and model uncertainties and sensitive to the faults under consideration.

The residual evaluation step utilizes properties of the residual to draw inferences regarding the presence of a fault and, possibly, its characteristics. A linear transformation or filter maybe used to increase the signal-to-noise ratio of the residual and extract any directional properties (of vector valued residuals) in order to aid the fault isolation step [124]. Ding provides two main classes of methods for evaluating the filtered residual: norm-based and statistics-based [119]. While the norm-based methods use one or more signal norms of the residual, the statistical methods use the statistical properties of the residual. Both classes of methods rely on thresholds to detect a fault.

In addition to the textbooks, there are many survey papers on the topic of fault diagnosis. For example, Isermann provides a survey of fault detection and diagnosis methods that are based on the estimation of the states and/or the parameters of the dynamic process under consideration [125]. Gertler provides an overview of fault detection and isolation using dynamic parity relations for additive and multiplicative faults and provides some connections between the parity relations and parameter estimation methods [124]. Gertler later extends this overview to include methods based on parameter estimation and principal component analysis [126]. Isermann and Ballé provide a short overview of the historical development of model-based fault detection and diagnosis methods, suggest some terminology for the field, and discuss trends in the applications of these methods [118]. Isermann revisits such an overview in [127].

Hwang et al. provide a more recent survey of fault detection, isolation, and reconfiguration methods [128]. To conduct the survey, they model faults originating in the

[1] This chapter exclusively deals with model-based approaches for fault detection.

actuators and the sensors as additive and those originating in the underlying process as multiplicative. In addition to the common parity space and observer-based methods, they describe optimization-based [129,130], Kalman filter-based [131], and artificial intelligence-based methods. They also describe methods for residual evaluation, e.g. sequential probability ratio test, cumulative sum algorithm, generalized likelihood ratio test, etc.

In addition to the aforementioned general literature, there are several papers that apply fault detection and diagnosis methods to aerospace-related problems. Patton and Chen review the application of parity space approaches for fault diagnosis on aerospace systems [132]. Traditionally, this involved checking the parity between the outputs of multiple sensors that measured the same quantity, i.e. *hardware redundancy* [133]. More recently, research efforts have shifted to using model-based and data-driven fault diagnosis methods to achieve the same ends, i.e. *analytical redundancy*. For example, Goupil explains the state of practice of fault detection at Airbus [134] and provides a specific example of analytical redundancy on-board the A380 [135]. In addition, the European ADDSAFE program investigated the applicability of advanced model-based fault detection and diagnosis methods to a commercial aircraft benchmark [136, 137]. In the academic world, several authors have tackled the problem of fault detection and isolation on commercial aircraft using observer-based methods that have their origins in feedback control theory, e.g. neural networks [138], H_∞ optimization [139–141], and sliding-mode [142]. Some of these papers consider faults affecting the inner control loops of the aircraft, e.g. in the elevator and the pitch rate sensor [139], while others consider faults affecting the outer loops, e.g. in the air data probes [140, 141]. Other authors have invoked concepts from parameter identification for fault isolation, e.g. see [143] for the description of a distributed fault diagnostic system for a business jet.

Analytical redundancy takes on additional significance for small UAS because of their limited size and payload capacity. The existing literature on fault diagnosis for UAS covers both actuator failures [144–148] and sensor failures [13–17, 145, 147]. Several methods have been proposed in this regard, e.g. a multiple-model Kalman filter that computes probabilities of different failure models [145], the superimposition of an excitation signal on the actuator commands [144], filters designed using H_∞ optimization [146–148], and multiple model adaptive estimation [149]. Some of these authors have demonstrated their methods using flight tests, with their algorithms implemented either online (i.e. running in real-time on-board the UAS) [146] or offline (running on a PC using prerecorded flight data) [147, 148]. For example, Pandita implemented a H_∞ fault detection filter in real-time on-board a small UAS in order to evaluate its performance in detecting and estimating aileron faults [146]. Freeman et al. compared the performances of a model-based method (an H_∞ filter) and a data-driven method (a Z-test on chosen closed-loop signals) in detecting aileron faults for a small UAS [147,148].

An important consideration when designing these algorithms is the effect of the feedback controller. For example, a well-designed feedback controller may suppress the fault of interest, thereby making it harder to detect. Stoustrup et al. showed that in the presence of model uncertainty, the designs of the controller and the FDI filter are coupled [150]. In other words, in order to meet all the tracking and detection objectives, the controller and the FDI filter must be designed simultaneously. Although this approach is appealing, it has some drawbacks, e.g. the FDI filter cannot be retrofitted onto an existing feedback controller. Thus this chapter eschews the aforementioned integrated approach in favor of a sequential approach, wherein the FDI filter is designed in consideration of the nominal controller already described in Chapter 4. (Section 7.3 of [146] provides a brief discussion on the advantages and disadvantages of each approach.) The interested reader may refer to [151, 152], and the references therein, for more information on FDI performance under closed-loop control, performance metrics for FDI filters, and controller-filter interaction in the presence of model uncertainty.

6.3 Motivation and Scope

The problem of fault detection, isolation, and fault-tolerant control is motivated by the low reliability of small UAS [153]. The UAV Lab previously quantified the reliability of a fixed-wing UAS called the Ultra Stick 120. In particular, a comprehensive fault tree analysis yielded a failure rate of 2.2×10^{-2} failures-per-flight-hour [2] for this aircraft [3]. This figure is orders of magnitude above the 10^{-9} level required of manned commercial aircraft. This is because the Ultra Stick 120 uses low-cost and hobby-grade

[2] This analysis provides a theoretical estimate of the reliability and no loss of aircraft has occurred to date.

components that suffer from single points of failure. The Vireo also uses low-cost and hobby-grade components (see Table 2.1). Although a similar fault tree analysis has not been conducted for the particular Vireo considered in this thesis, it is expected that its reliability is of the same order as that of the Ultra Stick 120.

As such, each component on the Vireo (see Table 2.1) can fail in a number of different ways, thereby contributing to the failure rate of the overall aircraft. In the interest of tractability, the scope of this chapter is limited both in the number and in the type of faults considered. Specifically, this chapter is limited to the detection and isolation of stuck faults in either of the elevons of the Vireo. The remainder of this section explains the rationale for this limited scope.

Of all the avionics components on the Vireo, the servomotors are the least reliable. Apart from regular wear and tear, they sustain large impact loads if the wingtips strike the ground during landing. To provide some empirical evidence, consider that on the particular Vireo operated by the UAV Lab, three servo motors have failed over the course of 35 flights conducted for the research presented in this thesis. The first servomotor became stuck due to broken internal gears. The second servomotor suffered a faulty potentiometer and thus stopped responding to control commands. The third servomotor suffered from a condition wherein it traveled only in one direction relative to its neutral position. All the three servomotors were new and had no operational history at the time of installation. Their failures were discovered during routine maintenance checks conducted between flights. Although the failure times are unknown, flight data suggests that the failures did not occur in-flight. Despite the small sample size, this empirical evidence suggests that the servomotors are highly susceptible to failure. In contrast, no other component listed in Table 2.1 has failed over the course of the 35 flights.

In addition to the three servomotor failure modes described above, the literature offers a few others, e.g. bias, stuck, hard-over, floating, oscillatory, and increased deadband or stiction [4]. These failure modes differ not only in their respective causes and effects but also in their likelihoods and severities. In general, failure modes with high likelihood and high severity pose the greatest risk. Researchers at the UAV Lab previously conducted a failure modes and effects analysis (FMEA) of hobby-grade servomotors and concluded that the stuck and the floating failure modes pose the greatest risk. In particular, loss of control [5] is likely to occur if either of these failure modes is left unchecked. (The interested reader may refer to Appendix A of [4] and Appendix 1 of [3] for the FMEA.) While both of these failure modes are interesting in their own right, this chapter will be limited to the stuck failure mode.

Stuck faults impose constraints on the flight envelope of the aircraft. Section III of [153] shows one example of the impact of stuck faults on the allowable flight envelope for a fixed wing aircraft. In particular, there may be fault magnitudes where it is not possible to trim the aircraft in the conventional sense, e.g. if an elevon gets stuck at one of its extreme positions. For simplicity, this chapter only considers stuck faults for which a steady, wings-level, constant altitude, and constant airspeed flight condition exists[3]. In order to achieve such a flight condition, the operable elevon must be re-trimmed at the same position as that of the faulted elevon. Thus the magnitude of the stuck fault implicitly defines the trim point. The subset of tolerable stuck faults is obtained by referring to the trim analysis results of Section 2.4.1. Recall that the stall airspeed of the Vireo is 12 $\mathrm{m\,s^{-1}}$ and its high speed limit is 20.5 $\mathrm{m\,s^{-1}}$. From Figure 2.14, the corresponding subset of tolerable stuck faults is centered at the nominal elevon trim position and has the range $[-7, +5]\,^{\circ}$.

When one of the elevons of the Vireo experiences a stuck fault, the fault detection and isolation algorithm: (1) detects that the fault has occurred, (2) isolates whether it is left or the right elevon that has failed, and (3) switches the control law from the nominal controller (NC) to the fault-tolerant controller (FTC) (see Section 4.2.3). The FDI algorithms discussed in this chapter are designed to satisfy a set of common requirements, as described next.

6.4 Requirements

In the time intervening the fault occurrence and the controller switching, the aircraft is operating under the influence of the nominal controller. Since the nominal controller is not explicitly designed to manage the fault, loss of control (LOC) is likely to occur if the controller is not switched in a timely manner. In order to integrate notions of LOC into the FDI requirements, this section makes use of the work of Wilborn and Foster [5].

[3] The existence of such a flight condition is sufficient, but not necessary, for recovery. For instance, it may be possible to safely land the aircraft after a fault without having to fly it around an equilibrium point or flight condition.

In particular, Wilborn and Foster defined LOC using five envelopes relating to aircraft flight dynamics, aerodynamics, structural integrity, and control surface usage [5]. This section co-opts three of their five envelopes, particularly the unusual attitude (UA), the dynamic pitch control (DPC), and the dynamic roll control (DRC) envelopes. The adverse aerodynamics envelope is not chosen because the Vireo does not have sensors that measure the angle-of-attack and the angle-of-sideslip. The structural integrity envelope is not chosen because the load factor requirements of the Vireo are unknown.

It must be noted here that Wilborn and Foster defined their envelopes for commercial transport aircraft. An equivalent set of LOC definitions for small UAS does not currently exist. However, since UAS are uninhabited, it is reasonable to expect that any set of LOC definitions made in the future for small UAS will be no more stringent than Wilborn and Foster's existing definitions for commercial aircraft. For example, the Vireo has an aspect ratio of 4.46 (see Table 2.2), which is roughly one-half of that of a typical large passenger aircraft. As a result, the limit load factor of the Vireo is likely to be larger than the $[-1, +2.5]$ range stipulated by 14 CFR Part 25 for transport category aircraft. Thus the structural integrity envelope of the Vireo (and perhaps of other UAS) is also likely to be larger than Wilborn and Foster's current definition for commercial aircraft. The FDI requirements given in this section are, in this sense, conservative.

Before the fault occurrence, the nominal controller ensures that the aircraft state remains within the peripheries of the UA, DPC, and DRC envelopes. After the fault occurrence, the aircraft state is no longer guaranteed to remain within these three envelopes. Thus, the main requirement for the FDI algorithm is to switch the controller before the aircraft states exits at least one of the three envelopes. The remainder of this section formalizes this requirement.

6.4.1 Flight Envelopes

Figure 6.1 shows the definitions of the UA, DPC, and DRC envelopes used in this section. The UA envelope is directly borrowed from [5] and maps the roll angle ϕ against the pitch angle θ. The boundaries are: $\phi \in [-45, +45]\,^{\circ}$ and $\theta \in [-10, +25]\,^{\circ}$.

The DPC and the DRC envelopes are slightly modified from the ones given in [5]. In particular, it is assumed for illustrative purposes that the failure occurs in the right elevon. Thus the DPC and the DRC envelopes plot the left elevon deflection along

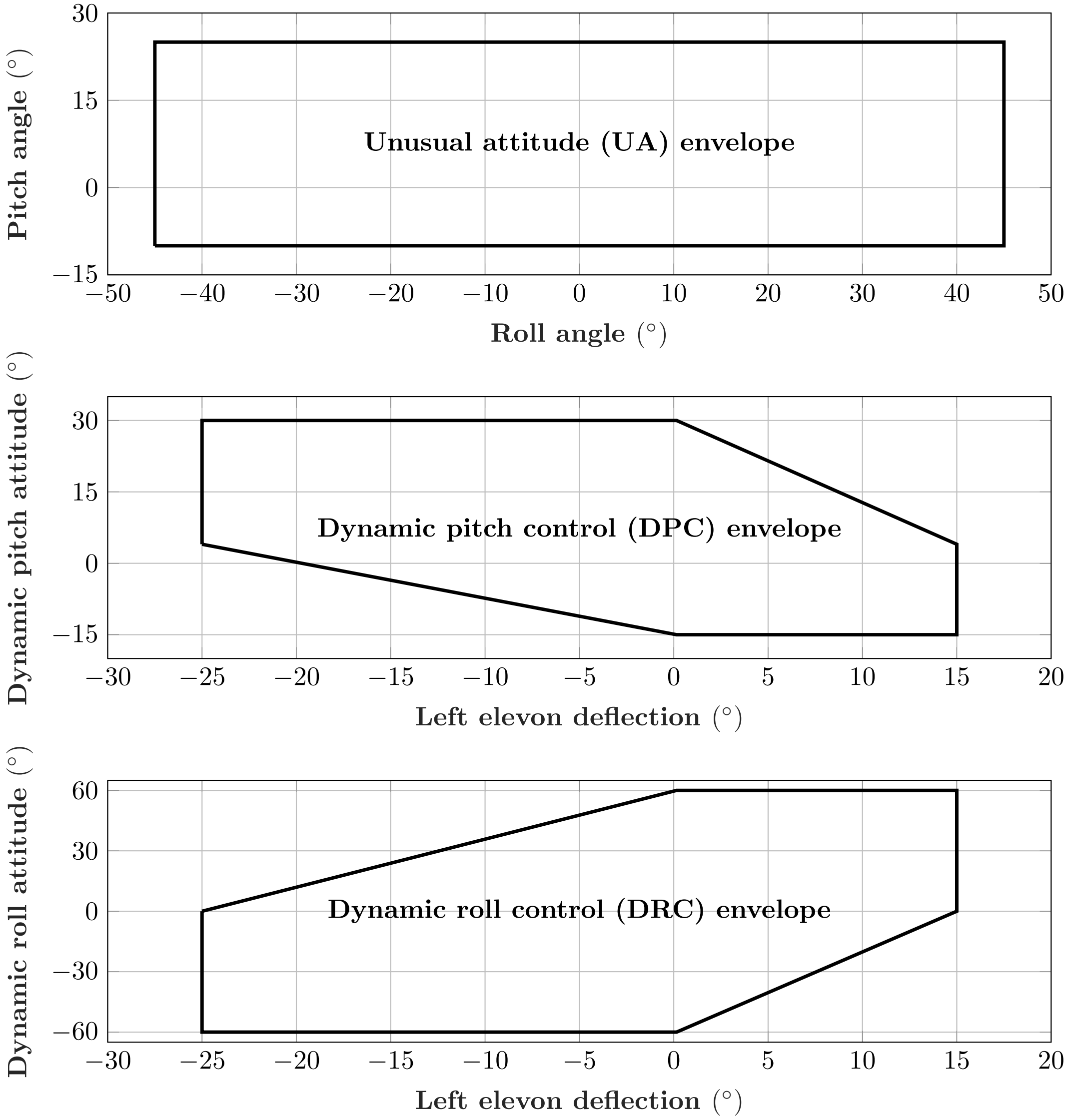


Figure 6.1: The boundaries of the unusual attitude (UA), the dynamic pitch control (DPC), and the dynamic roll control (DRC) envelopes. It is assumed for illustrative purposes that the failure occurs in the right elevon. Note that attitudes *outside* the box shown in the topmost plot are considered *unusual*.

their horizontal axes, with the limits $[-25, +15]^\circ$. These limits are 5° smaller, on either side, than the $[-30, +20]^\circ$ saturation bounds used within the nominal controller. The vertical axis of the DPC envelope is the dynamic pitch attitude, defined as the sum of the current pitch angle and its expected change after one second. Similarly, the vertical axis of the DRC envelope is the dynamic roll attitude, defined as the sum of the current roll angle and its expected change after one second. In order to account for the additional maneuverability of a small UAS such as the Vireo, the limits on the dynamic pitch and roll attitudes shown in Figure 6.1 are slightly larger than those used in [5]. In particular, the limits on the dynamic pitch attitude are $[-15, +30]^\circ$ and the limits on the dynamic roll attitude are $[-60, +60]^\circ$.

Denoting the aircraft state by x and the control input by u, the UA envelope is given by the set $\mathbb{F}_{UA} := \{(x, u) : \phi \in [-45, +45]^\circ,\ \theta \in [-10, +25]^\circ\}$. Similar set definitions follow for $\mathbb{F}_{DPC}$ and $\mathbb{F}_{DRC}$. Finally, $\mathbb{F} := \mathbb{F}_{UA} \cap \mathbb{F}_{DPC} \cap \mathbb{F}_{DRC}$ denotes the intersection.

6.4.2 Switching Time Requirement

Let t_f denote the time of fault occurrence, t_{det} denote the time of fault detection, t_{iso} denote the time of fault isolation, and t_{swi} denote the time of controller switching. Since these events occur sequentially, the relation $t_f < t_{det} \leq t_{iso} \leq t_{swi}$ holds. The second inequality includes cases where FDI filter detects and isolates the fault simultaneously. The third inequality includes cases where the control law switching immediately follows the fault isolation. This relation implicitly assumes that there are no false alarms, i.e. that the controller switching always occurs *after* the fault. While this may appear to be a strong assumption, it is inconsequential here because this section is only concerned with preventing LOC due to late controller switching.

When a fault occurs, the aircraft will deviate from its trim point. If the controller is not switched in a timely manner, the aircraft may not be recoverable. In order to set the switching time requirement, the following question is posed. What is the maximum allowable t_{swi} (denoted by $\bar{t}_{swi}$) such that: (1) the Vireo remains within the UA, the DPC, and the DRC envelopes and (2) a given FTC is able to successfully recover the aircraft? The first constraint is readily interpreted in terms of the set $\mathbb{F}$. The second constraint may also be interpreted in terms of sets. In particular, there exists some set of (x, u) for which a given FTC can successfully recover the aircraft.

The answer to the above question then follows by finding the intersection of these two sets. However, the set of recoverable aircraft states is difficult to model. (It is essentially the region-of-attraction of the fault-tolerant closed-loop.) It is therefore hypothesized that the set of recoverable states contains $\mathbb{F}$. This hypothesis is tested by initializing the nominal nonlinear model of the Vireo (Section 2.3) at each vertex of $\mathbb{F}$, injecting the maximum tolerable stuck fault in either direction, and immediately engaging the fault-tolerant controller (Chapter 7). In all the simulations, which include sensor noise and atmospheric turbulence, the FTC is able to stabilize the Vireo and track the reference command. This builds confidence that the set of recoverable states contains $\mathbb{F}$, although it may be larger. It is thus required that any stuck fault be detected, isolated, and the controller switched before the aircraft departs any of the three envelopes.

6.4.3 Computing $\bar{t}_{swi}$ via Simulations

At this stage, it is evident that the value of $\bar{t}_{swi}$ is a function of the stuck fault magnitude and the maneuver being performed at the time of the fault. For instance, a large fault magnitude may result in a quick breach of one of the three envelopes (UA, DPC, and DRC). Similarly, a fault that occurs during a maneuver may result in a quick breach of the envelopes. The effects of both of these parameters are studied using the nonlinear simulation. In particular, the tolerable stuck fault range of $[-7, +5]^\circ$ is discretized in intervals of 0.5°. In addition, the following five flight maneuvers are considered, all of which are at constant altitude and constant airspeed:

1. (FM-1) Steady, wings-level flight,

2. (FM-2) Steady, left-banked turn at a bank angle of 20°,

3. (FM-3) Steady, right-banked turn at a bank angle of 20°,

4. (FM-4) Transition from wings-level to a left-banked turn of 15°, and

5. (FM-5) Transition from wings-level to a right-banked turn of 15°.

Multiple simulations are conducted by initializing the nonlinear model of the Vireo (Section 2.3) at each of the five flight conditions described above and injecting a stuck fault in the right elevon. The nominal controller (Chapter 4) is active throughout all the simulations. For each simulation, the time that the Vireo takes to depart each of the three flight envelopes (see Figure 6.1) is computed. The lowest of the three values is thus a direct measure of $\bar{t}_{swi}$. For these computations, the dynamic pitch attitude is calculated as $(\theta + \hat{q})$, where $\hat{q}$ is the filtered pitch rate. Similarly, the dynamic roll attitude is calculated as $(\phi + \hat{p})$, where $\hat{p}$ is the filtered roll rate. Both the pitch and the roll rates are filtered using a fourth-order, low-pass Butterworth filter with a cutoff frequency of 12 $\mathrm{rad\,s^{-1}}$ in order to filter out sensor noise. The signals are filtered both forwards and backwards in time to ensure zero phase difference.

Figure 6.2 shows plots of the flight envelope departure times as a function of the stuck fault magnitude in the right elevon, for each of the five flight maneuvers listed earlier. Each plot shows the earliest time that the Vireo departs the UA, the DPC, and the DRC envelopes. The horizontal axis of each plot is normalized such that zero corresponds to the nominal trim elevon deflection, i.e. an airspeed of 15.4 $\mathrm{m\,s^{-1}}$ (see Section 2.4.1). The normalization is performed because the simulation and the real aircraft have slightly different nominal trim elevon deflections owing to variations in the CG location. For all the three envelopes, the departure times decrease as the fault magnitudes increase. This is expected because larger fault magnitudes tend to produce larger deviations in the Euler angles and the angular rates, thereby resulting in quicker departures. In addition, the departure times are very large for small fault magnitudes.

As mentioned earlier, the minimum departure time across the three envelopes is a direct measure of $\bar{t}_{swi}$. As seen in Figure 6.2, different envelopes drive this minimum, depending on the fault magnitude and the flight maneuver. For positive faults, the minimum departure time is driven by the UA and the DRC envelopes in all cases except FM-5. For FM-5, the minimum departure time is driven by the DPC envelope. For negative faults, the departure times for the three envelopes are roughly similar for fault magnitudes between -5° and -3°. However, for larger negative faults, the minimum departure time is driven by the DPC envelope. Furthermore, for fault magnitudes that are in some small neighborhood of the nominal trim elevon deflection, the Vireo never departs any of the flight envelopes over the simulation time horizon of 100 s. The size of this neighborhood varies depending on the flight maneuver, e.g. $[-3, +2]^\circ$ for FM-1.

The flight experiments, presented later in Section 6.8, consider stuck faults that are within $\pm 1^\circ$ of the nominal trim elevon deflection. In order to derive a common switching

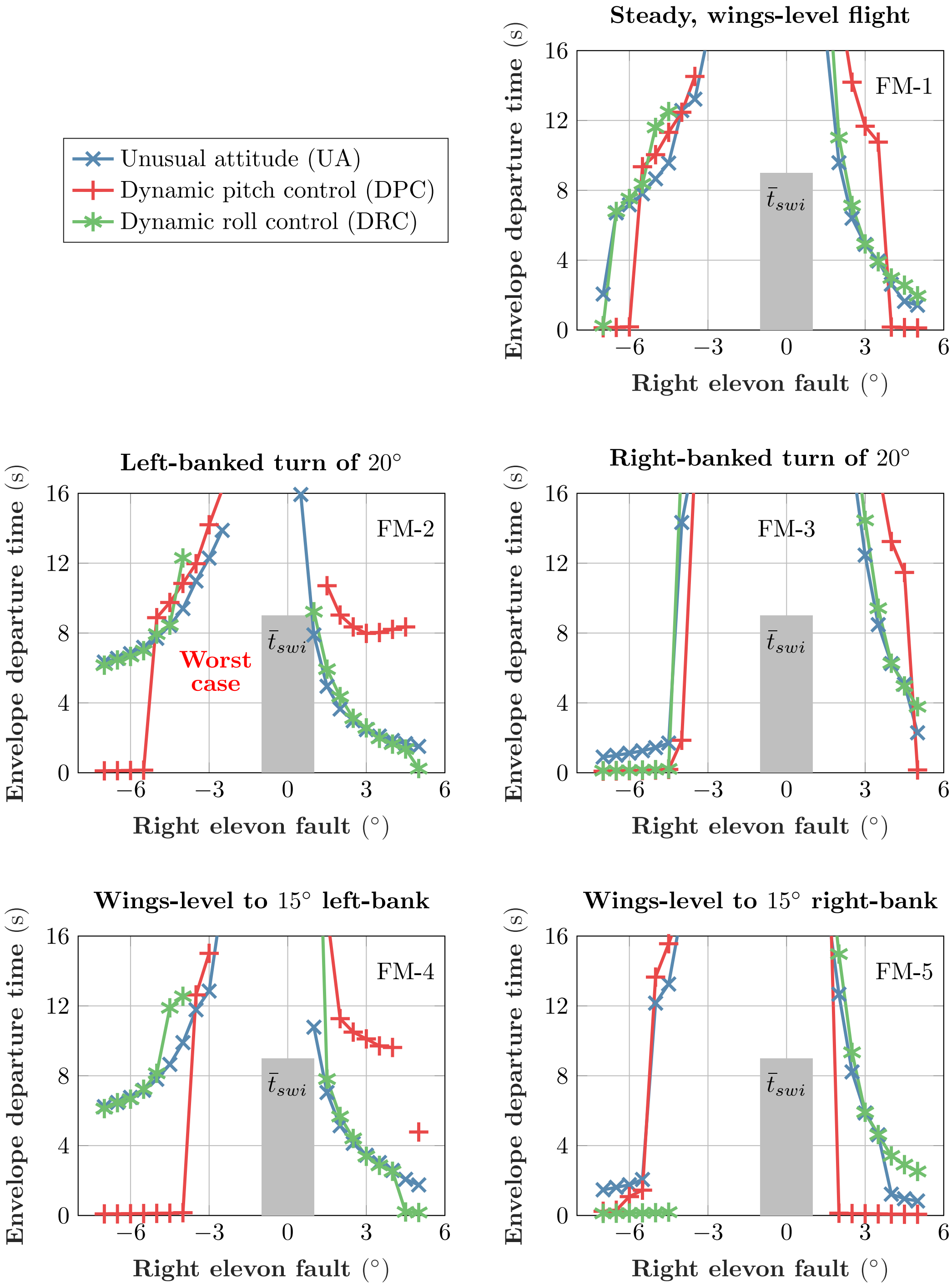


Figure 6.2: The flight envelope departure times as a function of the fault magnitude.

time requirement, the worst-case envelope departure time is considered across all the envelopes and all the flight maneuvers. For the fault interval $[-1, +1]^{\circ}$, Figure 6.2 indicates that the worst-case envelope departure time is 9 s and occurs for FM-2. Thus the maximum allowable switching time $\bar{t}_{swi}$ is 9 s, as shown by the shaded rectangles.

Finally, the approach for computing $\bar{t}_{swi}$ that is described in this section should be interpreted as a guideline. There are two main limitations to this approach. First, the peripheries of the flight envelopes shown in Figure 6.1 are not aircraft-specific. Rather, they are the result of generally accepted definitions within the aerospace industry. Further, these envelopes were originally defined for commercial aircraft [5], and have been co-opted for use with a small UAS in this chapter. Second, there may be alternatives to using the envelope departure time as a basis for setting the controller switching time requirement. The large discontinuities seen in Figure 6.2 point to one of the drawbacks of this approach. It may be worth exploring approaches that, in addition to the departure time, consider the distance of $(x(t), u(t))$ from the periphery of $\mathbb{F}$.

6.5 FDI Architecture

The fault detection and isolation (FDI) algorithm has two stages, as shown in Figure 6.3. The residual generation stage uses some subset of the controller reference commands r, the sensor measurements y, the navigation state estimates $\hat{x}$, and the control commands u to generate a residual e_*[4]. These four types of signals $(r, y, \hat{x}, u)$ are either the inputs or the outputs of the nominal controller. Thus, they are readily available during runtime. The residual evaluation stage uses the residual e_* and the threshold T_* to detect the fault. In particular, a fault is declared if $|e_*| \geq T_*$. The thresholds may be state-dependent or time-varying. In addition, they may be combined with up-down counters to decrease the probability of false alarms. This chapter uses constant thresholds for simplicity. As such, the threshold is an important parameter that controls the trade-off between the rates of false alarms and missed detections. The threshold is selected to ensure that the FDI algorithm does not declare false alarms when applied to flight data wherein no faults are injected (see Section 6.8.2). Depending on the method, the residual evaluation stage may also isolate the fault. The *fault information* produced by

[4] The asterisk is used here as a placeholder. The remainder of this section will use specific variables.

the residual evaluation stage is used to reconfigure the guidance and the control laws, as described previously in Section 4.2.

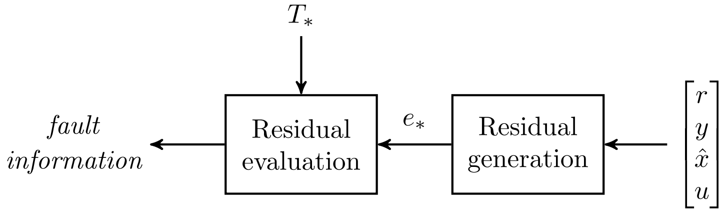


Figure 6.3: The fault detection and isolation algorithm consists of two stages.

Two designs of the residual generation stage are presented next. The first is parity-space based (Design A) and only performs fault detection. The second is observer-based (Design B) and performs both fault detection and isolation. Both the designs use the same residual evaluation principle of threshold crossing detection, albeit with different values for the threshold. Overall, Design A serves as the baseline for fault detection and Design B seeks to improve upon this baseline.

6.6 Residual Generation: Parity Space-Based (Design A)

Design A is pictured in Figure 6.4 and consists of two main blocks. The F_A block uses the roll rate p (measured by the IMU) and the aileron command δ_{ac} (produced by the nominal roll attitude controller K_A; see Figure 4.6) to generate a roll rate residual $\bar{e}_p$. The F_A block comprises models describing the open-loop lateral-directional aircraft dynamics G_{lat} and the aileron actuator G_a. Equation (4.2) provides the state-space representation of G_{lat} at the nominal trim airspeed of 15.4 $\mathrm{m\,s^{-1}}$. The output of G_{lat} is the model-predicted roll rate $\hat{p}$, which is used to compute the roll rate residual as $\bar{e}_p = \hat{p} - p$. The actuator model G_a incorporates not only the second-order transfer function identified in Section 3.2.4, but also the servo position limits of $[-30, +20]^\circ$, the rate limits of $\pm 338^\circ\,\mathrm{s^{-1}}$, and the time delay of 0.05 s. Thus the output $\bar{\delta}_a$ of G_a is a prediction of the virtual aileron position, based purely on the aileron command. The R_A block filters $\bar{e}_p$ to produce the final residual e_p, which is subsequently sent to the residual evaluation stage. Note that Design A uses the lateral-directional model over the longitudinal model owing to its higher modeling accuracy (see Section 3.3.7).

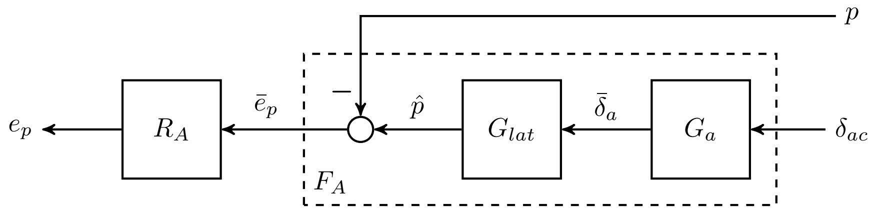


Figure 6.4: The architecture of Design A of the FDI algorithm.

6.7 Residual Generation: Observer-Based (Design B)

Design B is observer-based and generates two sets of estimates of the left and right elevon positions (δ_l, δ_r). In the absence of a fault, these two sets of estimates are nearly equal and differ only at high frequencies, i.e. due to noise. In the presence of a fault, one set continues to be a reliable estimate of the actual elevon positions, while the other set exhibits a low frequency divergence. The difference between the two sets of estimates serves as the residual. This residual is filtered and evaluated using thresholds.

Design B is pictured in Figure 6.5 and consists of two main blocks. The F_B block comprises the observer F and the model describing the elevon actuator G_a. As with Design A, G_a is the nonlinear actuator model that incorporates the servo position limits, the rate limits, and the time delay. The inputs to $\operatorname{diag}(G_a, G_a)$ are the left and right elevon deflection commands $(\delta_{lc}, \delta_{rc})$ generated by the nominal controller (see Figure 4.4). The outputs $\left(\bar{\delta}_l, \bar{\delta}_r\right)$ of $\operatorname{diag}(G_a, G_a)$ are the first set of estimates of the left and right elevon positions. Since they account only for the actuator model, they are reliable only in the absence of faults.

The observer F uses the commanded roll attitude ϕ_{cmd}, the estimated roll attitude ϕ, and the angular rates p and q to generate the second set of estimates of the left and right elevon positions $\left(\hat{\delta}_l, \hat{\delta}_r\right)$. Since the observer accounts for the closed-loop aircraft dynamics (explained shortly), $\hat{\delta}_l$ and $\hat{\delta}_r$ are reliable even in the presence of a stuck elevon fault. The output $(\bar{e}_l, \bar{e}_r)$ of F_B is the difference between the two sets of estimates and thus serves as the raw residual. The $\operatorname{diag}(R_B, R_B)$ block filters $(\bar{e}_l, \bar{e}_r)$ to produce the final residual (e_l, e_r), which is subsequently sent to the residual evaluation stage.

The observer F comprises two other filters, as shown in Figure 6.6. F_{lat} uses $[\phi_{cmd}, \phi, p]^T$ to estimate the position of the virtual aileron $\hat{\delta}_a$. Thus F_{lat} is designed

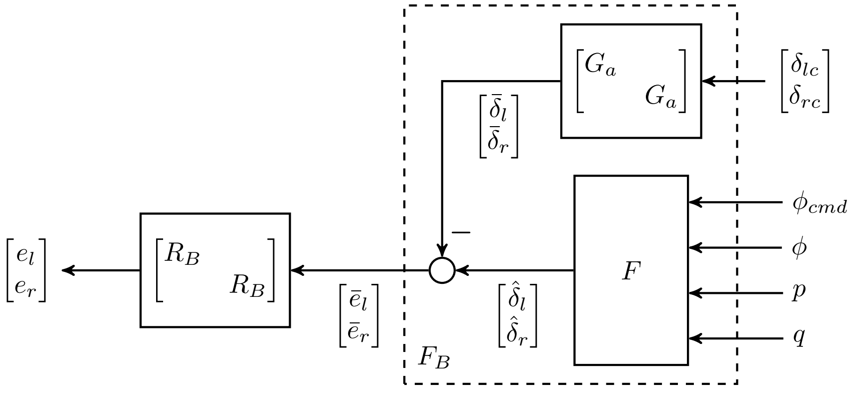

$\begin{bmatrix} G_a & \\ & G_a \end{bmatrix}$
$\begin{bmatrix} \delta_{lc} \\ \delta_{rc} \end{bmatrix}$
$\begin{bmatrix} \bar{\delta}_l \\ \bar{\delta}_r \end{bmatrix}$
F
ϕ_{cmd}
ϕ
p
q
$-$
$\begin{bmatrix} e_l \\ e_r \end{bmatrix}$
$\begin{bmatrix} R_B & \\ & R_B \end{bmatrix}$
$\begin{bmatrix} \bar{e}_l \\ \bar{e}_r \end{bmatrix}$
$\begin{bmatrix} \hat{\delta}_l \\ \hat{\delta}_r \end{bmatrix}$
F_B

Figure 6.5: The architecture of Design B of the FDI algorithm.

using the lateral-directional aircraft model G_{lat}, whose state-space representation is given by Equations (3.18), (3.19), and (3.20). F_{long} uses q to estimate the position of the virtual elevator $\hat{\delta}_e$. Recall from Section 3.3.5 that the phugoid mode is not experimentally characterized on the Vireo. Thus F_{long} is designed using only the short period model G_{sp}. The state-space representation of G_{sp} is obtained from Equations (3.15) and (3.16) after truncating the states u and θ. The transformation block $T_{lr\leftarrow ea}$ converts $\hat{\delta}_a$ and $\hat{\delta}_e$ into the elevon position estimates $\left(\hat{\delta}_l, \hat{\delta}_r\right)$. In addition to G_{lat} and G_{sp}, the designs of F_{lat} and F_{long} account for the nominal controller described in Chapter 4.

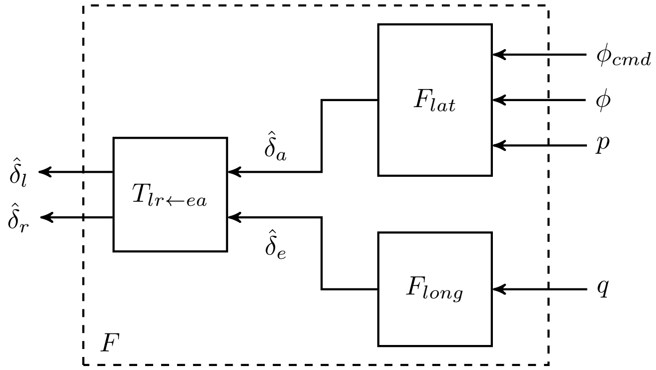


Figure 6.6: The observer F comprises two other observers: F_{lat} and F_{long}.

The exclusion of the phugoid modal dynamics in designing F_{long} slightly impacts the accuracy of the elevon position estimates $\left(\hat{\delta}_l, \hat{\delta}_r\right)$. As such, this is a shortcoming of the current design that is managed within the leeway afforded by the thresholds. It may be remedied by modeling the phugoid modal dynamics to a higher degree of accuracy.

Finally, as discussed previously in Section 2.4.2, the aircraft dynamics are described by LTI models, which are parametrized by the airspeed V. These models are not perfectly known and have some degree of uncertainty. Thus the subsequent sections treat G_{lat} and G_{sp} as *nominal* gridded, linear parameter-varying (LPV) systems that are interconnected with block structured *uncertainties*. In this general setting, F_{lat} and F_{long} are synthesized by invoking the concepts discussed in Section 5.4.1.

6.7.1 LPV Modeling

The trim analysis conducted in Section 2.4.1 yielded the stall airspeed of the Vireo as 12 $\mathrm{m\,s^{-1}}$ and the high speed limit as 20 $\mathrm{m\,s^{-1}}$. Thus, in order to obtain LPV representations of G_{lat} and G_{sp}, 20 grid points are chosen in the airspeed domain $[12, 20]$ $\mathrm{m\,s^{-1}}$. In addition, past flight data indicates that the rate-of-variation of the airspeed is bounded by ±8 $\mathrm{m\,s^{-2}}$ during typical flight maneuvers. These rate bounds restrict the airspeed trajectories to only those that are realistic. Moreover, when incorporated during the synthesis of F_{lat} and F_{long}, they decrease the conservatism of the result. The nonlinear model of the Vireo (Section 2.3) is trimmed and linearized (Section 2.4) at the steady, wings-level, constant altitude, and constant airspeed flight condition corresponding to each grid point. The resulting collections of LTI models, along with the airspeed domain, constitute the LPV representations of G_{lat} and G_{sp}. LPVTools (a Matlab toolbox) implements algorithms for the modeling, analysis, and synthesis of LPV systems [42,43].

Figure 6.7 shows the Bode diagrams of the open-loop plant at each grid point in the airspeed domain $[12, 20]$ $\mathrm{m\,s^{-1}}$. Both the aileron-to-roll rate response (left) and the elevator-to-pitch rate response[5] (right) vary to some degree between the stall speed of 12 $\mathrm{m\,s^{-1}}$ (solid line) and the high speed limit of 20 $\mathrm{m\,s^{-1}}$ (dashed line). In particular, the frequencies of both the dutch roll mode and the short period mode increase with increasing airspeed. This is consistent with the observations made in Section 2.4.2. Section 6.8 later demonstrates using flight data that FDI filters that account for these airspeed variations (i.e. LPV) perform better than those that do not (i.e. LTI).

[5] The elevator-to-pitch rate response does not exhibit the peak associated with the phugoid mode because this section only uses the second-order short period model.

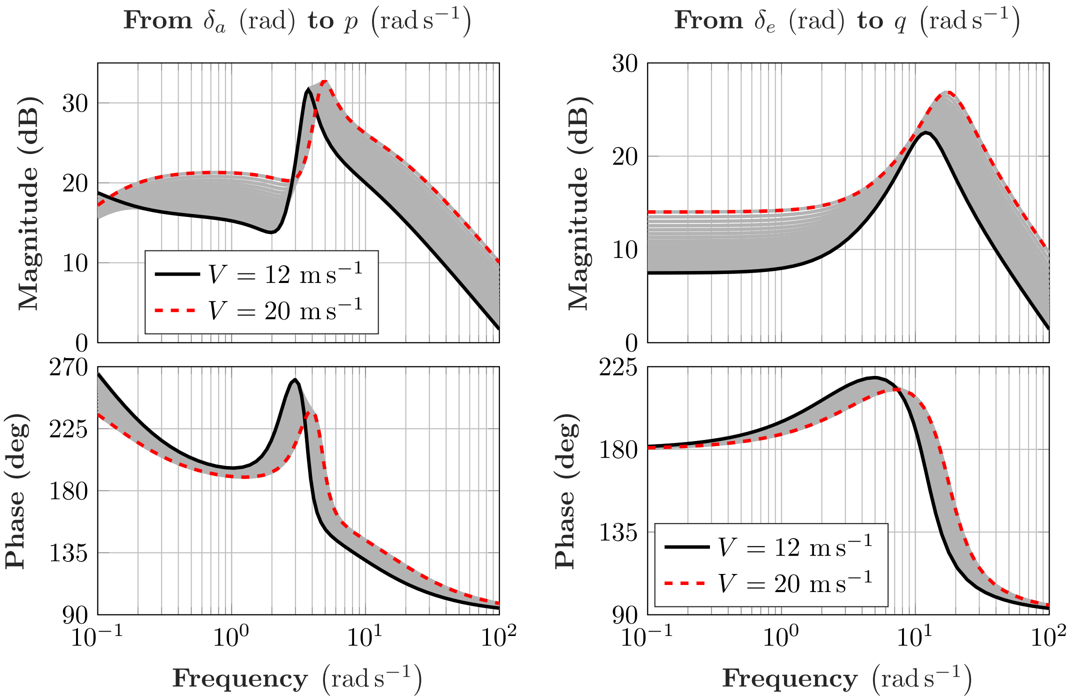


Figure 6.7: The Bode diagrams of the open-loop plant at each grid point in the airspeed domain $[12, 20]\ \mathrm{m\,s^{-1}}$. Left: The lateral-directional model from the aileron δ_a to the roll rate p. Right: The short period model from the elevator δ_e to the pitch rate q.

6.7.2 Synthesis Framework

F_{lat} and F_{long} are designed using the generic block diagram shown in Figure 6.8. This block diagram corresponds to the output estimation problem introduced previously as Figure 5.4 in Section 5.4.1. The specific block diagrams for the synthesis of F_{lat} and F_{long} are obtained by replacing $\star$ in Figure 6.8 with the appropriate subscript, as explained shortly. $H_\star$ is a nominal LPV plant that includes either G_{lat} or G_{sp}, the actuator model, the nominal controller, and any synthesis weighting functions. Δ is a block-structured perturbation that includes any uncertainties in the aircraft model. The input-output behavior of Δ is described using IQCs, as explained in Chapter 5. Further, d denotes the generalized disturbances, y denotes the measurements sent to the filter, $\delta_\star$ denotes the actual aileron or elevator position, and $\hat{\delta}_\star$ denotes the estimated aileron or elevator position. $W_\star$ filters the estimation error $e_\star = \hat{\delta}_\star - \delta_\star$ over some desired frequency range.

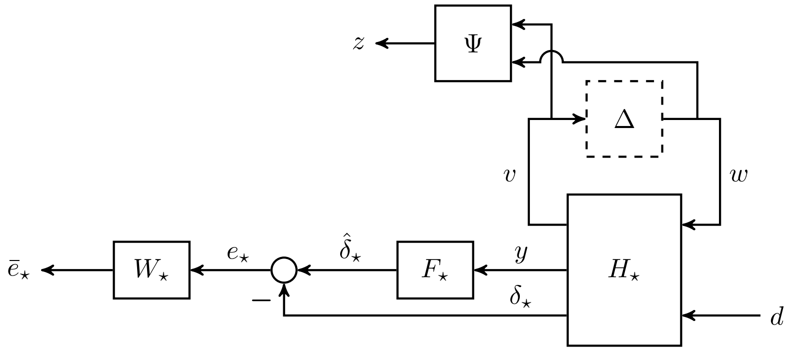


Figure 6.8: F_{lat} and F_{long} are synthesized by formulating an output estimation problem.

The synthesis objective is to find the filter $F_\star$ that yields the smallest possible upper bound on the worst-case gain from d to $\bar{e}_\star$. Since the design model $\mathcal{F}_u(H_\star, \Delta)$ is *uncertain and LPV*, $F_\star$ is synthesized using Theorem 2 of Chapter 5. In order to motivate the uncertain, LPV design model, three other design models are considered by placing restrictions on the model uncertainty and/or the airspeed domain. Table 6.1 lists each design model, the corresponding synthesis method, and an identifier for the resulting filter. For example, a design model that accounts for the aircraft dynamics at a fixed airspeed of 15.4 $\mathrm{m\,s^{-1}}$, along with some model uncertainty, results in a robust-LTI filter. Sections 6.7.5 and 6.8 later demonstrate that, by virtue of making the fewest assumptions, the robust-LPV filter that results from the uncertain, LPV design model outperforms the other filters. The particular choices of $H_\star$ and Δ for the uncertain, LPV design model are explained next.

Table 6.1: F_{lat} and F_{long} are synthesized using four different design models.

Aircraft model	**Uncertainty**	**Method**	**Resulting filter**
LTI at 15.4 $\mathrm{m\,s^{-1}}$	$\Delta = 0$	H_∞ synthesis [154]	Nominal-LTI
LTI at 15.4 $\mathrm{m\,s^{-1}}$	$\Delta \neq 0$	Theorem 2	Robust-LTI
LPV over $[12, 20]$ $\mathrm{m\,s^{-1}}$	$\Delta = 0$	LPV synthesis [100]	Nominal-LPV
LPV over $[12, 20]$ $\mathrm{m\,s^{-1}}$	$\Delta \neq 0$	Theorem 2	Robust-LPV

6.7.3 Lateral-Directional Filter F_{lat}

To synthesize the later-directional filter, Figure 6.8 is used with the systems H_{lat}, F_{lat}, and W_{lat}, and the signals δ_a, $\hat{\delta}_a$, e_a, and $\bar{e}_a$. The measurement signal is $y = [\phi_{cmd}, \phi, p]^T$.

Generalized plant

Figure 6.9 shows the generalized plant that is used for synthesizing F_{lat}. The dashed box shows the contents of $\mathcal{F}_u(H_{lat}, \Delta)$, which includes the nominal lateral-directional aircraft model G_{lat}, the actuator model G_a^L (see Section 3.3.9), and the nominal roll attitude controller K_A (see Section 4.4). Ideally, the uncertainty in the lateral-directional aircraft model should be quantified in terms of the uncertainties in the underlying aircraft parameters, e.g. the moments of inertia and the aerodynamic coefficients. However, since Chapter 2 did not model these parametric uncertainties, the nominal plant G_{lat} is assumed to be affected by multiplicative uncertainty at its input. In particular, Δ is assumed to be norm-bounded and LTI such that $\|\Delta\|_\infty \leq 1$. The uncertain plant is thus given by $G_{lat}(1 + \Delta W_\Delta)$.

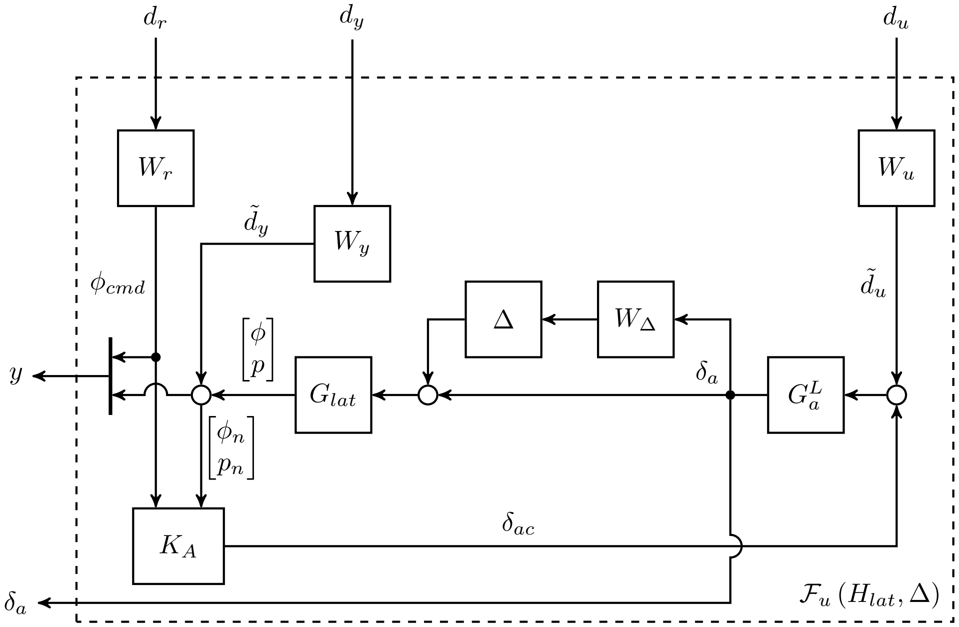


Figure 6.9: The generalized plant that is used for synthesizing F_{lat}.

The input δ_a to the uncertain plant is the quantity to be estimated. The feedback loop involving G_a^L, G_{lat}, and K_A represents the closed-loop lateral-directional aircraft dynamics (discussed previously as Figure 4.6). This feedback loop is affected by: the disturbance $\tilde{d}_u$ at the input of G_a^L, the disturbance $\tilde{d}_y$ at the output of G_{lat}, and the reference command ϕ_{cmd}. The performance weights W_u, W_y, and W_r relate the disturbances $\tilde{d}_u$, $\tilde{d}_y$, and ϕ_{cmd} to their respective normalized counterparts d_u, d_y, and d_r.

