NONLINEAR ADAPTIVE DYNAMIC INVERSION CONTROL FOR VARIABLE STABILITY SMALL UNMANNED AIRCRAFT SYSTEMS

A Thesis

by

JOSHUA A. HARRIS

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	John Valasek
Committee Members,	Srinivas Rao Vadali
	Manoranjan Majji
	Gregory Huff
Head of Department,	Rodney Bowersox

December 2017

Major Subject: Aerospace Engineering

ABSTRACT

In-flight simulation and variable stability aircraft provide useful capabilities for flight controls development such as testing control laws for new aircraft earlier, identification of adverse conditions such as pilot-induced oscillations, and handling qualities research. While these capabilities are useful they are not without cost. The expense and support activities needed to safely operate in-flight simulators has limited their availability to military test pilot schools and a few private companies. Modern computing power allows the implementation of advanced flight control systems on size, weight, and power constrained platforms such as small uninhabited aerial systems used by universities and research organizations. This thesis aims to develop a flight control system that brings in-flight simulation capability to these platforms. Two control systems based on model reference and $\mathcal{L}_1$ adaptive augmentation of baseline nonlinear dynamic inversion controllers are proposed and evaluated against a command augmentation system design and in-flight simulation cases for a variety of linear and nonlinear models. Simulation results demonstrate that both proposed control architectures are able to meet the control objectives for tracking and in-flight simulation and performance and stability robustness in the presence of severe turbulence.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a thesis committee consisting of Professors John Valasek, Srinivas Rao Vadali, and Manoranjan Majji of the Department of Aerospace Engineering and Professor Gregory Huff of the Department of Electrical and Computer Engineering.

All other work conducted for the thesis was completed by the student independently.

Funding Sources

Graduate study was supported in part by the National Defense, Science, and Engineering Graduate Fellowship program.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 Motivation

Variable stability (VS) aircraft, also referred to as in-flight simulators (IFS), are aircraft which can simulate the dynamic and kinematic performance of other aircraft in flight. That is, a VS aircraft can behave like another aircraft when flown. This ability is particularly advantageous for flying qualities and flight controls research, where new designs can be inexpensively tested in real-world conditions to provide additional insights to computer simulation at earlier program stages. Variable stability aircraft are also an important part of test pilot training at institutions such as the U.S. Air Force and U.S. Navy Test Pilot Schools [1], allowing test pilots to experience flight in aircraft with unstable modes. One of the most useful functions of VS/IFS aircraft is the testing of flight control systems for new aircraft in development. Examples of this use include both the Advanced Tactical Fighter (YF-22/F-22A) and Joint Strike Fighter (X-35/F-35) programs and the NASA X-38 Crew Return Vehicle program [2]. A particularly noteworthy example is the identification of a pilot induced oscillation (PIO) in the landing mode of the YF-17 flight control system, which was identified using an IFS aircraft and corrected prior to the aircraft's first flight; additionally, another PIO caused by digital sampling issues in the follow-on F/A-18 Hornet was similarly detected [1].

Despite the many benefits offered by VS/IFS aircraft their use is somewhat limited in scope, with aircraft availability limited to government or a small number of private companies. In addition to the low availability, manned flight testing remains expensive and risky, both of which are detrimental in the current budgetary environment. A potential solution to this problem is the use of small uninhabited aerial systems (SUAS) flight test aircraft. Advances in consumer electronics technology, particularly smartphones and computing, have resulted in large cost reductions in Micro-Electro-Mechanical Systems (MEMS) sensors and processors, allowing for the development of small and inexpensive autopilot systems (e.g. common open-source autopilot hardware and software costs roughly \$200, whereas previously proprietary autopilot hardware costing many thousands of dollars was required). These autopilots enabled the development and operation of SUAS in the commercial and research sectors, enabling new industries (e.g. precision agriculture and infrastructure assessment) as well as experimental research in autonomy, flight control, and robotics.

Unmanned flight testing is increasingly commonplace, as it is both safer since the pilot and test engineers are not onboard the aircraft and cheaper due to the lower cost of unmanned aircraft than manned flight testing. In particular, the Vehicle Systems & Control Laboratory (VSCL), part of the Department of Aerospace Engineering at Texas A&M University, operates an extensive unmanned flight testing program, with vehicles ranging from modified radio control aircraft to custom-designed SUAS in the 100 lb weight class. Established manned flight testing organizations such as the USAF Test Pilot School are also increasingly using SUAS for reasons including cost reduction and support of Remotely Piloted Aircraft (RPA) programs. As a result, there is growing interest in bringing the technology of variable stability aircraft to the world of SUAS due to the capabilities outlined at the beginning of this section. Another usage for a variable stability SUAS particular to academia is use as a teaching platform for flight dynamics courses.

An early step toward this goal was the design of the VSCL Pegasus UAS, which features a movable wing so that the longitudinal static stability (C_{m_α}) of the aircraft can be changed on the ground [3]. While this allows the Pegasus UAS to be flown at differing levels of longitudinal stability, it does not allow for it to replicate arbitrary dynamics, although the multiple redundant control surfaces make the aircraft well-suited for fault tolerant and reconfigurable control flight testing. VSCL researchers also desire the ability to change the simulated dynamics in-flight and not simply on the ground. The largest problem for using Pegasus for experimental flight control testing is that the vehicle is a high-value asset. This further serves as reasoning for an inexpensive variable stability SUAS, as such a platform could be used to prove new control laws for the more expensive aircraft prior to actual implementation on those aircraft.

Accordingly, requirements for a VS SUAS can be determined:

1. it should be based on a readily-available, inexpensive Commercial, Off-the-Shelf (COTS) airframe,
2. it should have simple, uncomplicated inherent dynamics,
3. it should be able to simulate the dynamics of common SUAS,
4. it should use inexpensive, COTS flight control hardware, and
5. the Variable Stability System (VSS) should be based on some form of model following controller.

The rationale for these requirements follows. Primarily, the IFS SUAS should meet requirements while being as inexpensive as possible, so that the vehicle can be used in low-budget environments (e.g. universities) and so that loss-of-vehicle is not cost prohibitive. Using a COTS airframe also reduces the development and integration time required compared to a bespoke design. The aircraft should have inherently stable dynamics and a conventional design, so that the aircraft can be reverted to direct stick-to-surface manual control in the event of an upset and so that publicly available aerodynamic prediction methods can be used for model development. In order to be useful, the platform should be able to simulate the dynamics of common SUAS configurations. This translates to the requirement that the base platform should have a sufficiently expansive flight envelope and fast actuators. Use of COTS flight control hardware for the control law implementation reduces cost by not requiring a custom flight control system, allowing the work to be focused on the control law development itself. Finally, using a model-following approach instead of RFB allows for the leverage of modern control theoretic techniques, making the VSS system design and implementation simpler. This thesis focuses on the design of the VSS for an SUAS-class in-flight simulator using a fusion of adaptive control and nonlinear dynamic inversion flight control approaches.

1.2 Historical and Literature Review

This section provides a review of the literature on variable stability/in-flight simulation aircraft and adaptive dynamic inversion flight control. For the former, a brief historical overview of methods and airframes used is presented.

1.2.1 Variable Stability Aircraft and In-Flight Simulators

Variable stability aircraft have a long history beginning with aeronautics research undertaken shortly after the Second World War. The first variable stability aircraft was a Grumman F6F-3 Hellcat modified by the Ames Aeronautical Laboratory to allow for varying dihedral effect (C_{ℓ_β}) by servo actuation of the ailerons. The objective of this program was to determine the safe limits of dihedral effect for normal fighter aircraft operation at different flight conditions. This aircraft had a single axis of variable stability in roll [4].

The first three-axis variable stability aircraft was the USAF/Calspan NT-33, based on the Lockheed T-33 Shooting Star trainer, as seen in Fig. 1.1.

Figure 1.1: NT-33A variable stability aircraft. (Reprinted from [5].)

The NT-33 originally used an analog flight computer to implement a VSS with two components: a variable feel system and a Response Feedback (RFB) flight control system to modify the base NT-33 dynamics [2]. The variable feel system consists of a set of sensors and servos which record the pilot's inputs and generate an artificial "feel" of the control system. This system allows the simulation of feel characteristics varying from a traditional irreversible flight control system to a zero-deflection force control system such as those found on modern fly-by-wire aircraft. The pilot's inputs are passed to the VSS flight control system [6]. The RFB technique uses classical SISO feedback loops to change the frequency and damping of the aircraft modes to achieve desired values. This approach for RFB is described in Refs. [7, 8].

A full, six degree-of-freedom (6-DOF) in-flight simulator was realized with the Total In-Flight Simulator (TIFS) aircraft, based on the C-131 transport. Direct force control is obtained by a combination of direct lift flaps and sideforce surfaces. The TIFS uses a model following approach implemented with feedback and feedforward gains to replicate the desired dynamics to be simulated. These gains are either determined for the maximum dynamic pressure to be encountered in the simulation or are scheduled based on the dynamic pressure. With the direct force effectors, the TIFS aircraft has can accurately simulate 6-DOF of motion [9].

The current state-of-the-art in variable stability aircraft is the Variable-stability, In-flight Simulator Test Aircraft, “VISTA”, first proposed in 1984 to replace the NT-33A. This was largely driven by the need for a newer aircraft to reduce logistics issues and the limitations of the NT-33A performance envelope for simulating modern fighter aircraft designs [1]. The VISTA proposal ultimately resulted in an in-flight simulator based on the Lockheed Martin F-16D fighter aircraft, shown in Fig. 1.2.

Figure 1.2: NF-16D VISTA aircraft in near 90 degree bank. (Reprinted from [10].)

VISTA was developed by the USAF Flight Dynamics Laboratory (now AFRL), Lockheed Martin (then General Dynamics) and Calspan. In addition to the traditional RFB control law structure, VISTA was envisioned to add a model following controller to reduce the need for accurate modeling of the simulator vehicle’s stability and control derivatives [11]. VISTA was used to support several development and research programs, and is currently operated by the USAF TPS [2].

Calspan additionally developed a fleet of variable stability Learjets. These aircraft were modified to have a 3-DOF (roll, pitch, yaw) VSS based on previous variable stability aircraft; an additional DOF for the throttle was subsequently added. The Learjets added a digital configuration control system to change the VSS gains; previously potentiometers required manual adjustment. This system was eventually replaced with a digital flight control system using model following and implemented with the MATLAB®/Simulink® environment [2, 12]. These aircraft were used for several projects, including use as UAV surrogates and testing of adaptive controllers. [13–15]

A dynamic inversion approach for in-flight simulators is used in Ref. [16] for a proposed in-flight simulator based on the T-6A Texan II trainer aircraft. This approach used a simplified form of dynamic inversion (Ref. [17]) to track linear state-space reference models. The dynamic inversion model following scheme is shown as a superior approach to the traditional response feedback methods.

1.2.2 Nonlinear Adaptive Dynamic Inversion Control

The aerodynamics of an aircraft change with flight condition based on parameters such as dynamic pressure, angle-of-attack, sideslip angle, and Mach number. Flight control systems must safely operate over the entire flight envelope. For linear control techniques the common approach is to generate linear models at trim conditions over the flight envelope, design controllers at each of these points, and then change the current controller gains based off of the current flight condition. This approach is called gain scheduling [18]. This is a time-consuming and costly process, motivating research intro control techniques that avoid the need for gain scheduling. Two approaches that have proven particularly useful for flight control are adaptive control and dynamic inversion.

Adaptive control methods modify the gains of a baseline control law online based on sensor feedback [19]. Adaptive controllers can be classified into two main classes: *direct* adaptive controllers, which estimate controller parameters, and *indirect* adaptive controllers, which synthesize controller parameters from estimated plant parameters. Indirect adaptive controllers are also referred to as *self-tuning regulators* [18]. Both methods can be viewed as a combination of a baseline control law and an online parameter estimator.

Dynamic inversion control is a subset of feedback linearization control, which uses the mechanism of feedback to produce linear dynamics [20]. Nonlinear dynamic inversion (NDI) in theory allows the substitution of desired dynamics (i.e. desired derivatives of the controlled variables) perfectly for the inherent dynamics. In practice, considerations such as modeling error, unmodeled dynamics, actuator dynamics, and sensor noise prevent perfect cancellation of the inherent dynamics from occurring. The lack of robustness of NDI requires an outer loop control for robustness [21]. One choice is to use adaptive control methods to provide robustness to the NDI baseline controller.

The combination of adaptive control methods with NDI is increasingly prevalent. A simple SISO example of an adaptive dynamic inversion controller is presented in Ref. [22] showing a direct model reference adaptive control (DMRAC) approach to dynamic inversion. Reference [23] uses a DMRAC approach to generate nonlinear adaptive dynamic inversion control laws to design state-constrained Command Augmentation Systems (CAS) for a generic hypersonic vehicle. This approach is extended to the output feedback case in Ref. [24]. References [25, 26] introduce Structured Adaptive Model Inversion (SAMI) control, which uses DMRAC techniques with an NDI control law while avoiding adaptation on perfectly-known kinematic equations. A fault-tolerant formulation for SAMI is presented in Ref. [27], and is applied to the problem of autonomous air refueling in Ref. [28]. A different approach is found in Refs. [22, 29–31], which introduce adaptive dynamic inversion controllers with neural networks using single hidden layer and radial basis function network architectures. Reference [32] provides $\mathcal{L}_1$ adaptive augmentation to a baseline dynamic inversion controller for the X-48B blended-wing body aircraft. Unlike other approaches above, the $\mathcal{L}_1$ adaptive control signal is added to the baseline NDI control law, which is unmodified. A similar approach is used in Ref. [33] for the design of a missile autopilot. These latter two works are sampled-data controllers. An $\mathcal{L}_1$ adaptive NDI controller for the NASA Langley Generic Transport Model is introduced in Ref. [34]. This control architecture uses the $\mathcal{L}_1$ controller to directly adapt the NDI baseline control law in a manner similar to that of the previous DMRAC techniques. The $\mathcal{L}_1$ adaptive controller is also applied to proportional-integral desired dynamics which are used to convert pilot commands to desired rates of controlled variables.

1.3 Research Objectives

The objective of this thesis is the design of a variable stability system suitable for use on SUAS-class vehicles. The intended application is the support of the VSCL unmanned flight testing program and the AERO 425 Flight Test Engineering and AERO 445 Vehicle Management Systems courses by adding an IFS capability. In particular, the capability to use in-flight simulation for verification and validation of experimental control laws for high-value assets such as the Pegasus UAS is strongly desired. While the scope of this thesis is limited to a simulation study of candidate control laws for the VSS, the ultimate objective is the implementation of a VSS on a SUAS and operational use.

The first objective is the development of a six degree-of-freedom (6-DOF) nonlinear flight simulation for evaluating the VSS designs. While the ultimate application is SUAS, the control laws are tested on models for full-size manned aircraft. This choice is made given the relative lack of freely available high-fidelity SUAS models. By contrast, linear and nonlinear models are widely available in the literature for “traditional” manned aircraft. This availability allows for testing the VSS against a wide variety of aircraft models. The simulation is developed using the MATLAB/Simulink environment, which enables model-based control design and offers the potential for autocoding of the actual flight software. The Simulink Aerospace Blockset additionally provides many useful building blocks, reducing the effort required to develop the simulation. For these reasons, MATLAB/Simulink has become the *de facto* standard for control law development in the aerospace industry.

The second objective is to develop nonlinear adaptive dynamic inversion (NADI) control laws to form the basis of the VSS design. The NADI approach was chosen for several reasons. An NDI baseline has several advantages:

1. nonlinear dynamic inversion controllers are valid over the entire flight envelope and do not require gain scheduling,
2. the structure of dynamic inversion allows the substitution of desired dynamics for the inherent dynamics, and
3. dynamic inversion is conceptually simple.

Meanwhile, adding an adaptive element allows for the compensation of modeling errors and unmodeled dynamics and offers improved robustness. An adaptive approach is particularly useful for SUAS since high-fidelity models are rarely available and the manufacturers of radio controlled aircraft that generally form the basis of research SUAS are either unable or unwilling to provide information on their products. Two NADI control laws are to be developed: a DMRAC-based design similar to those used in VSCL hypersonic control research, and an $\mathcal{L}_1$ adaptive control-based design.

The third objective is the development of the variable stability system. The VSS design primarily consists in the selection of desired dynamics for the dynamic inversion baseline. Designs for basic variable stability (e.g. changing damping response) and for in-flight simulation of different aircraft are considered. In particular, it is desired to be able to simulate both linear and nonlinear aircraft models. Current model reference control schemes generally consider only linear reference models. Additionally, the integration of the VSS within the larger flight control system is discussed. This latter point is important as the requirements for safe operation of a manned IFS aircraft do not necessarily directly map over to a SUAS IFS aircraft.

1.4 Outline of the Thesis

This thesis is intended such that a new graduate student with basic familiarity in systems theory can understand the control laws developed in this thesis sufficiently to be able to apply them to SUAS flight control research. Accordingly the thesis is written in a pseudo-pedagogical manner, beginning with a review of basic flight dynamics and aircraft modeling in Section 2. This is followed by Section 3, a description of the Simulink aircraft model used for simulation examples throughout the thesis. The section also describes the outline of a basic Command Augmentation System (CAS) design that is used to provide common example for comparison of the different control laws. Returning to the requisite background material, Sections 4–5 introduce nonlinear dynamic inversion and basic adaptive control design, respectively. The main contributions of this thesis appear in Section 6, which introduces nonlinear adaptive dynamic inversion control laws and applies them to the common CAS example, and in Section 7, which presents simulation results for a variety of variable stability and in-flight simulation controller designs. These results feature the in-flight simulation of two dissimilar aircraft, a jet fighter and a subscale research UAS. Section 8 presents conclusions and is followed by Section 9 which presents recommendations for future work.

Additional in-flight simulation examples with both linear and nonlinear models are provided in Appendix A. Model data for the in-flight simulation examples are presented in Appendix B. Finally, descriptions of disturbance models are provided in Appendix C.

2. AIRCRAFT FLIGHT DYNAMICS AND MODELING

2.1 Introduction

The dynamic inversion baseline control laws considered in this thesis require knowledge of the aircraft equations of motion and forces in order to obtain the desired control demands. Accordingly, a review of the field of aircraft flight dynamics and modeling is a required preliminary. The goal of this section is to provide a concise review of the field as an in-depth study is necessarily out-of-scope of this thesis. For more detail on the material presented below the reader is suggested to consult one of the texts by McRuer et al. [35], Roskam [36], or Schmidt [37].

The section is split into two main "thrusts": the dynamic equations, and the modeling of the forces and moments. These can be thought of simply as the "$m\boldsymbol{a}$" and "$\boldsymbol{F}$" terms, respectively, in Newton's Second Law. First the six degree-of-freedom nonlinear equations of motion for a rigid-body, fixed-wing aircraft are developed, followed by discussion of trimmed flight conditions and locally-linear models traditionally used for flight control design. Modeling of aerodynamic forces and moments is presented second, particularly the development of aerodynamic databases for simulation and use as onboard models for dynamic inversion control laws. Propulsive forces and moments are neglected in this section. Finally, simple models for aircraft actuators and their failure modes are introduced as the effects of these dynamics are non-negligible.

2.2 Equations of Motion

This section derives the translational and rotational equations of motion for a conventional aircraft configuration. Ignoring flexible-body effects, aircraft can be modeled as rigid bodies using standard techniques from classical mechanics. While true aircraft are not rigid bodies and can have significant structural deformation in flight, the rigid body approximation is useful for most flight control design problems. This section develops the translational and rotational equations of motion for a rigid aircraft under the assumptions of constant mass and a flat earth. These equations of motion are also referred to as kinetic equations as they describe the effect of the forces and moments on the system of interest. For modeling of flexible aircraft the reader is invited to consult Chapters 4 and 7 of the textbook by Schmidt [37].

The equations of motion provide six first order ordinary differential equations that describe the velocity level state variables. Since the behavior of a rigid body with six degrees of freedom requires six second order ODEs or twelve first-order ODEs to completely describe its motion, six further equations are needed. These are the kinematic equations, which are a set of ODEs that relate the velocity and position level variables. The kinematic equations are consequences of the geometry of the problem and are known exactly.

2.2.1 Preliminaries

This section discusses the frames of reference and attitude representation used prior to deriving the equations of motion.

2.2.1.1 Frames of Reference

Three reference frames will be considered in this section:

- a North-East-Down (NED) frame considered to be inertial relative to the aircraft,
- an aircraft-fixed non-inertial frame aligned with the aircraft body, and
- an aircraft-fixed non-inertial frame aligned with the projection of the aircraft velocity vector.

These reference frames are described further below.

2.2.1.1.1 North East Down Frame: The NED reference frame is a tangent-plane coordinate system defined with its x-axis pointing to geographic North, its y-axis pointing toward geographic East, and its z axis pointing down toward the Earth center, with the origin of the coordinate system located on a tangent plane to the Earth surface [31]. This coordinate system is, strictly speaking, non-inertial, but can be considered inertial over the scale of the aircraft flight control problem. The NED frame will be denoted as $\mathcal{N}$, with basis vectors $\{\hat{\boldsymbol{x}}_\mathrm{n}, \hat{\boldsymbol{y}}_\mathrm{n}, \hat{\boldsymbol{z}}_\mathrm{n}\}$.

2.2.1.1.2 Aircraft Body-Fixed Axis System: The standard aircraft body-fixed frame is defined as a right-handed frame with the x-axis pointing out the aircraft nose, the y-axis pointing out the right wing, and the z-axis pointing down such that an orthogonal basis is formed. In this thesis, the aircraft-fixed frame will be denoted as $\mathcal{A}$ with basis vectors $\{\hat{\boldsymbol{x}}_\mathrm{a}, \hat{\boldsymbol{y}}_\mathrm{a}, \hat{\boldsymbol{z}}_\mathrm{a}\}$. The origin of the aircraft body-fixed frame is the aircraft mass center, which is assumed to be constant over the flight. Figure 2.1 shows the aircraft body axis system, as well as the stability axis system, described below. In Fig. 2.1, V_T is the aircraft velocity

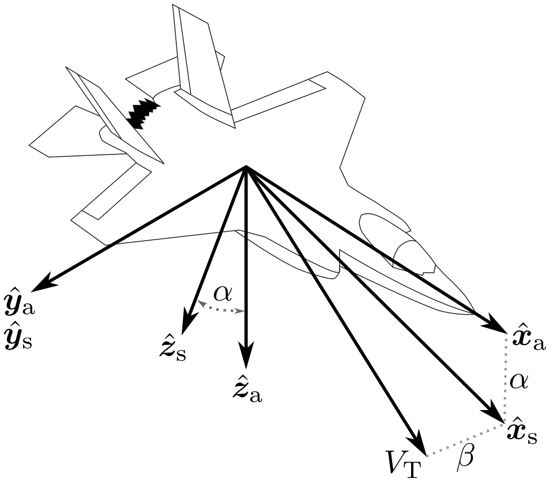


Figure 2.1: Aircraft body and stability axis systems.

(true airspeed), α is the angle-of-attack (AOA), and β is the sideslip angle.

2.2.1.1.3 Aircraft Stability Axis System: The stability axis system is a special body-fixed frame where the x-axis is rotated to align with the projection of the aircraft velocity vector on the xz plane for a reference flight condition [37]. The stability y-axis is the same as in the body frame, while the stability z-axis completes a right handed system. The stability axes are obtained from the body axes by a negative rotation of α about the $\hat{\boldsymbol{y}}_{\mathrm{a}} = \hat{\boldsymbol{y}}_{\mathrm{s}}$ axis. In this thesis, the stability frame will be denoted as $\mathcal{S}$ with basis vectors $\{\hat{\boldsymbol{x}}_{\mathrm{s}}, \hat{\boldsymbol{y}}_{\mathrm{s}}, \hat{\boldsymbol{z}}_{\mathrm{s}}\}$. Figure 2.1 shows the stability axes in addition to the body axes.

2.2.1.2 Attitude Representation

The attitude of a rigid body describes its orientation with respect to some reference frame, and can thus be viewed mathematically as a change of basis. Consequently the fundamental representation of attitude is the direction cosine matrix (DCM). Direction cosine matrices are orthogonal matrices in the group SO(3) which represent right-handed coordinate transformations. The orthogonality of a DCM $[C]$ implies the constraint $[C][C]^{\mathsf{T}} = [C]^{\mathsf{T}}[C] = [I_3]$, which results in three independent parameters and six dependent parameters. Since a DCM contains six redundant parameters, it is common to parameterize the DCM with a reduced set of attitude coordinates.

The most common method for representing a DCM is through the use of Euler angles, which parameterize the DCM in terms of simple rotations about the coordinate axes. Euler angle representations are broadly grouped into two categories: asymmetric sets of rotations (each rotation is about a distinct axis) and symmetric representations (the first and third rotations are about the same axis, while the second differs). Aircraft attitude is commonly described using the asymmetric 3–2–1 set of Euler angles. The 3–2–1 Euler angles consist of a right-handed rotation about the third axis, followed by a rotation about the second axis and then a rotation about the first axis. These angles are referred to as the yaw angle Ψ, pitch angle Θ, and roll angle Φ, respectively. The direction cosine matrix may be formed from the 3–2–1 Euler angles as $[C(\Psi, \Theta, \Phi)] = [R_1(\Phi)][R_2(\Theta)][R_3(\Psi)]$, where $[R_i(\xi)]$ consists of a rotation about the i-th axis by an angle ξ. Expanding this, it is straightforward to show that the DCM can be represented in terms of the 3–2–1 Euler angles as

$$[C(\Psi,\Theta,\Phi)]=\begin{bmatrix} \cos\Theta\cos\Psi & \cos\Theta\sin\Psi & -\sin\Theta \\ \sin\Phi\sin\Theta\cos\Psi-\cos\Phi\sin\Psi & \sin\Phi\sin\Theta\sin\Psi+\cos\Phi\cos\Psi & \sin\Phi\cos\Theta \\ \cos\Phi\sin\Theta\cos\Psi+\sin\Phi\sin\Psi & \cos\Phi\sin\Theta\sin\Psi-\sin\Phi\cos\Psi & \cos\Phi\cos\Theta \end{bmatrix} \tag{2.1}$$

and that the Euler angles can be extracted from the DCM as

$$\Psi = \tan^{-1}\left(\frac{C_{12}}{C_{11}}\right), \tag{2.2}$$

$$\Theta = -\sin^{-1}(C_{13}), \tag{2.3}$$

$$\Phi = \tan^{-1}\left(\frac{C_{23}}{C_{33}}\right). \tag{2.4}$$

Note that Eq. (2.2) and Eq. (2.4) both become indeterminate when $|\Theta| = 90°$. In this case, Ψ and Φ cannot be uniquely determined and thus this is referred to as an orientation singularity.

2.2.2 Translational Equations of Motion

Newton's Second Law can be expressed as

$$\frac{{}^N\mathrm{d}(m\boldsymbol{V})}{\mathrm{d}t} = \boldsymbol{F}, \tag{2.5}$$

where m is the aircraft mass, $\boldsymbol{V}$ is the aircraft velocity in the body frame, and $\boldsymbol{F}$ is the net applied force. Assuming constant mass, Eq. (2.5) becomes

$$m\,\frac{{}^N\mathrm{d}\boldsymbol{V}}{\mathrm{d}t} = \boldsymbol{F}. \tag{2.6}$$

Since the aircraft frame $\mathcal{A}$ rotates relative to the inertial frame $\mathcal{N}$ with angular velocity $\boldsymbol{\omega}$, the inertial derivative of the translational velocity $\boldsymbol{V}$ using the kinematic transport theorem (consult one of Refs. [38, 39]) is

$$\frac{^N\mathrm{d}\boldsymbol{V}}{\mathrm{d}t} = \dot{\boldsymbol{V}} + \boldsymbol{\omega} \times \boldsymbol{V}\,, \tag{2.7}$$

resulting in

$$m\left(\dot{\boldsymbol{V}} + \boldsymbol{\omega} \times \boldsymbol{V}\right) = \boldsymbol{F}\,. \tag{2.8}$$

Here, $\dot{\boldsymbol{V}}$ represents the time derivative of the velocity vector expressed in the aircraft body-axis frame $\mathcal{A}$. The vector $\boldsymbol{V}$ is given body-axis components $\boldsymbol{V} = U\hat{\boldsymbol{x}}_\mathrm{a} + V\hat{\boldsymbol{y}}_\mathrm{a} + W\hat{\boldsymbol{z}}_\mathrm{a}$, and the angular velocity is given body-axis components $\boldsymbol{\omega} = P\hat{\boldsymbol{x}}_\mathrm{a} + Q\hat{\boldsymbol{y}}_\mathrm{a} + R\hat{\boldsymbol{z}}_\mathrm{a}$. Substituting these values into Eq. (2.8) results in the three scalar equations

$$m(\dot{U} + QW - VR) = F_\mathrm{x}\,, \tag{2.9a}$$

$$m(\dot{V} + RU - PW) = F_\mathrm{y}\,, \tag{2.9b}$$

$$m(\dot{W} + PV - QU) = F_\mathrm{z}\,. \tag{2.9c}$$

Conventionally, the force terms are further subdivided into forces due to aerodynamics, thrust, and gravity [36]. Doing so results in the rigid-body aircraft translational equations of motion,

$$m(\dot{U} + QW - VR) = F_{\mathrm{A_x}} + F_{\mathrm{T_x}} + F_{\mathrm{G_x}}\,, \tag{2.10a}$$

$$m(\dot{V} + RU - PW) = F_{\mathrm{A_y}} + F_{\mathrm{T_y}} + F_{\mathrm{G_y}}\,, \tag{2.10b}$$

$$m(\dot{W} + PV - QU) = F_{\mathrm{A_z}} + F_{\mathrm{T_z}} + F_{\mathrm{G_z}}\,. \tag{2.10c}$$

Equations (2.10) are a set of three first-order nonlinear ordinary differential equations that govern the translational dynamics of the aircraft. For simulation, the equations of motion are rewritten in the form Eq. (2.11), shown in the vector form,

$$\dot{\boldsymbol{V}} = -\boldsymbol{\omega} \times \boldsymbol{V} + \frac{1}{m}\left(\boldsymbol{F}_{\mathrm{A}} + \boldsymbol{F}_{\mathrm{T}} + \boldsymbol{F}_{\mathrm{G}}\right) \tag{2.11}$$

which can be directly integrated to obtain the body-axis velocities.

Applying the assumption that the Earth can be represented as flat, the gravity term is defined as acting along the $\hat{\boldsymbol{z}}_{\mathrm{n}}$ direction. The gravitational force vector is then converted to the body-frame via a coordinate transformation for use with Eq. (2.10) or Eq. (2.11). This approximation is valid over small distances and is thus useful for models used to develop flight control laws.

2.2.3 Translational Kinematics

The translational kinematics are commonly formulated in terms of the NED coordinates $\{X_{\mathrm{n}}, Y_{\mathrm{n}}, Z_{\mathrm{n}}\}$, and can be simply expressed using the DCM and velocity vector as

$$\begin{Bmatrix} \dot{X}_{\mathrm{n}} \\ \dot{Y}_{\mathrm{n}} \\ \dot{Z}_{\mathrm{n}} \end{Bmatrix} = \begin{bmatrix} C \end{bmatrix}^{\mathsf{T}} \begin{Bmatrix} U \\ V \\ W \end{Bmatrix} . \tag{2.12}$$

Equation (2.12) can then simply be integrated to obtain the NED position. The aircraft altitude $H = -Z_{\mathrm{n}}$.

2.2.4 Rotational Equations of Motion

The rotational motion about the aircraft center of mass can be described using Eulerian mechanics, relating the angular momentum and moments:

$$\frac{{}^{N}\mathrm{d}\boldsymbol{h}_{\mathrm{c}}}{\mathrm{d}t} = \boldsymbol{\ell}_{\mathrm{c}}\,. \tag{2.13}$$

Using the definition of the angular momentum $\boldsymbol{h} = [I]\boldsymbol{\omega}$, where $[I]$ is the moment of inertia of the aircraft, and applying the kinematic transport theorem, Eq. (2.13) can be written as

$$[I]\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times [I]\boldsymbol{\omega} = \boldsymbol{\ell}_{\mathrm{c}}\,, \tag{2.14}$$

assuming that inertia tensor remains constant. The representation of the inertia tensor has elements

$$[I] = \begin{bmatrix} I_{\mathrm{xx}} & -I_{\mathrm{xy}} & -I_{\mathrm{xz}} \\ -I_{\mathrm{xy}} & I_{\mathrm{yy}} & -I_{\mathrm{yz}} \\ -I_{\mathrm{xz}} & -I_{\mathrm{yz}} & I_{\mathrm{zz}} \end{bmatrix}, \tag{2.15}$$

which for a conventionally-configured aircraft with a plane of symmetry xz reduces to

$$[I] = \begin{bmatrix} I_{\mathrm{xx}} & 0 & -I_{\mathrm{xz}} \\ 0 & I_{\mathrm{yy}} & 0 \\ -I_{\mathrm{xz}} & 0 & I_{\mathrm{zz}} \end{bmatrix}. \tag{2.16}$$

The moment vector is given components $\boldsymbol{\ell}_{\mathrm{c}} = L\hat{\boldsymbol{x}}_{\mathrm{a}} + M\hat{\boldsymbol{y}}_{\mathrm{a}} + N\hat{\boldsymbol{z}}_{\mathrm{a}}$ and is partitioned into moments due to aerodynamics and thrust effects, similarly to the partitioning of the forces in the translational EOM. Since the gravitational force is assumed to act through the aircraft mass center it does not produce a moment. Accordingly, the vector rotational equations of motion produce three rotational equations,

$$I_{\mathrm{xx}}\dot{P} - I_{\mathrm{xz}}(\dot{R} + PQ) + (I_{\mathrm{zz}} - I_{\mathrm{yy}})QR = L_{\mathrm{A}} + L_{\mathrm{T}}\,, \tag{2.17a}$$

$$I_{\mathrm{yy}}\dot{Q} + I_{\mathrm{xz}}(P^2 - R^2) + (I_{\mathrm{xx}} - I_{\mathrm{zz}})PR = M_{\mathrm{A}} + M_{\mathrm{T}}\,, \tag{2.17b}$$

$$I_{\mathrm{zz}}\dot{R} - I_{\mathrm{xz}}(\dot{P} - QR) + (I_{\mathrm{yy}} - I_{\mathrm{xx}})PQ = N_{\mathrm{A}} + N_{\mathrm{T}}\,. \tag{2.17c}$$

Note that these equations are coupled through the I_{xz} terms (called the I_{xz} effect for obvious reasons) and the differences between the principle inertias e.g. $I_{\mathrm{zz}} - I_{\mathrm{yy}}$. Aircraft with poorly balanced inertias are heavily coupled and can exhibit unsatisfactory flying qualities. To allow direct integration to obtain the angular rates the equations are solved for the angular accelerations $\dot{P}$, $\dot{Q}$, and $\dot{R}$. In the vector form, the uncoupled equations are

$$\dot{\boldsymbol{\omega}} = [I]^{-1}\left(-\boldsymbol{\omega} \times [I]\boldsymbol{\omega} + \boldsymbol{\ell}_{\mathrm{A}} + \boldsymbol{\ell}_{\mathrm{T}}\right)\,. \tag{2.18}$$

2.2.5 Rotational Kinematics

The attitude kinematics are a set of exactly-known differential equations relating the angular velocities of the aircraft to the attitude rates. Generally, for a set of attitude parameters u, the attitude kinematics differential equations are a set $\dot{u} = [A(u)]\omega$, where ω is the angular velocity vector and $[A(u)]$ is the attitude influence matrix [39].

