

Test Platforms for Model-Based Flight Research

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

Andrei Dorobantu

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

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

September 2013

© Andrei Dorobantu 2013

Acknowledgments

I want to thank all those around me who supported me through the years. It has been a long, crazy journey. I would never have been able to get any research together without your constant encouragement and cheer. For that I want to say thanks to my parents, my brother, Joe, and everyone else in the lab for dealing with me.

I also want to thank my advisers Gary Balas and Pete Seiler for their guidance. None of this would have been possible without your pushing and helping me forward, especially in the beginning. It is difficult for me to describe the scope of the influence you have had on my outlook. Perhaps we should propose a new metric for this. And thanks for reading every paper for the n^{th} time – I know it was thrilling!

Finally, I want to thank my co-authors who helped me with my research and journal papers. The system identification results could not have been attained without input from Bérénice Mettler. Similarly, insight provided by Tryphon Georgiou on the gap metric was essential in formulating the model validation result. It was truly an honor to work with and learn from you.

Dedication

To My Family

Abstract

Demonstrating the reliability of flight control algorithms is critical to integrating unmanned aircraft systems into the civilian airspace. For many potential applications, design and certification of these algorithms will rely heavily on mathematical models of the aircraft dynamics. Therefore, the aerospace community must develop flight test platforms to support the advancement of model-based techniques. The University of Minnesota has developed a test platform dedicated to model-based flight research for unmanned aircraft systems. This thesis provides an overview of the test platform and its research activities in the areas of system identification, model validation, and closed-loop control for small unmanned aircraft.

Contents

List of Tables	vii
List of Figures	vii
Nomenclature	x
Chapter 1 Introduction	1
Chapter 2 UAS Test Platform	4
2.1 Flight Test System	5
2.1.1 Vehicles	5
2.1.2 Avionics and Sensors	7
2.1.3 Flight Operations	11
2.2 Simulation Environment	12
2.2.1 Nonlinear Aircraft Simulation	13
2.2.2 Software-in-the-loop	14
2.2.3 Hardware-in-the-loop	14
2.3 Flight Software	15
2.3.1 Real-time Software	15

2.3.2	Navigation State Estimation	16
2.3.3	Version Control and Documentation	19
Chapter 3	System Identification	20
3.1	Fixed-Wing Aircraft Dynamics	22
3.1.1	Longitudinal Dynamics	23
3.1.2	Lateral/Directional Dynamics	24
3.2	Preliminary Analysis	25
3.2.1	Longitudinal Dynamics	27
3.2.2	Lateral/Directional Dynamics	28
3.3	Design of Flight Experiments	29
3.3.1	Frequency Sweep Design	30
3.4	Frequency Domain System Identification	32
3.4.1	Extracting Frequency Responses	33
3.4.2	Parametric Identification	35
3.5	Identification Results	36
3.5.1	Longitudinal Dynamics	37
3.5.2	Lateral/Directional Dynamics	39
3.6	Sensitivity and Residual Analysis	43
3.6.1	Sensitivity to Parameter Variation	43
3.6.2	Residual Disturbance Modeling	47
Chapter 4	Model Validation	50
4.1	Theoretical Background	51

4.2	A Framework for Model Validation	57
4.2.1	A Simple Example	59
4.3	Flight Data Results	61
Chapter 5	Baseline Control Design	67
5.1	Inner-Loop Controller	67
5.2	Outer-Loop Controller	70
5.3	Waypoint Guidance	72
Chapter 6	Conclusion	75
Bibliography		77

List of Tables

2.1	Key Physical Properties of Ultra Stick Vehicles	6
2.2	Summary of Core Avionics and Sensors	10
2.3	Summary of additional sensor equipment	10
3.1	Moments of inertia for Thor.	26
3.2	Estimated modes of the longitudinal dynamics.	27
3.3	Estimated modes of the lateral/directional dynamics.	28
3.4	Identified longitudinal dynamics.	39
3.5	Identified lateral/directional dynamics.	42
3.6	Cramér-Rao bounds for the identified aircraft model.	46
4.1	Validation results for raw and averaged flight data.	64

List of Figures

1.1	University of Minnesota UAS test vehicle.	2
2.1	University of Minnesota UAS test platform.	4
2.2	Three versions of flight test vehicles.	6
2.3	Core avionics and sensor array.	8
2.4	Integrated flight control system (FCS).	9
2.5	Airfield and visual range.	11
2.6	Three levels of simulation environments.	12
2.7	Real-time software schedule and order diagram of code modules.	15
2.8	Block diagram of INS/GPS integration algorithm.	17
2.9	Navigation filter solution.	18
3.1	University of Minnesota Ultra Stick 25e code name Thor.	21
3.2	Baseline model frequency response with range for system identification.	31
3.3	Sample aileron frequency sweep input signal and response.	32
3.4	Diagram of a two-input, single-output system.	33
3.5	Longitudinal dynamics identification for elevator input.	37
3.6	Lateral/directional dynamics identification for roll rate output.	40
3.7	Lateral/directional dynamics identification for yaw rate output.	41

3.8	Lateral/directional dynamics identification for y-axis acceleration output.	42
3.9	Selected uncertain frequency responses.	45
3.10	System diagram for output disturbance modeling.	48
3.11	Output disturbance models for angular rate measurements.	49
4.1	Comparison of the model $P_1(s)$ and the true aircraft dynamics $P_2(s)$.	60
4.2	Guaranteed robustness margins for control.	61
4.3	Experimental frequency responses and linear model.	62
4.4	Perturbations and uncertainty disks.	64
4.5	Time domain simulation of uncertainty model versus flight data.	66
5.1	Longitudinal dynamics control architecture.	68
5.2	Lateral/directional dynamics control architecture.	68
5.3	Inner-loop control flight test vs. SIL simulation.	69
5.4	Aircraft outer-loop control architecture.	70
5.5	Validation of outer-loop flight management system.	71
5.6	Satellite view of flight management system validation results.	72
5.7	GPS waypoint guidance flight test result.	73

Nomenclature

Abbreviations and Acronyms

UAS	Unmanned Aircraft System
UMN	University of Minnesota
IMU	Inertial Measurement Unit
SVN	Subversion
FCS	Flight Control System
DOF	Degree of Freedom
SIL	Software-in-the-loop
HIL	Hardware-in-the-loop
RTOS	Real-time Operating System
INS	Inertial Navigation System
SNR	Signal-to-Noise Ratio
CR	Cramer Rao
LTI	Linear Time-Invariant
SISO	Single-Input Single-Output

List of Symbols

m	Mass, kg
I	Moment of inertia, $kg\ m^2$
g	Gravity, m/s^2
ϕ, θ, ψ	Attitude angles, rad
u, v, w	Body-axis velocities, m/s
p, q, r	Body-axis angular rates, rad/s
a_x, a_y, a_z	Body-axis measured accelerations, m/s^2
δ_{elev}	Elevator deflection, rad
δ_{ail}	Aileron deflection, rad
δ_{rud}	Rudder deflection, rad
X, Y, Z	Aerodynamic forces, N
L, M, N	Aerodynamic moments, $N\ m$
T	Thrust force, N

Chapter 1

Introduction

The ascent of unmanned aircraft systems (UASs) has enabled the possibility for novel and exciting applications across many industries. For the first time in 2015, these aircraft will be allowed to fly in the civilian airspace [1]. Accordingly, regulatory agencies have begun to award type certificates for non-military UASs [2]. Demonstrating the reliability and integrity of flight control algorithms is a critical step to obtain such a certificate. This process has traditionally been very long and expensive for the aircraft designer, often relying on extensive ground and flight test campaigns. In many potential UAS applications, the design and certification of flight control algorithms will rely heavily on mathematical models of the aircraft dynamics. A model-based approach is necessary in order to reduce the development cost, which is key to the UAS value proposition.

Modern aircraft like the F-35 have already taken advantage of model-based design, leading the F-35 to become the most complex and sophisticated aircraft ever built. At the same time, test pilots on the program attribute many of the costly delays and budget overruns on models that failed to recognize design problems as expected [3]. One of the major lessons learned from the F-35 program is that better handling of models and their accuracy is required to efficiently rely on model-based design. In response to this, the aerospace community must develop more effective tools for modeling and validation.

Test platforms dedicated to model-based flight research are critical to the continued development and integration of UASs in the civilian airspace. They serve as a nec-

essary proving ground for novel solutions to all aspects of the technology, including flight control. In addition, they are ideally suited to study the underlying models that support model-based design and certification. Using on these platforms, more effective analysis and validation techniques can be developed to address the shortcomings of existing approaches. The University of Minnesota (UMN) has developed a UAS test platform dedicated to model-based flight research [4]. One of the test aircraft is shown in Figure 1.1. Flight tests are routinely carried out for research experiments in the areas of guidance, navigation, control, and fault detection. A vital component in this activity is having access to accurate models of the aircraft dynamics. Therefore, a complementary research direction is centered on identifying and validating models for small unmanned aircraft.



Figure 1.1: University of Minnesota UAS test vehicle.

This thesis describes the development of a UAS test platform dedicated to model-based flight research. It is largely based on three major publications in the areas of flight test platform development [5], system identification [6], and model validation [7]. A major emphasis is placed on modeling and validation approaches for small aircraft. The primary contribution of this work is twofold. First, new and better techniques are proposed where the current approach is not effective or poorly suited for model-based design. An example of this is the model validation framework described in Chapter 4. Therein, a new framework is proposed that brings more analytic rigor to the model

validation process. Second, engineering insight is provided on the development process and challenges that pertain to small UASs and flight control. The application of system identification techniques and flight test design in Chapter 3 belong to this category. In this work, the impact of size, weight, and cost restrictions on the ability to determine accurate models of the aircraft dynamics is examined. Both contributions are useful to the aerospace community because they fill engineering knowledge gaps uncovered by the ascent of UASs.

The chapters in this thesis follow the engineering tasks associated with developing flight control algorithms for UASs. Chapter 2 introduces the UMN UAS test platform. A brief overview is provided on the flight test vehicles, the integrated hardware and software suite, and the simulation environment. Chapter 3 tailors existing system identification techniques for application to small, low-cost unmanned aircraft. The approach is focused on engineering insight to support the identification of accurate models using limited sensing. Chapter 4 proposes a new model validation framework based on the gap metric. The proposed approach is an analytically rigorous way to validate models of aircraft dynamics using flight data. Finally, Chapter 5 describes the design of a baseline controller and its validation through flight testing.

Chapter 2

UAS Test Platform

The University of Minnesota has developed a UAS test platform dedicated to model-based flight research. The test platform has advanced capabilities to support research in the areas of guidance, navigation, control, and fault detection. This chapter describes key components of the infrastructure in order to provide context on the functions of the platform. These include a flight test system, a high-fidelity nonlinear simulation, software- and hardware-in-the-loop simulations, and a real-time flight software suite. The test platform relies on a seamless integration of its components to carry out model-based flight research. Figure 2.1 shows one of the primary test vehicles on the left, and the research laboratory at the Twin Cities campus on the right.



(a) Flight test vehicle on approach



(b) Research laboratory

Figure 2.1: University of Minnesota UAS test platform.

The UMN UAS test platform is managed by an academic research group founded under a set of guiding principles. These principles include supporting open-source development, providing freely available flight data, and enabling resource and information sharing. Consequently, all development is publicly available online [4]: www.uav.aem.umn.edu. The website includes wiki-style documentation, the full sub-version repository (SVN) for access to the simulation environments and the real-time flight software suite, and a database of flight test data.

2.1 Flight Test System

The flight test system is the flagship component of the UMN UAS test platform. It is comprised of three integrated sub-components: vehicles, avionics and sensors, and flight operations. A fleet of test vehicles supports research experiments with respect to a variety of airframe size and payload requirements. An avionics and sensor array supports fundamental flight and communication needs, as well as specific research experiments. Finally, a set of flight operations ensures the safety of the entire system.

2.1.1 Vehicles

The current fleet includes three versions of conventional fixed-wing aircraft that belong to the Ultra Stick family. The Ultra Stick family is a commercially available group of radio-controlled aircraft. Each airframe is modified to fit and carry the necessary avionics and sensors. The Ultra Stick 120 is the largest and heaviest airframe, with a 1.92 m wing span and 7.4 kg mass. It is capable of carrying the most payload and is equipped with the largest array of sensors. The Ultra Stick 25e is a 66% scale version of the Ultra Stick 120, with a 1.27 m wing span and 1.9 kg mass. It serves as the primary flight test vehicle due to its convenient size and is equipped with a core avionics and sensor array. Subsequent chapters of this thesis focus on the flight dynamics of the Ultra Stick 25e. The Ultra Stick Mini is a 52% scale version of the Ultra Stick 120, with a 0.98 m wing span. This aircraft is used as a wind tunnel model and is not equipped with any flight avionics or sensors. Figure 2.2 shows an example of each airframe version.

All three vehicles have conventional fixed-wing airframes with aileron, rudder, elevator, and flap control surfaces. Each control surface is actuated using an electric servo, with a maximum deflection of 25 degrees in each direction. The propulsion systems



(a) Ultra Stick Mini

(b) Ultra Stick 25e

(c) Ultra Stick 120

Figure 2.2: Three versions of flight test vehicles.

consist of electric motors (with varying power depending on the airframe size) that drive fixed-pitch propellers. The aircraft systems are battery powered, designed to allow for approximately 30 minutes of power on a single charge. Some of the key physical properties of the three Ultra Stick aircraft are given in Table 2.1.

Table 2.1: Key Physical Properties of Ultra Stick Vehicles

Parameter	Mini	25e	120
Mass	0.62 kg	1.90 kg	7.41 kg
Wing Span	0.98 m	1.27 m	1.92 m
Wing Chord	0.21 m	0.30 m	0.43 m
Wing Area	0.21 m ²	0.32 m ²	0.77 m ²
Length	0.87 m	1.05 m	1.32 m
Endurance	10 – 15 min	15 – 20 min	15 – 20 min
Cruise Speed	12 m/s	17 m/s	25 m/s
Cost	\$120	\$170	N/A

Thrust for the vehicles is generated by electric outrunner brushless DC motors, which require electronic speed controllers. Ultra Stick 120 vehicles are powered by Actro 40-4 motors along with Castle Creations ICE2 HV80 speed controllers. These motors require two 5S 5000 mAh lithium polymer (LiPo) batteries. The avionics and servos are powered by a single 4S 1650 mAh LiPo battery. Ultra Stick 25e vehicles are powered by Eflite Power 25 motors along with Castle Creations ICE LITE 50 speed controllers. These motors require a single 3S 3000 mAh LiPo battery, which also powers the servos. The avionics are powered by a single 3S 800 mAh LiPo battery.

The Ultra Stick 120 aircraft was initially used as a low-cost flight test platform at

NASA Langley Research Center [8]. Aerodynamic modeling efforts have included extensive static wind tunnel tests, which were later complemented with dynamic wind tunnel tests [9, 10]. The aerodynamic model for the Ultra Stick 120 is a nonlinear look-up table that includes effects due to the basic airframe, control surfaces, thrust, and angular rates. This high-fidelity model was made publicly available by NASA. Unfortunately, the Ultra Stick 120 model airframe is currently out of production. To ensure the continuity of the Ultra Stick 120 as a flight test vehicle, a stock of three spare airframes has been acquired.

The Ultra Stick 25e was co-developed as a low-cost flight test platform by the University of Minnesota [11, 12] along with researchers at the Budapest University of Technology and Economics in Hungary. Over time, the needs of the two groups have evolved, and, hence, the vehicles are currently equipped with different avionics and sensors. However, the similarity in airframes allows for cooperation in critical research areas, such as control and navigation algorithms. The aerodynamic model for this aircraft was derived using frequency domain system identification techniques based on flight test data [6]. Chapter 3 is focused on this work and provides details on the technical approach.

The Ultra Stick Mini is used primarily as a wind tunnel model. It serves as an educational tool for undergraduate courses and laboratories for the Department of Aerospace Engineering & Mechanics. For example, undergraduates use the airframe mounted on a sting in a wind tunnel to estimate basic aerodynamic coefficients.

2.1.2 Avionics and Sensors

The architecture of the core avionics and sensor array is shown in Figure 2.3. This hardware combination is installed onboard each Ultra Stick 120 and 25e airframe and represents the minimum requirement for research experiment flight tests. Some individual airframes have additional sensors to support specific experimental functions. These specific sensor outfits will be highlighted following a description of the core avionics and sensor array.

At the center of the avionics and sensor array is the flight computer, a phyCore MPC5200B 32-bit PowerPC microcontroller [13]. It has a clock frequency of 400 MHz, 760 MIPS of processing power, and performs floating point computation. The flight computer utilizes a real-time operating system called eCos [14], and the flight

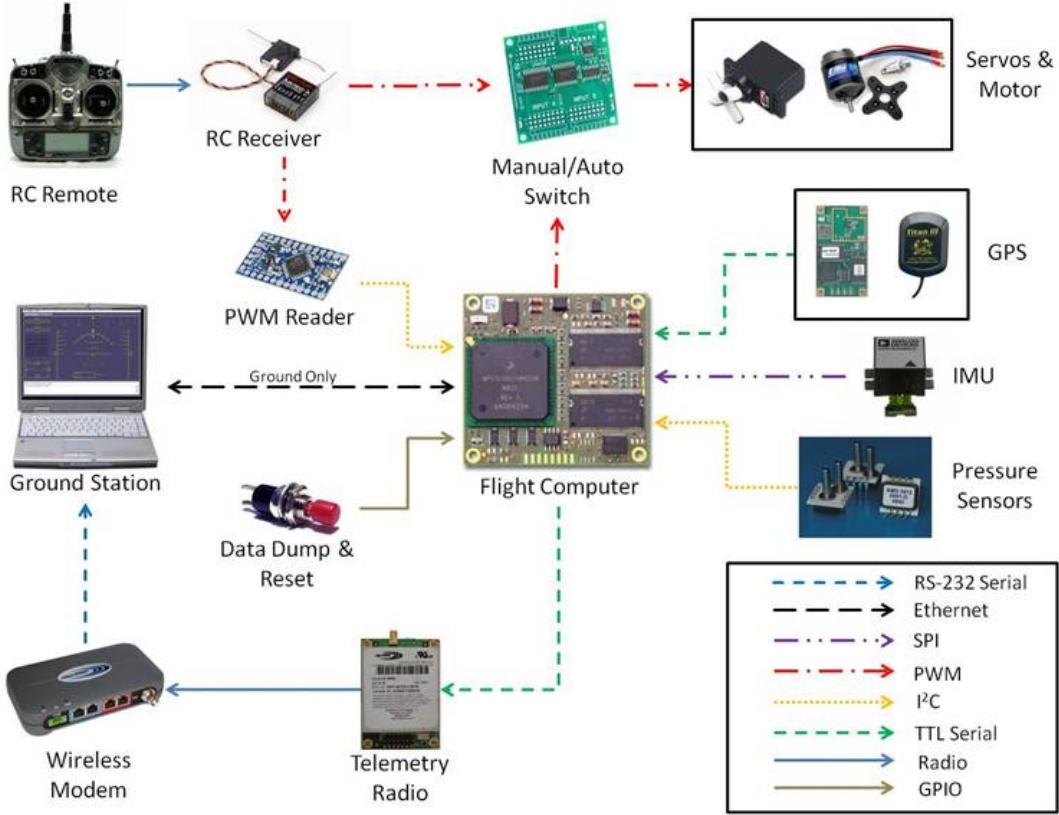


Figure 2.3: Core avionics and sensor array.

software is written in C. The flight software is modularized with standard interfaces, allowing different modules (e.g. different control or fault detection algorithms) to be easily interchanged. More details on the software architecture are presented in Section 2.3. The current software utilizes runs at a framerate of 50 Hz.

The MPC5200B has a wide range of input-output capabilities. It supports communication with external devices via TTL and RS232 serial, SPI, I²C, and Ethernet. Communication with servo actuators is handled with PWM. Flight data is recorded at 50 Hz and stored in the 64 MB SRAM available onboard. The flight data is downloaded after each flight via Ethernet connection to a ground station laptop. The Ethernet connection is also used to load flight software onto the flight computer. The flight computer is mounted on an interface board, which is a custom design and handles power and the communication interface with external devices. Digital production and fabrication documents for the custom interface board are available on the research group website [4].

A failsafe board [15] is used to switch control of the aircraft between manual mode (human pilot stick-to-surface control) and flight computer automatic mode. In both modes, pilot commands are recorded and provided to the flight computer. This enables the option for piloted closed-loop control or signal augmentation experiments. Telemetry is sent to a ground station laptop through a wireless radio at 10 Hz [16]. The transmitted data is visualized on a custom developed synthetic heads-up display. The display provides real-time information about attitude, altitude, airspeed, and GPS performance.

In addition to the flight computer and mode switch, each flight test vehicle is equipped with a core set of onboard sensors. Measurements of static and dynamic air pressure from a Pitot probe are used to estimate airspeed and altitude. Pressure transducers [17] communicate with the flight computer over I²C. Angular rates and translational accelerations are measured with an inertial measurement unit (IMU) [18], which communicates through SPI. This sensor is comprised of gyroscope and accelerometer triads. The IMU is aligned with the body axis of the vehicle and located near the center of gravity. A GPS receiver provides position and velocity information at 1 Hz and communicates over a TTL serial line [19, 20]. Table 2.2 summarizes the core avionics and sensor array. This array is assembled and integrated into a single flight control system (FCS) that is common to all test vehicles, shown in Figure 2.4. Once again, digital production and fabrication documents for the FCS are available online [4].

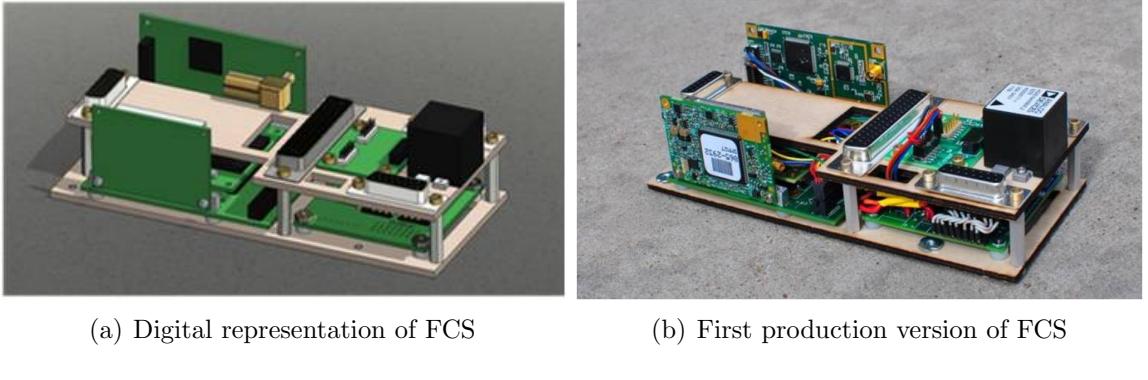


Figure 2.4: Integrated flight control system (FCS).

Several vehicles are equipped with additional sensors to enhance their research capabilities. An Ultra Stick 120 (code name UMN FASER), and an Ultra Stick 25e (code name Thor), are equipped with 5-hole Pitot probes [21]. These probes were

Table 2.2: Summary of Core Avionics and Sensors

Component	Module	Cost
Flight Computer	Phytec MPC5200B Microcontroller	\$250
Failsafe Switch	AcroName Robotics RxMux	\$300
Interface Board	AEM Custom Design	\$250
Receiver	Spektrum AR7010	\$90
Telemetry Radio	Free Wave MM2-T 900 MHz Modem	\$590
IMU	Analog Devices iSensor ADIS16405	\$860
GPS Receiver	Hemisphere GPS Crescent Board	\$300
GPS Antenna	GPS Outfitters Titan 3 Antenna	\$70
Pitot Probe	Eagletree	\$10
Pressure Trans.	AMSYS AMS 5812 (x2)	\$60
Total Cost		\$2,780

originally used as wind tunnel prototypes by Goodrich Corporation in Burnsville, MN (now UTC Aerospace), after which they were donated to the UMN UAS test platform. Each 5-hole Pitot probe takes 4 additional air pressure measurements along with the standard 2 (the name refers to 5 pressure taps on front of sensor). The additional measurements are used to estimate angle-of-attack and angle-of-sideslip. UMN FASER is also equipped with wingtip sensor booms that measure angle-of-attack and angle-of-sideslip directly [22]. Another Ultra Stick 120 (code name GPS FASER) is equipped with 2 additional GPS antenna/receiver systems. This aircraft is specialized for navigation and GPS research. Table 2.3 summarizes the additional sensor equipment currently available onboard each vehicle.