Weight selection

The weighted, uncertain closed-loop shown in Figure 6.8 has the input $d = (d_u, d_y, d_r)$ and the output $\bar{e}_a$. The synthesis is an iterative process that involves weight selection and tuning. Table 6.2 lists the final values of all the weights, along with their interpretations. For instance, the weights W_u, W_y, and W_r are selected as the typical disturbances expected in the signals $\tilde{d}_u$, $\tilde{d}_y$, and ϕ_{cmd}, respectively.

Table 6.2: The final weights selected for synthesizing F_{lat}.

Weight	**Final value**	**Weight interpretation:**
W_u	$3\,(\pi/180)$	Typical aileron disturbance.
W_y	$\begin{bmatrix} 6 & 0 \\ 0 & 6 \end{bmatrix} (\pi/180)$	Typical disturbances in the roll angle and the roll rate.
W_r	$30\,(\pi/180)$	Typical roll angle command.
W_Δ	$\frac{s+3.924}{s+392.4}$	Shapes the uncertainty in G_{lat} across frequency.
W_{lat}	$\frac{s+4}{1.122s+0.07113}$	Inverse of the desired sensitivity function from $-\delta_a$ to e_a.

The weight W_Δ shapes the uncertainty in G_{lat} across frequency. In general, the model uncertainty is low at low frequencies and high at high frequencies. For this problem, it is assumed that the uncertainty in G_{lat} is 1% at frequencies below the dutch roll mode $\left(4.1 \text{ rad}\,\text{s}^{-1}\right)$ and increases to 100% at high frequencies. Thus W_Δ is selected as shown in Figure 6.10, where the numbers within the parentheses specify the particular levels of uncertainty at the natural frequencies of the dutch roll mode, the roll subsidence mode, and the actuator.

In order to select W_{lat} (the filter on e_a in Figure 6.8), let S_a and $\bar{S}_a$ denote the sensitivity function $-\delta_a \rightarrow e_a$ and its upper bound, respectively. Overall, it is more

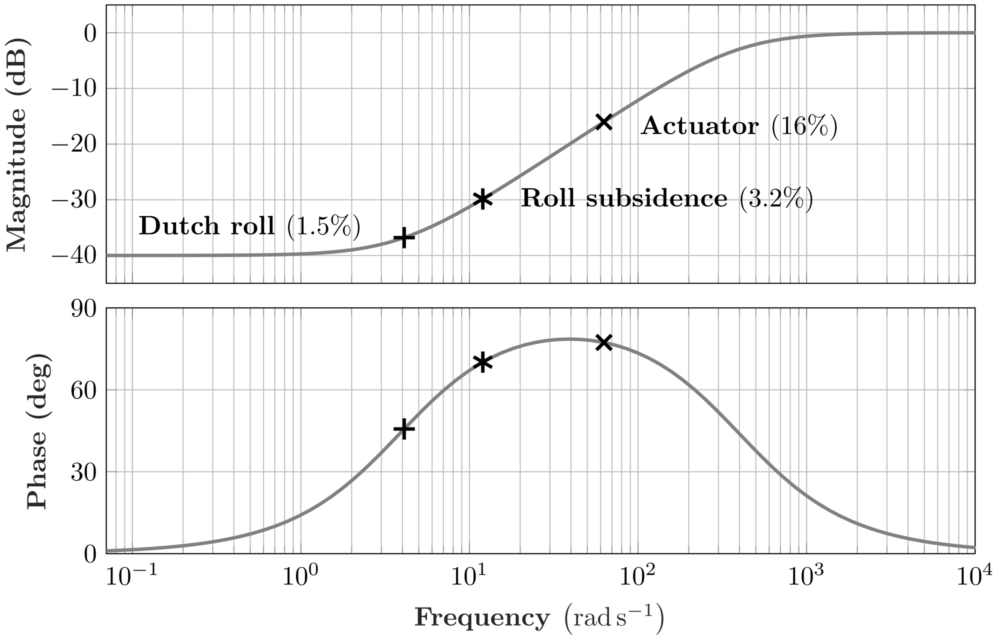


Figure 6.10: The Bode diagram of the weight W_Δ that shapes the uncertainty in G_{lat}.

important to minimize e_a at low frequencies as compared to high frequencies. Thus $\bar{S}_a$ is specified as a first-order transfer function with a DC gain of −35 dB, a high-frequency gain of 1 dB, and a bandwidth of 4 rad s^{-1}. The bandwidth here refers to the −3 dB point with respect to the high frequency gain and corresponds to the natural frequency of the dutch roll mode. Subsequently W_{lat} is selected as $\bar{S}_a^{-1}$.

Incorporation of rate bounds and IQCs

Recall that Theorem 2 of Chapter 5 is invoked to synthesize F_{lat}. Although this theorem is worded for cases where the scheduling parameters have unbounded rates-of-variation, it is extended to account for the finite rate bounds specified for the airspeed in Section 6.7.1. One of the consequences of this extension is that the matrices X and Z (see Theorem 2) become continuous functions of the parameter, in this case the airspeed V. The implementation of the corresponding LMIs requires X and Z to be expressed using basis functions (see, for example, Section 5 of [100]). This section uses the bases $\left(1, V, V^2\right)$ so that $X\left(V\right) = X_0 + VX_1 + V^2X_2$ and $Z\left(V\right) = Z_0 + VZ_1 + V^2Z_2$.

In addition, since Δ is a LTI uncertainty with $\|\Delta\|_\infty \leq 1$, it satisfies all IQCs defined by multipliers of the form:

$$\Pi\left(j\omega\right) = \begin{bmatrix} x\left(j\omega\right) I_{n_v} & 0 \\ 0 & -x\left(j\omega\right) I_{n_w} \end{bmatrix}, \tag{6.1}$$

where $x\left(j\omega\right) \geq 0$ is a bounded measurable function [99].

To obtain time-domain IQCs, Π is factorized as $\Pi\left(j\omega\right) = \Psi\left(j\omega\right)^\sim M \Psi\left(j\omega\right)$, where

$$\Psi\left(j\omega\right) = \begin{bmatrix} \psi\left(j\omega\right) & 0 \\ 0 & \psi\left(j\omega\right) \end{bmatrix} \quad \text{and} \quad M = \begin{bmatrix} \bar{M} & 0 \\ 0 & -\bar{M} \end{bmatrix}. \tag{6.2}$$

Equations (6.1) and (6.2) together imply that $x\left(j\omega\right) I_{n_v} = \psi\left(j\omega\right)^\sim \bar{M}\psi\left(j\omega\right) \geq 0$.

If $x\left(j\omega\right)$ is factorized as $\psi_x\left(j\omega\right)^\sim M_x \psi_x\left(j\omega\right)$, then

$$\begin{aligned} x\left(j\omega\right) I_{n_v} &= x\left(j\omega\right) \otimes I_{n_v} & (6.3) \\ &= \left(\psi_x\left(j\omega\right)^\sim M_x \psi_x\left(j\omega\right)\right) \otimes I_{n_v} & (6.4) \\ &= \left(\psi_x\left(j\omega\right) \otimes I_{n_v}\right)^\sim \left(M_x \otimes I_{n_v}\right)\left(\psi_x\left(j\omega\right) \otimes I_{n_v}\right) & (6.5) \\ &= \psi\left(j\omega\right)^\sim \bar{M}\psi\left(j\omega\right). & (6.6) \end{aligned}$$

Theorem 2 is implemented by choosing $\psi\left(j\omega\right) = \left(\psi_x\left(j\omega\right) \otimes I_{n_v}\right)$ and $\bar{M} = \left(M_x \otimes I_{n_v}\right)$, where $\psi_x\left(j\omega\right) = \left[\psi_1\left(j\omega\right), \psi_2\left(j\omega\right), \ldots, \psi_{n_\psi}\left(j\omega\right)\right]^T$ is a vector of user-defined scalar basis functions and M_x is a symmetric matrix that is determined during the optimization.

The uncertainty in this problem is scalar, i.e. $n_v = n_w = 1$. This section uses two basis functions for $\psi_x\left(j\omega\right)$: $\psi_1\left(j\omega\right) = 1$ and $\psi_2\left(j\omega\right) = \frac{1}{j\omega + 0.031}$, where the second basis function is the result of trial and error.

Synthesis results

The synthesized filter F_{lat} is a quadratically stable LPV system (see Definition 3) that is scheduled by the airspeed V and its time derivative $\dot{V}$. In order to assess its performance, the *nominal, unweighted closed-loop* is constructed by setting all the weights to unity and Δ to zero in Figures 6.8 and 6.9. Thus this closed-loop consists only of G_{lat}, K_A, G_a^L, and F_{lat}. Figure 6.11 shows the Bode diagrams of the sensitivity functions S_a from

$-\delta_a$ to e_a at each grid point in the airspeed domain along with the sensitivity bound $\bar{S}_a$ that was described earlier. The plot indicates that F_{lat} satisfies the desired sensitivity response at each point in the domain.

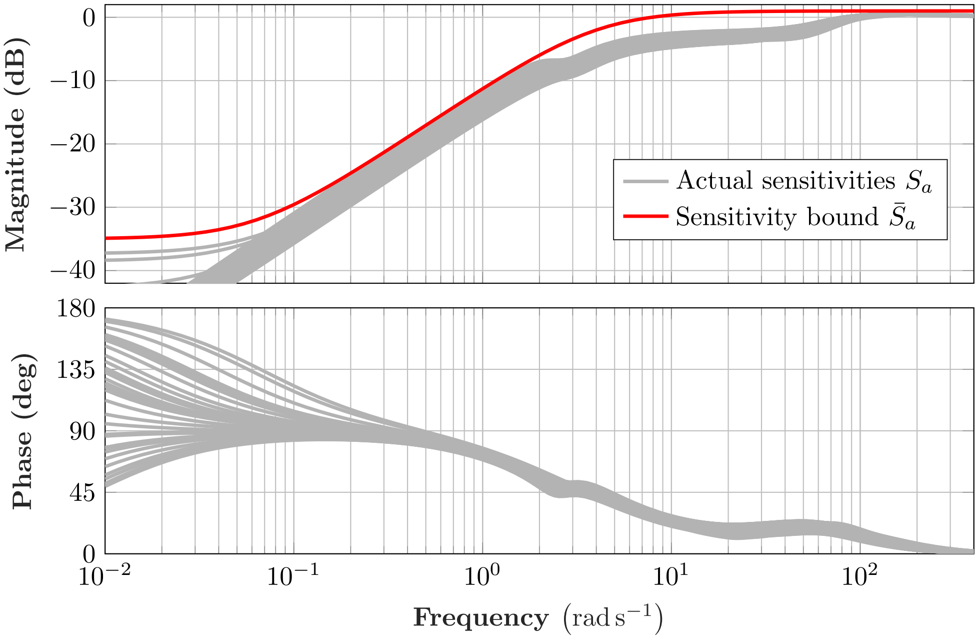


Figure 6.11: The Bode diagrams of the sensitivity functions S_a from $-\delta_a$ to e_a at each grid point and their upper bound $\bar{S}_a$.

The time-domain performance of the filter is evaluated using step responses as shown in Figure 6.12. The left-side of this figure shows an ensemble of filter responses to a 10° step change in the aileron position for multiple constant airspeed trajectories that span the domain of $[12, 20]$ $\mathrm{m\,s^{-1}}$. This plot indicates that the filter responds slightly faster at higher airspeeds. For example, the 90%-rise time of the filter output is 0.6 s at $V = 20$ $\mathrm{m\,s^{-1}}$ and 0.9 s at $V = 12$ $\mathrm{m\,s^{-1}}$. With the exception of the rise time, however, the filter responses are largely similar over the entire domain.

The upper-right corner of Figure 6.12 shows the filter response to a 10° step change in the aileron position δ_a that occurs at $t = 0.1$ s. This step response corresponds to the sinusoidal airspeed trajectory $V(t) = 16 + 4\sin\left(\frac{2\pi}{\pi}t + \frac{2\pi}{3}\right)$ $\mathrm{m\,s^{-1}}$ that is shown in the lower-right corner. This particular trajectory has a time period of π s, spans the entirety of the airspeed domain $[12, 20]$ $\mathrm{m\,s^{-1}}$, and achieves a maximum rate-of-variation

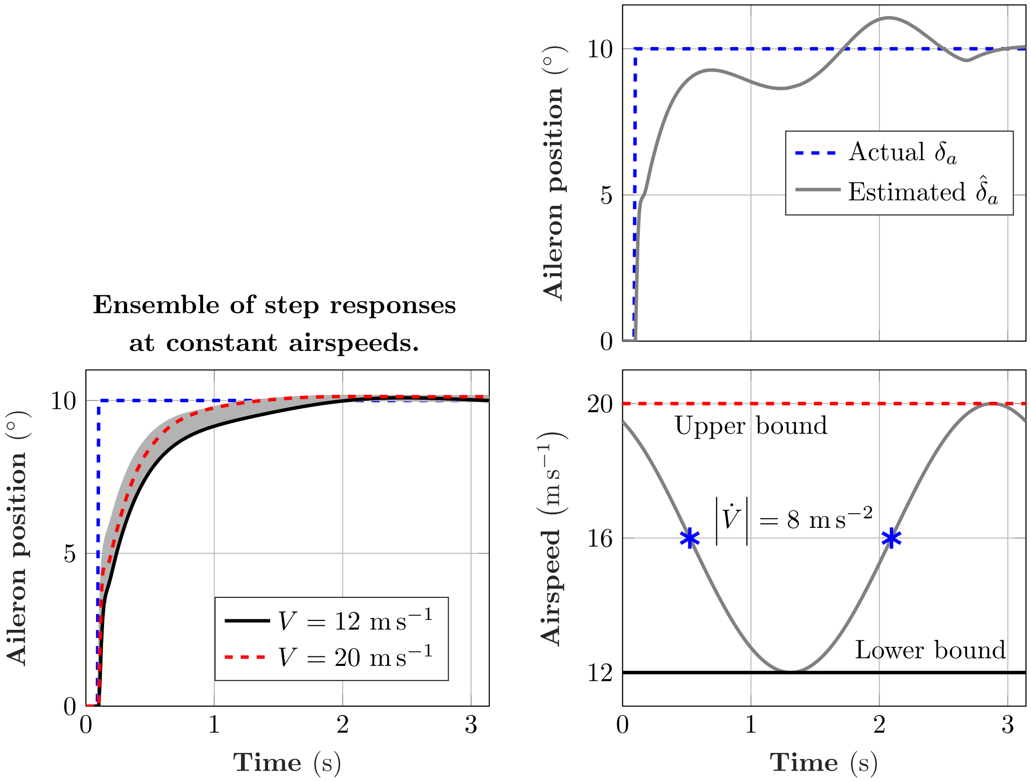


Figure 6.12: Left: An ensemble of filter responses to a 10° step change in the aileron position for multiple constant airspeed trajectories. Right: The filter response to a 10° step change in the aileron position (top) for a sinusoidal airspeed trajectory (bottom).

of ±8 m s^{-2} (marked by $*$). The estimated aileron position $\hat{\delta}_a$ exhibits a 90%-rise time of 0.53 s, an overshoot of 10%, and converges to 10° by $t = 3$ s. Note the correlation between the instantaneous rate-of-variation of the airspeed and the filter response. For example, the overshoot in the filter response (around 2 s) does not occur until after one full oscillation because that is when the airspeed reaches its peak rate-of-variation.

6.7.4 Longitudinal Filter F_{long}

To synthesize the longitudinal filter, Figure 6.8 is used with the systems H_{long}, F_{long}, and W_{long}, and the signals δ_e, $\hat{\delta}_e$, e_e, and $\bar{e}_e$. The measurement signal is $y = q$.

Generalized plant

Figure 6.13 shows the generalized plant that is used for synthesizing F_{long}. The dashed box shows the contents of $\mathcal{F}_u\left(H_{long}, \Delta\right)$, which includes the nominal short period model G_{sp}, the actuator model G_a^L (see Section 3.3.9), and the nominal pitch damper K_{PD} (see Section 4.4). Since Chapter 2 did not model the parametric uncertainties in the short period model, the nominal plant G_{sp} is assumed to be affected by multiplicative uncertainty at its input. In particular, Δ is assumed to be norm-bounded and LTI such that $\|\Delta\|_\infty \leq 1$. The uncertain plant is thus given by $G_{sp}\left(1 + \Delta W_\Delta\right)$.

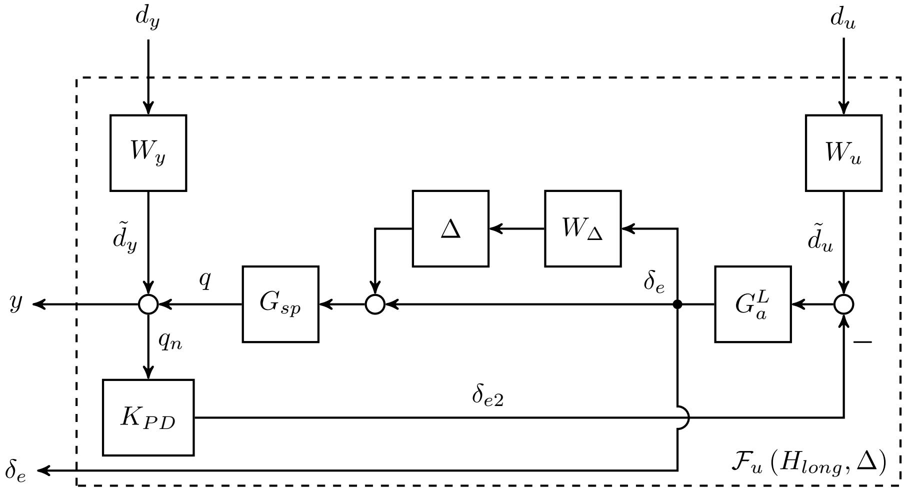


Figure 6.13: The generalized plant that is used for synthesizing F_{long}.

The input δ_e to the uncertain plant is the quantity to be estimated. The feedback loop involving G_a^L, G_{sp}, and K_{PD} represents the closed-loop short period modal dynamics (the innermost loop of Figure 4.5). This feedback loop is affected by: the disturbance $\tilde{d}_u$ at the input of G_a^L and the disturbance $\tilde{d}_y$ at the output of G_{sp}. The performance weights W_u and W_y relate the disturbances $\tilde{d}_u$ and $\tilde{d}_y$ to their respective normalized counterparts d_u and d_y.

Weight selection

The weighted, uncertain closed-loop shown in Figure 6.8 has the input $d = (d_u, d_y)$ and the output $\bar{e}_e$. The synthesis is an iterative process that involves weight selection and tuning. Table 6.3 lists the final value of all the weights, along with their interpretations. For instance, the weights W_u and W_y are selected as the typical disturbances expected in the signals $\tilde{d}_u$ and $\tilde{d}_y$, respectively.

Table 6.3: The final weights selected for synthesizing F_{long}.

Weight	**Final value**	**Weight interpretation:**
W_u	$4\,(\pi/180)$	Typical elevator disturbance.
W_y	$10\,(\pi/180)$	Typical pitch rate disturbance.
W_Δ	$\frac{s^2+53.43s+213.5}{s^2+436s+427}$	Shapes the uncertainty in G_{sp} across frequency.
W_{long}	$\frac{s+14.5}{1.122s+9.149}$	Inverse of the desired sensitivity function from $-\delta_e$ to e_e.

The weight W_Δ shapes the uncertainty in G_{sp} across frequency. In general, the model uncertainty is low at low frequencies and high at high frequencies. However, recall that the longitudinal system identification (Section 3.3.5) characterized the elevator-to-pitch rate response only near the short period mode. Consequently, it is assumed that the uncertainty in G_{sp} is 50% at frequencies below the phugoid mode $\left(0.87 \text{ rad}\,\text{s}^{-1}\right)$, decreases to around 12% near the short period mode $\left(14.5 \text{ rad}\,\text{s}^{-1}\right)$, and increases to 100% at high frequencies. Thus W_Δ is selected as shown in Figure 6.14, where the numbers within the parentheses specify the particular levels of uncertainty at the natural frequencies of the phugoid mode, the short period mode, and the actuator.

In order to select W_{long} (the filter on e_e in Figure 6.8), let S_e and $\bar{S}_e$ denote the sensitivity function $-\delta_e \to e_e$ and its upper bound, respectively. Overall, it is more important to minimize e_e at low frequencies as compared to high frequencies. Thus $\bar{S}_e$ is specified as a first-order transfer function with a DC gain of -4 dB, a high-frequency gain of 1 dB, and a bandwidth of 14.5 $\text{rad}\,\text{s}^{-1}$. Note that the DC gain of $\bar{S}_e$ is specified to a much higher value when compared to $\bar{S}_a$ in the previous section. This is because G_{sp} only includes the short period modal dynamics and thus lower DC gains cannot be achieved in S_e. The bandwidth here refers to the -3 dB point with respect to the high frequency gain and corresponds to the natural frequency of the short period mode. Subsequently W_{long} is selected as $\bar{S}_e^{-1}$.

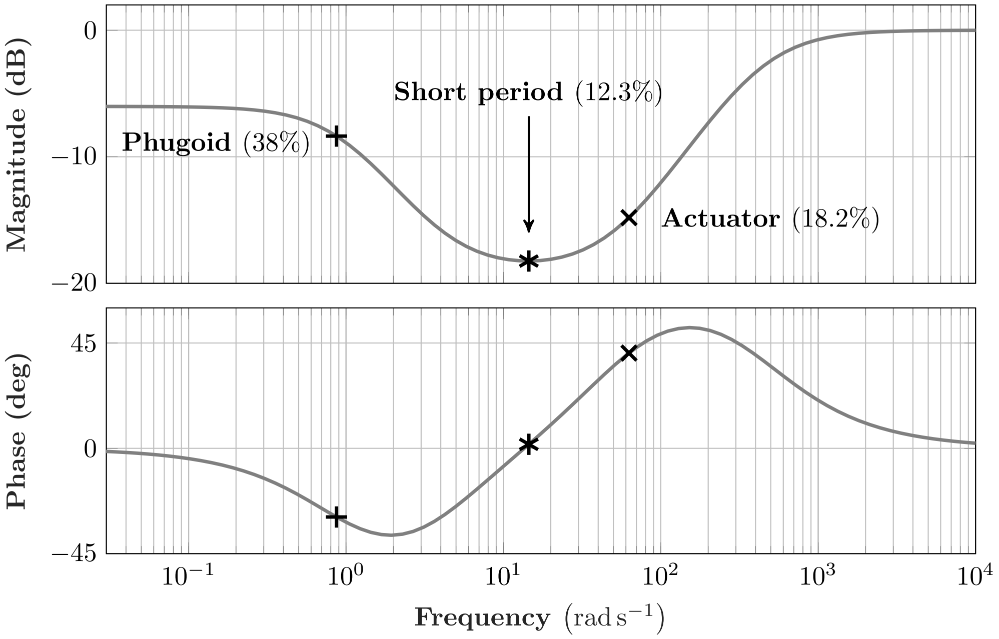


Figure 6.14: The Bode diagram of the weight W_Δ that shapes the uncertainty in G_{sp}.

Incorporation of rate bounds and IQCs

The basis functions for incorporating the rate bounds and the IQCs are selected identically to the lateral-directional case. In particular, the matrices X and Z in Theorem 2 are expressed using the bases $\left(1, V, V^2\right)$. The norm-bounded LTI uncertainty Δ satisfies all IQCs defined by multipliers of the form given in Equation (6.1). The multiplier is parametrized as explained previously in Equation (6.6).

Synthesis results

The synthesized filter F_{long} is a quadratically stable LPV system (see Definition 3) that is scheduled by the airspeed V and its time derivative $\dot{V}$. In order to assess its performance, the *nominal, unweighted closed-loop* is constructed by setting all the weights to unity and Δ to zero in Figures 6.8 and 6.13. Thus this closed-loop consists only of G_{sp}, K_{PD}, G_a^L, and F_{long}. Figure 6.15 shows the Bode diagrams of the sensitivity functions S_e from $-\delta_e$ to e_e at each grid point in the airspeed domain along with the sensitivity bound $\bar{S}_e$ that was described earlier. The plot indicates that F_{long} satisfies the desired sensitivity response at each point in the domain. Finally, F_{long} and F_{lat} are integrated into F (Figure 6.6), which is in turn integrated into F_B (Figure 6.5).

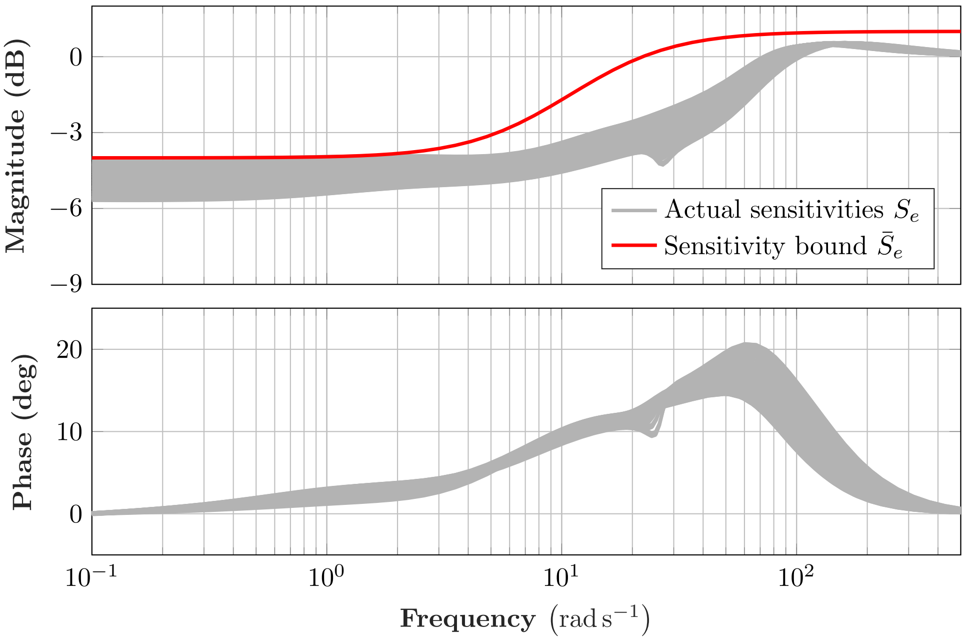


Figure 6.15: The Bode diagrams of the sensitivity functions S_e from $-\delta_e$ to e_e at each grid point and their upper bound $\bar{S}_e$.

6.7.5 Worst-Case Analysis

A worst-case analysis is conducted to illustrate the benefits of using the uncertain, LPV design model to synthesize the lateral-directional filter F_{lat}. Four filters are synthesized using the four design models listed in Table 6.1. Each filter is then interconnected with the uncertain, LPV design model, such that the final system consists of $G_{lat}\left(1 + \Delta W_{\Delta}\right)$, K_A, G_a^L, and F_{lat}. The worst-case performance of each filter is analyzed by computing bounds on the worst-case gain from $-\delta_a$ to e_a, which accounts for all possible norm-bounded LTI uncertainties. Recall that the channel $-\delta_a \rightarrow e_a$ quantifies the sensitivity of the estimation error to the true aileron position and is only one of the channels considered during the synthesis. Upper bounds, which account for all allowable airspeed trajectories, are computed by conducting a LPV worst-case gain analysis using Theorem 2 of [102]. Lower bounds, which account for constant airspeed trajectories, are computed by conducting a LTI worst-case gain analysis at each fixed airspeed in the domain and then choosing the largest such gain.

The left-side of Figure 6.16 shows the upper bound on the worst-case gain as a function of the upper bound on $\|\Delta\|_\infty$. As expected, larger uncertainty norm bounds result in larger worst-case gains across all four filter types. However, the increase in the worst-case gain is markedly less pronounced for the robust-LPV filter. In particular, the upper bounds on the worst-case gain for $\Delta = 0$ are between 1.2 and 1.3 for the nominal-LTI, the robust-LTI, and the nominal-LPV filters and around 1.1 for the robust-LPV filter. In contrast, the upper bound on the worst-case gain for $\|\Delta\|_\infty \leq 1$ is around 2.1 for the nominal-LTI and the robust-LTI filters, around 1.6 for the nominal-LPV filter, and around 1.2 for the robust-LPV filter. The rapid performance degradations seen in the nominal-LTI and the robust-LTI filters are due to the fact that their respective design models are LTI, whereas this analysis considers all allowable airspeed trajectories. The nominal-LPV filter accounts for the airspeed variations and thus achieves lower worst-case gain than the two LTI filters. The robust-LPV filter achieves the lowest worst-case gain across all uncertainty levels because its design model accounts not only for the airspeed variations but also for the model uncertainty.

The right-side of Figure 6.16 shows the lower bound on the worst-case gain as a function of the airspeed for $\|\Delta\|_\infty \leq 1$. Each point on this plot represents a lower bound on the worst-case gain of the uncertain, LTI system at the corresponding fixed airspeed. These bounds are computed using the command `wcgain` of Matlab's Robust Control Toolbox [155]. This function not only returns the lower-bound on the worst-case gain, but also the worst-case uncertainty that achieves this gain. Thus the lower bounds shown in Figure 6.16 are *true* lower bounds at the corresponding fixed airspeed. The largest such value across all constant airspeeds is thus a lower bound on the worst-case gain of the uncertain, LPV system. The dashed rectangles indicate the largest lower bounds and the corresponding worst-case airspeeds. Once again, the robust-LPV filter has the smallest lower bound across all airspeeds. The largest lower bound on the worst-case gain is 1.42 for the nominal-LTI filter, 1.32 for the robust-LTI and the nominal-LPV filters, and occurs at an airspeed of 20 $\mathrm{m\,s^{-1}}$ in all three cases. In contrast, the largest lower bound on the worst-case gain is 1.13 for the robust-LPV filter and occurs at an airspeed of 15.78 $\mathrm{m\,s^{-1}}$. The lower bound for the robust-LPV filter is relatively invariant

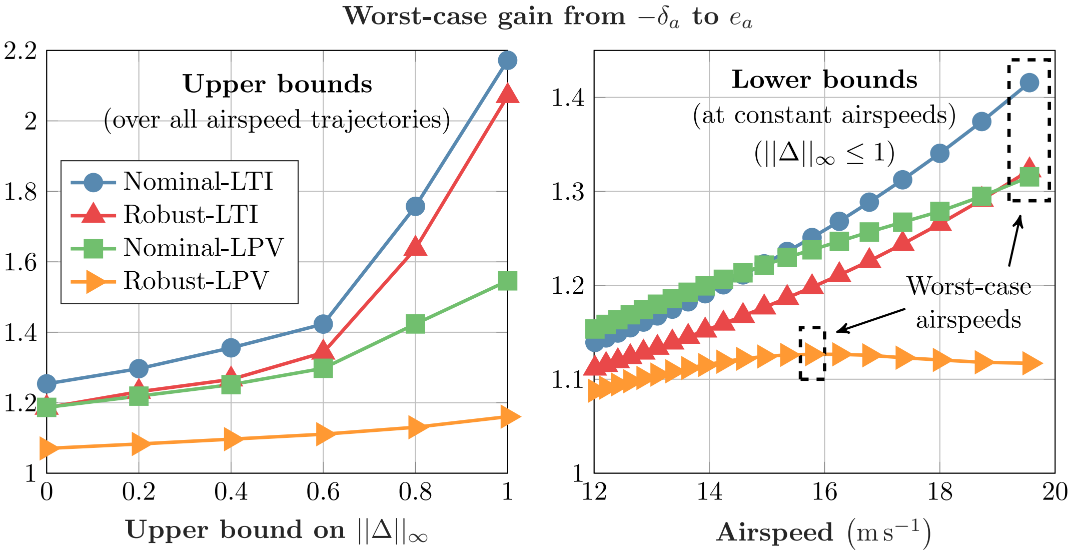


Figure 6.16: Left: The upper bound on the worst-case gain from $-\delta_a$ to e_a, computed over all allowable airspeed trajectories, as a function of the upper bound on $\|\Delta\|_\infty$. Right: The lower bound on the worst-case gain from $-\delta_a$ to e_a, computed at constant airspeeds, as a function of the airspeed. The worst-case airspeeds are marked.

to the airspeed. Moreover, the upper and the lower bound plots together indicate that the true worst-case gain for the robust-LPV case lies in the interval $[1.13, 1.16]$.

Figure 6.16 further indicates that the relative performances of the nominal-LTI, the robust-LTI, and the nominal-LPV filters differ depending on the analysis. It is particularly interesting to compare the robust-LTI and the nominal-LPV filters because the former accounts only for the model uncertainty while the latter accounts only for the airspeed variations. The upper bound plot indicates the nominal-LPV filter performs better when all allowable airspeed trajectories are considered. On the other hand, the lower bound plot indicates that the robust-LTI filter performs better when only fixed airspeeds below 18 $\mathrm{m\,s^{-1}}$ are considered.

In addition to the lower bound, the worst-case uncertainty satisfying $\|\Delta\|_\infty \leq 1$ is computed at 15.78 $\mathrm{m\,s^{-1}}$ for the robust-LPV case and at 20 $\mathrm{m\,s^{-1}}$ for the other three cases. The time-domain performances of the four filters, with Δ substituted by their respective worst-case uncertainties, are evaluated using step responses as shown in Figure 6.17. In particular, responses of the four filters to a 10° step change in the aileron position are shown at the worst-case airspeeds of 15.8 $\mathrm{m\,s^{-1}}$ (left) and 20 $\mathrm{m\,s^{-1}}$ (right). The responses of the nominal-LTI, the robust-LTI, and the nominal-LPV filters clearly degrade, with overshoot and/or transients, at their worst-case airspeed of 20 $\mathrm{m\,s^{-1}}$. In contrast, the response of the robust-LPV filter at its worst-case airspeed of 15.8 $\mathrm{m\,s^{-1}}$ is largely similar to its response at 20 $\mathrm{m\,s^{-1}}$. This shows that the robust-LPV filter has more consistent worst-case performance across the airspeed domain.

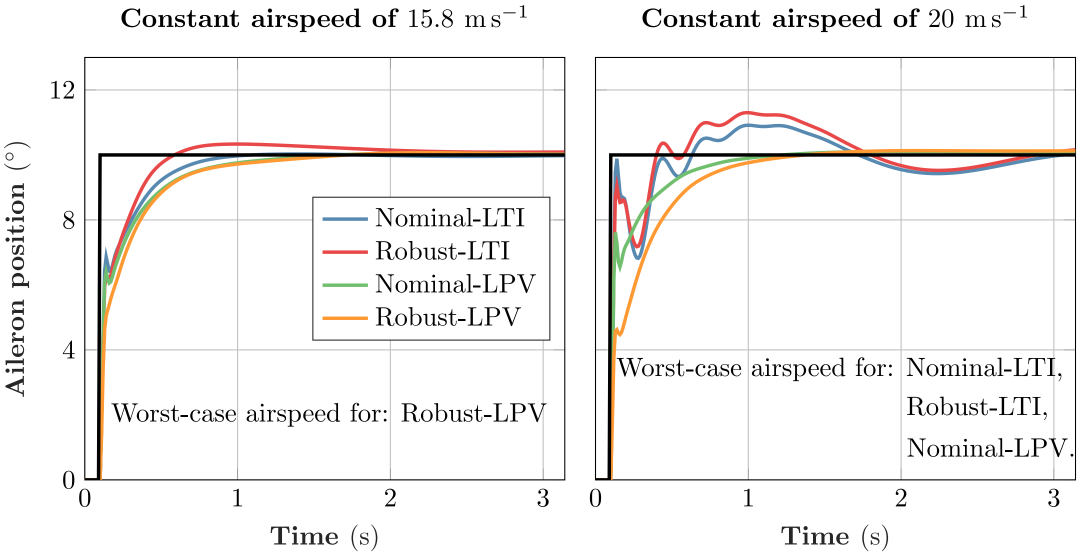


Figure 6.17: The responses of the four filters to a 10° step change in the aileron position at the worst-case airspeeds of 15.8 $\mathrm{m\,s^{-1}}$ (left) and 20 $\mathrm{m\,s^{-1}}$ (right).

6.8 Flight Test Results: Offline Validation

6.8.1 Summary of Flight Experiments

This section validates the FDI algorithms using flight test data in an offline setting. The FDI algorithms operate in open-loop until they declare a switch from the nominal to the fault-tolerant controller. Thus evaluating the FDI algorithm performance (up to the switching time) offline is equivalent to evaluating their performance online, i.e. in real-time. Offline validation entails feeding prerecorded flight data to the FDI algorithms and assessing their performance. The validation is performed using data from FLT32 and FLT33 on 2017-10-05. (See Appendix C for a list of flight tests.) In both flights, the Vireo tracks a rectangular flight path at constant altitude using the nominal controller (Chapter 4). The winds during both the flights are 2.7 $\mathrm{m\,s^{-1}}$ from the South, as reported by the National Weather Service [6].

Both flights consist of a number of test points wherein stuck faults are injected in the right elevon. To avoid overspeeding, only trailing edge up (negative) stuck faults are injected. Each test point begins when the fault is injected and ends when the fault is removed. Table 6.4 lists the fault magnitude, the fault injection time, and the fault duration for each test point, along with the detection times obtained using Design A and the robust-LPV version of Design B. Section 6.8.5 provides a more detailed comparison of the designs. The table and the validation results exclude the third test point of FLT32 due to issues with its execution. Since the nominal controller is not designed to manage stuck faults, the Vireo strays off its intended flight path during each test point.

Table 6.4: The description of the test points conducted in FLT32 and FLT33. The detection times obtained using Designs A and B are provided for a quick comparison.

Test point	**Right elevon fault**	**Injection time** (s)	**Fault duration** (s)	**Detection time** (s)	
				Design A	**Design B Robust-LPV**
		FLT32			
T1	−5°	1308	102	8.51	6.93
T2	−5°	1456	43	3.56	3.03
T4	−5°	1601	22	12.9	8.26
T5	−5°	1649	41	2.54	1.89
		FLT33			
T1	−5°	992	51	8.78	8.23
T2	−5°	1070	50	5.34	5.06
T3	−5°	1159	28	5.22	4.5
T4	−6°	1237	48	2.78	1.84
T5	−7°	1321	7	2.48	1.57
T6	−7°	1344	10	2.20	1.26

Before evaluating the FDI algorithms using the test points described above, it is

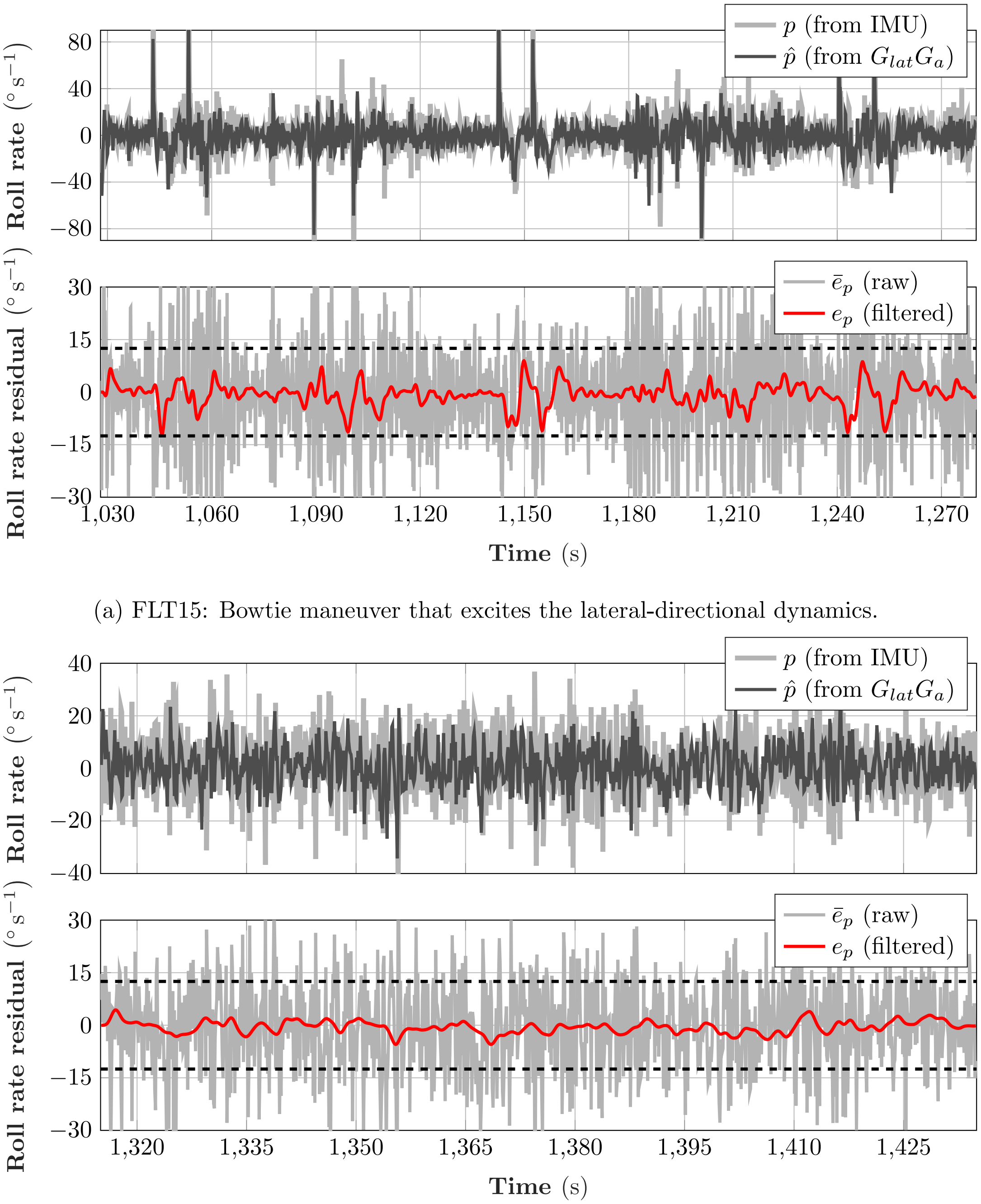
[6] `https://www.wunderground.com/weather/us/mn/kmsp`

important to select the residual filter and the thresholds. The next section considers the selection of these parameters using data from two earlier flights wherein no faults are injected. The reader who is primarily interested in the performance of the FDI algorithms during the test points of FLT32 and FLT33 may skip ahead to Section 6.8.3.

6.8.2 Filter and Threshold Selection

Recall from Sections 6.6 and 6.7 that the raw residuals of Designs A and B are filtered using R_A (Figure 6.4) and R_B (Figure 6.5) in order to attenuate high frequencies unrelated to the fault. The residual evaluation stage applies the thresholds T_* to the filtered residuals to detect the fault (Figure 6.3). The thresholds and filters for both designs are selected to ensure that the FDI algorithm does not declare false alarms when applied to flight data wherein no faults are injected. Of all such flight data, FLT15 and FLT18 are chosen because they involve large maneuvers, such as banked turns, and altitude and airspeed changes. In particular, FLT15 includes a bowtie maneuver that excites the lateral-directional dynamics and FLT18 includes step commands in the altitude and the airspeed that excite the longitudinal dynamics. FLT15 and FLT18 also happen to be the flights that were used to validate the nominal controller in Section 4.7.

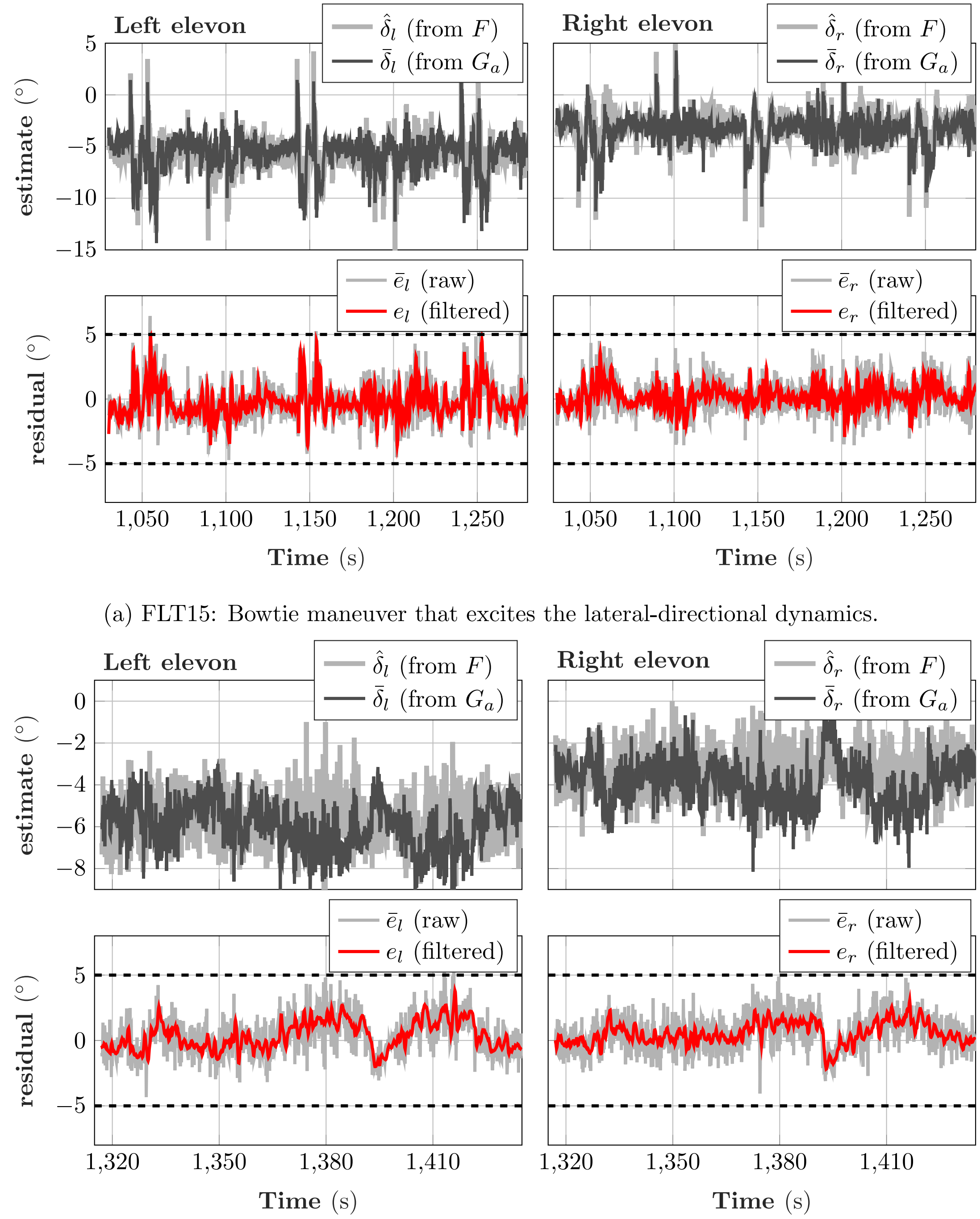
Figures 6.18a and 6.18b show the results of applying Design A to FLT15 and FLT18, respectively. Within each figure, the top plot shows the measured and the model-predicted roll rates and the bottom plot shows the raw and the filtered residuals. The notation is from Figure 6.4. Since the raw residual exhibits high variance, R_A is selected as a fifth-order, low-pass Bessel filter with a bandwidth of $2\,\mathrm{rad\,s^{-1}}$. The filtered residual e_p has a mean close to zero in both flights, but exhibits larger and more frequent deviations away from zero during the bowtie maneuvers in FLT15. This is because of the modeling errors in the aileron-to-roll rate response, which become particularly visible during banked turns. For example, the Vireo executes a right-banked turn of 35° at 1043 s (see Figure 4.9). During the transition to the turn, the peak measured roll rate is around $97^\circ\,\mathrm{s^{-1}}$ and the peak model-predicted roll rate is around $83^\circ\,\mathrm{s^{-1}}$. The corresponding peak e_p is $-12^\circ\,\mathrm{s^{-1}}$. In contrast, the altitude and airspeed step commands in FLT18 do not excite the lateral-directional dynamics much. Thus e_p is largely close to zero for the entirety of the flight. The thresholds for e_p are selected as $\pm 12.5^\circ\,\mathrm{s^{-1}}$ (dashed lines) in order to avoid false alarms.



(a) FLT15: Bowtie maneuver that excites the lateral-directional dynamics.

(b) FLT18: Altitude and airspeed step commands that excite the longitudinal dynamics.

Figure 6.18: The thresholds for Design A ($\pm 12.5^\circ\,\mathrm{s}^{-1}$ on the roll rate residual e_p) are selected such that no false alarms are declared during FLT15 and FLT18.



(a) FLT15: Bowtie maneuver that excites the lateral-directional dynamics.

(b) FLT18: Altitude and airspeed step commands that excite the longitudinal dynamics.

Figure 6.19: The thresholds for Design B (±5° on the left e_l and right e_r elevon residuals) are selected such that no false alarms are declared during FLT15 and FLT18.

Figures 6.19a and 6.19b show the results of applying Design B to FLT15 and FLT18, respectively. Within each figure, the top row of plots show the two sets of estimates of the left and right elevon positions and the bottom row of plots show the raw and the filtered residuals. The notation is from Figure 6.5. In order to attenuate high frequencies, R_B is selected as a fifth-order, low-pass Bessel filter with a bandwidth of 10 $\mathrm{rad\,s^{-1}}$. In both flights, the filtered residuals (e_l, e_r) are close to zero for the majority of the time segment shown, but deviate from zero during the maneuvers. Once again, this is because of errors in the design models G_{lat} and G_{sp}, which are used in the synthesis of F_{lat} and F_{long}. The right-banked turn of 35° at 1043 s is once again considered as an example. During the transition to the turn, the peak raw residual in the left elevon position estimate is 4.4°. The corresponding peak e_l is 3.7°.

In contrast, the peak residuals are not as high during the altitude and airspeed step commands in FLT18. However, these longitudinal maneuvers do expose the shortcoming of using only the pitch rate in F_{long} (see Section 6.7). In particular, the altitude and airspeed step commands result in pitch attitude changes that F_{long} and, in turn, F do not account for. For example, the altitude command increases by 15 m at 1391 s (see Figure 4.10). The nominal pitch tracker K_{PT} (see Figure 4.5b) responds to the ensuing changes in the commanded and estimated pitch attitudes by commanding an elevator deflection. This is reflected in the elevon position estimates generated from G_a $\left(\bar{\delta}_l, \bar{\delta}_r\right)$, but not in those generated from F $\left(\hat{\delta}_l, \hat{\delta}_r\right)$. This mismatch is seen as the sharp drop in the residuals e_l and e_r around 1390 s. The thresholds for e_l and e_r are selected as ±5° (dashed lines) in order to avoid false alarms and to manage this shortcoming.