The Euler angle kinematic equations can be derived by examining the effect of the Euler angle rates $\dot{\Psi}$, $\dot{\Theta}$, and $\dot{\Phi}$ in intermediate reference frames and then using simple rotations to express the result in the aircraft body frame $\mathcal{A}$. This results in an expression of the form $\omega = [B(u)]\dot{u}$, which can be inverted to obtain the desired expression $\dot{u} = [A(u)]\omega$. These steps are left as an exercise for the reader, and the final result for the

3–2–1 set is presented below as Eq. (2.19), reproduced from Ref. [38]:

$$\begin{Bmatrix} \dot{\Psi} \\ \dot{\Theta} \\ \dot{\Phi} \end{Bmatrix} = \frac{1}{\cos\Theta} \begin{bmatrix} 0 & \sin\Phi & \cos\Phi \\ 0 & \cos\Theta\cos\Phi & -\cos\Theta\sin\Phi \\ \cos\Theta & \sin\Theta\sin\Phi & \sin\Theta\cos\Phi \end{bmatrix} \begin{Bmatrix} P \\ Q \\ R \end{Bmatrix}. \tag{2.19}$$

Note that Eq. (2.19) features a singularity corresponding to $|\Theta| = 90°$. This singularity in the kinematic equations is referred to as a kinematic singularity, and corresponds with the orientation singularity observed in Eq. (2.2) and Eq. (2.4).

2.2.6 Six Degree-of-Freedom Dynamics

The 6-DOF aircraft dynamics are thus given by the combination of the kinetic and kinematic equations for translation and rotation. These dynamics are then written generally as the first-order vector ODE

$$\dot{\boldsymbol{X}} = \boldsymbol{F}(\boldsymbol{X}, \boldsymbol{U}), \tag{2.20}$$

where $\boldsymbol{X} \in \mathbb{R}^n$ is the state vector, $\boldsymbol{U} \in \mathbb{R}^m$ is the control vector (manifested in the force and moment terms of the kinetic equations), and the nonlinear map $\boldsymbol{F} : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^n$ encodes the dynamics.

Using the 3–2–1 Euler angle attitude parameterization, the total state vector of the aircraft is given as

$$\boldsymbol{X} = [U, V, W, P, Q, R, \Psi, \Theta, \Phi, X_{\mathrm{n}}, Y_{\mathrm{n}}, Z_{\mathrm{n}}]^{\mathsf{T}}. \tag{2.21}$$

Equation (2.20) is then a set of 12 equations that represent the velocity and acceleration level dynamics for all six degrees of freedom.

The aircraft dynamics are generally expressed in terms of the true airspeed V_{T}, the aerodynamic angle-of-attack (AOA), α, and the aerodynamic sideslip angle, β. These values can be directly measured in flight (airspeed via a pitot-static probe, incidence angles with a five hole probe or vanes), and the AOA and sideslip angles are critical aerodynamic parameters.

For the purposes of simulation, the airspeed is calculated using Eq. (2.22)

$$V_{\mathrm{T}} = \sqrt{U^2 + V^2 + W^2}\,, \tag{2.22}$$

the AOA is calculated as Eq. (2.23),

$$\alpha = \tan^{-1}\left(\frac{W}{U}\right)\,, \tag{2.23}$$

and the sideslip angle is calculated as Eq. (2.24)

$$\beta = \sin^{-1}\left(\frac{V}{V_{\mathrm{T}}}\right)\,. \tag{2.24}$$

The equations of motion can then be written using V_{T}, α, and β as state variables by differentiating Eqs. (2.22–2.24) with respect to time and then substituting in the equations for $\dot{U}$, $\dot{V}$, $\dot{W}$. These expressions can then be written entirely in terms of V_{T}, α, and β by inverting the expressions Eqs. (2.22–2.24) and simplifying. In doing so it becomes easier to write the force and moment terms in the wind axes, which makes this the formulation of the equations of motion in wind axes [40].

Another option is to leave the EOM in terms of the body axis state variables and compute the airspeed and incidence angles as part of the system outputs. A typical output vector is then defined as

$$\boldsymbol{Y} = [V_{\mathrm{T}}, \alpha, \beta, P, Q, R, \Psi, \Theta, \Phi, X_{\mathrm{n}}, Y_{\mathrm{n}}, Z_{\mathrm{n}}]^{\mathsf{T}}. \tag{2.25}$$

In the most general case the output vector is an algebraic function of the state and control vectors:

$$\boldsymbol{Y} = \boldsymbol{H}(\boldsymbol{X}, \boldsymbol{U})\,, \tag{2.26}$$

where $\boldsymbol{Y} \in \mathbb{R}^p$ and $\boldsymbol{H} : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^p$. Together, Eq. (2.20) and Eq. (2.26) provide a (nonlinear) state-space representation of the aircraft dynamics.

2.2.7 Trim

The dynamics Eq. (2.20) have equilibrium states and controls $\boldsymbol{X}_1$ and $\boldsymbol{U}_1$ such that

$$\boldsymbol{0} = \boldsymbol{F}(\boldsymbol{X}_1, \boldsymbol{U}_1)\,. \tag{2.27}$$

This equilibrium point is referred to as a trim condition or trim point. In trimmed flight the aircraft maintains its current flight condition without pilot input. The trimmed output is $\boldsymbol{Y}_1 = \boldsymbol{H}(\boldsymbol{X}_1, \boldsymbol{U}_1)$. Multiple trim conditions exist throughout the aircraft flight envelope, although depending on the aircraft dynamics and the control authority there may be areas where the aircraft is unable to trim. These trim conditions are usually found through optimization methods that start from an initial guess and then vary the states and controls until the derivatives of the state variables become zero or very near to zero.

2.2.8 Linear Dynamics

The nonlinear state-space equations Eq. (2.20) and Eq. (2.26) can be linearized about a trim point to provide locally linear dynamics. The basic approach is to partition the total state, control, and output vectors as the sum of a trim (steady-state) part and a perturbation from the trimmed values: $\boldsymbol{X} = \boldsymbol{X}_1 + \boldsymbol{x}$, $\boldsymbol{U} = \boldsymbol{U}_1 + \boldsymbol{u}$, and $\boldsymbol{Y} = \boldsymbol{Y}_1 + \boldsymbol{y}$. Jacobian linearization is then applied to the functions $\boldsymbol{F}$ and $\boldsymbol{G}$ to obtain

$$\begin{aligned}
\dot{\boldsymbol{x}} &= \boldsymbol{F}(\boldsymbol{X}_1, \boldsymbol{U}_1) + \left.\frac{\partial \boldsymbol{F}(\boldsymbol{X},\boldsymbol{U})}{\partial \boldsymbol{X}}\right|_{\substack{\boldsymbol{X}=\boldsymbol{X}_1\\ \boldsymbol{U}=\boldsymbol{U}_1}} \boldsymbol{x} + \left.\frac{\partial \boldsymbol{F}(\boldsymbol{X},\boldsymbol{U})}{\partial \boldsymbol{U}}\right|_{\substack{\boldsymbol{X}=\boldsymbol{X}_1\\ \boldsymbol{U}=\boldsymbol{U}_1}} \boldsymbol{u}\,, \\
\boldsymbol{Y}_1 + \boldsymbol{y} &= \boldsymbol{H}(\boldsymbol{X}_1, \boldsymbol{U}_1) + \left.\frac{\partial \boldsymbol{H}(\boldsymbol{X},\boldsymbol{U})}{\partial \boldsymbol{X}}\right|_{\substack{\boldsymbol{X}=\boldsymbol{X}_1\\ \boldsymbol{U}=\boldsymbol{U}_1}} \boldsymbol{x} + \left.\frac{\partial \boldsymbol{H}(\boldsymbol{X},\boldsymbol{U})}{\partial \boldsymbol{U}}\right|_{\substack{\boldsymbol{X}=\boldsymbol{X}_1\\ \boldsymbol{U}=\boldsymbol{U}_1}} \boldsymbol{u}\,.
\end{aligned}$$

Ignoring the steady-state parts, and defining

$$\begin{aligned}
[A] &\triangleq \left.\frac{\partial \boldsymbol{F}(\boldsymbol{X},\boldsymbol{U})}{\partial \boldsymbol{X}}\right|_{\substack{\boldsymbol{X}=\boldsymbol{X}_1\\ \boldsymbol{U}=\boldsymbol{U}_1}}\,, \\
[B] &\triangleq \left.\frac{\partial \boldsymbol{F}(\boldsymbol{X},\boldsymbol{U})}{\partial \boldsymbol{U}}\right|_{\substack{\boldsymbol{X}=\boldsymbol{X}_1\\ \boldsymbol{U}=\boldsymbol{U}_1}}\,, \\
[C] &\triangleq \left.\frac{\partial \boldsymbol{H}(\boldsymbol{X},\boldsymbol{U})}{\partial \boldsymbol{X}}\right|_{\substack{\boldsymbol{X}=\boldsymbol{X}_1\\ \boldsymbol{U}=\boldsymbol{U}_1}}\,, \\
[D] &\triangleq \left.\frac{\partial \boldsymbol{H}(\boldsymbol{X},\boldsymbol{U})}{\partial \boldsymbol{U}}\right|_{\substack{\boldsymbol{X}=\boldsymbol{X}_1\\ \boldsymbol{U}=\boldsymbol{U}_1}}\,,
\end{aligned}$$

the linear state-space equations can be written as

$$\dot{\boldsymbol{x}} = [A]\boldsymbol{x} + [B]\boldsymbol{u}\,, \tag{2.28}$$

$$\boldsymbol{y} = [C]\boldsymbol{x} + [D]\boldsymbol{u}\,. \tag{2.29}$$

Since Eq. (2.20) was formulated as an autonomous ODE, Eqs. (2.28–2.29) form a Linear Time Invariant (LTI) system. The matrix $[A]$ is referred to as the state matrix, the matrix $[B]$ as the control distribution or control effectiveness matrix, the matrix $[C]$ as the output matrix, and the matrix $[D]$ as the carry-through or feed-forward matrix. In this thesis the matrix $[B]$ will be exclusively referred to as the control distribution matrix to avoid confusion with a separate measure of control effectiveness that will be introduced during the control law design.

The 4-tuple $([A], [B], [C], [D])$ is a state-space representation of the linearized dynamics, and is non-unique. This non-uniqueness is easily seen by changing the order of state variables to obtain a different representation of the same dynamics. Given an invertible linear transformation $[T]$ such that a new state $\boldsymbol{z} = [T]\boldsymbol{x}$, one can easily obtain the state-space representation

$$\dot{\boldsymbol{z}} = [T][A][T]^{-1}\boldsymbol{z} + [T][B]\boldsymbol{u}\,, \tag{2.30}$$

$$\boldsymbol{y} = [C][T]^{-1}\boldsymbol{z} + [D]\boldsymbol{u}\,. \tag{2.31}$$

This transformation is useful when changing state variables in LTI models of an aircraft; an example is replacing the body-axis velocities u and w with V_{T} and α.

By taking the Laplace transform of Eqs. (2.28–2.29) the unique transfer function matrix is obtained:

$$[G(s)] = [C]\left(s[I_n] - [A]\right)^{-1}[B] + [D]\,, \tag{2.32}$$

where s is the Laplace variable. The uniqueness of the transfer function can be proven by calculating the transfer function matrix of the transformed system Eqs. (2.30–2.31) and using matrix algebra to arrive at Eq. (2.32). These manipulations are left as an exercise to the reader.

Linear models for conventional aircraft at a steady, level flight condition at near-zero bank angle can be decoupled into two sets of linear models: longitudinal (i.e. pitch axis) and lateral/directional (lat/d, i.e. roll and yaw axes). The longitudinal state vector consists of the variables U, W, Q, and Θ in body-axes, while the lateral/directional state vector consists of the variables V, P, R, Φ, and Ψ in body-axes. In stability/wind axes, the body axis velocities are replaced by V_{T}, α, and β. Conventional longitudinal controls are throttle δ_{T} and elevator δ_{E}, and conventional lat/d controls are aileron δ_{A} and rudder δ_{R} [36]. In the stability axis system the parametric longitudinal linear state-space equations are given by

$$\left\{\begin{matrix}\dot{u}\\ \dot{\alpha}\\ \dot{q}\\ \dot{\theta}\end{matrix}\right\} = \begin{bmatrix} X'_{\mathrm{u}} & X'_{\alpha} & X'_{\mathrm{q}} & -g\cos\Theta_1 \\ Z'_{\mathrm{u}} & Z'_{\alpha} & Z'_{\mathrm{q}} & -g\sin\Theta_1 \\ M'_{\mathrm{u}} & M'_{\alpha} & M'_{\mathrm{q}} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \left\{\begin{matrix}u\\ \alpha\\ q\\ \theta\end{matrix}\right\} + \begin{bmatrix} X'_{\delta_{\mathrm{E}}} & X'_{\delta_{\mathrm{T}}} \\ Z'_{\delta_{\mathrm{E}}} & Z'_{\delta_{\mathrm{T}}} \\ M'_{\delta_{\mathrm{E}}} & M'_{\delta_{\mathrm{T}}} \\ 0 & 0 \end{bmatrix} \left\{\begin{matrix}\delta_{\mathrm{E}}\\ \delta_{\mathrm{T}}\end{matrix}\right\}. \tag{2.33}$$

The "primed" quantities result from decoupling the $\dot{\alpha}$ and $\dot{q}$ equations. The form of the lat/d linear state-space model is

$$\left\{\begin{matrix}\dot{\beta}\\ \dot{p}\\ \dot{r}\\ \dot{\phi}\\ \dot{\psi}\end{matrix}\right\} = \begin{bmatrix} \dfrac{Y_{\beta}}{U_1} & \dfrac{Y_{\mathrm{p}}}{U_1} & 1+\dfrac{Y_{\mathrm{r}}}{U_1} & \dfrac{g\cos\Theta_1}{U_1} & 0 \\ L'_{\beta} & L'_{\mathrm{p}} & L'_{\mathrm{r}} & 0 & 0 \\ N'_{\beta} & N'_{\mathrm{p}} & N'_{\mathrm{r}} & 0 & 0 \\ 0 & 1 & \tan\Theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \left\{\begin{matrix}\beta\\ p\\ r\\ \phi\\ \psi\end{matrix}\right\} + \begin{bmatrix} \dfrac{Y_{\delta_{\mathrm{A}}}}{U_1} & \dfrac{Y_{\delta_{\mathrm{R}}}}{U_1} \\ L'_{\delta_{\mathrm{A}}} & L'_{\delta_{\mathrm{R}}} \\ N'_{\delta_{\mathrm{A}}} & N_{\delta_{\mathrm{R}}} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \left\{\begin{matrix}\delta_{\mathrm{A}}\\ \delta_{\mathrm{R}}\end{matrix}\right\}. \tag{2.34}$$

Similarly to the longitudinal equations Eq. (2.33), the primed terms result from decoupling the $\dot{p}$ and $\dot{r}$ equations. The elements of the $[A]$ and $[B]$ matrices in Eqs. (2.33–2.34) are dimensional stability and control derivatives, respectively, and will be be discussed further in the section on aerodynamic modeling. The linear models developed in this section are useful for the design of linear control systems and estimators.

2.3 Aerodynamic Forces and Moments

As described in the preceding section, there are three aerodynamic forces and three aerodynamic moments acting on an aircraft. In the body axis system, the forces are denoted as $F_{\mathrm{A_x}}$, $F_{\mathrm{A_y}}$, and $F_{\mathrm{A_z}}$, and the moments are denoted as L_{A}, M_{A}, and N_{A}. While aerodynamics are generally written in the stability axis system in terms of drag D, sideforce Y, and lift L, the simulation models used in this these express the coefficients in the body axis system and this section will proceed accordingly.

2.3.1 Aerodynamic Coefficients

The basic modeling approach used in the literature is to define nondimensional force and moment coefficients. For forces, the dynamic pressure

$$\bar{q} = \frac{1}{2}\rho V_{\mathrm{T}}^2 \tag{2.35}$$

and the wing planform area S are used to nondimensionalize:

$$C_{\mathrm{X}} = \frac{F_{\mathrm{A_x}}}{\bar{q}S}\,, \tag{2.36}$$

$$C_{\mathrm{Y}} = \frac{F_{\mathrm{A_y}}}{\bar{q}S}\,, \tag{2.37}$$

$$C_{\mathrm{Z}} = \frac{F_{\mathrm{A_z}}}{\bar{q}S}\,. \tag{2.38}$$

These coefficients are denoted the X-axis, sideforce, and Z-axis coefficients, respectively. Wind tunnel and CFD data may output aerodynamic data in terms of axial and normal force coefficients, where $C_{\mathrm{A}} = -C_{\mathrm{X}}$ and $C_{\mathrm{N}} = -C_{\mathrm{Z}}$. Note that the dynamic pressure accounts for effects due to airspeed (through the V_{T}^2 term) and altitude (through the density ρ). For moments, the dynamic pressure, wing area, and a reference length are used to nondimensionalize. The reference length for the pitching moment is the mean geometric chord length $\bar{c}$ while the reference length for the rolling and yawing moments is the aircraft wingspan b. These reference lengths are shown in Fig. 2.2. The dimensionless moment

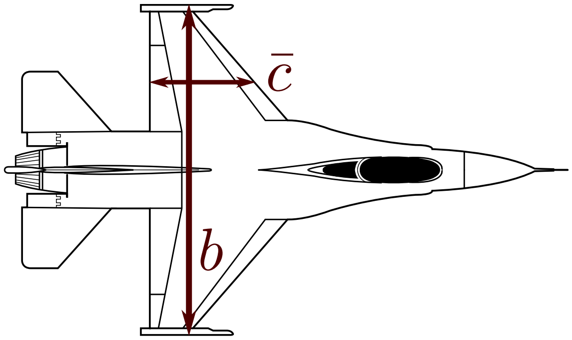


Figure 2.2: Reference lengths for nondimensionalization.

coefficients are therefore given by

$$C_\ell = \frac{L_\mathrm{A}}{\bar{q}Sb}\,, \tag{2.39}$$

$$C_\mathrm{m} = \frac{M_\mathrm{A}}{\bar{q}S\bar{c}}\,, \tag{2.40}$$

$$C_\mathrm{n} = \frac{N_\mathrm{A}}{\bar{q}Sb}\,, \tag{2.41}$$

which are the rolling, pitching, and yawing moment coefficients.

The total aerodynamic moment results from two sources: the pure moments, and the moments generated by the forces. The latter contribution arises from the fact that the aerodynamic forces are generally computed acting through the aerodynamic center and not the center of gravity. Let $\boldsymbol{x}_\mathrm{ac}$ and $\boldsymbol{x}_\mathrm{cg}$ be the position vectors of the aerodynamic center and center of gravity, respectively, expressed in the body axis frame. Then, the total body axis moments can be computed as

$$\begin{Bmatrix} L_\mathrm{A} \\ M_\mathrm{A} \\ N_\mathrm{A} \end{Bmatrix} = \begin{Bmatrix} C_\ell \bar{q}Sb \\ C_\mathrm{m}\bar{q}S\bar{c} \\ C_\mathrm{n}\bar{q}Sb \end{Bmatrix} \bar{q}S + (\boldsymbol{x}_\mathrm{ac} - \boldsymbol{x}_\mathrm{cg}) \times \begin{Bmatrix} C_\mathrm{X}\bar{q}S \\ C_\mathrm{Y}\bar{q}S \\ C_\mathrm{Z}\bar{q}S \end{Bmatrix}\,. \tag{2.42}$$

2.3.2 Component Build-up Method

The six aerodynamic coefficients C_{X}, C_{Y}, C_{Z}, C_{ℓ}, C_{m}, C_{n} are further decomposed in what is known as the component build-up approach. Here, the coefficients are parameterized with the aerodynamic angles α and β, angular rates P, Q, R, control effector positions, and other modeling terms. The terms representing the effect of changes in state variables are commonly termed stability derivatives, and terms representing the effect of control effectors are termed control derivatives. These buildups can be linear approximations of the aircraft aerodynamics valid for quasi-steady flow [40], or can be nonlinear models. Further discussion of aerodynamic models will be limited to nonlinear models; for an in-depth coverage of linear aerodynamic models consult one of Refs. [35, 36].

In general, the aerodynamic coefficients are nonlinear. Nonlinear aerodynamic models are usually expressed in one of two ways. The first and most common, especially with computational aerodynamic prediction software such as HASC2002, is to represent the coefficients as functions of key parameters such as AOA, sideslip angle, and Mach number ($M = V_{\mathrm{T}}/a$, where a is the speed-of-sound) with linear dependencies on other terms. Using this method, the six coefficients can be represented in forms such as

$$C_{\xi} = C_{\xi_{\mathrm{base}}}(\alpha,\beta,M) + C_{\xi_{\mathrm{q}}}(\alpha,\beta,M)\hat{q} + \Delta C_{\xi_{\mathrm{ground}}}(\alpha,\beta,M,H) + \sum_{i} \Delta C_{\xi}(\alpha,\beta,M,\delta_i) \tag{2.43}$$

for $\xi = X, Z, m$ and

$$\begin{aligned} C_{\xi} = C_{\xi_{\mathrm{base}}}(\alpha,\beta,M) + C_{\xi_{\mathrm{p}}}(\alpha,\beta,M)\hat{p} + C_{\xi_{\mathrm{r}}}(\alpha,\beta,M)\hat{r} \\ + \Delta C_{\xi_{\mathrm{ground}}}(\alpha,\beta,M,H) + \sum_{i} \Delta C_{\xi}(\alpha,\beta,M,\delta_i) \end{aligned} \tag{2.44}$$

for $\xi = Y, l, n$. In Eqs. (2.43–2.44), the term $C_{\xi_{\mathrm{base}}}$ describes the baseline aerodynamic effects due to AOA, sideslip, and Mach and the terms for $\hat{p}$, $\hat{q}$, and $\hat{r}$ describe dynamic aerodynamic effects. The dimensionless angular rates $\hat{p}$ and $\hat{r}$ are obtained from p and r by multiplying by $b/(2V_{\mathrm{T}})$, while $\hat{q} = q\bar{c}/(2V_{\mathrm{T}})$. The term $\Delta C_{\xi_{\mathrm{ground}}}$ models the incremental change in the coefficient due to ground effect (h being the altitude of the aircraft). Finally, the terms $\Delta C_{\xi}(\alpha, \beta, M, \delta_i)$ describe the incremental change in the coefficient due to deflection of the control effectors (δ_i, $i = 1 \ldots m$). In general, the control effectors are coupled, with deflections changing the effectiveness of other effectors.

Equations (2.43–2.44) contain several assumptions. Effects on the longitudinal axis coefficients due to the AOA rate ($\dot{\alpha}$) are assumed to be included in the pitch rate terms. The longitudinal equations are assumed to depend on AOA, sideslip angle, pitch rate, and longitudinal controls (conventionally elevator). The lateral/directional terms are assumed to depend on AOA, sideslip angle, roll rate, yaw rate, and lat/d control effectors, traditionally aileron and rudder. Generally, AOA and sideslip effects are nonlinear, rate terms are linear in the rates themselves, and control terms are linear in the controls. Mach effects are generally neglected for subsonic aircraft [40].

Using this representation the stability and control derivatives are commonly implemented in the form of multidimensional tables. These tables are generally the output of computational aerodynamic methods or wind tunnel data, which produce values of the coefficients at a finite number of AOA and sideslip angles, Mach numbers, and control effector deflections. Values are then obtained using interpolation. These selected values of the independent variables are known as breakpoints. Aerodynamic tables are generally large. Using the baseline aerodynamic term $C_{\xi_{\mathrm{base}}}(\alpha, \beta, M)$ as an example and denoting the numbers of breakpoints as n_{α}, n_{β}, and n_{M}, the size of a table is

$$n_{\mathrm{bytes}} \times n_{\alpha} \times n_{\beta} \times n_{\mathrm{M}} \, , \tag{2.45}$$

where n_{bytes} is the number of bytes required to represent the data type of a table entry, usually a `float` or `double`. The size of the aerodynamics database is important when using dynamic inversion control laws (see Section 4), as the tables must be stored onboard the aircraft for use by the flight control system. The model stored in the aircraft's digital flight control computer is referred to as the *onboard model* or OBM.

The second approach to modeling nonlinear aerodynamic coefficients is to represent the coefficients with constant stability and control derivatives multiplied by nonlinear functions of the states and controls. These coefficients are obtained by curve-fitting the aerodynamic tables or by using parameter identification methods such as MMLE on flight data [41, 42]. An example of this approach is the Generic Nonlinear Aerodynamic (GNA) model developed at NASA Langley Research Center [43]. The GNA model was developed by fitting a candidate pool of basis functions formed by combinations of the state variables up to fourth order terms against high-fidelity nonlinear simulations. The component build-up terms for the six coefficients in the GNA model are shown in Eqs. (2.46–2.51).

$$\begin{aligned} C_{\mathrm{D}} = {} & C_{\mathrm{D}_1} + C_{\mathrm{D}_\alpha}\alpha + C_{\mathrm{D}_{\alpha\mathrm{q}}}\alpha\hat{q} + C_{\mathrm{D}_{\alpha\delta_\mathrm{E}}}\alpha\delta_\mathrm{E} + C_{\mathrm{D}_{\alpha^2}}\alpha^2 + C_{\mathrm{D}_{\alpha^2\mathrm{q}}}\alpha^2\hat{q} \\ & + C_{\mathrm{D}_{\alpha^2\delta_\mathrm{E}}}\alpha^2\delta_\mathrm{E} + C_{\mathrm{D}_{\alpha^3}}\alpha^3 + C_{\mathrm{D}_{\alpha^3\mathrm{q}}}\alpha^3\hat{q} + C_{\mathrm{D}_{\alpha^4}}\alpha^4 \, , \end{aligned} \tag{2.46}$$

$$C_{\mathrm{Y}} = C_{\mathrm{Y}_\beta}\beta + C_{\mathrm{Y}_\mathrm{p}}\hat{p} + C_{\mathrm{Y}_\mathrm{r}}\hat{r} + C_{\mathrm{Y}_{\delta_\mathrm{A}}}\delta_\mathrm{A} + C_{\mathrm{Y}_{\delta_\mathrm{R}}}\delta_\mathrm{R} \, , \tag{2.47}$$

$$C_{\mathrm{L}} = C_{\mathrm{L}_1} + C_{\mathrm{L}_\alpha}\alpha + C_{\mathrm{L}_\mathrm{q}}\hat{q} + C_{\mathrm{L}_{\delta_\mathrm{E}}}\delta_\mathrm{E} + C_{\mathrm{L}_{\alpha\mathrm{q}}}\alpha\hat{q} + C_{\mathrm{L}_{\alpha^2}}\alpha^2 + C_{\mathrm{L}_{\alpha^3}}\alpha^3 + C_{\mathrm{L}_{\alpha^4}}\alpha^4 \, , \tag{2.48}$$

$$C_{\ell} = C_{\ell_\beta}\beta + C_{\ell_\mathrm{p}}\hat{p} + C_{\ell_\mathrm{r}}\hat{r} + C_{\ell_{\delta_\mathrm{A}}}\delta_\mathrm{A} + C_{\ell_{\delta_\mathrm{R}}}\delta_\mathrm{R} \, , \tag{2.49}$$

$$\begin{aligned} C_{\mathrm{m}} = {} & C_{\mathrm{m}_1} + C_{\mathrm{m}_\alpha}\alpha + C_{\mathrm{m}_\mathrm{q}}\hat{q} + C_{\mathrm{m}_{\delta_\mathrm{E}}}\delta_\mathrm{E} + C_{\mathrm{m}_{\alpha\mathrm{q}}}\alpha\hat{q} + C_{\mathrm{m}_{\alpha^2\mathrm{q}}}\alpha^2\hat{q} + C_{\mathrm{m}_{\alpha^2\delta_\mathrm{E}}}\alpha^2\delta_\mathrm{E} \\ & + C_{\mathrm{m}_{\alpha^3\mathrm{q}}}\alpha^3\hat{q} + C_{\mathrm{m}_{\alpha^3\delta_\mathrm{E}}}\alpha^3\delta_\mathrm{E} + C_{\mathrm{m}_{\alpha^4}}\alpha^4 \, , \end{aligned} \tag{2.50}$$

$$C_{\mathrm{n}} = C_{\mathrm{n}_\beta}\beta + C_{\mathrm{n}_\mathrm{p}}\hat{p} + C_{\mathrm{n}_\mathrm{r}}\hat{r} + C_{\mathrm{n}_{\delta_\mathrm{A}}}\delta_\mathrm{A} + C_{\mathrm{n}_{\delta_\mathrm{R}}}\delta_\mathrm{R} + C_{\mathrm{n}_{\beta^2}}\beta^2 + C_{\mathrm{n}_{\beta^3}}\beta^3 \, . \tag{2.51}$$

The advantage of this approach is that fewer parameters are required to describe the aerodynamic model. This is especially useful for onboard models on memory-constrained flight computers. Again using the example of the GNA model, only 45 parameters are required to describe the aerodynamics, a significant reduction compared to the lookup table approach. In general, the nonlinear aerodynamics can be represented using a candidate pool of regressors from the state and control variables and a suitable curve-fitting method such as least-squares or neural networks. An example of this using a least-squares approach can be found in Ref. [44].

3. NONLINEAR 6-DOF SIMULATION MODEL AND BASELINE COMMAND AUGMENTATION SYSTEM

3.1 Introduction

This section describes the basic nonlinear six degree-of-freedom (6-DOF) model used for simulations in this thesis. The model is based off of a low speed (less than Mach 0.6) model of a Lockheed Martin F-16 jet fighter. This model is well-known due to its inclusion in the classic textbook by Stevens and Lewis [31]. A Simulink model is built around this simulation model to serve as the standard simulation for work done in this thesis, with the F-16 serving as a "surrogate" UAS.

The section is organized as follows: first, the custom Simulink model is presented. This includes the model layout, actuator models, and disturbance models. Second, open loop simulation results for a series of doublet inputs are presented to demonstrate the inherent dynamics of the model. These input sets will be used for variable stability system simulation results in Section 7. Finally, an overview of a basic design for a 3-axis Command Augmentation System (CAS) is presented. This CAS design is used as a baseline flight control system design to allow comparison between the different control laws presented in this thesis.

3.2 F-16 Simulink Model

As previously mentioned, the aircraft model chosen is an F-16 model popularized by Ref. [31], in which it is distributed as a FORTRAN simulation code. This model uses a simplified aerodynamic database based off of NASA wind tunnel tests found in Ref. [45], and is valid for low (subsonic) speeds. The Simulink model created for this thesis is based off of a MATLAB conversion of the original FORTRAN program by Ying Huo from the University of Southern California, which also includes an S-function implementing the F-16 dynamics for use with Simulink. This simulation is distributed on the website accompanying the book *Adaptive Control Tutorial* [46] and is available for noncommercial use.

3.2.1 Model Description

This section describes the aerodynamic database and propulsion, actuator, and environmental models. Additionally, mass properties and geometry are listed in Table 3.1. In the table, the parameter h_{eng} is the engine angular momentum, assumed constant.

Table 3.1: Nominal F-16 model mass properties and geometry.

Parameter	**Value**
m (lbf)	20500
I_{xx} (slug·ft^2)	9456
I_{yy} (slug·ft^2)	55814
I_{zz} (slug·ft^2)	63100
I_{xz} (slug·ft^2)	982
S_{ref} (ft^2)	300
b_{ref} (ft)	30
$\bar{c}$ (ft)	11.32
$\bar{x}_{\mathrm{cg}}$ ($\%\bar{c}$)	35
h_{eng} (slug·ft^2/s)	160

3.2.1.1 Aerodynamic Database

The F-16 aerodynamic database is provided as lookup tables for aerodynamic coefficients in the body axis system, i.e. C_{X}, C_{Y}, C_{Z}, C_{ℓ}, C_{m}, and C_{n}. The aerodynamic coefficients are functions of angle-of-attack, angle-of-sideslip, nondimensional body axis roll rates, and virtual aileron, elevator, and rudder. Angle-of-attack breakpoints are from $-10°$ to $45°$ in $5°$ increments, while angle-of-sideslip breakpoints are either from $0°$ to

30° in 5° increments for symmetric terms or from -30° to 30° in 10° increments for symmetric terms. The aerodynamic coefficients are coupled between the longitudinal and lateral/directional axes. The component build-ups for the six aerodynamic coefficients are given by Eq. (3.1).

$$\begin{aligned}
C_{\mathrm{X}}(\alpha, \hat{q}, \delta_{\mathrm{E}}) &= C_{\mathrm{X}}(\alpha, \delta_{\mathrm{E}}) + C_{\mathrm{X_q}}(\alpha)\hat{q}\,, \\
C_{\mathrm{Y}}(\beta, \hat{p}, \hat{r}, \delta_{\mathrm{A}}, \delta_{\mathrm{R}}) &= C_{\mathrm{Y}_\beta}\beta + C_{\mathrm{Y_p}}(\alpha)\hat{p} + C_{\mathrm{Y_r}}(\alpha)\hat{r} + C_{\mathrm{Y}_{\delta_\mathrm{A}}}\delta_{\mathrm{A}} + C_{\mathrm{Y}_{\delta_\mathrm{R}}}\delta_{\mathrm{R}}\,, \\
C_{\mathrm{Z}}(\alpha, \beta, \hat{q}, \delta_{\mathrm{E}}) &= C_{\mathrm{Z}}(\alpha)(1-\beta^2) + C_{\mathrm{Z_q}}(\alpha)\hat{q} + C_{\mathrm{Z}_{\delta_\mathrm{E}}}\delta_{\mathrm{E}}\,, \\
C_{\ell}(\alpha, \beta, \hat{p}, \hat{r}, \delta_{\mathrm{A}}, \delta_{\mathrm{R}}) &= C_{\ell}(\alpha, \beta) + C_{\ell_\mathrm{p}}(\alpha)\hat{p} + C_{\ell_\mathrm{r}}(\alpha)\hat{r} + C_{\ell_{\delta_\mathrm{A}}}(\alpha, \beta)\delta_{\mathrm{A}} + C_{\ell_{\delta_\mathrm{R}}}(\alpha, \beta)\delta_{\mathrm{R}}\,, \\
C_{\mathrm{m}}(\alpha, \hat{q}, \delta_{\mathrm{E}}) &= C_{\mathrm{m}}(\alpha, \delta_{\mathrm{E}}) + C_{\mathrm{m_q}}(\alpha)\hat{q}\,, \\
C_{\mathrm{n}}(\alpha, \beta, \hat{p}, \hat{r}, \delta_{\mathrm{A}}, \delta_{\mathrm{R}}) &= C_{\mathrm{n}}(\alpha, \beta) + C_{\mathrm{n_p}}(\alpha)\hat{p} + C_{\mathrm{n_r}}(\alpha)\hat{r} + C_{\mathrm{n}_{\delta_\mathrm{A}}}(\alpha, \beta)\delta_{\mathrm{A}} + C_{\mathrm{n}_{\delta_\mathrm{R}}}(\alpha, \beta)\delta_{\mathrm{R}}\,.
\end{aligned} \tag{3.1}$$

Figures 3.1–3.3 plot the nonlinear terms of the build-up equations. Note that some plots are symmetric with sideslip while others are not. Several terms are linear; these are shown in Table 3.2.