Table 2.3: Summary of additional sensor equipment

Vehicle Type	Code Name	Sensor
Ultra Stick 120	UMN FASER	5-hole Pitot Probe
Ultra Stick 120	UMN FASER	Wingtip Sensor Booms
Ultra Stick 120	GPS FASER	GPS Antenna/Receiver (2x)
Ultra Stick 25e	Thor	5-hole Pitot Probe

2.1.3 Flight Operations

Typical flight experiments are divided into three segments: take-off, research experiments, and landing. Each flight begins with a manual take-off by the pilot. For safety, winds below 10 mph with no gusts are required. Once airborne, the pilot flies the aircraft into a race track pattern with constant altitude (below 200 m) and obtains a steady trim. The race track pattern is generally used to maximize available straight and level flight time. Dimensions of the pattern are defined by line of sight requirements. In an emergency, the pilot must always be able to visually guide the aircraft back to safe operation. As a result of these safety constraints, the Ultra Stick 120 and 25e can only achieve about a 20 second maximum of straight and level flight.

Figure 2.5 shows a satellite view [23] of a hobby R/C airfield where the UMN UAS test platform performs research experiments. The runway is centered at the origin of the map, near the “Airfield” label, and oriented in a north/south direction. A flight test was conducted to determine the maximum line of sight range, and this data is shown in Figure 2.5. Direct GPS position measurements are shown along with the refined position estimate from a navigation filter. More details on the navigation algorithm are provided in Section 2.3.2. The range test indicates that the pilot can safely fly inside a semi-circle with approximate radius of 500 m. Safety rules prohibit flying behind (or over) the flight-line, hence the operating flight range is limited to a semi-circle. To ensure a continued safe operation, a collection of documents has been

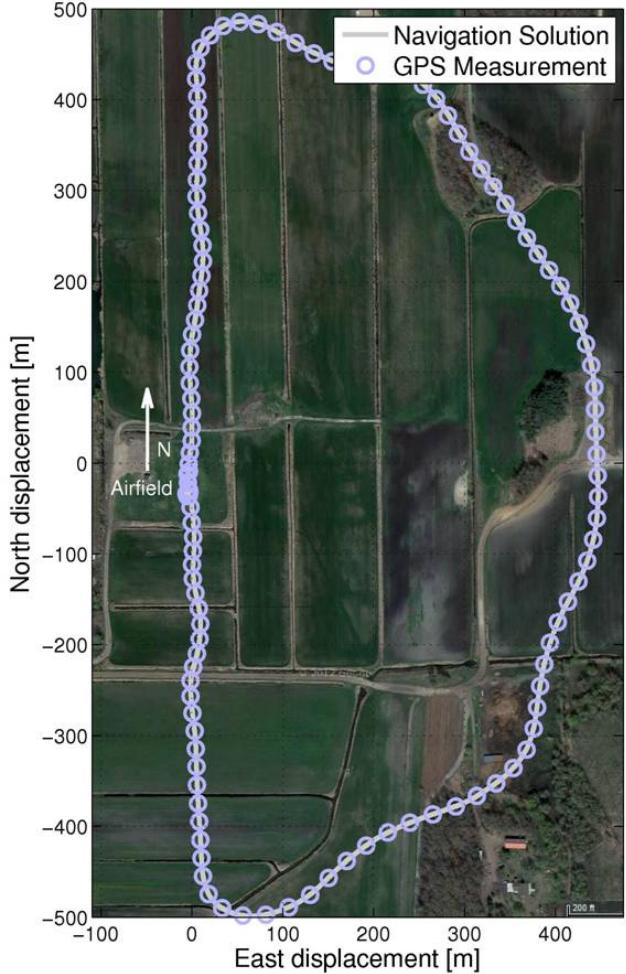


Figure 2.5: Airfield and visual range.

drafted to standardize procedures and maintenance plans.

2.2 Simulation Environment

A simulation environment provides an important complement to the flight test system. Simulation-based development and validation prior to flight testing reduces the total design cycle time for experimental research algorithms. The UMN UAS research group maintains three simulations, illustrated by the block diagram in Figure 2.6. A common Matlab/Simulink [24] implementation of the aircraft dynamics is shared between the three simulations. This shared implementation includes flight dynamics, actuator models, sensor models, and an environmental model. All experimental research algorithms must pass through a validation test in each simulation before consideration for flight testing.

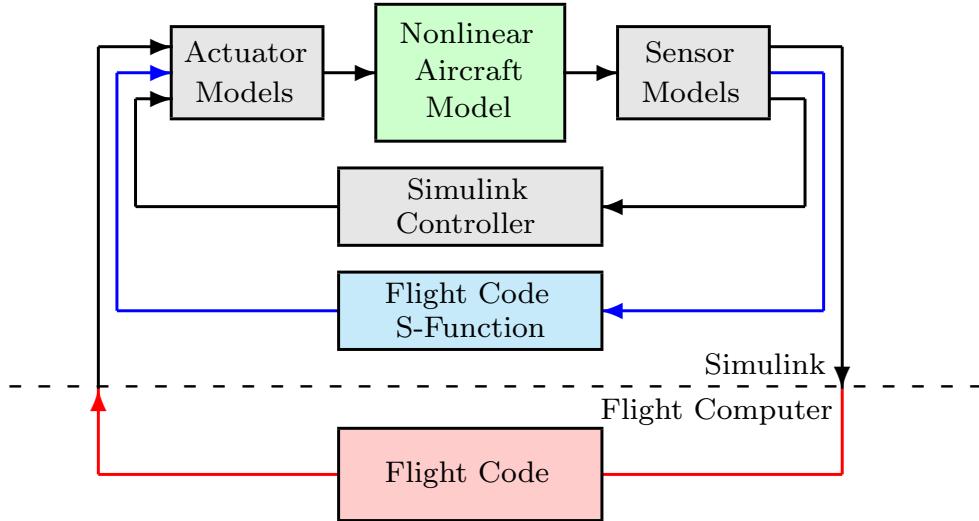


Figure 2.6: Three levels of simulation environments.

The lowest-level and most basic simulation allows for control algorithms to be implemented in Simulink. This frequently serves as a first step in the design process of new control algorithms. The mid-level simulation is a software-in-the-loop (SIL) simulation. The SIL simulation allows a research algorithm, written as flight code in C and interfaced via S-function, to be validated in Simulink. Finally, the highest-level simulation is a hardware-in-the-loop (HIL) simulation. The HIL simulation allows a research algorithm, written as flight code in C and implemented on a flight computer, to be interfaced with Simulink and validated. The latest versions of all three simu-

lations can be downloaded as a package from the SVN, which is directly accessible from the research group website [4].

2.2.1 Nonlinear Aircraft Simulation

A 6 degree-of-freedom (DOF) nonlinear simulation model of the aircraft dynamics is implemented in Simulink. This model represents a set of conventional rigid-body equations of motion for generic fixed-wing aircraft. Forces and moments due to aerodynamics, propulsion, and the environment are integrated numerically to solve the nonlinear differential equations. The environmental model includes a detailed model of Earth's atmosphere, gravity, magnetic field, wind, and turbulence. Models of the aircraft subsystems, such as actuators and sensors are also included.

Each test vehicle is associated with three simulation components: physical properties, a propulsion model, and an aerodynamic model. This allows the nonlinear simulation model to be easily reconfigured for a particular test vehicle. Physical properties for each airframe are determined in the lab, where moments of inertia are found using bifilar pendulum swing tests. Wind tunnel tests are used to characterize the motor and propeller thrust, torque, and power for each aircraft.

The aerodynamic models vary depending on the airframe. The Ultra Stick 120 aerodynamic model is derived from extensive wind tunnel data obtained at NASA Langley Research Center [8]. This is a high-fidelity model that covers large ranges of angle-of-attack and angle-of-sideslip aerodynamics, and is implemented as a look-up table. The Ultra Stick 25e aerodynamic model is derived using flight test data and frequency domain system identification techniques [6]. This model is linear and assumes constant aerodynamic coefficients. The Ultra Stick Mini aerodynamic model is based strictly on wind tunnel data obtained by the UMN UAS research group.

Linear models of the aircraft dynamics (about an operating point) are frequently desired for the design of control algorithms. The 6 DOF nonlinear simulation model is set up for trimming and linearization. Automatic functions to perform these tasks are provided in the simulation package. After the performance of a typical control algorithm has been verified using the linearized dynamics, it must be verified using the nonlinear simulation. The gray-shaded controller shown in Figure 2.6 illustrates this verification process.

2.2.2 Software-in-the-loop

The SIL simulation uses the 6 DOF nonlinear simulation model in feedback with a control algorithm implemented as flight code in C. This implementation of the control algorithm is interfaced with Simulink through an S-function block. Figure 2.6 represents the SIL simulation with the blue-shaded controller. The flight control algorithm alone is linked to the S-function; the remainder of the flight software is not included. The primary purpose of the SIL simulation is to verify the accuracy of a control algorithm transition from Simulink (mathematical discrete-time model) to flight code written in C.

2.2.3 Hardware-in-the-loop

The HIL simulation is an extension of SIL simulation that includes the flight software and flight computer. In Figure 2.6, this simulation environment is represented by the red-shaded controller. The entire flight software suite is compiled and runs on the flight computer in sync with the nonlinear simulation model. The MathWorks Real-Time Windows Target toolbox [25] is used to ensure the simulation runs in real-time on a Windows PC. This is crucial to obtain meaningful results when the flight computer is included in the simulation loop.

The nonlinear simulation model, in Simulink, interfaces with the flight computer using a serial connection. The flight software is modified in two ways in order to interface correctly with the HIL simulation. First, the data acquisition code (which normally solicits the onboard sensors) reads sensor data from the nonlinear simulation. Second, the actuator commands (which are normally delivered to the actuators via PWM signals) are sent back to the nonlinear simulation. Through the HIL verification process, any implementation issues or bugs associated with a control algorithm are identified and resolved. The HIL simulation is also useful in testing attitude and navigation state estimation algorithms, such as the one described in Sec. 2.3.2.

The HIL simulation provides an interface for an R/C pilot through a USB R/C-style remote. The aircraft state can be visualized via FlightGear [26], which is an open-source flight simulator. This interface can be used to evaluate the performance and handling qualities of a control algorithm prior to flight testing.

2.3 Flight Software

The software implemented on the MPC5200B flight computer is programmed as a single-thread, real-time process executing at 50 Hz. The entire real-time software suite and simulation environment (described in the previous section) are managed by a version control and documentation management system.

2.3.1 Real-time Software

The flight software is divided into code modules that are called in sequence by the main function. Each module is dedicated to a certain type of computation, e.g. attitude/navigation estimation or control algorithm. All candidate modules use an interface layer, which allows the software engineer to easily select which modules to compile in order to build the full program. In general, each type of module must be present in the compiled code, even if no computation native to a given module type is required. Figure 2.7 shows a schedule and order diagram of the code modules implemented in the flight software.

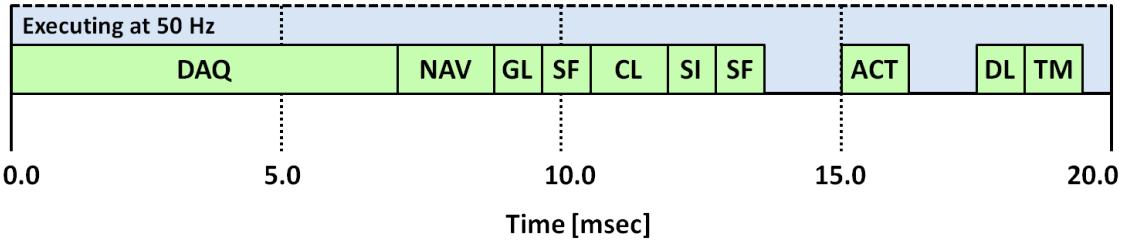


Figure 2.7: Real-time software schedule and order diagram of code modules.

A real-time clock is managed by an open-source, real-time operating system (RTOS) called eCos [14]. The RTOS provides alarms to the flight software that trigger code modules to execute. Three alarms are scheduled and validated in order to allow enough time for the software to execute on a 0.02 second frame. The data acquisition module (DAQ) is triggered by the first alarm immediately at the start of a new frame. This module reads data from the onboard sensors and is allowed the longest time to execute. For the HIL simulation, a different DAQ module is compiled in order to interface with Simulink. An INS/GPS algorithm is executed in the navigation filter module (NAV). Commands to the control system are generated in the guidance law module (GL). Potential sensor faults, such as biases, can be added in the sensor fault module (SF). The main control algorithm is computed next (CL). Signals required for

system identification (SI) can be augmented after the control law. Potential surface faults can be added in the surface fault module (SF). Once this sequence of modules has executed, the software waits for the next alarm.

The actuator module (ACT), which sends PWM signals to the actuators, is triggered by the second eCos alarm. For HIL simulation, a different ACT module is compiled in order to interface with Simulink. It is important that the time in between the DAQ and ACT modules is consistent; it represents the time delay of the flight computer. After the ACT module has executed, the software waits for the next alarm. A third eCos alarm allows the data logging (DL) and telemetry (TM) modules to execute. The sequence of modules then repeats as the software waits for the DAQ alarm from eCos, which indicates the start of a new frame.

The flight software is built and compiled using a `makefile`. This `makefile` specifies which version of each code module is included in the program. For example, to compile software for HIL simulation instead of flight, the HIL versions of DAQ and ACT are selected in the `makefile`. This approach allows for modularity and software flexibility.

2.3.2 Navigation State Estimation

The attitude state of an aircraft must be estimated from measurements provided by the onboard sensors. Position and velocity states, on the other hand, can be measured directly with a GPS receiver. Typical receivers, however, do not provide data at a sufficiently high rate for use in feedback control of aircraft. For example, the GPS receiver used by the UMN UAV Research Group provides data to the flight computer at 1 Hz. A sensor fusion algorithm is thus required to provide accurate and high bandwidth estimates of the aircraft attitude, position, and velocity. These states are known as the navigation states.

The navigation state estimates are computed with an algorithm that integrates an inertial navigation system (INS) with GPS. More in-depth description of INS/GPS integration can be found in the literature [27–30]. An INS provides measurements at higher bandwidth than a typical GPS receiver. For example, the IMU used by the UMN FCS provides measurements at 800 Hz. Due to flight software limitations, however, the IMU is only sampled at 50 Hz by the flight computer. Numerically integrating IMU measurements to obtain the navigation state estimates leads to un-

bounded errors that grow over time. A combined INS/GPS algorithm provides a solution that has the high bandwidth of the INS and the drift-free long-term stability of the GPS measurement. The structure of the algorithm is shown by the diagram in Figure 2.8.

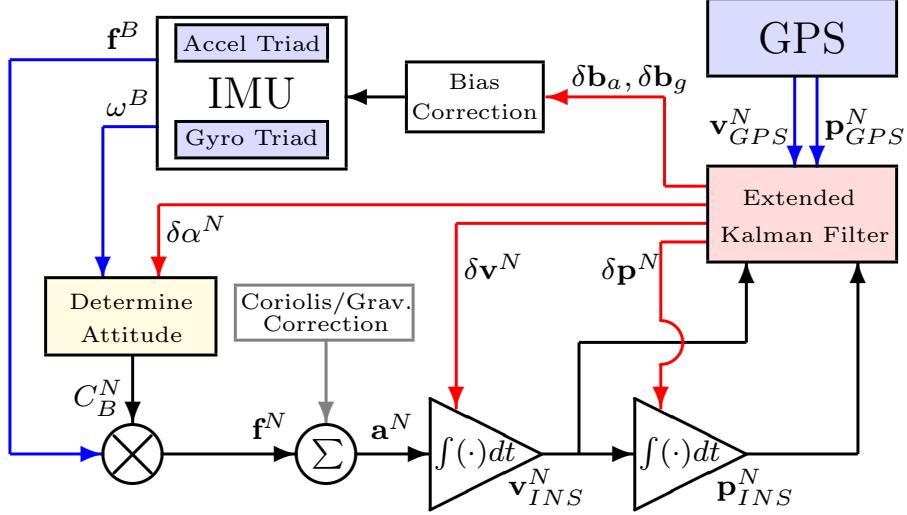


Figure 2.8: Block diagram of INS/GPS integration algorithm.

Attitude can be equivalently described by Euler angles or the quaternion. The INS/GPS estimation algorithm utilizes the quaternion for computation, and converts the solution to Euler angles for control. Attitude determination using an IMU calls for integrating the angular velocity measurement ω^B to propagate the attitude forward in time. Rate gyroscopes measure inertial rotation and should be compensated to account for the Earth's rotation rate and the transport rate due to the Earth's curvature [28]. For consumer/automotive IMUs used in low-cost UAV applications, however, these terms are small ($\sim 10^{-5}$ rad/s) compared to the noise level in the sensors, and are thus neglected.

The INS uses measurement of the acceleration to generate position and velocity estimates. A triad of accelerometers in the IMU provides measurement of the force-over-mass (specific force) acting on the aircraft. When the IMU is rigidly attached to the aircraft, this specific force is measured in the body frame (\mathbf{f}^B) and thus needs to be rotated into the navigation frame (\mathbf{f}^N) before integration and propagation in time. This rotation uses the aircraft attitude to formulate the required transformation matrix C_B^N . After compensating for gravity and the Coriolis effect, the force can be integrated once to yield velocity, and twice to yield position.

An Extended Kalman Filter (EKF) [31] is used to correct the attitude, velocity, and position estimates for errors. In order to improve the navigation solution between GPS measurements, and to allow coasting during short GPS outages, the EKF makes frequent corrections to compensate for the inertial sensor errors. Although more sophisticated sensor error models exist, a simplified model presented in [32] is used. This model is robust to parameters that are unobservable when the aircraft is not accelerating [27]. Using this model, the estimated sensor bias not only represents the true bias corrupting the measurement, but also accounts for all unmodeled errors that corrupt the sensor measurement.

The INS/GPS algorithm is initialized as soon as a valid GPS measurement becomes available (when implemented on the flight computer). Accordingly, position and velocity estimates are initialized at the first available position and velocity measurements. Since the initialization is set to occur on the ground before a flight test begins, the attitude is initialized to an approximate attitude of the aircraft on the ground. Figure 2.9 shows a portion of navigation flight data obtained during an experiment test with Thor.

The flight test results in Figure 2.9 show the navigation solution (computed in real-time onboard the aircraft) along with GPS measurements as markers. The INS/GPS integration algorithm connects the GPS measurements with a smooth, high

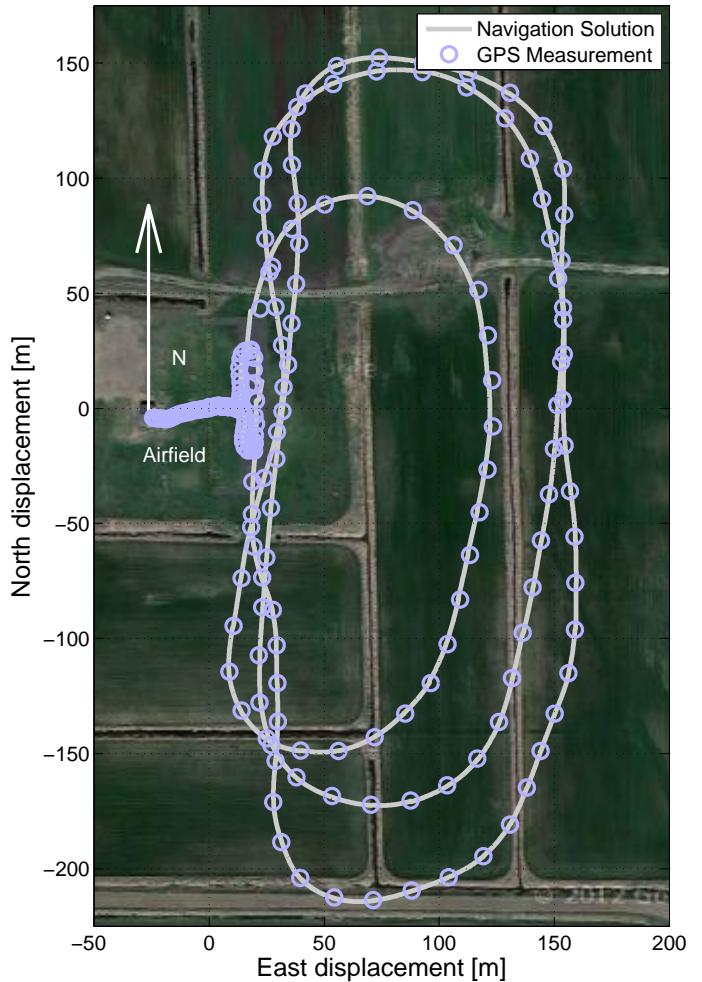


Figure 2.9: Navigation filter solution.

bandwidth position estimate. Innovations in the position estimate from the EKF are small (typically below 2 m for position, and 0.5 m/s for velocity), which indicates convergence in the algorithm. Although this flight test was conducted with an Ultra Stick 25e test vehicle, numerous data sets exist in the SVN of Ultra Stick 120 flight tests that validate the same attitude and navigation algorithm.

2.3.3 Version Control and Documentation

The flight software and simulation environments are managed by a version control program. The UMN UAS test platform utilizes the open-source subversion [33] server to manage a software development repository. This repository is available publicly on the research group website [34]. The flight software is automatically documented using Doxygen [35]. This utility allows documentation to be generated directly from the source code. Each file in the flight software suite has a special Doxygen header that allows the automatic documentation to be generated.

Chapter 3

System Identification

An important part for the development of a UAS test platform is modeling the aircraft dynamics. System identification techniques, which rely on experimental data obtained in flight, have been developed by the aerospace community to accomplish this task efficiently. However, the application of system identification techniques to small, low-cost unmanned aircraft poses a challenge. Physical airframe size and cost restrictions limit the availability and quality of onboard sensors. For example, sensors required to reliably measure angle-of-attack or angle-of-sideslip may not be available. Furthermore, the available sensors may be susceptible to high levels of noise. This challenge is addressed with a practical procedure to identify the dynamics of a small unmanned aircraft based on flight data obtained using a low-cost IMU.

Foundations and mathematical background of system identification theory are covered in detail by References 36 and 37. References 38, 39, and 40 summarize various engineering approaches to system identification, including time domain and frequency domain methods. Novel identification techniques have been published in recent literature, detailing advancements in areas such as real-time identification and efficient control surface input design [41–47]. Computational software tools are also readily available to automate parts of the system identification process [36, 38, 39]. What remains unclear, however, is the viability of the various techniques for application to small, low-cost unmanned aircraft. For example, some techniques assume that highly accurate sensor measurements are available, while others require measurement of the full aerodynamic state. These assumptions are unrealistic for low-cost platforms.

The technical approach described in this chapter is based on system identification in the frequency domain. A linear state-space model is derived from the generic non-linear equations of motion for aircraft. Parameters in the linear model are identified by fitting the model to frequency responses extracted from flight data, based on the approach in Reference 38. This approach is similar to the frequency domain output-error methods in References 39 and 47, but relies on a different cost function. It also implements a different algorithm to transform flight data into the frequency domain. Both methods share the advantage of fitting models over frequency ranges relevant to aircraft dynamics, and neither requires measurement of every state in the model. Using frequency responses, however, preserves insightful ties to flight dynamics and Bode plots. Its drawbacks include the need for longer and less efficient flight experiments [47]. The approach is applied to an Ultra Stick 25e aircraft code named Thor, shown in Figure 3.1. It is based on similar analysis applied to small, low-cost, rotorcraft [41–43]. The work in this chapter is drawn primarily from Reference 6.



Figure 3.1: University of Minnesota Ultra Stick 25e code name Thor.

This chapter is organized as follows: Section 3.1 provides the nonlinear equations of motion, linearized model, and simplifying assumptions. Physical properties of the the Ultra Stick 25e and the first principles analysis used to obtain a baseline model are described in Section 3.2. Flight experiments used to collect informative data are designed using the baseline model and operational constraints in Section 3.3. Section 3.4 briefly describes the theory behind frequency domain system identification, and results are given in Section 3.5. Finally, Section 3.6 describes a sensitivity and residual analysis. The system identification task is complemented with a model validation technique, which is presented in the next chapter.

3.1 Fixed-Wing Aircraft Dynamics

A nonlinear model for fixed-wing aircraft dynamics can be derived from the generic rigid-body equations of motion. Conventional aircraft are subject to external forces and moments due to gravity, propulsion, and aerodynamics. The central modeling task is to determine expressions for these external forces and moments. A simple nonlinear model is obtained when the equations of motion are written in the vehicle body-axis (see References 39 and 48). Standard nomenclature is used for the states: x-y-z body-axis velocities (u, v, w), x-y-z body-axis angular rates (p, q, r), and a standard 3-2-1 ordered rotation sequence of Euler angles (ϕ, θ, ψ). The x-y-z body-axis aerodynamic forces are denoted X, Y , and Z , and the x-y-z body-axis aerodynamic moments are denoted L, M , and N .