6.8.3 Results Using Design A

Figures 6.20 and 6.21 show the filtered roll rate residual e_p of Design A during the test points of FLT32 and FLT33, respectively. The raw residual $\bar{e}_p$ is not shown to avoid clutter. Each subplot shows the residual before and after the fault, with the vertical line marking the fault injection time. For clarity, the residuals are shown only for about 20 s after the fault injection and not for the entire duration of the fault. The following discussion uses the nomenclature FLT**X**-T**Y** to refer to test point **Y** of flight **X**.

As seen in both the figures, the residual crosses one of the thresholds $\left(\pm 12.5^\circ\,\mathrm{s^{-1}}\right)$ a short time after the fault injection. There are no false alarms or missed detections in

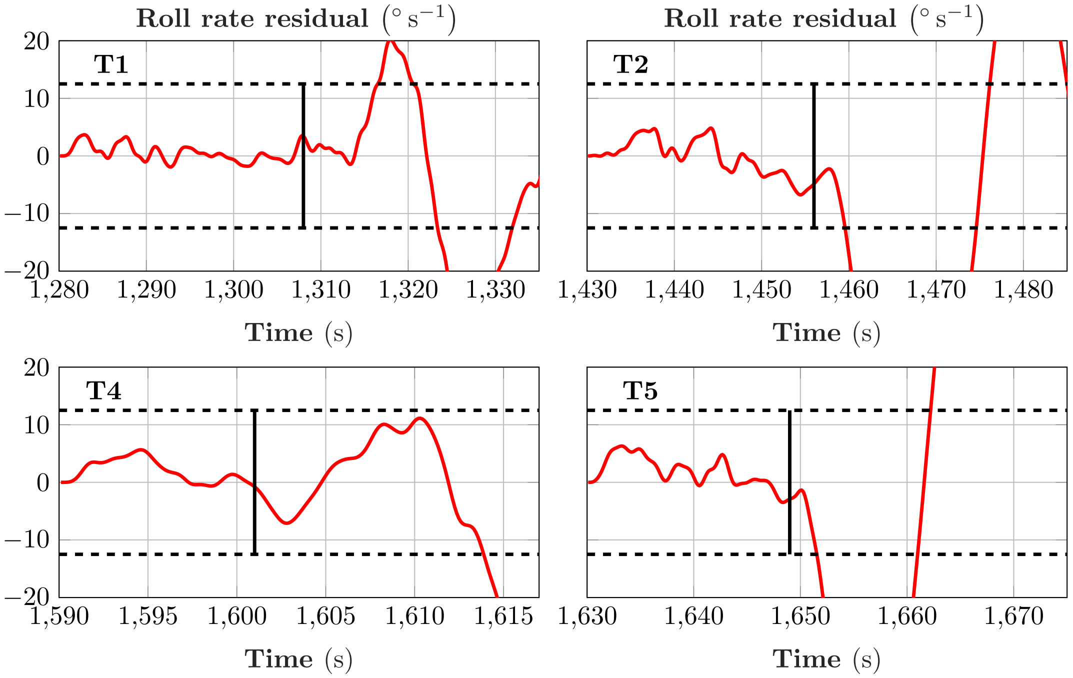


Figure 6.20: The filtered roll rate residual e_p of Design A during the test points of FLT32. The vertical line marks the fault injection time.

any of the test points. Although the fault magnitudes are similar in all the test points (see Table 6.4), the detection time varies depending on the maneuver being performed immediately prior to the fault. In particular, recall that the Vireo tracks a rectangular flight path at constant altitude in both the flights. The Vireo flies straight and level when tracking the four sides of this rectangle and banks sharply when navigating the four corners. It takes longer to detect a fault that is injected during the straight and level segment as compared to one that is injected during the banked turns. For example, the fault in FLT32-T5 is injected during a banked turn and is detected in 2.54 s. In contrast, the fault in FLT32-T4 is injected during a straight and level segment and is detected only after 12.9 s. The longer detection time in FLT32-T4 is due to the lack of excitation of the filter during the straight and level segment. For instance, although the residual in FLT32-T4 reaches a peak at 1610 s, it does not cross the positive threshold (see Figure 6.20). Rather, it reverses direction and crosses the negative threshold a few seconds later, thereby prolonging the fault detection.

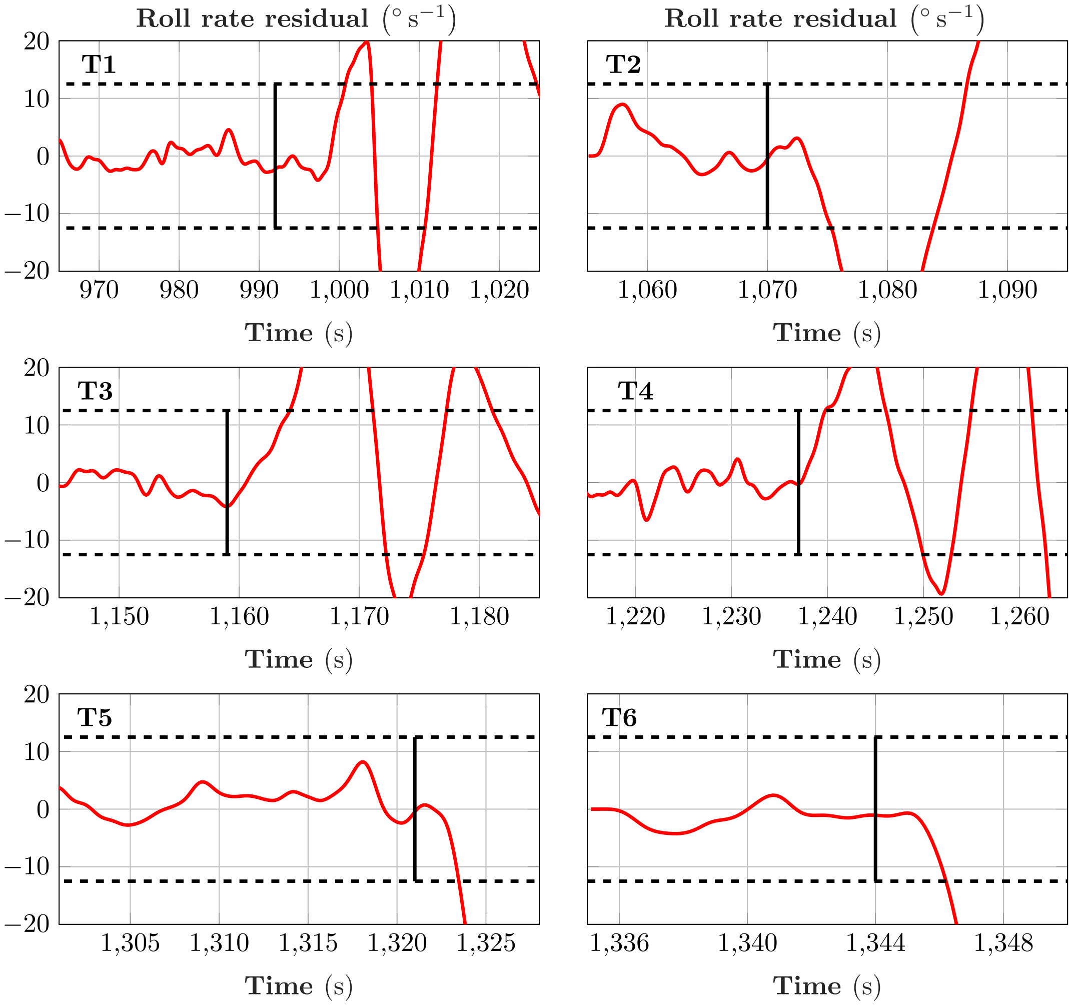


Figure 6.21: The filtered roll rate residual e_p of Design A during the test points of FLT33. The vertical line marks the fault injection time.

Similar observations are made for the test points of FLT33 (see Figure 6.21). The faults in the test points T1, T2, and T3 are injected when the bank angle of the aircraft is relatively shallow, i.e. in the interval $[0, 10]\,^\circ$. Design A detects these faults in 8.8 s, 5.34 s, and 5.2 s, respectively. In contrast, the faults in the test points T4, T5, and T6 are injected when the bank angle of the aircraft is relatively steep, i.e. in the interval $[10, 30]\,^\circ$. Design A detects these faults in 2.8 s, 2.5 s, and 2.2 s, respectively. Although the detection time can be decreased by choosing tighter thresholds, this comes at the expense of increased probability of false alarms. One possible solution is to use time-varying or state-dependent thresholds or up-down counters, all of which can favorably alter the trade-off between the detection time and the probability of false alarm. Another possible solution is to use a more sophisticated algorithm, such as Design B, as demonstrated next.

6.8.4 Results Using Design B

Figures 6.22, 6.23, and 6.24 show the filtered left e_l and right e_r elevon residuals of Design B during the test points of FLT32 and FLT33. (FLT33 is split over Figures 6.23 and 6.24.) The raw residuals are not shown to avoid clutter. As before, each subplot shows the residual before and after the fault, with the vertical line marking the fault injection time. The time spans in each subplot are the same as those used in the previous figures with Design A. All the plotted residuals are generated using the robust-LPV filter. The next section compares the performances of the different filter types.

As seen in all the figures, the left elevon residual remains within the thresholds ($\pm 5°$) both before and after the fault injection. On the other hand, the right elevon residual crosses one of the thresholds a short time after the fault injection. There are no false alarms or missed detections in any of the test points. As with Design A, the detection time varies depending on the maneuver being performed immediately prior to the fault. However, for any given test point, Design B detects the fault sooner than Design A. For example, Design B detects the fault in FLT32-T5 (injected during a banked turn) in 1.9 s, whereas Design A detects the same fault in 2.54 s. The benefits of Design B are more readily apparent when the fault is injected during a straight and level segment, e.g. in FLT32-T4. For this particular test point, Design B detects the fault in 8.26 s, whereas Design A takes 12.9 s. This illustrates that Design B is better suited for fault scenarios that have low persistency of excitation. Similar observations are made for the test points of FLT33 (see Figures 6.23 and 6.24).

6.8.5 Comparison of the Algorithms

In addition to considering the time histories of the residuals during individual test points, it is insightful to compare the aggregate performance of the FDI algorithms across all

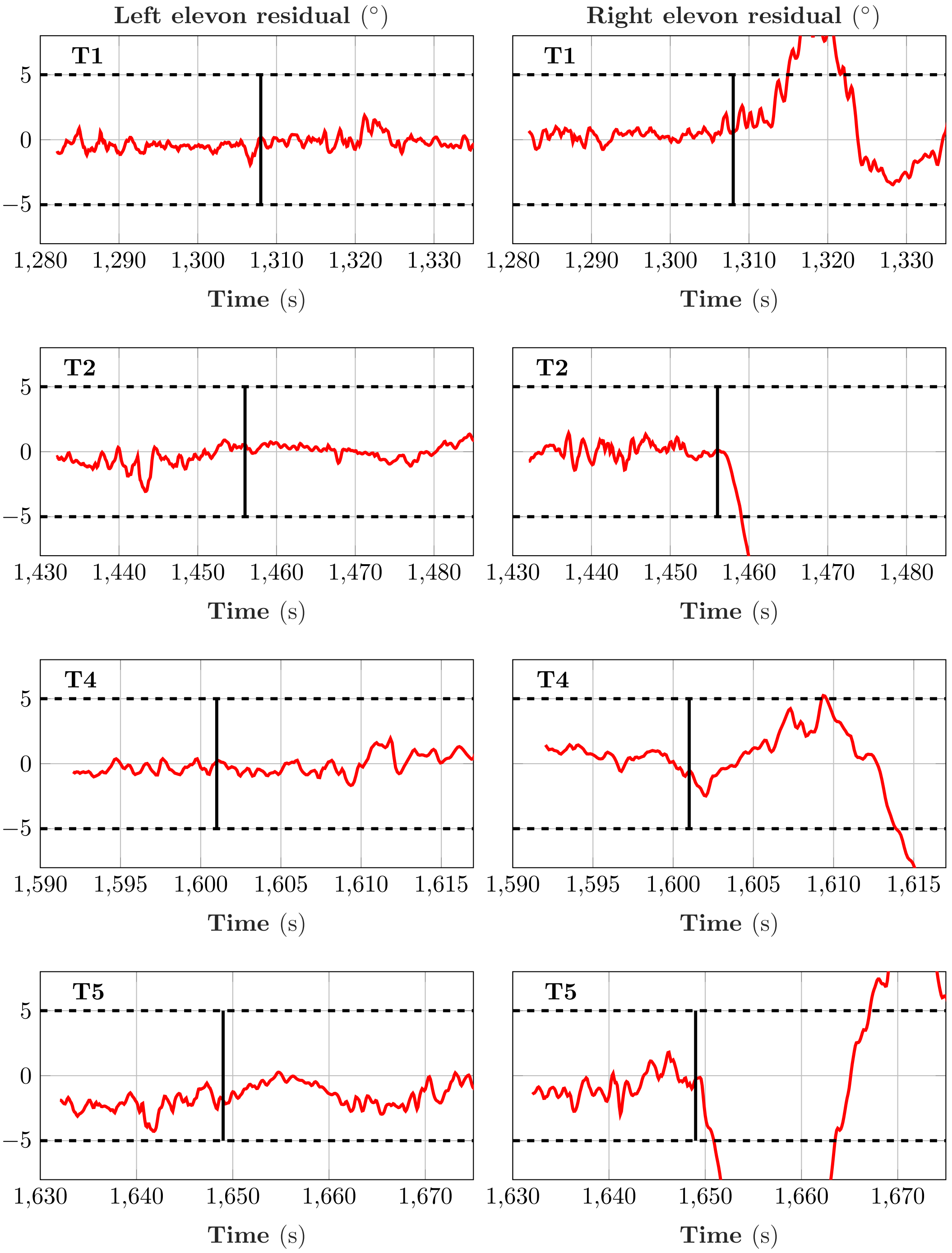


Figure 6.22: The filtered left e_l and right e_r elevon residuals of Design B during the test points of FLT32. The vertical line marks the fault injection time.

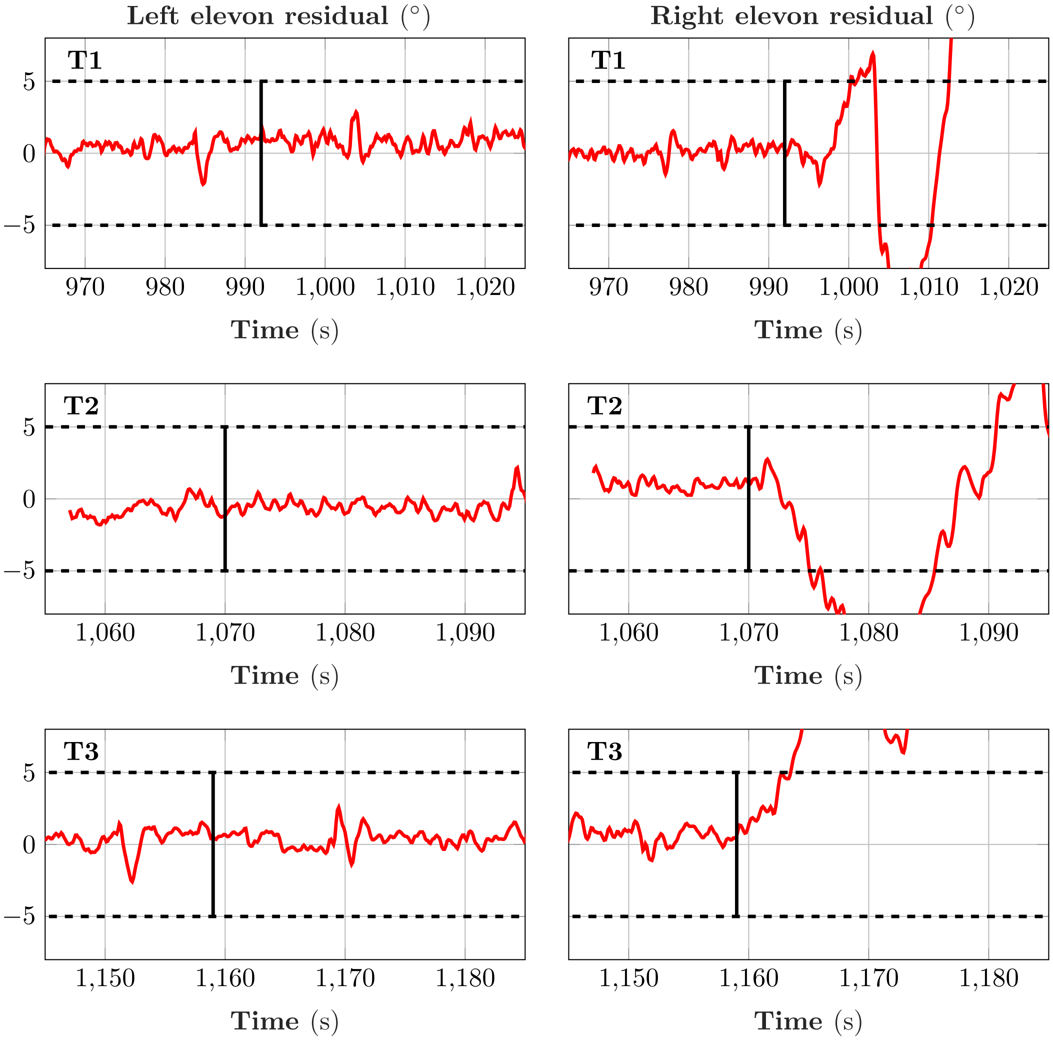


Figure 6.23: The filtered left e_l and right e_r elevon residuals of Design B during the test points T1, T2, and T3 of FLT33. The vertical line marks the fault injection time.

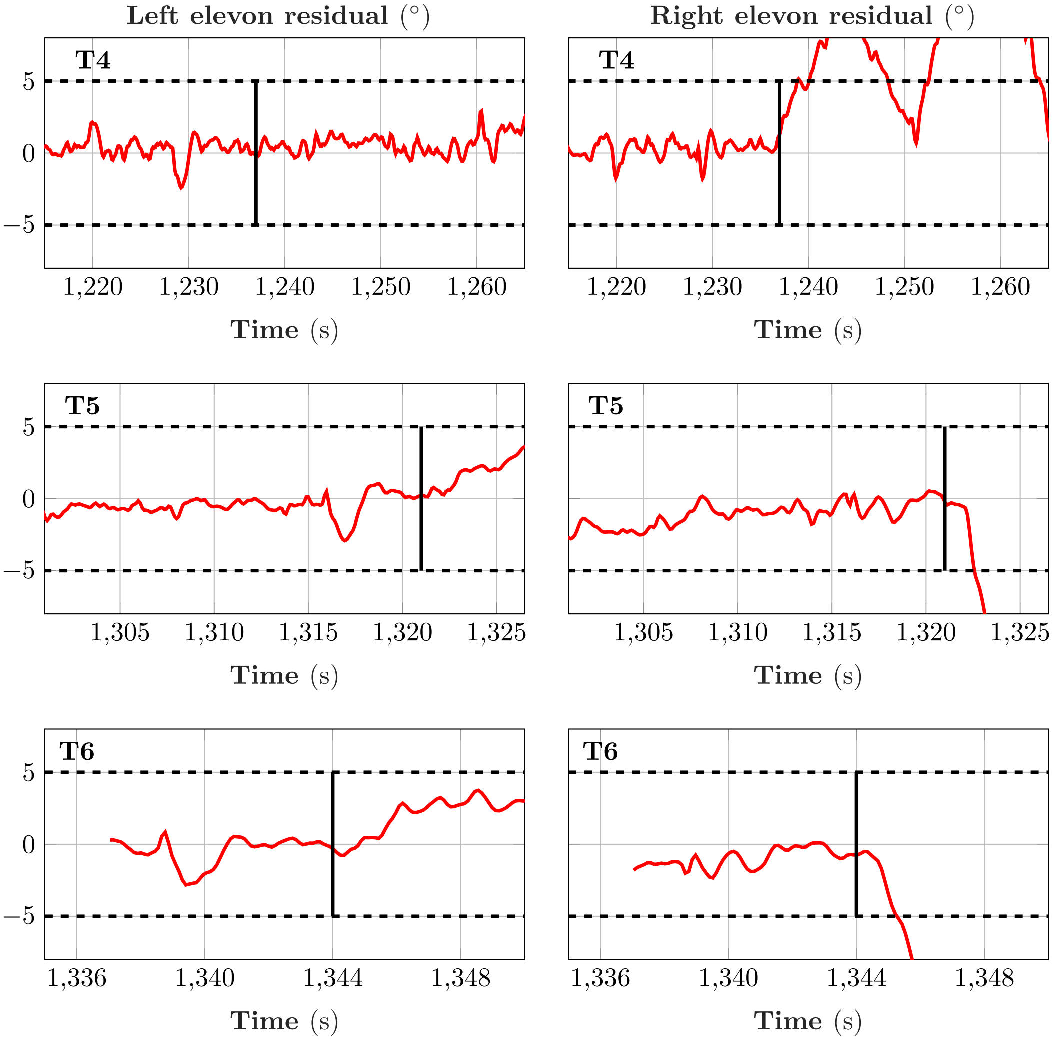
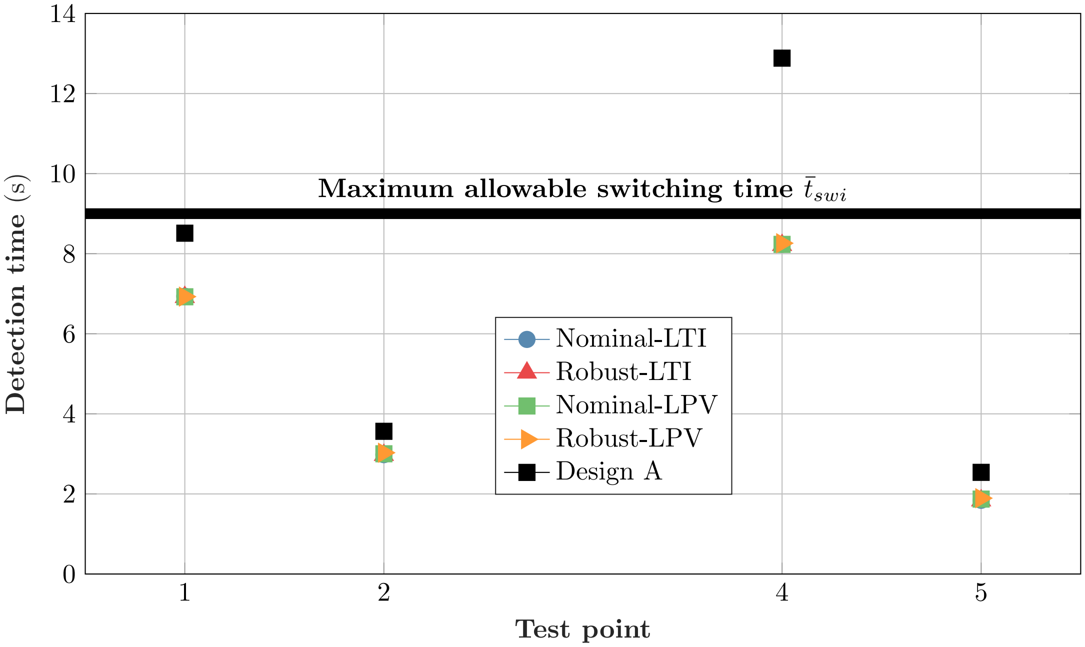


Figure 6.24: The filtered left e_l and right e_r elevon residuals of Design B during the test points T4, T5, and T6 of FLT33. The vertical line marks the fault injection time.

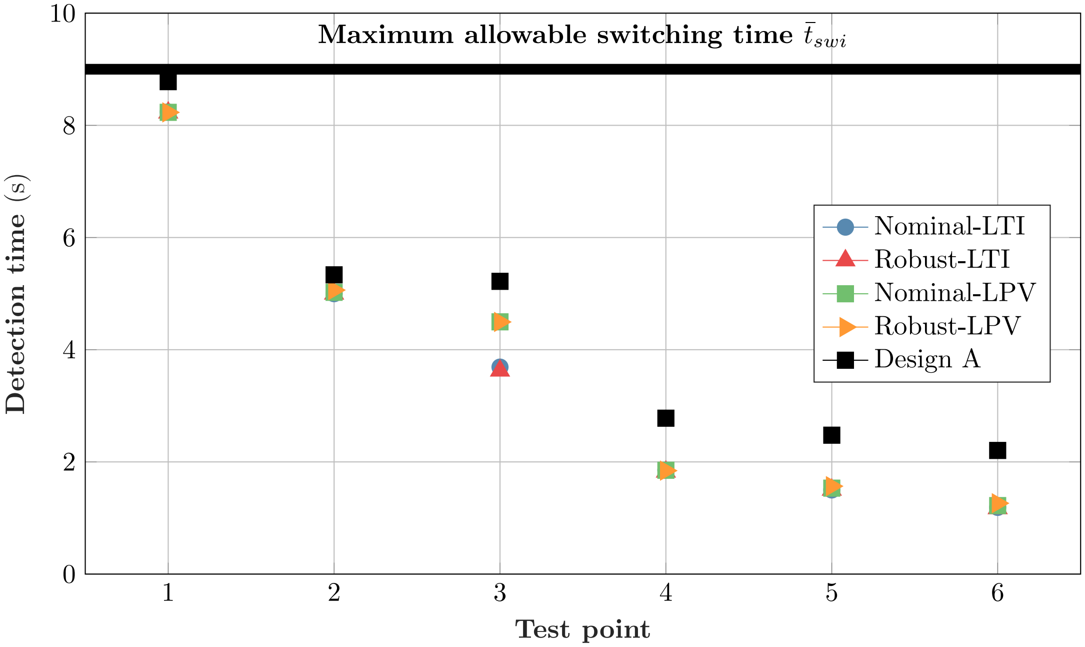
the test points. Figures 6.25a and 6.25b show the fault detection times obtained using Design A and Design B (which has four variants; see Table 6.1) during the test points of FLT32 and FLT33. The figures indicate that Design B detects the fault sooner than Design A in all the test points. The difference between the detection times is in the range $[0.5, 1.5]$ s depending on the test point. In addition, both the designs detect the faults within the maximum allowable switching time of 9 s (see Section 6.4.3) in all the test points except FLT32-T4. As explained previously, the fault in FLT32-T4 is injected during a straight and level segment and is detected by Design A only after 12.9 s. Design B detects this fault 4.6 s sooner and is thus within the maximum allowable switching time. Further, the four variants of Design B achieve very similar detection times in all the test points except FLT33-T3, where the two LTI filters achieve lower detection times. This is because the airspeed in FLT33-T3 is close to the design airspeed of the LTI filters $\left(15.4\ \mathrm{m\,s^{-1}}\right)$. It is also likely that the LTI filters have a different trade-off between the detection time and the probability of false alarm. The performances of the four variants, however, differ in the absence of faults, as explained next.

In the absence of faults, it is desired that the residuals are as small as possible in order to avoid false alarms. To this end, Figures 6.26a and 6.26b compare the performances of the four variants of Design B using the summary statistics of the filtered left and right elevon residuals *before* the fault injections in FLT32 and FLT33, respectively. For each test point, the summary statistics are computed over a time segment before the fault injection and are displayed using box plots. The box plots show the median (black circle), the 25th percentile (lower edge of box), and the 75th percentile (upper edge of box) of the residuals. The plots show that all the percentiles are contained within the thresholds of ±5°. Smaller interquartile ranges are indicative of smaller residuals.

As with the detection time metric, the interquartile range of a particular filter varies between the test points depending on the maneuver being performed. For example, the interquartile ranges are larger during banked turns (e.g. FLT32-T5) as compared to straight and level segments (e.g. FLT32-T4). Further, the differences between the interquartile ranges of the four filters are more or less pronounced depending on the test point. In particular, the nominal controller tracks an airspeed command of $15.4\ \mathrm{m\,s^{-1}}$ in both the flights. However, due to variations in wind gusts and atmospheric turbulence, the measured airspeed is slightly different between the test points. The nominal-LPV

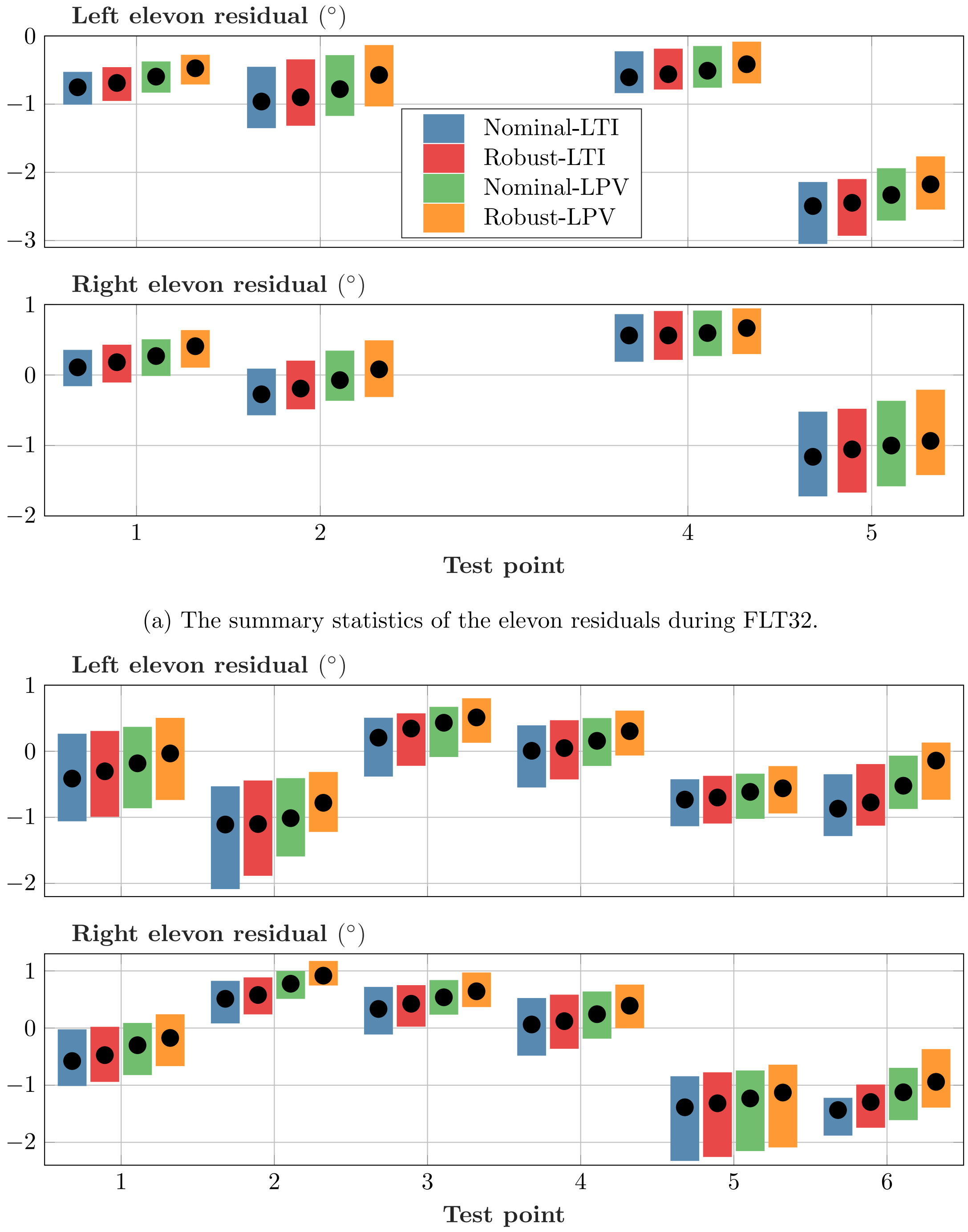


(a) The detection times for the test points of FLT32.



(b) The detection times for the test points of FLT33.

Figure 6.25: The fault detection times obtained using Designs A and B during the test points of FLT32 and FLT33. The maximum allowable switching time $\bar{t}_{swi}$ is 9 s.



(a) The summary statistics of the elevon residuals during FLT32.

(b) The summary statistics of the elevon residuals during FLT33.

Figure 6.26: The summary statistics of the filtered left and right elevon residuals of Design B *before* the fault injections in FLT32 and FLT33.

and the robust-LPV filters account for these airspeed variations and thus yield smaller interquartile ranges, as compared to the two LTI filters, in test points T1 and T5 of FLT32 and T1, T2, T3, and T4 of FLT33. Between the two LPV filters, the robust-LPV filter yields the smaller interquartile range because it also accounts for the model uncertainty. The differences are less pronounced for the other test points, e.g. the interquartile ranges of the four filters are very similar in FLT32-T2 and FLT32-T4.

6.8.6 Concluding Remarks

In conclusion, Design B not only offers faster fault detection but also isolates the source of the fault. Therefore, it is better suited for the particular problem of fault detection and isolation on the Vireo. In addition, the flight tests corroborate the comparisons drawn between the four variants of Design B during the worst-case analysis of Section 6.7.5. In particular, the robust-LPV filter achieves the smallest upper and lower bounds on the worst-case gain. This is a direct consequence of incorporating both the airspeed variations and the model uncertainty in the synthesis. Thus, in the absence of faults, the robust-LPV filter yields the smallest residuals, especially at the flight test points that witness varying airspeeds.

Finally, the better fault detection performance of Design B comes at the expense of a more complicated design process (Section 6.7). The four variants of Design B themselves vary in the level of complexity, with the robust-LPV filter being the most complex to design. In addition, there are some implementation considerations with the nominal-LPV and robust-LPV filters, e.g. interpolating the state-space matrices at every sample time and accounting for the bias in the measurement of the airspeed. Given the offline nature of the validation, these implementation considerations are, unfortunately, left unexplored in this chapter. Any future effort at online implementation, however, will necessarily have to address these considerations.

Chapter 7

Fault-Tolerant Controller

7.1 Introduction

Stuck elevon faults are critical fault modes because of their high likelihood and high severity (Section 6.3). They are manageable as long as the autopilot detects, identifies, and tolerates the fault. While Chapter 6 focuses on the fault detection and identification, this chapter focuses on the fault-tolerance. Section 7.2 presents related work on fault-tolerant flight control and highlights those that include flight demonstrations. Section 7.3 formulates the problem and Section 7.4 discusses the challenges of controlling the Vireo with a single elevon. This informs the design of the fault-tolerant controller in Sections 7.5, 7.6, and 7.7. This chapter concludes with flight tests demonstrating the autonomous landing of the Vireo in the presence of a stuck elevon (Section 7.9).

7.2 Related Work

Fault-tolerant control (FTC) is interdisciplinary and borrows from reliability engineering and control theory. Some textbooks offer a general treatment of the subject [120, 122, 123, 156–160]. For example, [160] classifies FTC systems based on whether they are passive or active, whether the faults occur in the actuators or sensors, and whether the faults are continuous or discrete events. In addition, the authors of [160, 161] use graph theory to explore the structure and other fundamental properties of a system to fully utilize any redundancy in the process. The control of systems with actuator failures is treated, in a general setting, using adaptive control in [162,163].

Aside from these general treatments, major advances in fault-tolerant control have been a result of the practical requirements of engineering systems. Consequently, most of the literature in FTC is application-specific. For aircraft-specific applications, actuator/control surface failures, loss of thrust, loss of lifting surfaces, and structural damage have historically served as the motivations behind fault-tolerant flight control. Most advances in this field have been since the 1980s and have been reported primarily via conferences and journals. The 1980s saw FTC solutions involving explicit fault detection, isolation, estimation, and control reconfiguration. The 1990s saw the increasing popularity of adaptive and intelligent control approaches. While these early works almost exclusively focused on manned commercial and military aircraft, the 2000s saw the rise of fault-tolerant control for small UAS.

Survey papers provide some perspective to the evolution of fault-tolerant flight control. Steinberg [164] provides a historical overview of research in reconfigurable and fault-tolerant flight control laws, and discusses the challenges in their certification. Zhang and Jiang [165] provide a comprehensive bibliographical review of fault-tolerant control, including aircraft applications. Apart from making a distinction between passive and active FTC, they classify the FTC literature based on the control design methodology and the field of application. They also provide a list of research programs and industrial benchmarks that have compared different fault-tolerant control methods. In addition to the survey papers, textbooks such as [166–168] provide an overview of the field. For example [167] provides the rationale for FTC, a survey of aircraft accidents, the differentiation of active and passive FTC, and a list of control design methods. In addition, [168] provides a case-study involving the fault-tolerant control of a Boeing 747.

The fault-tolerant flight control design methods that are reported in the literature include adaptive control [169–174], receding horizon control (RHC) [175–178], H_∞ optimal control [179,180], gain scheduled/LPV control [181,182], and nonlinear dynamic inversion (NDI) [183–186]. In addition, indirect adaptive control has been enabled by the advances made in real-time parameter identification [170,176,187–190]. The identified plant parameters are then used adaptively in other control techniques, such as

RHC [176, 178] and NDI [185, 186]. All of these FTC techniques involve either replacing or adapting the baseline controller after the occurrence of one or more faults. On the other hand, retrofit fault-tolerant controllers are designed to add-on to, and operate alongside, the baseline controller [177, 191]. In contrast to the model-based FTC techniques discussed thus far, Handelman and Stengel [192, 193] propose a data-driven, rule-based expert system approach to solving FTC by drawing on concepts from artificial intelligence.

In addition to these theoretical developments, some research efforts have included flight demonstrations of FTC in the presence of actuator, thrust, and airframe failures. Table 7.1 provides a non-exhaustive list of FTC demonstrations on fixed-wing aircraft. Most of these demonstrations are the result of programs sponsored by NASA and DARPA and are motivated by real-life failures. For example, the simultaneous loss of all the aerodynamic control surfaces in a commercial aircraft led NASA to develop the propulsion-controlled aircraft [195,197]. More recently, [198] explored the possibility of controlling a commercial aircraft under the complete loss of its vertical tail.

As with any field, different authors and research groups advocate for different approaches to fault-tolerant flight control. While there is no one-size-fits-all solution, the approaches that are successfully flight tested are typically informed by the dynamics of the specific aircraft, the mission requirements, and the expected failure modes. Finally, it is acknowledged that this section covers only a sample of all the available literature on fault-tolerant flight control.

7.3 Problem Formulation

The Vireo is designed for commercial application in precision agriculture. This application is used to formulate the fault-tolerant control problem. Figure 7.1 depicts a precision agriculture mission and the actions that follow after a fault is detected. In particular, the Vireo is deployed on an aerial photography mission over farmland within a so called nominal mission zone. For example, the nominal mission could require the Vireo to fly a lawnmower-like pattern (solid ground track). At some point, a stuck fault occurs in one of the elevons. A fault diagnosis algorithm detects the fault and isolates its source (Chapter 6). Subsequently, the mission is aborted and the aircraft is directed

Table 7.1: Non-exhaustive list of flight test demonstrations of fault-tolerant control on fixed-wing aircraft.

Reference	**Aircraft**	**Fault scenarios**	**FTC approach**
Stewart & Shuck (1990) [194]	F-15 HIDEC	Right horizontal stabilator: locked at trim; locked at 6° from trim; lost due to damage	Multiple model
Burken et al. (1996) [195]	McDonnell Douglas MD-11	Loss of all aerodynamic control surfaces	Multiple model
Monaco et al. (1997) [176]	F-16 VISTA	Missing left horizontal tail	Real-time parameter ID + model-following RHC
Brinker & Wise (2001) [186]	X-36	Locked control surfaces	Adaptive neural network + NDI
Schierman et al. (2004) [196]	X-40A	Locked control surfaces	Trajectory reshaping + adaptive guidance + NDI
Shore & Bodson (2005) [174]	Hobbico Twinstar UAV	Locked left elevator; locked left aileron; left/right engine failure; separation of left stabilizer	Real-time parameter ID + adaptive control
Page et al. (2006) [177]	F/A-18C	Locked right aileron; locked right stabilator	Retrofitted RHC
Jourdan et al. (2010) [172]	Sub-scale F-18 UAV	Locked, floating, and detached aileron; loss of up to 80% wing area moment; loss of thrust	All-attitude control + MRAC + supervisory adaptive control + trajectory reshaping
Chowdhary et al. (2013) [173]	GT Twinstar UAV	Loss of all aerodynamic control surfaces; loss of 50% right wing; increased actuator time-delay	State-dependent guidance + neural network-based MRAC

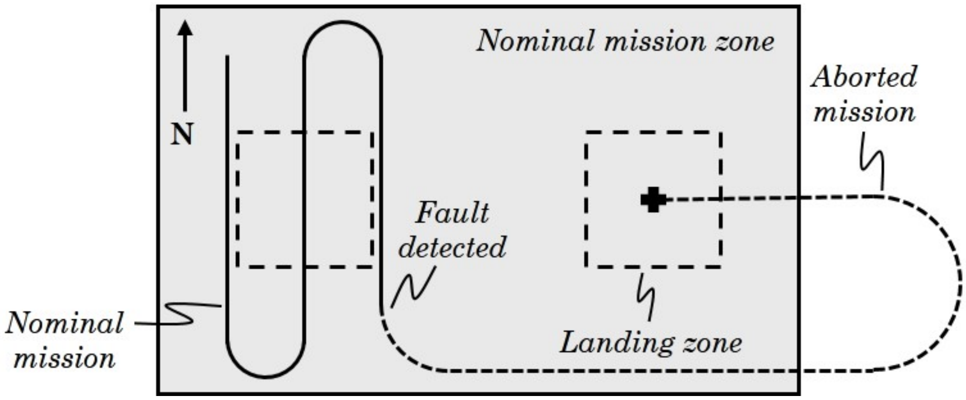


Figure 7.1: A depiction of the ground track of the aircraft during a precision agriculture mission. The fault-tolerant controller is operational during the dashed ground track.

to fly to a suitable landing zone (dashed ground track). The particular fault mode that has occurred may require the fault-tolerant controller to limit the maximum achievable bank angle, and thus the minimum achievable turn radius. Further, due to the aircraft dynamics, it may be preferable to bank in one direction, as opposed to the other. This may in turn make some landing zones more easily accessible than others. Figure 7.1 depicts a fault scenario where left banked turns are preferred and thus the easterly landing zone is easier to access. Finally, the Vireo touches down at a designated spot.

The following questions are posed to narrow the scope of the problem stated above. In posing these questions, it is assumed that the fault diagnosis algorithm successfully diagnoses the fault and reports the fault information to the control law.

1. Given the terrain below and the prevailing winds, what is the best landing zone?

2. Given the best landing zone, what is the best route to take to reach it?

3. Given the best route, how should a fault-tolerant controller be designed so as to track it closely?

The first two questions deal with optimization, e.g. in defining what is meant by "best" while remaining within the allowable flight envelope of the faulted aircraft. These questions are not considered in this thesis. The path and motion planning literatures [199–202] provide the theoretical foundations to answer these two questions. Hence this chapter focuses on answering the third question, which deals primarily with control.

7.4 Input-Output Controllability

The nominal flight controller is designed and analyzed with the aid of the decoupled longitudinal and lateral-directional models (Section 4.3). The dynamics are separable because of the input decoupling afforded by the virtual elevator and aileron (see Equation (3.13) in Section 3.3.2). In contrast, the dynamics are not separable when one of the elevons is stuck, because the operable elevon excites motion in both the longitudinal and the lateral-directional axes. Hence, the fault-tolerant flight controller must be considered using the full, coupled linear model. The inputs of this model are the throttle δ_t and either the left δ_l or the right δ_r elevon, depending on the failure scenario. For the purpose of illustration, consider the right elevon to be stuck at trim. Thus, the inputs are $u = [\delta_t, \delta_l]^T$, the states are $x = [\phi, \theta, p, q, r, u, v, w, Z_e]^T$, and the outputs are $y = [V, h, \phi, \theta, p, q, r]^T$[1]. The state vector excludes ψ, X_e, and Y_e because they are tracked by the guidance law (Section 4.2.2). At the nominal trim airspeed of 15.4 m s^{-1}, the model G_f for a stuck right elevon has the state-space representation:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 & 0.068 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -11 & 0 & 2.5 & 0 & -3.99 & 0 & 0 \\ 0 & 0 & 0 & -5.49 & 0 & 0.88 & 0 & -12.9 & 0 \\ 0 & 0 & -1.5 & 0 & -0.944 & 0 & 0.311 & 0 & 0 \\ 0 & -9.78 & 0 & -1.02 & 0 & -0.15 & 0 & 0.753 & 0 \\ 9.78 & 0 & 1.12 & 0 & -15.3 & 0 & -0.573 & 0 & 0 \\ 0 & -0.67 & 0 & 13.9 & 0 & -0.88 & 0 & -5.69 & -9.43\times10^{-4} \\ 0 & -15.4 & 0 & 0 & 0 & -0.068 & 0 & 0.998 & 0 \end{bmatrix}, \; B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 100 \\ 0 & -92.9 \\ 0 & 4.81 \\ 6.53 & 0.073 \\ 0 & 0.244 \\ 0 & -12.3 \\ 0 & 0 \end{bmatrix} \tag{7.1}$$

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0.998 & 0 & 0.0681 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \; D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}. \tag{7.2}$$

At this stage, it is clear that the number of outputs and states far exceed the number of available inputs. Since, the fault-tolerant flight controller can reasonably track only a limited number of outputs, it is important to design its architecture to prioritize the most important outputs. The following questions are posed to aid this process.

1. Are all the states controllable and observable, and to what extent?

2. What outputs are the most impacted by the available inputs?

[1] Although the Euler angles are estimated by the EKF, they are treated as measurable outputs. Further, the linear accelerations are excluded from y since they are not used for feedback control.

3. Given the single operable elevon, are some outputs more important to control than others? If so, how should they be prioritized?

4. What model properties can help mitigate the impact of uncontrolled outputs?

The answers to these questions are obtained from the principles of flight dynamics [25,26] and the concept of input-output controllability (e.g. see Chapter 5 in [55]).

The first question deals purely with state controllability and observability. For this analysis, the state Z_e is truncated from G_f in order to make A Hurwitz. The controllability W_c and the observability W_o gramians are both positive definite, thus indicating that all the states are controllable and observable. The maximum and minimum singular values of W_c are 1588 and 9, which indicate that some states are weakly controllable. In particular, when W_c is recomputed with only the throttle input, its diagonal entries indicate that the forward speed u is by far the most controllable state. Moreover, as expected from flight dynamics, δ_t has zero controllability over the lateral-directional states. On the other hand, when W_c is recomputed with only the left elevon input, its diagonal entries indicate that, apart from the body-axis velocities, p and q are highly controllable. In addition, ϕ is roughly seven times more controllable than θ.

The second question is partially answered by the first. However, to complete the discussion, note that δ_t has the most impact on V and h. Since good tracking cannot be achieved in both these variables using one input, the fault-tolerant controller uses the total energy as a surrogate. On the other hand, δ_l has roughly equal impact on p and q, and slightly more impact on ϕ than θ. In order to decide on what to control among ϕ, θ, p, and q, the third question needs to be answered. Specifically, the final goal of this thesis is to perform an autonomous landing using a single elevon. A given landing zone can be reached only if the Vireo has the ability to change and maintain course. Thus, roll attitude control (ϕ, p) is arguably more important than pitch attitude control (θ, q).

In finding other means to control θ and q, the throttle is not useful because the thrust line is very close to the CG (Section 2.3.4). Moreover, the bandwidth of the throttle actuator (1 Hz) is too low to control the short period mode (2.3 Hz). Thus, θ and q are relegated to being uncontrolled outputs, which leads to the fourth and final question. In this regard, the most exploitable model property is the stability of the short period mode ($\zeta = 0.39$, see Table 4.3), which implies that θ and q asymptotically approach the trim point in the absence of external disturbances and control inputs. However, since a single elevon is used, it is conceivable that in seeking to control ϕ and p, the fault-tolerant controller inadvertently destabilizes θ and q. Such scenarios are avoided by exploiting the modal frequency separation, as described next.

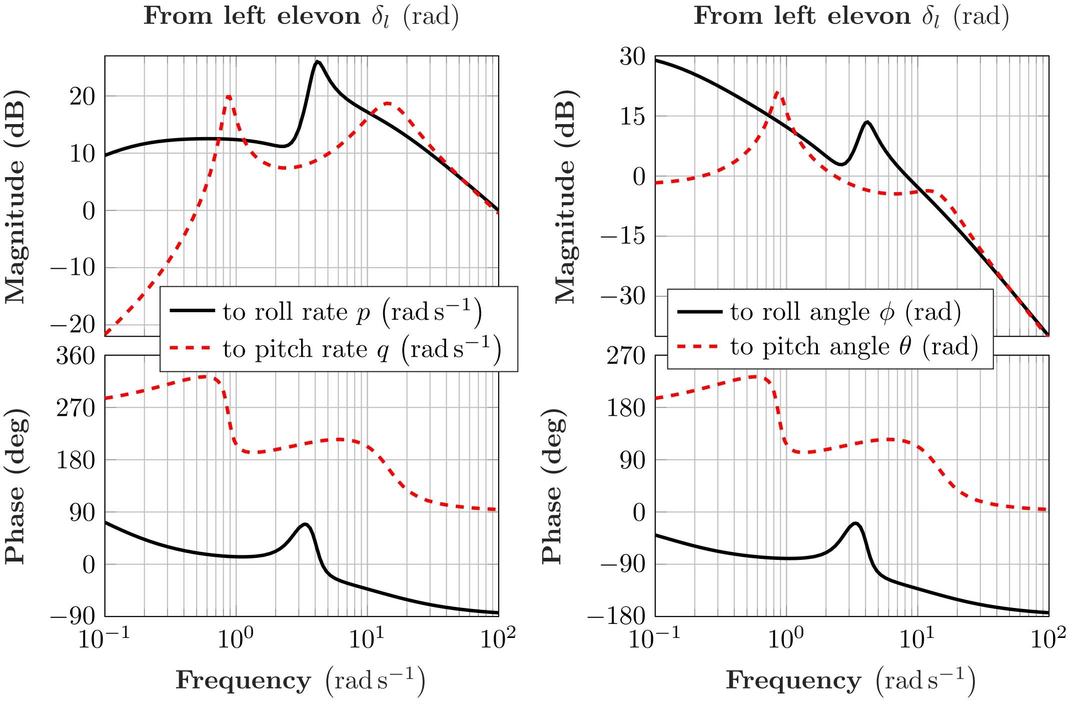


Figure 7.2: The Bode diagrams of the open-loop plant from the left elevon to the roll and the pitch rates (left) and the roll and the pitch angles (right).