Table 3.2: F-16 linear stability and control derivatives.

Parameter	**Value**
C_{Y_β}	$-0.02/\mathrm{rad}$
$C_{\mathrm{Y}_{\delta_\mathrm{A}}}$	$0.00105/^\circ$
$C_{\mathrm{Y}_{\delta_\mathrm{R}}}$	$0.00287/^\circ$
$C_{\mathrm{Z}_{\delta_\mathrm{E}}}$	$-0.0076/^\circ$

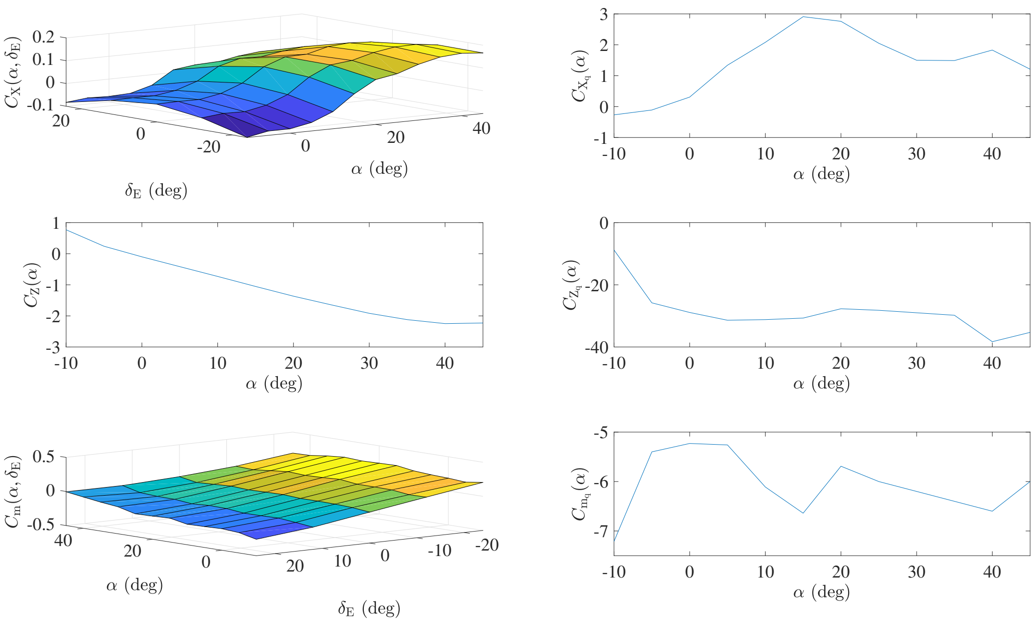


Figure 3.1: Nonlinear build-up terms for C_{X}, C_{Z}, and C_{m}.

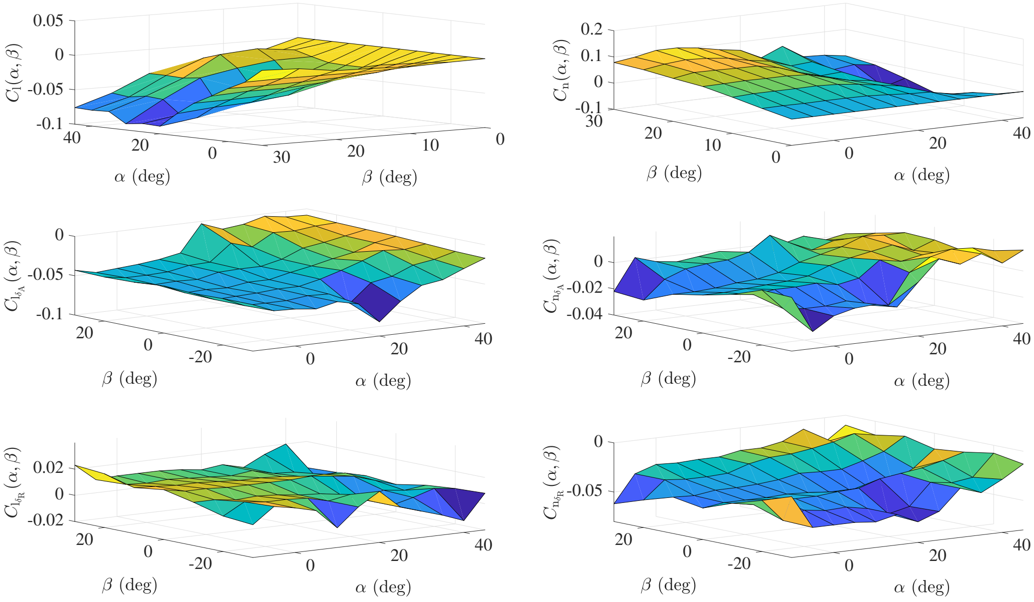


Figure 3.2: Nonlinear build-up terms for C_ℓ, and C_{n}.

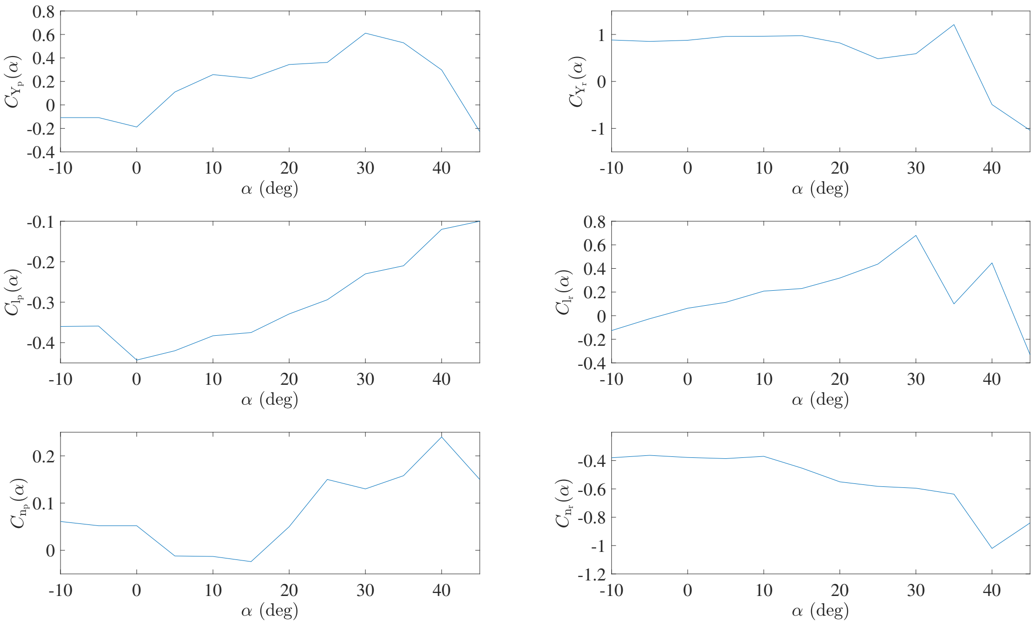


Figure 3.3: Nonlinear damping derivatives for C_{Y}, C_{ℓ}, and C_{n}.

3.2.1.2 Engine Model

The F-16 model contains a simple engine model with first-order dynamics with throttle gearing and thrust modeled as a function of engine power, altitude, and Mach number. The thrust data contains data for Mach numbers from 0 to 1 and altitudes from 0 to $50{,}000\,\mathrm{ft}$. Three power levels are modeled: idle, military, and maximum; these data correspond roughly to throttle settings of 0%, 50%, and 100%, respectively. The engine dynamics are discussed in detail in Appendix A.7 of Ref. [31], which the interested reader is invited to consult.

3.2.1.3 Actuator Model

Flight simulations generally approximate actuator dynamics using first or second order linear models. A first-order approximation is generally useful, which leads to the following servo model, Eq. (3.2):

$$\dot{\delta} = \omega_\delta(\delta_{\text{cmd}} - \delta)\,. \tag{3.2}$$

Here ω_δ is the actuator bandwidth, which can be expressed in terms of a time constant as

$$\tau_\delta = \frac{1}{\omega_\delta}\,.$$

The servos have rate and position limits such that $\delta_{\min} \leqslant \delta \leqslant \delta_{\max}$ and $\dot{\delta}_{\min} \leqslant \dot{\delta} \leqslant \dot{\delta}_{\max}$. Note that the provided rate limits are the no-load limits and that nonlinearities due to hinge moments are ignored. These limits are implemented using a saturation function on the computed rate and integral as shown in Fig. 3.4.

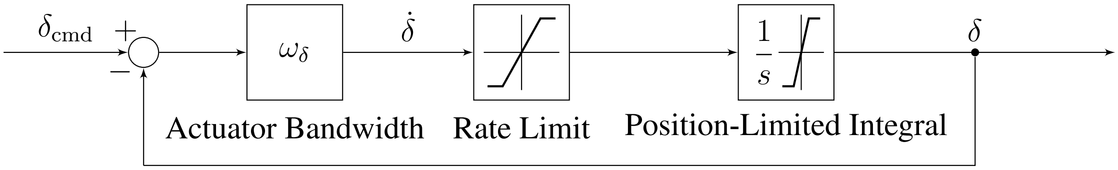


Figure 3.4: Servo model block diagram.

Actuator failures can be modeled as Eq. (3.3),

$$\delta_{\text{applied}} = e\delta_{\text{demanded}} + d \tag{3.3}$$

where δ_{demanded} is the demanded actuator position, δ_{applied} is the actual actuator position, and e and d are parameters used to model different failure types [27]. Common actuator failure types are:

- in-place failures: the actuator remains frozen in its current position
- hardover failures: the actuator remains frozen at one of its limits

- floating failures: the actuator moves freely with the airmass
- partial loss-of-effectiveness: the actuator cannot produce its nominal effect

Table 3.3 lists the values of e and d used to model these types of failures.

Table 3.3: Actuator failure types.

Failure Type	$\boldsymbol{e}$	$\boldsymbol{d}$
in-place	0	$\delta_{\min} \leqslant d \leqslant \delta_{\max}$
hardover	0	$\delta_{\min}$ or $\delta_{\max}$
floating	0	0
partial effectiveness	$0 < e < 1$	0

The F-16 model has three aerodynamic pseudo-controls for aileron, elevator, and rudder. Actuator time constants and position and rate limits for each surface are shown in Table. 3.4.

3.2.1.4 Disturbance Models

Wind and gust disturbances are included in the model using blocks in the Aerospace Blockset. To accommodate disturbances, the original S-function is modified to accept additional inputs for body axis translational and angular velocity gusts. Specifications of

Table 3.4: F-16 actuator specifications

Actuator	Time Constant (s)	Position Limit (deg)	Rate Limit (deg/s)
Aileron	0.0495	± 21.5	± 52.0
Elevator	0.0769	± 25.0	± 60.0
Rudder	0.0495	± 30.0	± 120.0

the gust and shear models are based on MIL-F-8785C [47]. These models are described in Appendix C.

3.2.2 Simulink Model Layout

The Simulink model is organized into subsystems, with blocks for the airframe, actuators, inputs, outputs, and control laws. The different subsystems common to the simulation models are described below. Subsystems are described in approximate "left-to-right" order based on the block diagram. Figure 3.5 shows the overall layout of the Simulink block diagram.

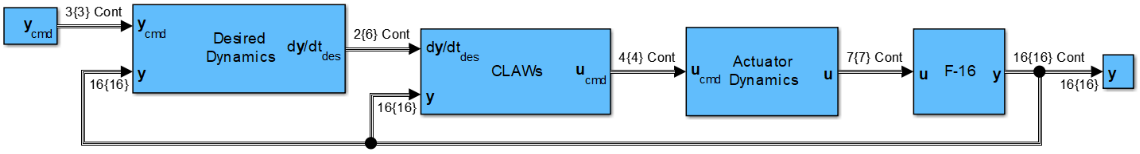


Figure 3.5: High level block diagram of 6-DOF simulation.

3.2.2.1 Command Generator Subsystem

The command generator subsystem generates reference inputs for the control systems. Inputs are specified using either recorded control inceptor commands from piloted simulation using X-Plane® 10 or are generated using built-in Simulink sources such as step inputs, sinusoids, or doublets. For ease of use a custom doublet block is created as part of a simulation library for this thesis.

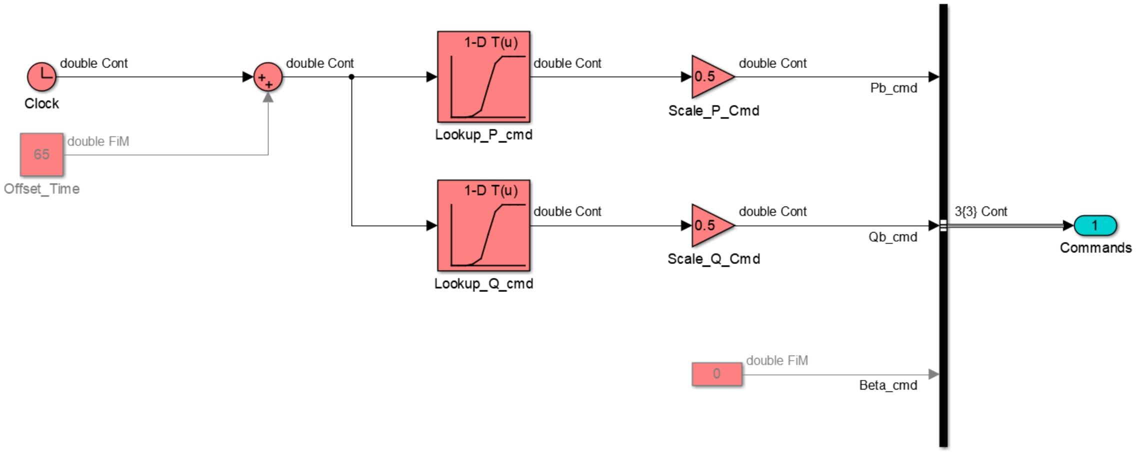


Figure 3.6: Command generation subsystem.

3.2.2.2 Desired Dynamics Subsystem

The desired dynamics (or regulator) subsystem implements different models to generate commanded rates for controlled variables for dynamic inversion controllers. Different types of desired dynamics are discussed in-depth in Ref. [48]. This subsystem is in particular where the reference models for the in-flight simulator designs discussed in Section 7 are implemented. The desired dynamics for the example flight control system design used in this thesis are discussed in detail in Section 3.4; the implementation of these dynamics is shown in Fig. 3.7.

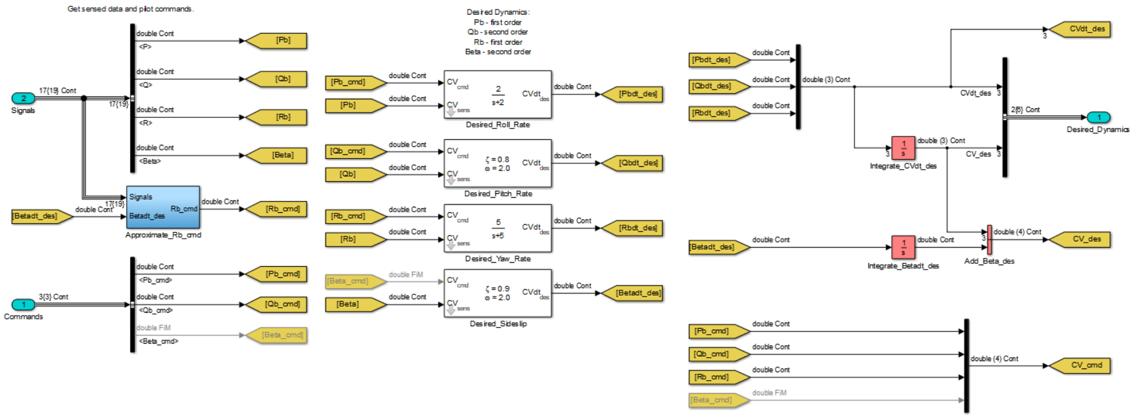


Figure 3.7: Desired dynamics subsystem.

3.2.2.3 *Various Control Law Subsystems*

Various subsystems are used to implement control laws presented in this thesis. These include subsystems for inner-loop control and robust outer-loop controls where needed. For adaptive controllers a subsystem is used to implement adaptive laws, and for $\mathcal{L}_1$ adaptive controllers a state predictor subsystem is required. Figure 3.8 shows an example block diagram for an inner loop nonlinear dynamic inversion control law.

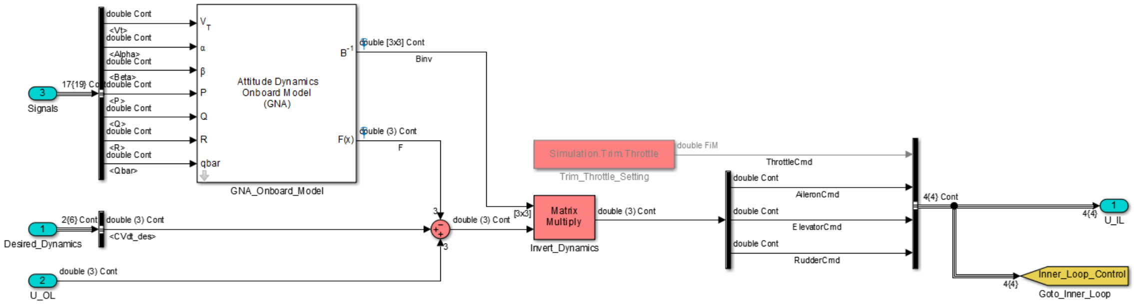


Figure 3.8: Example nonlinear dynamic inversion control law subsystem.

3.2.2.4 Actuator Subsystem

The actuator subsystem, shown in Fig. 3.9, implements first order actuator dynamics as described in Section 3.2.1.3 for the aerodynamic control effectors. The subsystem additionally implements limiting on the throttle command to the closed interval [0, 1].

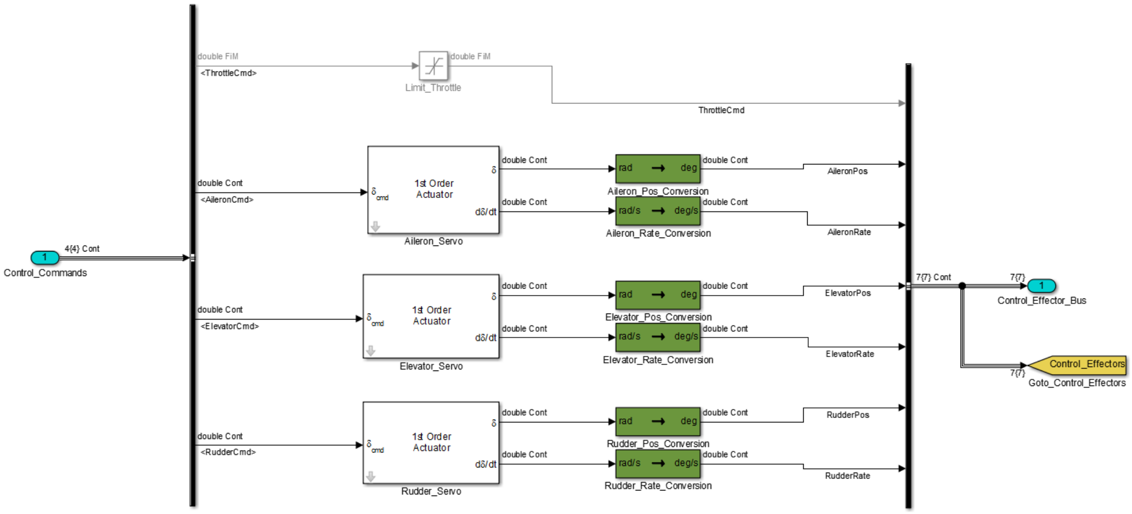


Figure 3.9: Actuator subsystem.

3.2.2.5 Airframe Subsystem

The base airframe subsystem consists of the S-function implementing the F-16 dynamics, along with a masked subsystem for exogenous disturbances. The subsystem takes a control effector bus containing signals for the throttle, aileron, elevator, and rudder deflections as an input, and outputs the aircraft states. This system is shown in Fig. 3.10.

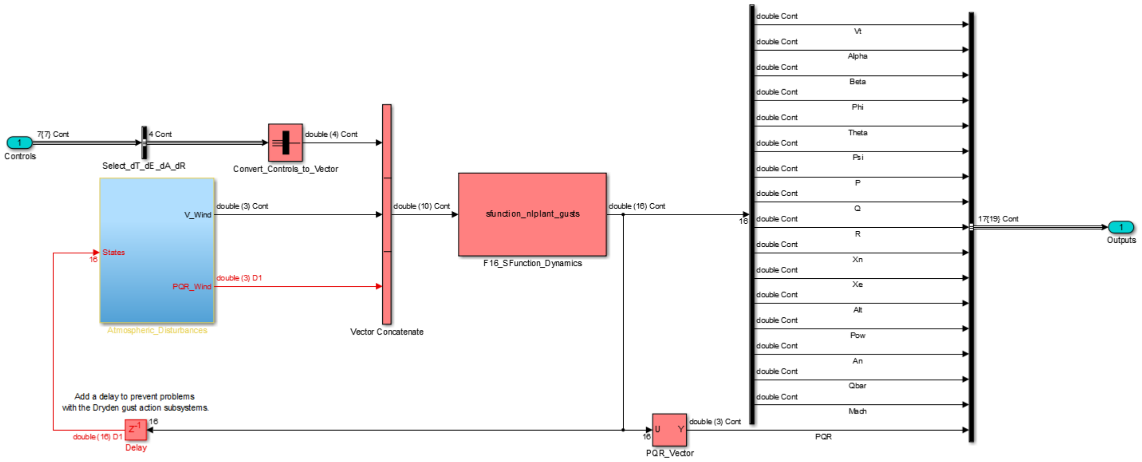


Figure 3.10: Airframe subsystem.

3.2.2.6 Output Subsystem

The output subsystem (Fig, 3.11) records all signals of interest for post-simulation analysis. This subsystem is built using native Simulink logging functionality, and supports real-time plotting of signals using the Simulink Data Inspector in recent versions of MATLAB.

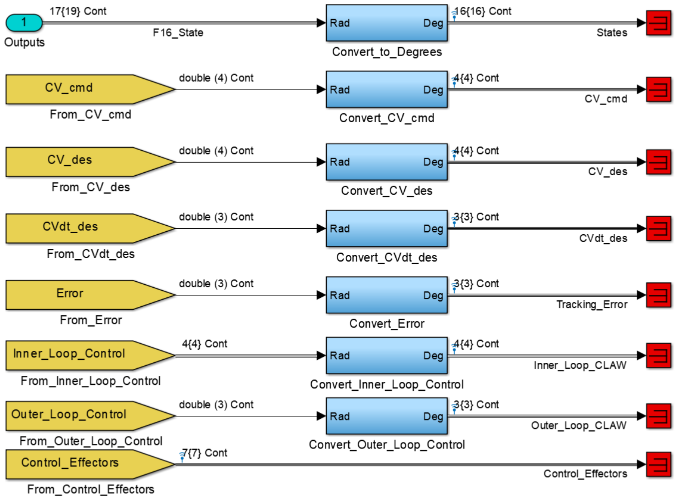


Figure 3.11: Output signals.

3.3 Open Loop Simulation Results

Open loop simulation results are obtained by perturbing the aerodynamic control effectors (aileron, elevator, rudder) by $2°$ doublets lasting a total of $2\,\mathrm{s}$. The throttle is held constant at its trimmed position, and the doublets are added relative to the trimmed effector positions. Figure 3.12 plots the open-loop response from the trimmed flight condition $V_{\mathrm{T}_1} = 750\,\mathrm{ft/s}$ at $20{,}000\,\mathrm{ft}$, showing the airspeed, AOA, sideslip, angular rates, Euler angles, and effector commands and positions.

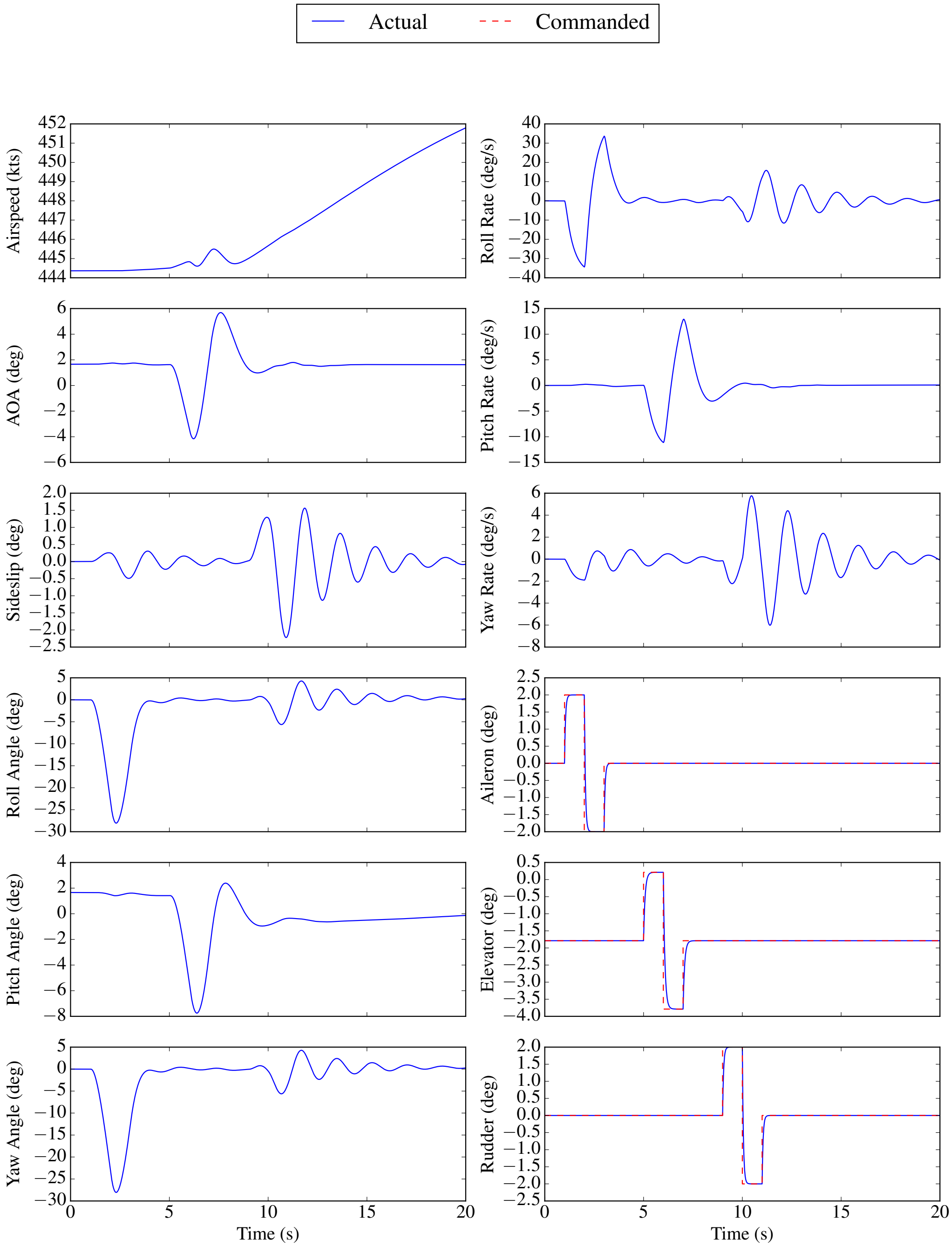


Figure 3.12: F-16 simulation open-loop response.

3.4 Command Augmentation System Design

A Command Augmentation System is a type of flight control system that interprets pilot control inceptor inputs as commanded state variables or functions of state variables. This is opposed to Stability Augmentation Systems (SAS), in which the pilot commands control effector positions which are then augmented with a feedback control signal to modify the inherent dynamics of the aircraft [31]. A CAS is a *full authority* system, where the control law has complete control over the effector commands, while a SAS is a *limited authority* system which can only command a subset of the effectors' deflection. While not required to implement a CAS, the popularity of fly-by-wire systems has seen the CAS eclipse the SAS as the primary flight control system architecture.

As a result, a CAS is the system of choice for the example control design presented herein. This leads to the problem of selection of controlled variables. Reference [49] is perhaps the most complete reference for flight control system design available in the open literature, and provides guidelines for the selection of controlled variables largely based on work done at Honeywell and Lockheed Martin. The report defines roll, pitch, and yaw control variables that combine different states in order to achieve decoupled responses that are suitable for flight at low and high dynamic pressures. However, since this CAS design is primarily intended as an example a simpler approach is chosen. The controlled variables are selected as body axis roll rate P, body axis pitch rate Q, and sideslip angle β. The P and Q control variables are used to maneuver the aircraft, while β is commanded zero to maintain coordinated flight, as $\beta \approx 0$ is a close proxy for $n_{\mathrm{y}} \approx 0$. This corresponds to flying "feet on the floor", i.e. the flight control system controls the directional axis and the pilot does not use the rudder. For the β commands simplified $\dot{\beta}$ dynamics are used to obtain body axis yaw rate R commands based on the desired sideslip angle. Consider the

$\dot{\beta}$ equation of motion:

$$\begin{aligned}\dot{\beta} = \frac{1}{V_{\mathrm{T}}}\bigg[\frac{D}{m}\sin\beta + \frac{Y}{m}\cos\beta + g\left(\cos\alpha\sin\beta\sin\theta + \cos\beta\sin\phi\cos\theta\right.\\ \left. - \sin\alpha\sin\beta\cos\phi\cos\theta\right) + P\sin\alpha - R\cos\alpha\bigg]\,.\end{aligned}$$

Neglecting the effects of the forces D and Y and assuming that $\beta \approx 0$ results in the simplified relation

$$\dot{\beta} \approx P\sin\alpha - R\cos\alpha + \frac{g}{V_{\mathrm{T}}}\sin\phi\cos\theta\,,$$

which can be solved to obtain the yaw rate command:

$$R_{\mathrm{cmd}} = P\tan\alpha - \dot{\beta}_{\mathrm{des}}\sec\alpha + \frac{g}{V_{\mathrm{T}}}\sin\phi\cos\theta\sec\alpha\,, \tag{3.4}$$

where $\dot{\beta}_{\mathrm{des}}$ is the desired sideslip rate. The approach in Eq. (3.4) is an approximate dynamic inversion (see Section 4) of the $\dot{\beta}$ dynamics, and serves as a slow inversion loop compared to fast inversion loops for the angular rates. Yaw rate commands are used since the aircraft rotational equations of motion are used for the control law synthesis. Figure 3.13 is a high-level block diagram of the CAS design. In the figure $\boldsymbol{CV}$ is a vector of control variables, $\boldsymbol{u}$ is the control vector, $\boldsymbol{\delta}$ is the control effector vector, and the “des”, “cmd”, and “sens” subscripts refer to desired, commanded, and sensed values, respectively. The “CLAWs” block implements the control laws, the “EB” block is the effector blender/control allocator, and the “ACT” block is the actuator dynamics.

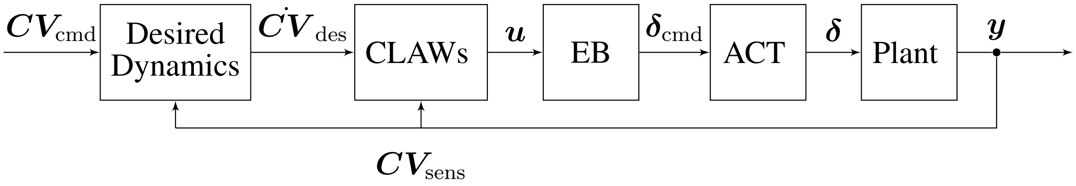


Figure 3.13: Example command augmentation system block diagram.

In a piloted aircraft, the right hand inceptor (i.e. sidestick) would be used to generate P and Q commands via lateral and longitudinal deflections, respectively. The left hand inceptor (i.e. throttle) would be used by the pilot to control the engine. For batch simulation, the P and Q commands are generated using recorded flight simulation data or various function generators, while the throttle is commanded by a simple autothrottle system described further in Section 3.4.6. The example maneuver used for the CAS simulation examples begins with the aircraft in trimmed, level flight at $444\,\mathrm{kt}$ at $20{,}000\,\mathrm{ft}$ with the CG location at 35% chord. The aircraft then rolls inverted and pulls into a dive, followed by a slow roll back to wings level. Figure 3.14 shows time histories of the pilot commands recorded using the X-Plane flight simulator.

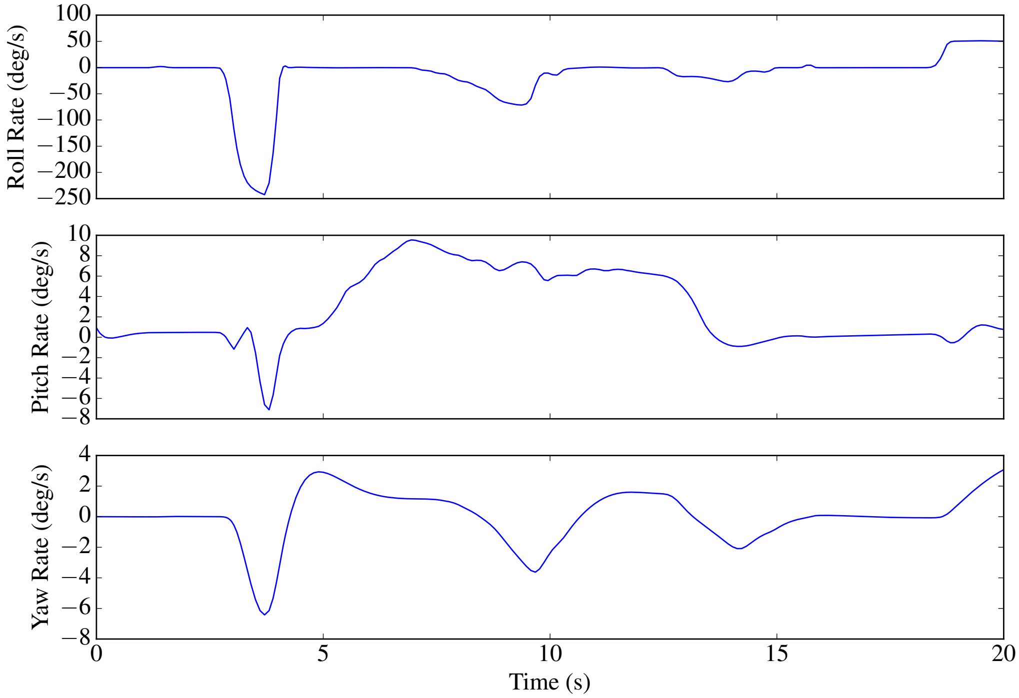


Figure 3.14: Pilot angular rate commands from X-Plane simulation.