For simplicity, gyroscopic effects of the rotating mass of the motor are assumed to be insignificant, and the thrust T is assumed to act through the center of gravity and coincide with the body x-axis. The resulting system is summarized by the following equations [39]:

Force Equations:

$$\dot{u} = (rv - qw) + X/m - g \sin \theta + T/m \quad (3.1)$$

$$\dot{v} = (pw - ru) + Y/m + g \cos \theta \sin \phi \quad (3.2)$$

$$\dot{w} = (qu - pv) + Z/m + g \cos \theta \cos \phi \quad (3.3)$$

Moment Equations:

$$\dot{p} - (I_{xz}/I_x)\dot{r} = -qr(I_z - I_y)/I_x + qpI_{xz}/I_x + L/I_x \quad (3.4)$$

$$\dot{q} = -pr(I_x - I_z)/I_y - (p^2 - r^2)I_{xz}/I_y + M/I_y \quad (3.5)$$

$$\dot{r} - (I_{xz}/I_z)\dot{p} = -pq(I_y - I_x)/I_z - qrI_{xz}/I_z + N/I_z \quad (3.6)$$

Kinematic Equations:

$$\dot{\phi} = p + \tan \theta(q \sin \phi + r \cos \phi) \quad (3.7)$$

$$\dot{\theta} = q \cos \phi - r \sin \phi \quad (3.8)$$

$$\dot{\psi} = \sec \theta(q \sin \phi + r \cos \phi) \quad (3.9)$$

The dynamic response of an aircraft can be recorded with an IMU. Low-cost sensors

measure angular rates (p, q, r) and translational accelerations (a_x, a_y, a_z) [18]. The measured accelerations, however, exclude the effect of gravity [39]:

$$a_x = \dot{u} - (rv - qw) + g \sin \theta \quad (3.10)$$

$$a_y = \dot{v} - (pw - ru) - g \cos \theta \sin \phi \quad (3.11)$$

$$a_z = \dot{w} - (qu - pv) - g \cos \theta \cos \phi \quad (3.12)$$

The nonlinear model is linearized by assuming small perturbations from a steady, level trim condition. For simplicity, state variables in the nonlinear equations of motion are recast using the same notation as perturbation states in the linear model. The longitudinal dynamics are decoupled from the lateral/directional dynamics, and the thrust is assumed to be constant.

3.1.1 Longitudinal Dynamics

The longitudinal dynamics are described by the states $x_{lon} = [u, w, q, \theta]^\top$, which correspond to Equations 3.1, 3.3, 3.5, and 3.8. The forces X and Z , and the moment M are assumed to be linear functions of u, w, q , and the elevator deflection δ_{elev} , resulting in the following system:

$$\dot{x}_{lon} = A_{lon}x_{lon} + B_{lon}\delta_{elev} \quad (3.13)$$

where

$$A_{lon} = \begin{bmatrix} X_u & X_w & X_q - W_e & -g \cos \theta_e \\ Z_u & Z_w & Z_q + U_e & -g \sin \theta_e \\ M_u & M_w & M_q & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad B_{lon} = \begin{bmatrix} X_{\delta_{elev}} \\ Z_{\delta_{elev}} \\ M_{\delta_{elev}} \\ 0 \end{bmatrix}$$

The terms W_e , U_e , and θ_e represent the trim condition. The X , Z , and M terms with subscripts are the dimensional aerodynamic derivatives to be identified. Coefficients in the A_{lon} matrix are the stability derivatives, and the B_{lon} matrix holds the control derivatives. Finally, the linearized acceleration measurements are given by:

$$a_x = \dot{u} + qW_e + g \cos \theta_e \theta + g \sin \theta_e \quad (3.14)$$

$$a_z = \dot{w} - qU_e + g \sin \theta_e \theta - g \cos \theta_e \quad (3.15)$$

The longitudinal dynamics can be decoupled further into the phugoid and the short-period modes. The phugoid mode is typically very slow, lightly damped, and dominates the response in u , θ , and a_x . The short-period mode is typically fast, moderately damped, and dominates the response in w , q , and a_z . For control applications, accurate knowledge of the phugoid mode is not crucial due to the low frequency of the oscillation, which is compensated for with feedback control. Stability and performance characteristics also depend primarily on the short-period mode [48]. System identification is applied to the short-period model shown in the following system, where the state vector is $x_{lon} = [w, q]^\top$:

$$A_{lon} = \begin{bmatrix} Z_w & Z_q + U_e \\ M_w & M_q \end{bmatrix} \quad B_{lon} = \begin{bmatrix} Z_{\delta_{elev}} \\ M_{\delta_{elev}} \end{bmatrix} \quad (3.16)$$

Terms in Equations 3.14 and 3.15 that depend on θ are neglected for the short-period model. Furthermore, the short-period aircraft response captured by a_x is small relative to the measurement noise on a low-cost IMU. Hence, a_x is not used in this identification analysis.

3.1.2 Lateral/Directional Dynamics

The lateral/directional dynamics are described by the states $x_{lat} = [v, p, r, \phi, \psi]^\top$, which correspond to Equations 3.2, 3.4, 3.6, 3.7, and 3.9. Force Y , and moments L and N are described by linear functions of v , p , r , and aileron and rudder deflections (δ_{ail} and δ_{rud} , respectively). The resulting system is given by the following:

$$M_{lat}\dot{x}_{lat} = A'_{lat}x_{lat} + B'_{lat} \begin{bmatrix} \delta_{ail} \\ \delta_{rud} \end{bmatrix} \quad (3.17)$$

where

$$M_{lat} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -I_{xz}/I_x & 0 & 0 \\ 0 & -I_{xz}/I_z & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A'_{lat} = \begin{bmatrix} Y_v & Y_p + W_e & Y_r - U_e & g \cos \theta_e & 0 \\ L_v & L_p & L_r & 0 & 0 \\ N_v & N_p & N_r & 0 & 0 \\ 0 & 1 & \tan \theta_e & 0 & 0 \\ 0 & 0 & \sec \theta_e & 0 & 0 \end{bmatrix} \quad B'_{lat} = \begin{bmatrix} Y_{\delta_{ail}} & Y_{\delta_{rud}} \\ L_{\delta_{ail}} & L_{\delta_{rud}} \\ N_{\delta_{ail}} & N_{\delta_{rud}} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

The Y , L , and N terms with subscripts in A'_{lat} and B'_{lat} are the dimensional aerodynamic derivatives to be identified. Unlike the longitudinal dynamics, the lateral/directional dynamics cannot be decoupled into independent modes. They are governed by a slow spiral mode, a fast lightly damped dutch roll mode, and an even faster roll mode. Finally, the linearized acceleration measurement is given by:

$$a_y = \dot{v} - pW_e + rU_e - g \cos \theta_e \phi \quad (3.18)$$

The longitudinal and lateral/directional systems form a linear parametric model that is used as a basis for frequency domain system identification. Stability and control derivatives are identified by fitting these models to frequency responses extracted from flight data.

3.2 Preliminary Analysis

The task of system identification is simplified by incorporating a priori knowledge of the aircraft dynamics into the analysis. A baseline model is generated to gain insight into the general characteristics of the system, such as its gain and bandwidth. This information is used as a guide to design informative and efficient system identification flight experiments, for which a rough approximation of the dynamics is sufficient.

Recall that Thor is an Ultra Stick 25e conventional fixed-wing airframe with aileron, rudder, and elevator control surfaces. The aircraft is instrumented with an IMU as

part of the FCS. A ground test, with the throttle set to around 70%, resulted in a noise amplitude is approximately 2 deg/s in each angular rate channel, and 0.5 m/s² in each acceleration channel. The flight computer records manual pilot command before delivering the signals to the actuators. This is particularly useful here, for system identification flight experiments, where augmenting manual pilot commands with automatically generated excitation signals helps maintain the aircraft near the trim condition. Physical airframe properties for Thor were given in Table 2.1 in Section 2.1, and these parameters are complemented by moment of inertia values in Table 3.1. The moment of inertia values for Thor were estimated using bifilar pendulum swing tests.

Table 3.1: Moments of inertia for Thor.

Property	Symbol	Value	Units
Moment of Inertia	I_x	0.089	$kg\ m^2$
Moment of Inertia	I_y	0.144	$kg\ m^2$
Moment of Inertia	I_z	0.162	$kg\ m^2$
Cross Moment of Inertia	I_{xz}	0.014	$kg\ m^2$

Various methods can be applied to obtain a baseline model. If the airframe is similar to an already modeled aircraft, its model can be scaled. However, if the aircraft configuration and airfoils are new, empirical methods and/or simple wind tunnel tests can be implemented. For example, the Digital DATCOM [49] is a purely empirical guide to estimating stability and control derivatives based on aircraft configuration and the experience of engineers. Simple wind tunnel tests can also be used, in particular to obtain estimates of control derivatives and stability derivatives associated with the body velocity components u , v , and w . The key point is that various methods exist to obtain a baseline model, and depending on the available resources, a combination of methods can be used.

A baseline model for the flight dynamics is generated using aerodynamic data from two similar airframes. Control derivatives and stability derivatives associated with the body velocities are estimated from wind tunnel tests performed with an Ultra Stick Mini. This airframe is smaller than the 25e and fits in the wind tunnel available at the University of Minnesota. The 25e and the Mini have similar aerodynamics but are not exact geometric scales of each other. Stability derivatives associated with the angular rates are taken from an aerodynamic model for the Ultra Stick 120 [4]. This airframe is larger than the 25e, has similar aerodynamics, yet it also is not an exact

geometric scale. The aerodynamic model for the 120 was developed at NASA Langley Research Center, using both static and dynamic wind tunnel testing [50, 51]. Due to these approximations in the aerodynamics, the baseline model for Thor is only used as a guide to design flight experiments.

For the wind tunnel tests performed on the Ultra Stick Mini, airspeed is held constant while aerodynamic forces and moments on the aircraft are measured by a sensor. The first two tests consist of static variations in angle-of-attack and angle-of-sideslip. These tests are sufficient to obtain estimates of the stability derivatives associated with the body velocity components. A third wind tunnel test is conducted to estimate the control derivatives. In this test, each control surface is deflected independently while aerodynamic forces and moments are measured.

3.2.1 Longitudinal Dynamics

The longitudinal control derivatives, velocity stability derivatives, and equilibrium terms are estimated using wind tunnel data acquired with the Ultra Stick Mini. The angular rate stability derivatives are taken from values estimated for the Ultra Stick 120 [50, 51]. Along with mass data from Table 2.1, the longitudinal baseline model is constructed and given by:

$$A_{lon} = \begin{bmatrix} -0.38 & 0.60 & -0.36 & -9.80 \\ -0.98 & -7.81 & 15.32 & -0.21 \\ 0.18 & -8.31 & -35.21 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad B_{lon} = \begin{bmatrix} -0.36 \\ -3.62 \\ -106.32 \\ 0 \end{bmatrix}$$

The modes of the longitudinal dynamics are computed from an eigenvalue decomposition of the system state matrix and presented in Table 3.2.

Table 3.2: Estimated modes of the longitudinal dynamics.

Mode	Natural Frequency [rad/s]	Damping Ratio
Phugoid	0.48	0.43
Short-Period Pole 1	13.70	-
Short-Period Pole 2	29.28	-

The phugoid mode has a natural frequency of 0.48 rad/s with a damping ratio of 0.43. The typical short-period mode does not appear in the baseline model. Instead, it is

replaced by two stable real poles at 13.70 and 29.28 rad/s. This is unconventional for fixed-wing aircraft and is a result of estimating derivatives using the Ultra Stick Mini and 120. However, this result is not a major concern for the baseline model since the approximate bandwidth associated with the longitudinal dynamics can still be inferred.

3.2.2 Lateral/Directional Dynamics

The lateral/directional control derivatives, velocity stability derivatives, and equilibrium terms are estimated using wind tunnel data acquired with the Ultra Stick Mini. The angular rate stability derivatives are taken from values estimated for the Ultra Stick 120 [50, 51]. The populated matrices are shown by the following:

$$M_{lat} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -0.157 & 0 & 0 \\ 0 & -0.086 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A'_{lat} = \begin{bmatrix} -1.64 & 0.64 & -18.90 & 9.50 & 0 \\ -2.14 & -13.71 & 13.71 & 0 & 0 \\ 0.93 & -0.12 & -7.28 & 0 & 0 \\ 0 & 1 & 0.03 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad B'_{lat} = \begin{bmatrix} 0 & 6.99 \\ -68.65 & 18.32 \\ -8.03 & -19.01 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (3.19)$$

The modes of the lateral/directional dynamics are computed from an eigenvalue decomposition of the system state matrix, and are presented in Table 3.3.

Table 3.3: Estimated modes of the lateral/directional dynamics.

Mode	Natural Frequency [rad/s]	Damping Ratio
Spiral	0.05	-
Dutch Roll	6.03	0.77
Roll	12.38	-

The spiral mode is represented by a pole at 0.05 rad/s, the dutch roll mode has natural frequency of 6.03 rad/s with damping ratio 0.77, and the roll mode is represented by a pole at 12.38 rad/s.

3.3 Design of Flight Experiments

Research experiments begin when the pilot is ready to engage the on-board flight computer. Immediately following a turn in the race track pattern, the pilot trims the aircraft to a desired flight condition. In this case, the flight condition for Thor is straight and level flight at approximately 19 m/s. The pilot then engages the flight computer to execute the research experiment. Due to range and other spacial constraints, Thor can only achieve a 20 second maximum time window of straight and level flight. When the experiment is complete, the pilot disengages the flight computer and continues the race track pattern. The pilot can choose to realign the aircraft for additional experiments, or to conclude the flight test with a manual landing.

The aircraft dynamics must be excited during a flight experiment in order to successfully perform system identification. Automatic frequency sweep inputs are used to accomplish this over a broad frequency range. These inputs are computer generated sinusoids with frequencies that vary logarithmically with time. The flight computer applies signals for one control surface at a time in order to prevent correlation between the inputs. Frequency sweeps are designed using the default logarithmic chirp function in MATLAB, which implements the following equation:

$$\delta(t) = A \sin(f(t)t), \quad \text{where} \quad f(t) = f_0 \left(\frac{f_1}{f_0} \right)^{t/t_1} \quad (3.20)$$

In this relationship, the amplitude A is specified, as well as a frequency interval given by f_0 and f_1 (in Hz). A time vector t is required, where the final time is given by t_1 .

Frequency sweep inputs can take the aircraft away from the trim condition [38, 39]. To counter this effect, the inputs are augmented with a manual pilot input via the flight computer. The pilot counters the drift by ensuring that the nose and wings remain level over the course of the maneuver. Pilot augmentation is only permitted for the control surface on which the active frequency sweep is applied. All remaining control surfaces are fixed at their trim values throughout the experiment. If multiple inputs were active simultaneously, the extracted frequency response would need to be conditioned for the effect of the secondary input on the primary input-output response. To ensure that each experiment begins with airspeed close to 19 m/s, the throttle setting is fixed to 70%.

Several practical factors constrain the design of frequency sweeps experiments. Due

to trimming requirements before and after each turn, a 10 second experiment time window is the approximate limit for the Ultra Stick 25e. As a result, dynamics at frequencies below 0.1 Hz cannot be identified accurately. Given a data sampling rate of 50 Hz, the Nyquist limit indicates that signals above 25 Hz cannot be recorded accurately. A more practical limit for system identification is closer to 10 Hz [38]. The fastest pole in the baseline model is located around 30 rad/s, or about 5 Hz. Hence, the upper frequency limit is not a major concern. The phugoid and spiral modes are located below 0.1 Hz. They cannot be identified accurately due to a lack of excitation in this frequency range. However, this is not of great concern as the slow nature of these dynamics can be easily handled by a pilot or a control system.

Servo actuator dynamics must also be considered in the design of system identification experiments. Specifications on actuator bandwidth are often unavailable from low-cost hobby manufacturers. Ground tests on Thor indicated that the bandwidth of the servos is below 15 Hz. Therefore, frequency sweeps from 0.1 to 15 Hz would adequately excite the flight dynamics relevant for control applications as well as the actuator dynamics.

3.3.1 Frequency Sweep Design

Figure 3.2 shows the baseline model frequency response for each control surface to its corresponding primary angular rate response. Conventional aircraft dynamics are dominated by these input-output relationships. Accurate models for these relationships are thus a key requirement for control applications. The 0.1 to 15 Hz frequency range is highlighted by the solid curves.

To achieve a sufficiently high signal-to-noise ratio (SNR) on Thor, the measured output response must exceed 6 deg/s for the angular rates, and 1.5 m/s² for the accelerations. This accounts for about a factor of 3 between the magnitude of the noise and the magnitude of the response, which represents the minimum desired SNR [38]. Experience has shown that higher SNR, closer to a value of 6, can improve the quality of the frequency responses. A frequency sweep with amplitude of 4 degrees is chosen for safety and to satisfy SNR requirements. Higher amplitudes would generally be considered unsafe due to the uncertainty in the baseline model.

The identification frequency range is broken down into two intervals: low and high frequency. Each interval has a dedicated experiment to ensure that the entire fre-

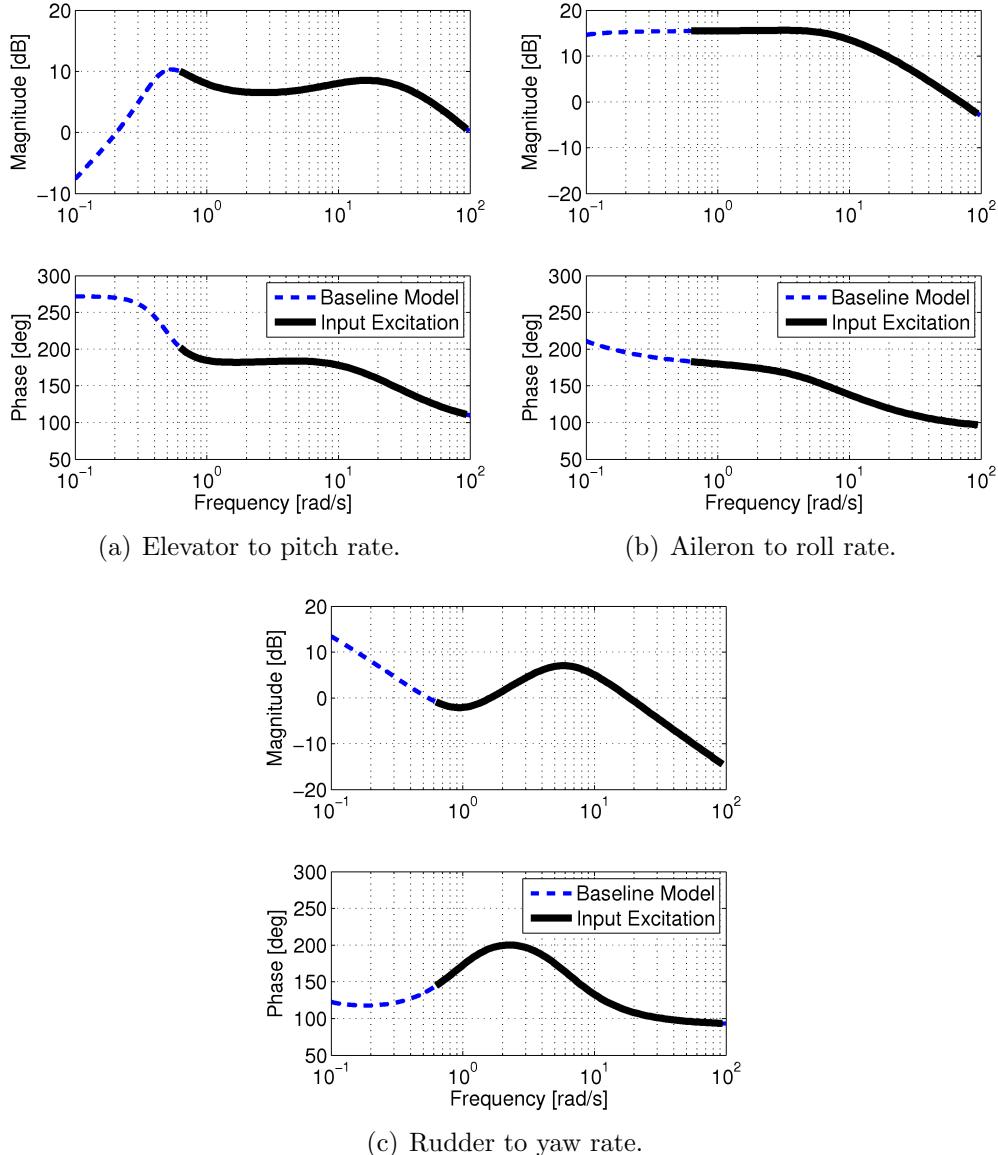
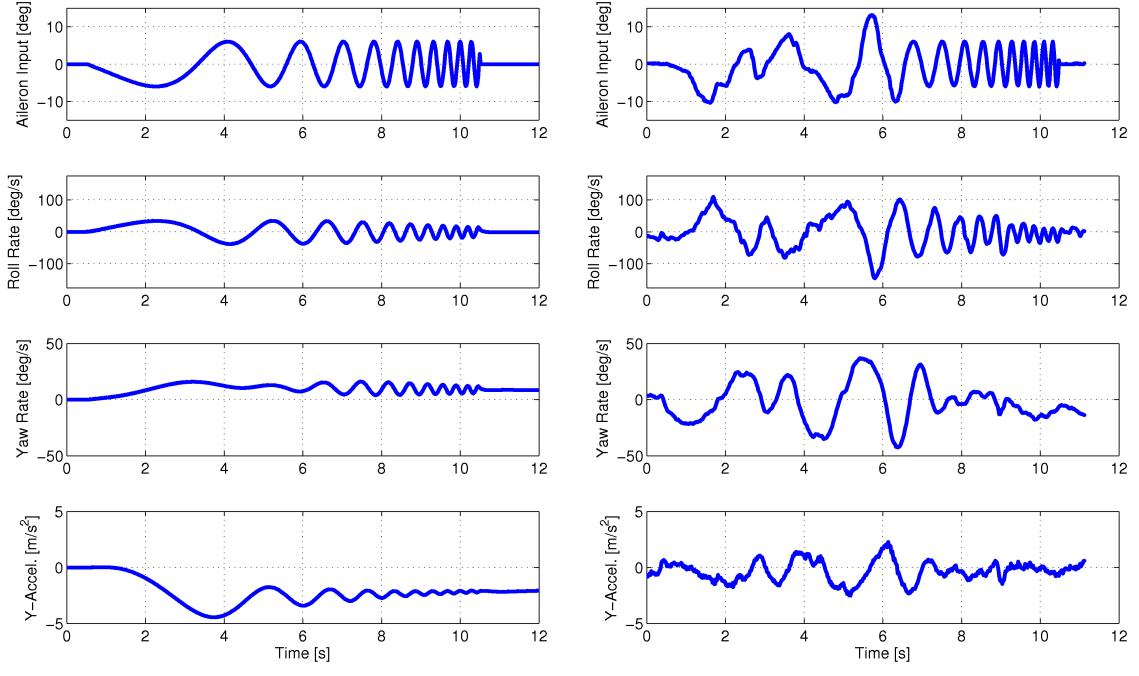


Figure 3.2: Baseline model frequency response with range for system identification.

frequency range is excited sufficiently. The low frequency interval spans 0.1 to 5 Hz, and the high frequency interval spans 4 to 15 Hz. Five frequency sweep experiments are conducted for each interval on each control surface to obtain a rich data set. Multiple runs are required because frequency responses are ultimately extracted from the flight data using an averaging process. Figure 3.3 shows both simulation and flight results from a low frequency aileron sweep. The pilot command augmentation is clearly visible on the right.



(a) Aileron sweep baseline model simulation.

(b) Aileron sweep flight experiment data.

Figure 3.3: Sample aileron frequency sweep input signal and response.

The flight data shown on the right in Figure 3.3 indicates that the manual pilot augmentation helps keep the aircraft around the desired trim condition during the experiment. The pilot is able to maintain oscillations about the trim condition without canceling out the input excitation, eliminating the bias noted at the end of the maneuver in the baseline simulation. The true aircraft exhibits higher gain in the roll rate channel than predicted by the baseline model. However, the baseline model accurately predicts low gain in the off-axis, particularly in the y-component acceleration response.

3.4 Frequency Domain System Identification

The frequency domain system identification process is comprised of two steps. The first identification step extracts frequency responses using spectral quantities computed from the input-output flight data. Control surface commands recorded by the flight computer are considered inputs because sensors are not available to measure the true surface deflections. The second identification step fits the linear state-space models to the extracted frequency responses. Parameters in the linear models are

identified through a nonlinear optimization that minimizes the fitting error in the frequency domain.

3.4.1 Extracting Frequency Responses

The basic frequency domain identification problem is cast for a two-input, single-output system without the loss of generality [52]. This formulation can easily be modified to include additional inputs. A multiple-output model, such as the lateral/directional aircraft model, is constructed by superposing sets of multiple-input, single-output relationships. The block diagram in Figure 3.4 shows the fundamental transfer functions and signals:

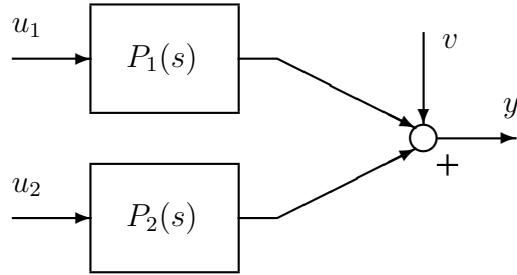


Figure 3.4: Diagram of a two-input, single-output system.