Figure 7.2 shows the Bode diagrams of the open-loop plant from the left elevon δ_l to the controlled outputs (ϕ, p) and the uncontrolled outputs (θ, q). In order to make the appropriate comparisons, p and q are shown on the left plot and ϕ and θ are shown on the right plot. Both plots indicate that the phugoid mode $\left(0.87\ \mathrm{rad\,s^{-1}}\right)$ and the short period mode $\left(14.5\ \mathrm{rad\,s^{-1}}\right)$ straddle the dutch roll mode $\left(4.1\ \mathrm{rad\,s^{-1}}\right)$. This frequency separation is exploited by controlling ϕ and p up to a bandwidth that is no greater than 4 $\mathrm{rad\,s^{-1}}$. This ensures that the left elevon does not inadvertently excite the short period mode and, in turn, that θ and q remain well-behaved due to the static stability of the aircraft. Further, owing to the large DC gain from δ_l to ϕ, a given roll attitude command can be achieved using a very small change to the steady-state value of δ_l. This ensures that the steady-state pitch angle, which defines the trim airspeed, is relatively unaltered by the fault-tolerant roll attitude controller. On the other hand, the phugoid mode is within the intended bandwidth of the fault-tolerant controller, and will thus get excited by the left elevon. This problem is managed by the total energy controller, wherein some modifications help increase the phugoid mode damping ratio.

7.5 Fault-Tolerant Controller Architecture

The previous section develops the strategy of controlling (V, h) using the throttle and (ϕ, p) using the operable elevon. Figure 7.3 shows the architecture of the fault-tolerant controller, which consists of a *modified* total energy controller $\bar{K}_{TECS}$ and a *single surface controller* K_S. This is different from the nominal controller architecture (Figure 4.4) in two ways. First, since the single surface controller prioritizes roll attitude control over pitch attitude control, $\bar{K}_{TECS}$ does not compute θ_{cmd}. Second, K_S directly issues commands to the operable elevon, and commands the stuck elevon to its estimated stuck position. Finally, it is practical to lead with the design and validation of $\bar{K}_{TECS}$, since it only involves a modification to the nominal total energy controller. Once $\bar{K}_{TECS}$ is validated, Section 7.7 considers the design and analysis of K_S.

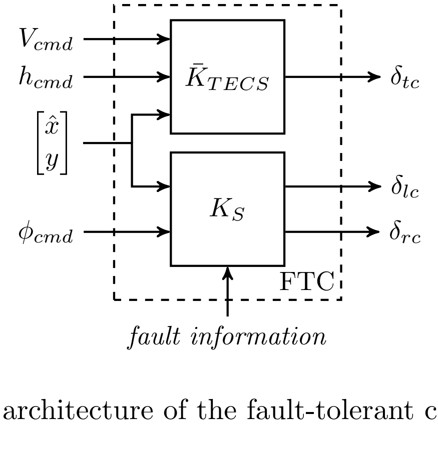


Figure 7.3: The architecture of the fault-tolerant controller (FTC).

7.6 Total Energy Controller

Recall that θ_{cmd} is the primary control variable for regulating the balance between the kinetic and the potential energies. Since $\bar{K}_{TECS}$ does not compute θ_{cmd}, the exchange between the two forms of energy is not actively regulated. Thus, even if the total energy is constant throughout a maneuver, an exchange between the kinetic and the potential energies is possible, and will consequently result in undesired variations in the airspeed V and the altitude h. Indeed, this is exactly what happens during the phugoid mode[2]. This is of concern for the Vireo since its phugoid mode is lightly damped ($\zeta = 0.094$).

In order to alleviate this problem, $\bar{K}_{TECS}$ is designed as shown in Figure 7.4, where G_{lon} models the longitudinal dynamics (Equation (4.1)) and G_t^P models the throttle actuator dynamics (Equation (3.26)). As before, the energy calculation block computes the total energy error ΔE and the balance energy error ΔB. However, a new energy measure called the *mixed energy* error is defined as $\Delta T := \Delta E + w_b \Delta B$, where the weight $w_b \in [-1, +1]$. K_{TE} implements the PI controller $6 \times 10^{-4} + \frac{4\times10^{-5}}{s}$ (same as in K_{TECS}) to drive ΔT to zero using the throttle command δ_{tc}. Thus, the design boils down to selecting w_b such that the phugoid mode damping ratio increases sufficiently.

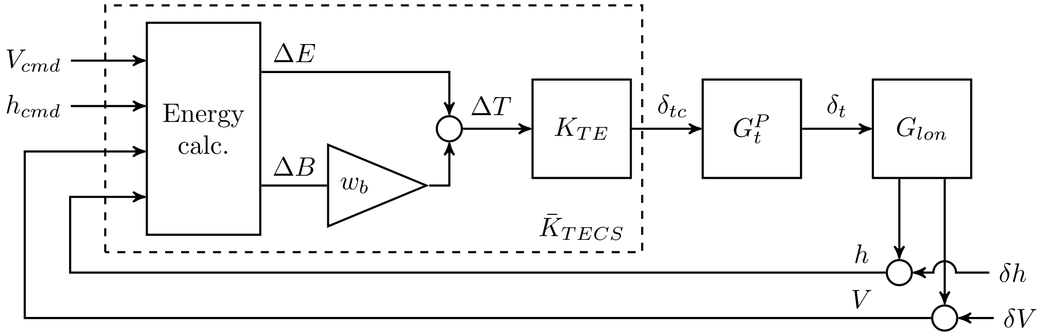


Figure 7.4: The architecture of the fault-tolerant total energy controller $\bar{K}_{TECS}$.

7.6.1 Analysis of Weight Selection

Figure 7.5 shows a root locus plot of the closed-loop phugoid mode as a function of w_b, with $\bar{K}_{TECS}$ linearized at the nominal trim airspeed of 15.4 $\mathrm{m\,s^{-1}}$. The grid lines are

[2] In fact, the total energy of the aircraft remaining constant was one of the key assumptions that enabled Lanchester to derive the first mathematical model of the phugoid mode [19].

drawn in the traditional polar coordinates of constant damping ratios ζ and natural frequencies ω_n. The open-loop phugoid mode (+) has $\zeta = 0.094$ and $\omega_n \approx 0.87$ rad s^{-1}. As seen from the plot, w_b has a much greater impact on ζ than on ω_n. Selecting $w_b = 0$ impacts neither ζ nor ω_n. Thus, *not* modifying the nominal total energy controller is not beneficial, because K_{TE} only drives ΔE to zero. Selecting $w_b < 0$ is also not beneficial because it decreases ζ. Thus, in order to increase the phugoid mode damping ratio, w_b should be selected in the interval $(0, 1]$. Although selecting a large positive value for w_b increases the phugoid mode damping, it comes at the expense of decreased altitude tracking performance. For example, selecting $w_b = 1$ results in $\zeta \approx 0.18$, but implies that $\Delta T = \Delta E + \Delta B = 2\Delta K$, where ΔK is the kinetic energy error (Equation (4.4)). In this case, the $\bar{K}_{TECS}$ does not respond to altitude reference commands.

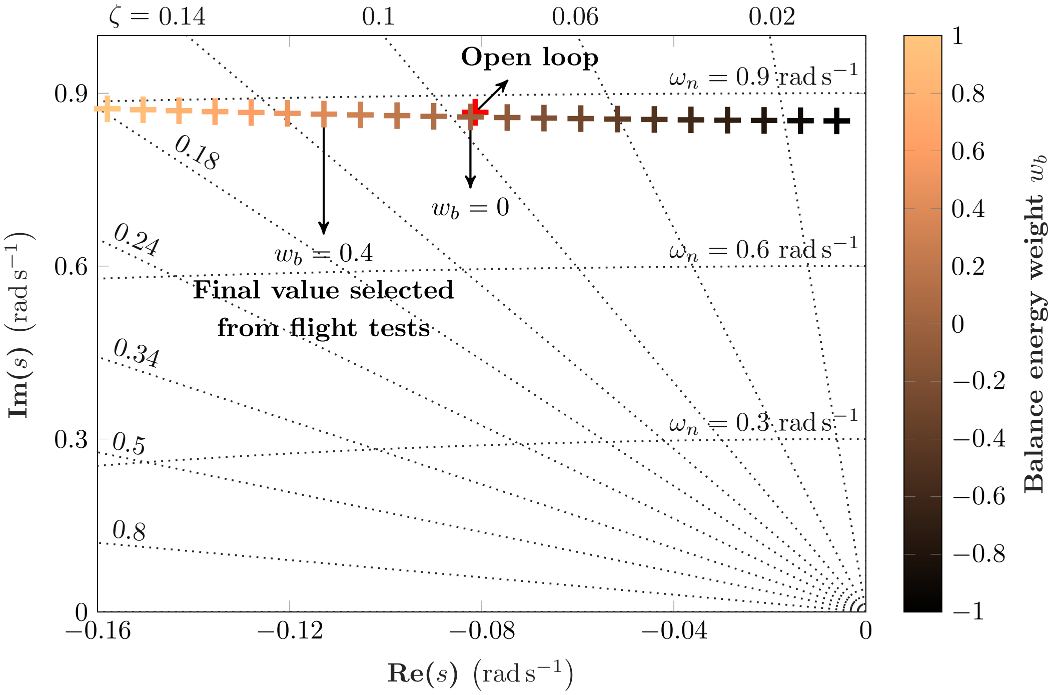


Figure 7.5: Root locus plot of the closed-loop phugoid mode as a function of w_b.

From a flight test, it is determined that $w_b = 0.4$ offers a good trade-off between the phugoid mode damping ratio and the altitude tracking performance. This trade-off is visualized using the output sensitivity function $[\delta V, \delta h]^T \rightarrow [V, h]^T$ associated with the closed-loop shown in Figure 7.4, where δV and δh denote disturbances in the airspeed and the altitude, respectively. Figure 7.6 shows the Bode diagrams of the four channels of the output sensitivity function for $w_b = \{0, 0.4\}$. In both cases, the magnitudes of the $\delta V \to V$ and the $\delta h \to h$ channels are close to 0 dB throughout most of the frequency range. Similarly, the magnitudes of the cross channels $\delta h \to V$ and $\delta V \to h$ both exhibit bandpass behavior near the phugoid mode (0.87 rad s^{-1}). This is expected since the coupling between the airspeed and the altitude (or alternatively, between the kinetic and potential energies) is one of the defining characteristics of the phugoid mode.

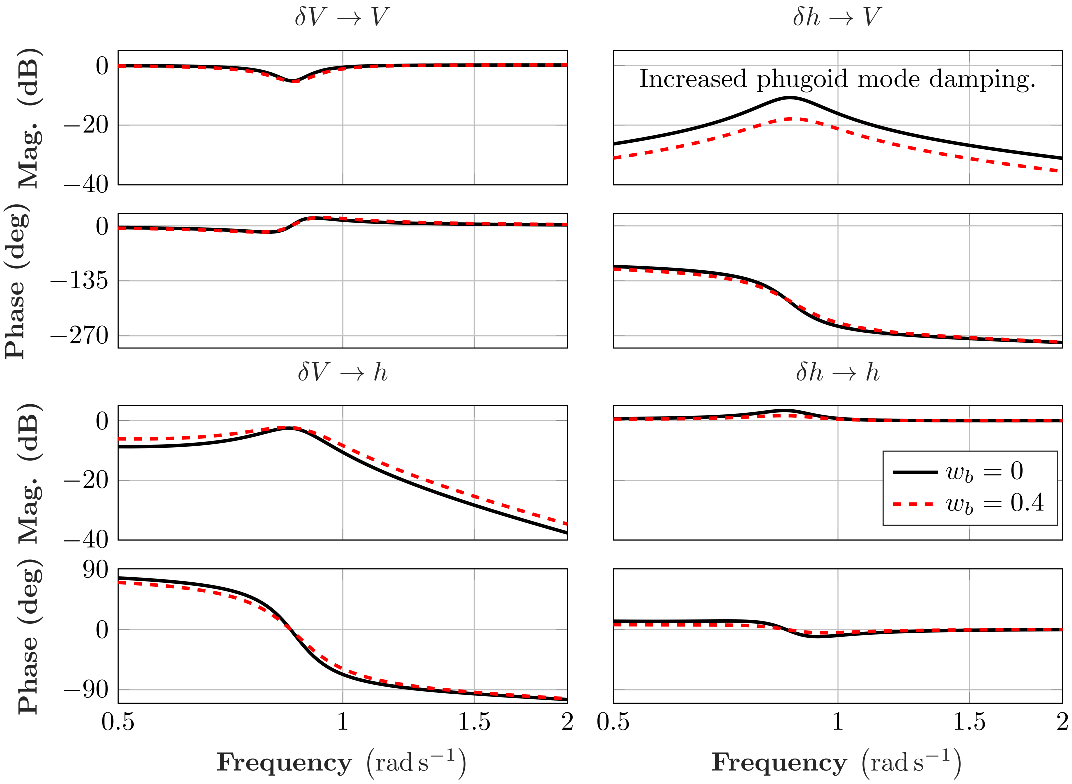


Figure 7.6: The Bode diagrams of the output sensitivity function $[\delta V, \delta h]^T \to [V, h]^T$.

On the other hand, the peak sensitivity in the $\delta h \to V$ channel is 7 dB lower for $w_b = 0.4$ than for $w_b = 0$. Moreover, the phugoid mode damping ratio increases from 0.096 for $w_b = 0$ to 0.13 for $w_b = 0.4$ (Figure 7.5). This benefit comes at the expense of increased sensitivity in the other cross channel $\delta V \to h$. However, the increases in this channel are concentrated at frequencies away from the phugoid mode, and are thus not a major concern. The flight test leading to the selection of $w_b = 0.4$ is presented next.

7.6.2 Flight Test for Weight Selection

Experiments to select w_b are performed during FLT26 of the Vireo on 2017-09-07. Since $\bar{K}_{TECS}$ does not compute θ_{cmd}, no pitch control is provided. However, lateral-directional control is provided by the nominal roll attitude controller K_A (Figure 4.6). The experiment involves selecting a value of w_b (called a test point) and issuing two step changes in the altitude command. Each test point is designed for 60 s, which is approximately 8 times the open-loop phugoid mode time period. Figure 7.7 shows the flight test results. The top-right subplot shows the altitude response, wherein the arrows indicate the time spans of the three test points. The first test point uses $w_b = 0$, and the altitude step command excites the phugoid mode. The second and the third test points use $w_b = 0.2$ and $w_b = 0.4$, respectively. While these two test points exhibit increased phugoid damping, they also exhibit increased overshoot in the altitude.

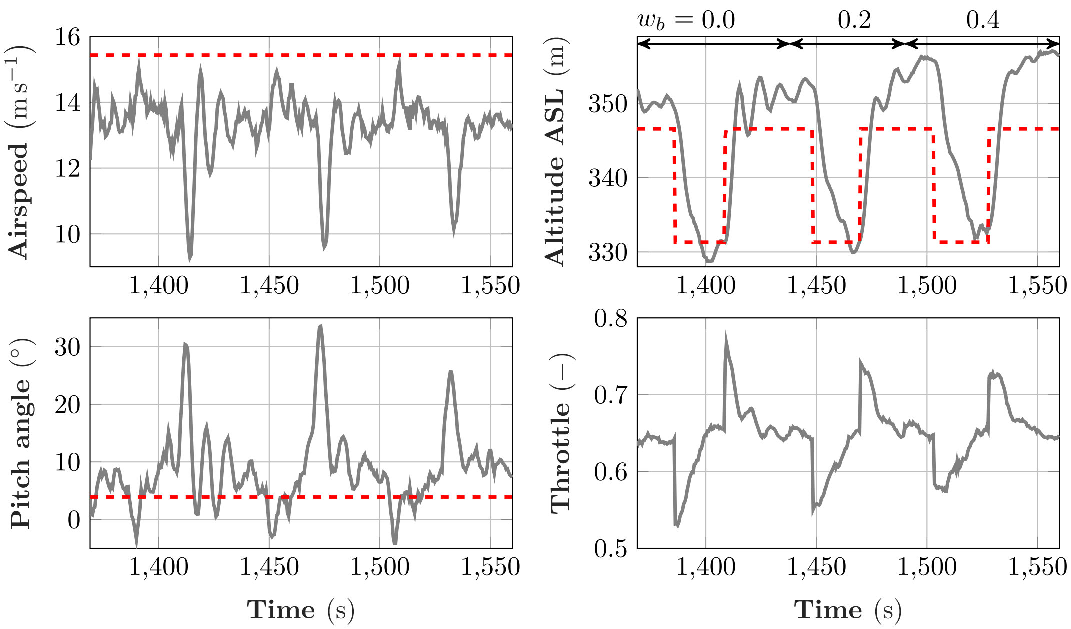


Figure 7.7: FLT26: The weight w_b is tuned using altitude step reference commands.

When considered in isolation, the phugoid excitation that is seen in the altitude response appears benign. However, it also results in large airspeed variations as seen in the top-left subplot. While the stall limit of 12 $\mathrm{m\,s^{-1}}$ (see Section 2.4.1) is breached during the final climb in all three test points, it is breached again at 1423 s in the first test point due to the low phugoid damping. Moreover, by treating the step reference commands as constant altitude disturbances, the cross channel sensitivity $\delta h \rightarrow V$ plot (Figure 7.6) explains why $w_b = 0.4$ elicits the lowest airspeed variations among the three test points. A similar observation applies to the bottom-left subplot, wherein the third test point yields a lower peak in the pitch angle when compared to the previous two test points. The bottom-right subplot, which shows the time history of the throttle command, indicates that the control effort required to track the altitude step progressively decreases as w_b increases. In conclusion, small variations in the airspeed, the pitch angle, and the throttle command are highly desirable when progressing to the more challenging task of controlling the Vireo with a single elevon. Thus, w_b is selected as 0.4 despite the higher overshoot that it produces in the altitude response.

On a related note, the airspeed V and the altitude h exhibit steady-state errors in Figure 7.7 because $\bar{K}_{TECS}$ does not directly regulate the balance energy using θ_{cmd}. Rather K_{TE} drives ΔT to zero (Figure 7.4) using δ_{tc}. For example, during the time span $[1370, 1384]$ s before the start of the experiment, the steady-state errors in V and h are 1.83 $\mathrm{m\,s^{-1}}$ and -3.4 m, respectively. In contrast, $\Delta T = -8.33$ J, which is less than 1% of the mixed energy. Thus, the steady-state errors in the airspeed and the altitude offset each other when computing ΔT. Unfortunately, there does not appear to be a way around this shortcoming, within the scope of the current architecture.

7.7 Single Surface Controller

Figure 7.8 shows the architecture of the single surface controller K_S, which comprises a roll attitude controller K_A and a switch. Depending on the fault information received, the switch sends the aileron command generated by K_A to the operable elevon and commands the stuck elevon to its estimated stuck position. Figure 7.8 shows an instance of the switch for the case of a stuck right elevon. To understand the rationale behind this architecture, first note that the inverse of Equation (3.13) (Section 3.3.2) yields

$\delta_{lc} = \delta_{ec} - \delta_{ac}$ and $\delta_{rc} = \delta_{ec} + \delta_{ac}$, and then note that the elevator command is zero.

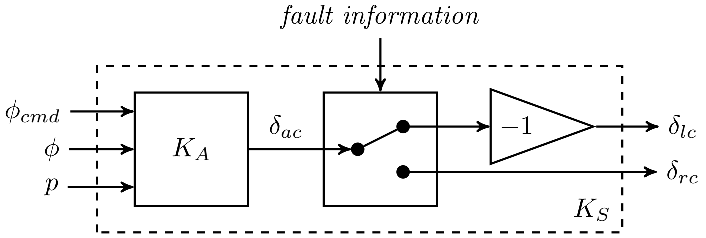


Figure 7.8: The architecture of the single surface controller K_S. Depending on the failure scenario, K_S commands either the left or the right elevon.

This section presents two design approaches for K_A, which is the key component of K_S. The first (Design A) is a repurposed version of the nominal roll attitude controller (Figure 4.6). The second (Design B) is designed using H_∞ synthesis. In addition to tracking ϕ_{cmd} and damping p, the H_∞ synthesis allows for objectives such as disturbance rejection to be included. Design A serves as a baseline for assessing the performance and robustness of the nominal roll attitude controller when only one elevon is operable. Design B seeks to improve upon this baseline. Both the designs use the feedback structure shown in Figure 7.9, where G_{lat} models the lateral-directional dynamics (Equation (4.2)), G_a^L models the actuator dynamics (Equation (3.25)), and $G := G_{lat}G_a^L$. The input $\tilde{d}_1$ and output $\tilde{d}_2 = [\delta\phi, \delta p]^T$ disturbances are used to define the following sensitivities. Note that K_A is partitioned as $[K_{Ar}\ K_{Ay}]$, where K_{Ar} only includes the input ϕ_{cmd} and K_{Ay} only includes the inputs $[\phi_n, p_n]^T$.

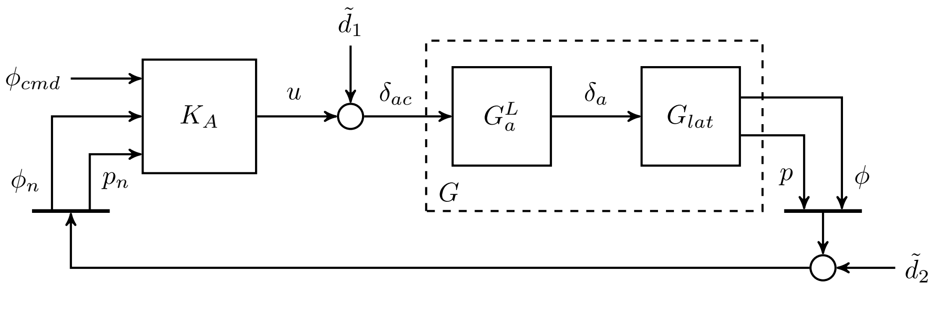


Figure 7.9: The K_A block of K_S is designed using the lateral-directional model.

- The input sensitivity S_i is the transfer function $\tilde{d}_1 \to \delta_{ac}$.

- The output sensitivity S_o is the transfer function $\tilde{d}_2 \rightarrow [\phi_n, p_n]^T$.
- The input complementary sensitivity T_i is the transfer function $\tilde{d}_1 \rightarrow u$.
- The output complementary sensitivity T_o is the transfer function $\tilde{d}_2 \rightarrow [\phi, p]^T$.
- The disturbance sensitivity $GS_i\,(= S_oG)$ is the transfer function $\tilde{d}_1 \rightarrow [\phi, p]^T$.
- The control sensitivity $K_{Ay}S_o\,(= S_iK_{Ay})$ is the transfer function $\tilde{d}_2 \rightarrow u$.

These sensitivity definitions use a negative sign because K_A is in positive feedback with G in Figure 7.9. The corresponding algebraic constraints are $S_i - T_i = I$ and $S_o - T_o = I$. While this is non-standard, e.g. when compared to textbooks such as [55], it does not affect the inferences that are drawn from these functions. Section 7.7.3 uses these sensitivities to compare the performance and robustness of the two designs.

7.7.1 Design A

Design A of K_S is a repurposed version of the nominal roll attitude controller (Figure 4.6), i.e. a PI controller K_{RT} tracks ϕ_{cmd} and a P-controller K_{RD} damps p. Despite having the same architecture as Figure 4.6, its gains are re-tuned using flight tests, guided by the performance and robustness targets given in Table 4.1. Since θ_{cmd} is not tracked by K_S, an additional tuning objective is to prevent excessive pitch excitations. The competing objectives lead to less aggressive tracking of ϕ_{cmd} as compared to the nominal controller. The final gains ($K_{RD} = -0.06$ and $K_{RT} = -0.29 - \frac{0.0573}{s}$) correspond to a bandwidth of 1.5 $\mathrm{rad\,s^{-1}}$ in the $\phi_{cmd} \rightarrow \phi$ channel. The loop transfer function $e_\phi \rightarrow \phi$ (Figure 4.6) has the gain margins ±13 dB, the phase margins ±85.4°, and the delay margin 0.98 s. In addition, at 15.4 $\mathrm{m\,s^{-1}}$, the closed-loop roll subsidence and spiral modes have natural frequencies of 21 $\mathrm{rad\,s^{-1}}$ and 0.22 $\mathrm{rad\,s^{-1}}$, respectively. The dutch roll mode has a damping ratio of 0.35 (Level-1 flying quality) and a natural frequency of 3.6 $\mathrm{rad\,s^{-1}}$. Finally, Design A is validated using flight tests (Section 7.9).

7.7.2 Design B

Since Design A yields promising flight test results, Design B retains its low frequency behavior but improves upon its robustness. Specifically, Design B of K_S is based on

H_∞ synthesis. Given a LTI system $H(s)$ with the input $d \in \mathbb{R}^{n_d}$ and output $e \in \mathbb{R}^{n_e}$, its H_∞ norm is defined as:

$$\|H(s)\| := \sup_{\omega} \bar{\sigma}\left(H(j\omega)\right) = \sup_{0 \neq d \in L_2^{n_d}} \frac{\|e\|_2}{\|d\|_2}, \tag{7.3}$$

where $\bar{\sigma}(\cdot)$ denotes the largest singular value. The second equality states that the H_∞ norm is equal to the induced L_2 norm, or largest gain, of the system $H(s)$.

The H_∞ norm provides a unified framework for expressing the different, and often competing, design objectives. Particularly, the generalized plant P comprises the plant to-be-controlled and the frequency-dependent weights that codify the design objectives. The *weighted closed-loop* is defined as $F_L(P, K)$, where K is the as-of-yet undetermined controller. The design problem then reduces to selecting K so that the closed-loop is stable and $\|F_L(P, K)\| < \gamma$ for some $\gamma > 0$. This is a convex optimization problem that is solved either via coupled Riccati equations [154] or via LMIs [116]. Matlab's Robust Control Toolbox implements algorithms for both approaches [155]. Since the solution hinges on bounding the single number $\|F_L(P, K)\|$, the designer exercises control over the process via the choice of P. The reader is referred to [55, 90] for more details.

Control design objectives

The controller K_A resulting from the H_∞ synthesis should:

1. track a roll angle command ϕ_{cmd} with a bandwidth of 1.4 $\mathrm{rad\,s^{-1}}$,
2. provide a closed-loop dutch roll mode damping ratio of at least 0.38 (Level-1 FQ),
3. ensure zero steady-state error in tracking ϕ_{cmd},
4. penalize excessive control effort in δ_{ac} above 1.4 $\mathrm{rad\,s^{-1}}$, and
5. be robust to disturbances at the plant input $\tilde{d}_1$ and output $\tilde{d}_2$ (see Figure 7.9).

The first and the second objectives are drawn from Table 4.1. The third objective seeks to mimic the integrator in Design A. The first three objectives ensure that Design B performs on par with Design A. Hence, the fourth and the fifth objectives differentiate the designs. In particular, the fourth objective ensures that Design B produces smaller elevon deflections, and thus smaller pitch excitations, as seen in the flight test results (Section 7.9). The fifth objective increases the robustness of Design B (Section 7.8).

Generalized plant

The design objectives are codified using weights, which are interconnected with $G_{lat}G_a^L$ to form a generalized plant as shown in Figure 7.10. The generalized plant P consists of all the systems shown in Figure 7.10, except for the as-of-yet undetermined controller $\bar{K}_A$, enclosed by the dashed box. Although the synthesis results in $\bar{K}_A$, the final controller of interest is K_A, which comprises $\bar{K}_A$, the integrator $\frac{1}{s}$, and the sum junction, as indicated by the shaded box. The integrator ensures zero-steady state error in ϕ. P has three generalized disturbance inputs (d_1, d_2, d_3), one control input (u), three generalized error outputs (e_1, e_2, e_3), and three measurement outputs (e_ϕ, its integral, p_n).

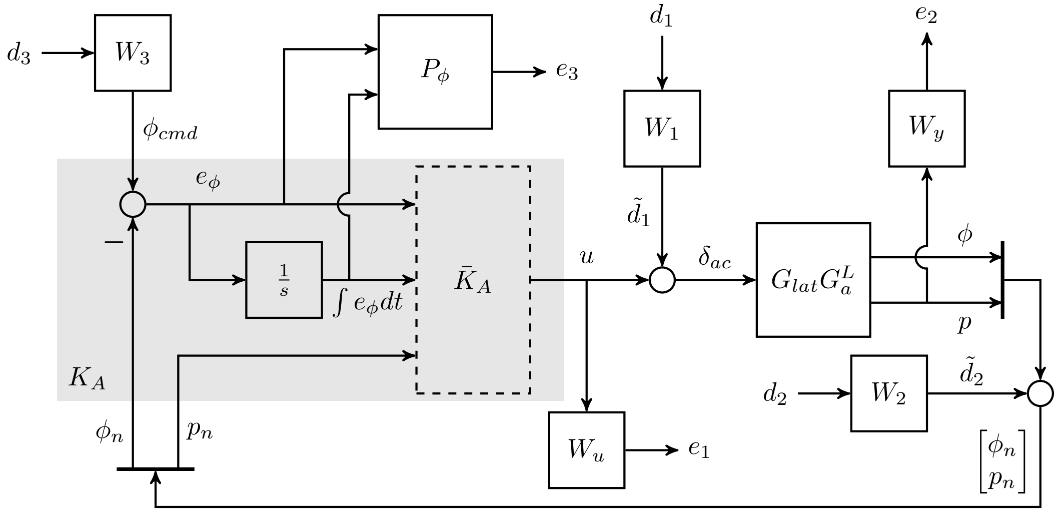


Figure 7.10: The generalized plant for Design B of the single surface controller.

The weight W_1 relates disturbances at the plant input $\tilde{d}_1$, e.g. atmospheric turbulence, to its normalized counterpart d_1. The weight W_2 relates disturbances at the plant output $\tilde{d}_2$, e.g. sensor noise, to its normalized counterpart d_2. W_1 and W_2 together codify the fifth objective and control the relation between the input and output margins. The weight W_3 relates ϕ_{cmd} to its normalized counterpart d_3. The noisy roll rate p_n is directly fed back to $\bar{K}_A$, whereas the noisy roll angle ϕ_n is first used to compute the tracking error $e_\phi = \phi_{cmd} - \phi_n$, which is then fed to $\bar{K}_A$ along with its integral. The weight P_ϕ penalizes large tracking errors. An appropriate penalty on e_ϕ codifies the first objective, while any non-zero penalty on its integral codifies the third objective. The weight W_y penalizes large roll rates and codifies the second objective. Finally, the weight W_u penalizes excessive control effort and codifies the fourth objective.

Weight selection and synthesis

The weighted closed-loop $F_L\left(P, \bar{K}_A\right)$ has the inputs (d_1, d_2, d_3) and outputs (e_1, e_2, e_3). The synthesis involves a search for $\bar{K}_A$ such that $\left\|F_L\left(P, \bar{K}_A\right)\right\|$ is minimized. This is an iterative process that involves weight selection and tuning. Table 7.2 lists the final values of all the weights, along with their interpretations. The goodness of the resultant controller is evaluated using: the nine transfer functions that compose $F_L\left(P, \bar{K}_A\right)$, the input and output sensitivities, the closed-loop step response, the damping ratio of the closed-loop dutch roll mode, the robustness margins, and the degree to which the low frequency behavior of K_A matches that of Design A.

Table 7.2: The final weights selected for the H_∞ synthesis.

Weight	Final value	Weight interpretation:
W_1	$6\,(\pi/180)$	Aileron disturbance at plant input.
W_2	$\begin{bmatrix} 5 & 0 \\ 0 & 50 \end{bmatrix} (\pi/180)$	Disturbance at the plant output.
W_3	$150\,(\pi/180)$	Roll angle command.
P_ϕ	$\begin{bmatrix} 0.4 & 1.6 \end{bmatrix}$	Inverse of the desired sensitivity function from ϕ_{cmd} to e_ϕ.
W_y	$(1/250)\,(180/\pi)$	Inverse of the desired roll rate.
W_u	$\frac{60.31(s+3778)(s+38.73)(s+8.762)(s+0.897)}{(s+2857)(s+872.2)(s+28.08)(s+1.887)}$	Inverse of the desired aileron commands across frequency.

Let S_ϕ and $\bar{S}_\phi$ denote the sensitivity function $\phi_{cmd} \rightarrow e_\phi$ and its upper bound, respectively. In order to select P_ϕ, the bound is first chosen as $\bar{S}_\phi = \frac{k_\phi s}{s+p_\phi}$. This choice has zero DC gain and thus satisfies the third control design objective. Given this bound, e_ϕ should be weighted using $\bar{S}_\phi^{-1} = k_\phi^{-1} + p_\phi k_\phi^{-1} s^{-1}$. However, since the generalized plant (Figure 7.10) contains an integrator, P_ϕ is simply set equal to $\begin{bmatrix} k_\phi^{-1} & p_\phi k_\phi^{-1} \end{bmatrix}$. The pole p_ϕ is selected as 4 $\mathrm{rad\,s^{-1}}$ for good tracking performance. This is close to the dutch roll mode $\left(4.1 \text{ rad s}^{-1}\right)$ but is sufficiently lower than the short period mode $\left(14.5 \text{ rad s}^{-1}\right)$. The gain k_ϕ $\left(= \left\|\bar{S}_\phi\right\|\right)$ is selected as 8 dB for good robustness. The left side of Figure 7.11 shows that S_ϕ, which is bounded by $\bar{S}_\phi$, has zero DC gain (indicates zero steady-state error), a peak gain of 2.4 dB at 6.9 rad s^{-1} (indicates good robustness), and a bandwidth of 1.5 rad s^{-1} (satisfies first control design objective).

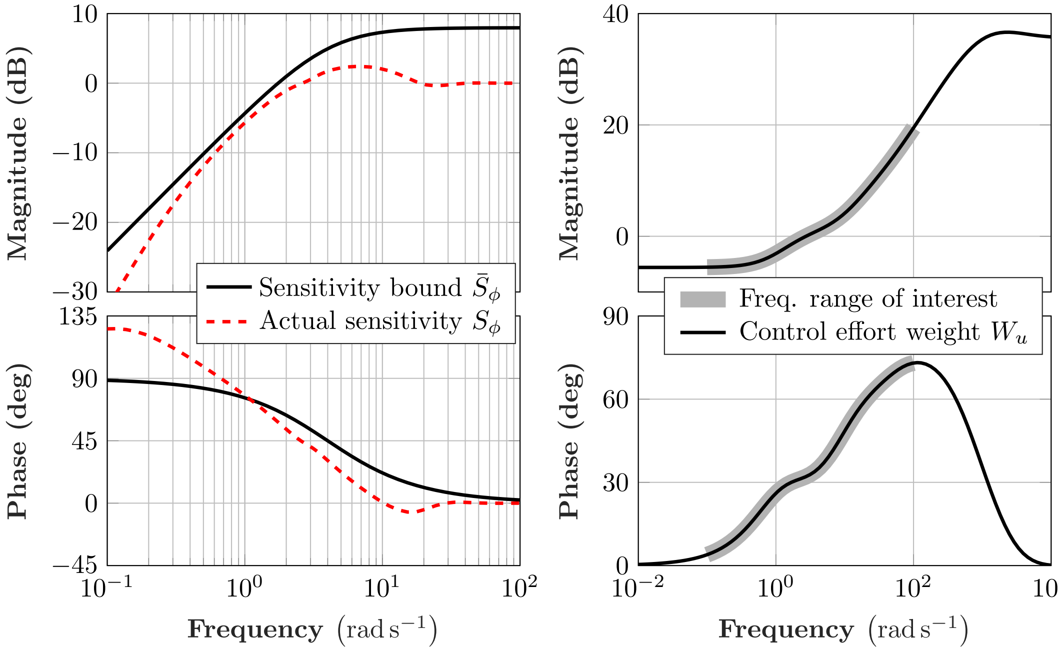


Figure 7.11: Left: The Bode diagrams of the sensitivity function S_ϕ from ϕ_{cmd} to e_ϕ and its upper bound $\bar{S}_\phi$. Right: The Bode diagram of the control effort weight W_u.

The weight W_u (right side of Figure 7.11) is selected to penalize excessive control effort in δ_{ac}. Thus, it has a small DC gain (−5.6 dB) and a large high frequency gain (35.6 dB). The magnitude of W_u at a given frequency may be interpreted as the inverse of the desired aileron command at that frequency. For example, the desired δ_{ac} at the actuator bandwidth $\left(57 \text{ rad s}^{-1}\right)$ is 10° and at high frequencies is 1°. W_u is selected as a fourth-order system (Table 7.2) to ensure that its gain rapidly increases with frequency. Lower order weights, while being simpler, unfortunately do not provide such increases. Figure 7.11 highlights the frequency range pertinent to this design problem.

Through iterative tuning, the remaining weights W_1, W_2, W_3, and W_y are selected to normalize the nine individual transfer functions that compose $F_L\left(P, \bar{K}_A\right)$ to a peak gain of around 0 dB. The weights may be interpreted using Table 7.2, e.g. the controller should treat a 6° aileron disturbance on par with a 5° roll angle disturbance and a $50^\circ\,\mathrm{s}^{-1}$ roll rate disturbance. Similarly, a 150° roll angle command should elicit a peak roll rate on the order of $250^\circ\,\mathrm{s}^{-1}$, i.e. a rise time of approximately 0.6 s.

7.7.3 Comparison of Designs A and B

Figure 7.12 shows the Bode diagrams of the two designs. The roll angle tracking performance of Design B is comparable to that of Design A at low frequencies. However, its gain rolls off above $2.7\ \mathrm{rad\,s^{-1}}$ because the control effort weight W_u has its zero crossover at this frequency. At the dutch roll mode $\left(4\ \mathrm{rad\,s^{-1}}\right)$, Design B is more than 3 dB lower than Design A. Similarly, at the short period mode $\left(14.5\ \mathrm{rad\,s^{-1}}\right)$, Design B is more than 9 dB lower than Design A. The roll rate channel largely follows a proportional-type behavior at low frequencies and rolls off at $42\ \mathrm{rad\,s^{-1}}$, thus attenuating sensor noise.

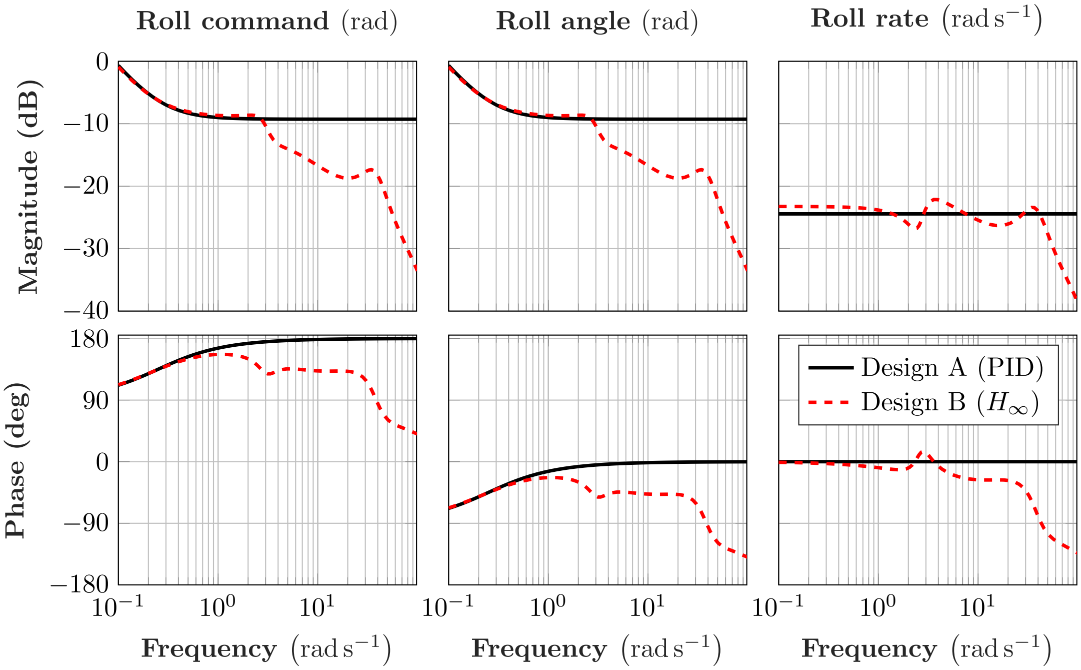


Figure 7.12: The Bode diagrams of the two designs of the single surface controller. The input channels are labeled at the top of each plot. The output is the aileron command.

Design B increases the dutch roll mode damping ratio to 0.3 (Level-1 flying quality) from its open-loop value of 0.13. This increase is visualized via the disturbance sensitivity G_pS_i from the aileron disturbance $\tilde{d}_1$ to the roll rate p, as shown on the left side of Figure 7.13. The closed-loop G_pS_i is lower for both the designs, when compared to the open-loop, near the dutch roll mode $\left(4 \text{ rad s}^{-1}\right)$. However, due to the constraints imposed by the Bode sensitivity integral [55], this benefit comes at the expense of higher disturbance sensitivity in the frequency range $[10, 30]$ rad s^{-1}.

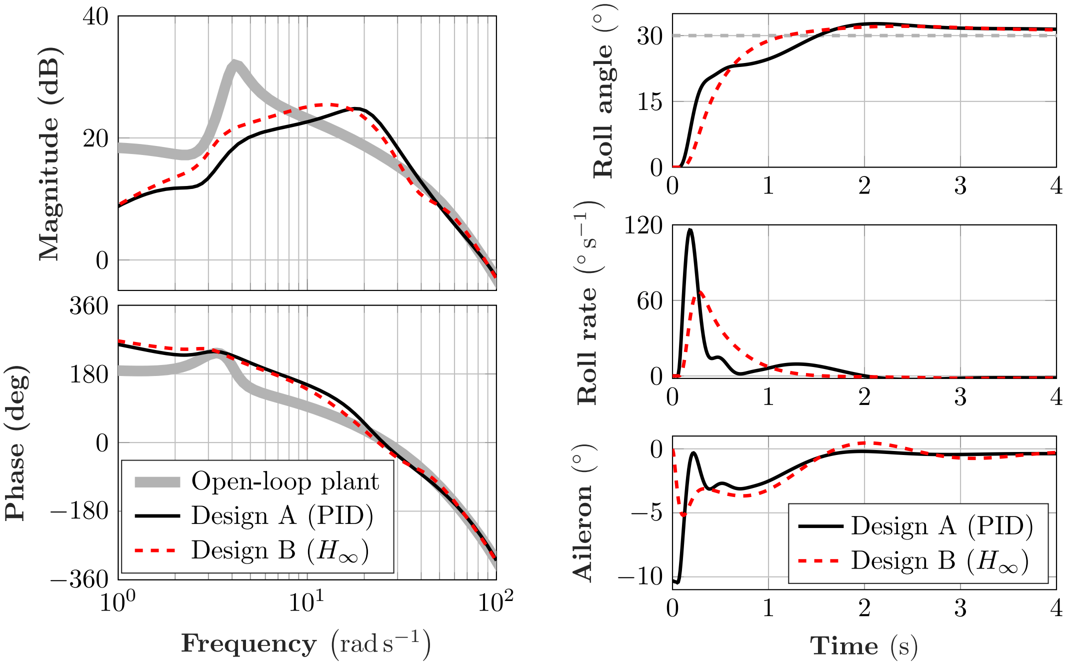


Figure 7.13: Left: The Bode diagrams of the open-loop and the closed-loop disturbance sensitivities G_pS_i from the aileron disturbance $\tilde{d}_1$ to the roll rate p. Right: The closed-loop aircraft response to a 30° step command in the roll angle.

In the time-domain, Designs A and B are compared using their closed-loop responses to a 30° step command in the roll angle. As seen on the right side of Figure 7.13, Design B exhibits better performance in all the variables. With Design A, the roll angle exhibits a rise time (from 10% to 90% of the steady-state value) of 1.1 s, a settling time (within 2% of the steady-state value) of 7.1 s, and an overshoot of 9%. With Design B, the roll angle exhibits a rise time of 0.63 s, a settling time of 7.2 s, and an overshoot of 7.1%.

Further, Design B elicits a maximum roll rate of 67° s^{-1}, which is much lower compared to the 116° s^{-1} seen with Design A. Similarly, Design B elicits a smaller peak aileron command of $-5.2°$ compared to the $-10.4°$ seen with Design A.

The elevon commands issued by the two designs inevitably excite the longitudinal dynamics. Figure 7.14 quantifies their impact on the longitudinal modes through the largest singular value of the sensitivity function from the lateral disturbances $[\delta\phi, \delta p]^T$ to the longitudinal variables $[\theta, q]^T$. The peak magnitude near the short period mode $\left(14.5 \text{ rad}\,\mathrm{s}^{-1}\right)$ is 12 dB for Design A and 3.3 dB for Design B. The phugoid mode is unaffected. The short period mode attenuation, which is a consequence of the high gain of W_u above 2.7 rad s^{-1}, is useful because (θ, q) are not in feedback with the fault-tolerant controller (Section 7.4). However, the attenuation is only with respect to lateral disturbances, i.e. there is no attenuation of longitudinal disturbances. This is a limitation of the current architecture and, perhaps, of this particular control problem.

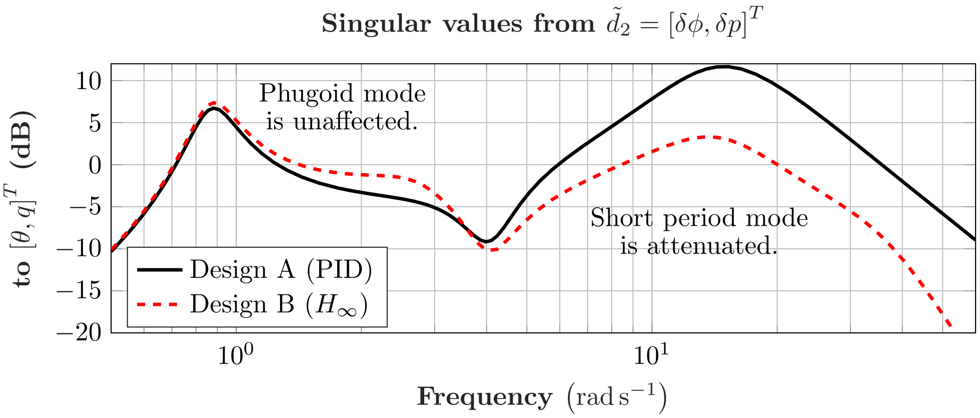


Figure 7.14: The largest singular value of the sensitivity function from the lateral disturbances $[\delta\phi, \delta p]^T$ to the longitudinal variables $[\theta, q]^T$ for Designs A and B.

Finally, Figure 7.15 compares Designs A and B using the input and output sensitivity functions defined in Section 7.7. The signal names in the titles and y-labels correspond to the closed-loop block digram shown in Figure 7.9. The higher robustness of Design B, when compared to Design A, is most prominent in the plots of S_o and $K_{Ay}S_o$. In particular, the peak gain of the largest singular value of S_o is 7.7 dB for Design B and 10.2 dB for Design A. The largest singular value of S_o is also lower for Design B near the dutch roll mode frequency of 4 rad s^{-1} and for all higher frequencies, implying that Design B has better roll attitude tracking and roll rate damping. The magnitude of $K_{Ay}S_o$ for Design B is slightly higher than that of Design A in the range $[1, 3]$ rad s^{-1}, but is significantly lower above 4 rad s^{-1}, indicating its good noise rejection property.

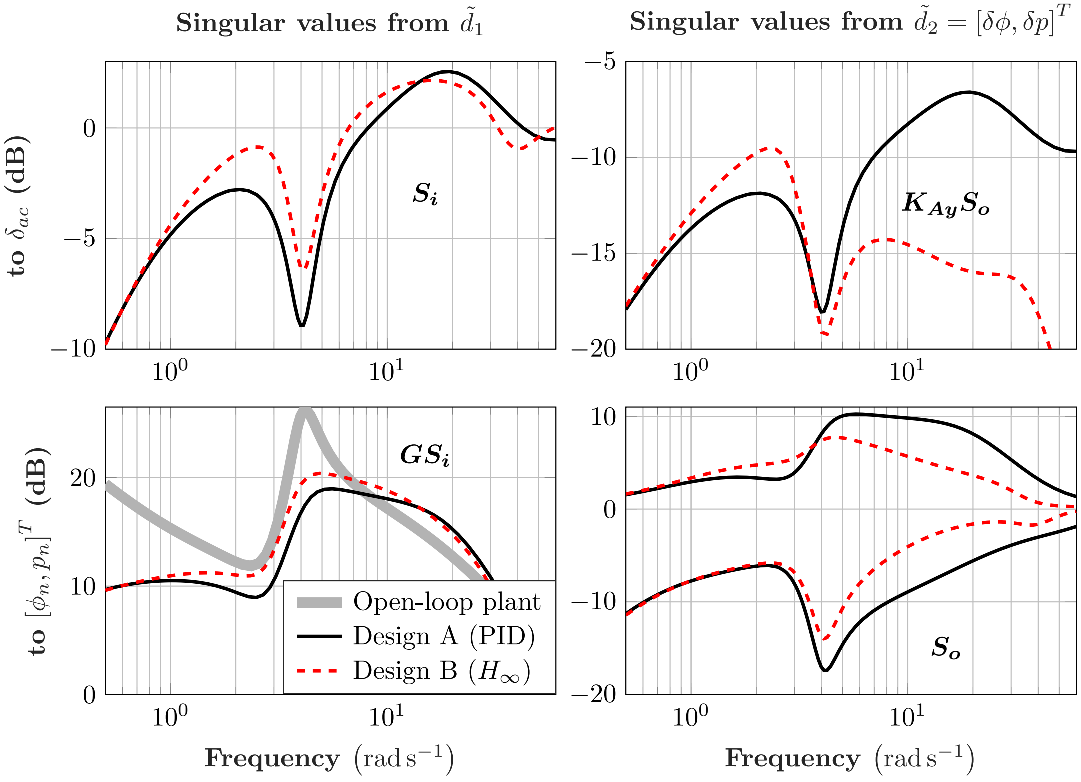


Figure 7.15: The singular value diagrams of the input sensitivity S_i, the control sensitivity $K_{Ay}S_o$, the disturbance sensitivity GS_i, and the output sensitivity S_o functions associated with the two designs of the single surface controller. The titles specify the input channels and the y-labels specify the output channels.

On the other hand, the higher robustness of Design B is less apparent in the plots of S_i and GS_i. In particular, the peak gain of S_i is 2.12 dB for Design B and 2.52 dB for Design A. The magnitude of S_i for Design B is slightly higher than that of Design A in the range $[1, 14]$ rad s^{-1} and slightly lower above 14 rad s^{-1}. Similar observations apply to the plot of GS_i, wherein the benefits of Design B are apparent only above 14 rad s^{-1}. In comparing the two designs to the open-loop plant, it is evident that feedback control is essential to reducing the impact of input disturbances on the lateral motion.