3.4.1 Design to Flying Qualities

The CAS design used in this thesis is based on a "design to flying qualities" approach. This approach uses model reference control techniques and designs the reference models to achieve the desired Level 1 flying qualities, usually based off of MIL-STD-1797 [50] or the older MIL-F-8785C [47]. The former introduces specifications for highly-augmented airplanes but has restricted distribution. Low order systems are used so that the reference models have the desired time constants, damping ratios, and natural frequencies, as appropriate. For the P, Q, β design introduced in this section, first and second order dynamics are specified. Desired roll and yaw rate responses are first order, while the pitch rate and sideslip responses are second order.

3.4.1.1 First Order Desired Dynamics

The first order desired dynamics are specified using the following form:

$$\dot{CV}_{\text{des}} = K\left(CV_{\text{cmd}} - CV_{\text{sens}}\right)\,, \tag{3.5}$$

The form Eq. (3.5) allows the use of sensor feedback in the computation of the desired dynamics. As the controller drives the control variable to the desired value, the closed-loop desired dynamics become

$$\frac{CV_{\text{des}}(s)}{CV_{\text{cmd}}(s)} = \frac{K}{s+K}$$

in the frequency domain and are immediately obvious as a classical first-order system. The gain K should be set to the reciprocal of the desired time constant:

$$K = \frac{1}{\tau}\,.$$

3.4.1.2 Second Order Desired Dynamics

Second order desired dynamics allow the specification of a desired damping ratio and natural frequency. To develop a form of second order desired dynamics in a form that allows sensor feedback similar to Eq. (3.5) start with the classical second order system:

$$\ddot{CV} + 2\zeta_{\text{des}}\omega_{\text{des}}\dot{CV} + \omega_{\text{des}}^2 CV = \omega_{\text{des}}^2 CV_{\text{cmd}}\,.$$

An error term can be introduced by moving the CV term to the right hand side:

$$\ddot{CV} + 2\zeta_{\text{des}}\omega_{\text{des}}\dot{CV} = \omega_{\text{des}}^2\left(CV_{\text{cmd}} - CV\right)\,.$$

The left hand side may be written in terms of the control variable rate as

$$(s + 2\zeta_{\text{des}}\omega_{\text{des}})\,\dot{CV} = \omega_{\text{des}}^2\,(CV_{\text{cmd}} - CV)\ ,$$

which then leads to the desired dynamics

$$\dot{CV}_{\text{des}} = \frac{\omega_{\text{des}}^2}{s + 2\zeta_{\text{des}}\omega_{\text{des}}}\,[CV_{\text{cmd}} - CV_{\text{sens}}]\ . \tag{3.6}$$

By working the above manipulations in the reverse order Eq. (3.6) can be shown to have a classic second order response when the loop is closed. For more information on various forms of desired dynamics the reader is invited to consult Ref. [48]; Chapter 8 of Ref. [51] is also particularly useful.

3.4.2 Roll Axis Desired Dynamics

The roll axis desired dynamics are specified using Eq. (3.5) as

$$\dot{P}_{\text{des}} = K_{\text{P}}(P_{\text{cmd}} - P_{\text{sens}})\,,$$

where $K_{\text{p}} = \tau_{\text{R}}^{-1}$ is chosen based off of the roll mode time constant requirements. For a Class IV aircraft at the air combat phase (CO), MIL-F-8785C specifies that $\tau_{\text{R}} < 1\,\text{s}$. In this thesis, $\tau_{\text{R}} = 0.5\,\text{s}$ is used. Figure 3.15 is a block diagram of the roll axis desired dynamics.

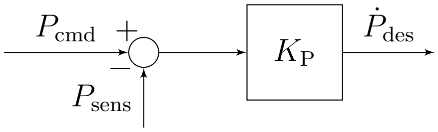


Figure 3.15: Roll axis desired dynamics block diagram.

3.4.3 Pitch Axis Desired Dynamics

The pitch axis dynamics are second order and are based off of the short period mode requirements in MIL-F-8785C:

$$\dot{Q}_{\text{des}} = \frac{\omega_{\text{sp}}^2}{s + 2\zeta_{\text{sp}}\omega_{\text{sp}}} \left[Q_{\text{cmd}} - Q_{\text{sens}}\right] .$$

MIL-F-8785C provides minimum damping ratios and natural frequencies based on the value of n/α. A damping ratio $\zeta_{\text{sp}} = 0.8$ and natural frequency $\omega_{\text{sp}} = 2\,\text{rad/s}$ are chosen for this thesis. Figure 3.16 is a block diagram of the roll axis desired dynamics.

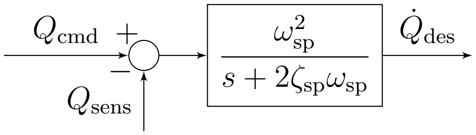


Figure 3.16: Pitch axis desired dynamics block diagram.

3.4.4 Yaw Axis Desired Dynamics

The yaw axis desired dynamics are the most complicated as the approach used in this thesis combines second order desired dynamics for sideslip, slow inversion of sideslip to obtain yaw rate commands, and finally yaw rate desired dynamics. First, the sideslip desired dynamics are given by

$$\dot{\beta}_{\text{des}} = \frac{\omega_{\text{dr}}^2}{s + 2\zeta_{\text{dr}}\omega_{\text{dr}}} \left[\beta_{\text{cmd}} - \beta_{\text{sens}}\right] .$$

The sideslip desired dynamics are based off of the MIL-F-8785C Dutch roll mode requirements. In the examples in this thesis $\zeta_{\text{dr}} = 0.9$ and $\omega_{\text{dr}} = 2\,\text{rad/s}$.

The sideslip desired dynamics $\dot{\beta}_{\text{des}}$ are then converted into yaw rate commands by a dynamic inversion process. The inversion is given by the previously shown Eq. (3.4); in this equation, R_{cmd} is effectively the "control" resulting from the inversion. The resulting yaw rate command is then fed to a first order system for the yaw rate desired dynamics, as shown in Fig. 3.17.

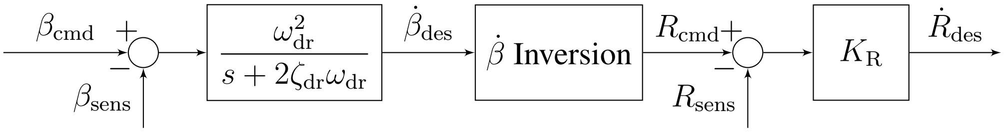


Figure 3.17: Yaw axis desired dynamics/slow inversion block diagram.

3.4.5 Summary of Desired Dynamics

Table 3.5 presents the time constants, damping ratios, and natural frequencies for the roll rate, pitch rate, yaw rate, and sideslip angle desired dynamics, as described previously.

Table 3.5: Desired dynamics for example CAS design.

Design Parameter	**Value**
τ_{P}	$0.5\,\text{s}$
ζ_{Q}	0.8
ω_{Q}	$2\,\text{rad/s}$
τ_{R}	$0.2\,\text{s}$
ζ_{β}	0.9
ω_{β}	$2\,\text{rad/s}$

3.4.6 Autothrottle

A simple autothrottle is included in the design to command the throttle in lieu of a pilot. For the CAS examples, the autothrottle is set to hold the airspeed constant throughout the maneuver. The autothrottle tracks the reference model airspeed for the in-flight simulation results.

The autothrottle uses a basic proportional-integral-derivative (PID) controller that generates incremental throttle commands based on airspeed error, defined as

$$\boldsymbol{e}_{\mathrm{V_T}} \triangleq V_{\mathrm{T_{cmd}}} - z^{-1} V_{\mathrm{T}}\,,$$

where a one-sample delay has been added to the velocity feedback to avoid an algebraic loop in Simulink. The PID controller is implemented in discrete-time[1] and takes the form

$$C(z) = K_{\mathrm{P}} + \frac{K_{\mathrm{I}} T_{\mathrm{s}}}{z-1} + \frac{K_{\mathrm{D}} K_{\mathrm{N}}}{1 + K_{\mathrm{N}} T_{\mathrm{s}} \left(\dfrac{1}{z-1} \right)} \tag{3.7}$$

where K_{P}, K_{I}, and K_{D} are the PID gains, T_{s} is the sampling time, and K_{N} is a filter coefficient for a discrete-time approximate derivative. The controller is implemented using the built-in Simulink "PID Controller" block. The output of the PID controller is limited to the interval $[-\delta_{\mathrm{T_1}}, 1 - \delta_{\mathrm{T_1}}]$ so that the total throttle command is limited to $[0, 1]$. The implementation of the autothrottle is shown in Fig. 3.18.

---

[1] The discrete-time implementation is chosen to allow the use of a unit delay on the airspeed feedback to avoid an algebraic loop in Simulink. Using a delay with a continuous-time solver suitable for stiff equations results in an unacceptable run time.

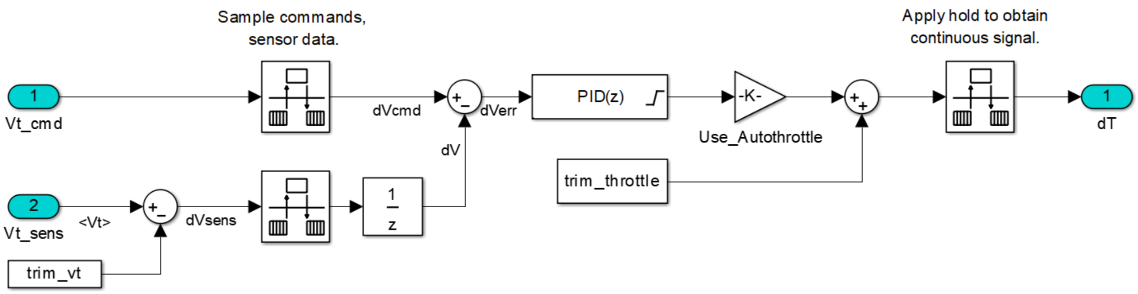


Figure 3.18: Autothrottle subsystem.

The gains for the PID controller are designed using a linearized model of the airspeed dynamics generated at the flight condition of interest. The autothrottle design system is the transfer function from δ_{T} to V_{T}, inclusive of throttle dynamics from the nonlinear model. A simple Simulink model of the closed-loop linear system is used for tuning the gains against a velocity step command. A sampling rate of $100\,\mathrm{Hz}$ is used for the autothrottle, which is applied to the continuous-time system with a zero-order hold. Table 3.6 lists the chosen gains.

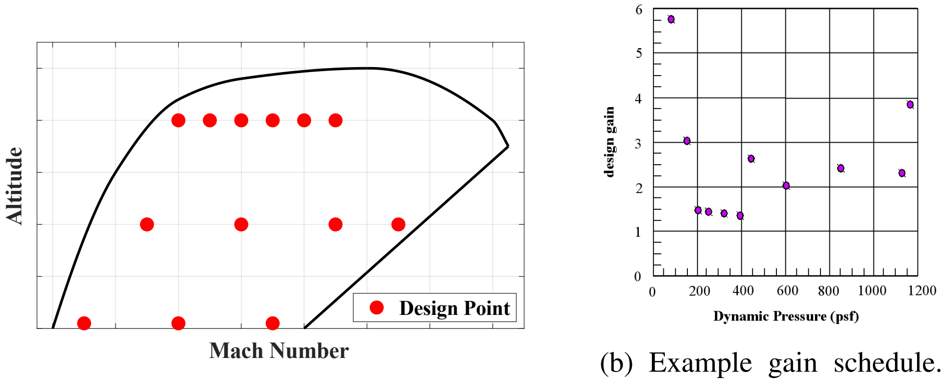
Table 3.6: Autothrottle gains.

Design Parameter	**Value**
K_{P}	0.7
K_{I}	0.9
K_{D}	0.6
K_{N}	10

4. DYNAMIC INVERSION

4.1 Introduction

Dynamic inversion (DI) is a special case of a nonlinear control technique called Feedback Linearization. DI is an increasingly popular scheme for flight control, particularly for aircraft that operate over multiple flight regimes, as it is a nonlinear technique valid over the entire flight envelope. This is in contrast to linear control methods such as classical PID or modern optimal control methods, which are used to design a controller for a single trim condition. Multiple controllers are then designed at different trim conditions covering the aircraft's flight envelope, and the resulting gains are scheduled based on the current flight condition (usually the dynamic pressure or a combination of Mach number and altitude). Interpolation is used to obtain gains for flight conditions in-between those which have been used to design gains; alternatively, function approximation techniques are used to provide functional representations of the scheduled gains. Figure 4.1 shows an example of gain scheduling applied to SISO digital controllers for an F-5A Freedom Fighter.



(a) Flight conditions for gain design.

(b) Example gain schedule. (Reprinted from [52].)

Figure 4.1: Gain scheduling example for F-5A Freedom Fighter.

Gain scheduling has the advantages of allowing linear control design techniques to be applied to linear systems via the use of several locally linear models and has thus been a primary method used for flight control in industry. In this respect, gain scheduling is a relatively easy method for controlling nonlinear systems. Despite this, several issues exist with gain scheduling:

1. selection of trim conditions requires expertise and domain knowledge,
2. selected trim conditions must be sufficiently dense to capture the system behavior,
3. the system dynamics must change relatively slowly,
4. controller performance must be evaluated for both the locally linear models and the global nonlinear model,
5. fast gain switching can introduce instabilities, and
6. multiple linear controllers must be designed.

This last item is a particularly important drawback when labor-intensive classical design techniques are used.

Nonlinear Dynamic Inversion (NDI) is a useful technique for controlling aerospace systems given their inherently nonlinear nature. This is especially true for high-performance aircraft with expansive flight envelopes, as well as hypersonics and reentry vehicles. NDI in theory allows the design of a single controller for the entire flight envelope, reducing the amount of control design necessary compared to traditional gain scheduling. In addition, in theory perfect cancellation of dynamics using NDI results in perfect tracking of the desired dynamics. NDI is not a panacea, however, as in practice modeling uncertainties, actuator dynamics, and sensor noise can prevent perfect cancellation of dynamics. Additionally, low control effectiveness can cause numerical issues with the controller and control effector saturation. NDI additionally provides no robustness guarantees, requiring an outer control loop [20]. Combining NDI with adaptive control methods helps address some of these issues, in particular model uncertainty and robustness. NDI provides a baseline controller with the best known model of the system, while adaptive corrections address modeling errors and errors tracking the desired dynamics.

DI for flight control is not a new technique, but widespread adoption did not occur until the advent of powerful digital flight control computers. Prior to this, limitations in onboard memory and computing power limited the use of the dynamic inversion methodology [53]. Modern high-performance fighters thus tend to use dynamic inversion-based flight control systems, such as the Boeing F-18 High Alpha Research Vehicle [54], Lockheed Martin F-35 Lightning II [55], and Boeing X-36 [56]. Reentry vehicles such as NASA's canceled X-38 Crew Return Vehicle also commonly use DI control laws [20]. In particular, the F-35 Lightning II has operationally proven the utility and effectiveness of NDI flight controllers.

This section introduces the underlying concept of feedback linearization, and then introduces dynamic inversion for both linear and nonlinear systems.

4.2 Feedback Linearization

At its core, feedback linearization is a nonlinear control technique that uses feedback to produce linear dynamics rather than using a linear model approximation to the nonlinear dynamics. Consider the SISO system

$$\begin{aligned} \dot{x} &= f(x) + g(x)u\,, \\ y &= h(x)\,, \end{aligned} \tag{4.1}$$

where y is the controlled output. The objective is to obtain an input-output mapping from u to y that is affine in the control u. Taking the time derivative of y, obtain

$$\dot{y} = \frac{\partial h}{\partial x} f(x) + \frac{\partial h}{\partial x} g(x) u = L_{\mathrm{f}} h(x) + L_{\mathrm{g}} h(x) u \,, \tag{4.2}$$

where

$$L_{\mathrm{f}} h(x) \triangleq \frac{\partial h}{\partial x} f(x) \quad \text{and} \quad L_{\mathrm{g}} h(x) \triangleq \frac{\partial h}{\partial x} g(x)$$

are *Lie derivatives* of the system. This notation make it easy to express repeated differentiations, e.g.

$$L_{\mathrm{f}}^2 h(x) = \frac{\partial (L_{\mathrm{f}} h)}{\partial x} f(x) \quad \text{and} \quad L_{\mathrm{g}} L_{\mathrm{f}} h(x) = \frac{\partial (L_{\mathrm{f}} h)}{\partial x} g(x) \,.$$

If $L_{\mathrm{g}} h(x) = 0$, then $\dot{y}$ is independent of u and further differentiation is required. This process is continued until $L_{\mathrm{g}} L_{\mathrm{f}}^{r-1} h(x) \neq 0$, which results in the dynamics

$$y^{(r)} = L_{\mathrm{f}}^r h(x) + L_{\mathrm{g}} L_{\mathrm{f}}^{r-1} h(x) u \,. \tag{4.3}$$

The number of differentiations required r is the *relative degree* of the system. Formally, the integer r is the relative degree of the system Eq. (4.1) if $L_{\mathrm{g}} L_{\mathrm{f}}^i h = 0$ for $i = 0, \ldots, r-2$ and $L_{\mathrm{g}} L_{\mathrm{f}}^{r-1} h \neq 0$ [57]. If the system Eq. (4.1) is controllable and of size n, then $r \leqslant n$.

Equation (4.3) results in the control law

$$u = \frac{1}{L_{\mathrm{g}} L_{\mathrm{f}}^{r-1} h(x)} \left(\nu - L_{\mathrm{f}}^r h(x) \right) \tag{4.4}$$

where ν is a pseudocontrol. This control law cancels the nonlinear dynamics and the pseudocontrol achieves the desired control objective. Substituting Eq. (4.4) into Eq. (4.3) results in the linear dynamics $y^{(r)} = \nu$, a chain of integrators. The pseudocontrol ν can be desired dynamics or error dynamics for the tracking problem. These results are easily extended to the case of MIMO systems provided that $L_{\mathrm{g}}L_{\mathrm{f}}^{r-1}h(x)$ is an invertible matrix.

If $r < n$, then part of the system dynamics is made unobservable by the feedback linearization process [58]. These unobservable dynamics are called the *internal dynamics* of the system and must be stable for Eq. (4.4) to be a stabilizing control law. In general, showing the stability of the internal dynamics is nontrivial. An alternative method is to use the *zero dynamics* of the system, which are the dynamics resulting from the control input required to hold the output of the system at zero. The zero dynamics are an extension of the concept of zeros in linear systems to nonlinear systems. Stable zero dynamics in a linear system (i.e. the zeros of the linear system are analytic in the right-half plane) results in stable internal dynamics. For the nonlinear system, stable zero dynamics results in local stability of the internal dynamics. Accordingly, feedback linearization requires the system to be minimum phase [58]. References [57, 58] are suggested for in-depth discussions of feedback linearization.

4.3 Linear Dynamic Inversion

Dynamic inversion is a special case of feedback linearization for systems that are of relative degree one. Flight control problems usually feature dynamics that are relative degree one, as the relevant dynamics are first-order and feature the control effectors. In this section, the basic approach to dynamic inversion for flight control is introduced using linearized aircraft dynamics; in this thesis this approach will be referred to as Linear Dynamic Inversion (LDI).

In the linear case, the state-space system is inverted to obtain the control $\boldsymbol{u}$ as a function of the system dynamics and the desired dynamics $\dot{\boldsymbol{x}}_{\mathrm{des}}$, as shown in Eq. (4.5):

$$\boldsymbol{u} = [B]^{-1}(\dot{\boldsymbol{x}}_{\mathrm{des}} - [A]\boldsymbol{x}) \,. \tag{4.5}$$

Note here the requirement for the existence of $[B]^{-1}$; this is equivalent to requiring an equal number of control effectors as controlled variables. Since the aerodynamic control effectors directly impact the rotational dynamics, this is equivalent to stating that there must be one control for each axis of rotation. For conventional aircraft, the ailerons are used to control roll, the elevator is used to control pitch, and the rudder is used to control yaw. For aircraft with complex control effector configurations, a pseudo-control can be defined for each control variable, and then the physical control effectors can be used via a control allocation process [51]. Such an approach can be seen in the "effector blender" for the X-35B in Ref. [55], which features a redistributed weighted pseudoinverse algorithm that accounts for rate and position limits and axis prioritization.

Consider the following example from Ref. [20], featuring simplified linear lat/d aircraft dynamics, shown in Eq. (4.6):

$$\begin{Bmatrix} \dot{p} \\ \dot{r} \end{Bmatrix} = \begin{bmatrix} L_{\mathrm{p}} & L_{\mathrm{r}} & L_{\beta} \\ N_{\mathrm{p}} & N_{\mathrm{r}} & N_{\beta} \end{bmatrix} \begin{Bmatrix} p \\ r \\ \beta \end{Bmatrix} + \begin{bmatrix} L_{\delta_{\mathrm{A}}} & L_{\delta_{\mathrm{R}}} \\ N_{\delta_{\mathrm{A}}} & N_{\delta_{\mathrm{R}}} \end{bmatrix} \begin{Bmatrix} \delta_{\mathrm{A}} \\ \delta_{\mathrm{R}} \end{Bmatrix} . \tag{4.6}$$

In Eq. (4.6) the terms in the matrices are the linear dimensional stability and control derivatives for the rolling and yawing moment equations. Inverting this equation as in Eq. (4.5), the control law is obtained as Eq. (4.7),

$$\begin{Bmatrix} \delta_{\mathrm{A_{cmd}}} \\ \delta_{\mathrm{R_{cmd}}} \end{Bmatrix} = \begin{bmatrix} L_{\delta_{\mathrm{A}}} & L_{\delta_{\mathrm{R}}} \\ N_{\delta_{\mathrm{A}}} & N_{\delta_{\mathrm{R}}} \end{bmatrix}^{-1} \left(\begin{Bmatrix} \dot{p}_{\mathrm{des}} \\ \dot{r}_{\mathrm{des}} \end{Bmatrix} - \begin{bmatrix} L_{\mathrm{p}} & L_{\mathrm{r}} & L_{\beta} \\ N_{\mathrm{p}} & N_{\mathrm{r}} & N_{\beta} \end{bmatrix} \begin{Bmatrix} p \\ r \\ \beta \end{Bmatrix} \right) , \tag{4.7}$$

where the matrix inverse is

$$\begin{bmatrix} L_{\delta_{\mathrm{A}}} & L_{\delta_{\mathrm{R}}} \\ N_{\delta_{\mathrm{A}}} & N_{\delta_{\mathrm{R}}} \end{bmatrix}^{-1} = \frac{1}{L_{\delta_{\mathrm{A}}} N_{\delta_{\mathrm{R}}} - N_{\delta_{\mathrm{A}}} L_{\delta_{\mathrm{R}}}} \begin{bmatrix} N_{\delta_{\mathrm{R}}} & -L_{\delta_{\mathrm{R}}} \\ -N_{\delta_{\mathrm{A}}} & L_{\delta_{\mathrm{A}}} \end{bmatrix} .$$

Note here that small control effectiveness terms can have the effect of creating a large inverse value, which can lead to problems such as control saturation. This requires addressing during the control law design.

The form of the dynamic inversion controller in Eq. (4.7) allows the designer to specify the desired dynamics. Various methods can be used to generate the $\dot{\boldsymbol{x}}_{\mathrm{des}}$ term, including using a reference model and specifying the desired flying qualities [59]. In the former case, DI is yet another form of a model-following controller. In the latter case, a common choice is to specify the desired dynamics in terms of the desired flying qualities found in MIL-STD-1797B [50] or its predecessor MIL-F-8785C [47]. Note that distribution of MIL-STD-1797B is restricted while distribution of MIL-F-8785C is unlimited, making the latter a more accessible document for academic research. Advantages of MIL-STD-1797B include updated flying qualities requirements for highly-augmented aircraft.

4.3.1 Linear Dynamic Inversion F-16 Simulation Example

An illustrative example is a bank angle controller designed for a linearized aircraft model of the Lockheed Martin F-16A Fighting Falcon. The flight condition is $M_1 = 0.18$

at $100\,\mathrm{ft}$, and the state-space formulation is

$$\left\{\begin{matrix}\dot{\beta}\\ \dot{p}\\ \dot{r}\\ \dot{\phi}\\ \dot{\psi}\end{matrix}\right\} = \begin{bmatrix} -0.132 & 0.324 & -0.94 & 0.149 & 0\\ -10.614 & -1.1719 & 1.0023 & 0 & 0\\ 0.997 & -0.00182 & -0.259 & 0 & 0\\ 0 & 1 & 0.34 & 0 & 0\\ 0 & 0 & 1.0561 & 0 & 0 \end{bmatrix} \left\{\begin{matrix}\beta\\ p\\ r\\ \phi\\ \psi\end{matrix}\right\} + \begin{bmatrix} 0.0069 & 0.0189\\ -5.935 & 1.203\\ -0.122 & -0.614\\ 0 & 0\\ 0 & 0 \end{bmatrix} \left\{\begin{matrix}\delta_{\mathrm{A}}\\ \delta_{\mathrm{R}}\end{matrix}\right\} . \tag{4.8}$$

Using Eq. (4.7) with Eq. (4.8), the controller takes the form

$$\left\{\begin{matrix}\delta_{\mathrm{A}_{\mathrm{cmd}}}\\ \delta_{\mathrm{R}_{\mathrm{cmd}}}\end{matrix}\right\} = \begin{bmatrix} -0.1620 & -0.3173\\ 0.0322 & -1.5656 \end{bmatrix} \left(\left\{\begin{matrix}\dot{p}_{\mathrm{des}}\\ \dot{r}_{\mathrm{des}}\end{matrix}\right\} - \begin{bmatrix} -1.1790 & 1.0023 & -10.6140\\ -0.0018 & -0.2590 & 0.9970 \end{bmatrix} \left\{\begin{matrix}p\\ r\\ \beta\end{matrix}\right\} \right) \tag{4.9}$$

Since the objective is to control the bank angle it is necessary to express the control law in terms of the commanded bank angle, ϕ_{cmd}. This is accomplished by taking the linearized Euler angle kinematics for the bank angle from Eq. (2.34), treating the roll rate as a control, and inverting:

$$p_{\mathrm{cmd}} = \dot{\phi}_{\mathrm{des}} - r\tan\Theta_1 \,.$$

For simplicity, first-order reference dynamics are used to obtain the desired rates from the commanded variables:

$$\dot{p}_{\mathrm{des}} = \omega_{\mathrm{p}}(p_{\mathrm{cmd}} - p)\,,$$
$$\dot{\phi}_{\mathrm{des}} = \omega_{\phi}(\phi_{\mathrm{cmd}} - \phi)\,.$$

The roll rate desired dynamics can then be expressed as

$$\dot{p}_{\mathrm{des}} = \omega_{\mathrm{p}}(-p - r\tan\Theta_1 + \omega_\phi(\phi_{\mathrm{cmd}} - \phi))\,. \tag{4.10}$$

It is also desirable for the controller to minimize the sideslip angle. Since the controller Eq. (4.7) uses p and r as the controlled variables, it is necessary to express the desired β in terms of $\dot{r}_{\mathrm{des}}$. Here the approximation $\dot{\beta} \approx -r$ is used to obtain

$$r_{\mathrm{cmd}} = -\dot{\beta}_{\mathrm{des}}\,, \tag{4.11}$$

where

$$\dot{\beta}_{\mathrm{des}} = \omega_\beta(\beta_{\mathrm{cmd}} - \beta) \tag{4.12}$$

relates the commanded sideslip (nominally zero) to the desired dynamics. Similarly to the roll rate, the yaw rate is assigned first-order desired dynamics:

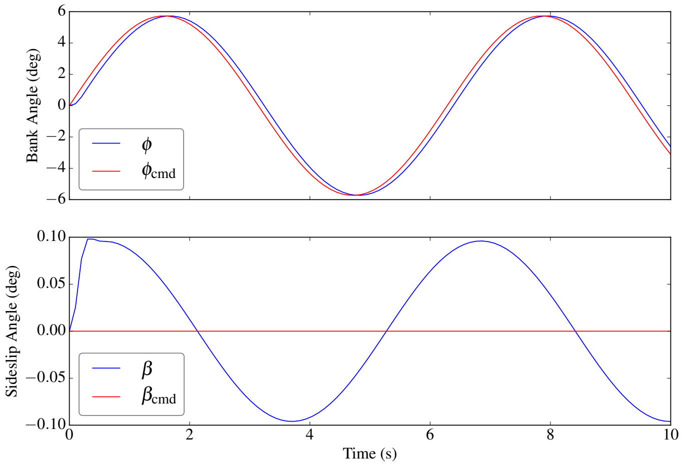
$$\dot{r}_{\mathrm{des}} = \omega_{\mathrm{r}}(r_{\mathrm{cmd}} - r)$$

resulting in

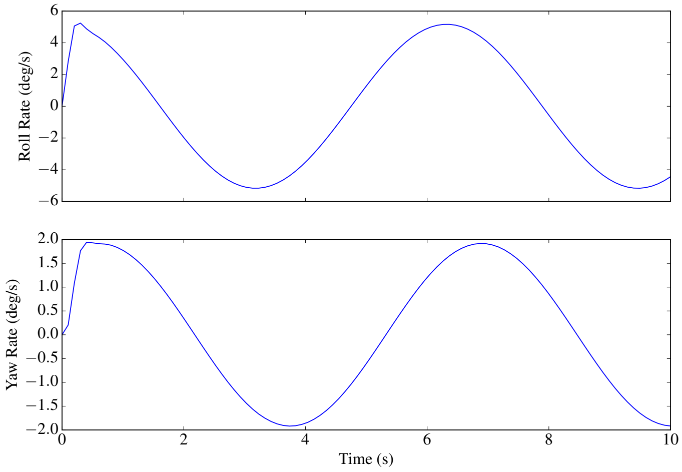
$$\dot{r}_{\mathrm{des}} = \omega_{\mathrm{r}}(\omega_\beta(\beta - \beta_{\mathrm{cmd}}) - r)\,. \tag{4.13}$$

Together, Eqs. (4.9–4.13) describe the linear dynamic inversion controller.

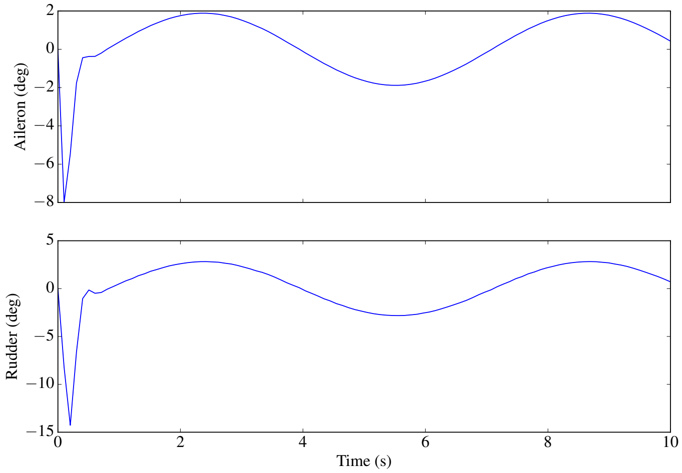
Figure 4.2a plots the bank angle and sideslip angle time histories against their commands. The linear DI controller tracks the commanded bank angle well with a slight phase lag due to the first-order desired dynamics, and regulates the sideslip angle to $0.1°$. The roll and yaw rates resulting from the bank angle and sideslip angle commands are shown in Fig. 4.2b, while the aileron and rudder time histories are shown in Fig. 4.2c.



(a) Bank angle and sideslip time histories and commands.



(b) Body-axis roll and yaw rate.



(c) Aileron and rudder time histories.

Figure 4.2: Linear DI controller time histories.

4.4 Nonlinear Dynamic Inversion

The linear DI method outlined in Section 4.3 can be easily extended to nonlinear systems to obtain a nonlinear control scheme that inverts the aircraft dynamics to obtain a control law that operates over the entire flight envelope (or at least the modeled subset of the flight envelope). If the system is affine in the control $\boldsymbol{U}$, the dynamics can be expressed as the form Eq. (4.14).

$$\dot{\boldsymbol{X}} = \boldsymbol{F}(\boldsymbol{X}) + [B(\boldsymbol{X})]\boldsymbol{U} \tag{4.14}$$

Here, $\boldsymbol{F}$ and $[B]$ are locally Lipschitz nonlinear functions of the state. Note that the notation $\boldsymbol{G}(\boldsymbol{X})$ is commonly used to represent the nonlinear affine-in-control term; in this thesis the notation of Ref. [51] is adopted and $[B]$ is used. The function $\boldsymbol{F} : \mathbb{R}^n \mapsto \mathbb{R}^n$ models the system dynamics and the function $[B] : \mathbb{R}^n \mapsto \mathbb{R}^{n \times m}$ models the control effectiveness and is assumed linear in the controls. If $[B]$ is invertible, then Eq. (4.14) may be solved for the control $\boldsymbol{U}$ to obtain the inverse dynamics, Eq. (4.15),

$$\boldsymbol{U} = [B(\boldsymbol{X})]^{-1} \left[\dot{\boldsymbol{X}}_{\text{des}} - \boldsymbol{F}(\boldsymbol{X})\right] , \tag{4.15}$$

where $\dot{\boldsymbol{X}}$ is replaced with the desired dynamics $\dot{\boldsymbol{X}}_{\text{des}}$ to obtain the NDI control law. If the system dynamics $\boldsymbol{F}$ and $[B]$ are known exactly, substitution of Eq. (4.15) into Eq. (4.14) results in perfect cancellation to obtain $\dot{\boldsymbol{X}} = \dot{\boldsymbol{X}}_{\text{des}}$. This also requires that the control effort requested be within the available bandwidth of the actuators.

Since the nonlinear dynamics model the system behavior over its flight envelope, the controller drives the system to match the desired dynamics without need for scheduling [20, 49]. Figure 4.3 shows a block diagram of a generic NDI controller.

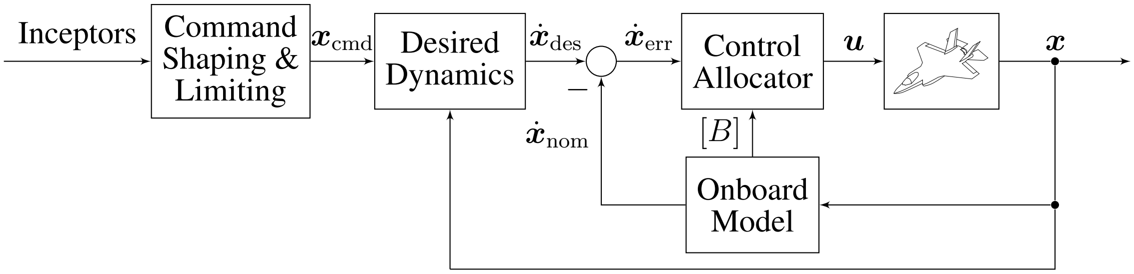


Figure 4.3: Block diagram of a generic NDI structure.