Signals u_1 and u_2 are inputs to the system, which are represented by transfer function blocks $P_1(s)$ and $P_2(s)$. Signal v introduces measurement noise on the output measurement signal y . It is assumed that the noise disturbance is white and uncorrelated with the inputs u_1 and u_2 .

Identifying transfer functions $P_1(s)$ and $P_2(s)$ is simple if the input signals u_1 and u_2 are uncorrelated. Uncorrelated inputs are obtained in practice by exciting each control surface independently, justifying why frequency sweep experiments are executed for one control surface at a time. To prevent biases due to correlation between the measurement noise and the inputs [37], it is best to perform open-loop experiments. Operating in open-loop is particularly beneficial for systems equipped with sensors that are susceptible to high levels of measurement noise. Under these conditions, the spectral input-output relationship is given by the following equation:

$$S_{y,y}(s) = |P_1(s)|^2 S_{u_1,u_1}(s) + |P_2(s)|^2 S_{u_2,u_2}(s) + S_{v,v}(s) \quad (3.21)$$

In this relationship, $S(s)$ represents a complex-valued spectral density function. $P_1(s)$ and $P_2(s)$ are obtained from cross- and auto-spectral density functions for the input and output signals:

$$P_1(s) = \frac{S_{y,u_1}(s)}{S_{u_1,u_1}(s)} \quad P_2(s) = \frac{S_{y,u_2}(s)}{S_{u_2,u_2}(s)} \quad (3.22)$$

Spectral quantities are estimated from the input and output flight data. For more details on spectral analysis, see References 38 and 52. Applying a standard Hanning window with 50% overlap is one simple approach to obtain a smooth frequency response estimate. Experience has shown that this windowing technique works well for aircraft systems. An appropriate window length remains to be selected. Window length is directly related to the low frequency limit of the estimated frequency response, where longer windows allow lower frequencies. The maximum window length is given by the data record length, which in this case is 10 seconds. Longer windows, however, reduce the total number of windows applied to the data record and diminish the averaging effect. As a result, the estimated frequency response exhibits more random error, particularly at frequencies where the SNR is low and averaging would have been the most helpful. Hence, there is no single optimal window length that provides both high accuracy and broad dynamic range in an estimated frequency response.

A frequency response estimate can be improved by using data from several individual frequency responses, each obtained with a different window length, to form a composite frequency response. The composite frequency response blends the averaging benefits of shorter windows with the dynamic range advantages of longer windows. In this analysis, 5 frequency responses are obtained using 2, 4, 6, 8, and 10 second windows. The basic principle used to generate the composite frequency response emphasizes frequency response data from the response with the highest coherence at each individual frequency point. Coherence functions, denoted $\gamma^2(s)$, measure the linear correlation between signals. For example, a coherence value of 1 indicates that the entire output response is accounted for by the input via a linear transfer function. Hence, the composite frequency response is generated by emphasizing data from the frequency response with the highest coherence. More complex optimization-based approaches have been developed to obtain more accurate composite frequency responses [38]. However, coherence weighting alone yields sufficiently accurate results for small unmanned aircraft.

In general, coherence values less than one imply the presence of “non-ideal” effects in the input-output relationship, such as nonlinear dynamics, unmeasured inputs, disturbances, or measurement noise. High coherence, in practice above 0.7, is desired for accurate frequency domain system identification. Coherence functions for the system in Figure 3.4 based on spectral density functions, assuming uncorrelated inputs, are given by:

$$\gamma_{u_1,y}^2(s) = \frac{|S_{y,u_1}(s)|^2}{S_{u_1,u_1}(s) S_{y,y}(s)} \quad \gamma_{u_2,y}^2(s) = \frac{|S_{y,u_2}(s)|^2}{S_{u_2,u_2}(s) S_{y,y}(s)} \quad (3.23)$$

Transfer functions $P_1(s)$ and $P_2(s)$ are replaced by experimentally obtained frequency responses in the subsequent parametric identification analysis. This basic insight on the spectral estimation process must be incorporated into the design of flight experiments in order to achieve the most accurate frequency response estimates.

3.4.2 Parametric Identification

The longitudinal and lateral/directional aircraft dynamics are identified by fitting their corresponding linear parametric models to estimated frequency responses. This is implemented as a nonlinear optimization that aims to minimize the error of the fit in the frequency domain. The decision variables in the optimization are the stability and control derivatives from the state and input matrices of the linear parametric models. Hence, the identification subspace is described by the set of aerodynamic parameters that determine the state-space representation of the aircraft dynamics. The optimization uses a cost function to capture errors in the frequency domain (over a desired interval) between the linear parametric models and the estimated frequency responses. The cost function is weighted based on high coherence to emphasize the fit where the estimated frequency responses accurately capture the system response.

The ability to identify a physically meaningful aircraft model depends on the number of free parameters in the linear model relative to the information captured by the estimated frequency responses. In general, the information captured is limited by the available sensor measurements and the experimental constraints. Small, low-cost unmanned aircraft are equipped with a limited quantity of sensors, which restricts the number of available estimated frequency responses. Furthermore, the 10 second experiment time window for Thor implies that the phugoid mode cannot be excited.

This mode is also significantly decoupled from the remaining longitudinal dynamics. Including free parameters in the optimization that describe this non-excited, decoupled mode would result in over-parametrization. To address this issue, parameters associated with the phugoid mode are fixed to their baseline values. This assumption can introduce errors in the identified model in the form of a low frequency mismatch. However, low frequency errors can easily be handled by feedback control and hence are not of great concern.

Flight dynamics relevant for control applications can be identified with the proposed approach using measurements from an IMU alone. This is a significant advantage for low-cost systems with limited sensor equipment. Other approaches, such as the state-space formulation of the equation-error method [39], require measurement of every state of the aircraft model. This is not a feasible requirement for a low-cost UAS. Frequency domain parametric identification also has some general advantages over time domain identification. Flight dynamics relevant to control are dominant in a particular frequency range. Identification in the frequency domain allows accurate modeling to be emphasized in this frequency range.

The parametric identification is performed using CIFER[®], a frequency domain system identification tool in the aerospace industry [38]. Originally developed for rotorcraft identification, this tool has also been applied to fixed-wing aircraft [47, 53]. A non-linear optimization is implemented for the parametric identification, emphasizing a close model fit in portions of the frequency response with high coherence. Additional known dynamics, such as actuator dynamics and system time delay, can be augmented to the linear parametric models. These additional dynamics allow the optimization to identify a model that closely matches the estimated frequency responses. Actuator dynamics and system time delay are identified separately using a method described in the next section.

3.5 Identification Results

The longitudinal and lateral/directional models are identified separately. The simple structure of the short-period model is exploited first to identify the actuator dynamics and system time delay using the concept of Low-Order Equivalent Systems (LOES). This approach which was originally developed in the 1970s to certify aircraft handling qualities [54]. Parametric identification in CIFER[®] is performed using

the linear aircraft models presented in Section 3.1 to determine the longitudinal and lateral/directional flight dynamics.

3.5.1 Longitudinal Dynamics

Figure 3.5 shows the estimated frequency responses and the final parametric identification results for the elevator input to the pitch rate and z-axis acceleration measurements. The frequency response for the pitch rate is on the left, and for the z-axis acceleration on the right. The baseline model is also shown for comparison, indicating a significant deviation from the final results.

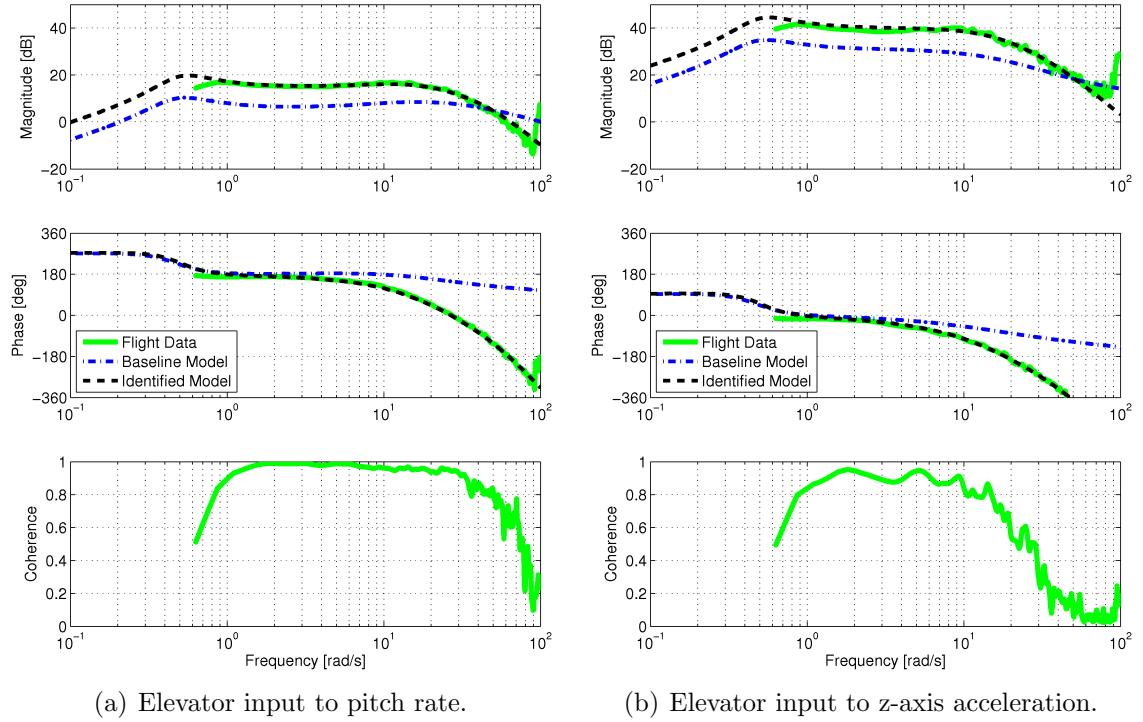


Figure 3.5: Longitudinal dynamics identification for elevator input.

The short-period mode has a known structure that can be fitted with a LOES [47]. Analysis based on LOES is performed first in order to identify the actuator dynamics and system time delay. Once these additional dynamics have been determined, they are appended to the short-period state-space model and fixed for the parametric identification in CIFER®. This approach is expected to introduce errors in the final identification results. However, it is necessary in order to avoid having too many free parameters that need to be identified. LOES modeling is based on using a transfer function plus a time delay to match an experimental frequency response.

The transfer function for the elevator to pitch rate response, which represents the short-period mode, is given by the following:

$$P_{sp}(s) = \frac{K_{sp}(s + a_{sp})}{s^2 + 2\xi_{sp}\omega_{sp}s + \omega_{sp}^2} \quad (3.24)$$

This transfer function has relative degree one, hence its frequency response must exhibit a first order roll-off. The corresponding estimated frequency response on the left in Figure 3.5, however, exhibits a third order roll-off. This mismatch suggests that the actuator dynamics should be modeled as a transfer function with relative degree two.

Parameters in Equation 3.24 are tuned to fit the estimated frequency response up to the bandwidth of the short-period mode. The resulting LOES short-period model has a natural frequency $\omega_{sp} = 17.3$ rad/s, damping ratio $\xi_{sp} = 0.65$, gain $K_{sp} = -107.4$, and zero $a_{sp} = 14.5$ rad/s. The actuator is identified as a second-order low-pass filter with a 50.27 rad/s bandwidth. Together, the LOES short-period model and the actuator dynamics provide a good fit of the magnitude curve. A 50 msec first-order Pade approximation is included to model the system time delay, which provides a good fit of the phase curve. Part of this delay is attributed to the 20 msec computer computation time. The remaining time delay captures unmodeled and higher-order dynamics, such as the nonlinear effect of actuator rate limits.

Parametric identification is used to fit the short-period state-space model (shown in Equation 3.16) to the estimated frequency responses, given fixed actuator dynamics and system time delay. The results are used to update the baseline model, which completes the longitudinal axis identification process. Derivatives corresponding to the phugoid mode are not updated since they are fixed to their baseline values. The final identified model is shown by the frequency responses in Figure 3.5. The results indicate that the identified model successfully captures the longitudinal flight dynamics, along with actuator dynamics and time delay. Based on the coherence functions, the model is expected to be accurate from 1 rad/s up to 70 rad/s in the angular rate channel, and up to 20 rad/s in the acceleration channel. Dynamics near the bandwidth are most important for control applications, and the results indicate that they are captured accurately by the identified model. Table 3.4 summarizes the modal characteristics of the longitudinal dynamics.

Table 3.4: Identified longitudinal dynamics.

Mode	Frequency [rad/s]	Damping	Time Constant [s]
Phugoid	0.51	0.38	12.32
Short-Period	16.33	0.83	0.39
Actuator	50.27	0.80	0.13
Time Delay	-	-	0.05

Equation 3.25 provides the system matrices that represent the identified model shown in Figure 3.5 and Table 3.4. Note that the stability and control derivatives corresponding to the phugoid dynamics are maintained at their baseline values.

$$A_{lon} = \begin{bmatrix} -0.38 & 0.60 & -0.36 & -9.80 \\ -0.98 & -10.65 & 16.74 & -0.21 \\ 0.18 & -5.39 & -16.55 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad B_{lon} = \begin{bmatrix} -0.36 \\ -3.62^1 \\ -141.57 \\ 0 \end{bmatrix} \quad (3.25)$$

The identified model deviates significantly from the baseline model, notably particularly from the magnitude plots in Figure 3.5. Further, a standard oscillatory short-period mode is identified and shown in Table 3.4. The poor accuracy of the baseline model is attributed to the approximations made during its construction.

3.5.2 Lateral/Directional Dynamics

The lateral/directional dynamics are more complicated to identify than the longitudinal dynamics. Although the spiral mode is very slow and cannot be excited given the experimental constraints, it cannot be decoupled in the state-space model and fixed to a baseline value. As a result, the lateral/directional model has a large number of free parameters relative to the information captured by the estimated frequency responses. Low gain in some cross-coupling relationships, such as in the rudder to roll rate channel, complicate the problem further.

The lateral/directional actuator model is assumed to be the same as in the longitudi-

¹Parametric identification found the cost function to be highly insensitive to the $Z_{\delta_{elev}}$ control derivative, indicating poor accuracy in the identified value of this parameter. The derivative was fixed at the baseline value and the optimization reconverged for the remaining parameters. More details in Section 3.6.

nal dynamics. This assumption is valid because all control surfaces are actuated by the same type of servo. The time delay, in general, captures unmodeled and higher-order effects and hence could be different in the lateral/directional axes. However, it is assumed that the time delay is the same throughout the aircraft to avoid introducing another unknown parameter. These assumptions simplify the lateral/directional identification problem, for which over-parametrization is already a significant concern. Figure 3.6 shows the identification results for the aileron and rudder control surface inputs to the roll rate response. Figure 3.7 shows the yaw rate response, and Figure 3.8 shows the y-axis acceleration response.

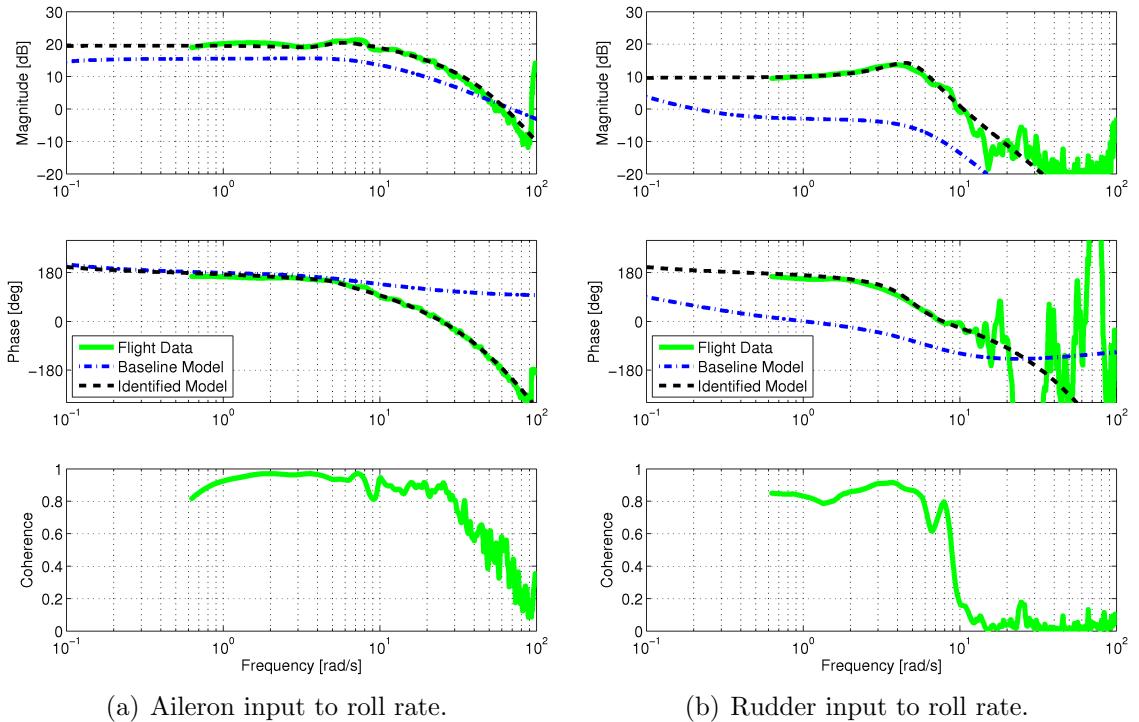


Figure 3.6: Lateral/directional dynamics identification for roll rate output.

Figure 3.6 shows the identification results for the control surface inputs to the roll rate response. Based on the coherence functions, the model is expected to be accurate from around 1 rad/s to 40 rad/s for the aileron channel, and from around 1 rad/s to 8 rad/s for the rudder channel. The roll rate is expected to be identified accurately throughout the bandwidth of the dynamics for both channels. Note that the aileron dynamics roll off at -20 dB/dec, while the rudder dynamics roll off at -40 dB/dec. As a result, gain in the rudder channel is attenuated more at higher frequencies. This is confirmed by the coherence function dropping at a lower frequency in the rudder

channel than in the aileron channel.

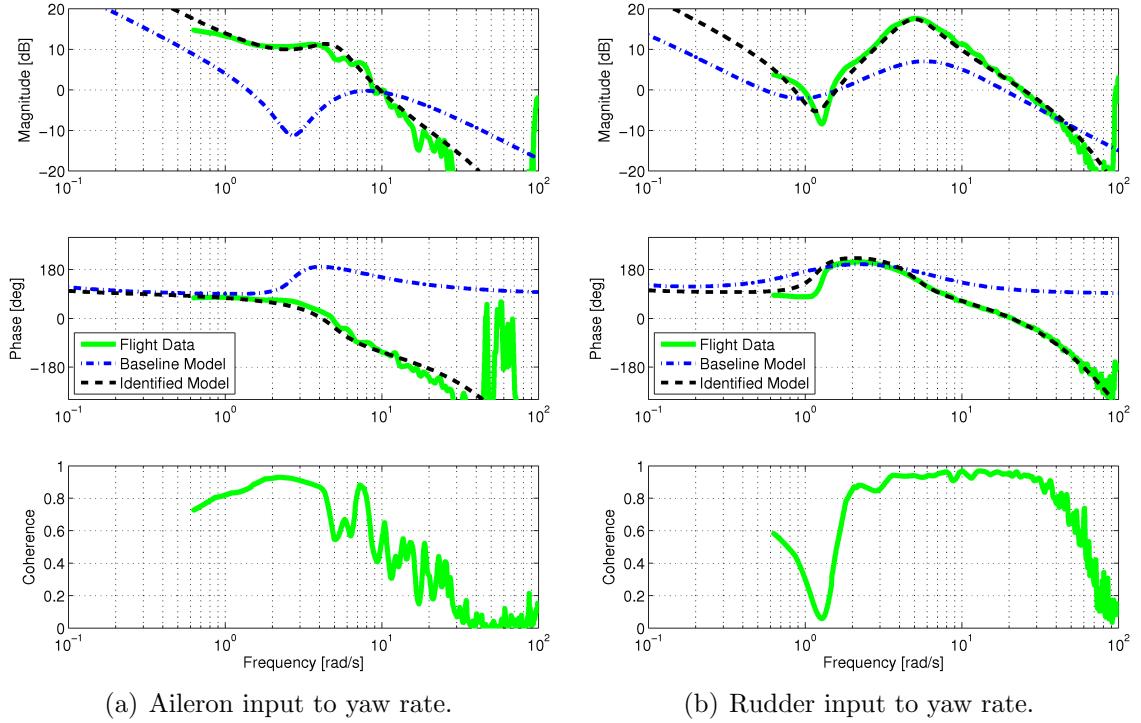


Figure 3.7: Lateral/directional dynamics identification for yaw rate output.

Figure 3.7 shows the identification results for the control surface inputs to the yaw rate response. Gain in the aileron to yaw rate channel is about 10 dB lower than in the aileron to roll rate channel, which reduces quality in the estimated frequency response due to lower SNR. The aileron to yaw rate channel is accurately modeled from around 1 rad/s to 10 rad/s, which captures the bandwidth of the dynamics. The rudder to yaw rate dynamics show a pair of complex-conjugate zeros near 1 rad/s. The frequency response has low gain near this frequency, which makes the damping of the zero dynamics challenging to identify. This effect is confirmed by the low coherence function in the neighborhood of 1 rad/s in the rudder to yaw rate channel. Overall, the rudder to yaw rate channel is accurately modeled from 2 rad/s to 70 rad/s, which captures the bandwidth of the dynamics.

Figure 3.8 shows the identification results for the control surface inputs to the y-axis acceleration response. The coherence function in the aileron channel is only sufficiently high from 1 rad/s to 4 rad/s, which is a narrower range than desired. The bandwidth of this frequency response is not accurately identified due to a sharp drop

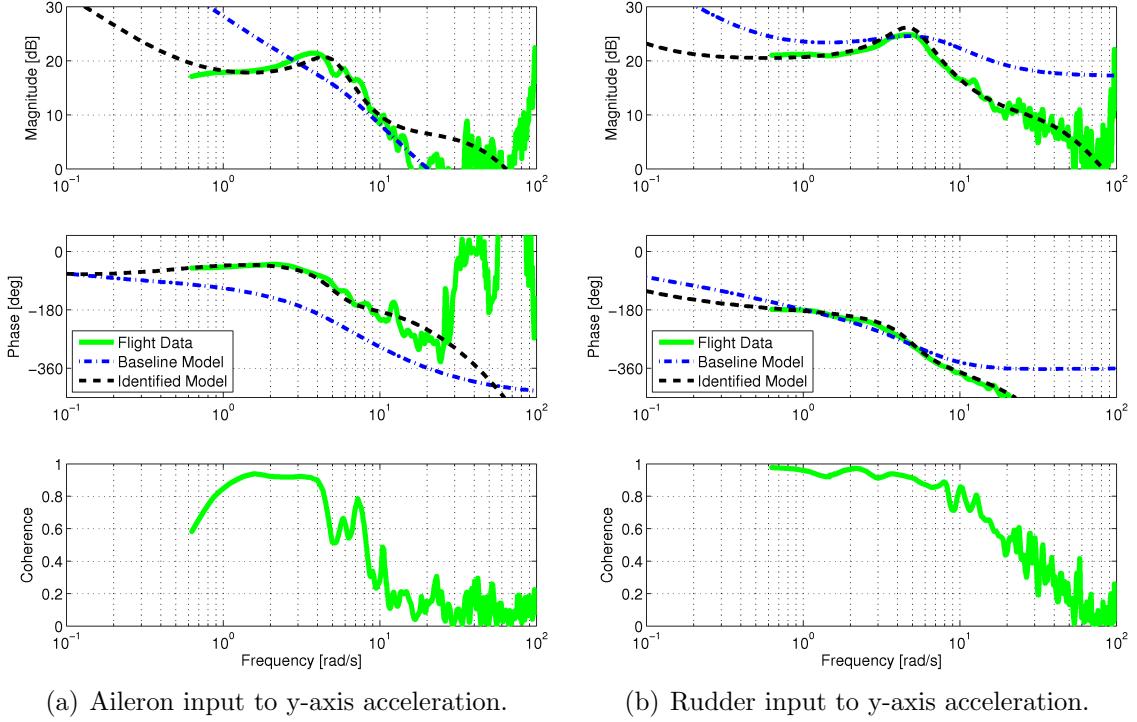


Figure 3.8: Lateral/directional dynamics identification for y-axis acceleration output.

in the gain. The rudder to y-component acceleration has higher gain than the aileron. Hence, the coherence function is high and the model is accurately identified from 1 rad/s to 20 rad/s, which covers the relevant dynamic range.

Table 3.5 summarizes the modal characteristics of the identified lateral/directional dynamics. The slow response associated with the spiral mode is impossible to excite given the prevailing experimental constraints. This is expected to lead to poor identification of the spiral mode. Similar to the phugoid mode, however, this is not a critical issue for control applications. Actuator dynamics and system time delay for the lateral/directional model are assumed to be the same as for the longitudinal model, which were given in Table 3.4.