7.7.4 Saturations

The autopilot includes saturation blocks to bound various commands. As in the nominal case, the guidance law saturates V_{cmd} in the interval $[13, 18]$ $\mathrm{m\,s^{-1}}$ to prevent the Vireo from breaching the stall and the high speed limits. Nominally, the guidance law saturates ϕ_{cmd} in the interval $[-35, +35]\,^\circ$. However, when one of the elevons is failed, the guidance law uses asymmetric saturation bounds. In particular, when the right elevon is failed, the saturation bounds for ϕ_{cmd} are $[-35, +20]\,^\circ$. Conversely, when the left elevon is failed, the saturation bounds for ϕ_{cmd} are $[-20, +35]\,^\circ$. The motivation for using asymmetric bounds arises from the preferred pitch direction of the aircraft. In particular, when one of the elevons is failed, it is preferred that the aircraft pitches nose-up rather than nose-down so as not to lose altitude. Hence, the larger of the two bounds is used for the roll direction that results from a trailing-edge up deflection of the operable elevon. In addition, the throttle command is bounded within $[0, 1]$ and the operable elevon deflection command is bounded within $[-20, +20]\,^\circ$.

7.8 Robustness Analysis

A disk margin analysis is conducted to assess the robustness of $\bar{K}_{TECS}$ and K_S separately, as shown in Table 7.3. For simplicity, only the input-to-plant loop transfer function L_i associated with each controller is analyzed. A good rule of thumb is a disk margin of at least 0.4 on L_i, which corresponds to a minimum disk gain margin of $[0.71, 1.67]$, a minimum disk phase margin of $\pm 23^\circ$, and a maximum H_∞ norm of 8 dB on the input-to-plant sensitivity function S_i [72]. The crossover frequency of L_i is a good measure of the bandwidth. $\bar{K}_{TECS}$ achieves good disk gain and phase margins, and a high time delay margin owing to the low critical frequency. Designs A and B of K_S achieve approximately the same bandwidth, which is a result of their similar characteristics at low frequencies (Figure 7.12). However, when considering $\|S_i\|$ and the disk margins, it is evident that Design B is more robust. Design B also has a higher time delay margin (0.042 s) than Design A (0.032 s). All the loops achieve the minimum desired disk gain and phase margins and the maximum desired $\|S_i\|$.

Table 7.3: The disk margins of the loops associated with the components of the fault-tolerant controller.

Controller	**Input-to-plant loop transfer func.** L_i	**Loop crossover** $(\mathrm{rad\,s^{-1}})$	$\lVert S_i \rVert$ (dB)	**Disk gain margin** $(-)$	**Disk phase margin** $(°)$	**Critical frequency** $(\mathrm{rad\,s^{-1}})$
Total energy controller $\bar{K}_{TECS}$	Break loop at δ_{tc} in Figure 7.4	0.062	2.87	$[0.47, 2.15]$	±40	0.06
Single surface controller K_S: Design A	Break loop at δ_{ac} in Figure 7.9	2.5	5.87	$[0.5, 2]$	±37	20
Single surface controller K_S: Design B	Break loop at δ_{ac} in Figure 7.9	2.3	4.9	$[0.43, 2.3]$	±43.4	18

7.9 Flight Test Results

$\bar{K}_{TECS}$ is tuned during FLT26 (Section 7.6.2). Designs A and B of K_S work satisfactorily in flight and are not tuned further. Thus, this section presents flight test data to validate the fault-tolerant controller. In this regard, Design A of K_S and $\bar{K}_{TECS}$ both use PID controllers and are easy to implement in real-time. On the other hand, Design B has some implementation considerations that are discussed in Appendix B.

7.9.1 Scope of the Demonstrations

Section 7.3 formulated the fault-tolerant control problem by basing it on a mission involving precision agriculture. The scope of the demonstrations presented in this section is limited in two important ways. First, the demonstrations do not implement a fault diagnosis algorithm. The fault is injected from the ground station and is used to directly reconfigure the guidance and control laws. Second, the UAS operator uses a priori knowledge about the airfield and the prevailing winds in order to select the target landing spot. Thus, the demonstrations are scoped to specifically validate the fault-tolerant controller, and consist of the following sequence of steps.

1. After a manual takeoff, the pilot takes the Vireo up to altitude and turns on the autopilot. By default, the nominal guidance and control laws are engaged.

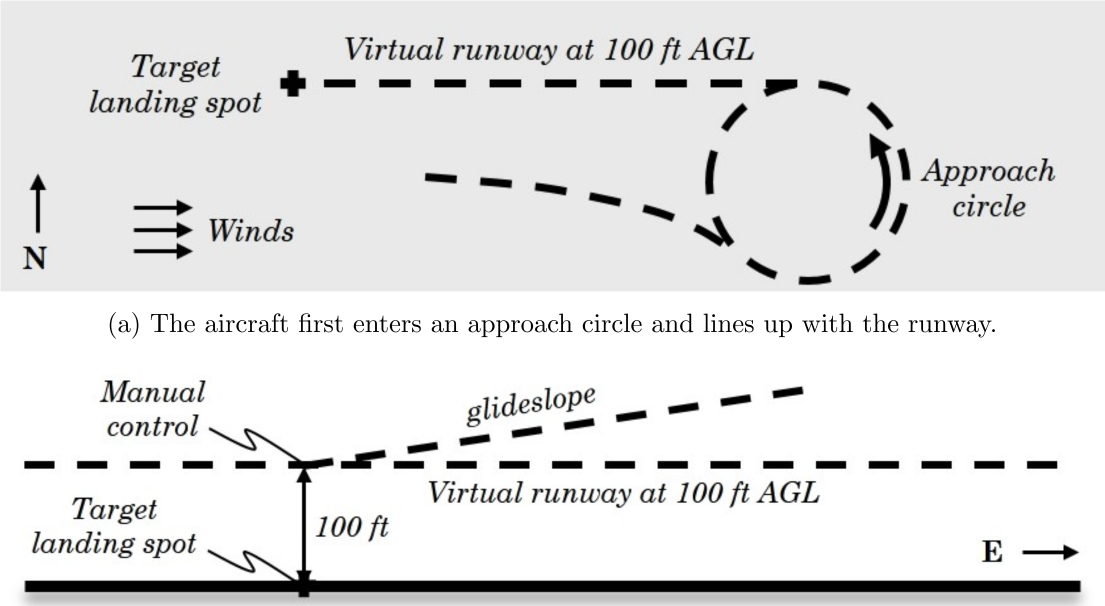
2. The Vireo enters a circle hold at constant altitude and constant airspeed.

3. A stuck fault is injected from the ground station in one of the elevons, causing a reconfiguration of the guidance and control laws.

4. The Vireo continues to fly the circle hold for a few minutes.

5. The Vireo completes an autonomous landing in the presence of the stuck elevon.

The highlight of the demonstrations is the closed-loop aircraft performance during the autonomous landing (autoland). The autoland is a mission manager feature of `AuraUAS` and consists of two main phases as shown in Figure 7.16. The aircraft first enters an approach circle and lines up with the runway leading up to the target landing spot. Upon exiting the approach circle, the aircraft tracks a glideslope until it reaches a point that is located 100 ft above the target landing spot. At this point, the pilot turns off the fault, takes manual control of the aircraft, and attempts a go-around. Although the autopilot is capable of tracking the glideslope all the way to the ground, it will result in a hard landing. In particular, a final flare is required to rapidly decrease the sink rate of the aircraft prior to touchdown. However, there does not appear to be a way to obtain the final flare using the single control surface, or the throttle, while also maintaining wings-level flight. By taking manual control, the pilot ensures a gentler landing. This reduces the amount of repair work required to prepare the aircraft for the next demonstration. If this were a real-life aborted mission, the hardness of the landing would be immaterial as long as the landing spot is far away from people and property.



(a) The aircraft first enters an approach circle and lines up with the runway.

(b) Upon exiting the approach circle, the aircraft tracks a glideslope until it reaches a point that is located 100 ft above the target landing spot. The pilot takes manual control at this point.

Figure 7.16: A schematic representation of the two phases of the autoland mode.

Since the airfield is shorter along the North-South direction, all the flights use a East-West virtual runway. The prevailing winds during the flight test dictate the landing direction, e.g. Westerly in Figure 7.16a. The location and radius of the approach circle are selected to avoid obstacles such as trees and poles. All demonstrations use a 100 m-radius approach circle. The approach speed is fixed by the trim speed corresponding to the stuck elevon deflection. For trailing edge up faults, this will typically lie between the stall speed $\left(12 \text{ m s}^{-1}\right)$ and the nominal cruise speed $\left(15.4 \text{ m s}^{-1}\right)$. The glideslope is specified based on the gliding performance of the Vireo. In particular, the power-off glideslope is approximately 8°, when computed using the lift-to-drag ratio of the Vireo at the nominal cruise airspeed [10]. Since this is too steep, all demonstrations use a glideslope of 6°, which is attained using a small amount of throttle. The elevation of the virtual runway is specified as 100 ft AGL so as to allow the pilot enough time to attempt a go-around. The airfield is located roughly 256 m above mean sea level (ASL).

7.9.2 Summary of Flight Demonstrations

A total of six flights are conducted wherein the right elevon is failed at 5° trailing edge up. Table 7.4 summarizes the flights and indicates whether Design A or B is active. The wind speed and direction are from the National Weather Service[3]. The battery used in the Vireo (see Table 2.1) provides for a total flight time of around 10 min. On average, the fault tolerant controller is active for about 9 min in each flight. An elevated autoland is attempted, and successfully completed, in four of the six flights.

Table 7.4: Summary of flight tests conducted using the single surface controller.

FLT	Wind speed & direction	Fault mode	FTC Controller		Autoland
			Design	Active time	
27	4.5 m s^{-1} from SW	Right elevon stuck at 5° trailing edge up	A	9.6 min	Yes
28	6.7 m s^{-1} from WWN	Right elevon stuck at 5° trailing edge up	A	9.1 min	Yes
29	6.7 m s^{-1} from NW	Right elevon stuck at 5° trailing edge up	A	8.6 min	Not attempted
30	6.7 m s^{-1} from NW	Right elevon stuck at 5° trailing edge up	A	7.7 min	Not attempted
34	2.7 m s^{-1} from S	Right elevon stuck at 5° trailing edge up	B	8.0 min	Yes
35	2.7 m s^{-1} from S	Right elevon stuck at 5° trailing edge up	B	9.6 min	Yes

[3] https://www.wunderground.com/weather/us/mn/kmsp

The next two subsections present the results of FLT27 and FLT35. Appendix D presents the results of the remaining flights from Table 7.4. Appendix C lists short descriptions of all the flights conducted to date. In all the time history plots, the dashed line denotes the autopilot command and the solid line denotes the aircraft response.

7.9.3 FLT27: Autoland Using Design A

An autoland demonstration using Design A of the single surface controller and the fault-tolerant total energy controller is conducted during FLT27 on 2017-09-07. Shortly after the manual takeoff, the pilot takes the Vireo up to 300 ft AGL and engages the autopilot at 746 s. Figure 7.17 shows the ground track of the Vireo from the autopilot engagement ($*$) until the final landing ($\diamond$). The ground track is plotted in a local North-East reference frame, whose origin coincides with the target landing spot chosen by the UAS operator prior to the start of the flight. The various events occurring in the flight are annotated on the ground track and are explained one-by-one.

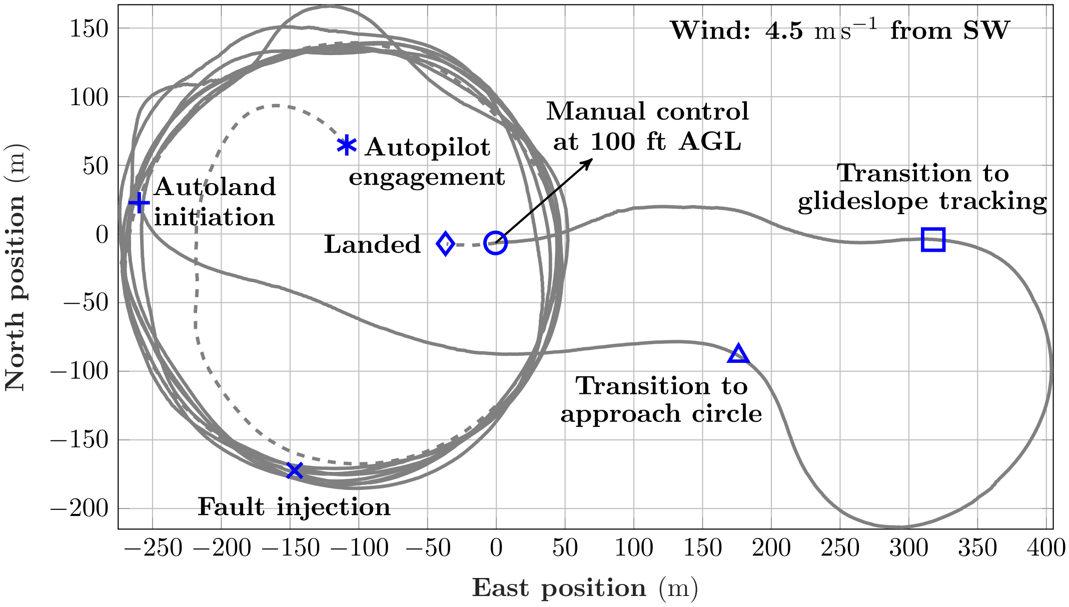


Figure 7.17: FLT27: The ground track of the Vireo indicating different flight events. The solid line indicates the times when the single surface controller is active.

The autopilot engagement ($*$) activates the nominal controller, which takes the Vireo into a left banked circle hold at an altitude of 300 ft AGL, an airspeed of 15.4 $\mathrm{m\,s^{-1}}$, and a turn radius of 150 m. The center of the circle $(-110, -17.5)$ m is automatically set by the flight computer to the point where the airplane climbs past the altitude of 150 ft AGL. The Vireo traces a counterclockwise circle when viewed in the North-East plane. This initial phase, wherein the nominal controller is active, is denoted by the dashed line starting at $*$ in Figure 7.17. The circle direction depends on the elevon that is faulted, e.g. left banked turns are preferred when the right elevon is faulted and vice-versa. (Section 7.7.4 explains the rationale.) At 841 s, the UAS operator injects a $-5°$ stuck fault in the right elevon ($\times$). The flight computer sends a constant PWM command to the right elevon to simulate the stuck fault. Since the demonstration excludes fault diagnosis, the fault-tolerant controller is activated immediately upon the fault injection. The ground track following the fault injection ($\times$) is denoted using the solid line. For the next 8.2 min, the Vireo continues to track the circle. The cross-track error standard deviations before and after the fault injection are 3 m and 7 m, respectively.

At 1332 s, the UAS operator initiates the autoland sequence ($+$) causing the Vireo to peel away from the circle hold and fly towards the approach circle. The approach circle is computed based on the target landing spot and direction set by the UAS operator (Section 7.9.1). Since the prevailing winds on that day are 4.5 $\mathrm{m\,s^{-1}}$ from the Southwest, the operator commands a Westerly landing to take advantage of the headwind. The Vireo enters the approach circle ($\triangle$) and traverses about three-quarters of the circle. Once the Vireo is heading due West, it exits the circle and transitions to glideslope tracking ($\square$). The remainder of the autoland is best explained by the next figure.

Figure 7.18 shows the terminal flight path of the Vireo as seen by an observer standing on the ground and facing due North. The approach speed is set to 13.9 $\mathrm{m\,s^{-1}}$. The Vireo tracks a 6° glideslope that intersects the virtual runway, which is located 100 ft above the ground, at an East position of zero. The standard operating procedure calls for the pilot to take manual control of the aircraft ($\circ$) once it reaches the virtual runway and attempt a go-around so as to provide for a gentler landing (Section 7.9.1). However, on this particular flight, the phugoid mode is excited throughout the glideslope tracking phase, as is evident in Figure 7.18. A trough of the phugoid mode causes the aircraft to intersect the virtual runway 18 m short of the intended touchdown point. Since the aircraft is lowered below the virtual runway, a go-around is not attempted and the pilot simply lands the aircraft (◇) at 37 m due West. This final phase of manual control is denoted by the dashed line between ○ and ◇. Next, it is instructive to evaluate the *short-term* and the *long-term* aircraft responses after the fault injection.

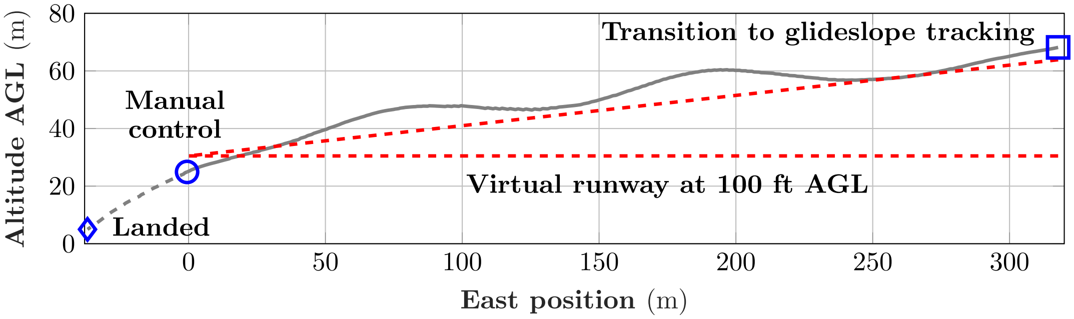
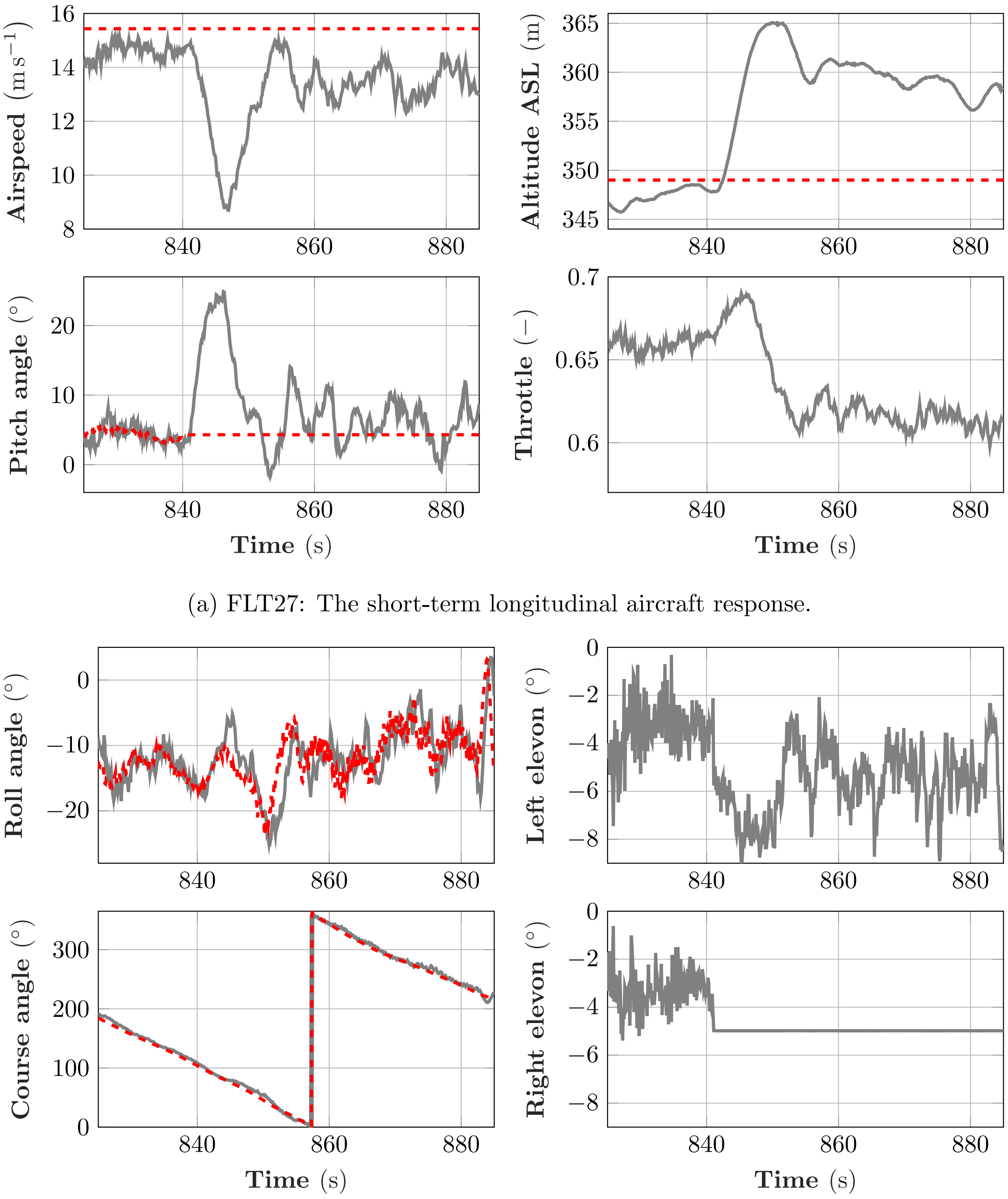


Figure 7.18: FLT27: The Vireo tracks a glideslope during the final approach. The pilot takes manual control of the aircraft once it reaches the virtual runway at 100 ft AGL.

Figure 7.19 shows the *short-term* longitudinal and lateral-directional aircraft responses as well as the control commands after the fault injection at 841 s. The longitudinal motion variables (Figure 7.19a) are close to their respective reference commands and the mean value of the right elevon (Figure 7.19b) is approximately −3.3° before the fault. The fault causes the aircraft to pitch nose-up to about 25° at 846 s, which in turn decreases the airspeed to about 8.7 $\mathrm{m\,s^{-1}}$ and increases the altitude by about 15 m. The fault-tolerant total energy controller responds by opening the throttle to 0.69. Around 12 s after the fault, the airspeed and pitch angle return closer to their respective reference commands, but exhibit higher variability than before the fault. Since the pitch angle is an uncontrolled variable (Section 7.4), its reference command is simply set equal to the trim pitch angle. As for the lateral-directional motion variables (Figure 7.19b), the fault causes the Vireo to momentarily roll rightward, as seen in the spike in the roll angle at 844 s. The single surface controller corrects this by temporarily deflecting the left elevon to about −8°. Finally, since the failed position of the right elevon (−5°) is slightly larger in magnitude than its mean value before the fault (−3.3°), the trim airspeed reduces. This is best explained by the next set of plots.

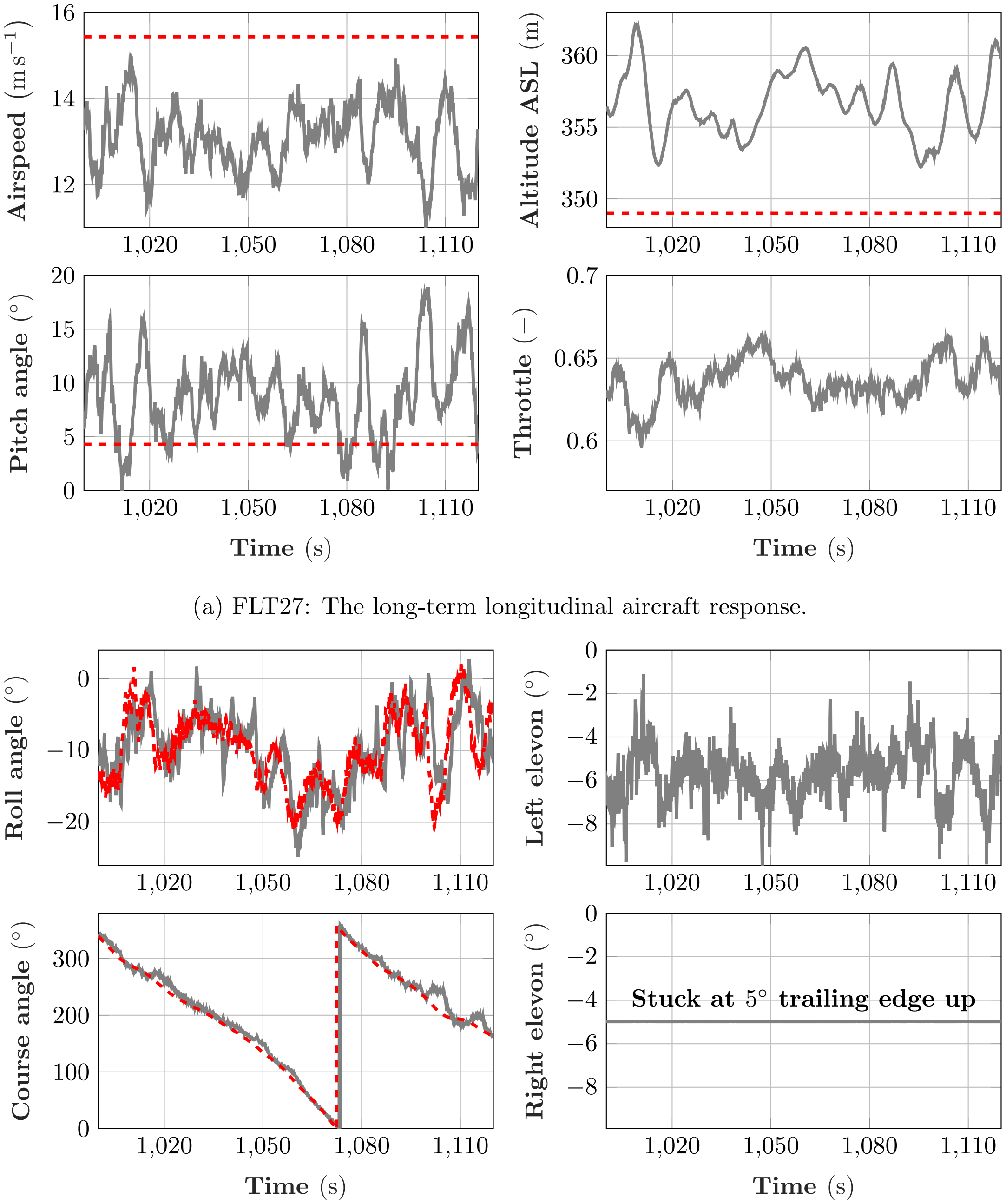
Figure 7.20 follows the same layout as Figure 7.19, except that it shows the long-term



(a) FLT27: The short-term longitudinal aircraft response.

(b) FLT27: The short-term lateral-directional aircraft response and the elevon commands.

Figure 7.19: FLT27: The short-term aircraft response after the fault injection at 841 s.



(a) FLT27: The long-term longitudinal aircraft response.

(b) FLT27: The long-term lateral-directional aircraft response and the elevon commands.

Figure 7.20: FLT27: The long-term aircraft response after the fault injection at 841 s.

aircraft responses over two minutes after the fault. The new trim airspeed (Figure 7.20a) is around 13 $\mathrm{m\,s^{-1}}$, which is a constraint imposed by the failed right elevon. Recall that $\bar{K}_{TECS}$ does not directly track the balance energy command using the pitch attitude (Section 7.6). Since the airspeed is below its reference value, the total energy controller compensates by increasing the altitude by about 7.6 m above its reference value. The steady-state errors in the airspeed and the altitude offset each other because $\bar{K}_{TECS}$ tracks the mixed energy, as shown in Figure 7.21. The phugoid mode is visible in the airspeed and the altitude, with a time period of about 10 s (Figure 7.20a). In particular, the airspeed varies within ± 2 $\mathrm{m\,s^{-1}}$ of 13 $\mathrm{m\,s^{-1}}$, the altitude varies within ± 5 m of 356.6 m, and the pitch attitude varies within $\pm 6°$ of 9.45°. The RMS tracking error in the mixed energy is 18.8 J, which is less than 1% of the reference value. Since the aircraft is rolled leftward, the left elevon is deflected slightly more (mean value of $-5.85°$ in Figure 7.20b) than the right elevon. In addition, the RMS tracking errors in the roll and the course angles (Figure 7.20b) are 4° and 9.3°, respectively.

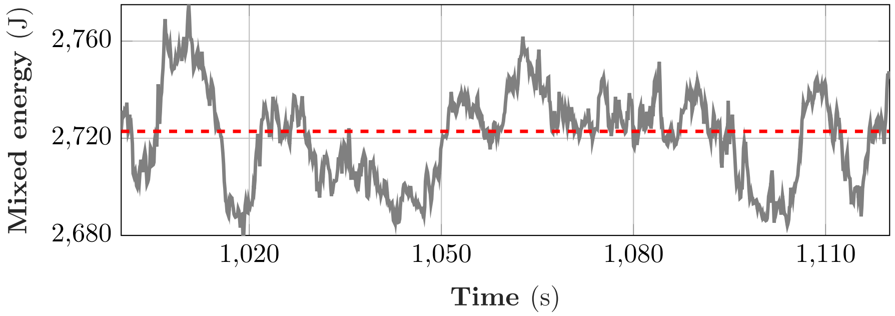


Figure 7.21: FLT27: The total-energy controller tracks the mixed energy after the fault.

7.9.4 FLT35: Autoland Using Design B

An autoland demonstration using Design B of the single surface controller and the fault-tolerant total energy controller is conducted during FLT35 on 2017-10-05. The discussion follows the same sequence as in Section 7.9.3. Shortly after the manual takeoff, the pilot takes the Vireo up to 250 ft AGL and engages the autopilot at 921 s. Figure 7.22 shows the ground track of the Vireo from the autopilot engagement ($*$) until the landing on the virtual runway ($\circ$). The ground track is plotted in a local North-East reference frame, whose origin coincides with the target landing spot chosen by the UAS operator prior to the start of the flight. The various events occurring in the flight are annotated on the ground track and are explained one-by-one.

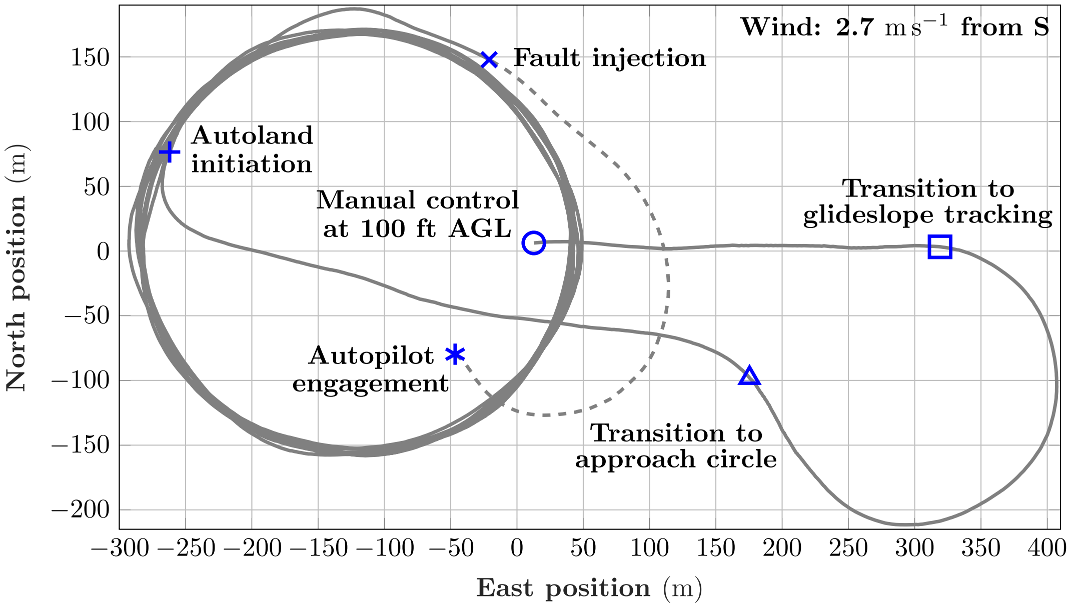


Figure 7.22: FLT35: The ground track of the Vireo indicating different flight events. The solid line indicates the times when the single surface controller is active.

The autopilot engagement ($*$) activates the nominal controller, which takes the Vireo into a left banked circle hold at an altitude of 250 ft AGL, an airspeed of 15.4 m s^{-1}, and a turn radius of 150 m. The center of the circle $(-120, 6.5)$ m is automatically set by the flight computer to the point where the airplane climbs past the altitude of 150 ft AGL. The Vireo traces a counterclockwise circle when viewed in the North-East plane. This initial phase, wherein the nominal controller is active, is denoted by the dashed line starting at $*$ in Figure 7.22. As with FLT27, a counterclockwise circle is chosen because the right elevon is faulted. At 945 s, the UAS operator injects a $-5°$ stuck fault in the right elevon ($\times$). The flight computer sends a constant PWM command to the right elevon to simulate the stuck fault. Since the demonstration excludes fault diagnosis, the fault-tolerant controller is activated immediately upon the fault injection. The ground track following the fault injection ($\times$) is denoted using the solid line. For the next 8.2 min, the Vireo continues to track the circle. The cross-track error standard deviation after the fault injection is 2.9 m, which is much better than FLT27. This is explained by the better controller used, and calmer winds observed, during FLT35.

At 1438 s, the UAS operator initiates the autoland sequence (+) causing the Vireo to peel away from the circle hold and fly towards the approach circle. The prevailing winds on that day are 2.7 $\mathrm{m\,s^{-1}}$ from the South. The operator commands a Westerly landing to be consistent with FLT27. The Vireo enters the approach circle (△) and traverses about three-quarters of the circle. Once the Vireo is heading due West, it exits the circle and transitions to glideslope tracking (□).

Figure 7.23 shows the terminal flight path of the Vireo as seen by an observer standing on the ground and facing due North. The target approach speed is 15.4 $\mathrm{m\,s^{-1}}$. The Vireo tracks a 6° glideslope that intersects the virtual runway at an East position of zero. The phugoid mode is excited during the glideslope tracking phase, but is less pronounced when compared to FLT27. However, there is a 5 m steady-state error in the altitude, which is a consequence of selecting a non-zero balance energy weight (see Section 7.6.2). The pilot follows the standard operating procedure and takes manual control of the aircraft (∘) once it is within 12 m of the virtual touchdown point and performs a go-around, which terminates with a gentle landing. The go-around is not shown in Figure 7.23 to avoid clutter. Once again, it is instructive to evaluate the short-term and the long-term aircraft responses.

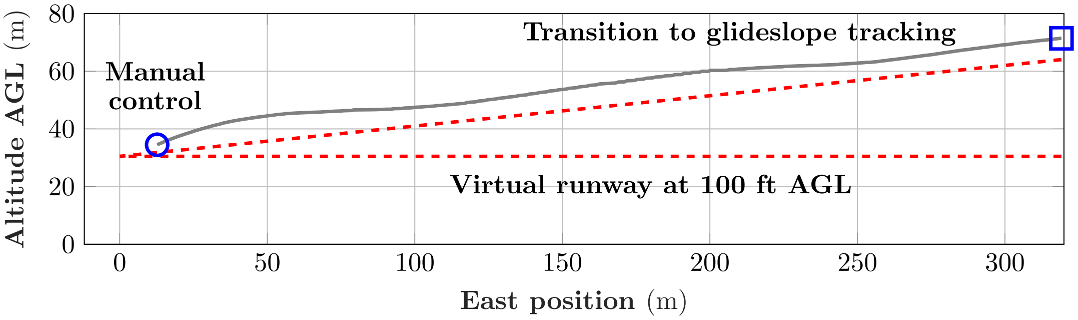
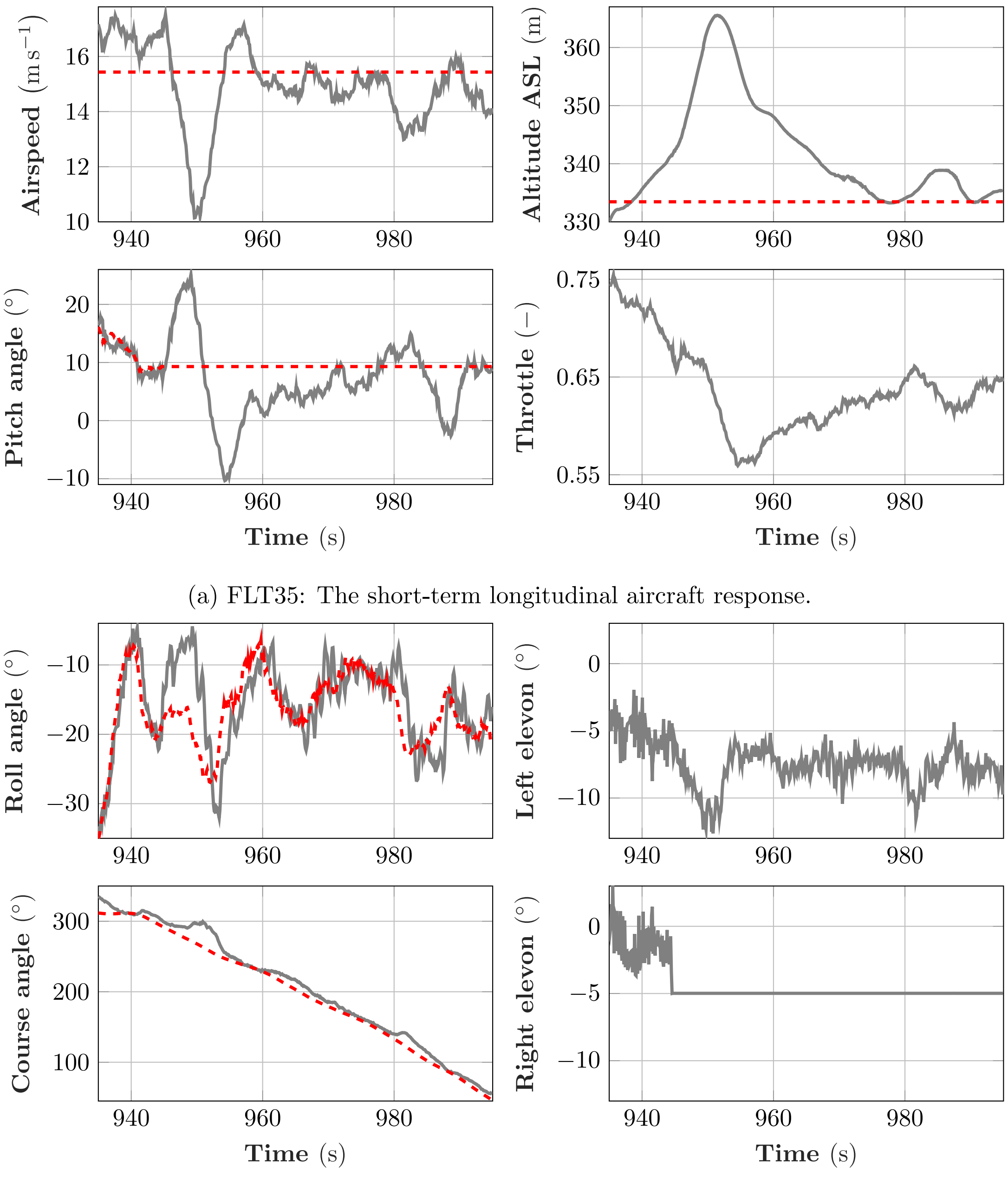


Figure 7.23: FLT35: The Vireo tracks a glideslope during the final approach. The pilot takes manual control of the aircraft once it reaches the virtual runway at 100 ft AGL.

Figure 7.24 shows the short-term longitudinal and lateral-directional aircraft responses as well as the control commands after the fault injection at 945 s. The longitudinal motion variables (Figure 7.24a) are close to their respective reference commands and the mean value of the right elevon (Figure 7.24b) is approximately $-2°$ before the fault. Since the right elevon gets stuck at $-5°$, the fault causes the aircraft to pitch nose-up to about $25°$ at 949 s. This in turn decreases the airspeed to about 10.2 $\mathrm{m\,s^{-1}}$ and increases the altitude by about 32 m. Since the altitude tracking error is much larger than the airspeed tracking error, the fault-tolerant total energy controller decreases the throttle from 0.73 to 0.56. At around 980 s, the airspeed and pitch angle return closer to their respective reference commands. As with FLT27, the pitch angle command is set equal to the trim pitch angle after the fault. As for the lateral-directional motion (Figure 7.24b), the fault causes the Vireo to momentarily roll rightward, as seen between 945 s and 950 s. The single surface controller corrects this by temporarily deflecting the left elevon to about $-12°$. This trailing edge up deflection is also partly responsible for the behavior seen in the longitudinal variables shortly after the fault injection.

There is another large deviation in all the motion variables near 980 s. This is explained by returning to Figure 7.22 and noting that the aircraft deviates from the circle hold near $(-290, 0)$ m. The guidance law corrects this deviation by commanding a sharp leftward roll at 980 s, which the single surface controller achieves by commanding the left elevon to about $-10°$. This in turn causes the airspeed to drop and the altitude to rise slightly near 980 s (Figure 7.24a). Finally, since the failed position of the right elevon ($-5°$) is slightly larger in magnitude than its mean value before the fault ($-2°$), the trim airspeed reduces. This is best explained by the next set of plots.

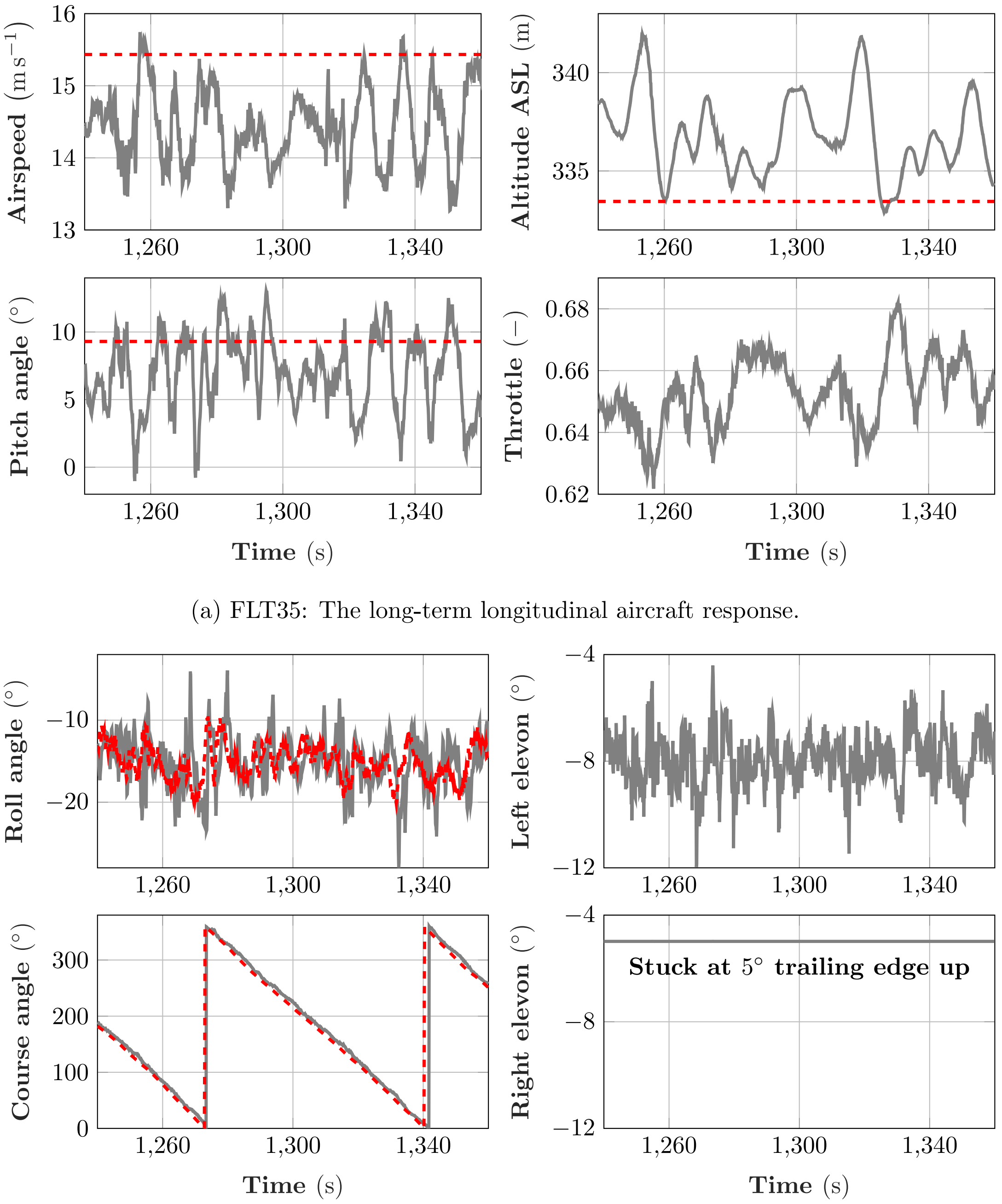
Figure 7.25 follows the same layout as Figure 7.24, except that it shows the long-term aircraft responses over two minutes after the fault. The new trim airspeed (Figure 7.25a) is around 14.4 $\mathrm{m\,s^{-1}}$, which is a constraint imposed by the failed right elevon. Note that this trim airspeed is 1.4 $\mathrm{m\,s^{-1}}$ faster than the corresponding value observed in FLT27 although both flights used the same fault magnitude. This is because the Vireo carries an on-board camera during FLT35 and is thus 45 g heavier than its nominal all-up weight. Since the airspeed is below its reference value, the total energy controller compensates by increasing the altitude by about 3.3 m above its reference value. The steady-state errors in the airspeed and the altitude offset each other because $\bar{K}_{TECS}$



(a) FLT35: The short-term longitudinal aircraft response.

(b) FLT35: The short-term lateral-directional aircraft response and the elevon commands.

Figure 7.24: FLT35: The short-term aircraft response after the fault injection at 945 s.



(a) FLT35: The long-term longitudinal aircraft response.

(b) FLT35: The long-term lateral-directional aircraft response and the elevon commands.

Figure 7.25: FLT35: The long-term aircraft response after the fault injection at 945 s.

tracks the mixed energy, as shown in Figure 7.26. The phugoid mode is visible in the airspeed and the altitude, with a time period of about 10 s (Figure 7.25a). In particular, the airspeed varies within ±1 $\mathrm{m\,s^{-1}}$ of 14.4 $\mathrm{m\,s^{-1}}$, the altitude varies within ±3.3 m of 336.8 m, and the pitch attitude varies within ±5° of 7.44°. The RMS tracking error in the mixed energy is 16 J, which is less than 1% of the reference value. Since the aircraft is rolled leftward, the left elevon is deflected slightly more (mean value of −8° in Figure 7.25b) than the right elevon. In addition, the RMS tracking errors in the roll and the course angles (Figure 7.25b) are 3.3° and 8°, respectively.

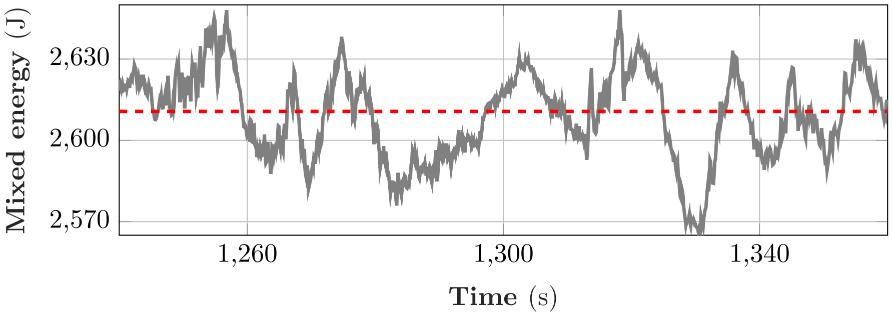


Figure 7.26: FLT35: The total-energy controller tracks the mixed energy after the fault.

7.9.5 Comparison of the Flight Tests

Apart from FLT27 and FLT35, there are four other flights that validate the fault-tolerant controller (see Table 7.4 and Appendix D). Figure 7.27 shows the three-dimensional views of the flight trajectories of FLT27, FLT28, FLT34, and FLT35, from the fault injection (×) until the landing on the virtual runway (○). Each trajectory is plotted in a local East-North-Up reference frame, whose origin coincides with the target landing spot chosen by the UAS operator prior to the start of the flight. The dashed lines indicate the glideslope and the virtual runway located at 100 ft AGL. As seen in the plots, FLT34 and FLT35 contain fewer deviations from the circle hold, when compared to FLT27 and FLT28. This is explained by the better controller (Design B) used, and the calmer winds observed, during FLT34 and FLT35. This also leads to better glideslope tracking performance in FLT34 and FLT35, when compared to FLT27 and FLT28. The remainder of this section makes these comparisons more concrete.

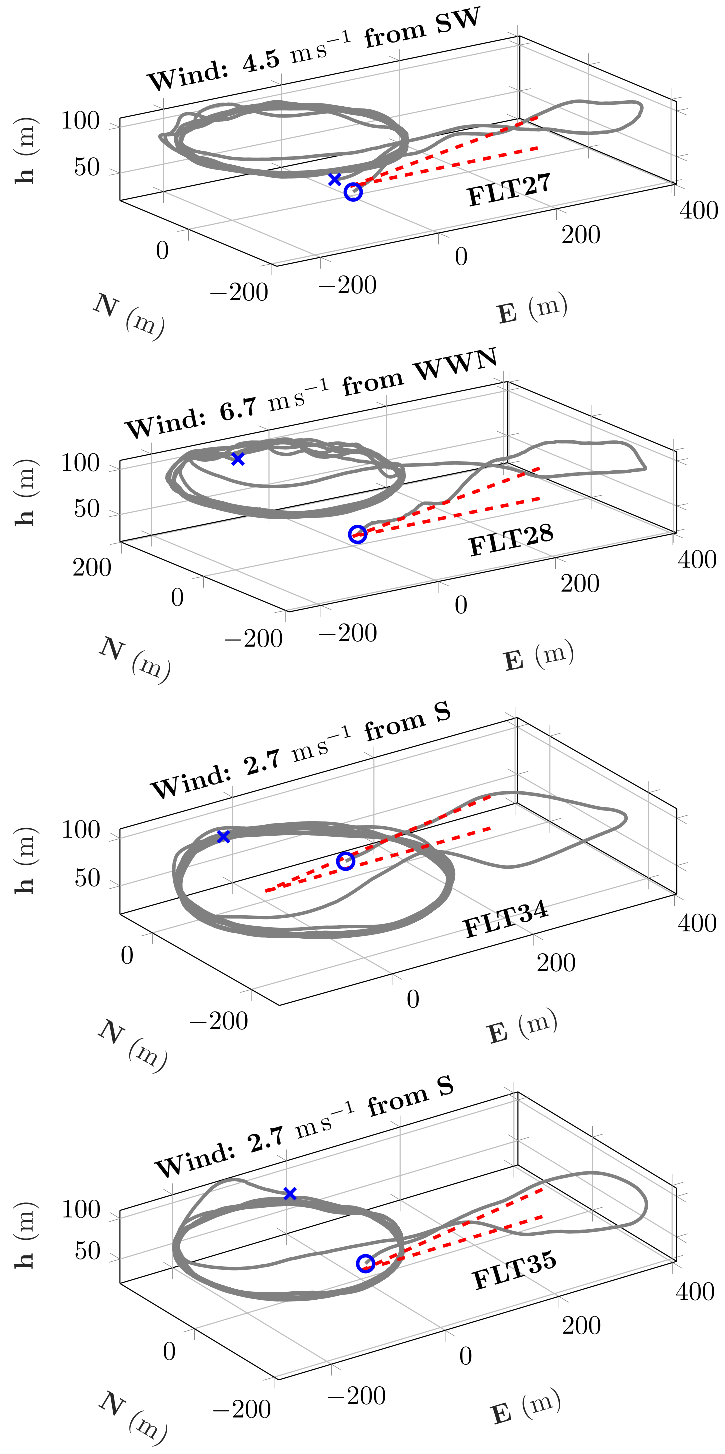


Figure 7.27: Three-dimensional views of the flight trajectories during the autoland.