4.4.1 Robustness

In practice, it is impossible to know the system dynamics exactly, and so the dynamics are replaced by their estimates $\hat{\boldsymbol{F}}$ and $[\hat{B}]$ to obtain the control law

$$\boldsymbol{U} = [\hat{B}(\boldsymbol{X})]^{-1}\left[\dot{\boldsymbol{X}}_{\text{des}} - \hat{\boldsymbol{F}}(\boldsymbol{X})\right] . \tag{4.16}$$

These estimates are generally based on modeling of the system (as described in Section 2) or result from parameter or system identification. Substituting Eq. (4.16) into Eq. (4.14) results in

$$\begin{aligned}\dot{\boldsymbol{X}} &= \boldsymbol{F}(\boldsymbol{X}) + [B(\boldsymbol{X})][\hat{B}(\boldsymbol{X})]^{-1}\left[\dot{\boldsymbol{X}}_{\text{des}} - \hat{\boldsymbol{F}}(\boldsymbol{X})\right] \\ \dot{\boldsymbol{X}} &= [B(\boldsymbol{X})][\hat{B}(\boldsymbol{X})]^{-1}\dot{\boldsymbol{X}}_{\text{des}} + \boldsymbol{F}(\boldsymbol{X}) - [B(\boldsymbol{X})][\hat{B}(\boldsymbol{X})]^{-1}\hat{\boldsymbol{F}}(\boldsymbol{X}) ,\end{aligned} \tag{4.17}$$

which demonstrates the imperfect cancellation. Actuator dynamics and sensor errors additionally reduce the control performance, but are generally neglected during the control law development [20]. The remainder of this subsection introduces techniques for robust NDI. Note that these techniques apply also to the linearized DI approach in Section 4.3.

4.4.1.1 Auxiliary Outer Loop Control Signal

To address this issue an outer loop is added for robustness. In this case, the control law Eq. (4.15) is modified by adding an auxiliary input $\boldsymbol{\nu}$ so that

$$\boldsymbol{U} = [\hat{B}(\boldsymbol{X})]^{-1}\left[\dot{\boldsymbol{X}}_{\text{des}} - \hat{\boldsymbol{F}}(\boldsymbol{X}) + \boldsymbol{\nu}\right] . \tag{4.18}$$

Assuming perfect cancellation of the dynamics for the purposes of design, the resulting closed-loop dynamics are then

$$\dot{\boldsymbol{X}} = \dot{\boldsymbol{X}}_{\text{des}} + \boldsymbol{\nu} .$$

In practice the inversion is imperfect and residual terms as discussed in Eq. (4.17) remain. One method for the design of this additional signal $\boldsymbol{\nu}$ is to design a robust outer loop controller for the error $\boldsymbol{E} = \boldsymbol{X} - \boldsymbol{X}_{\text{des}}$. This definition of the error signal results in the dynamics

$$\dot{\boldsymbol{E}} = \boldsymbol{\nu}$$

which can then be stabilized using any appropriate control technique [31]. Figure 4.4 is a block diagram of a robust NDI controller assuming a Linear Quadratic Regulator (LQR)-based outer loop. Common outer loop control methods include PI control [60], μ-synthesis [21] and linear quadratic control [48].

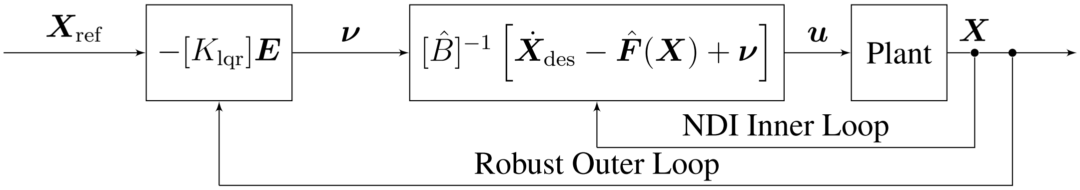


Figure 4.4: Robust NDI block diagram.

4.4.1.2 Adaptive Augmentation

Another option for robustness is to use adaptive control techniques to either estimate the plant parameters or to estimate the error between the model and the true dynamics. This approach will be used in this thesis and is discussed further in Section 6 in the context of a controller for an in-flight simulator.

4.4.2 Extension to Non-Affine Control Effectors

Equation (4.14) used to derive the baseline NDI control law assumes that the plant is affine in the control. In general, this is a valid assumption for aircraft. However in some cases such as where a high-fidelity nonlinear aerodynamic model is available it is desirable to use the more generic nonlinear model

$$\dot{\boldsymbol{X}} = \boldsymbol{F}(\boldsymbol{X}, \boldsymbol{U})\,. \tag{4.19}$$

Since the DI techniques require that the system be affine in the control, Eq. (4.19) is linearized about some control effector position $\boldsymbol{U}^\star$ to obtain a model in the form

$$\dot{\boldsymbol{X}} \approx \boldsymbol{F}(\boldsymbol{X}, \boldsymbol{U}^\star) + [B]\boldsymbol{u}\,, \tag{4.20}$$

where $[B]$ is a locally-linear approximation of the control effectiveness and $\boldsymbol{u}$ is a linearized control deflection such that the total nonlinear control $\boldsymbol{U} = \boldsymbol{U}^\star + \boldsymbol{u}$. The choice of $\boldsymbol{U}^\star$ is left to the designer, but common choices are the trimmed control vector $\boldsymbol{U}_1$, the previous value of the control deflection $z^{-1}\boldsymbol{U}$ or feedback of the effector position. This results in a control law

$$\begin{aligned} \boldsymbol{u} &= [B]^{-1}\left(\dot{\boldsymbol{X}}_{\text{des}} - \boldsymbol{F}(\boldsymbol{X}, \boldsymbol{U}^\star) + \boldsymbol{\nu}\right)\,, \\ \boldsymbol{U} &= \boldsymbol{U}^\star + \boldsymbol{u}\,. \end{aligned} \tag{4.21}$$

If $\boldsymbol{U}^\star$ is chosen as zero, then Eq. (4.21) collapses to the form Eq. (4.18). Note that Eq. (4.21) requires that $[B]$ be invertible; if $[B]$ is calculated online then it may be necessary to replace the inverse in Eq. (4.21) with a pseudoinverse. Methods for calculating $[B]$ online from an onboard model are presented in Refs. [17, 51]. In brief, a linear approximation of the dimensionless control derivative is calculated using a finite difference method about $\boldsymbol{U}^\star$, and the relevant column in the $[B]$-matrix is then obtained using by calculating the moment from the control power derivative.

4.4.3 Choice of Controlled Variables

The development of DI techniques in the preceding sections assumes full-state feedback and square control distribution (i.e. the number of states is the same as the number of controls). If a subset of the state variables are controlled, or other control variables (such as those described in Ref. [49]) are used, then the above formulations must be modified. In brief, the dynamics to be inverted become

$$\dot{\boldsymbol{CV}} = \frac{\partial \boldsymbol{H}(\boldsymbol{X})}{\partial \boldsymbol{X}} \dot{\boldsymbol{X}} \tag{4.22}$$

where $\boldsymbol{CV}$ is the vector of controlled variables determined from the states by the mapping $\boldsymbol{CV} = \boldsymbol{H}(\boldsymbol{X})$. The reader is invited to consult one of Refs. [31, 53] for discussion of NDI with controlled variables.

4.5 F-16 Simulation Examples

This section presents simulation examples using the nonlinear 6-DOF F-16C model and CAS design presented in Section 3. The CAS design is implemented using the baseline NDI control law Eq. (4.16) and the robustness technique presented in Section 4.4.1.1. Pilot commands are shown in Section 3 in Fig. 3.14.

4.5.1 Baseline NDI

The baseline NDI control law is based on Eq. (4.21), where the nonaffine F-16 dynamics are approximated via the approach of Section 4.4.2. The nonlinear aerodynamic and propulsion models in the F-16 simulation are used as the onboard model for the NDI control law, with

$$\hat{\boldsymbol{F}}(\boldsymbol{x}) = \boldsymbol{F}(\boldsymbol{X}, z^{-1}\boldsymbol{U})$$

used for the model of the dynamics. Explicitly, using the notation of Section 4.4.2, $\boldsymbol{U}^{\star} = z^{-1}\boldsymbol{U}$. A locally linear estimate of the control effectiveness is calculated online from the aerodynamic database using a central difference method. In this method, the aerodynamic coefficients C_{ℓ}, C_{m}, and C_{n} are calculated with $\pm 0.0001^{\circ}$ perturbations from the previous commanded control ($z^{-1}\boldsymbol{U}$). The control power derivative is then calculated as

$$C_{\mathrm{X}_{\delta}}(\alpha) = \frac{C_{\mathrm{X}}(\alpha, \delta + h) - C_{\mathrm{X}}(\alpha, \delta - h)}{2h}$$

where h is the perturbation. Note that the exact functional form of the coefficients vary. The $[B]$ matrix estimate is then formed by dimensionalizing the coefficients and multiplying by the inverse inertia tensor:

$$[B] = [I]^{-1} \begin{bmatrix} C_{\ell_{\delta_{\mathrm{A}}}} b & 0 & C_{\mathrm{n}_{\delta_{\mathrm{A}}}} b \\ 0 & C_{\mathrm{m}_{\delta_{\mathrm{E}}}} \bar{c} & 0 \\ C_{\ell_{\delta_{\mathrm{R}}}} b & 0 & C_{\ell_{\delta_{\mathrm{A}}}} b \end{bmatrix} \bar{q} S \, .$$

The control law is then explicitly

$$\boldsymbol{u}(k) = [B]^{\dagger} \left(\dot{\boldsymbol{X}}_{\mathrm{des}}(k) - \boldsymbol{F}(\boldsymbol{X}(k), \boldsymbol{U}(k-1)) \right) \, , \tag{4.23}$$

$$\boldsymbol{U}(k) = \boldsymbol{U}(k-1) + \boldsymbol{u}(k) \, , \tag{4.24}$$

where k is the index of the current time step in a discrete-time implementation. Note that for the baseline NDI control law the robust augmentation signal $\boldsymbol{\nu}$ is neglected. Additionally, the matrix inverse of $[B]$ is replaced with the pseudoinverse so that the control law remains valid if the online estimate of $[\hat{B}]$ becomes no longer invertible. For systems with more control effectors than control variables it is advisable to use a control allocator instead of a pseudoinverse as a well-designed control allocator such as that in Ref. [55] allows selection of effector displacements that satisfy position and rate limits, minimize the control demand, and allow for reconfigurability. For second- and higher-order desired dynamics, integrators used to calculate $\dot{\boldsymbol{X}}_{\mathrm{des}}(k)$ are replaced with discrete-time approximations.

The performance of the baseline NDI control law is shown in Fig. 4.5a, which plots the achieved angular rates against the desired rates, which are the integrals of the desired dynamics. Observing the figure, the baseline control law achieves relatively good tracking for the yaw axis, and adequate tracking for the pitch axis, with errors in the $2\,^{\circ}/\mathrm{s}$ range. The roll axis control objective is generally met, but the control law fails to achieve the desired roll rates when magnitudes are large ($\sim 200\,^{\circ}/\mathrm{s}$). The errors in the NDI control law tracking results from errors in the linearized control effectiveness estimates and actuator dynamics. Additionally, the actuator dynamics are nonlinear due to the position and no-load rate limit constraints. Figure 4.5a additionally plots the true airspeed, AOA, and sideslip angle. Airspeed and angle-of-attack are not directly commanded by the control law. The airspeed is kept nearly constant by the autothrottle, while the angle of attack is kept within limits. The sideslip angle is commanded to zero throughout the maneuver and is kept within 1°.

The NDI controller demands the aerodynamic control effector deflections shown in Fig. 4.5b. Observing the figure, all of the effectors remain withing rate and position limits for the entire simulation.

4.5.2 Robust NDI with LQR Outer Loop

This section presents a robust NDI controller with an LQR outer loop using Eq. (4.18). To illustrate the robustness added with the auxiliary outer loop control, the onboard model for the open-loop dynamics $\hat{\boldsymbol{F}}$ and control effectiveness $[\hat{B}]$ are obtained using the GNA model Eqs. (2.49–2.51) with the F-16C data listed in Ref. [43] instead of using the "true" dynamics as with the baseline configuration. This approach intentionally introduces modeling errors by neglecting the moments produced by forces and thrust effects in order to highlight robustness issues. Using the baseline NDI control law with this onboard model results in departure.

The robust augmentation is accomplished using infinite horizon LQR is used for this design example. The performance index is defined as

$$J = \frac{1}{2}\int_0^\infty \big(\boldsymbol{x}^\mathsf{T}[Q]\boldsymbol{x} + \boldsymbol{u}^\mathsf{T}[R]\boldsymbol{u}\big)\,\mathrm{d}t$$

where $[Q] \geqslant 0$ and $[R] > 0$ are weights on the state and control. The well-known solution to this optimal control problem is given by

$$\boldsymbol{u} = -[R]^{-1}[B]^\mathsf{T}[P]\boldsymbol{x} \tag{4.25}$$

where $[P]$ is the solution to the continuous algebraic Riccati equation

$$[0] = [A]^\mathsf{T}[P] + [P][A] + [Q] - [P][B][R]^{-1}[B]^\mathsf{T}[P]\,. \tag{4.26}$$

The matrix $[P]$ is constant and is computed offline. Control design software packages such as MATLAB provide efficient LQR routines making implementation straightforward. The theoretical basis for the LQR method can be found in any modern controls textbook such as one of Refs. [22, 31, 61, 62].

A proportional-integral LQR controller is designed by applying the LQR solution to the augmented error dynamics

$$\begin{Bmatrix} \dot{\boldsymbol{E}}_{\mathrm{I}} \\ \dot{\boldsymbol{E}} \end{Bmatrix} = \underbrace{\begin{bmatrix} [0_{3\times3}] & [I_3] \\ \multicolumn{2}{c}{[0_{3\times6}]} \end{bmatrix}}_{[\bar{A}]} \begin{Bmatrix} \boldsymbol{E}_{\mathrm{I}} \\ \boldsymbol{E} \end{Bmatrix} + \underbrace{\begin{bmatrix} [0_{3\times3}] \\ [I]_3 \end{bmatrix}}_{[\bar{B}]} \boldsymbol{\nu} \tag{4.27}$$

where $\boldsymbol{E}_{\mathrm{I}}$ is the integral of the error. If $([\bar{A}], [\bar{B}])$ is controllable and $([\bar{A}], [Q]^{1/2})$ is observable then standard LQR techniques can be used to obtain an outer loop controller of the form

$$\boldsymbol{\nu} = -[K_{\mathrm{e}}]\boldsymbol{E} - [K_{\mathrm{ei}}]\boldsymbol{E}_{\mathrm{I}} \tag{4.28}$$

where $[K_{\mathrm{e}}]$ and $[K_{\mathrm{ei}}]$ are the appropriate partitions from the LQR gain $[K_{\mathrm{lqr}}]$. The outer loop control signal Eq. (4.28) then augments the inner loop NDI control law Eq. (4.18). Table 4.1 presents the values of the weighting matrices for the LQR outer loop controller. The $[Q]$ and $[R]$ matrices are turned iteratively starting from identity until an acceptable response is achieved.

Table 4.1: LQR Outer Loop design parameters.

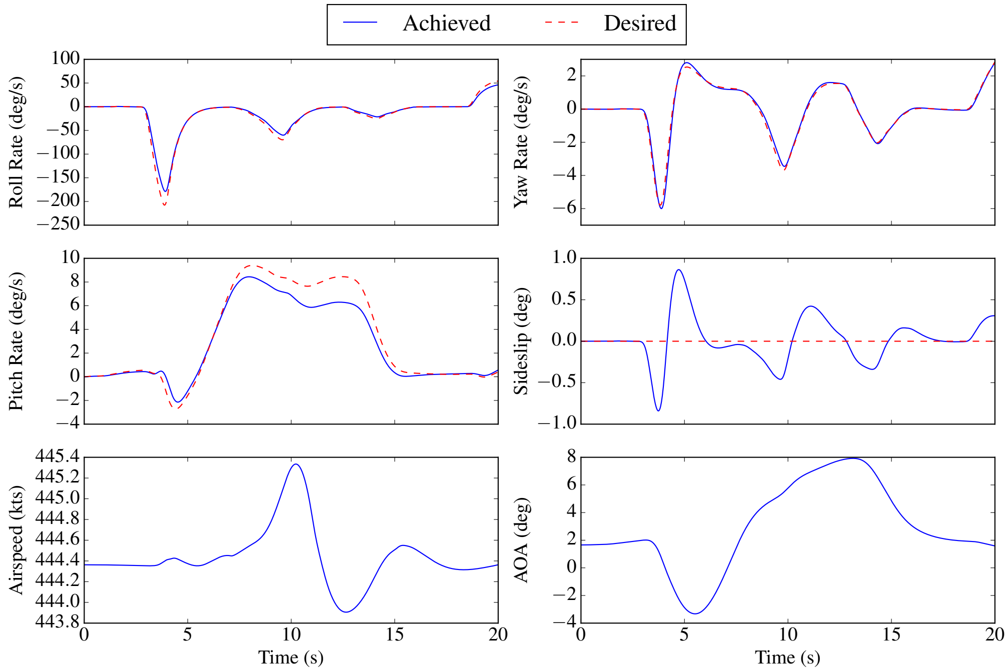
Design Parameter	**Value**
$[Q]$	$\mathrm{diag}(10, 5000, 50, 300, 250, 150)$
$[R]$	$\mathrm{diag}(15, 1, 1)$

The robust NDI method is compared against a standard NDI controller implemented with the same GNA-based approximate onboard model. This non-robust controller fails to achieve the control objective and the aircraft departs from controlled flight as seen in

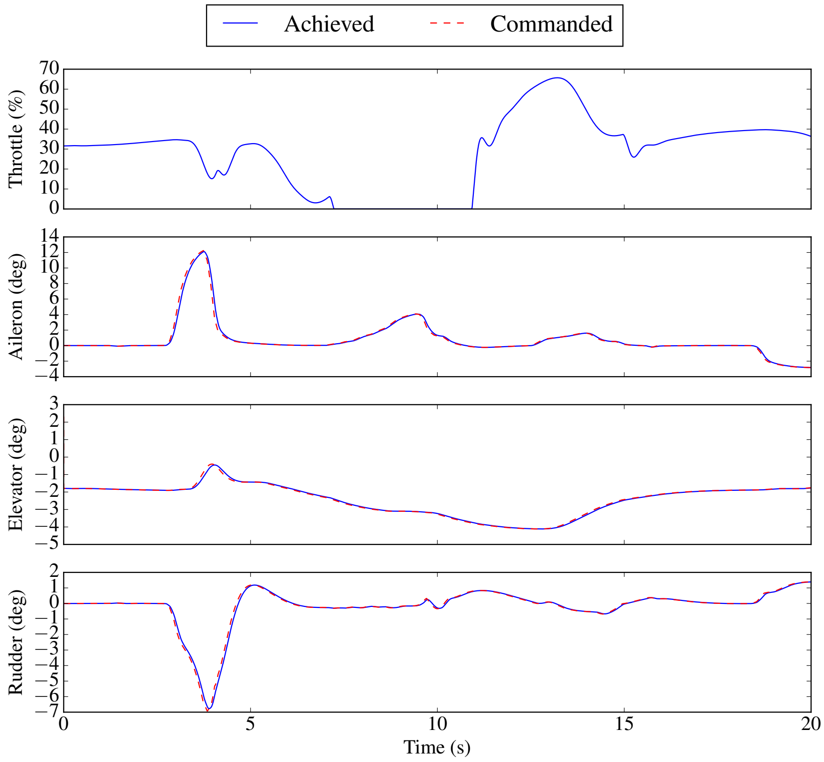
Fig. 4.6a. This simulation was terminated once the aircraft reached a $45°$ angle-of-attack at around the two second mark.

Figure 4.7a shows the performance of the robust NDI controller with the robustness loop described above. In contrast with the non-robust NDI control law, the robust control law achieves excellent tracking of the desired dynamics and performs better than the baseline DI control law with the "true" onboard model. The yaw axis shows the best performance, with errors less than $0.5\,°/\mathrm{s}$, followed by the pitch axis which exhibits errors less than $1\,°/\mathrm{s}$. The roll axis has larger errors of $\sim$$20\,°/\mathrm{s}$, which result from phase lag tracking the large roll rate command. This compares favorably to the roll rate error of $\sim$$30\,°/\mathrm{s}$ for the baseline controller. The airspeed hold and angle-of-attack changes remain reasonable, and the sideslip angle is regulated to within $1°$ with performance similar to the baseline NDI control law using the "true" OBM.

Finally, Fig. 4.7b presents the demanded control effector positions and the achieved positions. These are very similar to those of the baseline control law, showing that only minor modification of the control signals is required to achieve better performance. The control effectors remain within position and rate limits.

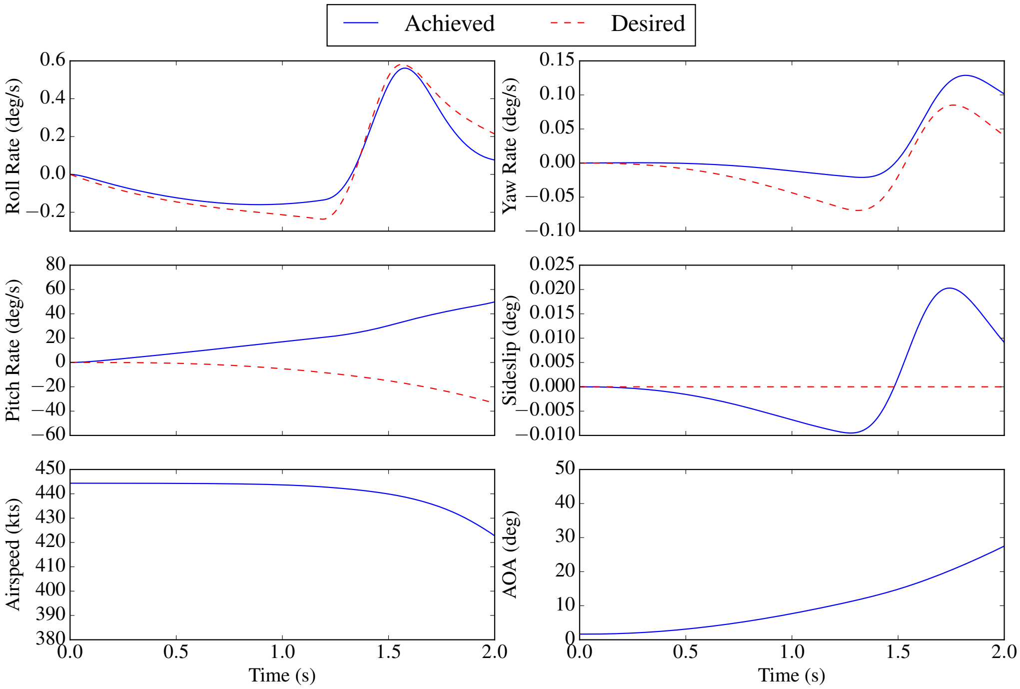


(a) Actual and desired state time histories.

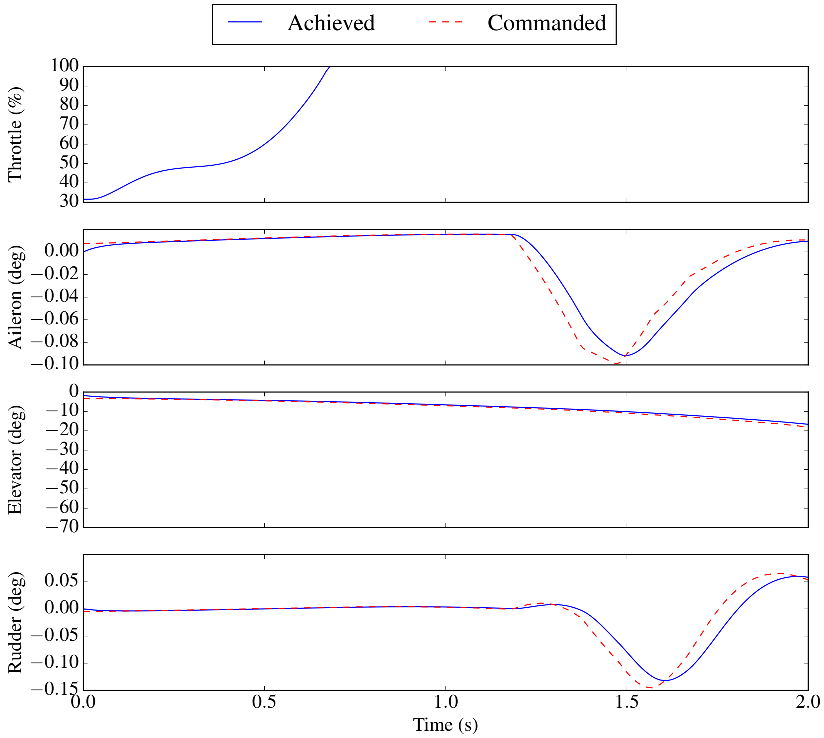


(b) Aerodynamic control effector time histories.

Figure 4.5: Exact NDI controller state and control time histories.

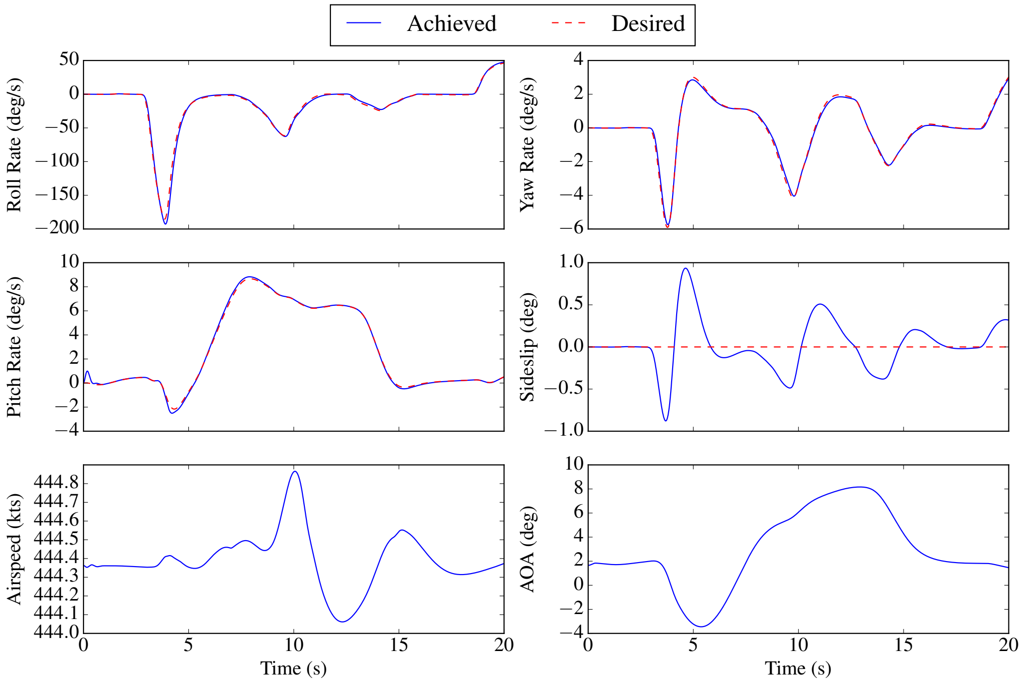


(a) Actual and desired state time histories.

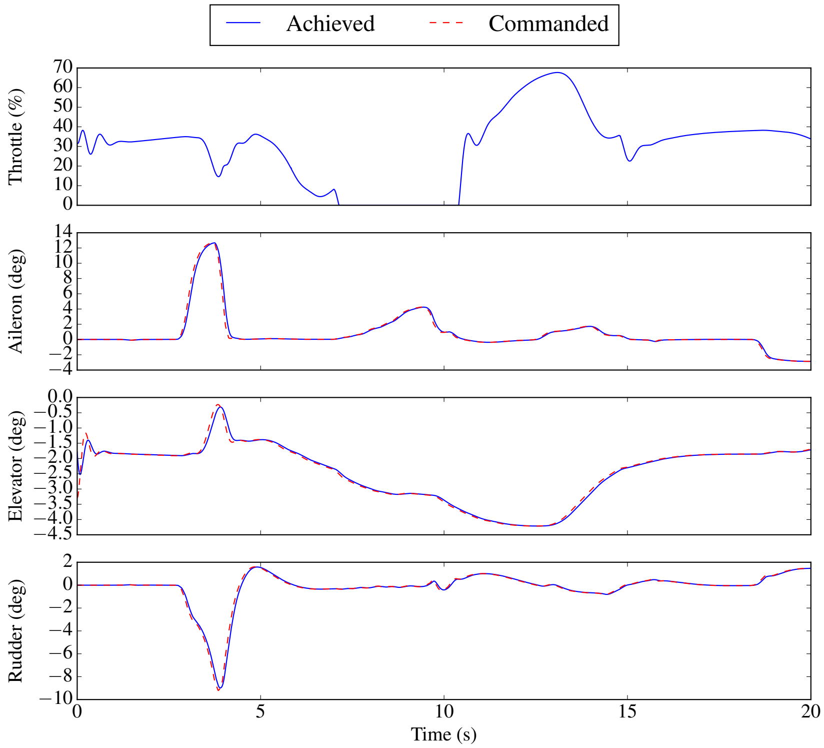


(b) Aerodynamic control effector time histories.

Figure 4.6: Approximate NDI controller state and control time histories.



(a) Actual and desired state time histories.



(b) Aerodynamic control effector time histories.

Figure 4.7: Robust NDI controller state and control time histories.

5. ADAPTIVE CONTROL

5.1 Introduction

Adaptive controllers can be succinctly described as the combination of a parameter estimator and a baseline control law. Accordingly, adaptive control is an enticing approach for systems with model uncertainty or unknown parameters. Adaptive control dates from the 1950s as a control methodology motivated by the large performance envelopes of emerging high-performance aircraft. The most (in)famous early use of an adaptive controller was with the NASA/USAF X-15 program [63, 64]. The first operational use was the adaptive controller for the F-111 [65].

The parameter estimator is used to either estimate plant parameters, from which controller parameters are computed, or to directly estimate controller parameters. The first approach is termed *indirect* adaptive control, while the second is *direct* adaptive control. Certain classes of adaptive controllers are also called "self-tuning regulators" because the parameter estimators determined the controller gains. Direct adaptive control tends to be more suited to minimum phase systems, although it is capable of of handling non-minimum phase systems with additional complexity, while indirect adaptive control works with both types of system. In this case indirect adaptive control suffers from the stabilizability problem, wherein the adaptive law cannot guarantee that the estimated parameters satisfy controllability and stabilizability requirements at each timestep. Combined direct/indirect schemes have been developed to rectify these issues [19].

The parameter estimators are generally realized in the forms of differential equations in the parameters called adaptive laws. As a result adaptive controllers are inherently nonlinear even in the case of a linear plant. Consequently, Lyapunov's direct method is usually used for design and analysis of adaptive controllers, although other methods exist.

Adaptive control research has seen the development of several classes of adaptive control architecture. The most common type of adaptive controller is Model Reference Adaptive Control (MRAC), which uses an adaptive controller to track the output of a reference model [19]. An adaptive control architecture based on an $\mathcal{L}_1$ norm condition developed in Refs. [66, 67], called $\mathcal{L}_1$ adaptive control ($\mathcal{L}_1$-AC) is increasingly popular. This approach combines a state predictor architecture with a filtering structure to decouple the estimation and control loops, allowing for large adaptive gains and the use of classical analysis tools. Structured Model Reference Control (SMRAC), a modification of MRAC that decomposes the dynamics into unknown kinetics and known kinematics, was developed at TAMU by Akella in Ref. [68]. Structured Adaptive Model Inversion (SAMI), SMRAC using a nominal dynamic inversion control law, was developed by Subbarao in Ref. [25].

This section presents a basic overview of adaptive control techniques for linear systems as preliminary material for Section 6, which develops nonlinear dynamic inversion based adaptive control laws. Topics covered include direct Model Reference Adaptive Control, $\mathcal{L}_1$ Adaptive Control, and two techniques for robust adaptive laws: the dead zone modification and the Projection Operator. Direct MRAC will be introduced first, followed by the Projection Operator as a prerequisite for the projection-based adaptive laws required by $\mathcal{L}_1$ control theory. A brief introduction to $\mathcal{L}_1$ adaptive controllers for linear systems follows, and the section concludes with brief remarks on the dead zone modification for practical implementation of adaptive controllers. All of the controllers presented are developed for MIMO systems. While SMRAC and SAMI are useful adaptive control structures, this thesis focuses on controlling aircraft attitude and thus the structured approach of these architectures is unneeded as the rotational kinematics are not considered.

5.2 Direct Model Reference Adaptive Control

Direct MRAC combines a model reference controller with adaptive laws to estimate the controller parameters. Figure 5.1 is a block diagram of an MRAC controller. The direct MRAC architecture consists of a stable known reference model, adaptive laws to estimate the controller parameters, and a control law. The adaptive laws update estimates of controller parameters based on the tracking error between the reference model and the measured plant output.

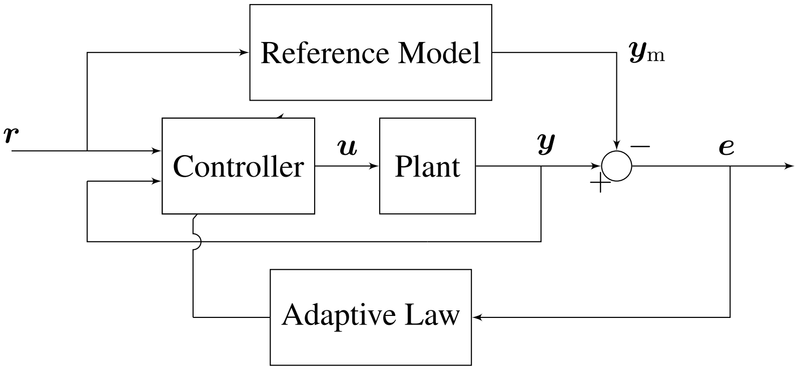


Figure 5.1: Direct model reference adaptive controller block diagram.

A Lyapunov-based direct MRAC design process can be summarized as a five step process:

1. Choose a nominal control law as if the unknown parameters were known. The adaptive control law is the nominal control law with the unknown parameters replaced by their adaptive estimates.
2. Determine matching conditions between the known reference model and the unknown closed-loop plant parameters.

3. Use the matching conditions to rewrite the unknown plant parameters in terms of the unknown controller parameters.
4. Determine the error dynamics between the state and the reference model in terms of the known parameters and adaptive estimation errors.
5. Use a candidate Lyapunov function to derive adaptive laws that make the error dynamics Lyapunov stable.

This process is now illustrated using a linear state-space system as an example, based on the approach in Ref. [22]. Consider a MIMO linear system of the form

$$\dot{\boldsymbol{x}} = [A]\boldsymbol{x} + [B][\Lambda](\boldsymbol{u} + \boldsymbol{f}(\boldsymbol{x})) \tag{5.1}$$

where the state matrix $[A]$ is constant and unknown, the control distribution matrix $[B]$ is constant and known, the control effectiveness matrix $[\Lambda]$ is constant and unknown, and $\boldsymbol{f}(\boldsymbol{x})$ is an unknown nonlinear matched disturbance.