Table 3.5: Identified lateral/directional dynamics.

Mode	Frequency [rad/s]	Damping	Time Constant [s]
Spiral	0.02	-	314.16
Dutch Roll	4.96	0.33	1.27
Roll	12.53	-	0.50

The identified A'_{lat} and B'_{lat} matrices are presented in Equation 3.26. Unlike for the

longitudinal case, the entire set of lateral/directional stability and control derivatives are updated. The matrix M_{lat} is not updated since the mass properties are assumed to be known.

$$A'_{lat} = \begin{bmatrix} -0.64 & 0.46 & -18.21 & 9.50 & 0 \\ -2.02 & -12.47 & 4.05 & 0 & 0 \\ 1.30 & 0.86 & -3.09 & 0 & 0 \\ 0 & 1 & 0.03 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad B'_{lat} = \begin{bmatrix} -2.06 & 2.98 \\ -139.10 & 6.52 \\ 17.22 & -26.42 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (3.26)$$

The results in Figures 3.6 - 3.8 indicate that the identified lateral/directional model is most accurate in the roll rate response due to both aileron and rudder inputs, and for the yaw rate and y-component acceleration due to rudder input. Overall, the identified model is significantly more accurate than the baseline model in all channels. This first iteration of frequency domain system identification provides crucial insight regarding the true aircraft dynamics of this Ultra Stick 25e. The findings also provide guidelines for the design of additional flight experiments to be used in refining the model.

3.6 Sensitivity and Residual Analysis

The system identification process is extended with a sensitivity and residual analysis of the results. Techniques presented in Reference 38 are used to construct an uncertainty model based on sensitivities to parameter variation. Residuals obtained from the spectral analysis are used to generate an output disturbance model. Together, the uncertainty and disturbance models complement the identified model and provide additional insight into the reliability of the result.

3.6.1 Sensitivity to Parameter Variation

Uncertainty in the identified model is quantified using properties of the converged cost function from the parametric identification process. This cost function depends on the identified parameters and captures the model fit error in the frequency domain. Its sensitivity to parameter variations, or combinations of parameter variations, is used to model uncertainty. Cramér-Rao (*CR*) bounds represent this type of sensitivity in the cost function, and hence are used as a basis for a parametric uncertainty model.

CR bounds depend on the diagonal entries of the inverse Hessian matrix \mathcal{H}^{-1} , evaluated for the converged cost function. The Hessian matrix is estimated and CR bounds are calculated as part of the identification process. A large CR_i bound indicates either low curvature in the cost function (high insensitivity) with respect to the i th parameter, or that the i th parameter is correlated with another parameter [38]. Separately, CR bounds also represent a lower bound on the standard deviation σ of the statistical scatter expected from running multiple experiments [37, 38]. This is shown by the following relationship:

$$\sigma_i \geq CR_i = \sqrt{(\mathcal{H}^{-1})_{ii}} \quad (3.27)$$

Experience has shown that a factor of 2 can be used to obtain an approximation of the standard deviation, resulting in Equation 3.28 [38]:

$$\sigma_i \approx 2CR_i = 2\sqrt{(\mathcal{H}^{-1})_{ii}} \quad (3.28)$$

A parametric uncertainty model is constructed based on CR bounds. The uncertainty describes a family of identified models expected from running multiple experiments. High CR bounds are attributed to several factors. For example, a parameter may be physically insignificant with respect to the measured aircraft response. This kind of parameter is difficult to identify and hence associated with a high CR bound. Reduced coherence in the estimated frequency response can also lead to high CR bounds. Due to weighting based on high coherence in the identification process, low coherence can mask the effect of varying a parameter. Finally, correlation between parameters indicates that they can vary together, making their individual values difficult to determine. Relating parametric uncertainty to CR bounds is one way to collect these identification problems and form a comprehensive uncertainty model.

The uncertainty model is constructed by letting the model parameters vary on an interval centered at their nominal identified values. Each parameter is modeled as a fixed interval uniform distribution. This type of model is useful since it can be analyzed directly with existing robustness tools, such as μ -analysis. To provide a conservative estimate of the uncertainty, the interval is selected to include 3 standard deviations from the nominal value [43]. Figure 3.9 illustrates how uncertainty in all the parameters manifests as a variation in the frequency response. The uncertainty

model is randomly sampled to generate a family of frequency responses. In this example, two families of frequency responses are shown. The aileron input to roll rate channel is on the left, and the rudder input to yaw rate channel is on the right. The nominal identified model is highlighted by the darker dashed curves.

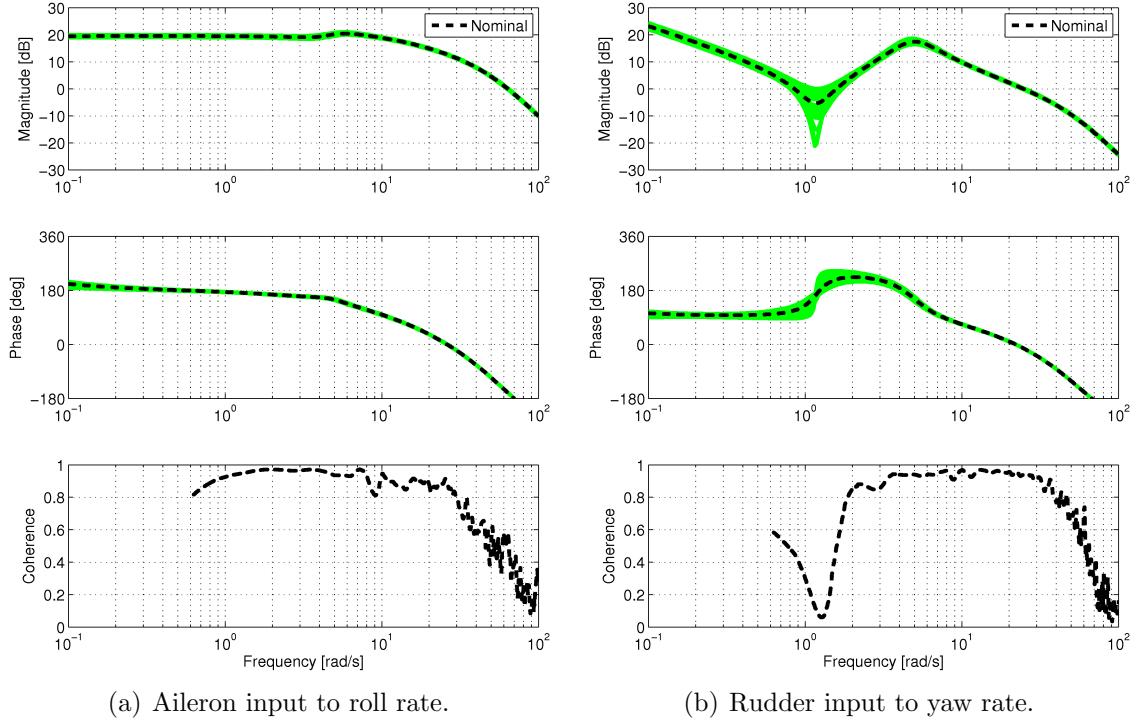


Figure 3.9: Selected uncertain frequency responses.

Figure 3.9 is useful because it provides a visual representation of the expected variation in the frequency response due to parametric uncertainty. The aileron to roll rate channel is accurately identified, as indicated by the high coherence function. Accordingly, low variation due to uncertainty is noted. The rudder to yaw rate model, however, is poorly identified near 1 rad/s. This is due to low gain in the transfer function. Uncertainty is high due to the low coherence, and significant variation is noted in the frequency response. The uncertainty shows that the damping of a pair of complex conjugate zeros in the transfer function is not accurately modeled. Due to the low coherence, the optimization is unable to determine the damping, which leads to high uncertainty in the identified parameters of that mode.

The effect of uncertainty must be considered in every relevant input-output channel. Exhaustive analysis of each remaining channel in the aircraft model is left out in the interest of brevity. However, visualizing the uncertainty is a straightforward process.

The full set of CR bounds obtained from the identification process (including the factor of 2) is given in Table 3.6. The bounds are given as percentage deviations from the nominal identified values of their corresponding aerodynamic derivatives, which were provided in the identified system matrices.

Table 3.6: Cramér-Rao bounds for the identified aircraft model.

Deriv.	$2CR$ (%)	Deriv.	$2CR$ (%)	Deriv.	$2CR$ (%)	Deriv.	$2CR$ (%)
Z_w	14.88	Y_v	5.17	L_v	9.16	N_v	3.49
Z_q	8.41	Y_p	20.20	L_p	8.20	N_p	16.30
$Z_{\delta_{elev}}$	(129.4)	Y_r	0.74	L_r	16.82	N_r	6.58
M_w	20.69	$Y_{\delta_{ail}}$	38.42	$L_{\delta_{ail}}$	7.25	$N_{\delta_{ail}}$	7.07
M_q	16.28	$Y_{\delta_{rud}}$	11.96	$L_{\delta_{rud}}$	41.94	$N_{\delta_{rud}}$	4.34
$M_{\delta_{elev}}$	8.49						

The CR bounds indicate that most parameters are identified accurately. Experience has shown that identification results are reliable when most $2CR_i < 20\%$ [38]. The rule of thumb is exceeded significantly by the $Z_{\delta_{elev}}$ parameter. This derivative represents the elevator z-axis force, which has a limited effect on the response relative to other forces and moments acting on the aircraft. The manifestation of the elevator z-axis force on a_z and q is negligible, and hence $Z_{\delta_{elev}}$ can vary greatly without having a significant impact on the measured input-output dynamics (or the cost function). Due to insensitivity in the cost function [38], noted by the 129.4% CR bound, the identified value is nearly arbitrary. Therefore, $Z_{\delta_{elev}}$ was fixed at the baseline value and the optimization reconverged for the remaining parameters. The original CR bound associated with $Z_{\delta_{elev}}$, however, was retained as uncertainty.

The $Y_{\delta_{ail}}$ and $L_{\delta_{rud}}$ derivatives represent cross-coupling relationships with low gain for the Ultra Stick 25e, which elevates their CR bounds. $Y_{\delta_{ail}}$ represents the aileron effectiveness in generating a y-axis force. $L_{\delta_{rud}}$ represents the rudder effectiveness in generating a rolling moment. It is difficult to obtain measurements with sufficiently high SNR and estimate accurate frequency responses for these relationships, primarily due to their low gain. As a result, errors and variations in the model fit are not significantly penalized in the parametric identification process, which results in elevated CR bounds for $Y_{\delta_{ail}}$ and $L_{\delta_{rud}}$.

3.6.2 Residual Disturbance Modeling

The effect of disturbances acting on the aircraft is captured by residual spectra computed for the output measurements. Residual spectra contain the portion of the output measurement that cannot be accounted for by the inputs via linear transfer functions. For this analysis, residual spectra are computed for the three angular rate measurements. An output disturbance model is generated, which complements the identified aircraft model.

The system diagram in Figure 3.4 shows the identification problem cast for a two-input, single output system. In the ideal scenario, the system is linear and subject to a white noise output disturbance v . Hence, the output disturbance is uncorrelated with the inputs. The inputs and outputs are related via linear transfer functions, and the coherence functions have values near 1. Therefore, in ideal conditions, the residual spectrum for the output is white noise. The output residual spectrum is computed with the following relationship [52]:

$$S_{v,v}(s) = [1 - \gamma_{u,y}^2(s)]S_{y,y}(s) \quad (3.29)$$

In practice, these ideal conditions cannot be fully satisfied because the aircraft is not a linear system and the output disturbance is not white noise. The effects of nonlinear dynamics, wind gusts, turbulence, and correlated noise on the measured output are compounded into the output residual spectrum. Hence, the inputs and outputs are no longer perfectly related via linear transfer functions, as assumed by the ideal scenario and in conjunction with Equation 3.29.

The residual spectrum of the output (computed with Equation 3.29) is used to infer a disturbance model for the aircraft dynamics. The disturbance on the measured output is modeled with a transfer function, denoted $D_y(s)$. This transfer function is driven by a unit amplitude white noise input signal v . The magnitude of the disturbance model transfer function is given by the following equation:

$$|D_y(s)| = \sqrt{|S_{v,v}(s)|} \quad (3.30)$$

Disturbance models are computed for the three angular rate measurements on Thor. Spectra and coherence functions obtained from the frequency sweep experimental

data are used in this analysis. Figure 3.10 shows a diagram that describes how the output disturbance enters the system. Three estimated transfer functions (obtained using Equation 3.22) relate the inputs to their primary angular rate responses. In the longitudinal axis, the disturbance model $D_p(s)$ is calculated for the elevator to pitch rate channel. In the lateral/directional axes, disturbance model $D_q(s)$ is calculated for the aileron to roll rate channel, and disturbance model $D_r(s)$ is calculated for the rudder to yaw rate channel.

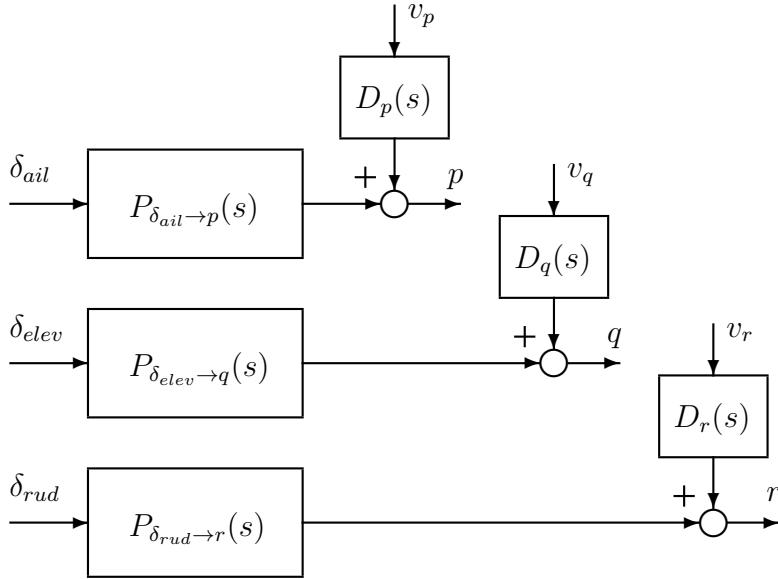


Figure 3.10: System diagram for output disturbance modeling.

The computed disturbance models are presented in Figure 3.11. Coherence functions are included below as a reference to indicate the predicted accuracy of the identified model. The roll rate disturbance model $D_p(j\omega)$ is shown on the left, the pitch rate disturbance model $D_q(j\omega)$ in the middle, and yaw rate disturbance model $D_r(j\omega)$ on the right. The results show that the output disturbance entering the system is not white noise, as assumed in the ideal scenario. This discrepancy can be attributed to the effects of nonlinear dynamics, wind gusts, turbulence, and correlated noise on the measured output.

The expected output disturbance is the measurement noise introduced by the IMU sensor. For each angular rate channel, noise with amplitude ± 2 deg/s is observed. This type of noise corresponds to a disturbance model with flat magnitude near -29 dB. However, the magnitudes of all three disturbance models in Figure 3.11 are greater than -29 dB, indicating the presence of additional disturbances. Moreover, the

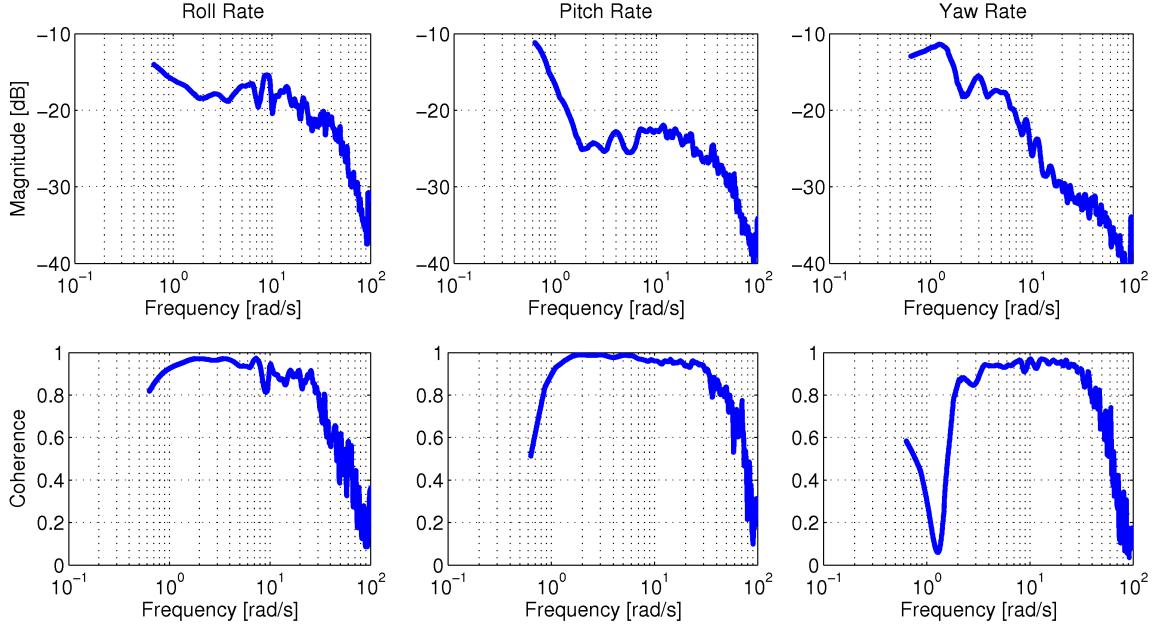


Figure 3.11: Output disturbance models for angular rate measurements.

disturbance models are qualitatively similar in frequency domain characteristics to their corresponding identified aircraft models. For example, the roll rate disturbance model on the left in Figure 3.11 is similar to the aileron to roll rate identified model on the left in Figure 3.6. The two models have similar bandwidth and roll-off characteristics. The same finding holds for the pitch and yaw rate disturbance models. This general observation suggests that input disturbances dominate the total disturbances captured in the measured aircraft response.

Input disturbances propagate through the aircraft dynamics. Their effects manifest directly on the residual output spectra, such as those shown in Figure 3.11. Each output residual spectrum is “colored” with the aircraft dynamics, which means they have similar frequency domain characteristics. The results in Figure 3.11 indicate that the disturbance models are “colored” with the identified aircraft dynamics. Therefore, disturbances acting on the input, such as wind gusts or turbulence, dominate the disturbances acting directly on the output, such as sensor noise. This result is important for understanding key performance limitations of small, low-cost unmanned aircraft. In particular, it shows that although low-cost sensors, like the IMU on Thor, are susceptible to high levels of noise, the presence of noise is not significant in comparison to the effect of wind gusts or turbulence experienced by the aircraft.

Chapter 4

Model Validation

The predictive accuracy of an identified model must be determined and quantified prior to the model being used for control design. Having precise knowledge of the expected accuracy is particularly important in model-based design applications. As noted with the F-35 program, relying on models that fail to identify design problems can cause major time and cost disruptions. Hence, the task of model validation is a critical component of the development process. In order to validate models efficiently, aerospace engineers need a practical framework that is easy to use with flight data. The underlying metric must provide a rigorous measure of the model quality in terms of robustness requirements for closed-loop control. The ultimate goal is therefore to develop a powerful model validation framework based on a metric that provides rigorous connections between model accuracy and control requirements. This chapter proposes a new metric that is similar to an existing and commonly used validation metric, yet also provides this type of connection.

Existing model validation techniques can generally be categorized by two mathematical approaches. One approach relies on linear matrix inequality optimizations [55–59]. Under this framework, a perturbation that accounts for the output error between a simulation and an experiment is identified. The result is said to *not invalidate the model* if the perturbation belongs to an allowable set. This approach provides rigorous conclusions about model quality, but is also limited by computational power and the types of model structures that can be analyzed. A more commonly used approach relies on statistical analysis [39, 60, 61]. Under this framework, the output error is analyzed directly based on raw statistics or a scaled signal norm. Conclusions

about model quality are drawn from the relative size of these statistics or the norm. Typically blind to model structure, this approach is more broadly applicable than optimization-based methods. However, it also provides less rigorous results. The most useful technique would combine analytic rigor with broad applicability in terms of model structures.

This chapter proposes the gap metric [62, 63] as a natural framework to validate aircraft models based on flight data. The gap metric can be thought of as a modified generalization of an existing and commonly used statistical validation metric: the Theil Inequality Coefficient (TIC) [60]. Building on properties of TIC analysis, however, validation in the gap metric is more analytically rigorous when applied to linear time-invariant (LTI) models. It is therefore ideally suited to support model-based design of flight control algorithms, which typically relies on linearized models of the aircraft dynamics. The approach is powerful in an engineering sense because it links the validation metric to a controls-centric notion of what constitutes an accurate model. It compares an identified aircraft model to flight data, via the gap metric, in order to derive a set of robustness requirements for closed-loop control. As a result, aerospace engineers gain confidence earlier in the development process that their control algorithms are safe to be implemented on the real aircraft.

4.1 Theoretical Background

The power of the gap metric as a tool for model validation stems from its roots in mathematics and control theory. Although the primary objective of this work is to build engineering perspective on the proposed framework, a brief description of the theoretical background is necessary. Many of the technical details are omitted and the reader is pointed to references for more information. However, certain key facts of the framework are explained so that engineers have sufficient context to use it in real model-based flight control applications.

Motivation for the proposed framework traces back to the TIC [60]. The TIC has been successfully used to validate models for a wide range of aerospace systems, including fixed-wing aircraft, helicopters, and missiles [40, 61, 64, 65]. Due to this experience, aerospace engineers feel comfortable with the metric and understand how to apply it to real flight data. One of the main goals of the framework is to propose a new metric that is similar to the TIC, namely the gap metric. The gap metric, however,

also provides a connection between model accuracy and robustness requirements.

For single-input single-output (SISO) systems, the TIC is defined as follows:

$$\text{TIC}(\hat{y}_1, \hat{y}_2) = \frac{\sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_{1i} - \hat{y}_{2i})^2}}{\sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_{1i})^2} + \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_{2i})^2}}, \quad (4.1)$$

where \hat{y}_1 is a sampled simulation time history and \hat{y}_2 is the corresponding output measurement obtained in flight. For this discrete time formulation, n represents the number of data points. Implicit in this definition is the assumption that the input to the simulation and the input to the aircraft is the same. Equation 4.2 shows a mathematically equivalent formulation of the TIC in continuous time for the output signals y_1 and y_2 :

$$\text{TIC}(y_1, y_2) = \frac{\| y_1 - y_2 \|_2}{\| y_1 \|_2 + \| y_2 \|_2}, \quad (4.2)$$

where $\|y\|_2 := \sqrt{\int_{-\infty}^{+\infty} y^2 dt}$ denotes a 2-norm.

Model validation using the TIC is very intuitive and easy to understand. The metric ranges from zero to one, where lower values indicate a better model. A value of zero corresponds to a perfect match, and a value of one indicates no correlation. Both transient and steady-state deviations are captured by the numerator, which computes a 2-norm on the output error. This norm measures the squared integral of the error. Finally, the TIC is scaled in the denominator by the individual 2-norms of the outputs being compared.

The TIC was originally proposed as a metric for economic forecasting [60]. In many economic forecasting applications, the *input* is often poorly defined and, hence, only output measurements are used for analysis. Accordingly, the TIC was not defined to consider input signals. For the analysis of aerospace systems, however, the input is an important quantity. Therefore, a reasonable extension of the TIC for application to aerospace systems is given by the following:

$$\text{TIC}_{sys}(y_1, y_2, u) = \frac{\left\| \begin{bmatrix} u \\ y_1 \end{bmatrix} - \begin{bmatrix} u \\ y_2 \end{bmatrix} \right\|_2}{\left\| \begin{bmatrix} u \\ y_1 \end{bmatrix} \right\|_2 + \left\| \begin{bmatrix} u \\ y_2 \end{bmatrix} \right\|_2}, \quad (4.3)$$

where u is the input signal applied to the simulation and to the real aircraft. Here, considering the input has no effect on the numerator and only modifies the denominator scaling. However, this system adaptation of the TIC is useful because, as shown later in this section, it is very similar to the gap metric.

A major shortcoming of the TIC, in any formulation, is the lack of rigorous connection to what constitutes a good model. It is difficult to relate a numerical value of this metric to a specific level of model accuracy. For example, there is no interpretation of the TIC in terms of a stability margin requirement for control. Engineers only know that, in general, lower TIC values indicate better models. One way to address this shortcoming is to use a metric similar to Equation 4.3: the gap metric. As a result, the validation framework gains analytic rigor through its connection between model accuracy and closed-loop robustness requirements provided by the gap metric.