Circle hold tracking performance

First, Figure 7.28 compares the six flights (Table 7.4) using the summary statistics of the closed-loop tracking errors observed during the circle hold. For each flight, the summary statistics are displayed using box plots for six motion variables, which are labeled along the vertical axes. The box plots show the median (black circle), the 25th percentile (lower edge of box), and the 75th percentile (upper edge of box) of the tracking errors before (green) and after (red) the fault injection. For both the pre- and post-fault cases, the statistics are computed over a time segment after the closed-loop has converged to a steady-state. Thus the median is representative of the trim condition and the interquartile range is representative of the closed-loop tracking performance about that trim condition. Figure 7.28 shows that the nominal airspeed command of 15.4 $\mathrm{m\,s^{-1}}$ is achieved within a steady-state error of ± 0.5 $\mathrm{m\,s^{-1}}$ in all the flights with the exception of FLT35. Although it may appear as such, FLT35 is not an anomaly because the fault is injected before the Vireo reaches its target altitude. Thus, the trim condition shown for FLT35 before the fault actually corresponds to a steady climb.

All the flights share the same fault mode of the right elevon getting stuck at 5° trailing edge up. However, the trim position of the right elevon before the fault (not shown in Figure 7.28) is slightly different across the flights due to small changes in the mass and the CG location that arise from variations in the locations of the battery and the on-board camera[4]. In particular, all the flights except FLT34 experience a net trailing edge up deflection of the right elevon upon the fault injection, which causes the trim (median) airspeed to decrease and the trim (median) pitch angle to increase. On the other hand, FLT34 experiences a net trailing edge down deflection of the right elevon, which causes the trim airspeed to increase and the trim pitch angle to decrease. The relative difference between the nominal and the faulty elevon positions also affects the relative change in the trim airspeed. For example, FLT29 and FLT30 experience small relative differences in their right elevon positions before and after the fault and thus small changes in their respective trim airspeeds. Further, the airspeed and the altitude exhibit non-zero steady-state errors after the fault injection, as explained previously. This is particularly visible in FLT27, FLT28, FLT34, and FLT35 in Figure 7.28.

[4] The Vireo carries a 45 g on-board camera in FLT29, FLT30, FLT34, and FLT35.

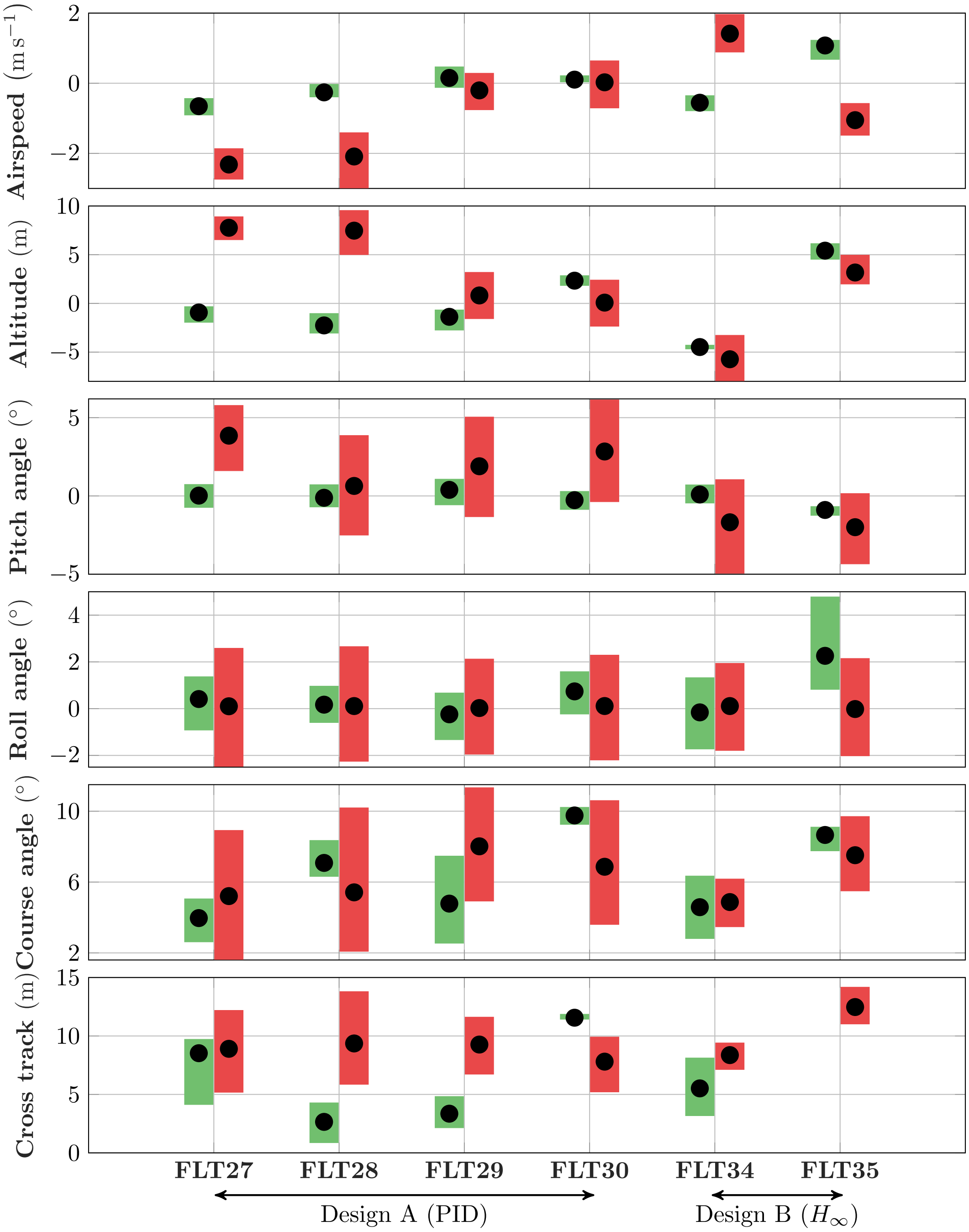


Figure 7.28: The summary statistics of the closed-loop tracking errors observed during the circle hold, before (green) and after (red) the fault injection.

For a given motion variable in Figure 7.28, the interquartile ranges before the fault (green) are largely similar across all flights. In addition, the interquartile ranges increase, as expected, after the fault injection (red). However, the increases in the interquartile ranges after the fault appear to be functions of: (1) the prevailing winds, (2) the controller used, and (3) the motion variables under consideration. In particular, the interquartile ranges in the longitudinal motion variables (airspeed, altitude, and pitch angle) appear to be strong functions of the prevailing winds but weak functions of the controller used. For example, FLT28, FLT29, and FLT30 experience high winds $\left(6.7 \mathrm{~m\,s^{-1}}\right)$ and thus exhibit larger interquartile ranges in all the longitudinal variables when compared to FLT27, which experiences moderate winds $\left(4.5 \mathrm{~m\,s^{-1}}\right)$. On the other hand, comparing FLT27 with FLT34 and FLT35, both of which also experience moderate winds $\left(2.7 \mathrm{~m\,s^{-1}}\right)$, reveals that Design B has very little impact in reducing the longitudinal tracking errors. This is because neither Design A nor Design B directly controls any of the longitudinal motion variables (see Section 7.7).

The benefits of Design B are more apparent in mitigating the lateral-directional tracking errors. In particular, FLT34 and FLT35 (Design B) exhibit smaller interquartile ranges in the tracking errors associated with the roll angle, the course angle, and the cross track when compared to the flights that use Design A. This is because Design B has lower output sensitivity (see Figure 7.15) than Design A. While the prevailing winds also affect the lateral-directional tracking errors, they do not seem to be the dominating factor. For example, the interquartile ranges of the course angle error and the cross track error in FLT27 are much higher than those in FLT34 and FLT35 despite the winds being only marginally higher in FLT27. Further, Design B in some cases yields tracking errors that are on par with, or lower than, those observed before the fault is injected. For example, in FLT34 the interquartile ranges in the course angle error and the cross track error are 3.6° and 5 m before the fault and 2.7° and 2.3 m after the fault. Figure 7.28 does not show the cross track error statistics before the fault for FLT35 because the fault is injected before the Vireo enters the circle hold.

Glideslope tracking performance

Next, Figure 7.29 compares the flights (Table 7.4) using the summary statistics of the closed-loop tracking errors observed during the glideslope phase of the autoland, after

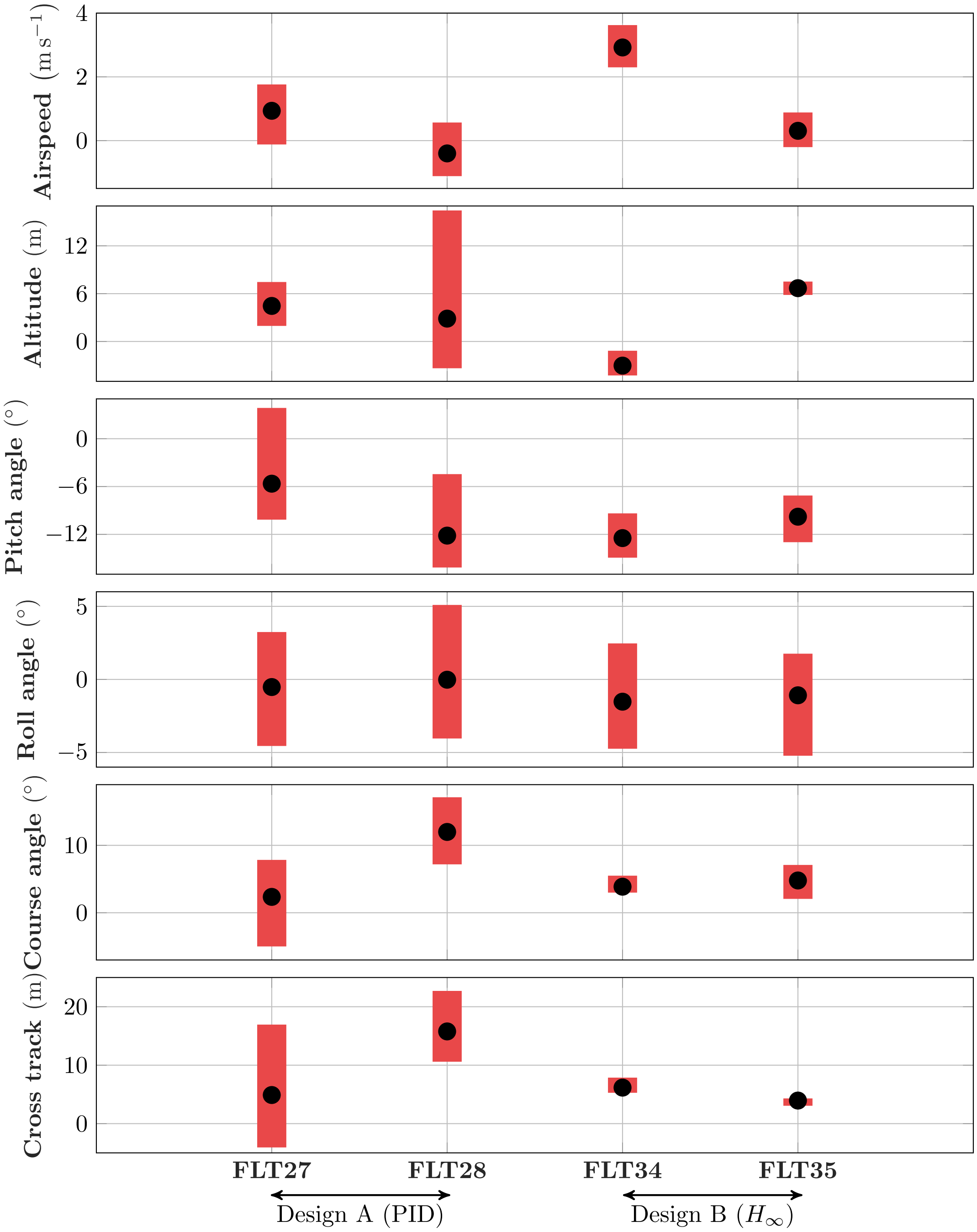


Figure 7.29: The summary statistics of the closed-loop tracking errors observed during the glideslope phase of the autoland, after the fault injection.

the fault injection. The figure excludes FLT29 and FLT30 since an autoland is not attempted in these two flights. As before, the summary statistics are displayed using box plots. As seen in the figure, FLT34 and FLT35 (Design B) exhibit smaller interquartile ranges in the tracking errors of all the motion variables when compared to FLT27 and FLT28 (Design A). However, the degree of reduction in the interquartile ranges depends on the variable. For the longitudinal variables, the reductions are observed mainly in the altitude and the pitch angle. The airspeed interquartile ranges appear similar across the four flights. Further, FLT28 exhibits a large interquartile range in the altitude tracking error due to the high winds. For the lateral-directional variables, the largest reductions in the interquartile ranges are observed in the course angle error and the cross track error. For example, the interquartile ranges of the course angle error and the cross track error are 12° and 21 m in FLT27, and 5° and 1.3 m in FLT35. Further, FLT28 exhibits the highest roll angle error due to the high winds. The interquartile ranges of the roll angle error in FLT27 (7.8°), FLT34 (7.2°), and FLT35 (7°) are largely similar.

Concluding remarks

In conclusion, the flight tests corroborate the comparisons drawn between Designs A and B in Section 7.7.3. In particular, Design B yields better tracking performance in both the circle hold and the glideslope phases. This is a direct consequence of incorporating weights that penalize the input/output disturbances and excessive control effort in the H_∞ synthesis (Section 7.7.2). The H_∞ synthesis results in a controller whose gain rolls off above 2.7 $\text{rad}\,\text{s}^{-1}$ in the roll angle tracking channel and above 42 $\text{rad}\,\text{s}^{-1}$ in the roll rate damping channel (Figure 7.12). This ensures that Design B actuates the single operable elevon only in the frequency range required for lateral control. Further, since none of the longitudinal motion variables are in feedback with the single surface controller, the better tracking performance of Design B is mainly observed in the lateral-directional motion variables. Consequently, the prevailing winds tend to exacerbate the longitudinal tracking errors more than the lateral-directional tracking errors.

The better tracking performance of Design B comes at the expense of a more complicated design process (Section 7.7.2). It also has some implementation considerations (Appendix B). Although Design A may not have all the performance and robustness advantages of Design B, it is arguably simpler to design, validate, and implement.

Moreover, although Design A may have larger tracking errors, it still works during the autoland, as is evident in FLT27 and FLT28 in Figure 7.27. The decision to follow Design A or B, or maybe even a completely new approach, ultimately depends on the requirements of the mission and the dynamics of the aircraft.

Finally, all the flights share the same fault mode of the right elevon getting stuck at 5° trailing edge up. Since this is close to the nominal trim point, the faults cause the trim airspeed to change by no more than $\pm 3\ \mathrm{m\,s^{-1}}$. Thus the LTI fault-tolerant controllers (Designs A and B) are sufficient for recovery. Faults that are farther away from the trim point, however, will result in larger changes in the airspeed. A LPV controller that is scheduled by the airspeed (much like the LPV fault detection filter of Chapter 6) could potentially address these large airspeed changes.

Chapter 8

Conclusions and Future Research

8.1 Conclusions

This thesis focuses on the problem of automatically detecting and managing stuck control surface faults on a small unmanned aircraft system (UAS) that is equipped with only two control surfaces (called *elevons*). A fault in either of the elevons of this UAS must, therefore, be managed using the only other elevon and the throttle. The thesis demonstrates the feasibility of this reconfiguration approach by autonomously landing this UAS in the presence of a stuck elevon.

The thesis models the flight dynamics of the aircraft using physics-based first-principles. In particular, the vortex lattice method (VLM) is used to estimate the aerodynamic stability and control derivatives. Given the limitations of VLM, flight experiments are conducted for system identification. The experiments reveal that the longitudinal dynamics predicted by VLM are sufficiently accurate in the frequency range desired for control design. On the other hand, the lateral-directional dynamics predicted by VLM differ from the experimental observations. Thus the lateral-directional stability and control derivatives are updated using the experimental data.

The fault detection and isolation algorithm is designed using a new convex method for synthesizing robust output estimators for continuous-time, uncertain, gridded, linear parameter-varying (LPV) systems. The uncertainty is described using integral quadratic constraints (IQC). The flight test results illustrate the benefits of modeling the aircraft dynamics across multiple flight conditions (LPV framework) and accounting for the model uncertainty (IQC framework) in designing the FDI algorithms.

The stuck elevon fault renders the aircraft under-actuated. Given that the final goal is to land the aircraft, the ability to change and maintain course is crucial. Consequently, the fault-tolerant controller is designed to sacrifice longitudinal tracking performance in favor of lateral-directional tracking performance. In order to control the throttle, the classical total energy controller is modified by creating a new measure called the mixed energy, which increases the closed-loop damping ratio of the phugoid mode.

Finally, although this thesis focuses on one particular small UAS, the concept of flight control using a single aerodynamic control surface is more generally applicable [203]. Previous work has demonstrated the automatic control of a larger UAS [204] and the manual control of another UAS [205], both using a single control surface.

8.2 Future Research

The investigations pursued in this thesis leave open several topics for future research.

8.2.1 Trajectory Generation

The flight demonstrations of the landing of the handicapped aircraft involve trajectory generation and control. Chapter 7 exclusively focuses on control. The trajectory generation, as currently implemented in `AuraUAS`, does not account for the dynamics of the aircraft. Consequently, the trajectory to the landing spot is formed by stitching together many individual trajectories, each of which corresponds to one of three flight conditions: Steady wings-level flight, steady turn, or steady descent. The focus on steady flight conditions limits the scope of this thesis in two main ways. First, this thesis is limited to stuck elevon faults for which a steady flight condition exists (see Section 6.3). Second, there does not appear to be a way to rapidly decrease the sink rate of the aircraft prior to touchdown, using a single elevon, by performing a steady descent (see Section 7.9.1).

More sophisticated trajectory generation methods could potentially address both of these limitations. In particular, the existence of steady flight conditions is sufficient, but not necessary, for recovery. Consequently, it may be possible to safely land the aircraft after a fault without having to fly it around an equilibrium point or flight condition. If such dynamic trajectories exist and can be found, then one could close the loop around them using the fault-tolerant controller. The challenge, however, lies in finding *feasible* trajectories, i.e. those that satisfy the constraints of the aircraft dynamics. The path planning literature has made significant advances in this area, as explained next.

Some nonlinear dynamical systems exhibit a property called differential flatness [206–208], which may be exploited for real-time trajectory generation [209]. One prominent aerospace example is the differentially flat dynamics of the quadrotor, which Mellinger and Kumar [210] exploit for trajectory generation and control. The dynamics of fixed-wing aircraft, on the other hand, are differentially flat only under somewhat restrictive assumptions [211–213]. Nevertheless, Bry et al. [214] experimentally demonstrate the efficacy of using a simplified, but differentially flat, representation of fixed-wing aircraft dynamics for trajectory generation and control.

One may also investigate sampling-based algorithms, such as probabilistic roadmaps (PRM) [215], rapidly exploring random trees (RRT) [199, 202, 216], its asymptotically optimal variant RRT* [217], and other randomized algorithms [218].

Other potential approaches include using a library of motion primitives to construct feasible trajectories (called trajectory libraries) [219–221], explicitly considering the stabilizability of the trajectory during the planning phase (called feedback motion planning) [222], and nonlinear model-predictive control [223].

8.2.2 Reliability Assessment

This thesis is motivated by the low reliability of small UAS. It is natural to then ask: To what extent does the solution proposed in this thesis increase the reliability of the particular small UAS considered? Answering this question is important for the certification of UAS or for comparing the benefits afforded by different fault-tolerant controllers. Although some preliminary work has been done in the area of reliability assessment [153], there is substantial room for development. For example, one could investigate how the design of the flight control law affects the trade-off between performance and reliability. In addition, one could account for the likelihood of the control surface being stuck at a particular position while setting design requirements for the fault-tolerant flight controller.

8.2.3 Other Areas

Uncertainty modeling

Chapter 6 assumed that the nominal aircraft model is affected by multiplicative uncertainty at its input. One could relax this assumption and instead quantify the model uncertainty in terms of the uncertainties in the underlying aircraft parameters, e.g. the moments of inertia and the aerodynamic coefficients. One could use Chapter 4 in [224] as a template for this uncertainty modeling exercise.

Fault diagnosis

Chapter 6 on fault diagnosis offers three broad areas for further investigation. First, there may be better alternatives to using the envelope departure time as a basis for setting the controller switching time requirement. For example, one could investigate approaches that, in addition to the departure time, consider the distance of the aircraft trajectory from the envelope boundaries. In addition, one could account for the need to avoid stall in setting the controller switching time requirement. Second, one could compare model-based and data-driven approaches to fault detection (see, for example, [148]). Data-driven approaches could allow for a much broader set of fault modes to be included without a substantial increase in effort. Third, one could analyze the performance of the fault diagnosis algorithm using receiver operating characteristic curves. Such analysis would better inform the selection of the residual filter and the threshold. Fourth, one could investigate the relation between persistency of excitation and the fault detection time.

Fault-tolerant control

Chapter 7 on fault-tolerance modifies the total energy controller by introducing a constant weight w_b that trades-off the altitude and the airspeed tracking performances. The problem of the high sink rate during touchdown could potentially be addressed by varying w_b as a function of the altitude. Specifically, one could decrease the airspeed command and simultaneously vary w_b to emphasize airspeed tracking as the aircraft begins the glideslope tracking phase. In addition, one could investigate other fault-tolerant control architectures and other control design methods such as adaptive control.

Scope of the demonstrations

The scope of the demonstrations may be expanded to further validate the algorithms developed in this thesis. One could start by demonstrating the fault diagnosis and the fault-tolerant control sequentially in the same flight test. The demonstrations could also include other (larger) fault magnitudes and other control surface fault modes, e.g. floating faults. Further, one could investigate the possibility of recovering the aircraft if both control surfaces were to fail or if the electric motor were to fail. The last scenario (loss of thrust) may have interesting ties with trajectory planning. For example, by finding trajectories that minimize the energy dissipation, one could maximize the range of the aircraft and thereby increase the probability of landing it safely.

Aircraft design

This thesis proposed a strategy of controlling a *given* fixed-wing UAS using one aerodynamic control surface. This is essentially an analysis problem. One could also investigate the corresponding *synthesis* problem: How must fixed-wing UAS be designed so as to maximize their capability of tolerating on-board faults, including control surface faults? While this seems like a very broad design challenge, one could place design constraints in order to narrow the design space. A group of undergraduate seniors at the University of Minnesota successfully completed a design-build-fly challenge to manually fly a small UAS using one aerodynamic control surface [205]. One could extend their design work to cover additional fault modes.

References

[1] European Aviation Safety Agency. Concept of operations for drones: A risk based approach to regulation of unmanned aircraft, 2015.

[2] Federal Aviation Administration. Final rule for 14 CFR Part 107, 2016.

[3] J. Amos, E. Bergquist, J. Cole, J. Phillips, S. Reimann, and S. Shuster. UAV for reliability. Technical report, University of Minnesota, December 2013.

[4] P. Freeman. *Reliability Assessment for Low-cost Unmanned Aerial Vehicles*. PhD thesis, University of Minnesota, 2014.

[5] J. E. Wilborn and J. V. Foster. Defining commercial transport loss-of-control: A quantitative approach. In *AIAA Atmospheric Flight Mechanics Conference*, 2004.

[6] UAV Laboratories, University of Minnesota. `www.uav.aem.umn.edu`.

[7] UAV Lab Photos and Videos. `www.flickr.com/photos/100936386@N02`.

[8] A. Dorobantu. *Test Platforms for Model-Based Flight Research*. PhD thesis, University of Minnesota, 2013.

[9] Sentera. `www.sentera.com`.

[10] A. Mahon, R. Venkataraman, and B. Taylor. Development of a research flight test vehicle based on a small two surface UAV. Technical report, University of Minnesota, 2016.

[11] C. Olson. AuraUAS. `https://github.com/AuraUAS`.

[12] D. Gebre-Egziabher. *Design and Performance Analysis of a Low-Cost Aided Dead Reckoning Navigator*. PhD thesis, Stanford University, 2001.

[13] F. A. P. Lie and D. Gebre-Egziabher. Synthetic air data system. *Journal of Aircraft*, 50:1234–1249, 2013.

[14] S. Bhattacharyya and D. Gebre-Egziabher. Vector loop RAIM in nominal and GNSS-stressed environments. *IEEE Transactions on Aerospace and Electronic Systems*, 50(2):1249–1268, April 2014.

[15] S. Bhattacharyya and D. Gebre-Egziabher. Integrity monitoring with vector GNSS receivers. *IEEE Transactions on Aerospace and Electronic Systems*, 50(4):2779–2793, October 2014.

[16] S. Bhattacharyya and D. Gebre-Egziabher. Kalman filter-based RAIM for GNSS receivers. *IEEE Transactions on Aerospace and Electronic Systems*, 51:2444–2459, 2015.

[17] T. Layh and D. Gebre-Egziabher. Design for graceful degradation and recovery from GNSS interruptions. *IEEE Aerospace and Electronic Systems Magazine*, 32:4–17, 2017.

[18] G. H. Bryan. *Stability in Aviation.* Macmillan and Co., 1911.

[19] F. W. Lanchester. *Aerodonetics.* Constable and Co. Ltd., 1908.

[20] G. J. Hancock and J. S. Y. Lam. Part 1 - Introduction to the concept of axiomatic aerodynamic modelling. *The Aeronautical Journal*, 91(901):1–3, 1987.

[21] G. J. Hancock and J. S. Y. Lam. Part 2 - Longitudinal aircraft motions with attached flow. *The Aeronautical Journal*, 91(901):4–20, 1987.

[22] G. J. Hancock and J. S. Y. Lam. Part 3 - Coupled longitudinal-lateral aircraft motions with attached flow. *The Aeronautical Journal*, 91(901):20–28, 1987.

[23] B. W. McCormick. *Aerodynamics, Aeronautics, and Flight Mechanics.* Wiley, 1995.

[24] B. Etkin and L. D. Reid. *Dynamics of Flight: Stability and Control.* Wiley, 1996.

[25] M. V. Cook. *Flight Dynamics Principles.* Elsevier, second edition, 2007.

[26] R. C. Nelson. *Flight Stability and Automatic Control.* McGraw-Hill, 1998.

[27] B. L. Stevens and F. L. Lewis. *Aircraft Control and Simulation.* Wiley, 2003.

[28] R. F. Stengel. *Flight Dynamics.* Princeton University Press, 2004.

[29] R. D. Kimberlin. *Flight Testing of Fixed-Wing Aircraft.* AIAA, 2003.

[30] V. M. Falkner. The calculations of aerodynamic loading on surfaces of any shape. Technical report, British Aeronautical Research Council, 1943.

[31] V. M. Falkner. The accuracy of calculations based on vortex lattice theory. Technical report, British Aeronautical Research Council, 1946.

[32] R. J. Margason and J. E. Lamar. Vortex-lattice FORTRAN program for estimating subsonic aerodynamic characteristics of complex planforms. Technical report, NASA Langley Research Center, 1972.

[33] J. D. Anderson. *Fundamentals of Aerodynamics.* McGraw-Hill, 2010.

[34] J. Katz and A. Plotkin. *Low-Speed Aerodynamics: From Wing Theory to Panel Methods.* McGraw-Hill, 1991.

[35] J. Bertin and R. M. Cummings. *Aerodynamics for Engineers.* Pearson, 2013.

[36] J. D. Anderson. *Introduction to Flight.* McGraw-Hill, eighth edition, 2015.

[37] R. M. Cummings, W. H. Mason, S. A. Morton, and D. R. McDaniel. *Applied Computational Aerodynamics: A Modern Engineering Approach.* Cambridge University Press, 2015.

[38] A. Marcos and G. Balas. Development of linear-parameter-varying models for aircraft. *Journal of Guidance, Control, and Dynamics*, 27:218–228, 2004.

[39] J. Shamma. *Analysis and design of gain scheduled control systems.* PhD thesis, Massachusetts Institute of Technology, 1988.

[40] J. Shamma and M. Athans. Gain scheduling: potential hazards and possible remedies. *IEEE Control Systems*, 12:101–107, 1992.

[41] F. Wu. *Control of Linear Parameter Varying Systems*. PhD thesis, University of California Berkeley, 1995.

[42] G. Balas, A. Hjartarson, A. Packard, and P. Seiler. LPVTools: A Toolbox for Modeling, Analysis, and Synthesis of Parameter Varying Control Systems. Technical report, Musyn, Inc., 2015.

[43] A. Hjartarson, A. Packard, and P. Seiler. LPVTools: A Toolbox for Modeling, Analysis, and Synthesis of Parameter Varying Control Systems. In *First IFAC Workshop on Linear Parameter Varying Systems*, 2015.

[44] Lennart Ljung. *System Identification: Theory for the User*. Prentice Hall PTR, 1999.

[45] L. Ljung. System Identification Toolbox: Users Guide. Technical report, The MathWorks Inc., 2017.

[46] T. Soderstrom and P. Stoica. *System Identification*. Prentice-Hall, 2001.

[47] M. Tischler and R. Remple. *Aircraft and Rotorcraft System Identification*. American Institute of Aeronautics and Astronautics, 2006.

[48] V. Klein and E. A. Morelli. *Aircraft System Identification: Theory And Practice*. American Institute of Aeronautics and Astronautics, 2006.

[49] R. V. Jategaonkar. *Flight Vehicle System Identification: A Time-Domain Methodology*. American Institute of Aeronautics and Astronautics, 2006.

[50] B. Mettler, M. B. Tischler, and T. Kanade. System identification modeling of a small-scale unmanned rotorcraft for flight control design. *Journal of the American Helicopter Society*, 47:50–63, 2002.

[51] C. R. Theodore, M. B. Tischler, and J. D. Colbourne. Rapid frequency-domain modeling methods for unmanned aerial vehicle flight control applications. *Journal of Aircraft*, 41:735–743, 2004.

[52] A. Dorobantu, A. Murch, B. Mettler, and G. Balas. System identification for small, low-cost, fixed-wing unmanned aircraft. *Journal of Aircraft*, 50(4):1117–1130, 2013.

[53] H. Pfifer and B. P. Danowsky. System identification of a small flexible aircraft. In *AIAA Atmospheric Flight Mechanics Conference*, 2016.

[54] G. Gremillion and J. Humbert. System identification of a quadrotor micro air vehicle. In *AIAA Atmospheric Flight Mechanics Conference*, 2010.

[55] S. Skogestad and I. Postlethwaite. *Multivariable Feedback Control: Analysis and Design*. Wiley, 2005.

[56] A. Dorobantu, W. Johnson, F. A. Lie, B. Taylor, A. Murch, Y. C. Paw, D. Gebre-Egziabher, and G. Balas. An airborne experimental test platform: From theory to flight. In *American Control Conference*, pages 659–673, 2013.

[57] S. Park, J. Deyst, and J. How. A new nonlinear guidance logic for trajectory tracking. In *AIAA Guidance, Navigation, and Control Conference*, 2004.

[58] S. Park, J. Deyst, and J. How. Performance and lyapunov stability of a nonlinear path-following guidance method. *Journal of Guidance, Control, and Dynamics*, 30:1718–1728, 2007.

[59] A. A. Lambregts. Functional integration of vertical flight path and speed control using energy principles. In *First Annual Nasa Aircraft Controls Workshop*, 1983.

[60] A. A. Lambregts. Vertical flight path and speed control autopilot design using total energy principles. In *AIAA Guidance, Navigation, and Control Conference*, 1983.

[61] A. A. Lambregts. Operational aspects of the integrated vertical flight path and speed control system. In *SAE Aerospace Congress and Exposition*, 1983.

[62] A. A. Lambregts. Integrated system design for flight and propulsion control using total energy principles. In *AIAA Aircraft Design, Systems, and Technology Meeting*, 1983.

[63] K. R. Bruce, J. R. Kelly, and L. H. Person. NASA B737 flight test results of the total energy control system. In *AIAA Astrodynamics Conference*, 1986.

[64] K. R. Bruce. NASA B737 flight test results of the total energy control system. Technical report, NASA CR-178285, 1987.

[65] A. Warren. Application of total energy control for high-performance aircraft vertical transitions. *Journal of Guidance, Control, and Dynamics*, 14:447–452, 1991.

[66] M. A. Bruzzini. Development of a TECS control-law for the lateral directional axis of the McDonnell Douglas F-15 Eagle. Master's thesis, University of Washington, 1994.

[67] M. E. Argyle. *Modeling and Control of a Tailsitter with a Ducted Fan.* PhD thesis, Brigham Young University, 2016.

[68] C. Voth and U. L. Ly. Total energy control system autopilot design with constrained parameter optimization. In *American Control Conference*, 1990.

[69] L. F. Faleiro and A. A. Lambregts. Analysis and tuning of a total energy control system control law using eigenstructure assignment. *Aerospace Science and Technology*, 3:127–140, 1999.

[70] Viswanathan, A. Kale, G. K. Singh, and V. V. Patel. L_1-adaptive output feedback based energy control. In *European Control Conference*, 2015.

[71] M. E. Argyle and R. W. Beard. Nonlinear total energy control for the longitudinal dynamics of an aircraft. In *American Control Conference*, 2016.

[72] P. Seiler. Class notes for automatic control systems. `http://www.aem.umn.edu/~SeilerControl/Courses/AEM4321/ClassicalControl_Master.pdf`, 2016.

[73] R. Venkataraman and P. Seiler. Convex LPV synthesis of estimators and feedforwards using duality and integral quadratic constraints. *International Journal of Robust and Nonlinear Control*, 28(3):953–975, 2018.

[74] B. Appleby, J. Dowdle, and W. VanderVelde. Robust estimator design using μ-synthesis. In *IEEE Conference on Decision and Control*, volume 1, pages 640–645, 1991.

[75] I. Petersen and D. McFarlane. Robust state estimation for uncertain systems. In *IEEE Conference on Decision and Control*, pages 2630–2631, 1991.

[76] L. Xie and C. de Souza. On robust filtering for linear systems with parameter uncertainty. In *IEEE Conference on Decision and Control*, pages 2087–2092, 1995.

[77] K. Sun and A. Packard. Robust H_2 and H_∞ filters for uncertain LFT systems. *IEEE Transactions on Automatic Control*, 50(5):715–720, 2005.

[78] K. Sun and A. Packard. Optimal, worst case filter design via convex optimization. In *Control of Uncertain Systems: Modelling, Approximation, and Design*, pages 293–315. Springer, 2006.

[79] H. Li and M. Fu. A linear matrix inequality approach to robust H_∞ filtering. *IEEE Transactions on Signal Processing*, 45(9):2338–2350, 1997.

[80] M. Ravuri and H. Asada. Robust state estimation for uncertain nonlinear systems described using integral quadratic constraints: An LMI formulation. In *American Control Conference*, volume 2, pages 1293–1297, 2000.

[81] G. Scorletti and V. Fromion. Further results on the design of robust H_∞ feedforward controllers and filters. In *IEEE Conference on Decision and Control*, pages 3560–3565, 2006.

[82] C. Scherer and I. Kose. Robustness with dynamic IQCs: An exact state-space characterization of nominal stability with applications to robust estimation. *Automatica*, 44(7):1666–1675, 2008.

[83] P. Seiler, B. Vanek, J. Bokor, and G. Balas. Robust H_∞ filter design using frequency gridding. In *American Control Conference*, pages 1801–1806, 2011.

[84] R. Mangoubi. *Robust Estimation and Failure Detection for Linear Systems.* PhD thesis, Massachusetts Institute of Technology, 1995.

[85] J. Veenman and C. Scherer. A synthesis framework for robust gain-scheduling controllers. *Automatica*, 50(11):2799–2812, 2014.

[86] R. Venkataraman and P. Seiler. Robust LPV estimator synthesis using integral quadratic constraints. In *2016 American Control Conference*, pages 4611–4616, 2016.

[87] J. Geromel. Optimal linear filtering under parameter uncertainty. *IEEE Transactions on Signal Processing*, 47(1):168–175, 1999.

[88] J. Geromel, J. Bernussou, G. Garcia, and M. de Oliveira. H_2 and H_∞ robust filtering for discrete-time linear systems. *SIAM Journal on Control and Optimization*, 38(5):1353–1368, 2000.

[89] J. Geromel and M. de Oliveira. H_2 and H_∞ robust filtering for convex bounded uncertain systems. *IEEE Transactions on Automatic Control*, 46(1):100–107, 2001.

[90] K. Zhou, J. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice-Hall, New Jersey, 1996.

[91] A. Helmersson. *IQC Synthesis based on Inertia Constraints*. Linköping University Electronic Press, 1998.

[92] J. Veenman and C. Scherer. IQC-synthesis with general dynamic multipliers. *International Journal of Robust and Nonlinear Control*, 24(17):3027–3056, 2014.

[93] S. Wang, H. Pfifer, and P. Seiler. Robust synthesis for linear parameter varying systems using integral quadratic constraints. *Automatica*, 68:111–118, 2016.

[94] A. Giusto and F. Paganini. Robust synthesis of feedforward compensators. *IEEE Transactions on Automatic Control*, 44(8):1578–1582, 1999.

[95] G. Ferreres and C. Roos. Robust feedforward design in the presence of LTI/LTV uncertainties. *International Journal of Robust and Nonlinear Control*, 17(14):1278–1293, 2007.

[96] I. Kose and C. Scherer. Robust L_2-gain feedforward control of uncertain systems using dynamic IQCs. *International Journal of Robust and Nonlinear Control*, 19(11):1224–1247, 2009.

[97] E. Prempain and I. Postlethwaite. A feedforward control synthesis approach for LPV systems. In *2008 American Control Conference*, pages 3589–3594, 2008.

[98] Y. Altun and K. Gulez. Linear parameter varying feedforward control synthesis using parameter-dependent lyapunov function. *Nonlinear Dynamics*, 78:2293–2307, 2014.

[99] A. Megretski and A. Rantzer. System analysis via integral quadratic constraints. *IEEE Transactions on Automatic Control*, 42:819–830, 1997.

[100] F. Wu, X. Yang, A. Packard, and G. Becker. Induced L_2 norm control for LPV systems with bounded parameter variation rates. *International Journal of Robust and Nonlinear Control*, 6:983–998, 1996.

[101] H. Pfifer and P. Seiler. Robustness analysis of linear parameter varying systems using integral quadratic constraints. *International Journal of Robust and Nonlinear Control*, 25(15):2843–2864, 2015.

[102] H. Pfifer and P. Seiler. Less conservative robustness analysis of linear parameter varying systems using integral quadratic constraints. *International Journal of Robust and Nonlinear Control*, 26(16):3580–3594, 2016.

[103] P. Seiler. Stability analysis with dissipation inequalities and integral quadratic constraints. *IEEE Transactions on Automatic Control*, 60(6):1704–1709, 2015.

[104] C. Scherer. *Robust controller synthesis is convex for systems without control channel uncertainties.* Springer, 2009.

[105] P. Apkarian and P. Gahinet. A convex characterization of gain-scheduled H_∞ controllers. *IEEE Transactions on Automatic Control*, 40:853–864, 1995.

[106] A. Packard. Gain scheduling via linear fractional transformations. *Systems and Control Letters*, 22:79–92, 1994.

[107] C. Scherer. *Advances in linear matrix inequality methods in control*, chapter Robust mixed control and linear parameter-varying control with full-block scalings, pages 187–207. SIAM, 2000.

[108] J. Veenman and C. Scherer. Stability analysis with integral quadratic constraints: A dissipativity based proof. In *IEEE Conference on Decision and Control*, pages 3770–3775, 2013.

[109] A. Megretski. Integral quadratic constraints. Technical report, Massachusetts Institute of Technology, 2009.

[110] B. Hu, M. Lacerda, and P. Seiler. Robustness analysis of uncertain discrete-time systems with dissipation inequalities and integral quadratic constraints. *International Journal of Robust and Nonlinear Control*, 27:1940–1962, 2017.

[111] R. Horn and C. Johnson. *Matrix Analysis.* Cambridge University Press, 1990.

[112] H. Bart, I. Gohberg, and M. Kaashoek. *Minimal Factorization of Matrix and Operator Functions.* Birkhäuser, 1979.

[113] G. Meinsma. J-spectral factorization and equalizing vectors. *Systems and Control Letters*, 25:243–249, 1995.

[114] T. Azizov and I. Iokhvidov. Linear operators in spaces with indefinite metric and their applications. *Journal of Mathematical Sciences*, 15:438–490, 1981.

[115] J. Carrasco and P. Seiler. A note on IQC factorizations. Submitted to Automatica, 2017.

[116] P. Gahinet and P. Apkarian. A linear matrix inequality approach to H_∞ control. *International Journal of Robust and Nonlinear Control*, 4:421–448, 1994.

[117] G. Dullerud and F. Paganini. *A Course in Robust Control Theory: A Convex Approach.* Springer, New York, 1999.

[118] R. Isermann and P. Ballé. Trends in the application of model-based fault detection and diagnosis of technical processes. *Control Engineering Practice*, 5(5):709–719, 1997.

[119] S. X. Ding. *Model-Based Fault Diagnosis Techniques: Design Schemes, Algorithms, and Tools.* Springer-Verlag, 2013.

[120] S. X. Ding. *Data-driven Design of Fault Diagnosis and Fault-tolerant Control Systems*. Springer-Verlag, 2014.

[121] J. J. Gertler. *Fault Detection and Diagnosis in Engineering Systems*. Routledge, first edition, 1998.

[122] J. Chen and R. J. Patton. *Robust Model-Based Fault Diagnosis for Dynamic Systems*. Springer, Boston, MA, 1999.

[123] R. Isermann. *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*. Springer, Germany, 2006.

[124] Gertler. Fault detection and isolation using parity relations. *Control Engineering Practice*, 5:653–661, 1997.

[125] R. Isermann. Process fault detection based on modeling and estimation methods - a survey. *Automatica*, 20:387–404, 1984.

[126] J. J. Gertler. *Fault Detection and Diagnosis*. John Wiley & Sons, 2014.

[127] R. Isermann. Model-based fault-detection and diagnosis status and applications. *Annual Reviews in Control*, 29(1):71–85, 2005.

[128] I. Hwang, S. Kim, Y. Kim, and C. E. Seah. A survey of fault detection, isolation, and reconfiguration methods. *IEEE Transactions on Control Systems Technology*, 18(3):636–653, 2010.

[129] A. Edelmayer, J. Bokor, and L. Keviczky. An H_∞ filtering approach to robust detection of failures in dynamical systems. In *IEEE Conference on Decision and Control*, volume 3, pages 3037–3039, 1994.

[130] J. Stoustrup and H. H. Niemann. Fault estimation - a standard problem approach. *International Journal of Robust and Nonlinear Control*, 12(8):649–673, 2002.

[131] R. K. Mehra and J. Peschon. An innovations approach to fault detection and diagnosis in dynamic systems. *Automatica*, 7:637–640, 1971.

[132] R. J. Patton and J. Chen. A review of parity space approaches to fault diagnosis for aerospace systems. *Journal of Guidance, Control, and Dynamics*, 17(2):278–285, 1994.

[133] R.P.G. Collinson. *Introduction to Avionics Systems.* Springer, 2nd edition, 2003.

[134] P. Goupil. AIRBUS state of the art and practices on FDI and FTC in flight control system. *Control Engineering Practice*, 19(6):524 – 539, June 2011.

[135] P. Goupil. Oscillatory failure case detection in the A380 electrical flight control system by analytical redundancy. *Control Engineering Practice*, 18(9):1110–1119, 2010.

[136] ADDSAFE. Advanced fault diagnosis for sustainable flight guidance and control, European 7th Framework Program. `http://addsafe.deimos-space.com`, 2012.

[137] P. Goupil and A. Marcos. The European ADDSAFE project: Industrial and academic efforts towards advanced fault diagnosis. *Control Engineering Practice*, 31:109–125, 2014.

[138] M. R. Napolitano, C. I. Chen, and S. Naylor. Aircraft failure detection and identification using neural networks. *Journal of Guidance, Control, and Dynamics*, 16(6):999–1009, 1993.

[139] A. Marcos, S. Ganguli, and G. J. Balas. An application of H_∞ fault detection and isolation to a transport aircraft. *Control Engineering Practice*, 13:105–119, 2005.

[140] P. Freeman, P. Seiler, and G. Balas. Robust fault detection for commercial transport air data probes. In *IFAC Proceedings Volumes*, volume 44, pages 13723–13728, 2011.

[141] P. Freeman, P. Seiler, and G. Balas. Air data system fault modeling and detection. *Control Engineering Practice*, 21(10):1290–1301, 2013.

[142] H. Alwi and C. Edwards. Fault detection and fault-tolerant control of a civil aircraft using a sliding-mode-based scheme. *IEEE Transactions on Control Systems Technology*, 16:499–510, 2008.

[143] S. Glavaski and M. Elgersma. Active aircraft fault detection and isolation. In *IEEE Systems Readiness Technology Conference*, 2001.

[144] F. Bateman, H. Noura, and M. Ouladsine. Actuators fault diagnosis and tolerant control for an unmanned aerial vehicle. In *IEEE International Conference on Control Applications*, pages 1061–1066, 2007.

[145] 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 *IEEE Conference on Decision and Control*, volume 4, pages 4208–4213, 1998.

[146] R. Pandita. *Dynamic flight envelope assessment with flight safety applications.* PhD thesis, University of Minnesota, 2010.

[147] P. Freeman and G. Balas. Analytical fault detection for a small UAV. In *AIAA Infotech@Aerospace Conference*, Boston, MA, USA, 2013.

[148] P. Freeman, R. Pandita, N. Srivastava, and G. Balas. Model-based and data-driven fault detection performance for a small UAV. *IEEE Transactions on Mechatronics*, 18(4):1300–1309, 2013.

[149] P. Bauer, R. Venkataraman, B. Vanek, P. Seiler, and J. Bokor. Fault detection and basic in-flight reconfiguration of a small UAV equipped with elevons. In *10th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes, SAFEPROCESS*, 2018.

[150] J. Stoustrup, M. J. Grimble, and H. Niemann. Design of integrated systems for the control and detection of actuator/sensor faults. *Sensor Review*, 1997.

[151] R. Pandita, J. Bokor, and G. Balas. Performance metrics for fault detection and isolation filters. In *American Control Conference*, pages 1390–1395, 2011.

[152] R. Pandita, J. Bokor, and G. Balas. Closed-loop performance metrics for fault detection and isolation filter and controller interaction. *International Journal of Robust and Nonlinear Control*, 23:419–438, 2013.

[153] R. Venkataraman, P. Seiler, M. Lukátsi, and B. Vanek. Reliability assessment of actuator architectures for unmanned aircraft. *Journal of Aircraft*, 54(3):955–966, 2017.

[154] J. Doyle, K. Glover, P. Khargonekar, and B. Francis. State-space solutions to standard H_2 and H_∞ control problems. *IEEE Transactions on Automatic Control*, 34(8):831–847, Aug 1989.

[155] G. Balas, R. Chiang, A. Packard, and M. Safonov. Robust Control Toolbox: Users Guide. Technical report, The MathWorks Inc., 2015.

[156] M. M. Mahmoud, J. Jiang, and Y. Zhang. *Active Fault Tolerant Control Systems: Stochastic Analysis and Synthesis*. Springer-Verlag, 2003.

[157] G. Tao, S. Chen, X. Tang, and S. M. Joshi. *Adaptive Control of Systems with Actuator Failures*. Springer-Verlag, 2004.

[158] H. Noura, D. Theilliol, J. Ponsart, and A. Chamseddine. *Fault-tolerant Control Systems*. Springer, 2009.

[159] M. S. Mahmoud and Y. Xia. *Analysis and Synthesis of Fault-Tolerant Control Systems*. Wiley, 2014.

[160] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki. *Diagnosis and Fault-Tolerant Control*. Springer-Verlag, 2016.

[161] M. Blanke, W. C. Frei, F. Kraus, R. Patton, and M. Staroswiecki. What is fault-tolerant control? *IFAC Proceedings Volumes*, 33:41–52, 2000.

[162] G. Tao, S. M. Joshi, and X. Ma. Adaptive state feedback and tracking control of systems with actuator failures. *IEEE Transactions on Automatic Control*, 46:78–95, 2001.

[163] G. Tao and S. M. Joshi. *Adaptive Control of Systems with Actuator Failures*, pages 111–156. Springer, 2001.

[164] M. Steinberg. Historical overview of research in reconfigurable flight control. *Journal of Aerospace Engineering*, 219:263–275, 2005.

[165] Y. Zhang and J. Jiang. Bibliographical review on reconfigurable fault-tolerant control systems. *Annual Reviews in Control*, 32:229–252, 2008.

[166] C. Hajiyev and F. Caliskan. *Fault Diagnosis and Reconfiguration in Flight Control Systems*. Springer-Verlag, 2003.

[167] C. Edwards, T. Lombaerts, and H. Smaili. *Fault Tolerant Flight Control: A Benchmark Challenge*. Springer-Verlag, 2010.

[168] A. Zolghadri, D. Henry, J. Cieslak, D. Efimov, and P. Goupil. *Fault Diagnosis and Fault-Tolerant Control and Guidance for Aerospace Vehicles*. Springer-Verlag, 2014.

[169] X. Yu, Z. Liu, and Y. Zhang. Fault-tolerant flight control design with finite-time adaptation under actuator stuck failures. *IEEE Transactions on Control Systems Technology*, 25:1431–1440, 2017.

[170] M. Pachter and Y. Huang. Fault tolerant flight control. *Journal of Guidance, Control, and Dynamics*, 26:151–160, 2003.