The control objective is to track a reference command $\boldsymbol{r}$ with the desired reference model

$$\dot{x}_{\mathrm{ref}} = [A_{\mathrm{m}}]\boldsymbol{x}_{\mathrm{ref}} + [B_{\mathrm{m}}]\boldsymbol{r} \tag{5.2}$$

where $[A_{\mathrm{m}}]$ is Hurwitz. For simplicity consider $\boldsymbol{x}$, $\boldsymbol{u}$, $\boldsymbol{r} \in \mathbb{R}^n$. A candidate control law to meet this objective is

$$\boldsymbol{u} = -[K]^{\mathsf{T}}\boldsymbol{x} + [G]^{\mathsf{T}}\boldsymbol{r} - \boldsymbol{f}(\boldsymbol{x}) \tag{5.3}$$

which includes a stabilizing state feedback term, a feedforward of the reference signal, and a term to cancel the matched nonlinear uncertainty.

To determine matching conditions, substitute Eq. (5.3) into Eq. (5.1) and compare with

Eq. (5.2). Skipping steps, this results in

$$[A_\mathrm{m}] = [A] - [B][\Lambda][K]^\mathsf{T}\,, \tag{5.4}$$

$$[B_\mathrm{m}] = [B][\Lambda][G]^\mathsf{T}\,. \tag{5.5}$$

The matching condition Eq. (5.4) allows writing Eq. (5.1) in terms of $[K]$ as

$$\dot{\boldsymbol{x}} = [A_\mathrm{m}]\boldsymbol{x} + [B][\Lambda](\boldsymbol{u} + [K]^\mathsf{T}\boldsymbol{x} + \boldsymbol{f}(\boldsymbol{x}))\,. \tag{5.6}$$

Equation (5.6) is left in terms of $[B][\Lambda]$ as this will result in simpler algebra to derive the error dynamics.

Define the tracking error as $\boldsymbol{e} \triangleq \boldsymbol{x} - \boldsymbol{x}_\mathrm{ref}$. Using Eq. (5.6) and Eq. (5.2), one obtains

$$\dot{\boldsymbol{e}} = [A_\mathrm{m}]\boldsymbol{e} + [B][\Lambda](\boldsymbol{u} + [K]^\mathsf{T}\boldsymbol{x} + \boldsymbol{f}(\boldsymbol{x})) - [B_\mathrm{m}]\boldsymbol{r}\,. \tag{5.7}$$

Equation (5.5) allows writing Eq. (5.7) as

$$\dot{\boldsymbol{e}} = [A_\mathrm{m}]\boldsymbol{e} + [B][\Lambda](\boldsymbol{u} + [K]^\mathsf{T}\boldsymbol{x} - [G]^\mathsf{T}\boldsymbol{r} + \boldsymbol{f}(\boldsymbol{x}))\,. \tag{5.8}$$

The adaptive control law consists of Eq. (5.3) with the unknown parameters $[K]$, $[G]$, and $\boldsymbol{f}(\boldsymbol{x})$ replaced with their adaptive estimates, which are denoted with over-hats:

$$\boldsymbol{u} = -[\hat{K}]^\mathsf{T}\boldsymbol{x} + [\hat{G}]^\mathsf{T}\boldsymbol{r} - \hat{\boldsymbol{f}}(\boldsymbol{x})\,. \tag{5.9}$$

Using Eq. (5.9) in Eq. (5.8) results in

$$\dot{\boldsymbol{e}} = [A_\mathrm{m}]\boldsymbol{e} + [B][\Lambda](-[\tilde{K}]^\mathsf{T}\boldsymbol{x} + [\tilde{G}]^\mathsf{T}\boldsymbol{r} - \tilde{\boldsymbol{f}}(\boldsymbol{x})) \tag{5.10}$$

where the estimation errors are defined as $[\tilde{K}] \triangleq [\hat{K}] - [K]$ and so forth. One final step is to assume that the unknown function $\boldsymbol{f}$ can be modeled as

$$\boldsymbol{f}(\boldsymbol{x}) = [\Theta]^{\mathsf{T}}\boldsymbol{\phi}(\boldsymbol{x})\,, \tag{5.11}$$

where $[\Theta] \in \mathbb{R}^{p\times n}$ is an unknown constant parameter matrix and $\boldsymbol{\phi}(\boldsymbol{x}) : \mathbb{R}^n \mapsto \mathbb{R}^p$ is a known vector of basis functions. This results in a linear parametric model of the nonlinear uncertainty. Using the representation Eq. (5.11) results in the final form of the error dynamics,

$$\dot{\boldsymbol{e}} = [A_{\mathrm{m}}]\boldsymbol{e} + [B][\Lambda](-[\tilde{K}]^{\mathsf{T}}\boldsymbol{x} + [\tilde{G}]^{\mathsf{T}}\boldsymbol{r} - [\tilde{\Theta}]^{\mathsf{T}}\boldsymbol{\phi}(\boldsymbol{x}))\,. \tag{5.12}$$

Equation (5.12) is in the desired form of known linear Hurwitz error dynamics perturbed by estimation error terms. Next, the objective is to design adaptive laws such that the estimation errors tend to zero, ideally resulting in the exponentially stable error dynamics $\dot{\boldsymbol{e}} = [A_{\mathrm{m}}]\boldsymbol{e}$.

The remaining step is to use Lyapunov's direct method to determine the adaptive laws. At a high level, this approach requires constructing a positive definite scalar function V whose time derivative is negative definite. A full discussion of Lyapunov stability theory is outside the scope of this thesis, and the reader is forthwith assumed to have at least basic knowledge of Lyapunov analysis; otherwise it is strongly suggested the reader consult one of Refs. [19, 22, 57, 69] prior to proceeding. The derivation of the adaptive laws begins by assuming a candidate Lyapunov function

$$V(\boldsymbol{e},[\tilde{K}],[\tilde{G}],[\tilde{\Theta}]) = \underbrace{\boldsymbol{e}^{\mathsf{T}}[P]\boldsymbol{e}}_{1} + \underbrace{\operatorname{tr}\left([\tilde{K}]^{\mathsf{T}}[\Gamma_{\mathrm{k}}]^{-1}[\tilde{K}][\Lambda]\right)}_{2} + \underbrace{\operatorname{tr}\left([\tilde{G}]^{\mathsf{T}}[\Gamma_{\mathrm{g}}]^{-1}[\tilde{G}][\Lambda]\right)}_{3} + \underbrace{\operatorname{tr}\left([\tilde{\Theta}]^{\mathsf{T}}[\Gamma_{\Theta}]^{-1}[\tilde{\Theta}][\Lambda]\right)}_{4} \quad (5.13)$$

where $[P] = [P]^{\mathsf{T}} > 0$ is a solution to the Lyapunov function

$$[A_{\mathrm{m}}]^{\mathsf{T}}[P] + [P][A_{\mathrm{m}}] = -[Q]$$

where $[Q] = [Q]^{\mathsf{T}} > 0$ is a design parameter, and $[\Gamma_{\mathrm{k}}] > 0$, $[\Gamma_{\mathrm{g}}] > 0$, and $[\Gamma_{\Theta}] > 0$ are adaptive gains. In Eq. (5.13), Term 1 ensures stable error dynamics, Term 2 ensures a stable adaptive law for the gain $[K]$, Term 3 ensures a stable adaptive law for the gain $[G]$, and Term 4 ensures a stable adaptive law for the parameters $[\Theta]$. The $[\Lambda]$ terms are included to allow the adaptive laws to be developed without requiring knowledge of the control effectiveness.

Taking the derivative of Eq. (5.13) results in

$$\dot{V} = \dot{\boldsymbol{e}}^{\mathsf{T}}[P]\boldsymbol{e} + \boldsymbol{e}^{\mathsf{T}}[P]\dot{\boldsymbol{e}} + 2\operatorname{tr}\left(\left([\tilde{K}]^{\mathsf{T}}[\Gamma_{\mathrm{k}}]^{-1}[\dot{\hat{K}}] + [\tilde{G}]^{\mathsf{T}}[\Gamma_{\mathrm{g}}]^{-1}[\dot{\hat{G}}] + [\tilde{\Theta}]^{\mathsf{T}}[\Gamma_{\Theta}]^{-1}[\dot{\hat{\Theta}}]\right)[\Lambda]\right) \quad (5.14)$$

where the derivatives of the estimation errors are replaced with the derivatives of the adaptive parameter estimates since the true values are assumed constant. Substitute Eq. (5.12) into Eq. (5.14) to obtain

$$\begin{aligned}\dot{V} = \boldsymbol{e}^{\mathsf{T}}[A_{\mathrm{m}}]^{\mathsf{T}}[P]\boldsymbol{e} - \boldsymbol{x}^{\mathsf{T}}[\tilde{K}][\Lambda]^{\mathsf{T}}[B]^{\mathsf{T}}[P]\boldsymbol{e} + \boldsymbol{r}^{\mathsf{T}}[\tilde{G}][\Lambda]^{\mathsf{T}}[B]^{\mathsf{T}}[P]\boldsymbol{e} - \boldsymbol{\phi}^{\mathsf{T}}(\boldsymbol{x})[\tilde{\Theta}][\Lambda]^{\mathsf{T}}[B]^{\mathsf{T}} \\ + \boldsymbol{e}^{\mathsf{T}}[P][A_{\mathrm{m}}]\boldsymbol{e} - \boldsymbol{e}^{\mathsf{T}}[P][B][\Lambda][\tilde{K}]^{\mathsf{T}}\boldsymbol{x} + \boldsymbol{e}^{\mathsf{T}}[P][B][\Lambda][\tilde{G}]^{\mathsf{T}}\boldsymbol{r} - \boldsymbol{e}^{\mathsf{T}}[P][B][\Lambda][\tilde{\Theta}]^{\mathsf{T}}\boldsymbol{\phi}(\boldsymbol{x}) \\ + 2\,\mathrm{tr}\left(\left([\tilde{K}]^{\mathsf{T}}[\Gamma_{\mathrm{k}}]^{-1}[\dot{\hat{K}}] + [\tilde{G}]^{\mathsf{T}}[\Gamma_{\mathrm{g}}]^{-1}[\dot{\hat{G}}] + [\tilde{\Theta}]^{\mathsf{T}}[\Gamma_{\Theta}]^{-1}[\dot{\hat{\Theta}}]\right)[\Lambda]\right)\end{aligned}$$

which can be simplified to

$$\begin{aligned}\dot{V} = -\boldsymbol{e}^{\mathsf{T}}[Q]\boldsymbol{e} - 2\boldsymbol{e}^{\mathsf{T}}[P][B][\Lambda][\tilde{K}]^{\mathsf{T}}\boldsymbol{x} + 2\boldsymbol{e}^{\mathsf{T}}[P][B][\Lambda][\tilde{G}]^{\mathsf{T}}\boldsymbol{r} - 2\boldsymbol{e}^{\mathsf{T}}[P][B][\Lambda][\tilde{\Theta}]^{\mathsf{T}}\boldsymbol{\phi}(\boldsymbol{x}) \\ + 2\,\mathrm{tr}\left([\Lambda]^{\mathsf{T}}\left([\tilde{K}]^{\mathsf{T}}[\Gamma_{\mathrm{k}}]^{-1}[\dot{\hat{K}}] + [\tilde{G}]^{\mathsf{T}}[\Gamma_{\mathrm{g}}]^{-1}[\dot{\hat{G}}] + [\tilde{\Theta}]^{\mathsf{T}}[\Gamma_{\Theta}]^{-1}[\dot{\hat{\Theta}}]\right)\right)\,. \qquad (5.15)\end{aligned}$$

Recall the property of the trace that, for vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^n$, $\boldsymbol{a}^{\mathsf{T}}\boldsymbol{b} = \mathrm{tr}(\boldsymbol{b}\boldsymbol{a}^{\mathsf{T}})$. This property allows writing Eq. (5.15) as

$$\begin{aligned}\dot{V} = -\boldsymbol{e}^{\mathsf{T}}[Q]\boldsymbol{e} + 2\,\mathrm{tr}\left([\tilde{K}]^{\mathsf{T}}\left([\Gamma_{\mathrm{k}}]^{-1}[\dot{\hat{K}}] - \boldsymbol{x}\boldsymbol{e}^{\mathsf{T}}[P][B]\right)[\Lambda]\right) \\ + 2\,\mathrm{tr}\left([\tilde{G}]^{\mathsf{T}}\left([\Gamma_{\mathrm{g}}]^{-1}[\dot{\hat{G}}] + \boldsymbol{r}\boldsymbol{e}^{\mathsf{T}}[P][B]\right)[\Lambda]\right) \\ + 2\,\mathrm{tr}\left([\tilde{\Theta}]^{\mathsf{T}}\left([\Gamma_{\Theta}]^{-1}[\dot{\hat{\Theta}}] - \boldsymbol{\phi}(\boldsymbol{x})\boldsymbol{e}^{\mathsf{T}}[P][B]\right)[\Lambda]\right)\,. \qquad (5.16)\end{aligned}$$

Choosing the adaptive laws

$$[\dot{\hat{K}}] = [\Gamma_{\mathrm{k}}]\boldsymbol{x}\boldsymbol{e}^{\mathsf{T}}[P][B]\,, \qquad (5.17)$$

$$[\dot{\hat{G}}] = -[\Gamma_{\mathrm{g}}]\boldsymbol{r}\boldsymbol{e}^{\mathsf{T}}[P][B]\,, \text{ and} \qquad (5.18)$$

$$[\dot{\hat{\Theta}}] = [\Gamma_{\Theta}]\boldsymbol{\phi}(\boldsymbol{x})\boldsymbol{e}^{\mathsf{T}}[P][B] \qquad (5.19)$$

results in

$$\dot{V} = -\boldsymbol{e}^{\mathsf{T}}[Q]\boldsymbol{e} \leqslant 0 \qquad (5.20)$$

as desired. From Eq. (5.13) and Eq. (5.20) it follows that $\boldsymbol{e} \in \mathcal{L}_\infty \cap \mathcal{L}_2$ and that the estimation errors $[\tilde{K}]$, $[\tilde{G}]$, $[\tilde{\Theta}] \in \mathcal{L}_\infty$. Since the true parameters are assumed bounded and known, it follows that $[\hat{K}]$, $[\hat{G}]$, $[\hat{\Theta}] \in \mathcal{L}_\infty$, i.e. the adaptive parameters are bounded. The reference signal and reference model state are assumed bounded, so from $\boldsymbol{e} \in \mathcal{L}_\infty$ it follows that $\boldsymbol{x} \in \mathcal{L}_\infty$, and from Eq. (5.9) that $\boldsymbol{u} \in \mathcal{L}_\infty$ since $\boldsymbol{\phi}(\boldsymbol{x})$ is assumed bounded for bounded $\boldsymbol{x}$. Then, using Eq. (5.12) it follows that $\dot{\boldsymbol{e}} \in \mathcal{L}_\infty$. The adaptive laws Eqs. (5.17–5.19) ensure that the adaptive parameter derivatives are bounded. Finally, since $\boldsymbol{e} \in \mathcal{L}_\infty \cap \mathcal{L}_2$ and $\dot{\boldsymbol{e}} \in \mathcal{L}_\infty$, by Barbalat's lemma the tracking error $\boldsymbol{e} \to \boldsymbol{0}$ as $t \to \infty$.

5.3 Projection Operator

This section introduces the projection operator prior to discussing $\mathcal{L}_1$ adaptive control, which makes heavy use of projection-based adaptive laws. The projection operator guarantees that the adaptive estimates remain bounded at all times and allows for adaptive laws that ensure the derivative of the Lyapunov function is negative semi-definite. Accordingly, projection-based adaptive laws are a mainstay of robust adaptive control theory. Given an adaptive parameter $\boldsymbol{\theta}$, the projection operator ensures that $\|\boldsymbol{\theta}\| \leqslant \theta_{\max}$. The continuous projection operator can be defined as

$$\mathrm{Proj}(\boldsymbol{\theta}, [\Gamma]\boldsymbol{y}, \boldsymbol{f}) = [\Gamma] \begin{cases} \boldsymbol{y} - \dfrac{\boldsymbol{\nabla f}(\boldsymbol{\nabla f})^{\mathsf{T}}}{\|\boldsymbol{\nabla f}\|_{[\Gamma]}^2}[\Gamma]\boldsymbol{y}f & \text{if } [f > 0 \wedge \boldsymbol{y}^{\mathsf{T}}[\Gamma]\boldsymbol{\nabla f} > \boldsymbol{0}] \\ \boldsymbol{y} & \text{otherwise} \end{cases} \tag{5.21}$$

where $\boldsymbol{y}$ defines the adaptive law dynamics, Γ is an adaptive gain, and $f(\boldsymbol{\theta})$ is a convex function [70]. In Eq. (5.21), the weighted norm $\|\boldsymbol{\nabla f}\|_{[\Gamma]}^2 \triangleq (\boldsymbol{\nabla f})^{\mathsf{T}}[\Gamma]\boldsymbol{\nabla f}$. In general Eq. (5.21) is written as $\mathrm{Proj}(\boldsymbol{\theta}, [\Gamma]\boldsymbol{y})$. The projection operator can be used to restrict $\boldsymbol{\theta}$ to a ball of radius $\theta_{\max}$ centered on $\boldsymbol{\theta}_{\mathrm{c}}$ by using $\boldsymbol{\theta} - \boldsymbol{\theta}_{\mathrm{c}}$ as the argument.

Two convex sets result from this definition of the projection operator:

$$\begin{aligned}\Omega_0 &\triangleq \{\boldsymbol{\theta} \in \mathbb{R}^n \mid f(\boldsymbol{\theta}) \leqslant 0\}\,,\\ \Omega_1 &\triangleq \{\boldsymbol{\theta} \in \mathbb{R}^n \mid f(\boldsymbol{\theta}) \leqslant 1\}\,.\end{aligned}$$

The projection operator ensures that for initial conditions $\boldsymbol{\theta}(0) \in \Omega_0$, the parameter estimates $\boldsymbol{\theta}(t) \in \Omega_1$ for all $t \geqslant 0$. For $\boldsymbol{\theta} \in \Omega_0$, the projection operator does not modify the adaptation law $\boldsymbol{y}$. As $0 \leqslant f(\boldsymbol{\theta}) \leqslant 1$, the projection operator subtracts a component normal to the boundary from $\boldsymbol{y}$ up until the boundary of Ω_1, as which point $\boldsymbol{y}$ becomes tangent to Ω_1 [70].

The convex function $f(\boldsymbol{\theta}, \theta_{\max}, \varepsilon_\theta)$ is used to ensure that the norm of the parameter vector $\boldsymbol{\theta}$ remains less than a specified maximum value $\theta_{\max}$, subject to a tolerance $\varepsilon_\theta \in (0, 1]$. In this thesis, the convex function

$$f(\boldsymbol{\theta}, \theta_{\max}, \varepsilon_\theta) = \frac{(1+\varepsilon_\theta)\boldsymbol{\theta}^{\mathsf{T}}\boldsymbol{\theta} - \theta_{\max}^2}{\varepsilon_\theta \theta_{\max}^2} \tag{5.22}$$

is used. For the convex function Eq. (5.22), the sets Ω_0 and Ω_1 are equivalent to

$$\begin{aligned}\Omega_0 &= \left\{\boldsymbol{\theta} \;\middle|\; \|\boldsymbol{\theta}\| \leqslant \frac{\theta_{\max}}{\sqrt{1+\varepsilon_\theta}}\right\},\\ \Omega_1 &= \{\boldsymbol{\theta} \mid \|\boldsymbol{\theta}\| \leqslant \theta_{\max}\}\,.\end{aligned}$$

The projection operator thus ensures that the adaptive parameter estimates $\boldsymbol{\theta}$ are uniformly ultimately bounded, allowing for more robust adaptive estimates.

The projection operator has the property that, for any $[\Gamma] = [\Gamma]^{\mathsf{T}} > 0$,

$$(\boldsymbol{\theta} - \boldsymbol{\theta}^{\star})^{\mathsf{T}} \left([\Gamma]^{-1}\operatorname{Proj}(\boldsymbol{\theta}, [\Gamma]\boldsymbol{y}) - \boldsymbol{y}\right) \leqslant 0 \tag{5.23}$$

where $\boldsymbol{\theta}^\star \in \Omega_1$ is an interior point of the convex set. This relation takes a form common in the development of adaptive laws, as can be seen in Eq. (5.16). Accordingly, Eq. (5.23) shows that the projection-based adaptive laws maintain the negative semi-definiteness of the Lyapunov function derivative, and thus maintain the boundedness and other stability properties of the non-projection-based adaptive laws.

Finally, a matrix version of the projection operator may be defined as

$$\mathrm{Proj}([\Theta], [\Gamma][Y], \boldsymbol{f}) = [\mathrm{Proj}(\boldsymbol{\theta}_1, [\Gamma]\boldsymbol{y}_1, f_1), \ldots, \mathrm{Proj}(\boldsymbol{\theta}_m, [\Gamma]\boldsymbol{y}_m, f_m)]$$

where $[\Theta] = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_m]$, $[Y] = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m]$, and $\boldsymbol{f} = [f_1, \ldots, f_m]^\mathsf{T}$. This form generalizes the projection operator for MIMO systems. Additional properties of the projection operator and their proofs may be found in Ref. [70].

5.4 $\mathcal{L}_1$ Adaptive Control

The $\mathcal{L}_1$ adaptive control theory introduced by Cao and Hovakimyan is a form of adaptive controller that combines a state predictor architecture and a filtering structure, as seen in Fig. 5.2 [66, 67]. The adaptive laws operate on the prediction error, and the filtering structure allows for high adaptive gains while remaining within the bandwidth of the control effectors. This fast adaptation allows the development of a closed-loop reference model which can be used with classical stability analysis techniques to evaluate transient performance.

The $\mathcal{L}_1$ adaptive control design process consists of

1. Designing a stable state predictor for the dynamical system.

2. Designing high-gain adaptive laws based on the prediction error.

3. Designing a low-pass filtered control law.

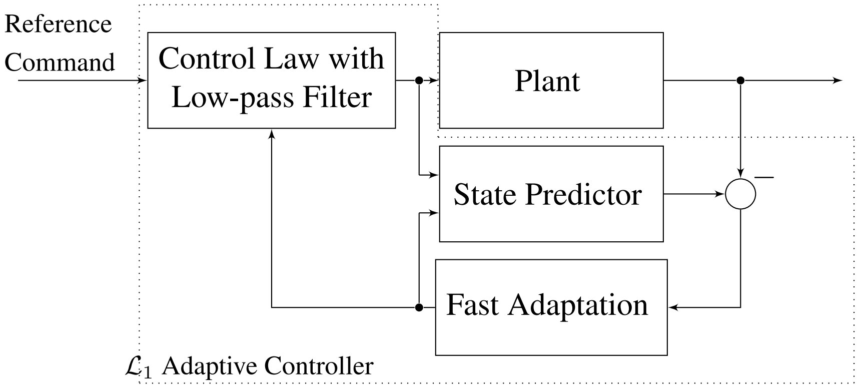


Figure 5.2: $\mathcal{L}_1$ adaptive controller block diagram. [69]

4. Analyzing stability using a closed-loop reference system.

An example of this process follows for a linear system with a nonlinear matched uncertainty and an uncertain control effectiveness. This example is loosely based on the $\mathcal{L}_1$ adaptive controller in Section 2.2 of Ref. [69]. Consider a dynamical system of the form

$$\begin{aligned}\dot{\boldsymbol{x}} &= [A]\boldsymbol{x} + [B]([\Lambda]\boldsymbol{u} + \boldsymbol{f}(\boldsymbol{x}))\,,\\ \boldsymbol{y} &= [C]\boldsymbol{x}\,,\end{aligned}$$

where $([A],[B])$ are known, $[\Lambda]$ is an unknown constant control effectiveness matrix confined to an *a priori* known compact set, $\boldsymbol{u}$ is the control law, and $\boldsymbol{f}(\boldsymbol{x})$ is an unknown nonlinear matched uncertainty. Assume that the function $\boldsymbol{f}(\boldsymbol{x})$ may be parameterized as $\boldsymbol{f}(\boldsymbol{x}) = [\Theta]^{\mathsf{T}}\boldsymbol{\phi}(\boldsymbol{x})$. Partition the control law as $\boldsymbol{u} = \boldsymbol{u}_{\text{bl}} + \boldsymbol{u}_{\text{ad}}$, where $\boldsymbol{u}_{\text{bl}}$ is a nonadaptive baseline control law and $\boldsymbol{u}_{\text{ad}}$ is an adaptive augmentation control signal. Assume that the control law $\boldsymbol{u}_{\text{bl}}$ results in a system of the form

$$\dot{\boldsymbol{x}} = [A_{\text{m}}]\boldsymbol{x} + [B]([\Lambda]\boldsymbol{u}_{\text{ad}} + [\Theta]^{\mathsf{T}}\boldsymbol{\phi}(\boldsymbol{x})) \tag{5.24}$$

where $[A_\mathrm{m}]$ is a Hurwitz matrix with desired dynamics.

Using Eq. (5.24), a state predictor can be formed as

$$\dot{\hat{\boldsymbol{x}}} = [A_\mathrm{m}]\hat{\boldsymbol{x}} + [B]([\hat{\Lambda}]\boldsymbol{u}_\mathrm{ad} + [\hat{\Theta}]^\mathsf{T}\boldsymbol{\phi}(\boldsymbol{x})) \tag{5.25}$$

where $\hat{\boldsymbol{x}}$ is the predicted state, and $[\hat{\Lambda}]$ and $[\hat{\Theta}]$ are the adaptive estimates of $[\Lambda]$ and $[\Theta]$, respectively. Defining the prediction error as $\boldsymbol{e} = \hat{\boldsymbol{x}} - \boldsymbol{x}$ and using Eqs. (5.24–5.25), the prediction error dynamics are

$$\dot{\boldsymbol{e}} = [A_\mathrm{m}]\boldsymbol{e} + [B]([\tilde{\Lambda}]\boldsymbol{u}_\mathrm{ad} + [\tilde{\Theta}]^\mathsf{T}\boldsymbol{\phi}(\boldsymbol{x})) \tag{5.26}$$

where $[\tilde{\Lambda}] = [\hat{\Lambda}] - [\Lambda]$ and $[\tilde{\Theta}] = [\hat{\Theta}] - [\Theta]$.

Consider the candidate Lyapunov function

$$V(\boldsymbol{e}, [\tilde{\Lambda}], [\tilde{\Theta}]) = \underbrace{\boldsymbol{e}^\mathsf{T}[P]\boldsymbol{e}}_{1} + \mathrm{tr}\left(\underbrace{[\tilde{\Lambda}]^\mathsf{T}[\Gamma_\Lambda]^{-1}[\tilde{\Lambda}]}_{2} + \underbrace{[\tilde{\Theta}]^\mathsf{T}[\Gamma_\Theta]^{-1}[\tilde{\Theta}]}_{3}\right) \tag{5.27}$$

where Term 1 ensures stable tracking dynamics, and Terms 2–3 ensure stable adaptive laws. The time derivative of Eq. (5.27) is

$$\dot{V} = \dot{\boldsymbol{e}}^\mathsf{T}[P]\boldsymbol{e} + \boldsymbol{e}^\mathsf{T}[P]\dot{\boldsymbol{e}} + 2\,\mathrm{tr}\left([\tilde{\Lambda}]^\mathsf{T}[\Gamma_\Lambda]^{-1}[\dot{\hat{\Lambda}}] + [\tilde{\Theta}]^\mathsf{T}[\Gamma_\Theta]^{-1}[\dot{\hat{\Theta}}]\right)$$

which, using Eq. (5.26) becomes

$$\begin{aligned}\dot{V} = -\boldsymbol{e}^\mathsf{T}[Q]\boldsymbol{e} &+ 2\,\mathrm{tr}\left[[\tilde{\Lambda}]^\mathsf{T}\left([\Gamma_\Lambda]^{-1}[\dot{\hat{\Lambda}}] + [B]^\mathsf{T}[P]\boldsymbol{e}\boldsymbol{u}_\mathrm{ad}^\mathsf{T}\right)\right] \\ &+ 2\,\mathrm{tr}\left[[\tilde{\Theta}]^\mathsf{T}\left([\Gamma_\Theta]^{-1}[\dot{\hat{\Theta}}] + \boldsymbol{\phi}(\boldsymbol{x})\boldsymbol{e}^\mathsf{T}[P][B]\right)\right]\end{aligned} \tag{5.28}$$

where $[A_\mathrm{m}]^\mathsf{T}[P] + [P][A_\mathrm{m}] = -[Q]$ and the trace property have been used to simplify the derivative as in the development of the direct MRAC adaptive laws. Choosing the adaptive laws

$$[\dot{\hat{\Lambda}}] = \operatorname{Proj}\left([\hat{\Lambda}], -[\Gamma_\Lambda][B]^\mathsf{T}[P]\boldsymbol{e}\boldsymbol{u}_\mathrm{ad}^\mathsf{T}\right) \tag{5.29}$$

$$[\dot{\hat{\Theta}}] = \operatorname{Proj}\left([\hat{\Theta}], -[\Gamma_\Theta]\boldsymbol{\phi}(\boldsymbol{x})\boldsymbol{e}^\mathsf{T}[P][B]\right) \tag{5.30}$$

and using the property Eq. (5.23) results in

$$\dot{V} \leqslant -\boldsymbol{e}^\mathsf{T}[Q]\boldsymbol{e} \leqslant 0$$

as desired.

The control law is

$$\boldsymbol{u}_\mathrm{ad}(s) = -[K][D(s)]\left(\boldsymbol{\eta}(s) - [K_\mathrm{g}]\boldsymbol{r}(s)\right) , \tag{5.31}$$

$$\boldsymbol{\eta}(t) = [\hat{\Theta}]^\mathsf{T}\boldsymbol{\phi}(\boldsymbol{x}) + [\hat{\Lambda}]\boldsymbol{u}_\mathrm{ad} , \tag{5.32}$$

where $[K_\mathrm{g}] = -([C][A_\mathrm{m}]^{-1}[B])^{-1}$, $[K] > 0$, and $[D(s)]$ is a strictly proper transfer function such that

$$[C(s)] = [\Lambda][K][D(s)]([I_n] + [\Lambda][K][D(s)])^{-1}$$

is strictly proper, stable, and has DC gain $[C(0)] = [I_n]$.

It now remains to establish that the state predictor states are bounded. In combination with the boundedness of the prediction error, shown above, boundedness of the state predictor implies that the true system states are bounded. Consider the state predictor system Eq. (5.25). The fast adaptation allows the assumption that the true system parameters are known instantaneously, which results in a reference system

$$\dot{\boldsymbol{x}}_{\text{ref}} = [A_{\text{m}}]\boldsymbol{x}_{\text{ref}} + [B]\left([\Lambda]\boldsymbol{u}_{\text{ref}} + [\Theta]^{\mathsf{T}}\boldsymbol{\phi}(\boldsymbol{x})\right) . \tag{5.33}$$

The Laplace transform $\mathscr{L}(\dot{\boldsymbol{x}}_{\text{ref}}) = s\boldsymbol{x}_{\text{ref}} - \boldsymbol{x}(0)$, which allows rewriting Eq. (5.33) as

$$\boldsymbol{x}_{\text{ref}}(s) = [H(s)]\left([\Lambda]\boldsymbol{u}_{\text{ref}}(s) + [\Theta]^{\mathsf{T}}\boldsymbol{\phi}(\boldsymbol{x})\right) + \boldsymbol{x}_{\text{in}}(s) \tag{5.34}$$

where $[H(s)] \triangleq (s[I_n] - [A_{\text{m}}])^{-1}[B]$ and $\boldsymbol{x}_{\text{in}}(s) \triangleq (s[I_n] - [A_{\text{m}}])^{-1}\boldsymbol{x}(0)$. Here the latter term represents the dynamics due to the initial conditions. Using Eq. (5.31), Eq. (5.34) becomes

$$\boldsymbol{x}_{\text{ref}}(s) = [H(s)]\left(-[C(s)]\boldsymbol{\eta}_{\text{ref}}(s) + [H(s)][C(s)][K_{\text{g}}]\boldsymbol{r}(s) + [\Theta]^{\mathsf{T}}\boldsymbol{\phi}(\boldsymbol{x})\right) + \boldsymbol{x}_{\text{in}}(s) \tag{5.35}$$

where $\boldsymbol{\eta}_{\text{ref}}(t) = [\Theta]^{\mathsf{T}}\boldsymbol{\phi}(\boldsymbol{x})$. Equation (5.35) simplifies to

$$\boldsymbol{x}_{\text{ref}}(s) = [G(s)]\boldsymbol{\eta}_{\text{ref}}(s) + [H(s)][C(s)][K_{\text{g}}]\boldsymbol{r}(s) + \boldsymbol{x}_{\text{in}}(s) \tag{5.36}$$

where $[G(s)] \triangleq [H(s)]([I_n] - [C(s)])$.