The gap metric was originally introduced as a mathematical way to compare two systems [62, 63]. It differs from the TIC by allowing the inputs (e.g. simulation and experimental inputs) to be different in the analysis. Consequently, deviations are measured between input-output signal pairs. Allowing the inputs to differ is convenient for model validation analysis. For example, consider the case where unmodeled time delays cause a miss-alignment in the output signals. This type of data discrepancy is commonly found in flight control applications. The resulting TIC would be large, even though the primary dynamics are captured accurately by the model. In practice, engineers address this issue in an ad-hoc way by time shifting the signals manually. The gap metric allows the inputs to differ slightly in order to align the outputs. The resulting gap would be small, which correctly indicates an accurate model.

To formally define the gap metric, consider two SISO systems P_1 and P_2 with inputs u_1 and u_2 and outputs y_1 and y_2 . No special assumptions are required on the model structures. The gap metric is defined as the maximum of two directed gaps $\vec{\delta}(P_1, P_2)$ and $\vec{\delta}(P_2, P_1)$ [62, 63], where

$$\vec{\delta}(P_1, P_2) = \sup_{\|u_1\|_2 \leq 1} \inf_{u_2} \frac{\left\| \begin{bmatrix} u_1 \\ y_1 \end{bmatrix} - \begin{bmatrix} u_2 \\ y_2 \end{bmatrix} \right\|_2}{\left\| \begin{bmatrix} u_1 \\ y_1 \end{bmatrix} \right\|_2}. \quad (4.4)$$

Accordingly, the gap metric is given by:

$$\delta(P_1, P_2) = \max \{ \vec{\delta}(P_1, P_2), \vec{\delta}(P_2, P_1) \}. \quad (4.5)$$

The constraint on the supremum in Equation 4.4 bounds the 2-norm of u_1 , which is a standard condition for max-min optimizations. Similar to the TIC, the gap metric ranges from zero to one and measures deviations based on 2-norms. Equation 4.4 shows that the system adaptation of the TIC is a nearly special case of the gap metric, namely where $u_1 = u_2$ and with a different scale factor in the denominator. Hence, the gap metric can be thought of as a modified generalization of the TIC.

The theory of the gap metric has been advanced by many authors [63, 66–78] and the gap metric has been used to analyze both linear [71, 75–77] as well as nonlinear and adaptive control systems [79, 80]. However, in general, it is difficult to compute the gap metric using Equation 4.4, and moreover, stronger assumptions are needed to enable model validation using flight data. Yet for P_1 and P_2 being restricted to SISO LTI systems with transfer functions $P_1(s)$ and $P_2(s)$, an equivalent formulation exists in the frequency domain [67, 71]. A modification to this frequency-domain formulation, known as the ν -gap metric, enables more direct and efficient computation [77]. Gap analysis in the frequency domain is particularly convenient because it aligns with many system identification techniques for UASs [6, 39, 61] which are also in the frequency domain. For early attempts to develop gap-based system identification techniques, see [81–84]. A frequency-domain recasting of the gap metric is given next in order to transition into the ν -gap which will be used in the sequel.

A brief introduction to normalized coprime factors follows to introduce the required frequency-domain theory. Coprime factors are used in control theory to model uncertainty and quantify stability and robustness properties of closed-loop systems [85, 86]. Given a transfer function $P_1(s)$, define its numerator and denominator (coprime) polynomials $\tilde{N}_1(s)$ and $\tilde{D}_1(s)$ such that $P_1(s) = \tilde{N}_1(s)/\tilde{D}_1(s)$. The normalized coprime factorization of $P_1(s)$ is:

$$P_1(s) = N_1(s)/D_1(s), \quad (4.6)$$

where the so-called *normalized coprime factors*

$$N_1(s) = \tilde{N}_1(s)/Z_1(s) \quad \text{and} \quad D_1(s) = \tilde{D}_1(s)/Z_1(s) \quad (4.7)$$

are stable proper transfer functions, instead of polynomials, and satisfy

$$N_1(s) N_1(s)^* + D_1(s) D_1(s)^* = 1. \quad (4.8)$$

Here, $N_1(s)^* := N_1(-s)^T$ denotes the complex-conjugate transpose of $N_1(s)$ (when evaluated along the imaginary axis), and where T is the transpose operator. This also requires $Z_1(s)$ to be stable and satisfy:

$$\tilde{N}_1(s) \tilde{N}_1(s)^* + \tilde{D}_1(s) \tilde{D}_1(s)^* = Z_1(s) Z_1(s)^*.$$

Normalized coprime factors are key to relating the time-domain gap metric to the frequency domain. Equation 4.8 will simplify the gap metric later to enable use with flight data.

The gap metric is expressed in the frequency domain as follows:

$$\vec{\delta}(P_1, P_2) = \inf_{Q \in \mathcal{H}_\infty} \| G_1 - G_2 Q \|_\infty, \quad (4.9)$$

which amounts to minimizing the peak gain (i.e., the \mathcal{H}_∞ -norm) of a transfer function representing mismatch between the two systems. Here, $G_1(s) = [D_1(s); N_1(s)]$ and $G_2(s) = [D_2(s); N_2(s)]$ are the normalized coprime factors stacked into column vectors. $Q(s)$ is the decision variable and restricted to be a stable transfer function. $G_1(s)$ and $G_2(s)$ have interpretations as “graph” operators representing the inputs and outputs of $P_1(s)$ and $P_2(s)$, thus linking the time-domain gap metric to the frequency domain. In fact, $Q(s)$ establishes a correspondence between the inputs in order to be consistent with the optimization in Equation 4.4. For more details on graph operators, see [79, 87]. Although the optimization in Equation 4.9 is convex and can therefore be solved efficiently, it cannot be used for model validation with flight data in its current form. Note that Equation 4.9 requires coprime factorizations, which are not directly available to engineers.

The ν -gap metric is a far reaching modification to the gap metric that connects the analysis to frequency responses [75, 87]. This is key for applications to model validation because frequency responses are easily obtained from flight data. Again, the mathematical details on the modification are beyond the scope of this work [87]. Briefly, the ν -gap relaxes the optimization constraint in Equation 4.9 from $Q(s) \in \mathcal{H}_\infty$ to allow $Q(s)$ to be an unstable transfer function. However, a new technical condition

is needed to counter an unintended side effect on the norm topology. This condition requires that $Q(s)$ has an equal number of unstable poles and zeros (referred to as a “winding number condition”). Accordingly, define \mathcal{S} as the set of all transfer functions that satisfy this condition. If $Q(s)$ is not in \mathcal{S} , then the ν -gap metric defaults to its worst-case value of one.

For conventional, fixed-wing, rigid-body aircraft, it can be assumed that the $Q(s)$ found by the converged optimization will be in \mathcal{S} . The value of $Q(s)$ is related to the flight dynamics of the aircraft, which are easily controlled (i.e. there are no “approximate” unstable pole-zero cancellations, which would affect the winding number and have a detrimental effect on robustness [87]). Moreover, typical aircraft models are reasonably representative of the true flight dynamics. It can therefore be assumed that the ν -gap is less than its worst-case value of one, which implies that $Q(s) \in \mathcal{S}$. This is assumed in the sequel for numerically calculated values of the metric. Formally, the ν -gap is expressed as:

$$\delta_\nu(P_1, P_2) = \inf_{Q \in \mathcal{S}} \| G_1 - G_2 Q \|_\infty, \quad (4.10)$$

and there is no distinction between directed formulas.

To explain the connection of the ν -gap to frequency responses, let $G_2^*(s)$ be the complex-conjugate transpose of $G_2(s)$. Recall that $G_2(s)$ contains normalized coprime factors, and by Equation 4.8, $G_2^*(s)G_2(s) = 1$. Also let $\tilde{G}_2(s) = [-N_2(s), D_2(s)]$ for which $\tilde{G}_2(s)G_2(s) = 0$. Premultiply the term $G_1 - G_2 Q$ in Equation 4.10 by $[G_2^*; \tilde{G}_2]$, which is a unitary matrix and therefore does not affect the magnitude of the norm:

$$\delta_\nu(P_1, P_2) = \inf_{Q \in \mathcal{S}} \left\| \begin{bmatrix} G_2^* G_1 - G_2^* G_2 Q \\ \tilde{G}_2 G_1 - \tilde{G}_2 G_2 Q \end{bmatrix} \right\|_\infty, \quad (4.11)$$

$$= \inf_{Q \in \mathcal{S}} \left\| \begin{bmatrix} G_2^* G_1 - Q \\ \tilde{G}_2 G_1 \end{bmatrix} \right\|_\infty. \quad (4.12)$$

The value of $\delta_\nu(P_1, P_2)$ turns out to be $\left\| \tilde{G}_2 G_1 \right\|_\infty$ since, by assumption, $G_2^* G_1 \in \mathcal{S}$ and therefore Q can be taken as $Q = G_2^* G_1$. Note that Q is eliminated from the optimization and the solution is an \mathcal{H}_∞ -norm. However, the expression still depends on normalized coprime factors. Reference 87 shows (cf. 63, 69) how the ν -gap metric

can be expressed directly in terms of transfer functions P_1 and P_2 :

$$\delta_\nu(P_1, P_2) = \left\| (1 + P_2 P_2^*)^{-\frac{1}{2}} (P_2 - P_1) (1 + P_1 P_1^*)^{-\frac{1}{2}} \right\|_\infty \quad (4.13)$$

The expression in Equation 4.13 is derived by expanding $\tilde{G}_2 G_1$, manipulating terms, and using the property defined in Equation 4.8:

$$\begin{aligned} \tilde{G}_2 G_1 &= \begin{bmatrix} N_2 & -D_2 \end{bmatrix} \begin{bmatrix} D_1 \\ N_1 \end{bmatrix} \\ &= N_2 D_1 - D_2 N_1 \\ &= D_2 (P_2 - P_1) D_1 \\ &= \left(\frac{D_2 D_2^*}{D_2 D_2^* + N_2 N_2^*} \right)^{\frac{1}{2}} (P_2 - P_1) \left(\frac{D_1 D_1^*}{D_1 D_1^* + N_1 N_1^*} \right)^{\frac{1}{2}} \\ &= (1 + P_2 P_2^*)^{-\frac{1}{2}} (P_2 - P_1) (1 + P_1 P_1^*)^{-\frac{1}{2}} \end{aligned}$$

Computing the ν -gap metric in this way is very convenient from an engineering perspective. The formula depends only on the transfer functions $P_1(s)$ and $P_2(s)$. Hence, this approach can be applied to flight data using frequency responses. Discretizing over frequency and plugging in $P_1(j\omega)$ and $P_2(j\omega)$ into Equation 4.13, the ν -gap metric corresponds to the peak magnitude of the norm over all frequency points.

4.2 A Framework for Model Validation

The ν -gap metric is useful in validating models because it links the validation metric to classical measures of robustness [76, 77]. As a result, it provides more insight to aerospace engineers than current validation metrics, such as the TIC, with regard to the practical meaning of the computed value. At the same time, it retains a similarity to the TIC via the gap metric and its time domain interpretation. For model validation, the computed value of the ν -gap metric is related to an uncertainty description that accounts for the difference between the identified aircraft model and flight data. The resulting uncertain model is analyzed using robust control techniques to derive stability margins for control. The relationship between the validation metric and robustness requirements is a major advantage of the ν -gap metric. However, certain assumptions are required on the structure of the uncertainty description.

Consider $P_1(s)$ as the identified aircraft model and $P_2(s)$ as the true aircraft dynamics. It is assumed that $P_2(s) \in P_\Delta(s)$. The family of models described by $P_\Delta(s)$ is defined as follows:

$$P_\Delta(s) := \{ P_1(s)(1 + \delta_1)/(1 - \delta_2) \}, \quad (4.14)$$

where δ_1 and δ_2 are complex numbers representing the uncertainty. The values for δ_i can depend on frequency and be thought of as weighting functions characterizing the uncertainty. This is a type of uncertainty model commonly used in robust control applications. If $|\delta_1| \leq \epsilon < 1$ and $|\delta_2| \leq \epsilon$, then the ν -gap between $P_1(s)$ and $P_2(s)$ is bounded by $\delta_\nu(P_1(s), P_2(s)) < \epsilon$. Hence, the model validation metric is bounded by the size of the uncertainty.

The model $P_\Delta(s)$, evaluated for all frequencies, forms a disk in the complex plane. Thus, at each frequency, the perturbation required to generate $P_2(s)$ based on $P_1(s)$ is contained in this disk. Therefore, stability margins can be guaranteed for a controller operating on $P_2(s)$ that was originally designed based on $P_1(s)$. If the stability margins for the controller are sufficiently large and exceed the worst-case perturbation due to the uncertainty in $P_\Delta(s)$, then the closed-loop control is guaranteed to stabilize the real aircraft. This has profound implications on model-based flight control design. Controllers designed for an aircraft model, regardless of its accuracy, can be guaranteed to be stable in closed-loop when applied on the real system. As a result, they can be safely implemented and flight tested earlier in the development program. The relationship between the ν -gap and classical stability margins is given by the following [76]:

$$\text{Gain margin} = 20 \log_{10} \frac{1 + \epsilon}{1 - \epsilon}, \quad (4.15)$$

$$\text{Phase margin} = 2 \arcsin \epsilon, \quad (4.16)$$

$$\text{Disk margin} = \frac{2\epsilon}{1 - \epsilon^2}. \quad (4.17)$$

These margins interpret the validation metric as a robustness requirement for control. For example, $\epsilon = 0.38$ is equivalent to standard stability margins of 6 dB gain margin and 45 degrees phase margin. If the validation result returns an ϵ value less than 0.38, then a controller with standard robustness margins is guaranteed to be stable on the real aircraft.

4.2.1 A Simple Example

Consider a simple example of the proposed model validation framework, where a nominal aircraft model $P_1(s)$ is given by the following transfer function:

$$P_1(s) = \frac{18.75s + 225}{s^2 + 9s + 225}. \quad (4.18)$$

This transfer function has a damping ratio of 0.3, a natural frequency of 15 rad/s, and a stable zero at 12 rad/s. This $P_1(s)$ is representative of short-period longitudinal dynamics of a small, fixed-wing aircraft. Assume now that the true system is a perturbation of $P_1(s)$. Uncertainty and errors are expected in the system gain, natural frequency, and damping ratio estimates. For simplicity and without loss of generality, uncertainty in the location of the zero is neglected in this example. Let the true aircraft dynamics be represented by:

$$P_2(s) = \frac{18.75s + 225}{s^2 + 7.22s + 246.5}. \quad (4.19)$$

This transfer function has a damping ratio of 0.23 and natural frequency of 15.7 rad/s. Note that real aircraft dynamics are not described by SISO LTI transfer functions.

Given exact knowledge of $P_1(s)$ and $P_2(s)$, the gap and the ν -gap metrics can be computed directly. Both metrics have the same value of 0.09. It is often the case that values of the gap and the ν -gap are identical. However, this computed value cannot be used directly to infer stability margins. The perturbation between $P_1(s)$ and $P_2(s)$ is not in the assumed coprime factor uncertainty structure required by $P_\Delta(s)$. Therefore, the true aircraft dynamics embodied in $P_2(s)$ must be overbounded by a coprime factor uncertainty model, for which an ϵ value of 0.14 is sufficient. This ensures the true perturbations are smaller and covered by the coprime factor uncertainty disk associated with $P_\Delta(s)$.

Figure 4.1 on the right shows the ϵ -disk as a solid circle in the complex plane. Multiplicative perturbations associated with the difference between $P_1(j\omega)$ and $P_2(j\omega)$, i.e. $P_1(j\omega)/P_2(j\omega)$, are shown for a set of discrete frequency points. The perturbations at 0.1, 15, and 100 rad/s are highlighted. Note that all perturbations are covered by the ϵ -disk. The low frequency perturbations are all near 0.91 on the real axis. The two systems are equivalent at high frequency, resulting in perturbations near 1 on the real axis. Bode plots of $P_1(j\omega)$ and $P_2(j\omega)$ are shown on the left.

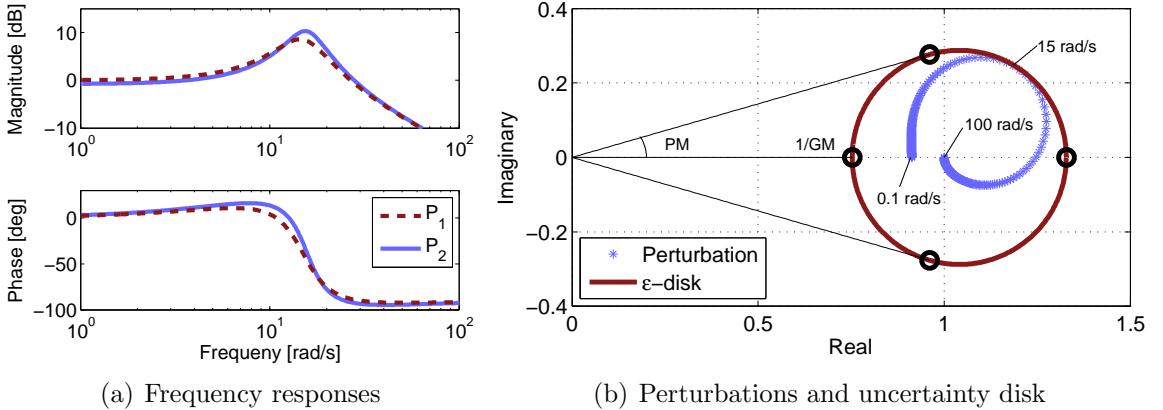


Figure 4.1: Comparison of the model $P_1(s)$ and the true aircraft dynamics $P_2(s)$.

Equations (4.15-4.17) have geometric interpretations using the ϵ -disk. The upper gain margin is given by the marker on the real axis at 1.33, which corresponds to +2.48 dB. Accordingly, the marker on the real axis at 0.75 corresponds to -2.48 dB gain margin. The phase margin is equivalent to the angle of the line tangent to the ϵ -disk that passes through the origin. The markers away from the real axis indicate the tangent points. Here, the phase margin is 16.2 degrees. Although the results are conservative due to the necessary overbounding, stability for the closed-loop system is guaranteed by these robustness requirements.

Robustness requirements derived from the model validation analysis can be examined further using Nichols and Nyquist diagrams. Figure 4.2 on the left shows the Nichols diagram. Gain and phase variations associated with the ϵ -disk form an elliptical region in the Nichols plane. This ellipse is inscribed in a traditional Nichols exclusion region marked by the dashed polygon. The Nichols exclusion region is a useful way to visualize the relationship between the validation analysis and robustness requirements. The elliptical region shows combinations of gain and phase variations that could exist on the real aircraft. Therefore, if a controller satisfies robustness requirements outside the Nichols exclusion region, then the closed-loop system will be stable. Note that the worst-case perturbation near 15 rad/s is on the boundary of the ellipse and inside the Nichols exclusion region.

A robust Nyquist diagram is shown on the right in Figure 4.2. In this case, the uncertainty associated with the ϵ -disk forms a tube around $P_1(s)$ that contains $P_2(s)$. The worst-case perturbation near 15 rad/s is accented with the circular markers. At

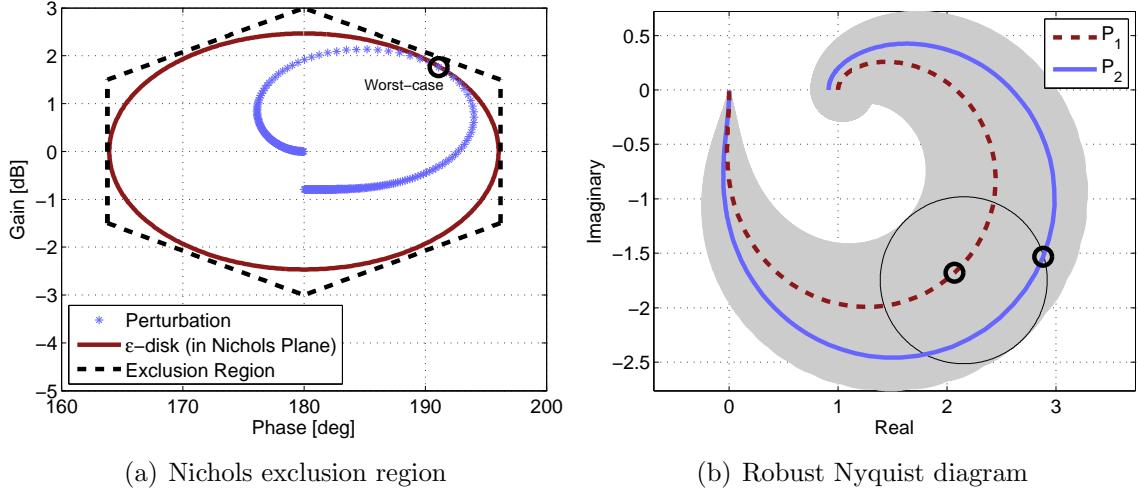


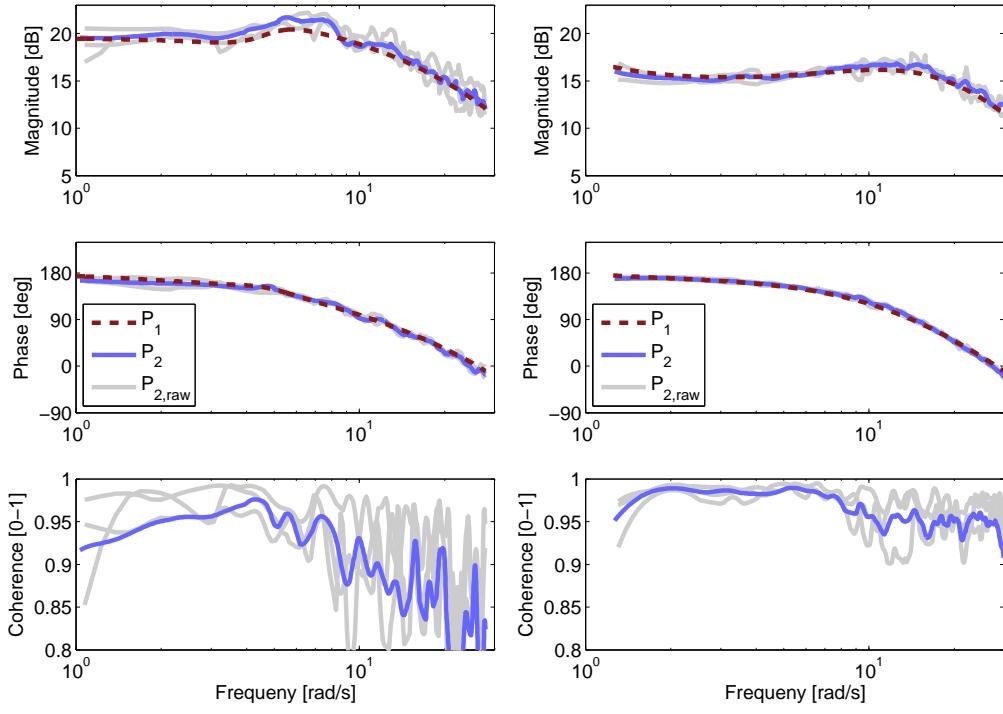
Figure 4.2: Guaranteed robustness margins for control.

this frequency, the ϵ -disk exactly captures $P_2(s)$. Nyquist analysis caters directly to robust control theory. It is important that the shaded region does not cross the critical point at -1 on the real axis, as this would prohibit a stability guarantee for the closed-loop system.

4.3 Flight Data Results

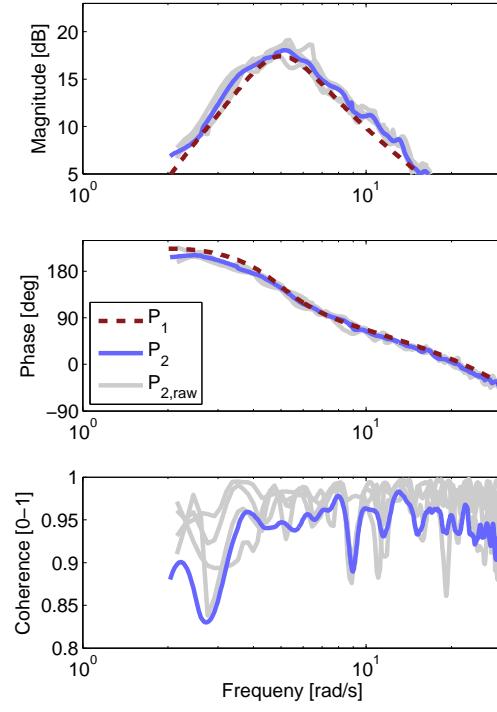
The proposed model validation framework is demonstrated for its intended application: evaluating the accuracy of aircraft models with respect to experimental flight data. A linear model based on constant aerodynamic coefficients was generated using frequency-domain system identification techniques in the previous chapter. The accuracy of this model is verified using the proposed validation framework. In order to be consistent with the previous example, consider $P_1(j\omega)$ as the linearized model frequency response to be validated. Let $P_2(j\omega)$ represent the experimental flight data in the form of a frequency response. The model $P_1(j\omega)$ and flight data $P_2(j\omega)$ are shown as Bode plots in Figure 4.3. Coherence plots are included below to indicate the quality and accuracy of the frequency responses.