[171] N. Nguyen, K. Krishnakumar, and J. Kaneshige. Flight dynamics and hybrid adaptive control of damaged aircraft. *Journal of Guidance, Control, and Dynamics*, 31:751–764, 2008.

[172] D. Jourdan, M. Piedmonte, V. Gavrilets, D. Vos, and J. McCormick. Enhancing UAV survivability through damage tolerant control. In *AIAA Guidance, Navigation, and Control Conference*, 2010. Paper No. AIAA 2010-7548.

[173] G. Chowdhary, E. N. Johnson, R. Chandramohan, M. S. Kimbrell, and A. Calise. Guidance and control of airplanes under actuator failures and severe structural damage. *Journal of Guidance, Control, and Dynamics*, 36:1093–1104, 2013.

[174] D. Shore and M. Bodson. Flight testing of a reconfigurable control system on an unmanned aircraft. *Journal of Guidance, Control, and Dynamics*, 28:698–707, 2005.

[175] J. M. Maciejowski and C. N. Jones. MPC fault-tolerant flight control case study: Flight 1862. *IFAC Proceedings Volumes*, 36:119–124, 2003.

[176] J. Monaco, D. Ward, R. Barron, and R. Bird. Implementation and flight test assessment of an adaptive, reconfigurable flight control system. In *AIAA Guidance, Navigation, and Control Conference*, 1997. Paper No. AIAA 1997-3738.

[177] A. Page, J. Monaco, and D. Meloney. Flight testing of a retrofit reconfigurable control law architecture using an F/A-18C. In *AIAA Guidance, Navigation, and Control Conference*, 2006. Paper No. AIAA 2006-6052.

[178] D. G. Ward, J. F. Monaco, and M. Bodson. Development and flight testing of a parameter identification algorithm for reconfigurable control. *Journal of Guidance, Control, and Dynamics*, 21:948–956, 1998.

[179] J. Cieslak, D. Henry, A. Zolghadri, and P. Goupil. Development of an active fault-tolerant flight control strategy. *Journal of Guidance, Control, and Dynamics*, 31:135–147, 2008.

[180] Y. Hitachi. Damage-tolerant control system design for propulsion-controlled aircraft. Master's thesis, University of Toronto, 2009.

[181] S. Bennani, R. van der Sluis, G. Schram, and J. Mulder. Control law reconfiguration using robust linear parameter varying control. In *AIAA Guidance, Navigation, and Control Conference*, 1999. Paper No. AIAA 1999-4137.

[182] S. Ganguli, A. Marcos, and G. Balas. Reconfigurable LPV control design for Boeing 747-100/200 longitudinal axis. In *American Control Conference*, pages 3612–3617, 2002.

[183] L. Zhong and F. Mora-Camino. On-line optimization for fault tolerant flight control. *Procedia Engineering*, 80:638–655, 2014.

[184] G. J. J. Ducard. *Fault-Tolerant Flight Control and Guidance Systems for a Small Unmanned Aerial Vehicle*. PhD thesis, ETH Zurich, 2007.

[185] T. J. J. Lombaerts. *Fault Tolerant Flight Control: A Physical Model Approach.* PhD thesis, TU Delft, 2010.

[186] J. S. Brinker and K. A. Wise. Flight testing of reconfigurable control law on the X-36 tailless aircraft. *Journal of Guidance, Control, and Dynamics*, 24:903–909, 2001.

[187] P. R. Chandler, M. Pachter, and M. Mears. System identification for adaptive and reconfigurable control. *Journal of Guidance, Control, and Dynamics*, 18:516–524, 1995.

[188] J. E. Bobrow and W. Murray. An algorithm for RLS identification of parameters that vary quickly with time. *IEEE Transactions on Automatic Control*, 38:351–354, 1993.

[189] D. G. Ward, R. L. Barron, M. P. Carley, and T. J. Curtis. Real-time parameter identification for self-designing flight control. In *NAECON*, pages 526–531, 1994.

[190] M. Bodson. An adaptive algorithm with information-dependent data forgetting. In *American Control Conference*, pages 3485–3489, 1995.

[191] J. D. Boskovic, R. Prasanth, and R. K. Mehra. Retrofit fault-tolerant flight control design under control effector damage. *Journal of Guidance, Control, and Dynamics*, 30:703–712, 2007.

[192] D. A. Handelman and R. F. Stengel. A theory for fault-tolerant flight control combining expert system and analytical redundancy concepts. In *Astrodynamics Conference, Fluid Dynamics and Co-located Conferences*, 1986. Paper No. AIAA 1986-2092.

[193] D. A. Handelman and R. F. Stengel. Combining expert system and analytical redundancy concepts for fault-tolerant flight control. *Journal of Guidance, Control, and Dynamics*, 12:39–45, 1989.

[194] J. F. Stewart and T. L. Shuck. Flight-testing of the self-repairing flight control system using the F-15 highly integrated digital electronic control flight research facility. Technical report, Nasa Dryden Flight Research Center, 1990.

[195] J. Burken, T. Maine, F. Burcham, Jr., and J. Kahler. Longitudinal emergency control system using thrust modulation demonstrated on an MD-11 airplane. In

32nd AIAA/ASME/SAE/ASEE Joint Propulsion Conference, 1996. Paper No. AIAA 1996-3062.

[196] J. Schierman, D. Ward, J. Hull, N. Gandhi, M. Oppenheimer, and D. Doman. Integrated adaptive guidance and control for re-entry vehicles with flight test results. *Journal of Guidance, Control, and Dynamics*, 27:975–988, 2004.

[197] T. Tucker. Touchdown: The development of propulsion controlled aircraft at nasa dryden. Nasa Monograph, Jan 1999.

[198] L. D. Crider. Control of commercial aircraft with vertical tail loss. In *AIAA 4th Aviation Technology, Integration and Operations (ATIO) Forum*, 2004. Paper No. AIAA 2004-6293.

[199] S. M. Lavalle. Rapidly-exploring random trees: A new tool for path planning. Technical report, Iowa State University, 1998.

[200] A. Richards and J. P. How. Aircraft trajectory planning with collision avoidance using mixed integer linear programming. In *American Control Conference*, volume 3, pages 1936–1941, 2002.

[201] T. McGee, S. Spry, and J. K. Hedrick. Optimal path planning in a constant wind with a bounded turning rate. In *AIAA Guidance, Navigation, and Control Conference*, 2005.

[202] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.

[203] R. Venkataraman, P. Seiler, and B. Taylor. Fault-tolerant aircraft flight control using a subset of aerodynamic control surfaces. United States Patent Application US 2017/0349267 A1, May 2017.

[204] R. Venkataraman and P. J. Seiler. Safe flight using one aerodynamic control surface. In *AIAA Guidance, Navigation, and Control Conference*, 2016.

[205] A. Krause, R. Condron, E. Day, D. Dessens, and P. Wagner. Single control surface UAV. Technical report, University of Minnesota, 2016.

[206] M. Fliess, J. Lévine, P. Martin, and P. Rouchon. Flatness and defect of non-linear systems: introductory theory and examples. *International Journal of Control*, 61:1327–1361, 1995.

[207] R. M. Murray, M. Rathinam, and W. Sluis. Differential flatness of mechanical control systems: A catalog of prototype systems. In *ASME International Congress and Exposition*, 1995.

[208] H. Sira-Ramírez and S. K. Agrawal. *Differentially Flat Systems.* CRC Press, 2004.

[209] M. J. Van Nieuwstadt and R. M. Murray. Real-time trajectory generation for differentially flat systems. *International Journal of Robust and Nonlinear Control*, 8:995–1020, 1998.

[210] D. Mellinger and V. Kumar. Minimum snap trajectory generation and control for quadrotors. In *IEEE International Conference on Robotics and Automation*, pages 2520–2525, 2011.

[211] P. Martin. Aircraft control using flatness. In *IMACS/IEEE-SMC Multiconference CESA Symposium on Control, Optimization and Supervision*, pages 194–199, 1996.

[212] P. Martin, S. Devasia, and B. Paden. A different look at output tracking: control of a VTOL aircraft. *Automatica*, 32(1):101–107, 1996.

[213] J. Hauser and R. Hindman. Aggressive flight maneuvers. In *IEEE Conference on Decision and Control*, volume 5, pages 4186–4191, 1997.

[214] A. Bry, C. Richter, A. Bachrach, and N. Roy. Aggressive flight of fixed-wing and quadrotor aircraft in dense indoor environments. *International Journal of Robotics Research*, 37:969–1002, 2015.

[215] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, Aug 1996.

[216] S. M. LaValle and J. J. Kuffner. Rapidly-exploring random trees: Progress and prospects, 2000.

[217] S. Karaman and E. Frazzoli. Sampling-based algorithms for optimal motion planning. *International Journal of Robotics Research*, 30(7):846–894, 2011.

[218] E. Frazzoli, M. Dahleh, and E. Feron. Real-time motion planning for agile autonomous vehicles. *Journal of Guidance, Control, and Dynamics*, 25:116–129, 2002.

[219] E. Frazzoli, M. A. Dahleh, and E. Feron. Robust hybrid control for autonomous vehicle motion planning. In *IEEE Conference on Decision and Control*, volume 1, pages 821–826, 2000.

[220] E. Frazzoli. *Robust Hybrid Control for Autonomous Vehicle Motion Planning.* PhD thesis, Massachusetts Institute of Technology, 2001.

[221] M. Stolle and C. G. Atkeson. Policies based on trajectory libraries. In *IEEE International Conference on Robotics and Automation*, pages 3344–3349, May 2006.

[222] R. Tedrake, I. R. Manchester, M. Tobenkin, and J. W. Roberts. LQR-trees: Feedback motion planning via sums-of-squares verification. *International Journal of Robotics Research*, 29:1038–1052, 2010.

[223] L. Singh and J. Fuller. Trajectory generation for a UAV in urban terrain, using nonlinear MPC. In *American Control Conference*, volume 3, pages 2301–2308 vol.3, 2001.

[224] A. Kotikalpudi. *Robust Flutter Analysis for Aeroservoelastic Systems.* PhD thesis, University of Minnesota, 2017.

[225] J. P. Hespanha. *Linear Systems Theory.* Princeton University Press, 2009.

[226] C. Moler and C. Van Loan. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Review*, 45(1):3–49, 2003.

[227] T. A. Bickart. Matrix exponential: Approximation by truncated power series. *Proceedings of the IEEE*, 56(5):872–873, 1968.

[228] M. L. Liou. A novel method of evaluating transient response. *Proceedings of the IEEE*, 54(1):20–23, 1966.

[229] W. Everling and M. L. Liou. On the evaluation of e^{AT} by power series. *Proceedings of the IEEE*, 55(3):413–413, 1967.

[230] E. Armstrong and A. Caglayan. An algorithm for the weighting matrices in the sampled-data optimal linear regulator problem. Technical report, Nasa Langley Research Center, 1976.

[231] J. Johnson and C. Phillips. An algorithm for the computation of the integral of the state transition matrix. *IEEE Transactions on Automatic Control*, 16(2):204–205, 1971.

[232] A. Levis. Some computational aspects of the matrix exponential. *IEEE Transactions on Automatic Control*, 14(4):410–411, 1969.

[233] C. Källström. Computing exp(A) and its integral. Report TFRT vol. 3053, Lund Institute of Technology, 1973.

[234] C. Van Loan. Computing integrals involving the matrix exponential. *IEEE Transactions on Automatic Control*, 23(3):395–404, 1978.

Appendix A

Supplementary Material for Chapter 5

A.1 Proof of Lemma 4

Proof. First, note that Π has the following frequency-domain representations for $i = 1, 2$:

$$\Pi\left(s\right) = \Psi_i^{\sim}\left(s\right) M_i \Psi_i\left(s\right) = \begin{bmatrix} \left(sI - A_i\right)^{-1} B_i \\ I \end{bmatrix}^{\sim} \begin{bmatrix} Q_i & S_i \\ S_i^T & R_i \end{bmatrix} \begin{bmatrix} \left(sI - A_i\right)^{-1} B_i \\ I \end{bmatrix}. \tag{A.1}$$

This yields the following two state-space realizations for Π:

$$\Pi \overset{s}{=} \left[\begin{array}{c|c} \bar{A}_i & \bar{B}_i \\ \hline \bar{C}_i & \bar{D}_i \end{array}\right] := \left[\begin{array}{cc|c} A_i & 0 & B_i \\ -Q_i & -A_i^T & -S_i \\ \hline S_i^T & B_i^T & R_i \end{array}\right] \text{ for } i = 1, 2. \tag{A.2}$$

These two realizations of Π are minimal since the A_i are the state matrices of the (assumed) minimal realizations for Ψ_i. Hence $\bar{A}_1$ and $\bar{A}_2$ share the same eigenvalues and hence are similar. Consequently, A_1 and A_2 share the same eigenvalues and hence are similar matrices. This proves the existence of a similarity transformation matrix $T_1 \in \mathbb{R}^{n \times n}$ such that:

$$A_2 = T_1 A_1 T_1^{-1}. \tag{A.3}$$

Moreover, the two minimal realizations of Π are also related by a similarity transformation:

$$\exists T \in \mathbb{R}^{2n \times 2n} : \left[\begin{array}{c|c} T\bar{A}_1 T^{-1} & T\bar{B}_1 \\ \hline \bar{C}_1 T^{-1} & \bar{D}_1 \end{array}\right] = \left[\begin{array}{c|c} \bar{A}_2 & \bar{B}_2 \\ \hline \bar{C}_2 & \bar{D}_2 \end{array}\right]. \tag{A.4}$$

Equating the $(1,1)$ blocks of (A.4) yields $T\bar{A}_1 = \bar{A}_2 T$. Using the partition $T = \left[\begin{smallmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{smallmatrix}\right]$, this is:

$$\begin{bmatrix} T_{11}A_1 - T_{12}Q_1 & -T_{12}A_1^T \\ T_{21}A_1 - T_{22}Q_1 & -T_{22}A_1^T \end{bmatrix} = \begin{bmatrix} A_2 T_{11} & A_2 T_{12} \\ -Q_2 T_{11} - A_2^T T_{21} & -Q_2 T_{12} - A_2^T T_{22} \end{bmatrix}. \tag{A.5}$$

Equating the $(1,2)$ blocks of (A.5) yields the relation $-T_{12}A_1^T = A_2 T_{12}$. However, A_1 and A_2 are also related by Equation (A.3). These two relations together yield the relation $-T_1^{-1}T_{12}A_1^T = A_1 T_1^{-1} T_{12}$. This can be rewritten as the Lyapunov Equation $A_1\bar{Z} + \bar{Z}A_1^T = 0$ where $\bar{Z} := T_1^{-1}T_{12}$. Since A_1 is Hurwitz it follows that $\bar{Z} = 0$ is the unique solution to this Lyapunov Equation. Moreover, $\bar{Z} = 0$ implies $T_{12} = 0$, i.e. T is block lower triangular. Equating the $(1,1)$ and $(2,2)$ blocks of (A.5) then implies $T_{11} = T_1$ and $T_{22} = T_1^{-T}$. Finally, denoting $\bar{X} := T_1^T T_{21}$ yields the block partitions $T = \left[\begin{smallmatrix} T_1 & 0 \\ T_1^{-T}\bar{X} & T_1^{-T} \end{smallmatrix}\right]$ and $T^{-1} = \left[\begin{smallmatrix} T_1^{-1} & 0 \\ -\bar{X}T_1^{-1} & T_1^T \end{smallmatrix}\right]$.

Equating the $(2,1)$ blocks of (A.5) yields the Lyapunov Equation $A_1^T\bar{X} + \bar{X}A_1 = Q_1 - T_1^T Q_2 T_1$. The solution $\bar{X} = \bar{X}^T$ to this Lyapunov Equation exists and is unique because A_1 is Hurwitz.

Equating the $(1,2)$ blocks of (A.4) yields:

$$B_2 = T_1 B_1 \text{ and } T_1^{-T}\bar{X}B_1 = T_1^{-T}S_1 - S_2. \tag{A.6}$$

Equating the $(2,2)$ blocks of (A.4) yields $\bar{D}_1 = \bar{D}_2$ which further implies $R_1 = R_2$. Finally, the following expressions are obtained for Q_2, S_2, and R_2:

$$Q_2 = T_1^{-T}\left(Q_1 - \bar{X}A_1 - A_1^T\bar{X}\right)T_1^{-1},\ S_2 = T_1^{-T}\left(S_1 - \bar{X}B_1\right), \text{ and } R_2 = R_1. \tag{A.7}$$

Equations (A.3), (A.6), and (A.7) prove statements (1), (2) and (3), respectively. □

A.2 Matrix Dilation Result

Lemma 7. *Let $X = X^T \in \mathbb{R}^{n\times n}$, $Y = Y^T \in \mathbb{R}^{n\times n}$, and a positive integer n_F be given. Then there exist matrices $X_2, Y_2 \in \mathbb{R}^{n\times n_F}$ and symmetric matrices $X_3, Y_3 \in \mathbb{R}^{n_F\times n_F}$, satisfying*

$$X_3 > 0 \text{ and } \begin{bmatrix} X & X_2 \\ X_2^T & X_3 \end{bmatrix}^{-1} = \begin{bmatrix} Y & Y_2 \\ Y_2^T & Y_3 \end{bmatrix} \tag{A.8}$$

if and only if

$$X - Y^{-1} \geq 0 \text{ and } \operatorname{rank}\left(X - Y^{-1}\right) \leq n_F. \tag{A.9}$$

Proof. For sufficiency, assume that the conditions given in (A.8) hold. By the matrix inversion lemma,

$$\begin{bmatrix} X & X_2 \\ X_2^T & X_3 \end{bmatrix}^{-1} = \begin{bmatrix} \left(X - X_2X_3^{-1}X_2^T\right)^{-1} & -\left(X - X_2X_3^{-1}X_2^T\right)^{-1}X_2X_3^{-1} \\ -\left(X_3 - X_2^TX^{-1}X_2\right)^{-1}X_2^TX^{-1} & \left(X_3 - X_2^TX^{-1}X_2\right)^{-1} \end{bmatrix}. \tag{A.10}$$

Comparing the expressions in (A.8) with (A.10) yields $Y = \left(X - X_2X_3^{-1}X_2^T\right)^{-1}$. Applying the matrix inverse to both sides of this relation and rearranging terms yields $X - Y^{-1} = X_2X_3^{-1}X_2^T$. Thus the assumption $X_3 > 0$ implies $X - Y^{-1} \geq 0$. Further, since $X_2 \in \mathbb{R}^{n\times n_F}$, rank $\left(X - Y^{-1}\right) \leq n_F$.

For necessity, assume that the conditions in (A.9) hold. Since rank $\left(X - Y^{-1}\right) \leq n_F$, $\exists X_2 \in \mathbb{R}^{n\times n_F}$ so that $X - Y^{-1} = X_2X_2^T \geq 0$. This relation can be rearranged to obtain $Y = \left(X - X_2X_2^T\right)^{-1}$. By the matrix inversion lemma, $\left[\begin{smallmatrix} X & X_2 \\ X_2^T & I \end{smallmatrix}\right]^{-1} = \left[\begin{smallmatrix} Y & -YX_2 \\ -X_2^TY & X_2^TYX_2+I \end{smallmatrix}\right]$. Hence, set $X_3 = I$, $Y_2 = -YX_2$, and $Y_3 = I + X_2^TYX_2$. □

Appendix B

Implementation Considerations

B.1 Controller Order Reduction

Design B of the single surface controller (Section 7.7.2) has 15 states. Its state order is reduced to decrease its computational overhead. In particular, a balanced residualization [55] of K_S results in the eighth-order controller K_S^8. The magnitude and the phase responses of all the channels of K_S^8 match those of K_S very well up to 150 $\mathrm{rad\,s^{-1}}$, which is more than twice the bandwidth of the elevon actuator. The fastest pole of K_S^8 is located at 15 Hz, which is sufficiently lower than the Nyquist frequency of 45 Hz of the flight computer (Table 3.1).

B.2 Real-Time Discretization

For a typical autopilot, the controller order reduction is followed by discretization, wherein the continuous-time controller K_S^8 is converted to discrete-time. The discretization needs to be performed only once if the flight computer has a fixed sample rate[1]. However, the flight computer of the Vireo has a sample rate that varies within any given flight between 85 Hz and 100 Hz. It is inadequate to simply discretize K_S^8 at the programmed sample rate of 100 Hz because there may be instances during the flight wherein the intersample time is greater than 0.01 s. Such *off-nominal* intersample times shift the frequency response of the discretized controller away from the response intended by

[1] The `C2D` function in Matlab's Control System Toolbox automates this process.

the designer. In order to avoid such scenarios, this thesis implements a real-time discretization algorithm, wherein the controller is discretized on-board the flight computer whenever a new measurement sample becomes available. This algorithm guarantees that the actual frequency response of the controller closely matches the intended frequency response up to the bandwidth of the elevon actuator. The remainder of this section provides the details of the real-time discretization algorithm.

B.2.1 Continuous-Time and Discrete-Time State-Space Realizations

Consider a continuous-time, LTI system with the following state-space realization:

$$\dot{x}(t) = Ax(t) + Bu(t), \tag{B.1}$$

$$y(t) = Cx(t) + Du(t), \tag{B.2}$$

and some initial condition $x(t_0) = x_0$. Here, $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ is the input vector, $y \in \mathbb{R}^p$ is the output vector, and $t \in \mathbb{R}^+$ is time. The solution to Equation B.1 defines the trajectory of the state vector and is given by [225]:

$$x(t) = e^{A(t-t_0)}x(t_0) + \int_{t_0}^{t} e^{A(t-s)}Bu(s)\,ds, \tag{B.3}$$

where e^A denotes the matrix exponential of A.

A digital sampling system observes the state of this system at the time instants t_k for $k = 1, 2, \dots$, where k is the enumeration index. Let t_k and t_{k+1} denote two consecutive sampling time instants. The intersample time $T_k \coloneqq t_{k+1} - t_k$ is a non-negative real number that varies with k, and is computed upon the arrival of the $(k+1)^{th}$ sample. The intersample trajectory of the state vector is exactly given by:

$$x(t_{k+1}) = e^{AT_k}x(t_k) + \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-s)}Bu(s)\,ds. \tag{B.4}$$

Under the assumption that the intersample behavior of the input is constant, i.e. zero-order hold, $u(s)$ is equal to $u(t_k)$ for all $s \in [t_k, t_{k+1})$. Further, applying the change of variables $s' = s - t_k$ to Equation (B.4) yields:

$$x\left(t_{k+1}\right) = e^{AT_k} x\left(t_k\right) + \int_0^{T_k} e^{A\left(T_k - s'\right)} ds' \; Bu\left(t_k\right). \tag{B.5}$$

Thus, the discrete-time system has the following state-space realization at time t_k:

$$\begin{aligned} x\left(t_{k+1}\right) &= \Phi\left(T_k\right) x\left(t_k\right) + \Gamma\left(T_k\right) u\left(t_k\right), & \text{(B.6)} \\ y\left(t_k\right) &= Cx\left(t_k\right) + Du\left(t_k\right), & \text{(B.7)} \end{aligned}$$

where $\Phi\left(T_k\right) := e^{AT_k}$ and $\Gamma\left(T_k\right) := \int_0^{T_k} e^{A\left(T_k - s'\right)} ds' \; B$ are the one-step state-transition and input-to-state matrices, respectively.

The matrix exponential e^{AT_k} is formally defined by the convergent power series:

$$e^{AT_k} = I + \frac{AT_k}{1!} + \frac{\left(AT_k\right)^2}{2!} + \cdots = \sum_{j=0}^{\infty} \frac{\left(AT_k\right)^j}{j!}. \tag{B.8}$$

Since e^{AT_k} is a transcendental function of AT_k, it is approximated using numerical methods. Moler and Van Loan [226] present a detailed survey of the plethora of methods available to approximate e^{AT_k}, e.g. inverse Laplace transform, Cayley-Hamilton, eigenvalue/eigenvector, Pade approximation[2], Chebyshev rational approximation, etc. This thesis uses the direct Taylor series approximation of Equation B.8 [227]:

$$e^{AT_k} \approx S_N\left(AT_k\right) = \sum_{j=0}^{N} \frac{\left(AT_k\right)^j}{j!}, \tag{B.9}$$

where N is an integer large enough so that $\left\| e^{AT_k} - S_N\left(AT_k\right) \right\|_2 \leq \epsilon$ for some prescribed tolerance ϵ [227–229]. Setting $N = 1$ simply yields the Euler discretization method, which is accurate only when $T_k \to 0$. Equation (B.9) approximates $\Phi\left(T_k\right)$ and is easy to implement in real-time since it only involves matrix multiplications and additions.

Next, consider the definition of $\Gamma\left(T_k\right)$. While several methods exist that directly approximate the integral of e^{AT_k} [230–233], it is also possible to express the integral itself as a matrix exponential [234]. This would allow a single algorithm to estimate both $\Phi\left(T_k\right)$ and $\Gamma\left(T_k\right)$. In order to do this, Theorem 1 from [234] is rephrased using

[2] The `EXPM` function in Matlab uses the Pade approximation after scaling and squaring its argument.

the notation of this section.

Theorem 4 ([234]). *Let the constant matrices $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ be given. If the matrix M is defined by:*

$$M = \begin{bmatrix} A & B \\ 0_{m\times n} & 0_{m\times m} \end{bmatrix}, \tag{B.10}$$

then for all $T_k \geq 0$

$$e^{MT_k} = \begin{bmatrix} \Phi\left(T_k\right) & \Gamma\left(T_k\right) \\ 0_{m\times n} & I_{m\times m} \end{bmatrix}, \tag{B.11}$$

where $\Phi\left(T_k\right) := e^{AT_k}$ and $\Gamma\left(T_k\right) := \int_0^{T_k} e^{A(T_k - s')} ds'\, B$.

This theorem and the Taylor series approximation together enable the implementation of the discrete-time system given in Equations (B.6) and (B.7). In particular, the flight computer stores the time-invariant matrices A, B, C, and D in its memory and implements an algorithm. When a new measurement sample arrives, the algorithm:

1. computes the intersample time $T_k = t_{k+1} - t_k$,

2. composes the matrix M given in Equation (B.10),

3. computes a N^{th}-order Taylor series approximation of e^{MT_k} using Equation (B.9),

4. partitions the resulting $(n+m)\times(n+m)$ matrix as per Equation (B.11) to yield the *estimates* of $\Phi\left(T_k\right)$ and $\Gamma\left(T_k\right)$,

5. and uses the estimates of $\Phi\left(T_k\right)$ and $\Gamma\left(T_k\right)$ in Equation (B.6) and C and D in Equation (B.7) to implement the discrete-time controller.

B.2.2 Selection of the Taylor Series Order

As such, the algorithm described above manages the variable sample rate of the flight computer. The next task is to select the order N of the Taylor series approximation, which controls the trade-off between the accuracy of the discretization and its computational complexity. One approach is to first prescribe the tolerance ϵ on the approximation error and then select N using the guidelines given in the literature [227–229].

Another approach is to compare the Bode diagrams of the controllers resulting from different orders of the Taylor series approximation against the *exact* discretization, which is obtained using the C2D function in Matlab. The C2D function uses the Pade approximation for the discretization and is equivalent to selecting a very large N. The latter approach is followed due to its simplicity. From a comparison of the Bode diagrams of the controllers, it is concluded that $N = 3$ yields a sufficiently accurate discretization.

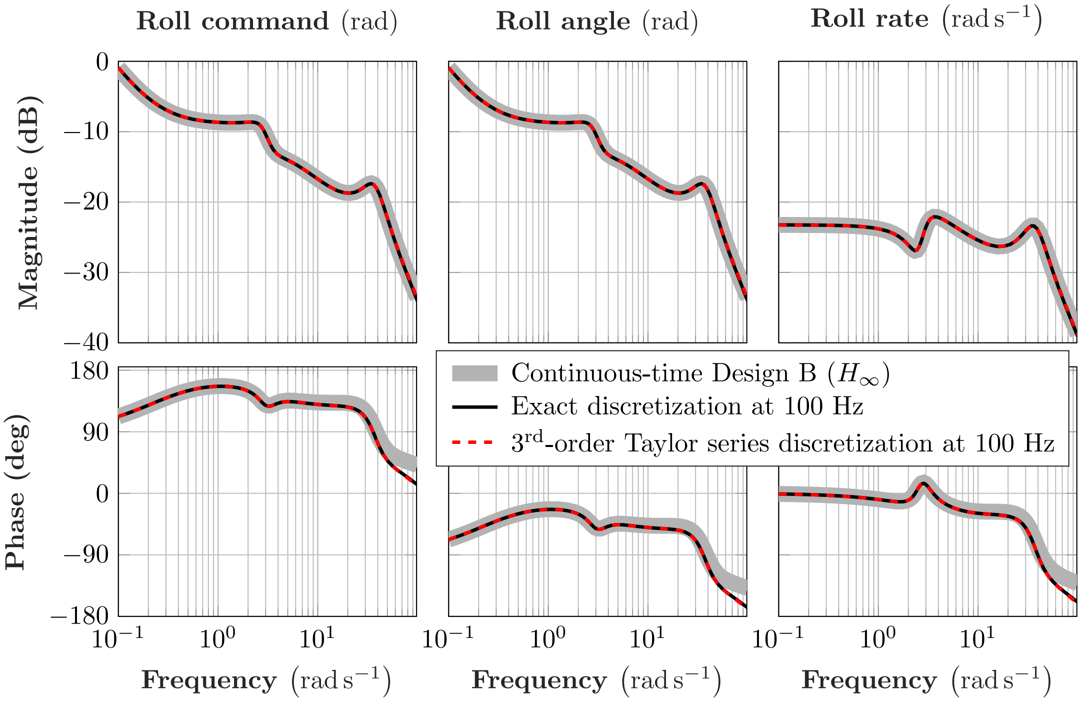


Figure B.1: The Bode diagrams of: (1) the continuous-time H_∞ controller, (2) the exact discretization, and (3) the third-order Taylor series discretization. The discretization is illustrated using an example sample rate of 100 Hz. The input channels are labeled at the top of each plot. The output is the aileron command.

Figure B.1 shows the Bode diagrams of the continuous-time H_∞ controller, the exact discretization, and the third-order Taylor series discretization. The figure indicates that there is virtually no difference between the exact discretization and the third-order Taylor series discretization. However, both discretizations inevitably deviate from the continuous-time controller for frequencies above 50 $\mathrm{rad\,s^{-1}}$, which is close to the bandwidth of the elevon actuator. Further, the figure depicts the discrete-time controllers at a sample rate of 100 Hz for illustrative purposes. The controllers match closely also for 85 Hz and for all sample rates in between. Overall, the third-order Taylor series approximation results in two multiplications and three additions involving a 11×11 matrix. Benchtop experiments reveal that the load average of the flight computer is significantly below one[3] even with this overhead, thereby clearing the controller for flight tests.

[3] A load average greater than one implies that computations overflow from one frame to the next.

Appendix C

List of Flight Tests

The Vireo has clocked a total of 35 flights, with each flight lasting around 10 min, for the research presented in this thesis. Around 300 min of the total flying time have been under autopilot control. Table C.1 lists the date and a short description of each flight. It also classifies the nature of each flight using an acronym, as defined below.

- **BAS** System checkout, testing estimates of gains, main loop timing, sensor calibration, navigation filter tuning.
- **SID** System identification experiments.
- **NC** Tuning, testing, and validation of the nominal controller and its components.
- **FTR** Testing new features, e.g. real-time discretization, control law switching, autoland, etc.
- **LTD** Limiting the authority of the nominal controller. This is a stepping-stone to evaluating the fault-tolerant controller.
- **FTC** Injecting an elevon fault and simultaneously reconfiguring the controller. Tuning, testing, and validation of the fault-tolerant controller and its components.
- **FDD** Injecting an elevon fault without controller reconfiguration. The data collected is used for off-line fault diagnosis.

Table C.1: List of flight tests conducted with the Vireo.

FLT	**Date**	**Type**	**Test Description**
1	2016-05-17	BAS	First flight. Checkout of airframe and hardware components. Verification of telemetry and data logging.
2	2016-07-01	BAS	First attempt at closing loop using estimates of gains. Calibration of IMU and magnetometer.
3	2016-07-29	BAS	Reattempt at closing loop using estimates of gains.
4	2016-08-17	BAS	Testing code changes affecting main loop timing and synchronization. Magnetometer integration with EKF.
5	2017-04-21	SID	Open-loop elevator chirps for system identification.
6	2017-04-21	SID	Open-loop aileron chirps for system identification.
7	2017-05-25	NC	Testing model-based gains for nominal controller.
8	2017-05-25	NC	Tuning airspeed tracker and roll controller.
9	2017-06-02	NC	Responses to altitude and airspeed step commands.
10	2017-06-02	NC	Tuning pitch and roll attitude controllers.
11	2017-06-09	NC	Tuning pitch, roll, airspeed, and altitude controllers.
12	2017-06-09	NC	Responses to altitude and airspeed step commands.
13	2017-08-02	FTR	Testing real-time discretization algorithm.
14	2017-08-02	FTR	Testing H_∞ controller using both control surfaces.
15	2017-08-11	NC	Lateral-directional controller validation.
16	2017-08-11	FTR	Testing control law switching logic.
17	2017-08-11	LTD	Disengaging pitch attitude controller.
18	2017-08-23	NC	Longitudinal controller validation.
19	2017-08-23	LTD	Testing altitude tracker and airspeed damper after disengaging pitch attitude controller.
20	2017-08-23	FTR	Testing autoland with nominal controller.
21	2017-08-23	LTD	Testing TECS without pitch attitude control.
22	2017-08-30	LTD	Tuning TECS without pitch attitude control.
23	2017-08-30	LTD	Testing TECS without pitch attitude control. First autoland without pitch attitude control.

Table C.1: List of flight tests conducted with the Vireo.

FLT	Date	Type	Test Description
24	2017-08-30	FTC	Right elevon stuck at 5° trailing-edge up. First flight of single surface controller.
25	2017-08-30	FTC	Right elevon stuck at 5° trailing-edge up. First autoland using single surface controller.
26	2017-09-07	LTD	Tuning w_b in fault-tolerant TECS.
27	2017-09-07	FTC	Right elevon stuck at 5° trailing-edge up with TECS and single surface controller (A). Autoland successful.
28	2017-09-20	FTC	Right elevon stuck at 5° trailing-edge up with TECS and single surface controller (A). Autoland successful.
29	2017-09-20	FTC	Right elevon stuck at 5° trailing-edge up with TECS and single surface controller (A). Autoland not attempted.
30	2017-09-20	FTC	Right elevon stuck at 5° trailing-edge up with TECS and single surface controller (A). Autoland not attempted.
31	2017-09-29	-	Launch failure due to insufficient airspeed.
32	2017-10-05	FDD	Right elevon stuck at 5° trailing-edge up with nominal controller. Data used for off-line fault diagnosis.
33	2017-10-05	FDD	Right elevon stuck at 5° trailing-edge up with nominal controller. Data used for off-line fault diagnosis.
34	2017-10-05	FTC	Right elevon stuck at 5° trailing-edge up with TECS and single surface controller (B). Autoland successful.
35	2017-10-05	FTC	Right elevon stuck at 5° trailing-edge up with TECS and single surface controller (B). Autoland successful.

The following are noted with regard to Table C.1.

1. Before TECS was introduced, the nominal controller tracked altitude using throttle and airspeed using pitch angle, e.g. FLT8.

2. Before Design B was tested as a single surface controller, it was tested to command both elevons together as a virtual aileron. This helped build confidence in the real-time discretization algorithm (FLT13) and the H_∞ controller (FLT14).

3. Some ideas that were tested never fully materialized into the final control architecture. For example, in order to increase the phugoid mode damping ratio, FLT19 evaluated an airspeed damper. The results were not promising, but paved the way for tuning the weight w_b in FLT26.

Flight data All flights are documented and the logged flight data is publicly available on the UMN Digital Conservancy. The data may be retrieved by visiting the following website and searching for the word "vireo" in the search bar.

`https://conservancy.umn.edu/handle/11299/163580`

Flight videos The Vireo carried and on-board camera in some of the flights that tested the fault detection and isolation (FDI) algorithm and the fault-tolerant controller (FTC). Table C.2 provides the URLs to three such flight videos. The videos have been shortened to capture only the highlights of the tests and are self-explanatory.

Table C.2: List of flight video summaries.

FLT	**Test point**	**Test Description**	**URL**
33	T2	FDI: Stuck fault is injected in the right elevon at its trim deflection of $-5°$.	`https://youtu.be/w-yjEvjMIck`
33	T6	FDI: Stuck fault is injected in the right elevon at $-7°$ (trim elevon deflection is $-5°$).	`https://youtu.be/3Nnbmtgg7SY`
35	–	FTC: Stuck fault is injected in the right elevon at -5 deg (trim elevon deflection is -2 deg). The fault-tolerant controller manages the fault and allows for an automated landing.	`https://youtu.be/rGIw71kiu4w`

Appendix D

Additional Flight Test Results of the Fault-Tolerant Controller

Section 7.9 presents the results of only FLT27 and FLT35. This appendix presents the results of the remaining flights from Table 7.4 validating the fault-tolerant controller. In all the time history plots, the dashed line denotes the autopilot command and the solid line denotes the aircraft response. The discussion follows the same sequence as in Sections 7.9.3 and 7.9.4.

D.1 FLT28: Autoland Using Design A

An autoland demonstration using Design A of the single surface controller and the fault-tolerant total energy controller is conducted during FLT28 on 2017-09-20. Shortly after the manual takeoff, the pilot takes the Vireo up to 300 ft AGL and engages the autopilot at 1072 s. Figure D.1 shows the ground track of the Vireo from the autopilot engagement ($*$) until the landing on the virtual runway ($\circ$). The ground track is plotted in a local North-East reference frame, whose origin coincides with the target landing spot chosen by the UAS operator prior to the start of the flight. The various events occurring in the flight are annotated on the ground track and are explained one-by one.

The autopilot engagement ($*$) activates the nominal controller, which takes the Vireo into a left banked circle hold at an altitude of 300 ft AGL, an airspeed of 15.4 $\mathrm{m\,s^{-1}}$,

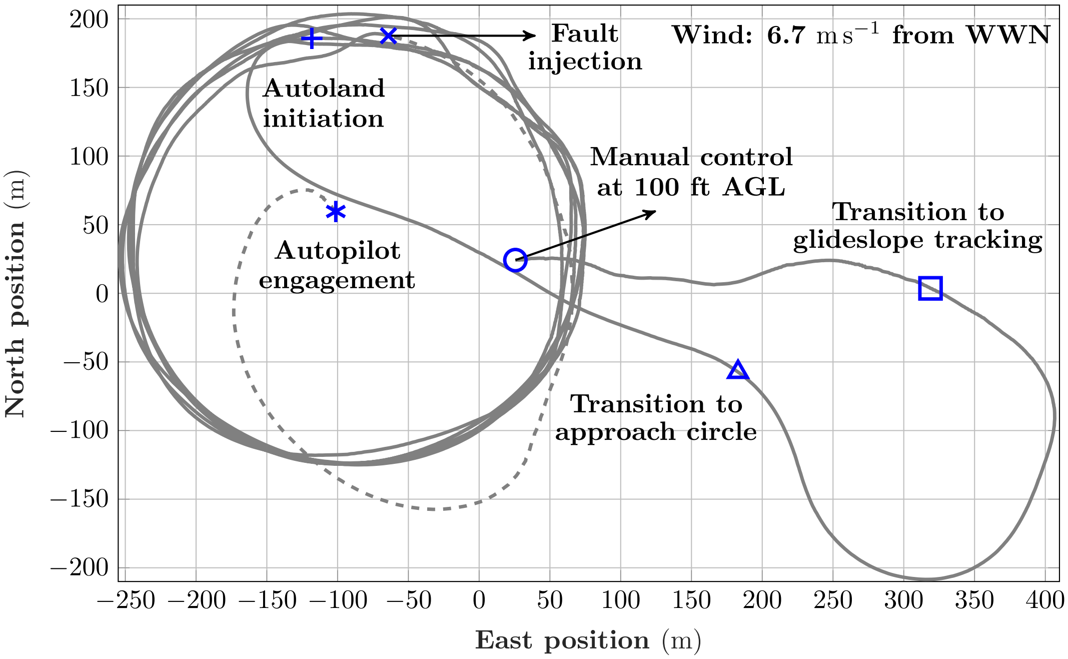


Figure D.1: FLT28: The ground track of the Vireo indicating different flight events. The solid line indicates the times when the single surface controller is active.

and a turn radius of 150 m. The center of the circle $(-87.8, 32.3)$ m is automatically set by the flight computer to the point where the airplane climbs past the altitude of 150 ft AGL. The Vireo traces a counterclockwise circle when viewed in the North-East plane. This initial phase, wherein the nominal controller is active, is denoted by the dashed line starting at $*$ in Figure D.1. As with the other flights, a counterclockwise circle is chosen because the right elevon is faulted. At 1128 s, the UAS operator injects a -5° stuck fault in the right elevon ($\times$). The flight computer sends a constant PWM command to the right elevon to simulate the stuck fault. Since the demonstration excludes fault diagnosis, the fault-tolerant controller is activated immediately upon the fault injection. The ground track following the fault injection ($\times$) is denoted using the solid line. For the next 7.7 min, the Vireo continues to track the circle. The cross-track error standard deviations before and after the fault injection are 5.68 m and 6.5 m, respectively.

At 1594 s, the UAS operator initiates the autoland sequence ($+$) causing the Vireo to peel away from the circle hold and fly towards the approach circle. Since the prevailing winds on that day are 6.7 $\mathrm{m\,s^{-1}}$ from the West-Northwest, the operator commands a Westerly landing to take advantage of the headwind. The Vireo enters the approach circle (△) and traverses about three-quarters of the circle. Once the Vireo is heading due West, it exits the circle and transitions to glideslope tracking (□). The remainder of the autoland is best explained by the next figure.

Figure D.2 shows the terminal flight path of the Vireo as seen by an observer standing on the ground and facing due North. The target approach speed is 15.4 $\mathrm{m\,s^{-1}}$ and the target flight path is a 6° glideslope that intersects the virtual runway at an East position of zero. As seen in Figure D.2, the Vireo is 24.36 m above the glideslope when it begins the glideslope tracking (□). Consequently, the total energy controller reduces the throttle, causing the Vireo to lose altitude and intersect the glideslope at around 173 m from the target touchdown point. The phugoid mode is excited at this point, most probably because of the high winds observed during this particular flight. As with FLT27, a trough of the phugoid mode causes the aircraft to intersect the virtual runway 38 m short of the intended touchdown point. Shortly thereafter, the pilot takes manual control of the aircraft (○). Since the aircraft is below the virtual runway, a go-around is not attempted and the pilot simply lands the aircraft. This final phase of manual control is not shown in Figure D.2 to avoid clutter. Next, it is instructive to evaluate the *short-term* and the *long-term* aircraft responses after the fault injection.

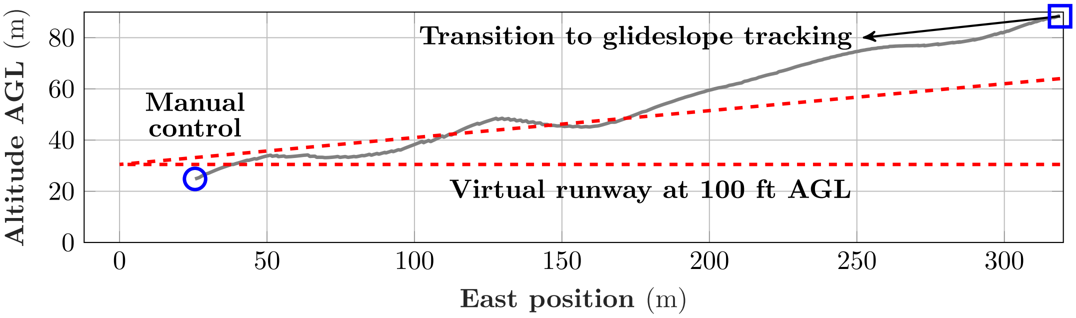
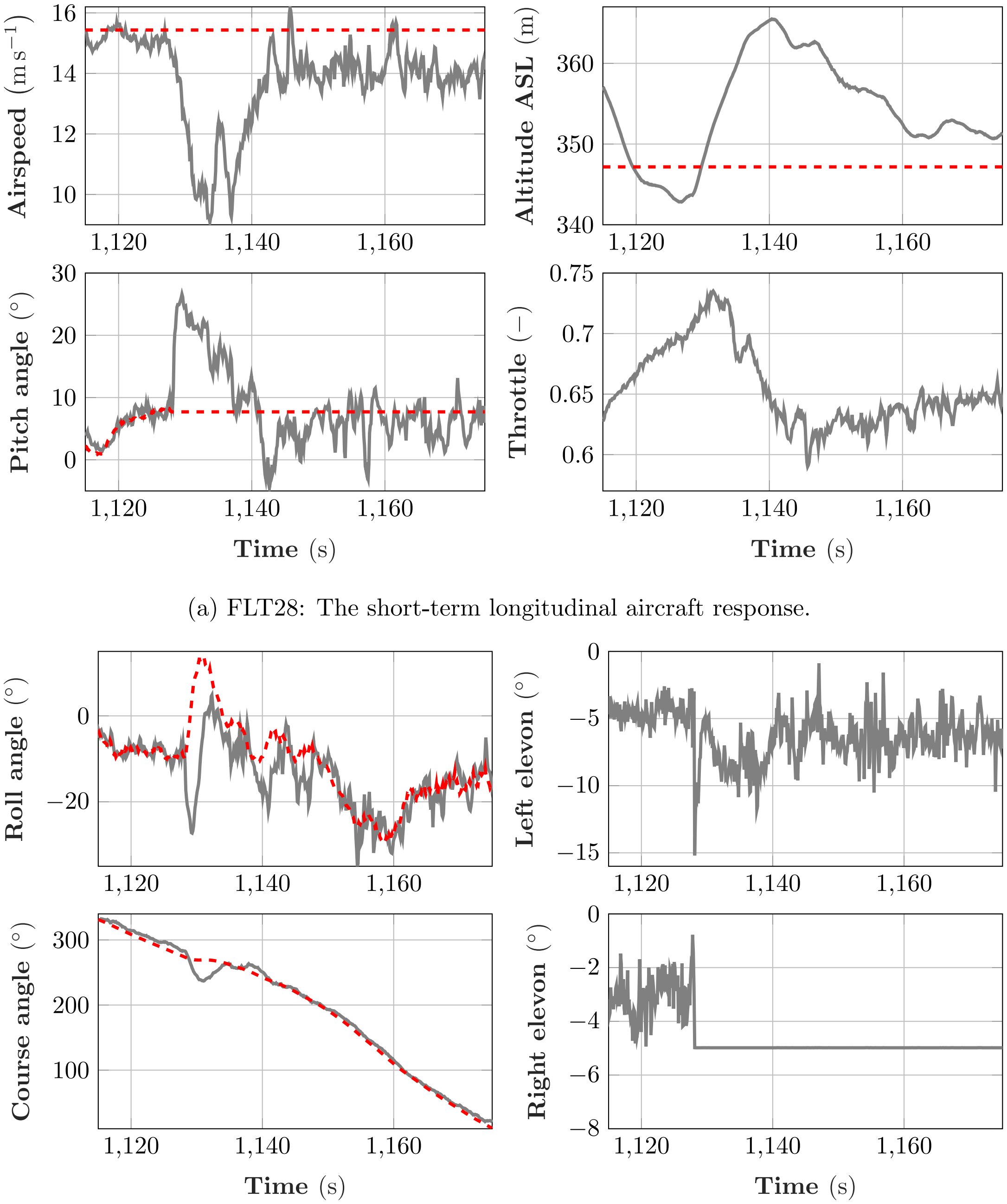


Figure D.2: FLT28: The Vireo tracks a glideslope during the final approach. The pilot takes manual control of the aircraft once it reaches the virtual runway at 100 ft AGL.

Figure D.3 shows the short-term longitudinal and lateral-directional aircraft responses as well as the control commands after the fault injection at 1128 s. The longitudinal motion variables (Figure D.3a) are close to their respective reference commands and the mean value of the right elevon (Figure D.3b) is approximately $-3°$ before the fault. The fault causes the aircraft to pitch nose-up to about 27° at 1130 s, which in turn decreases the airspeed to about 10 $\mathrm{m\,s^{-1}}$ and increases the altitude by about 18 m. The fault-tolerant total energy controller responds by opening the throttle to 0.725. Around 20 s after the fault, the airspeed, the pitch angle, and the throttle return closer to their respective reference commands, but exhibit higher variability than before the fault. The altitude takes an additional 20 s to return closer to its reference command. As for the lateral-directional motion variables (Figure D.3b), the fault causes the Vireo to momentarily roll leftward, as seen in the downward spike in the roll angle at 1129 s. This is quickly corrected by the single surface controller. Finally, since the failed position of the right elevon ($-5°$) is slightly larger in magnitude than its mean value before the fault ($-3°$), the trim airspeed reduces.