Consider a generic system with input/output relationship $y(s) = G(s)u(s)$. If the system dynamics $G(s) \in \mathcal{L}_1$, then the system satisfies

$$\|y_\tau(t)\|_\infty \leqslant \|G(s)\|_1 \|u_\tau(t)\|_\infty$$

where the subscript τ indicates truncation at $t \geqslant \tau$. This relationship states that for inputs u that are bounded without finite escapes, the output y is bounded. Applying this to the system Eq. (5.36) leads to

$$\left\|\boldsymbol{x}_{\mathrm{ref},\tau}(s)\right\|_\infty \leqslant \left\|[G(s)]\right\|_1 \left\|\boldsymbol{\eta}_{\mathrm{ref},\tau}(s)\right\|_\infty + \left\|[H(s)][C(s)][K_{\mathrm{g}}]\right\|_1 \left\|\boldsymbol{r}_\tau(s)\right\|_\infty + \left\|\boldsymbol{x}_{\mathrm{in},\tau}(s)\right\|_\infty$$

By the definition of $\boldsymbol{\eta}(t)$, $\left\|\boldsymbol{\eta}_{\mathrm{ref},\tau}(s)\right\|_\infty \leqslant \left\|[\theta]\right\|_1 \left\|\boldsymbol{\phi}_\tau(\boldsymbol{x})\right\|_\infty = L\left\|\boldsymbol{\phi}_\tau(\boldsymbol{x})\right\|_\infty$, where $L \triangleq \max_{\theta\in\Theta}\left\|[\theta]\right\|_1$ so

$$\left\|\boldsymbol{x}_{\mathrm{ref},\tau}(s)\right\|_\infty \leqslant \left\|[G(s)]\right\|_1 L \left\|\boldsymbol{\phi}_\tau(\boldsymbol{x})\right\|_\infty + \left\|[H(s)][C(s)][K_{\mathrm{g}}]\right\|_1 \left\|\boldsymbol{r}_\tau(s)\right\|_\infty + \left\|\boldsymbol{x}_{\mathrm{in},\tau}(s)\right\|_\infty . \tag{5.37}$$

Using Lemma A.8.1 of Ref. [69], assume that $\boldsymbol{\phi}(\boldsymbol{x})$ may be parameterized as

$$\boldsymbol{\phi}(\boldsymbol{x}) = \boldsymbol{\vartheta}(t)\left\|\boldsymbol{x}\right\|_\infty + \boldsymbol{\sigma}(t)$$

and thus

$$\left\|\boldsymbol{\phi}_\tau(\boldsymbol{x})\right\|_\infty \leqslant \left\|\boldsymbol{\vartheta}(t)\right\|_1 \left\|\boldsymbol{x}\right\|_\infty + \left\|\boldsymbol{\sigma}_\tau(t)\right\|_\infty = L_{\phi 1}\left\|\boldsymbol{x}\right\|_\infty + L_{\phi 2} . \tag{5.38}$$

Substituting Eq. (5.38) in Eq. (5.37) results in

$$\begin{aligned}\left\|\boldsymbol{x}_{\mathrm{ref},\tau}(s)\right\|_\infty \leqslant{}& \left\|[G(s)]\right\|_1 L\left(L_{\phi 1}\left\|\boldsymbol{x}_{\mathrm{ref},\tau}\right\|_\infty + L_{\phi 2}\right)\\ &+ \left\|[H(s)][C(s)][K_{\mathrm{g}}]\right\|_1 \left\|\boldsymbol{r}_\tau(s)\right\|_\infty + \left\|\boldsymbol{x}_{\mathrm{in},\tau}(s)\right\|_\infty .\end{aligned} \tag{5.39}$$

Solving for $\left\|\boldsymbol{x}_{\mathrm{ref},\tau}(s)\right\|_\infty$ results in

$$\left\|\boldsymbol{x}_{\mathrm{ref},\tau}(s)\right\|_\infty \leqslant \frac{\left\|[H(s)][C(s)][K_{\mathrm{g}}]\right\|_1 \left\|\boldsymbol{r}_\tau(s)\right\|_\infty \left\|[G(s)]\right\|_1 L L_{\phi 2} + \left\|\boldsymbol{x}_{\mathrm{in},\tau}(s)\right\|_\infty}{1 - \left\|[G(s)]\right\|_1 L L_{\phi 1}} \tag{5.40}$$

which is bounded provided that

$$\left\| [G(s)] \right\|_1 L L_{\phi 1} < 1\,.$$

5.5 Dead Zone Modification

The dead zone modification is used to create robust adaptive laws in the presence of unmatched disturbances. The dead zone modification changes the adaptive law from $\dot{\hat{\theta}} = \Gamma y$ to

$$\dot{\hat{\theta}} = \begin{cases} \Gamma y & \text{if } \|e\| > e_0 \\ 0 & \text{otherwise} \end{cases} \tag{5.41}$$

where $y = y(e)$ and the scalar case has been shown without loss of generality. This modification stops adaptation when the error is less than a specified threshold, which recovers uniform boundedness of the error e and estimation error $\tilde{\theta}$ in the presence of unmatched disturbances. Consequently, the dead zone modification prevents the phenomenon of parameter drift resulting from unmatched uncertainties/disturbances, although at the cost of losing asymptotic stability guarantees.

Note that Eq. (5.41) is not continuous, which can lead to difficulty with analysis and computation [19]. In Ref. [22], a Lipschitz-continuous version of the dead zone modification is proposed, where the function

$$\mu(\|e\|) = \max\left(0, \min\left(1, \frac{\|e\| - \delta e_0}{(1-\delta)e_0}\right)\right) \tag{5.42}$$

is inserted into the adaptive law to implement the dead zone modification. In Eq. (5.42), the parameter $0 < \delta < 1$ is a tolerance that parameterizes a ramp between zero and one in the dead zone modification which makes $\mu(\|e\|)$ piecewise-continuous. Using the adaptive law Eq. (5.30) as an example, the dead zone modification Eq. (5.42) results in an adaptive law

$$[\dot{\hat{\Theta}}] = \text{Proj}\left([\hat{\Theta}], -[\Gamma_\Theta]\boldsymbol{\phi}(\boldsymbol{x})\mu(\|\boldsymbol{e}\|)\boldsymbol{e}^\mathsf{T}[P][B]\right)$$

where the dead zone modification prevents parameter drift and the projection operator ensures boundedness of the adaptive parameters and provides anti-windup protection.

5.6 F-16 Simulation Examples

Two simulation examples are presented for the direct MRAC and linear $\mathcal{L}_1$ adaptive controllers described above. These examples use the same F-16 model described in Section 3 but are implemented in MATLAB instead of Simulink. Actuator dynamics are neglected as these have a destabilizing effect on the linear adaptive controllers due to the time delay and nonlinearities introduced by position and rate limiting. These examples do not include an autothrottle.

Additionally, these examples use a different form of desired dynamics. Second-order desired dynamics for the pitch rate and sideslip angle are replaced with first-order dynamics. This change is introduced due to the model-matching conditions required by the linear adaptive controllers: the form of the plant models and reference models must be compatible. Pilot commands remain those of Fig. 3.14. The linear controllers are designed with a plant model of the form

$$\dot{\boldsymbol{x}} = [A]\boldsymbol{x} + [B]\left[\boldsymbol{u} + [\Theta]^\mathsf{T}\boldsymbol{\phi}(\boldsymbol{x})\right] \tag{5.43}$$

where $\boldsymbol{x} = [p, q, r]^\mathsf{T}$, and thus the reference model is of the form

$$\dot{\boldsymbol{x}}_\text{m} = [A_\text{m}]\boldsymbol{x}_\text{m} + [B_\text{m}]\boldsymbol{r}$$

where $\boldsymbol{x}_\text{m} = [p_\text{des}, q_\text{des}, r_\text{des}]^\mathsf{T}$. The state-space form of the first-order desired dynamics is then

$$\begin{Bmatrix} \dot{p}_{\mathrm{des}} \\ \dot{q}_{\mathrm{des}} \\ \dot{r}_{\mathrm{des}} \end{Bmatrix} = \begin{bmatrix} -K_{\mathrm{p}} & 0 & 0 \\ 0 & -K_{\mathrm{q}} & 0 \\ 0 & 0 & -K_{\mathrm{r}} \end{bmatrix} \begin{Bmatrix} p_{\mathrm{des}} \\ q_{\mathrm{des}} \\ r_{\mathrm{des}} \end{Bmatrix} + \begin{bmatrix} K_{\mathrm{p}} & 0 & 0 \\ 0 & K_{\mathrm{q}} & 0 \\ 0 & 0 & K_{\mathrm{r}} \end{bmatrix} \begin{Bmatrix} p_{\mathrm{cmd}} \\ q_{\mathrm{cmd}} \\ r_{\mathrm{cmd}} \end{Bmatrix} .$$

As before, the commanded yaw rate is generated using sideslip desired dynamics. These are first-order as well, and the desired yaw rate is obtained using the simplified relation

$$r_{\mathrm{cmd}} \approx -\dot{\beta}_{\mathrm{des}} \,.$$

Table 5.1 lists the gains for the proportional desired dynamics.

Table 5.1: First-order desired dynamics for linear adaptive controllers.

Design Parameter	**Value**
K_{p}	2
K_{q}	5
K_{r}	2.5
K_{β}	2

5.6.1 Direct MRAC CAS

This example consists of a Direct MRAC linear adaptive controller based on Eq. (5.3) with adaptive laws Eqs. (5.17– 5.19). The controller is used to implement the CAS described in Section 3 with the modifications described above. Table 5.2 lists the adaptive controller design parameters for the direct MRAC controller. The basis function vector

Table 5.2: Direct MRAC controller design parameters.

Design Parameter	**Value**
$[Q]$	$[I_3]$
$[\Gamma_{\mathrm{k}}]$	$75[I_3]$
$[\Gamma_{\mathrm{g}}]$	$75[I_3]$
$[\Gamma_{\Theta}]$	$75[I_{16}]$

$\boldsymbol{\phi}(\boldsymbol{x})$ is defined as

$$\boldsymbol{\phi}(\boldsymbol{x}) = [\kappa_1, \alpha\kappa_1, \beta\kappa_2, \hat{p}\kappa_2, \hat{q}\kappa_1, \hat{r}\kappa_2, \alpha\hat{q}\kappa_1, \beta^2\kappa_2, \alpha^2\hat{q}\kappa_1, \beta^3\kappa_2, \alpha^3\hat{q}\kappa_1, \alpha^4\kappa_1, rq, pq, pr, p^2 - r^2]^{\mathsf{T}} \quad (5.44)$$

where $\kappa_1 \triangleq \bar{q}S\bar{c}$ and $\kappa_2 \triangleq \bar{q}Sb$. These values result from the basis functions used in the GNA model Eqs. (2.49–2.51) and nonlinear rate terms from the rotational dynamics Eq. (2.17). The terms in Eq. (5.44) are relatively large, which leads to small adaptive parameters. The initial values of the adaptive parameters are all set to zero, assuming no knowledge of the system. Since the adaptive parameters are relatively small, this has a relatively small impact on performance. The control influence matrix $[B]$ is calculated at the trim condition and is held constant throughout the simulation.

Figure 5.3a plots the p, q, r, β, V_{T}, and α time histories for the direct MRAC CAS. The controller shows almost exact tracking of the p, q, and r desired dynamics. This excellent performance must be balanced, however, against the fact that this is an academic example with perfect knowledge of the state and no actuator dynamics. The lack of actuator dynamics is a critical factor in this example; adding the actuator dynamics described in Section 3 is sufficient to destabilize the adaptive controller. Without projection, the adaptive estimates become unbounded, while with projection the performance degrades significantly and the adaptive parameter estimates oscillate between the projection bounds. In Fig. 5.3a the airspeed V_{T} is open-loop, and the AOA is influenced solely through the pitch rate channel. Both of these states remain at reasonable values throughout the flight time.

Figure 5.3b plots the time histories of the aerodynamic control effectors aileron, elevator, and rudder. The controls stay within the position limits for the entire elapsed time of the simulation, but violate rate constraints with high-frequency transients seen at various simulation times.

Finally, Figure 5.4 plots the time histories of the adaptive parameters for the controller matrices $[\hat{K}]$ and $[\hat{G}]$ and the nonlinear matched uncertainty parameter estimate $[\hat{\Theta}]$. Note that the large parameter matrix $[\hat{\Theta}]$ is plotted column-wise; these columns correspond to the parameters for the roll, pitch, and yaw axes. Note that the adaptive parameter estimates are all small. This is expected for the $[\hat{\Theta}]$ parameters as the elements of $\boldsymbol{\phi}(\boldsymbol{x})$ are relatively large in magnitude. The small magnitudes of the $[\hat{K}]$ and $[\hat{G}]$ estimates suggest that the DMRAC controller is primarily using the term $[\hat{\Theta}]^{\mathsf{T}}\boldsymbol{\phi}$ to control the system and uses the linear controller gain matrices to make small corrections.

5.6.2 $\mathcal{L}_1$ Adaptive Control CAS

This section presents another F-16 CAS design, this time using the $\mathcal{L}_1$-AC technique presented in this section. In order to formulate the nonlinear F-16 model in the form of Eq. (5.24), assume that the system can be represented by a reduced-order linear model in the form of Eq. (5.43) where the $[A]$ and $[B]$ matrices are obtained by linearizing the nonlinear model about a trim condition and the nonlinear matched uncertainty $[\Theta]^{\mathsf{T}}\boldsymbol{\phi}(\boldsymbol{x})$ captures the unmodeled linearities. The linearized $[A]$ matrix for the flight condition of interest is

$$[A] = \begin{bmatrix} -2.9491 & 0.0003 & 0.4977 \\ 0 & -1.1032 & -0.0029 \\ -0.0128 & 0.0025 & -0.3973 \end{bmatrix}.$$

First design a baseline, nonadaptive control $\boldsymbol{u}_{\mathrm{bl}}$ so that the partially closed-loop system takes the form of the $\mathcal{L}_1$ design system Eq. (5.24). In this example, the MATLAB `place` function, which implements the robust pole placement algorithm introduced by Ref. [71], is used to determine a gain matrix $[K_{\mathrm{bl}}]$ such that

$$[A] - [B][K_{\mathrm{bl}}] = [A_{\mathrm{m}}]\,,$$

where

$$[A_{\mathrm{m}}] = \begin{bmatrix} -K_{\mathrm{p}} & 0 & 0 \\ 0 & -K_{\mathrm{q}} & 0 \\ 0 & 0 & -K_{\mathrm{r}} \end{bmatrix}.$$

Using this approach results in the nonadaptive controller gain

$$[K_{\mathrm{bl}}] = \begin{bmatrix} 0.0183 & -0.0001 & -0.0786 \\ 0 & -0.3181 & 0.0002 \\ -0.0078 & -0.0006 & -0.4525 \end{bmatrix}.$$

Yaw rate commands are generated from the first order sideslip desired dynamics. As with the direct MRAC design, desired dynamics are generated using the parameters in Table 5.1. The gain for the tracking controller is then $[K_{\mathrm{g}}] = -([C][A_{\mathrm{m}}]^{-1}[B])^{-1} = -[B]^{-1}[A_{\mathrm{m}}]$ assuming full-state feedback with state vector $\boldsymbol{x} = [p, q, r]^{\mathsf{T}}$; explicitly, this matrix is

$$[K_{\mathrm{g}}] = \begin{bmatrix} -0.0377 & 0 & -0.0822 \\ 0 & -0.4081 & 0 \\ 0.0223 & 0 & -0.5445 \end{bmatrix}$$

for this example. The basis functions Eq. (5.44) are used for the function $\boldsymbol{f}$ as in the previous direct MRAC example.

The $\mathcal{L}_1$ adaptive controller is comprised of the state predictor Eq. (5.25), adaptive laws Eqs. (5.29–5.30), and control laws Eqs. (5.31–5.32). The transfer function $[D(s)]$ is chosen as $[D(s)] = [I_3]/s$, such that

$$\dot{\boldsymbol{u}}_{\mathrm{ad}} = [K_{\mathrm{L1}}] \left([B]^{-1}[A_{\mathrm{m}}]\boldsymbol{r}(t) - [\hat{\Theta}]^{\mathsf{T}}\boldsymbol{\phi}(\boldsymbol{x}) - [\hat{\Lambda}]\boldsymbol{u}_{\mathrm{ad}} \right) .$$

The adaptive law for $[\hat{\Lambda}]$ is centered on $[I_3]$ with $\theta_{\max} = 1$ to ensure that $[\hat{\Lambda}] > 0$. This approach allows for modeling errors of up to 100% for the control effectiveness. The control effectiveness estimate is initialized to identity, the adaptive parameters $[\hat{\Theta}(0)] = [0_{16\times3}]$, and the control signal $\boldsymbol{u}_{\mathrm{ad}}(0) = \boldsymbol{u}_{\mathrm{trim}}$.

Table 5.3 lists the values for the tunable parameters of the $\mathcal{L}_1$ controller design. Note the large adaptive gains $[\Gamma_\Lambda]$ and $[\Gamma_\Theta]$ that are the trademarks of $\mathcal{L}_1$ adaptive controllers. By comparison, the direct MRAC design in Table 5.2 uses gains on the order of 100.

Table 5.3: $\mathcal{L}_1$-AC controller design parameters.

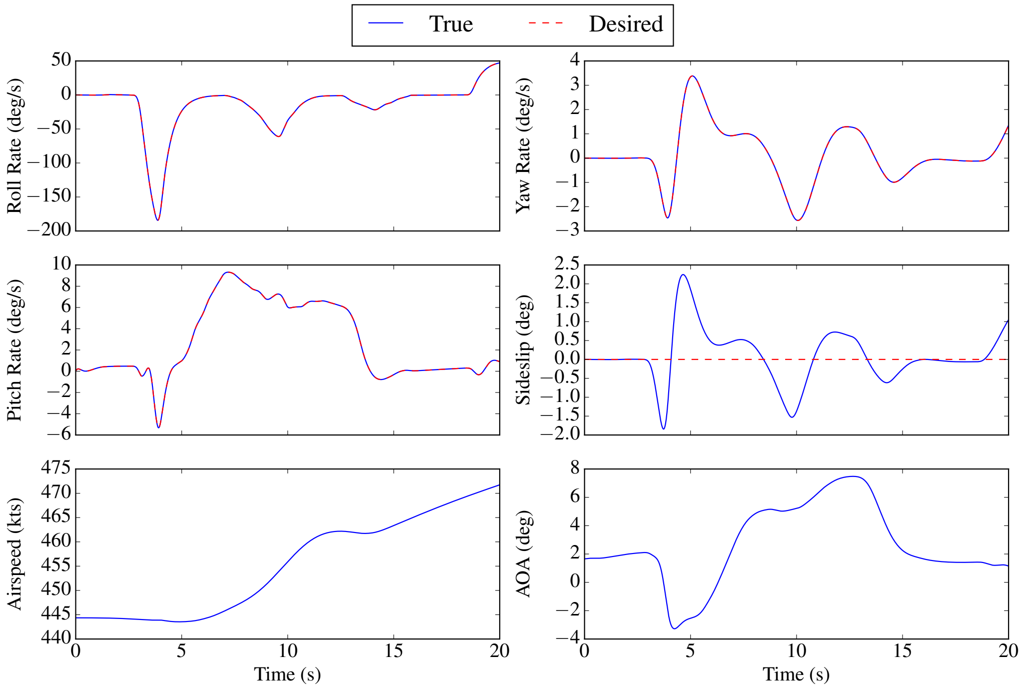
Design Parameter	**Value**
$[Q]$	$[I_3]$
$[K_{\mathrm{L1}}]$	$\mathrm{diag}(100, 150, 75)$
$[\Gamma_\Lambda]$	$1 \times 10^6 [I_3]$
$[\Gamma_\Theta]$	$1 \times 10^6 [I_{16}]$

Figure 5.5a plots the time histories of the aircraft states p, q, r,, β, V_{T}, and α. The controller shows excellent tracking performance of the angular rates, and is able to regulate sideslip to approximately $2°$. Airspeed and angle-of-attack responses are similar to those of the direct MRAC controller. The airspeed is again an open-loop response, and the angle-of-attack is not a controlled state. As with the MRAC controller actuator dynamics

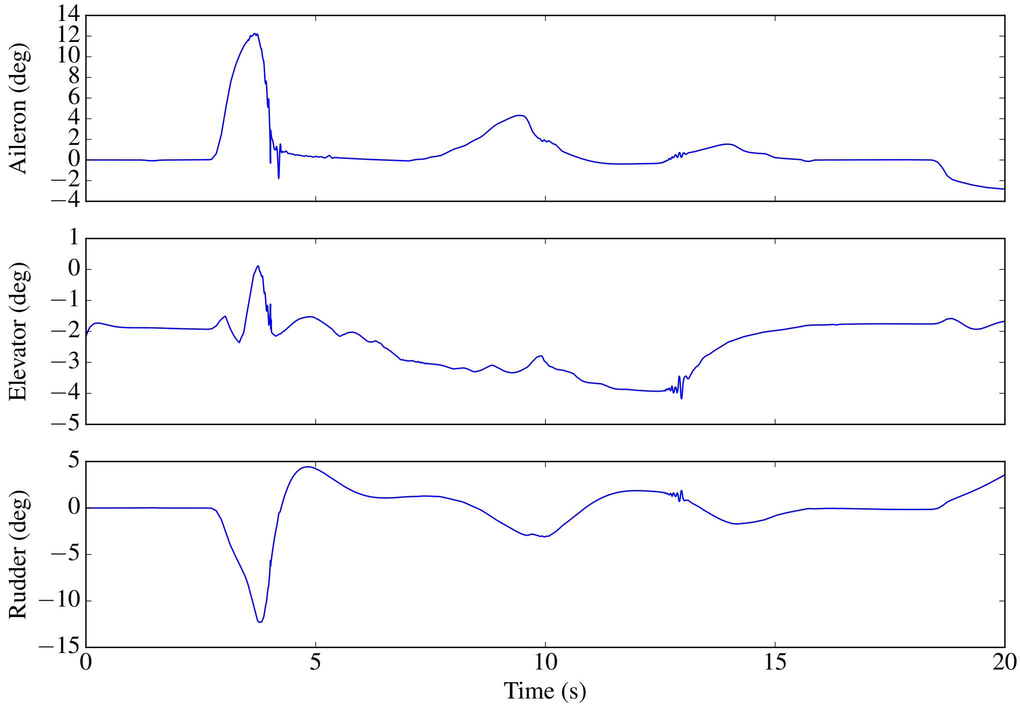
and other practical concerns are neglected, making this another academic example. In this light, the near-ideal tracking performance shown cannot be considered realizable.

Time histories of the aerodynamic control effectors are plotted in Fig. 5.5b. The total aileron, elevator, and rudder deflections are plotted, along with the commanded deflections resulting from the nonadaptive baseline control $\boldsymbol{u}_{\mathrm{bl}}$ and the adaptive augmentation $\boldsymbol{u}_{\mathrm{ad}}$. The $\mathcal{L}_1$ controller remains within the actuator limits and demands similar authority as the direct MRAC controller. Note however that in contrast with the direct MRAC controller in Fig. 5.3b, the control signals in Fig. 5.5b are smooth and relatively non-oscillatory. This results from the lowpass filtering structure in the $\mathcal{L}_1$ architecture, which limits the controller to compensation within the available bandwidth of the system.

Finally, time histories of the adaptive parameters $[\hat{\Lambda}]$ and $[\hat{\Theta}]$ are shown in Fig. 5.6. Note that for the control effectiveness matrix $[\hat{\Lambda}]$, the deviations $[\delta\hat{\Lambda}] = [\hat{\Lambda}] - [I_3]$ are plotted in order to have all of the adaptive elements have similar orders of magnitude. The values of $[\delta\hat{\Lambda}]$ are small since the linearized estimate of the $[B]$ matrix is accurate throughout the part of the flight envelope visited in the simulation. As in Fig. 5.4 the parameter matrix $[\hat{\Theta}]$ is plotted column-wise. These values are small since the basis function vector $\boldsymbol{\phi}(\boldsymbol{x})$ is comparatively large due to the $\bar{q}S$ factors. Additionally, inertial effects make the true values of $[\Theta]$ relatively small.



(a) Actual and desired state time histories.



(b) Aerodynamic control effector time histories.

Figure 5.3: Direct MRAC controller state and control time histories.

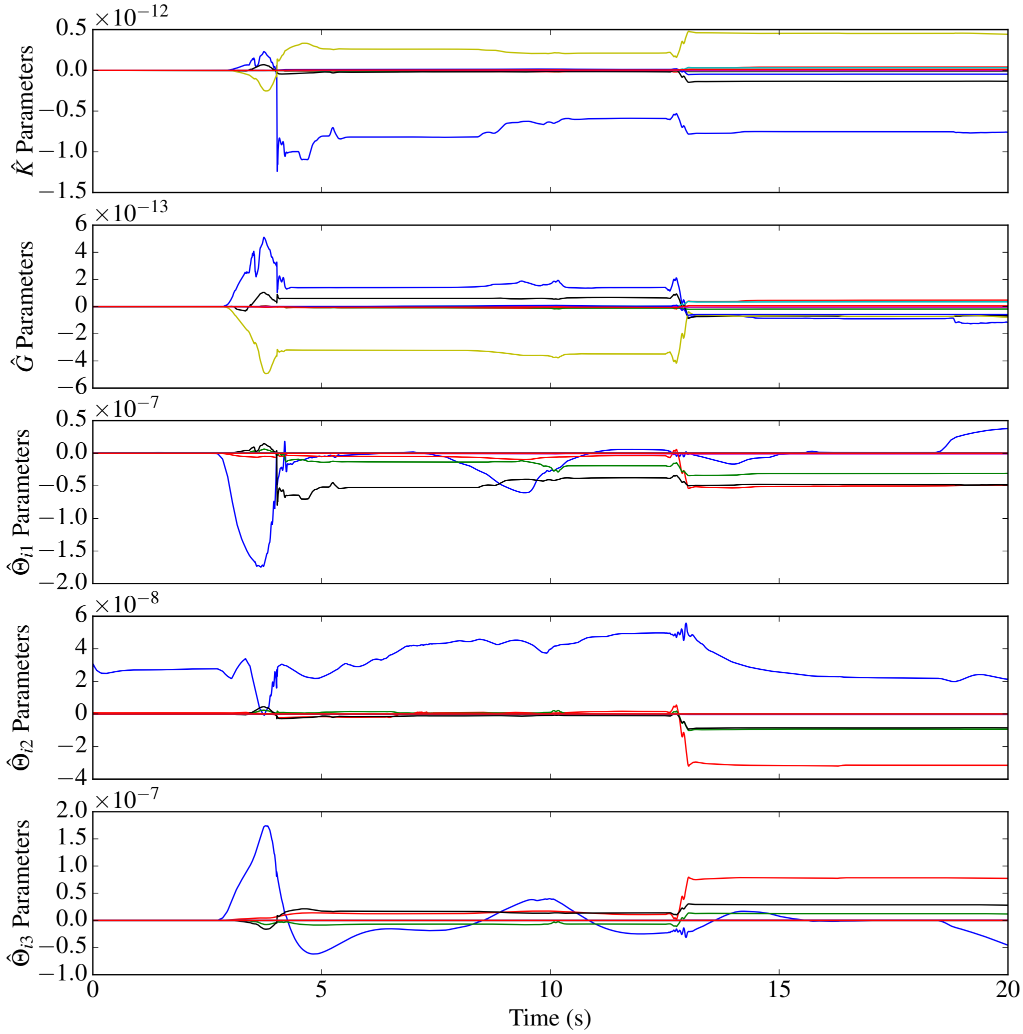
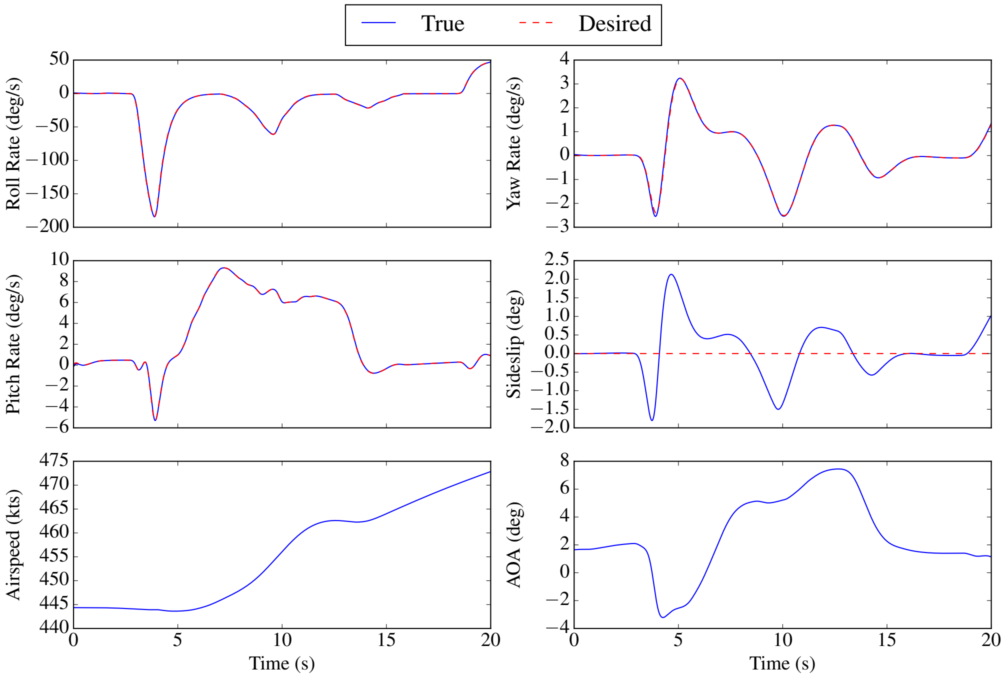
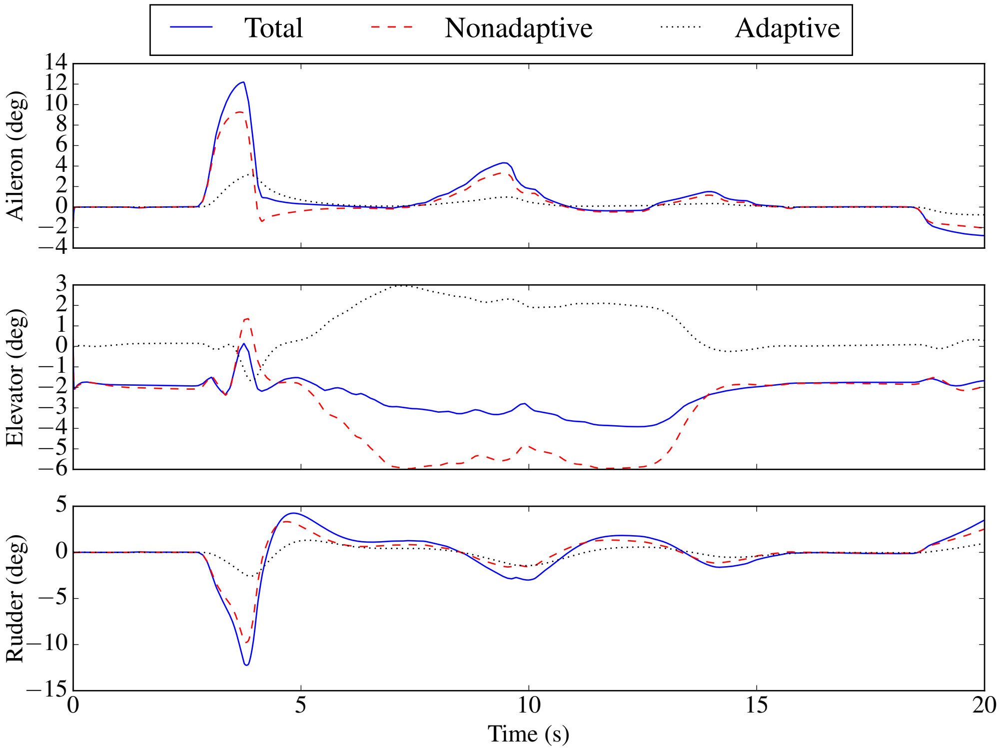


Figure 5.4: Direct MRAC adaptive parameter time histories.



(a) Actual and desired state time histories.



(b) Aerodynamic control effector time histories.

Figure 5.5: Linear $\mathcal{L}_1$-AC controller state and control time histories.

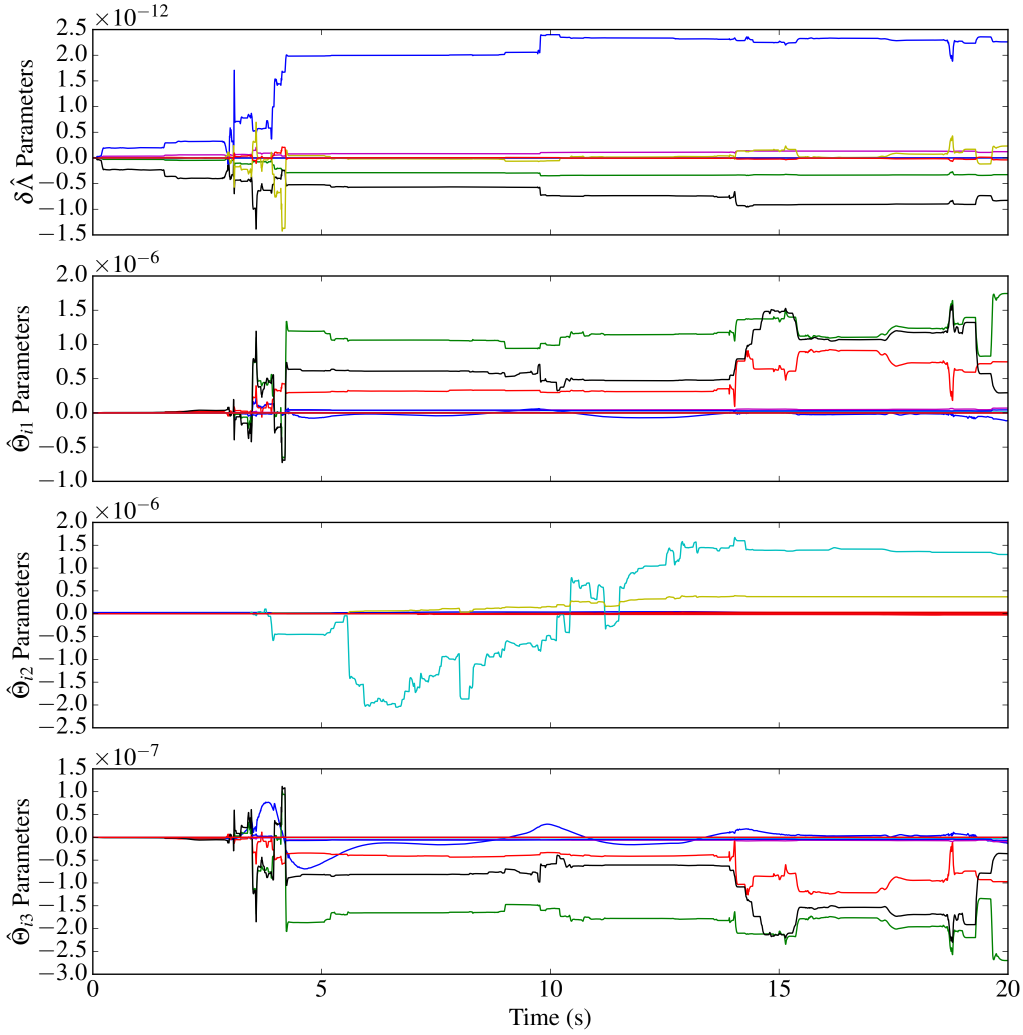


Figure 5.6: Linear $\mathcal{L}_1$-AC adaptive parameter time histories.

6. NONLINEAR ADAPTIVE DYNAMIC INVERSION CONTROL FOR IN-FLIGHT SIMULATION

6.1 Introduction

The theoretical perfect tracking of nonlinear dynamic inversion controllers is a strong motivation for their use for in-flight simulators (IFS). Simultaneously, the inherent lack of robustness suggests the use of adaptive control techniques in concert with NDI baseline controllers. The use of adaptive augmentation in particular is advantageous as the learning properties of adaptive controllers allow them to compensate for factors such as modeling errors and unmodeled dynamics. These benefits of adaptation can allow the use of lower fidelity onboard models, reducing cost of control law development, although doing so in a certifiable manner is still largely an open research problem. This is particularly useful for the SUAS application as in many cases high fidelity models of such vehicles are not available. The combination of NDI and adaptive augmentation with rapid aerodynamic model generation tools such as SimGen [72] have the potential to solve many of the roadblocks with model-based control design for SUAS.

This section presents two nonlinear adaptive dynamic inversion control laws designed for implementing in-flight simulation capabilities. The first is a direct MRAC based design similar to others found in the literature, while the second is an $\mathcal{L}_1$ adaptive control analogue novel to the best knowledge of the author. Both adaptive controllers are developed using Lyapunov-based techniques, with the $\mathcal{L}_1$ controller additionally using a stability analysis based on signal norms similar to those found in Ref. [69]. The motivation of these two control laws is variable stability and IFS systems for SUAS but they are applicable to a wider range of flight control problems. The basic command augmentation system design problem used as an example for nonadaptive NDI controllers and linear adaptive controllers in Sections 4–5 is again used to illustrate these controllers. Results specific for the variable stability system and IFS are presented in Section 7.