The lightly shaded frequency responses represent the raw data from individual sinusoidal sweep experiments. In general, data obtained from a single experiment would be sufficient to apply the proposed model validation framework. However, frequency responses based on single experiments often exhibit significant random error. In Fig-



(a) Aileron to roll rate

(b) Elevator to pitch rate



(c) Rudder to yaw rate

Figure 4.3: Experimental frequency responses and linear model.

ure 4.3, for example, the roll rate responses are noisy, particularly at high frequency. Similarly, the yaw rate responses are noisy near the peak of the magnitude. Errors like this contribute directly to more conservative robustness requirements in the validation analysis. This is a direct consequence of higher levels of uncertainty required to overbound the difference between $P_1(j\omega)$ and $P_2(j\omega)$.

The handling of frequency response errors is an engineering decision that depends on the application. In this example, the individual frequency responses are averaged to reduce the effect of random error. The darker solid line in Figure 4.3 shows this result. Where the individual roll rate responses are noisy, the average is smooth. A key feature of the proposed validation framework, however, is that it can be applied regardless of data quality. This is particularly beneficial in cases where model accuracy is not crucial and controllers are designed with large stability margins. In these cases, robust stability is guaranteed despite poor data quality. This can significantly reduce the scope of a flight test campaign, which is critical in many UAS applications.

Figure 4.4 shows the model validation result for the primary input-output relationships on the aircraft. The lightly shaded markers represent the raw data from individual experiments. Recall that each marker corresponds to the difference between $P_1(j\omega)$ and $P_2(j\omega)$ at a single frequency point. The markers are also related to the lightly shaded frequency responses in Figure 4.3. The lightly shaded ϵ -disk indicates the main validation result. It shows the amount of uncertainty necessary to account for the perturbations associated with the raw data.

The darker markers in Figure 4.4 represent the averaged flight data. These markers also correspond to the darker frequency responses in Figure 4.3. The darker ϵ -disk indicates the necessary amount of uncertainty necessary to account for the perturbations associated with the averaged data. Note that the darker ϵ -disk is significantly smaller than its lighter counterpart. This confirms that averaging the flight data yielded a less conservative uncertainty description. As a result, less stringent robustness requirements have to be satisfied by the controller.

The ϵ -disks in Figure 4.4 indicate that the elevator to pitch rate model has the highest quality, and that the rudder to yaw model has the lowest quality. In all three cases, averaging the experimental data reduced the amount of uncertainty necessary to account for the perturbations. Averaging also reduced the random error manifested as scatter, which is indicative of a smoother experimental frequency response. This

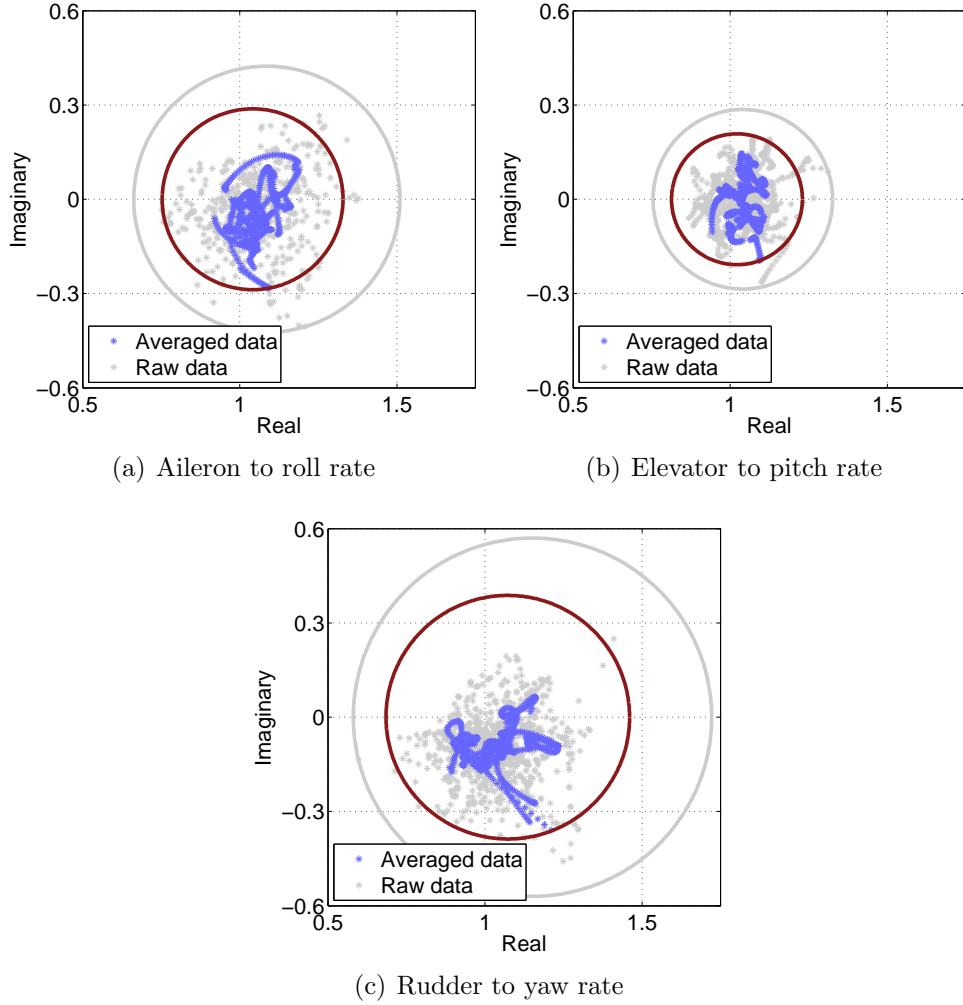


Figure 4.4: Perturbations and uncertainty disks.

is particularly noticeable in the roll and yaw rate responses, where the averaged data is qualitatively less scattered and more continuous than the raw data. Table 4.1 provides a summary of the quantitative model validation results.

Table 4.1: Validation results for raw and averaged flight data.

		Roll Rate		Pitch Rate		Yaw Rate	
		Raw	Avg.	Raw	Avg.	Raw	Avg.
Model Quality	Gap Metric ϵ	0.20	0.14	0.14	0.10	0.27	0.19
Controller Requirements	Gain Margin [dB]	3.58	2.47	2.45	1.80	4.70	3.29
	Phase Margin [deg]	23.43	16.21	16.10	11.82	30.73	21.29
	Disk Margin	0.42	0.28	0.29	0.21	0.57	0.39

The results in Table 4.1 show how the model validation metric ϵ is related to robust-

ness requirements for closed-loop control. Results are shown both for the raw and the averaged data. In comparison to the others, the higher quality of the elevator to pitch rate model is evident from the lower stability margins required for the control system. For the averaged data case, as an example, only about 12 degrees of phase margin are required for a controller to be stable. In the rudder to yaw rate model, however, about 21 degrees of phase margin are required. Table 4.1 also shows that a less conservative controller can be implemented if the averaged data is used in the analysis. This assumption is typically valid in conventional aerospace applications because, in general, it only reduces the adverse effects of turbulence and sensor noise on the flight data. Regardless of the data quality, however, robust stability requirements are obtained from the validation analysis.

The final step in the validation process is to verify that the uncertainty model, which was determined in the frequency domain, translates appropriately into the time domain. A Monte Carlo simulation is executed using input signals recorded from an open-loop doublet flight maneuver. The uncertainty is sampled along the boundary of the ϵ -disks and realized as all-pass transfer functions perturbing the nominal model. For this simulation, ϵ -disks computed based on the averaged flight data are used. The uncertainty sampling ensures that all possible perturbations in the flight dynamics are represented in the Monte Carlo simulation. For the flight test, a pitch doublet was executed first, followed by a roll doublet, and completed with a yaw doublet. Figure 4.5 shows the time-domain verification results by comparing flight data to the Monte Carlo simulations. The identified linear state-space model of the aircraft dynamics was used in order to account for the cross-coupling between the lateral and the directional axes.

Control surface input signals are also included in Figure 4.5. These input signals are measured in degrees, but scaled 5 times in the plot to match the system gain and the order of magnitude of the outputs. The linear aircraft model is represented by the thick dashed line. The flight data is shown by the thick solid line. A lightly shaded tube shows the collection of 500 individual Monte Carlo simulations. This tube corresponds to the family of responses described by the uncertainty model.

The time-domain simulation results provide complementary insights to the frequency-domain analysis. The flight data matches the roll and pitch rate simulation results closely. However, the validation results in Figure 4.4 and Table 4.1 predict the roll

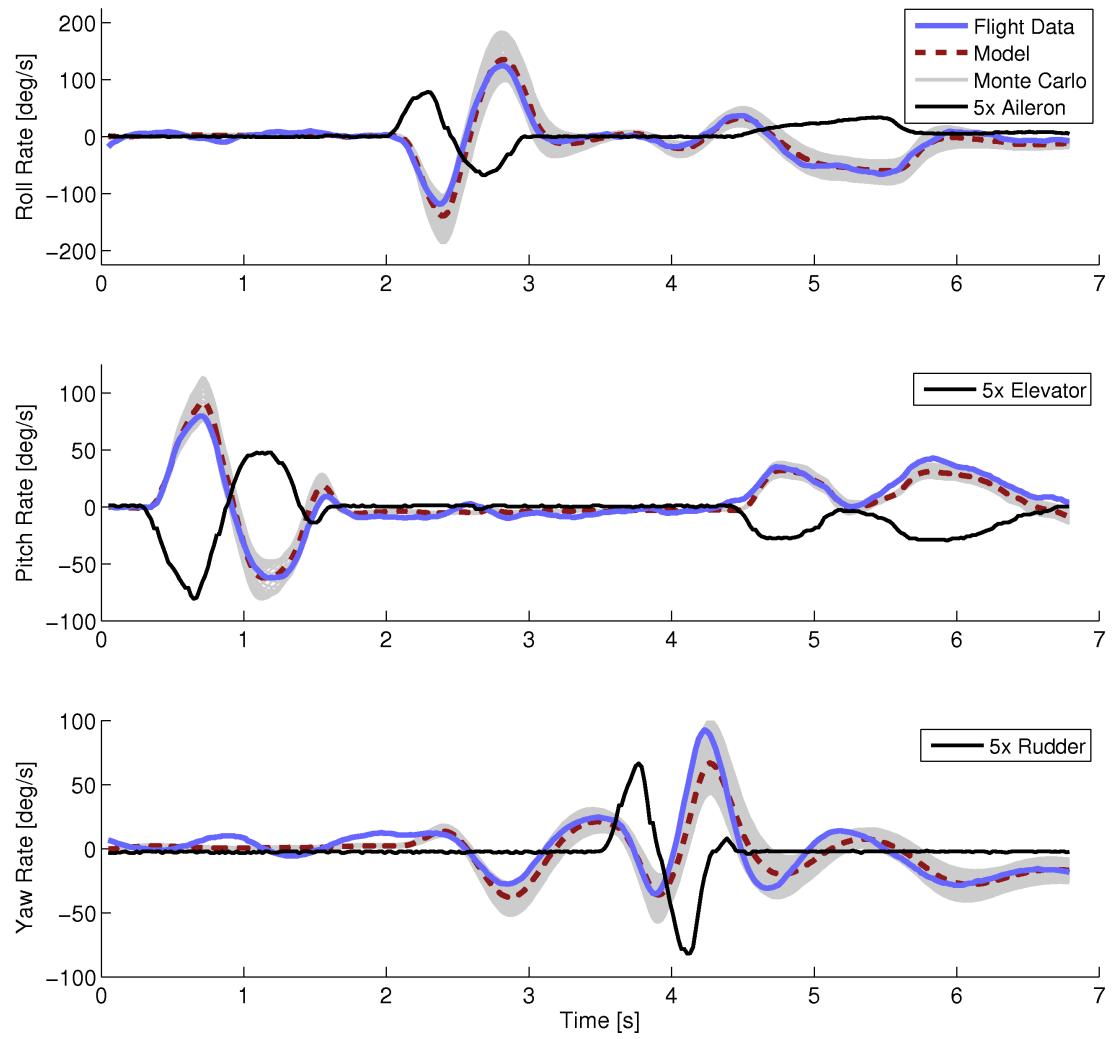


Figure 4.5: Time domain simulation of uncertainty model versus flight data.

model to be significantly less accurate than the pitch model. The discrepancy points to poor quality in the roll rate frequency response, which is related to lowered coherence at high frequency. In this case, although a good model was identified, the poor quality of the frequency response led to overly conservative robustness requirements. The pitch rate model represents the ideal case, where the frequency-domain analysis correctly predicts a close match in the time-domain simulation. The yaw rate uncertainty model successfully accounts for the significant deviation noted with respect to the flight data. In this case, a relatively poor model was identified due to fitting error on a high quality frequency response.

Chapter 5

Baseline Control Design

A simple baseline controller is designed as a benchmark for the UAS test platform. This controller was designed and validated through simulation and flight tests. It serves as the standard for any research experiment that requires closed-loop control. The design is two-tiered: an inner-loop attitude controller and an outer-loop flight management system. The inner-loop controller tracks desired pitch and roll angles of the aircraft while damping out oscillations present in the open-loop dynamics. The outer-loop controller maintains desired altitude, airspeed, and heading direction.

5.1 Inner-Loop Controller

The inner-loop attitude controller tracks desired pitch and roll angles while damping oscillations in the open-loop dynamics. The desired rise times are 1 second, with overshoot less than 5%. Standard robustness of ± 6 dB gain and ± 45 degrees phase margin are satisfied. This is sufficient to comply with robustness requirements obtained from the model validation analysis. For the control design, the nonlinear simulation model is trimmed for level flight at 19 m/s and linearized. The linear dynamics are decoupled into longitudinal and lateral/directional sub-systems, and inner-loop controllers are designed separately.

The longitudinal controller is shown in Figure 5.1, where AC_{lon} is the linearized aircraft model. The inputs to the model are throttle setting (δ_{thr}) and elevator deflection (δ_{ele}). The outputs used in feedback are pitch angle (θ) and pitch rate (q). A proportional gain pitch rate damper (K_q) is applied to increase damping. The pitch angle

tracking controller (K_θ) uses proportional-integral gain. Integrator anti-windup logic (A/W) is implemented to handle actuator saturation. The inputs to the closed-loop system are pitch angle reference (θ_{ref}) and δ_{thr} . The throttle not controlled.

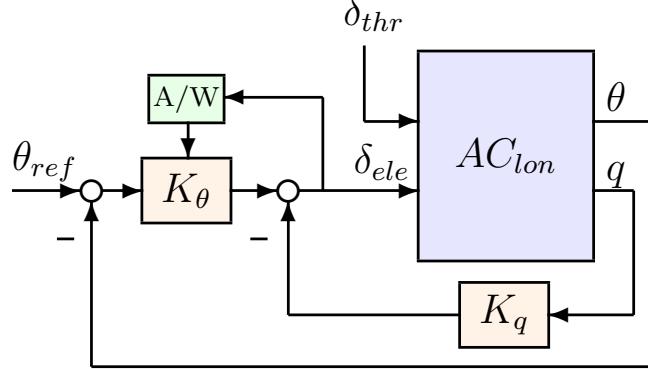


Figure 5.1: Longitudinal dynamics control architecture.

The lateral/directional controller is shown in Figure 5.2, where $AC_{lat/dir}$ is the linearized aircraft model. The inputs to the model are aileron (δ_{ail}) and rudder (δ_{rud}) deflections. The outputs used in feedback are roll angle (ϕ), roll rate (p), and yaw rate (r). A proportional gain roll rate damper (K_p) is applied to reject disturbances in turbulent conditions. A proportional gain yaw damper (K_r) is implemented to increase damping in the Dutch roll mode. A washout filter is also required to avoid an adverse yaw effect during turns. The roll angle tracking controller (K_ϕ) uses proportional-integral gain and A/W logic handles actuator saturation. The resulting input to the closed-loop system is roll angle reference (ϕ_{ref}).

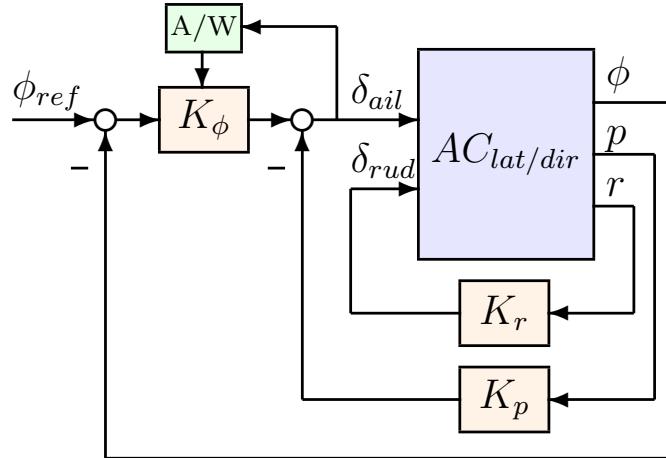
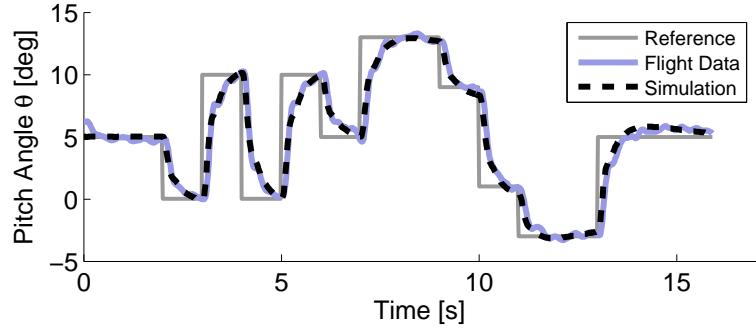
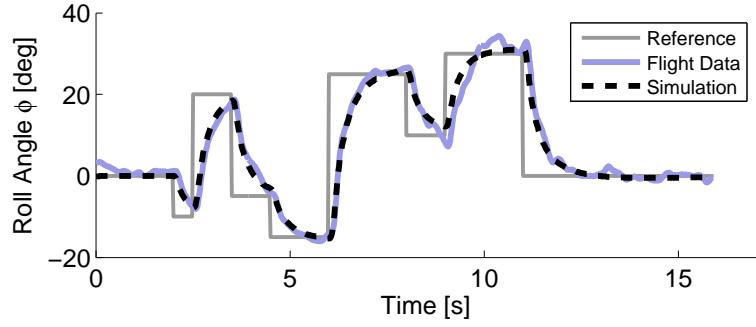


Figure 5.2: Lateral/directional dynamics control architecture.

Performance of the combined inner-loop controller was validated using simulation and flight tests. The test scenario was a step reference pattern of different amplitudes and lengths. Pitch axis commands were applied independently from roll axis commands. This type of pattern was used to excite the closed-loop dynamics over a broad frequency range. Simulation and flight tests validated that the controller, which was designed using a linear model, performed as expected in the simulation and in flight. Figure 5.3 shows the combined flight test and SIL simulation results for Thor.



(a) Pitch angle tracking pattern



(b) Roll angle tracking pattern

Figure 5.3: Inner-loop control flight test vs. SIL simulation.

The flight test results confirm that the inner-loop attitude tracking controller objectives are satisfied. The rise times for the pitch and roll axes are around 1 second. There is 5% overshoot in the pitch response, and no overshoot in the roll response. Strong agreement between the flight data and the simulation result affirms the accuracy of the aircraft model and the reliability of the controller.

5.2 Outer-Loop Controller

The outer-loop controller is a flight management system that tracks altitude, airspeed, and ground track angle. Its characteristics include no overshoot on the ground track angle, and maintaining altitude and airspeed within ± 5 m and ± 2 m/s, respectively. The architecture is shown in Figure 5.4.

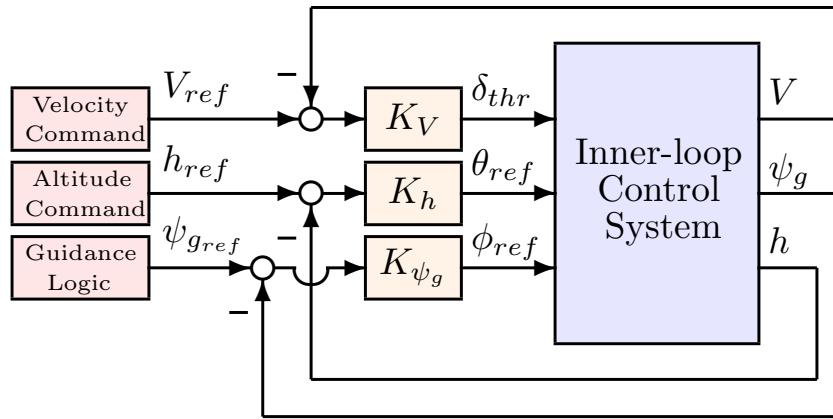


Figure 5.4: Aircraft outer-loop control architecture.

The inputs to the inner-loop control system are throttle setting (δ_{thr}), pitch angle reference (θ_{ref}), and roll angle reference (ϕ_{ref}). The outputs utilized for outer-loop control are indicated airspeed (V), altitude (h), and ground track angle (ψ_g). The ground track angle is defined as $\psi_g = \arctan(v_e/v_n)$, where v_e and v_n are the east and north velocities estimated by the navigation filter.

The three guidance blocks on the left provide the flight management system with commands V_{ref} , h_{ref} , and ψ_{gref} . The altitude controller (K_h) produces a pitch angle reference command, and the airspeed controller (K_V) produces a throttle command. Although non-ideal for engine-out scenarios [11], this architecture was selected for simplicity. Both K_h and K_V use proportional-integral control and implement integrator A/W logic (not shown in Figure 5.4) in order to safely limit the commands provided to the inner-loop control system. The throttle command is constrained between 0 and 1, and the pitch angle reference is constrained to $\pm 20^\circ$. The ground track angle controller (K_{ψ_g}) uses proportional gain, and, hence, the roll angle reference ϕ_{ref} can be constrained directly at $\pm 45^\circ$. This limiting is required to prevent the aircraft from rolling over due to large ground track angle step commands.

A flight test was conducted with Thor to verify the performance of the flight man-

agement system. The test consisted of a series of 90 degree ground track angle step commands. In the absence of wind, this should result in a square pattern. In the presence of wind, the airplane flies a rectangular pattern that drifts in the direction of the wind. Figure 5.5 shows a comparison between flight data and SIL simulation of the ground track angle, airspeed, and altitude signals throughout the maneuver.

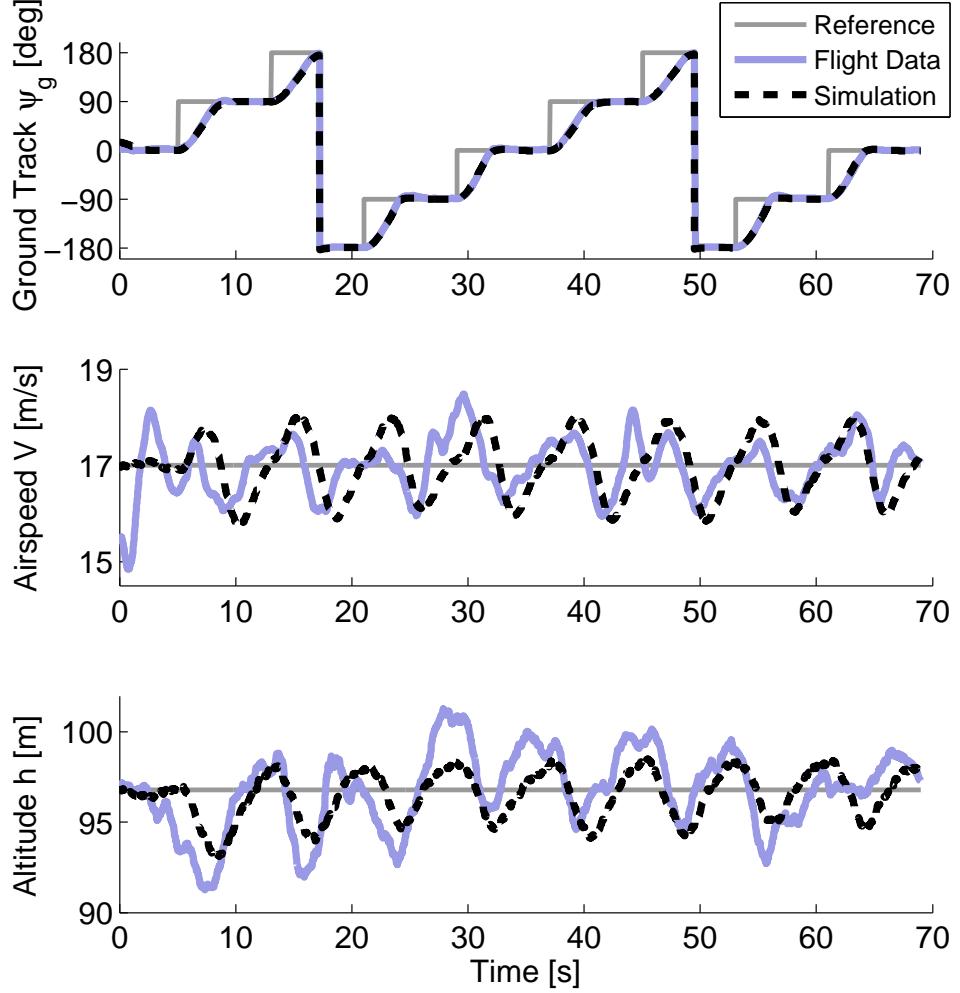


Figure 5.5: Validation of outer-loop flight management system.