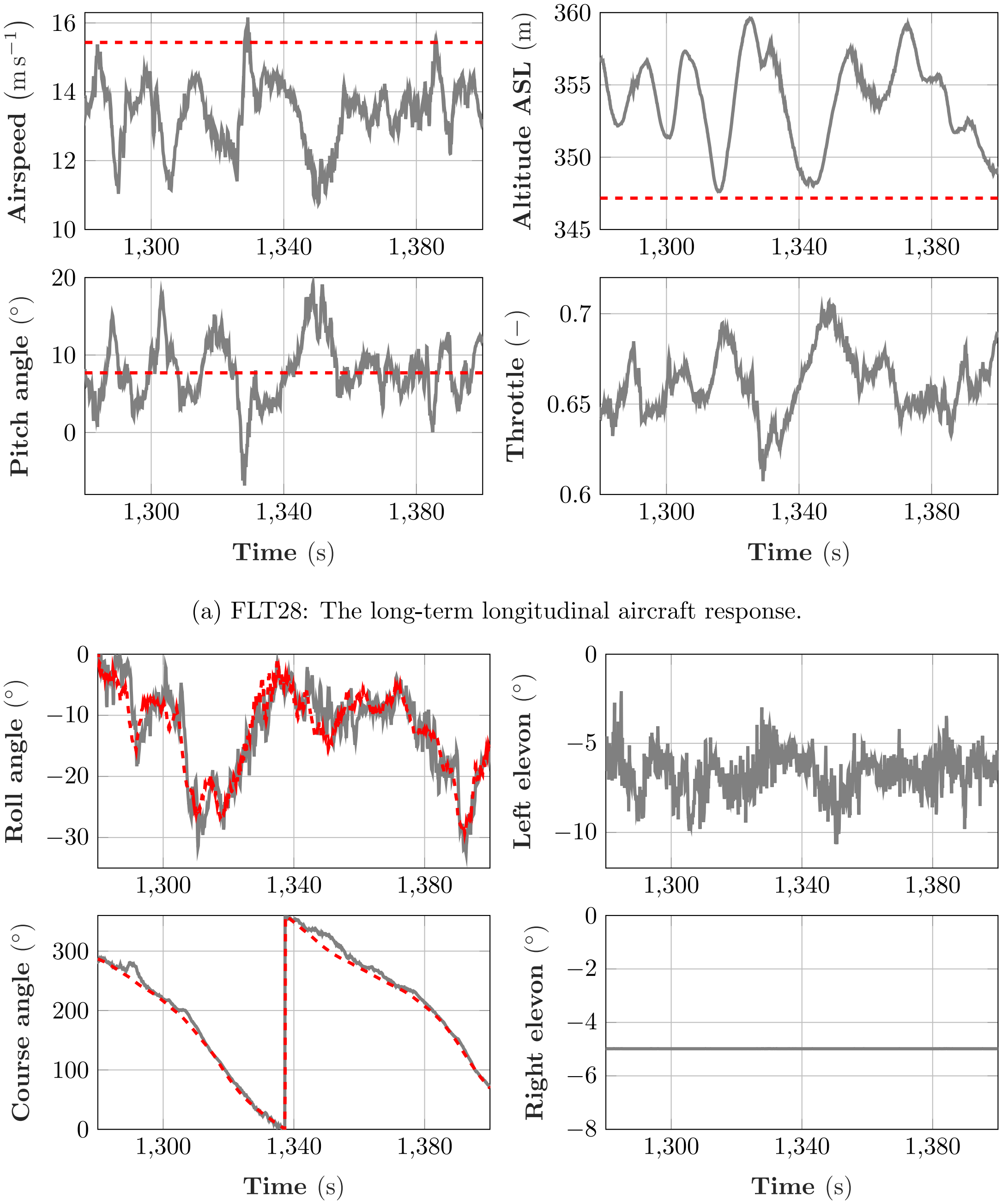
Figure D.4 follows the same layout as Figure D.3, except that it shows the long-term aircraft responses over two minutes after the fault. The new trim airspeed (Figure D.4a) is around 13.4 $\mathrm{m\,s^{-1}}$, which is a constraint imposed by the failed right elevon. Since the airspeed is below its reference value, the total energy controller compensates by increasing the altitude by about 7 m above its reference value. The steady-state errors in the airspeed and the altitude offset each other because $\bar{K}_{TECS}$ tracks the mixed energy, as shown in Figure D.5. The phugoid mode is visible in the airspeed and the altitude, with a time period of about 10 s (Figure D.4a). In particular, the airspeed varies within ± 2 $\mathrm{m\,s^{-1}}$ of 13.4 $\mathrm{m\,s^{-1}}$, the altitude varies within ± 5 m of 354 m, and the pitch attitude varies within $\pm 6°$ of 8.8°. The RMS tracking error in the mixed energy is 26.6 J, which is around 1% of the reference value. Since the aircraft is rolled leftward, the left elevon is deflected slightly more (mean value of $-6.8°$ in Figure D.4b) than the right elevon. In addition, the RMS tracking errors in the roll and the course angles (Figure D.4b) are 3° and 10.4°, respectively.



(a) FLT28: The short-term longitudinal aircraft response.

(b) FLT28: The short-term lateral-directional aircraft response and the elevon commands.

Figure D.3: FLT28: The short-term aircraft response after the fault injection at 1128 s.



(a) FLT28: The long-term longitudinal aircraft response.

(b) FLT28: The long-term lateral-directional aircraft response and the elevon commands.

Figure D.4: FLT28: The long-term aircraft response after the fault injection at 1128 s.

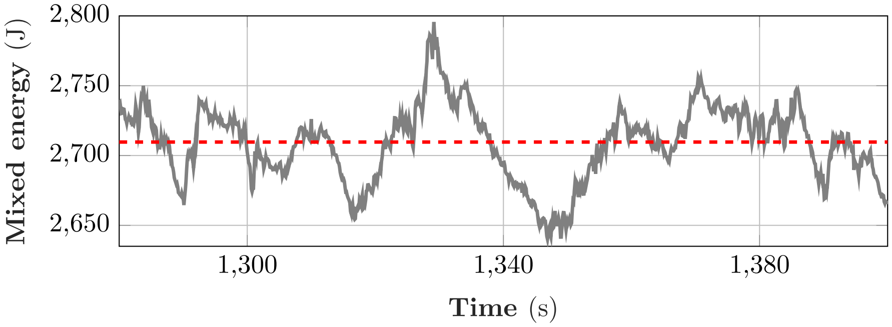


Figure D.5: FLT28: The total energy controller tracks the mixed energy after the fault.

D.2 FLT29: Flight Using Design A

FLT29 on 2017-09-20 validates Design A of the single surface controller and the fault-tolerant total energy controller. However, unlike FLT28, an autoland is not attempted. Shortly after the manual takeoff, the pilot takes the Vireo up to 225 ft AGL and engages the autopilot at 1420 s. Figure D.6 shows the ground track of the Vireo from the autopilot engagement ($*$) until the pilot takes manual control ($\circ$). The ground track is plotted in a local North-East reference frame and annotated with various flight events.

The autopilot engagement ($*$) activates the nominal controller, which takes the Vireo into a left banked circle hold at an altitude of 225 ft AGL, an airspeed of 15.4 $\mathrm{m\,s^{-1}}$, and a turn radius of 100 m. The center of the circle is at the origin of the North-East reference frame (Figure D.6) and the Vireo traces a counterclockwise circle. This initial phase, wherein the nominal controller is active, denoted by the dashed line starting at $*$ in Figure D.6. As with the other flights, a counterclockwise circle is chosen because the right elevon is faulted. At 1478 s, the UAS operator injects a $-5°$ stuck fault in the right elevon ($\times$). The flight computer sends a constant PWM comand to the right elevon to simulate the stuck fault. Since the demonstration excludes fault diagnosis, the fault-tolerant controller is activated immediately upon the fault injection. The ground track following the fault injection ($\times$) is denoted using the solid line. For the next 7.7 min, the Vireo continues to track the circle. The cross-track error standard deviations before and after the fault injection are 4.37 m and 4.54 m, respectively. Unlike the previous flights, an autoland is not attempted. Thus, the pilot takes manual control ($\circ$) at 1940 s and

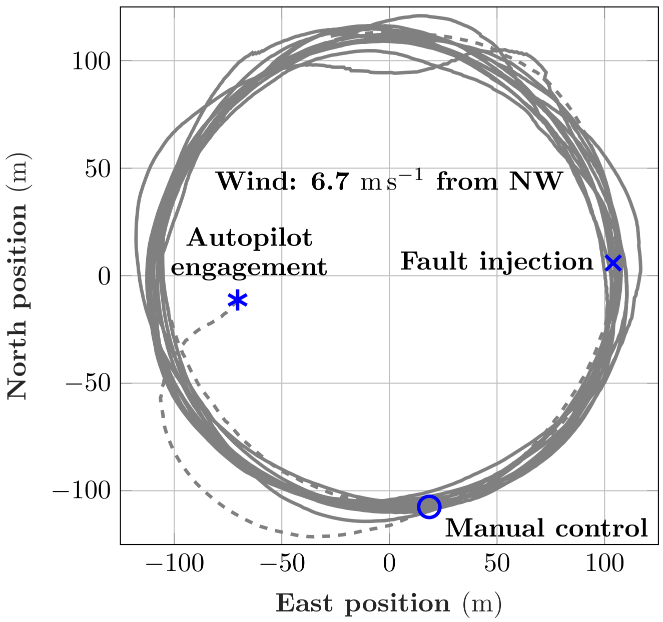
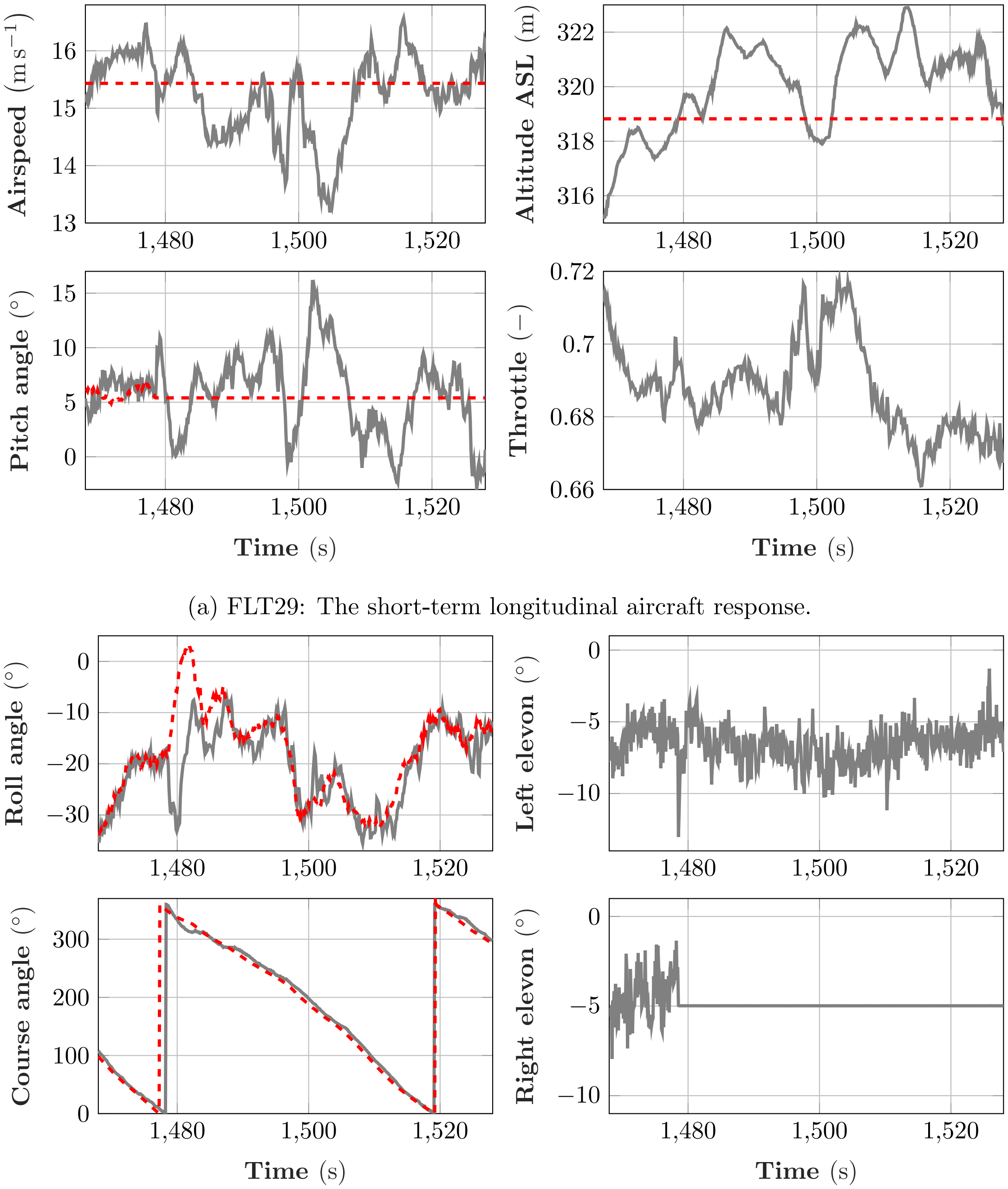


Figure D.6: FLT29: The ground track of the Vireo indicating different flight events. The solid line indicates the times when the single surface controller is active.

lands the aircraft. Next, it is instructive to evaluate the *short-term* and the *long-term* aircraft responses after the fault injection.

Figure D.7 shows the short-term longitudinal and lateral-directional aircraft responses as well as the control commands after the fault injection at 1478 s. The longitudinal motion variables (Figure D.7a) are close to their respective reference commands and the mean deflection of the right elevon (Figure D.7b) is approximately −4.3° before the fault. Since this is close to the fault magnitude (−5°), large variations are not seen in the airspeed and the altitude immediately after the fault injection. However, about 20 s after the fault injection, the airspeed and the altitude start to exhibit higher variability. In addition, the single surface controller momentarily deflects the left elevon to about −12° (Figure D.7b) which in turn causes the aircraft to roll sharply to the left at 1480 s. However, the roll angle quickly recovers and starts tracking its reference command. Finally, since the failed position of the right elevon (−5°) is slightly larger in magnitude than its mean value before the fault (−4.3°), the trim airspeed reduces.

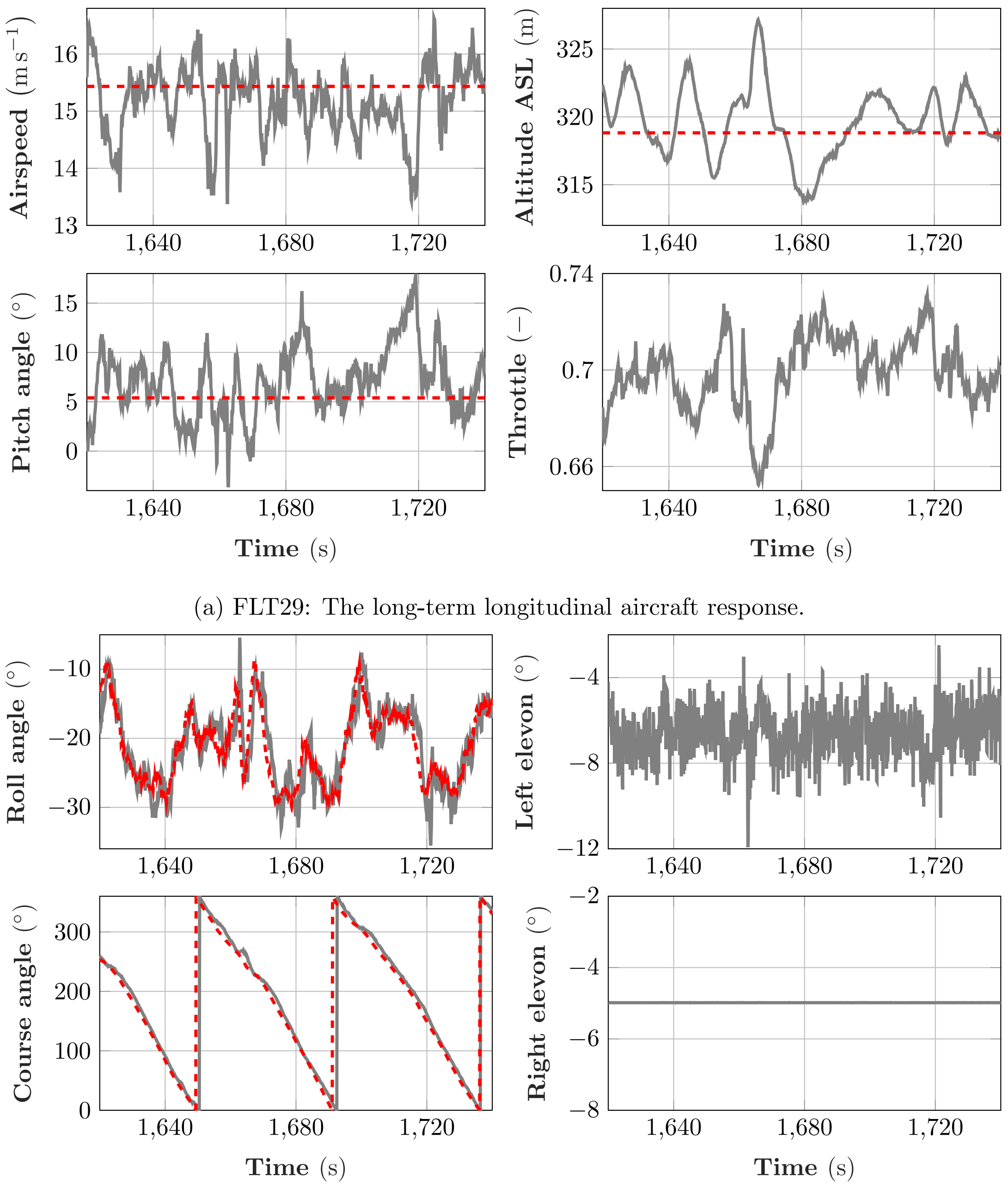
Figure D.8 follows the same layout as Figure D.7, except that it shows the long-term



(a) FLT29: The short-term longitudinal aircraft response.

(b) FLT29: The short-term lateral-directional aircraft response and the elevon commands.

Figure D.7: FLT29: The short-term aircraft responses after the fault injection at 1478 s.



(a) FLT29: The long-term longitudinal aircraft response.

(b) FLT29: The long-term lateral-directional aircraft response and the elevon commands.

Figure D.8: FLT29: The long-term aircraft response after the fault injection at 1478 s.

aircraft responses over two minutes after the fault. The new trim airspeed (Figure D.8a) is around 15.2 $\mathrm{m\,s^{-1}}$, which is a constraint imposed by the failed right elevon. Since the airspeed is slightly below its reference value, the total energy controller compensates by increasing the altitude by about 1 m above its reference value. The steady-state errors in the airspeed and the altitude offset each other because $\bar{K}_{TECS}$ tracks the mixed energy, as shown in Figure D.9. The phugoid mode is visible in the airspeed and the altitude, with a time period of about 10 s (Figure D.8a). In particular, the airspeed varies within ±1.5 $\mathrm{m\,s^{-1}}$ of 15.2 $\mathrm{m\,s^{-1}}$, the altitude varies within ±7 m of 320 m, and the pitch attitude varies within ±7.2° of 7.8°. The RMS tracking error in the mixed energy is 21.4 J, which is less than 1% of the reference value. Since the aircraft is rolled leftward, the left elevon is deflected slightly more (mean value of −6.7° in Figure D.8b) than the right elevon. In addition, the RMS tracking errors in the roll and the course angles (Figure D.8b) are 2.8° and 9.55°, respectively.

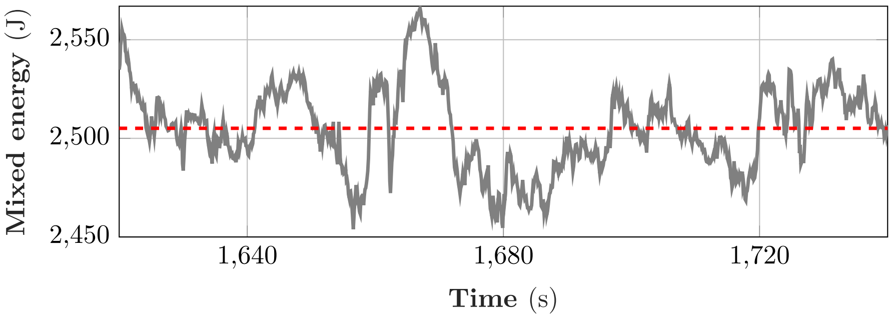


Figure D.9: FLT29: The total energy controller tracks the mixed energy after the fault.

D.3 FLT30: Flight Using Design A

FLT30 on 2017-09-20 validates Design A of the single surface controller and the fault-tolerant total energy controller. However, unlike FLT28, an autoland is not attempted. Shortly after the manual takeoff, the pilot takes the Vireo up to 225 ft AGL and engages the autopilot at 1020 s. Figure D.10 shows the ground track of the Vireo from the autopilot engagement (∗) until the pilot takes manual control (○). The ground track is plotted in a local North-East reference frame and annotated with various flight events.

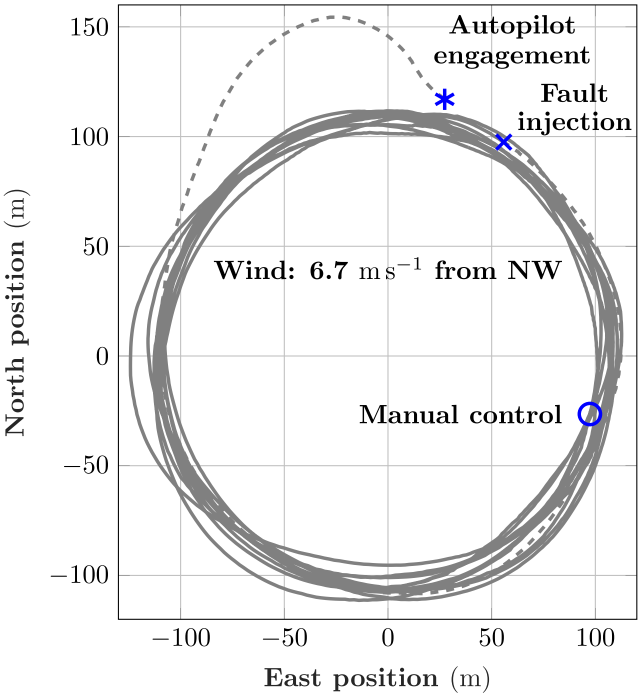
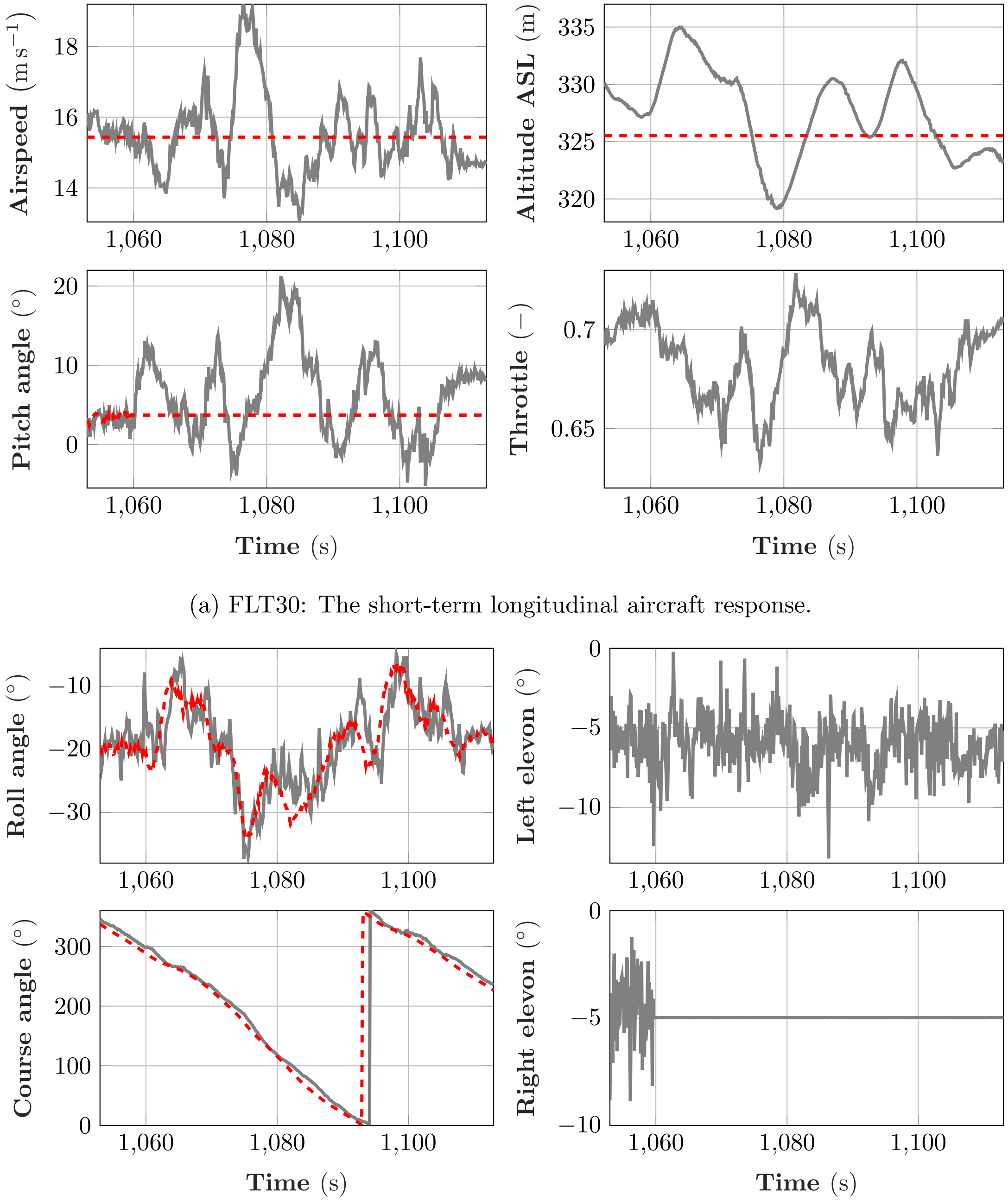


Figure D.10: FLT30: The ground track of the Vireo indicating different flight events. The solid line indicates the times when the single surface controller is active.

The autopilot engagement ($*$) activates the nominal controller, which takes the Vireo into a left banked circle hold at an altitude of 225 ft AGL, an airspeed of 15.4 $\mathrm{m\,s^{-1}}$, and a turn radius of 100 m. The center of the circle is at the origin of the North-East reference frame (Figure D.10) and the Vireo traces a counterclockwise circle. This initial phase, wherein the nominal controller is active, denoted by the dashed line starting at $*$ in Figure D.10. As with the other flights, a counterclockwise circle is chosen because the right elevon is faulted. At 1060 s, the UAS operator injects a $-5°$ stuck fault in the right elevon ($\times$). The flight computer sends a constant PWM comand to the right elevon to simulate the stuck fault. Since the demonstration excludes fault diagnosis, the fault-tolerant controller is activated immediately upon the fault injection. The ground track following the fault injection ($\times$) is denoted using the solid line. For the next 7.7 min, the Vireo continues to track the circle. The cross-track error standard deviations before and after the fault injection are 2.3 m and 4.3 m, respectively. Unlike the previous flights, an autoland is not attempted. Thus, the pilot takes manual control ($\circ$) at 1521 s and lands the aircraft. Next, it is instructive to evaluate the *short-term* and the *long-term* aircraft responses after the fault injection.

Figure D.11 shows the short-term longitudinal and lateral-directional aircraft responses as well as the control commands after the fault injection at 1060 s. The longitudinal motion variables (Figure D.11a) are close to their respective reference commands and the mean deflection of the right elevon (Figure D.11b) is approximately $-4.5°$ before the fault. Since this is close to the fault magnitude ($-5°$), large variations are not seen in the airspeed and the altitude immediately after the fault injection. However, about 15 s after the fault injection, the airspeed and the altitude start to exhibit higher variability. In addition, the fault causes the aircraft to momentarily roll rightwards (Figure D.11b). However, this is soon corrected by the single surface controller. Finally, since the fault position of the right elevon ($-5°$) is very close to its mean value before the fault ($-4.5°$), there is very little change in the trim airspeed.

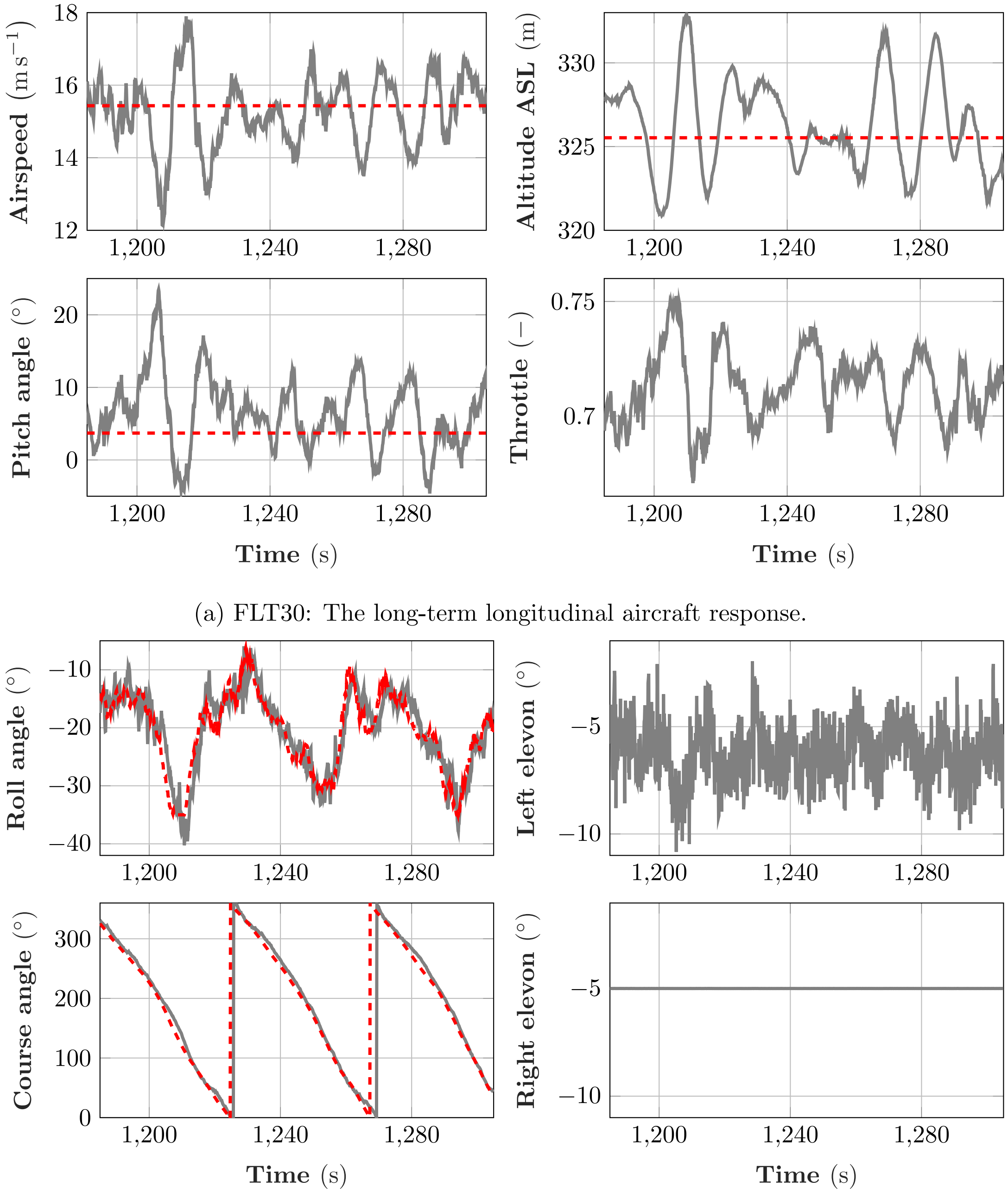
Figure D.12 follows the same layout as Figure D.11, except that it shows the long-term aircraft responses over two minutes after the fault. The new trim airspeed of 15.3 $\mathrm{m\,s^{-1}}$ (Figure D.12a) is very close to the nominal trim airspeed. Consequently, the mean altitude (326.3 m) is very close to its reference value (325.5 m). The phugoid mode is visible in the airspeed and the altitude, with a time constant of 10 s (Figure D.12a). In particular, the airspeed varies within ± 2 $\mathrm{m\,s^{-1}}$ of 15.3 $\mathrm{m\,s^{-1}}$, the altitude varies within ± 7 m of 326.3 m, and the pitch attitude varies within $\pm 8°$ of $8°$. The RMS tracking error in the mixed energy (Figure D.13) is 23.5 J, which is around 1% of the reference value. Since the aircraft is roll leftward, the left elevon is deflected slightly more (mean value of $-6.4°$ in Figure D.12b) than the right elevon. In addition, the RMS tracking errors in the roll and the course angles (Figure D.12b) are $3°$ and $7.8°$, respectively.



(a) FLT30: The short-term longitudinal aircraft response.

(b) FLT30: The short-term lateral-directional aircraft response and the elevon commands.

Figure D.11: FLT30: The short-term aircraft responses after the fault injection at 1060 s.



(a) FLT30: The long-term longitudinal aircraft response.

(b) FLT30: The long-term lateral-directional aircraft response and the elevon commands.

Figure D.12: FLT30: The long-term aircraft response after the fault injection at 1060 s.

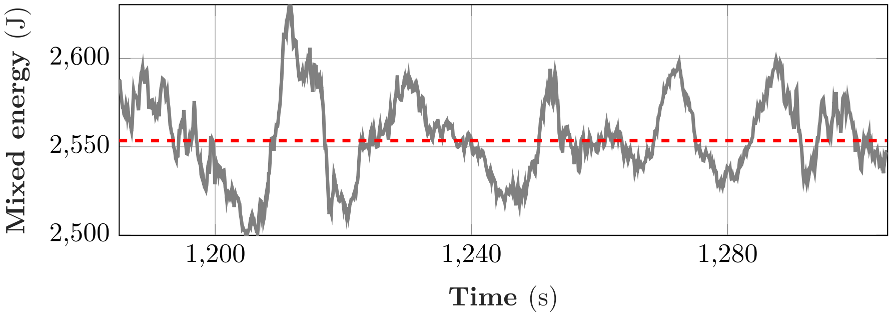


Figure D.13: FLT30: The total energy controller tracks the mixed energy after the fault.

D.4 FLT34: Autoland Using Design B

An autoland demonstration using Design B of the single surface controller and the fault-tolerant total energy controller is conducted during FLT34 on 2017-10-05. Shortly after the manual takeoff, the pilot takes the Vireo up to 250 ft AGL and engages the autopilot at 946 s. Figure D.14 shows the ground track of the Vireo from the autopilot engagement ($*$) until the landing on the virtual runway ($\circ$). The ground track is plotted in a local North-East reference frame, whose origin coincides with the target landing spot chosen by the UAS operator prior to the start of the flight. The various events occurring in the flight are annotated on the ground track and are explained one-by-one.

The autopilot engagement ($*$) activates the nominal controller, which takes the Vireo into a left banked circle hold at an altitude of 250 ft AGL, an airspeed of 15.4 $\mathrm{m\,s^{-1}}$, and a turn radius of 150 m. The center of the circle $(1.92, -95.6)$ m is automatically set by the flight computer to the point where the airplane climbs past the altitude of 150 ft AGL. The Vireo traces a counterclockwise circle when viewed in the North-East plane. This initial phase, wherein the nominal controller is active, is denoted by the dashed line starting at $*$ in Figure D.14. As with the other flights, a counterclockwise circle is chosen because the right elevon is faulted. At 1010 s, the UAS operator injects a $-5°$ stuck fault in the right elevon ($\times$). The flight computer sends a constant PWM command to the right elevon to simulate the stuck fault. Since the demonstration excludes fault diagnosis, the fault-tolerant controller is activated immediately upon the fault injection. The ground track following the fault injection ($\times$) is denoted using the solid line. For

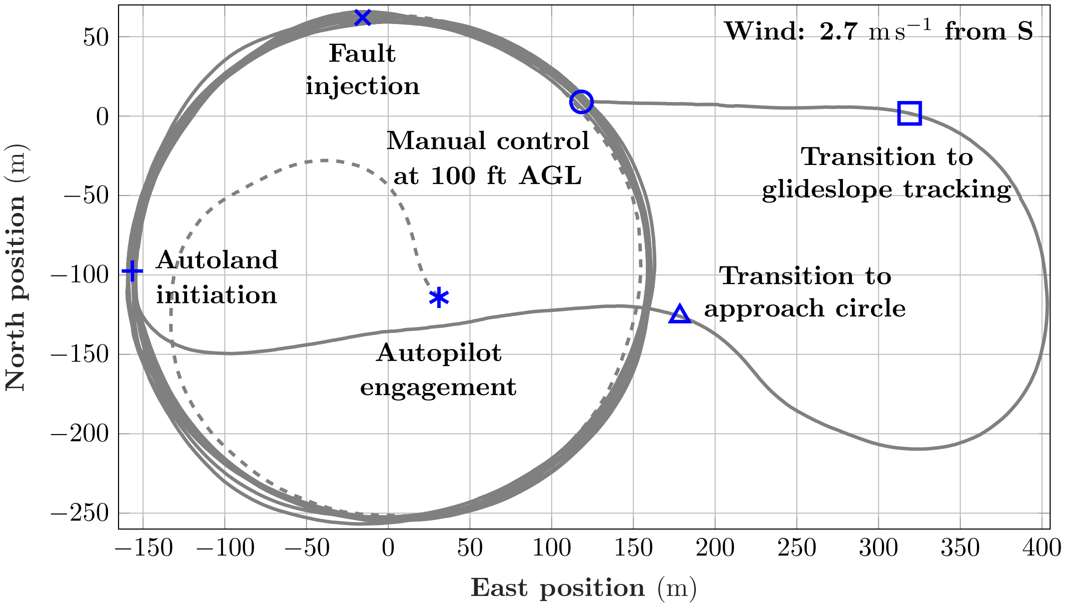


Figure D.14: FLT34: The ground track of the Vireo indicating different flight events. The solid line indicates the times when the single surface controller is active.

the next 7 min, the Vireo continues to track the circle. The cross-track error standard deviations before and after the fault injection are 2.3 m and 2.6 m, respectively.

At 1431 s , the UAS operator initiates the autoland sequence (+) causing the Vireo to peel away from the circle hold and fly towards the approach circle. The prevailing winds on that day are 2.7 m s^{-1} from the Sourth. The UAS operator commands a Westerly landing to remain consistent with the previous flights. The Vireo enters the approach circle (△) and traverses about three-quarters of the circle. Once the Vireo is heading due West, it exits the circle and transitions to glideslope tracking (□). The remainder of the autoland is best explained by the next figure.

Figure D.15 shows the terminal flight path of the Vireo as seen by an observer standing on the ground and facing due North. The target approach speed is 15.4 m s^{-1} and the target flight path is a 6° glideslope that intersects the virtual runway at an East position of zero. As seen in Figure D.15, although the tracking is good in the beginning (□), the Vireo soon drops below the glideslope. The pilot takes manual control (○) when the Vireo is 3.5 m above the virtual runway and 118 m short of the intended touchdown point. The pilot performs a go-around, which terminates in a gentle landing. This final phase of manual control is not shown to avoid clutter. Next, it is instructive to evaluate the *short-term* and the *long-term* aircraft responses after the fault-injection.

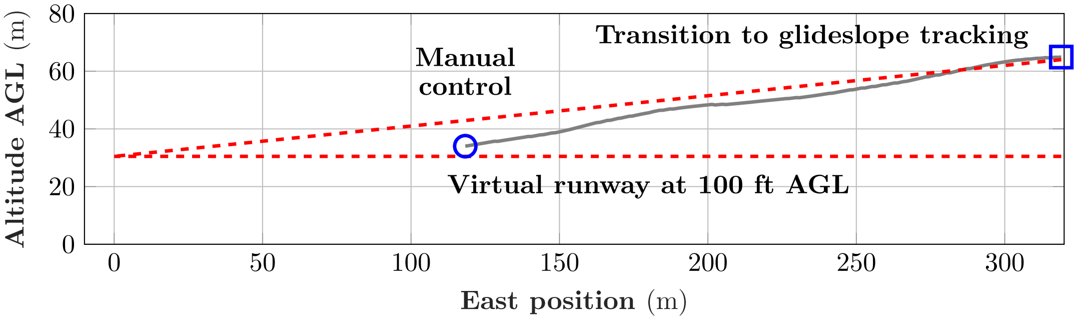
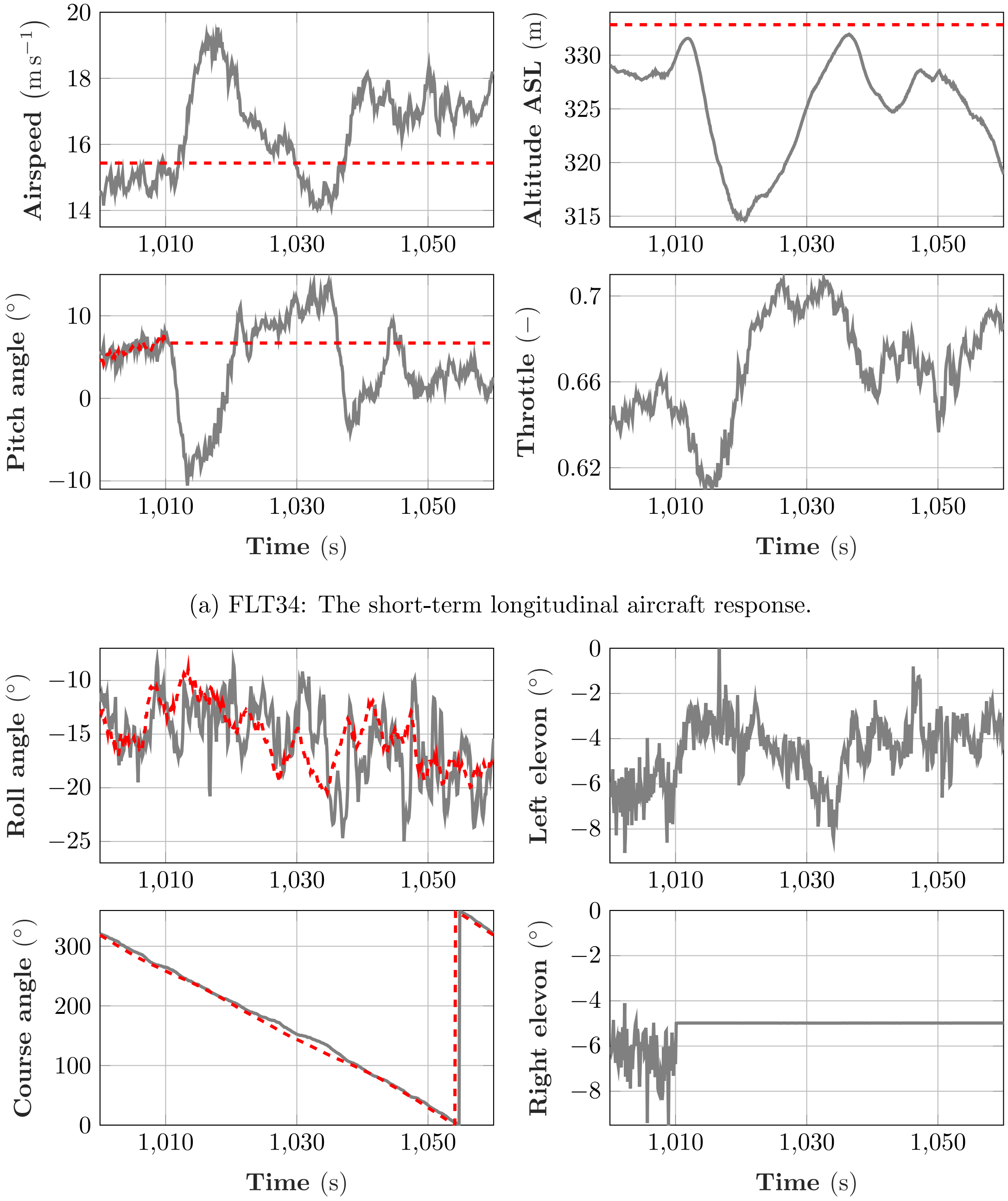


Figure D.15: FLT34: The Vireo tracks a glideslope during the final approach. The pilot takes manual control of the aircraft once it reaches the virtual runway at 100 ft AGL.

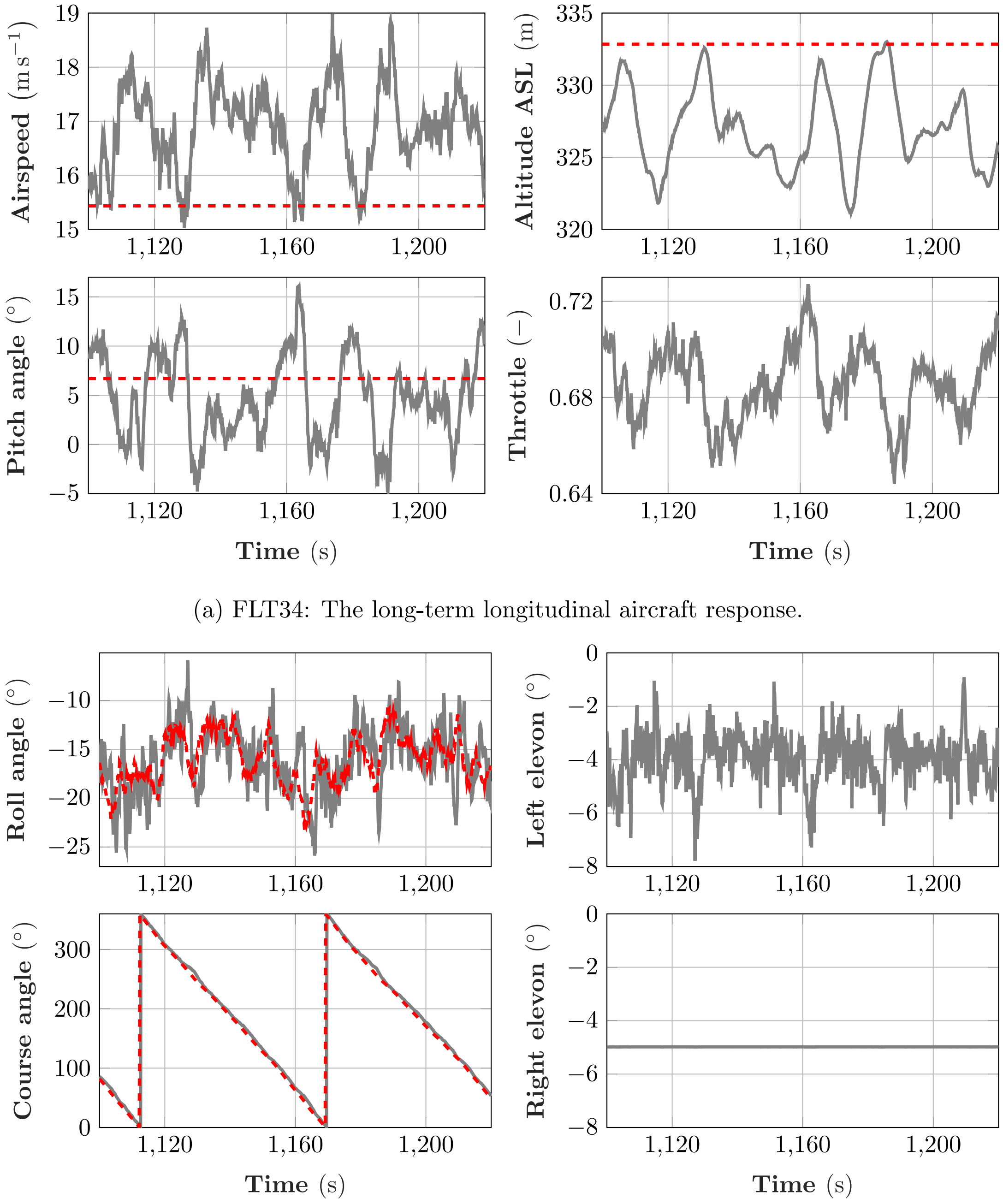
Figure D.16 shows the shot-term longitudinal and lateral-directional aircraft responses as well as the control commands after the fault injection at 1010 s. The longitudinal motion variables (Figure D.16a) are close to their respective reference commands and the trim deflection of the right elevon (Figure D.16b) is approximately $-6°$ before the fault. This is larger than the trim deflections in FLT27 and FLT28 because the Vireo carries an on-board camera during FLT34, which increases its all-up weight by 45 g. Since the fault ($-5°$) is smaller in magnitude than the nominal trim deflection, the aircraft pitches nose down to about $-10°$ as seen in Figure D.16a. This in turn causes the airspeed to rise to about 19 $\mathrm{m\,s^{-1}}$ and the altitude to drop by about 18 m. The fault-tolerant total energy controller responds by decreasing the throttle to about 0.62. Around 20 s after the fault, the airspeed and the pitch angle return closer to their respective reference commands, but exhibit higher variability than before. The altitude takes an additional 20 s to stabilize. As for the lateral-directional motion variables (Figure D.16b), the fault causes the Vireo to momentarily roll rightward, as seen in the upward spike in the roll angle at 1011 s. This is quickly corrected by the single surface controller. Finally, since the failed position of the right elevon ($-5°$) is slightly smaller in magnitude than its mean value before the fault ($-6°$), the trim airspeed increases.



(a) FLT34: The short-term longitudinal aircraft response.

(b) FLT34: The short-term lateral-directional aircraft response and the elevon commands.

Figure D.16: FLT34: The short-term aircraft response after the fault injection at 1010 s.



(a) FLT34: The long-term longitudinal aircraft response.

(b) FLT34: The long-term lateral-directional aircraft response and the elevon commands.

Figure D.17: FLT34: The long-term aircraft response after the fault injection at 1010 s.

Figure D.17 follows the same layout as Figure D.16, except that it shows the long-term aircraft response over two minutes after the fault. The new trim airspeed (Figure D.17a) is around 17 $\mathrm{m\,s^{-1}}$, which is a constraint imposed by the failed right elevon. Since the airspeed is above its reference value, the total energy controller compensates by decreasing the altitude by about 5.8 m below its reference value. The steady-state errors in the airspeed and the altitude offset each other because $\bar{K}_{TECS}$ tracks the mixed energy, as shown in Figure D.18. The phugoid mode is visible in the airspeed and the altitude, with a time period of about 10 s (Figure D.17a). In particular, the airspeed varies within ±2 $\mathrm{m\,s^{-1}}$ of 17 $\mathrm{m\,s^{-1}}$, the altitude varies within ±5 m of 327 m, and the pitch attitude varies within ±8.6° of 6.4°. The RMS tracking error in the mixed energy is 22.7 J, which is less than 1% of the reference value. The trim value of the left elevon after the fault is around −4°. In addition, the RMS tracking errors in the roll and the course angles (Figure D.17b) are 3° and 5.4°, respectively.

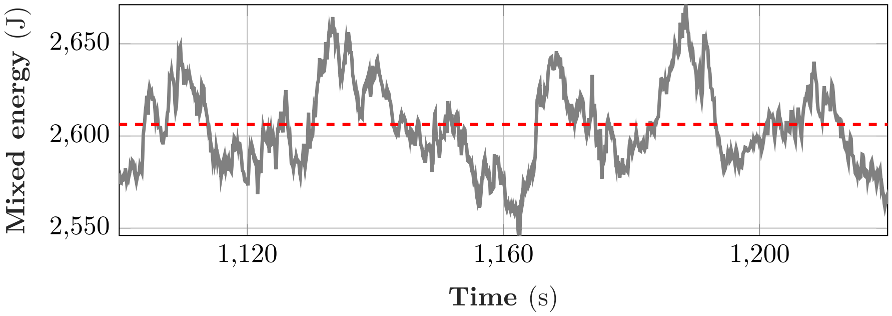


Figure D.18: FLT34: The total-energy controller tracks the mixed energy after the fault.