The simulation and flight test results confirm that the outer-loop flight management system objectives are satisfied. The ground track angle is followed by the aircraft, while altitude is held within ± 5 m, and airspeed within ± 1.5 m/s. Figure 5.6 shows a visualization of the aircraft trajectory across a satellite image of the airfield.

The rectangular trajectory begins at the origin in Figure 5.6. Using the flight data,

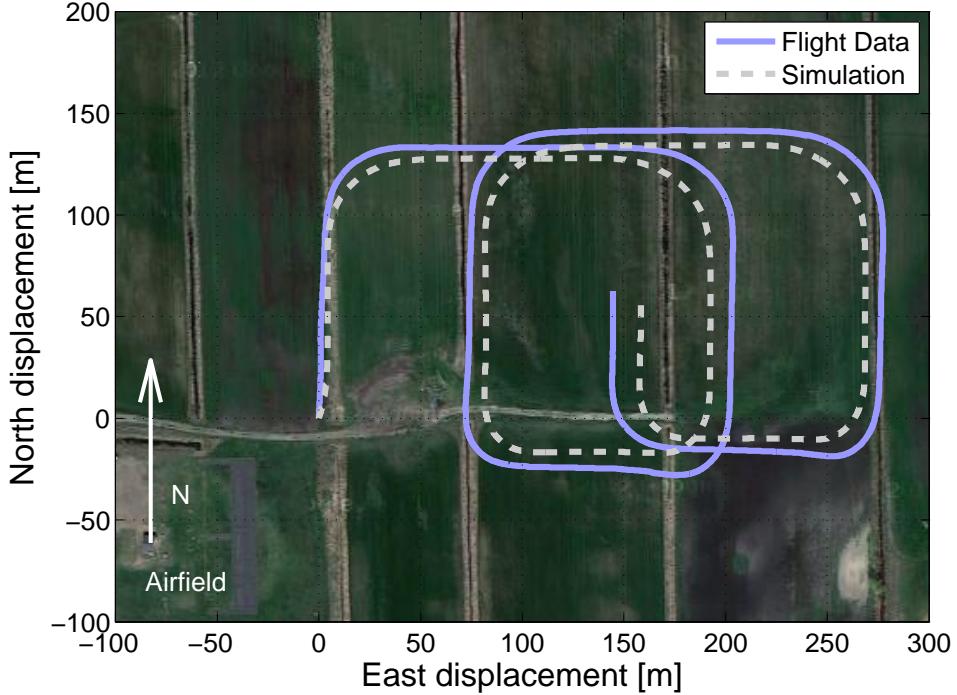


Figure 5.6: Satellite view of flight management system validation results.

an estimate of wind direction and speed is obtained by examining the drift. The estimated wind components are applied in the nonlinear simulation, which shows strong agreement with the flight test.

5.3 Waypoint Guidance

A GPS waypoint guidance algorithm was implemented and flight tested. The guidance logic was originally developed by researchers at the Budapest University of Technology and Economics in Hungary [88, 89] and later ported to the UMN UAS test platform. After completing modifications required for integration, a set of demonstration flight tests were carried out using Thor to validate the performance of the algorithm.

The waypoint guidance logic ensures the reachability of a GPS waypoint by operating in two modes that account for the closed-loop aircraft dynamics. Mode 1 is used when the target GPS waypoint is within the line-of-sight of the aircraft. Line-of-sight is a tuning parameter in the guidance logic, which in this case was defined as $\pm 20^\circ$ relative to the nose of the vehicle. While operating in Mode 1, the baseline flight

control system is commanded to follow a ground track vector pointed directly at the target GPS waypoint.

Mode 2 is used when the target GPS waypoint is outside the line-of-sight of the aircraft. The worst-case scenario occurs when the next waypoint is near and also behind the aircraft. Mode 2 logic checks for this scenario first. If the waypoint is not reachable with respect to the closed-loop turning radius of the aircraft (about 40 m for the Ultra Stick 25e), the aircraft continues in forward flight until reachability is achieved. When the waypoint becomes reachable, Mode 2 logic commands the aircraft to turn. As soon as the waypoint enters the line-of-sight of the aircraft, Mode 2 is terminated and the logic switches to Mode 1. A waypoint is considered to be captured when the aircraft flies within a 20 m safety zone of the target.

Figure 5.7 shows the flight test result from a 4 GPS waypoint pattern, arranged in a counter-clockwise direction at the corners of a 100x100 m square. The 20 m safety zones around each waypoint are also shown. The guidance algorithm is initiated at the origin on the map (lower left waypoint), with the aircraft pointing due east. Hence, the initial trajectory towards the lower right waypoint differs from the steady-state trajectory. The flight test result shown represents 4 loops around the pattern.

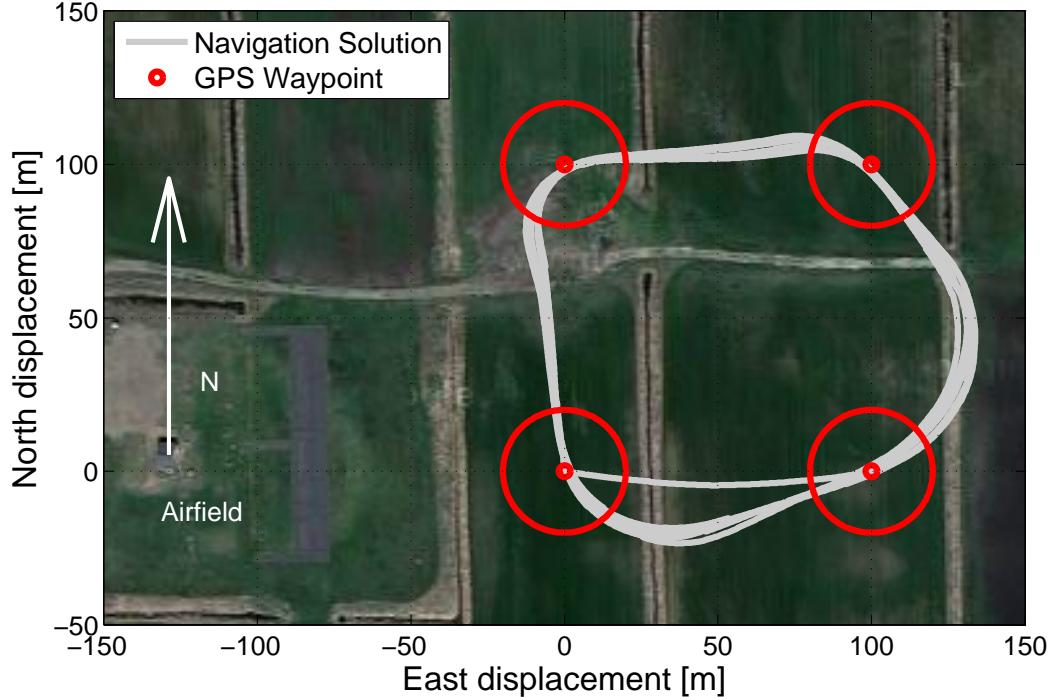


Figure 5.7: GPS waypoint guidance flight test result.

The flight trajectory in Figure 5.7 is distorted by the steady winds that prevailed during the test. Without the presence of wind, the flight trajectory would have been symmetric about the center of the pattern. The flight test results suggest that a 100 m distance between GPS waypoints is near the limit of how aggressively the baseline flight controller can perform.

Chapter 6

Conclusion

It is important for the aerospace community to continue developing flexible and efficient platforms that support testing and validation of new flight control algorithms. The University of Minnesota operates such a platform with advanced research capabilities in the areas of control, guidance, navigation, and fault detection. Collaborating with researchers, using open-source software, and flight testing with these platforms expedites the development and application of new theory that could one day revolutionize aerospace technology.

This thesis has provided an overview of the test platform developed at the University of Minnesota. The primary purpose of this work is to facilitate the expansion and creation of new test platforms dedicated to model-based flight research. These platforms are key to studying and maturing model-based approaches to flight control design and certification. In turn, model-based design and certification techniques are set to play a crucial role in the integration of unmanned aircraft systems into the civilian airspace.

As a technical contribution, an efficient and practical approach to system identification and model validation was developed to support model-based flight control design. The system identification work examined some of the challenges faced by low-cost unmanned aircraft. Effects due to quantity and quality restrictions on sensing equipment were described, particularly on the ability to obtain an accurate model of the aircraft dynamics. The model validation work proposed a new metric for comparing models to flight data. This metric is very useful because it provides a connection between

model accuracy and robustness requirements for closed-loop control.

Overall, the proposed approach makes a step towards an overall more rigorous handling of model-based design for unmanned aircraft systems. Researchers in the aerospace community must continue building upon and extending this work. Unmanned aircraft technology will mature alongside techniques and requirements for model-based design, and it is critical for the community to support a seamless integration.

The main future direction of this research is to develop an integrated set of tools for model-based design and certification. Currently, the fusing of various tasks in the development process are only partially linked. With tighter integration, the approach to model-based design and certification can be improved. The integrated tools can then be used by aerospace engineers to efficiently guide new aircraft through the modeling, validation, and control design process.

Bibliography

- [1] *FAA Modernization and Reform Act of 2012*, Library of Congress, H.R.685.
- [2] Lowy, J., “FAA Certifies First 2 Drones For Domestic Flight,” *The Associated Press*, 2013.
- [3] Hennigan, W. J. and Vartabedian, R., “F-35 fighter jet struggles to take off,” *Los Angeles Times*, 2013.
- [4] Taylor, B., “University of Minnesota UAV Research Group,” Online at: www.uav.aem.umn.edu.
- [5] Dorobantu, A., Johnson, W., Lie, F. A., Taylor, B., Murch, A., Paw, Y. C., Gebre-Egziabher, D., and Balas, G., “An Airborne Experimental Test Platform: From Theory to Flight,” *American Control Conference*, 2013.
- [6] Dorobantu, A., Murch, A., Mettler, B., and Balas, G., “System Identification for a Small, Low-Cost, Unmanned Aircraft,” *Journal of Aircraft*, 2013.
- [7] Dorobantu, A., Balas, G., and Georgiou, T., “Validating Aircraft Models in the Gap Metric,” *Submitted to the Journal of Aircraft*, 2013.
- [8] Owens, D. B., Cox, D. E., and Morelli, E. A., “Development of a Low-Cost Sub-Scale Aircraft for Flight Research: The FASER Project,” *25th AIAA Aerodynamic Measurement Technology and Ground Testing Conference*, No. 2006-3306, San Francisco, CA, 2006.
- [9] Morelli, E. A. and DeLoach, R., “Wind Tunnel Database Development using Modern Experiment Design and Multivariate Orthogonal Functions,” *41st AIAA Aerospace Sciences Meeting and Exhibit*, No. 2002-2795, Reno, NV, 2002.

- [10] Hoe, G., Owens, B., and Denham, C., “Forced Oscillation Wind Tunnel Testing for FASER Flight Research Aircraft,” *AIAA Atmospheric Flight Mechanics Conference*, No. 2012-4645, Minneapolis, MN, 2012.
- [11] Paw, Y. C., *Synthesis and Validation of Flight Control for UAV*, Ph.D. thesis, University of Minnesota, 2009.
- [12] Murch, A. M., Paw, Y. C., Pandita, R., Li, Z., and Balas, G. J., “A Low Cost Small UAV Flight Research Facility,” *Advances in Aerospace Guidance, Navigation and Control*, Springer Berlin Heidelberg, 2011, pp. 29–40.
- [13] Phytec, “phyCORE MPC5200B-tiny microcontroller datasheet,” <http://www.phytec.com/products/sbc/PowerPC/phyCORE-MPC5200B-tiny.html>.
- [14] eCos, “Real-time Operating System,” <http://ecos.sourceforge.org/>.
- [15] AcroName Robotics, “RxMux failsafe manual/auto switch datasheet,” <http://www.acroname.com/robotics/parts/R306-RXMUX.html>.
- [16] FreeWave, “MM2-T 900MHz Modem datasheet,” <http://www.freewave.com/products/allproducts/mm2.aspx>.
- [17] AMSYS, “AMS 5812 pressure transducer datasheet,” http://www.amsys.info/sheets/amsys.en.ams5812_e.pdf.
- [18] “ADIS16405 Triaxial Inertial Sensor with Magnetometer,” Tech. rep., Analog Devices.
- [19] GPS Outfitters, “Titan 3 GPS antenna datasheet,” <http://www.gpsoutfitters.com/Titan3.html>.
- [20] Hemisphere GPS, “Crescent GPS receiver board datasheet,” https://www.navtechgps.com/Downloads/Crescent_OEM.pdf.
- [21] Goodrich Corporation, “Model 0858 Multi-Function Probes datasheet,” .
- [22] Kershner, D. D., “Miniature Flow-direction and Airspeed Sensor for Airplanes and Radio-Controlled Models in Spin Studies,” *NASA Technical Paper*, , No. 1467, 1979.

- [23] Google, Inc., “Satellite View of Airfield,” <https://maps.google.com/>.
- [24] The MathWorks, Inc., “MATLAB/Simulink,” r2010a,” <http://www.mathworks.com>.
- [25] The MathWorks, Inc., “Real Time Windows Target,” <http://www.mathworks.com/products/rtwt/>.
- [26] FlightGear, “Open-Source Flight Simulator,” <http://www.flightgear.org/>.
- [27] Gleason, S. and Gebre-Egziabher, D., *GNSS Applications and Methods*, Artech House, 2009.
- [28] Groves, P., *Principles of GNSS, Inertial, and Integrated Navigation Systems*, Artech House, 2008.
- [29] Farrell, J. and Barth, M., *The Global Positioning System and Inertial Navigation*, McGraw-Hill Professional, 1999.
- [30] Titterton, D. and Weston, J., *Strapdown Inertial Navigation Technology*, Institution of Engineering and Technology, 2004.
- [31] Simon, D., *Optimal State Estimation: Kalman, H_∞ , and Nonlinear Approaches*, Wiley, 2006.
- [32] Gebre-Egziabher, D., *Design and Performance Analysis of Low-Cost Aided Dead Reckoning Navigator*, Ph.D. thesis, Department of Aeronautics and Astronautics, Stanford University, December 2001.
- [33] Apache Subversion, “SVN repository,” <http://subversion.apache.org/>.
- [34] University of Minnesota, “UAV Research Group Subversion Repository,” <http://svn.umnaem.webfactional.com>.
- [35] Doxygen, “Automatic documentation generated from source code,” <http://www.doxygen.org>/.
- [36] Ljung, L., *System Identification: Theory for the User*, Prentice Hall, 2nd ed., 1999.
- [37] Soderstrom, T. and Stoica, P., *System Identification*, Prentice Hall, 2001.

- [38] Tischler, M. and Remple, R., *Aircraft and Rotorcraft System Identification*, AIAA, 2006.
- [39] Klein, V. and Morelli, E., *Aircraft System Identification: Theory and Practice*, AIAA, 2006.
- [40] Jategaonkar, R., *Flight Vehicle System Identification: A Time Domain Methodology*, AIAA, 2006.
- [41] Mettler, B., Tischler, M., and Kanade, T., “System Identification of a Small-Scale Unmanned Rotorcraft for Flight Control Design,” *Journal of the American Helicopter Society*, Vol. 47, No. 1, 2002.
- [42] Theodore, C., Tischler, M., and Colbourne, J., “Rapid Frequency-Domain Modeling Methods for Unmanned Aerial Vehicle Flight Control Applications,” *Journal of Aircraft*, Vol. 41, No. 4, 2004.
- [43] Dadkhah, N. and Mettler, B., “System Identification Modelling and Flight Characteristics Analysis of Miniature Co-Axial Helicopter,” *Accepted for Publication in Journal of the American Helicopter Society*, 2011.
- [44] Gremillion, G. and Humbert, J. S., “System Identification of a Quadrotor Micro Air Vehicle,” *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, Toronto, Canada, 2010.
- [45] Bachelder, E., Thompson, P., Klyde, D., and Alvarez, D., “A New System Identification Method Using Short Duration Flight Test Inputs,” *AIAA Atmospheric Flight Mechanics Conference*, Portland, OR, United States, 2011.
- [46] Morelli, E. and Smith, M., “Real-Time Dynamic Modeling: Data Information Requirements and Flight-Test Results,” *Journal of Aircraft*, Vol. 46, No. 6, 2009.
- [47] Morelli, E., “Low-Order Equivalent System Identification for the Tu-144LL Supersonic Transport Aircraft,” *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 2, 2003.
- [48] Cook, M., *Flight Dynamics Principles*, Elsevier Ltd., 2nd ed., 2007.
- [49] Hoak, D. E. et al., “The USAF Stability and Control Datcom,” Tech. Rep. AFWAL-TR-83-3048.

- [50] Owens, B., Cox, D., and Morelli, E., “Development of a Low-Cost Sub-Scale Aircraft for Flight Research: The FASER Project,” *25th AIAA Aerodynamic Measurement Technology and Ground Testing Conference*, San Francisco, CA, United States, 2006.
- [51] Morelli, E. and DeLoach, R., “Wind Tunnel Database Development Using Modern Experiment Design and Multivariate Orthogonal Functions,” *41st Aerospace Sciences Meeting and Exhibit*, Reno, NA, United States, 2003.
- [52] Bendat, J. and Piersol, A., *Engineering Applications of Correlation and Spectral Analysis*, John Wiley & Sons, 1993.
- [53] Dorobantu, A., Murch, A., Mettler, B., and Balas, G., “Frequency Domain System Identification for a Small, Low-Cost, Fixed-Wing UAV,” *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, Portland, Oregon, 2011.
- [54] Hodgkinson, J., “History of Low-Order Equivalent Systems for Aircraft Flying Qualities,” *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 4, 2005.
- [55] Smith, R. and Doyle, J., “Model Validation: A Connection Between Robust Control and Identification,” *IEEE Transactions on Automatic Control*, Vol. 37, No. 7, 1992, pp. 942–952.
- [56] Poolla, K., Khargonekar, P., Tikku, A., Krause, J., and Nagpal, K., “A Time-Domain Approach to Model Validation,” *IEEE Transactions on Automatic Control*, Vol. 39, No. 4, 1994, pp. 951–959.
- [57] Smith, R., Dullerud, G., Rangan, S., and Poolla, K., “Model Validation for Dynamically Uncertain Systems,” *Mathematical Modeling of Systems*, Vol. 3, No. 1, 1997, pp. 43–58.
- [58] Rangan, S. and Poolla, K., “Time-Domain Validation for Sample-Data Uncertainty Models,” *IEEE Transactions on Automatic Control*, Vol. 41, No. 7, 1996, pp. 980–991.
- [59] Dullerud, G. and Smith, R., “Sampled-Data Model Validation: An Algorithm and Experimental Application,” *International Journal of Robust and Nonlinear Control*, Vol. 6, 1996, pp. 1065–1078.

- [60] Theil, H., *Economic Forecasts and Policy*, North-Holland Publishing Company, 1970.
- [61] Tischler, M. and Remple, R., *Aircraft and Rotorcraft System Identification*, AIAA, 2006.
- [62] Zames, G. and El-Sakkary, A. K., “Unstable Systems and Feedback: The Gap Metric,” *Allerton Conference*, 1980, pp. 380–385.
- [63] El-Sakkary, A. K., “The Gap Metric: Robustness of Stabilization of Feedback Systems,” *IEEE Transactions on Automatic Control*, Vol. 30, No. 3, 1985, pp. 240–247.
- [64] Mota, S. D. J. and Botez, R. M., “New Helicopter Model Identification Method Based on Flight Test Data,” *The Aeronautical Journal*, Vol. 115, No. 1167, 2011.
- [65] Kheir, N. and Holmes, W. M., “On Validating Simulation Models of Missile Systems,” *SIMULATION*, Vol. 30, No. 4, 1978, pp. 117–128.
- [66] Vidyasagar, M., “The Graph Metric for Unstable Plants and Robustness Estimates for Feedback Stability,” *IEEE Transactions on Automatic Control*, Vol. 29, No. 5, 1984, pp. 403–418.
- [67] Georgiou, T. T., “On the Computation of the Gap Metric,” *Conference on Decision and Control*, 1988, pp. 1360–1361.
- [68] Zhu, S., Hautus, M., and Praagman, C., “Sufficient Conditions for Robust BIBO Stabilization: Given by the Gap Metric,” *Systems & Control Letters*, Vol. 11, No. 1, 1988, pp. 53–59.
- [69] El-Sakkary, A. K., “Estimating Robustness on the Riemann Sphere,” *International Journal of Control*, Vol. 49, No. 2, 1989, pp. 561–567.
- [70] Glover, K. and McFarlane, D., “Robust Stabilization of Normalized Coprime Factor Plant Descriptions with H_∞ -bounded Uncertainty,” *IEEE Transactions on Automatic Control*, Vol. 34, No. 8, 1989, pp. 821–830.
- [71] Georgiou, T. T. and Smith, M. C., “Optimal Robustness in the Gap Metric,” *IEEE Transactions on Automatic Control*, Vol. 35, No. 6, 1990, pp. 673–686.

- [72] Qiu, L. and Davison, E., “Feedback Stability Under Simultaneous Gap Metric Uncertainties in Plant and Controller,” *Systems & Control Letters*, Vol. 18, No. 1, 1992, pp. 9–22.
- [73] Foias, C., Georgiou, T. T., and Smith, M. C., “Robust stability of feedback systems: A geometric approach using the gap metric,” *SIAM Journal on Control and Optimization*, Vol. 31, No. 6, 1993, pp. 1518–1537.
- [74] Sefton, J. and Ober, R. J., “On the Gap Metric and Coprime Factor Perturbations,” *Automatica*, Vol. 29, No. 3, 1993, pp. 723–734.
- [75] Vinnicombe, G., “Frequency Domain Uncertainty and the Graph Topology,” *IEEE Transactions on Automatic Control*, Vol. 38, No. 9, 1993, pp. 1371–1383.
- [76] Glover, K., Vinnicombe, G., and Papageorgiou, G., “Guaranteed Multi-Loop Stability Margins and the Gap Metric,” *Conference on Decision and Control*, 2000, pp. 4084–4085.
- [77] Vinnicombe, G., *Uncertainty and Feedback: H_∞ Loop-shaping and the ν -gap Metric*, World Scientific, 2001.
- [78] Cantoni, M. and Vinnicombe, G., “Controller Discretization: A gap Metric Framework for Analysis and Synthesis,” *IEEE Transactions on Automatic Control*, Vol. 49, No. 11, 2004, pp. 2033–2039.
- [79] Georgiou, T. T. and Smith, M. C., “Robustness Analysis of Nonlinear Feedback systems: An Input-Output Approach,” *IEEE Transactions on Automatic Control*, Vol. 42, No. 9, 1997, pp. 1200–1221.
- [80] French, M., “Adaptive Control and Robustness in the Gap Metric,” *IEEE Transactions on Automatic Control*, Vol. 53, No. 2, 2008, pp. 461–478.
- [81] Georgiou, T. T., Shankwitz, C. R., and Smith, M. C., “Identification of Linear Systems: A Graph Point of View,” *American Control Conference*, IEEE, 1992, pp. 307–312.
- [82] Date, P. and Vinnicombe, G., “Algorithms for worst case identification in H and in the ν -gap metric,” *Automatica*, Vol. 40, No. 6, 2004, pp. 995–1002.

- [83] Steele, J. H. and Vinnicombe, G., “Closed-loop Time-domain Model Validation in the ν -gap Metric,” *Conferenc on Decision and Control*, Vol. 5, IEEE, 2001, pp. 4332–4337.
- [84] Bombois, X., Gevers, M., and Scorletti, G., “A Measure of Robust Stability for an Identified Set of Parametrized Transfer Functions,” *IEEE Transactions on Automatic Control*, Vol. 45, No. 11, 2000, pp. 2141–2145.
- [85] Vidyasagar, M., *Control system synthesis: a factorization approach*, Morgan & Claypool Publishers, 2011.
- [86] Doyle, J. C., Francis, B. A., and Tennenbaum, A. R., *Feedback Control Theory*, Dover, 2009.
- [87] Vinnicombe, G., *Measuring the Robustness of Feedback Systems*, Ph.D. thesis, University of Cambridge, 1992.
- [88] Bauer, P. and Bokor, J., “Tuning and Improvements in a Waypoint and Trajectory Tracking Algorithm,” *AIAA Guidance, Navigation, and Control Conference*, No. 2012-4604, Minneapolis, MN, 2012.
- [89] Bauer, P. and Dorobantu, A., “Optimal Waypoint Guidance, Trajectory Design and Tracking,” *American Control Conference*, 2013.