6.2 Direct Model Reference Adaptive Control NDI

This section presents an adaptive NDI control law designed using a direct MRAC approach similar to that used by Rollins [23] and Famularo [24, 73]. Consider the nonlinear dynamics in the form

$$\dot{\boldsymbol{X}} = \boldsymbol{F}(\boldsymbol{X}, \boldsymbol{U}^\star) + [B][\Lambda]\boldsymbol{u}\,, \tag{6.1}$$

where $\boldsymbol{F}(\boldsymbol{X}, \boldsymbol{U}^\star)$ is a vector of the dynamics evaluated at state $\boldsymbol{X}$ and some control deflection $\boldsymbol{U}^\star$, $[B]$ is a known control distribution that is locally linear about $\boldsymbol{U}^\star$, and $[\Lambda]$ is an unknown control effectiveness matrix that is assumed constant and diagonally dominant.

6.2.1 Control Law

The resulting dynamic inversion control law is based off of Eq. (4.21), where the pseudo-control $\boldsymbol{\nu}$ is assumed to consist of proportional error dynamics and an adaptive signal $\boldsymbol{\nu}_{\text{ad}}$ yet to be designed:

$$\begin{aligned} \boldsymbol{u} &= ([B][\Lambda])^{-1}\left(\dot{\boldsymbol{X}}_{\text{des}} + [K]\boldsymbol{E} + \boldsymbol{\nu}_{\text{ad}} - \boldsymbol{F}(\boldsymbol{X}, \boldsymbol{U}^\star)\right)\,, \\ \boldsymbol{U} &= \boldsymbol{U}^\star + \boldsymbol{u}\,. \end{aligned} \tag{6.2}$$

In Eq. (6.2) the vector $\boldsymbol{E} \triangleq \boldsymbol{X} - \boldsymbol{X}_{\text{des}}$ is the error between the true state and the desired state and the Hurwitz matrix $[K]$ is chosen by the designer to specify the desired error dynamics. The desired dynamics are assumed to have a reference model of the form

$$\dot{\boldsymbol{X}}_{\text{des}} = [A_{\text{m}}]\boldsymbol{X}_{\text{des}} + [B_{\text{m}}]\boldsymbol{R} \tag{6.3}$$

where $[A_{\text{m}}]$ is Hurwitz and $\boldsymbol{R}$ is a reference command.

In practice, $\boldsymbol{F}$ and $[\Lambda]$ are replaced by their estimates $\hat{\boldsymbol{F}}$ and $[\hat{\Lambda}]$ to obtain the implementable control law

$$\begin{aligned} \hat{\boldsymbol{u}} &= \left([B][\hat{\Lambda}]\right)^{-1} \left(\dot{\boldsymbol{X}}_{\text{des}} + [K]\boldsymbol{E} + \boldsymbol{\nu}_{\text{ad}} - \hat{\boldsymbol{F}}(\boldsymbol{X}, \boldsymbol{U}^{\star})\right) , \\ \boldsymbol{U} &= \boldsymbol{U}^{\star} + \hat{\boldsymbol{u}} . \end{aligned} \tag{6.4}$$

In this case, $\hat{\boldsymbol{F}}(\boldsymbol{X}, \boldsymbol{U}^{\star})$ is the estimate of $\boldsymbol{F}(\boldsymbol{X}, \boldsymbol{U}^{\star})$ from the onboard model, and $[\hat{\Lambda}]$ is an adaptive estimate of $[\Lambda]$. The adaptive signal $\boldsymbol{\nu}_{\text{ad}}$ will be designed to account for the error $\tilde{\boldsymbol{F}} \triangleq \hat{\boldsymbol{F}} - \boldsymbol{F}$. This approach is chosen over using an adaptive law to estimate $\hat{\boldsymbol{F}}$ as it allows the use of an existing OBM over a nonlinear envelope to approximate the dynamics.

6.2.2 Error Dynamics

The error dynamics are used to determine the adaptive laws. The time derivative of the error $\dot{\boldsymbol{E}} = \dot{\boldsymbol{X}} - \dot{\boldsymbol{X}}_{\text{des}}$ can be expressed using Eq. (6.1) as

$$\dot{\boldsymbol{E}} = \boldsymbol{F}(\boldsymbol{X}, \boldsymbol{U}^{\star}) + [B][\Lambda]\boldsymbol{u} - \dot{\boldsymbol{X}}_{\text{des}} . \tag{6.5}$$

Next, add and subtract the term $[B][\hat{\Lambda}]\boldsymbol{u}$ to obtain

$$\dot{\boldsymbol{E}} = \boldsymbol{F}(\boldsymbol{X}, \boldsymbol{U}^{\star}) + [B][\hat{\Lambda}]\boldsymbol{u} - [B][\tilde{\Lambda}]\boldsymbol{u} - \dot{\boldsymbol{X}}_{\text{des}} , \tag{6.6}$$

where $[\tilde{\Lambda}] \triangleq [\hat{\Lambda}] - [\Lambda]$. Using Eq. (6.4) with Eq. (6.6) results in

$$\begin{aligned} \dot{\boldsymbol{E}} = \boldsymbol{F}(\boldsymbol{X}, \boldsymbol{U}^{\star}) + [B][\hat{\Lambda}] \left([B][\hat{\Lambda}]\right)^{-1} \left(\dot{\boldsymbol{X}}_{\text{des}} + [K]\boldsymbol{E} + \boldsymbol{\nu}_{\text{ad}} - \hat{\boldsymbol{F}}(\boldsymbol{X}, \boldsymbol{U}^{\star})\right) \\ - [B][\tilde{\Lambda}]\boldsymbol{u} - \dot{\boldsymbol{X}}_{\text{des}} , \end{aligned}$$

which simplifies to

$$\dot{\boldsymbol{E}} = [K]\boldsymbol{E} + \boldsymbol{\nu}_{\mathrm{ad}} - \tilde{\boldsymbol{F}}(\boldsymbol{X}, \boldsymbol{U}^{\star}) - [B][\tilde{\Lambda}]\boldsymbol{u}\,.$$

Assume that the error in the dynamics $\tilde{\boldsymbol{F}}$ can be parameterized as

$$\tilde{\boldsymbol{F}} \triangleq [\Theta]^{\mathsf{T}}\boldsymbol{\phi}\,,$$

where $[\Theta] \in \mathbb{R}^{p\times n}$ is a set of unknown constant parameters and $\boldsymbol{\phi} \in \mathbb{R}^{p}$ is a set of p known basis functions that are functions of the state and the aerodynamic angles. Choosing

$$\boldsymbol{\nu}_{\mathrm{ad}} \triangleq [\hat{\Theta}]^{\mathsf{T}}\boldsymbol{\phi}\,,$$

where $[\hat{\Theta}]$ is the adaptive estimate of $[\Theta]$ results in the error dynamics

$$\dot{\boldsymbol{E}} = [K]\boldsymbol{E} + [\tilde{\Theta}]^{\mathsf{T}}\boldsymbol{\phi} - [B][\tilde{\Lambda}]\boldsymbol{u}\,. \tag{6.7}$$

Equation (6.7) is the error dynamics in a useful form for determining the adaptive laws, as it contains known Hurwitz dynamics and error terms between the known and adaptive parameters.

6.2.3 Lyapunov Design of Adaptive Laws

To determine the adaptive laws, consider the candidate Lyapunov function

$$V(\boldsymbol{E}, [\tilde{\Theta}], [\tilde{\Lambda}]) = \boldsymbol{E}^{\mathsf{T}}[P]\boldsymbol{E} + \mathrm{tr}\left([\tilde{\Theta}]^{\mathsf{T}}[\Gamma_{\Theta}]^{-1}[\tilde{\Theta}]\right) + \mathrm{tr}\left([\tilde{\Lambda}]^{\mathsf{T}}[\Gamma_{\Lambda}]^{-1}[\tilde{\Lambda}]\right)\,. \tag{6.8}$$

In Eq. (6.8), the first term of the Lyapunov function represents the reference tracking error dynamics, the second term represents the the adaptive estimation error of $[\hat{\Theta}]$, and the third term represents the adaptive estimation error of $[\hat{\Lambda}]$. The symmetric matrix $[P]$ is a solution to a Lyapunov equation

$$[K]^{\mathsf{T}}[P] + [P][K] = -[Q]$$

where $[Q] = [Q]^{\mathsf{T}} > 0$ is chosen by the designer, and the matrices $[\Gamma_\Theta] = [\Gamma_\Theta]^{\mathsf{T}} > 0$ and $[\Gamma_\Lambda] = [\Gamma_\Lambda]^{\mathsf{T}} > 0$ are the rates of adaptation for $[\hat{\Theta}]$ and $[\hat{\Lambda}]$, respectively.

Taking the time derivative of Eq. (6.8), obtain

$$\dot{V} = \dot{\boldsymbol{E}}^{\mathsf{T}}[P]\boldsymbol{E} + \boldsymbol{E}^{\mathsf{T}}[P]\dot{\boldsymbol{E}} + 2\,\mathrm{tr}\left([\tilde{\Theta}]^{\mathsf{T}}[\Gamma_\Theta]^{-1}[\dot{\hat{\Theta}}]\right) + 2\,\mathrm{tr}\left([\tilde{\Lambda}]^{\mathsf{T}}[\Gamma_\Lambda]^{-1}[\dot{\hat{\Lambda}}]\right)$$

where the fact that the true values of $[\Lambda]$ and $[\Theta]$ are assumed constant has been used to express the derivatives of the adaptive estimation errors in terms of the derivatives of the adaptive parameter estimates. Substitute Eq. (6.7) to obtain, after some simplification,

$$\begin{aligned}\dot{V} = {} & -\boldsymbol{E}^{\mathsf{T}}[Q]\boldsymbol{E} + 2\boldsymbol{E}^{\mathsf{T}}[P][\tilde{\Theta}]^{\mathsf{T}}\boldsymbol{\phi} - 2\boldsymbol{u}^{\mathsf{T}}[\tilde{\Lambda}]^{\mathsf{T}}[B]^{\mathsf{T}}[P]\boldsymbol{E} \\ & + 2\,\mathrm{tr}\left([\tilde{\Theta}]^{\mathsf{T}}[\Gamma_\Theta]^{-1}[\dot{\hat{\Theta}}]\right) + 2\,\mathrm{tr}\left([\tilde{\Lambda}]^{\mathsf{T}}[\Gamma_\Lambda]^{-1}[\dot{\hat{\Lambda}}]\right) . \end{aligned} \tag{6.9}$$

Recall the trace identity that, for two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$,

$$\boldsymbol{a}^{\mathsf{T}}\boldsymbol{b} = \mathrm{tr}(\boldsymbol{b}\boldsymbol{a}^{\mathsf{T}})\,,$$

which will allow combining the last four terms in Eq. (6.9). Equation (6.9) is repeated below with the "$\boldsymbol{a}$" and "$\boldsymbol{b}$" terms shown explicitly:

$$\dot{V} = -\boldsymbol{E}^{\mathsf{T}}[Q]\boldsymbol{E} + 2\,\underbrace{\boldsymbol{E}^{\mathsf{T}}[P]}_{\boldsymbol{a}^{\mathsf{T}}}\underbrace{[\tilde{\Theta}]^{\mathsf{T}}\boldsymbol{\phi}}_{\boldsymbol{b}} - 2\,\underbrace{\boldsymbol{u}^{\mathsf{T}}}_{\boldsymbol{a}^{\mathsf{T}}}\underbrace{[\tilde{\Lambda}]^{\mathsf{T}}[B]^{\mathsf{T}}[P]\boldsymbol{E}}_{\boldsymbol{b}} \\ + 2\,\mathrm{tr}\left([\tilde{\Theta}]^{\mathsf{T}}[\Gamma_{\Theta}]^{-1}[\dot{\hat{\Theta}}]\right) + 2\,\mathrm{tr}\left([\tilde{\Lambda}]^{\mathsf{T}}[\Gamma_{\Lambda}]^{-1}[\dot{\hat{\Lambda}}]\right)\,.$$

This then results in

$$\dot{V} = -\boldsymbol{E}^{\mathsf{T}}[Q]\boldsymbol{E} + 2\,\mathrm{tr}\left[[\tilde{\Theta}]^{\mathsf{T}}\left([\Gamma_{\Theta}]^{-1}[\dot{\hat{\Theta}}] + \boldsymbol{\phi}\boldsymbol{E}^{\mathsf{T}}[P]\right)\right] \\ + 2\,\mathrm{tr}\left[[\tilde{\Lambda}]^{\mathsf{T}}\left([\Gamma_{\Lambda}]^{-1}[\dot{\hat{\Lambda}}] - [B]^{\mathsf{T}}[P]\boldsymbol{E}\boldsymbol{u}^{\mathsf{T}}\right)\right]\,, \quad (6.10)$$

which with Eq. (5.23) suggests the projection-based adaptive laws

$$[\dot{\hat{\Theta}}] = \mathrm{Proj}\left([\hat{\Theta}], -[\Gamma_{\Theta}]\boldsymbol{\phi}\mu(\|\boldsymbol{E}\|)\boldsymbol{E}^{\mathsf{T}}[P]\right)\,, \quad (6.11)$$

$$[\dot{\hat{\Lambda}}] = \mathrm{Proj}\left([\hat{\Lambda}], [\Gamma_{\Lambda}][B]^{\mathsf{T}}[P]\mu(\|\boldsymbol{E}\|)\boldsymbol{E}\boldsymbol{u}^{\mathsf{T}}\right)\,, \quad (6.12)$$

where the dead zone modification is also used. Since the dead zone modification is used the adaptive controller defined by Eq. (6.4) and Eqs. (6.11–6.12) can only guarantee that the error $\boldsymbol{E}$ is uniformly ultimately bounded and not that $\boldsymbol{E} \to \mathbf{0}$ as $t \to \infty$. The reduced theoretical stability is offset by the robustness of the adaptive laws for practical implementation. Table 6.1 presents a summary of the MRAC adaptive NDI control law.

6.3 $\mathcal{L}_1$ Adaptive Dynamic Inversion

This section introduces an adaptive nonlinear dynamic inversion controller using an $\mathcal{L}_1$ approach. The controller uses the fast adaptation to learn a parametric model of the dynamics $\hat{\boldsymbol{F}}(\boldsymbol{X}, \boldsymbol{U}^{\star})$ and control effectiveness $[\Lambda]$. A state predictor architecture is developed first, followed by robust adaptive laws, an $\mathcal{L}_1$ control law, and finally a stability

Table 6.1: Model reference adaptive nonlinear dynamic inversion control law.

Model	$\dot{\boldsymbol{X}} = \boldsymbol{F}(\boldsymbol{X}, \boldsymbol{U}^\star) + [B][\Lambda]\boldsymbol{u}$
Reference Model	$\dot{\boldsymbol{X}}_{\text{des}} = [A_{\text{m}}]\boldsymbol{X}_{\text{des}} + [B_{\text{m}}]\boldsymbol{r}$
Error Dynamics	$\boldsymbol{E} = \boldsymbol{X} - \boldsymbol{X}_{\text{des}}$ $\dot{\boldsymbol{E}} = [K]\boldsymbol{E} + [\tilde{\Theta}]^{\mathsf{T}}\boldsymbol{\phi} - [B][\tilde{\Lambda}]\boldsymbol{u}$
Adaptive Laws	$[\dot{\hat{\Theta}}] = \operatorname{Proj}\Big([\hat{\Theta}], -[\Gamma_\Theta]\boldsymbol{\phi}\mu(\lVert\boldsymbol{E}\rVert)\boldsymbol{E}^{\mathsf{T}}[P]\Big)$ $[\dot{\hat{\Lambda}}] = \operatorname{Proj}\Big([\hat{\Lambda}], [\Gamma_\Lambda][B]^{\mathsf{T}}[P]\mu(\lVert\boldsymbol{E}\rVert)\boldsymbol{E}\boldsymbol{u}^{\mathsf{T}}\Big)$
Control Law	$\boldsymbol{u} = \Big([B][\hat{\Lambda}]\Big)^{-1}\Big(\dot{\boldsymbol{X}}_{\text{des}} + [K]\boldsymbol{E} + [\hat{\Theta}]^{\mathsf{T}}\boldsymbol{\phi} - \hat{\boldsymbol{F}}(\boldsymbol{X}, \boldsymbol{U}^\star)\Big)$ $\boldsymbol{U} = \boldsymbol{U}^\star + \boldsymbol{u}$

analysis.

6.3.1 State Predictor

The $\mathcal{L}_1$ Adaptive Control theory uses a state-predictor architecture, so it is first necessary to determine the state predictor model. Consider the dynamics of Eq. (6.1), but with the vector $\boldsymbol{u}$ as a ganged pseudocontrol vector such that u_i controls the i-th state. Let $\boldsymbol{\delta}$ be the control effector deflection vector such that

$$\boldsymbol{u} = [G]\boldsymbol{\delta}\,. \tag{6.13}$$

As a consequence it follows that $[B] = [I_n]$. This formulation is introduced for two reasons. First, it ensures a stable lowpass filter for the $\mathcal{L}_1$ control architecture. Second, it allows for the use of a control allocation algorithm for aircraft with control effectors beyond the conventional aileron-elevator-rudder triplet.

Add and subtract a term $[A_{\text{m}}]\boldsymbol{X}$, where $[A_{\text{m}}]$ is a Hurwitz matrix chosen by the designer, to the dynamics Eq. (6.1) so that

$$\dot{\boldsymbol{X}} = [A_{\mathrm{m}}]\boldsymbol{X} + \boldsymbol{G}(\boldsymbol{X}, \boldsymbol{U}^{\star}) + [\Lambda]\boldsymbol{u} \tag{6.14}$$

where $\boldsymbol{G} \triangleq \boldsymbol{F} - [A_{\mathrm{m}}]\boldsymbol{X}$. The state predictor is then

$$\dot{\hat{\boldsymbol{X}}} = [A_{\mathrm{m}}]\hat{\boldsymbol{X}} + \hat{\boldsymbol{G}}(\boldsymbol{X}, \boldsymbol{U}^{\star}) + [\hat{\Lambda}]\boldsymbol{u} \quad ; \quad \hat{\boldsymbol{X}}(0) = \boldsymbol{X}(0) \tag{6.15}$$

where $\hat{\boldsymbol{G}}(\boldsymbol{X}, \boldsymbol{U}^{\star}) = \hat{\boldsymbol{F}}(\boldsymbol{X}, \boldsymbol{U}^{\star}) - [A_{\mathrm{m}}]\boldsymbol{X}$.

6.3.2 Adaptive Laws

Defining the state prediction error $\boldsymbol{E} \triangleq \hat{\boldsymbol{X}} - \boldsymbol{X}$, Eqs. (6.14–6.15) then result in the prediction error dynamics

$$\dot{\boldsymbol{E}} = [A_{\mathrm{m}}]\boldsymbol{E} + \tilde{\boldsymbol{G}}(\boldsymbol{X}, \boldsymbol{U}^{\star}) + [\tilde{\Lambda}]\boldsymbol{u}\,.$$

Note that the $\mathcal{L}_1$ adaptive controller structure derived in this section differs from the MRAC-based controller in that the model is estimated and not the model error. Since $\mathcal{L}_1$ uses fast adaptation for nonlinear parametric models it may be advantageous in computation and memory to avoid the need for table lookups as the model parameters can be initialized to the best known *a priori* estimates. If this turns out not to be the case for a particular application the $\mathcal{L}_1$ controller may be formulated to correct the model errors as in the previous section by following the modifications presented in Section 6.3.5. Assume that the function $\boldsymbol{G}$ and its estimate $\hat{\boldsymbol{G}}$ may be parameterized as $[\Theta]^{\mathsf{T}}\boldsymbol{\phi}$ and $[\hat{\Theta}]^{\mathsf{T}}\boldsymbol{\phi}$, respectively, where as before $\boldsymbol{\phi}$ is a function of the states and aerodynamic parameters. Then, the error dynamics may be expressed as

$$\dot{\boldsymbol{E}} = [A_\mathrm{m}]\boldsymbol{E} + [\tilde{\Theta}]^\mathsf{T}\boldsymbol{\phi} + [\tilde{\Lambda}]\boldsymbol{u}\,. \tag{6.16}$$

Consider the same Lyapunov function candidate as previous, Eq. (6.8), reproduced below:

$$V(\boldsymbol{E}, [\tilde{\Theta}], [\tilde{\Lambda}]) = \boldsymbol{E}^\mathsf{T}[P]\boldsymbol{E} + \operatorname{tr}\left([\tilde{\Theta}]^\mathsf{T}[\Gamma_\Theta]^{-1}[\tilde{\Theta}]\right) + \operatorname{tr}\left([\tilde{\Lambda}]^\mathsf{T}[\Gamma_\Lambda]^{-1}[\tilde{\Lambda}]\right)\ .$$

Note that the error vector has a different definition than from the MRAC control law. Taking the time derivative and using Eq. (6.16) results in

$$\begin{aligned}\dot{V} = -\boldsymbol{E}^\mathsf{T}[Q]\boldsymbol{E} &+ 2\operatorname{tr}\left[[\tilde{\Theta}]^\mathsf{T}\left([\Gamma_\Theta]^{-1}[\dot{\hat{\Theta}}] + \boldsymbol{\phi}\boldsymbol{E}^\mathsf{T}[P]\right)\right] \\ &+ 2\operatorname{tr}\left[[\tilde{\Lambda}]^\mathsf{T}\left([\Gamma_\Lambda]^{-1}[\dot{\hat{\Lambda}}] + [P]\boldsymbol{E}\boldsymbol{u}^\mathsf{T}\right)\right]\ ,\end{aligned} \tag{6.17}$$

which results in similar adaptive laws

$$[\dot{\hat{\theta}}] = \operatorname{Proj}\left([\hat{\Theta}], -[\Gamma_\Theta]\boldsymbol{\phi}\mu(\|\boldsymbol{E}\|)\boldsymbol{E}^\mathsf{T}[P]\right)\ , \tag{6.18}$$

$$[\dot{\hat{\Lambda}}] = \operatorname{Proj}\left([\hat{\Lambda}], -[\Gamma_\Lambda][P]\mu(\|\boldsymbol{E}\|)\boldsymbol{E}\boldsymbol{u}^\mathsf{T}\right)\ . \tag{6.19}$$

These adaptive laws are again based on the projection operator and the dead zone modification. As previous, the matrix $[Q]$ is chosen by the designer such that the Lyapunov equation $[A_\mathrm{m}]^\mathsf{T}[P] + [P][A_\mathrm{m}] = -[Q]$ is satisfied.

6.3.3 Control Law

The $\mathcal{L}_1$ Adaptive Dynamic Inversion control law is then given by

$$\boldsymbol{\eta}(t) = \dot{\boldsymbol{X}}_{\mathrm{des}} - [K_\epsilon]\boldsymbol{\epsilon} - [A_{\mathrm{m}}]\boldsymbol{X} - [\hat{\Theta}]^{\mathsf{T}}\boldsymbol{\phi} - [\hat{\Lambda}]\boldsymbol{u}\,, \tag{6.20a}$$

$$\boldsymbol{u}(s) = [K][D(s)][\boldsymbol{\eta}(t)]\,, \tag{6.20b}$$

where $[K]$ is a gain matrix and the strictly proper transfer function matrix $[D(s)]$ operates on the Laplace transform of the signal $\boldsymbol{\eta}(t)$. The signal $[K_\epsilon]\boldsymbol{\epsilon}$ is a proportional feedback of the tracking error $\boldsymbol{\epsilon} \triangleq \boldsymbol{X} - \boldsymbol{X}_{\mathrm{des}}$ and the matrix $[K_\epsilon]$ is a small gain chosen by the designer such that $-[K_\epsilon]\boldsymbol{\epsilon}$ is stabilizing. This term aids the tracking performance of the control law as otherwise the controller Eq. (6.20) has no feedback of the tracking error. The total pseudocontrol is then $\boldsymbol{U}(t) = \boldsymbol{U}^\star + \boldsymbol{u}(t)$, and the value of the control effectors is given by

$$\boldsymbol{\delta} = [G]^\dagger \boldsymbol{U} \tag{6.21}$$

where $[G]^\dagger$ is the pseudoinverse of the ganging matrix $[G]$.

Under the standard $\mathcal{L}_1$ assumption that the fast adaptation causes the adaptive parameters to tend to their true values nearly instantaneously the control law Eq. (6.20) can be shown to be approximately equivalent to a lowpass-filtered nonlinear dynamic inversion control law. Taking the Laplace transform of $\boldsymbol{\eta}(t)$ results in

$$\boldsymbol{\eta}(s) = s\boldsymbol{X}_{\mathrm{des}} - [K_\epsilon]\boldsymbol{\epsilon} - [A_{\mathrm{m}}]\boldsymbol{X} - [\Theta]^{\mathsf{T}}\boldsymbol{\phi} - [\Lambda]\boldsymbol{u}(s)\,,$$

and substituting in Eq. (6.20) results in

$$\boldsymbol{u}(s) = [K][D(s)]\left[\dot{\boldsymbol{X}}_{\mathrm{des}} - [K_\epsilon]\boldsymbol{\epsilon} - [A_{\mathrm{m}}]\boldsymbol{X} - [\Theta]^{\mathsf{T}}\boldsymbol{\phi} - [\Lambda]\boldsymbol{u}(t)\right]$$

where the notation $\boldsymbol{y}(s) = [G(s)][\boldsymbol{u}(t)]$ is used to indicate that the transfer function $[G(s)]$ operates on the time-domain signal $\boldsymbol{u}(t)$. Solving for the pseudocontrol signal, obtain

$$([I_n] + [K][D(s)][\Lambda])\, \boldsymbol{u}(s) = [K][D(s)] \left[\dot{\boldsymbol{X}}_{\text{des}} - [K_\epsilon]\boldsymbol{\epsilon} - [A_{\text{m}}]\boldsymbol{X} - [\hat{\Theta}]^{\mathsf{T}}\boldsymbol{\phi}\right] .$$

Since $[\Lambda]$ is assumed constant, introduce a multiplying factor $[\Lambda][\Lambda]^{-1}$ to allow writing the control law in an NDI-like form:

$$([I_n] + [K][D(s)][\Lambda])\, \boldsymbol{u}(s) = [K][D(s)][\Lambda] \left[[\Lambda]^{-1} \left(\dot{\boldsymbol{X}}_{\text{des}} - [K_\epsilon]\boldsymbol{\epsilon} - [A_{\text{m}}]\boldsymbol{X} - [\Theta]^{\mathsf{T}}\boldsymbol{\phi}\right)\right] , \tag{6.22}$$

and, finally,

$$\boldsymbol{u}(s) = [C(s)] \left[[\Lambda]^{-1} \left(\dot{\boldsymbol{X}}_{\text{des}} - [K_\epsilon]\boldsymbol{\epsilon} - [A_{\text{m}}]\boldsymbol{X} - [\Theta]^{\mathsf{T}}\boldsymbol{\phi}\right)\right] \tag{6.23}$$

where

$$[C(s)] = ([I_n] + [K][D(s)][\Lambda])^{-1} [K][D(s)][\Lambda] . \tag{6.24}$$

Table 6.2 presents a summary of this $\mathcal{L}_1$ adaptive NDI control law.

6.3.4 Stability Analysis

The stability analysis of the $\mathcal{L}_1$ adaptive dynamic inversion control law proceeds much like Section 5.4. The adaptive laws guarantee that the state prediction error is bounded, so it remains to show the state predictor is bounded for the true states to also be bounded. Using the standard fast adaptation assumption, consider the reference system

$$\dot{\boldsymbol{X}}_{\text{ref}} = [A_{\text{m}}]\boldsymbol{X}_{\text{ref}} + [\Theta]^{\mathsf{T}}\boldsymbol{\phi} + [\Lambda]\boldsymbol{u}_{\text{ref}} \tag{6.25}$$

Table 6.2: $\mathcal{L}_1$ adaptive nonlinear dynamic inversion control law.

Model	$\dot{\boldsymbol{X}} = [A_\mathrm{m}]\boldsymbol{X} + [\Theta]^\mathsf{T}\boldsymbol{\phi} + [\Lambda]\boldsymbol{u}$
State Predictor	$\dot{\hat{\boldsymbol{X}}} = [A_\mathrm{m}]\hat{\boldsymbol{X}} + [\hat{\Theta}]^\mathsf{T}\boldsymbol{\phi} + [\hat{\Lambda}]\boldsymbol{u}$
Error Dynamics	$\boldsymbol{E} = \hat{\boldsymbol{X}} - \boldsymbol{X}$ $\dot{\boldsymbol{E}} = [A_\mathrm{m}]\boldsymbol{E} + [\tilde{\Theta}]^\mathsf{T}\boldsymbol{\phi} + [\tilde{\Lambda}]\boldsymbol{u}$
Adaptive Laws	$[\dot{\hat{\Theta}}] = \mathrm{Proj}\left([\hat{\Theta}], -[\Gamma_\theta]\boldsymbol{\phi}\mu(\\|\boldsymbol{E}\\|)\boldsymbol{E}^\mathsf{T}[P]\right)$ $[\dot{\hat{\Lambda}}] = \mathrm{Proj}\left([\hat{\Lambda}], -[\Gamma_\Lambda][P]\mu(\\|\boldsymbol{E}\\|)\boldsymbol{E}\boldsymbol{u}^\mathsf{T}\right)$
Control Law	$\boldsymbol{\epsilon} = \boldsymbol{X} - \boldsymbol{X}_\mathrm{des}$ $\boldsymbol{u}(s) = [K][D(s)]\boldsymbol{\eta}(s)$ $\boldsymbol{\eta}(t) = \dot{\boldsymbol{X}}_\mathrm{des} - [K_\epsilon]\boldsymbol{\epsilon} - [A_\mathrm{m}]\boldsymbol{X} - [\hat{\Theta}]^\mathsf{T}\boldsymbol{\phi} - [\hat{\Lambda}]\boldsymbol{u}$
Control Allocation	$\boldsymbol{\delta} = \boldsymbol{\delta}^\star + [B]^\dagger\boldsymbol{U}$

with Laplace transform

$$(s[I_n] - [A_\mathrm{m}])\,\boldsymbol{X}_\mathrm{ref}(s) = [\Theta]^\mathsf{T}\boldsymbol{\phi} + [\Lambda]\boldsymbol{u}_\mathrm{ref}(s) + (s[I_n] - [A_\mathrm{m}])^{-1}\,\boldsymbol{X}(0)\,. \tag{6.26}$$

Define $[H(s)] \triangleq (s[I_n] - [A_\mathrm{m}])^{-1}$ and $\boldsymbol{X}_\mathrm{in}(s) = H(s)\boldsymbol{X}(0)$ to obtain

$$\boldsymbol{X}_\mathrm{ref}(s) = [H(s)][\Theta]^\mathsf{T}\boldsymbol{\phi} + [H(s)][\Lambda]\boldsymbol{u}_\mathrm{ref}(s) + \boldsymbol{X}_\mathrm{in}(s)\,. \tag{6.27}$$

The reference control law can be shown to be Eq. (6.23). Substituting this results in

$$\begin{aligned}\boldsymbol{X}_\mathrm{ref}(s) = [H(s)][\Theta]^\mathsf{T}\boldsymbol{\phi} + [H(s)][\Lambda][C(s)]\Bigg[[\Lambda]^{-1}\Bigg(&\dot{\boldsymbol{X}}_\mathrm{des} - [K_\epsilon](\boldsymbol{X}_\mathrm{ref} - \boldsymbol{X}_\mathrm{des})\\ &- [A_\mathrm{m}]\boldsymbol{X}_\mathrm{ref}(s) - [\Theta]^\mathsf{T}\boldsymbol{\phi}\Bigg)\Bigg] + \boldsymbol{X}_\mathrm{in}(s)\,.\end{aligned}$$

This can further be expanded as

$$\begin{aligned}([I_n]+[H(s)][C_1(s)]\left([A_\mathrm{m}]+[K_\epsilon]\right))\boldsymbol{X}_\mathrm{ref}(s) = [G(s)][\Theta]^\mathsf{T}\boldsymbol{\phi} \\ +[H(s)][C_1(s)]\left[\dot{\boldsymbol{X}}_\mathrm{des}+[K_\epsilon]\boldsymbol{X}_\mathrm{des}\right]+\boldsymbol{X}_\mathrm{in}(s) \quad (6.28)\end{aligned}$$

where $[C_1(s)] \triangleq [\Lambda][C(s)][\Lambda]^{-1}$ and $[G(s)] \triangleq [H(s)]([I_n]-[C_1(s)])$. To simplify manipulations it is useful to define some quantities: $[H_1(s)] \triangleq ([I_n]+[H(s)][C_1(s)]([A_\mathrm{m}]+[K_\epsilon]))$, $[G_1(s)] \triangleq [H_1(s)]^{-1}[G(s)]$, $[H_2(s)] \triangleq [H_1(s)]^{-1}[H(s)]$, and $[H_3(s)] \triangleq [H_1(s)]^{-1}$. This allows rewriting Eq. (6.28) as

$$\boldsymbol{X}_\mathrm{ref}(s) = [G_1(s)][\Theta]^\mathsf{T}\boldsymbol{\phi}+[H_2(s)][C_1(s)]\left[\dot{\boldsymbol{X}}_\mathrm{des}+[K_\epsilon]\boldsymbol{X}_\mathrm{des}\right]+[H_3(s)]\boldsymbol{X}_\mathrm{in}(s)\,. \quad (6.29)$$

The norm of Eq. (6.29) is

$$\begin{aligned}\|\boldsymbol{X}_\mathrm{ref,t}\|_\infty \leqslant \|[G_1(s)]\|_1\,\|[\Theta]\|_1\,\|\boldsymbol{\phi}_\mathrm{t}\|_\infty + \|[H_2(s)][C_1(s)]\|_1\,\|\dot{\boldsymbol{X}}_\mathrm{des,t}\|_\infty \\ + \|[H_2(s)][C_1(s)][K_\epsilon]\|_1\,\|\boldsymbol{X}_\mathrm{des,t}\|_\infty + \|[H_3(s)]\|_1\,\|\boldsymbol{X}_\mathrm{in,t}(s)\|_\infty\,. \quad (6.30)\end{aligned}$$

Recalling the use of Lemma A.8.1 of Ref. [69] in Section 5, assume that $\boldsymbol{\phi}(\boldsymbol{x})$ may be parameterized as

$$\boldsymbol{\phi}(\boldsymbol{x}) = \boldsymbol{\vartheta}(t)\,\|\boldsymbol{X}\|_\infty + \boldsymbol{\sigma}(t)$$

where $\boldsymbol{\vartheta}(t)$ and $\boldsymbol{\sigma}(t)$ are bounded signals that allow the nonlinear function $\boldsymbol{\phi}$ to be approximated as linear time-varying. It follows that $\|\boldsymbol{\phi}_\mathrm{t}\|_\infty \leqslant \|\boldsymbol{\vartheta}\|_1\|\boldsymbol{X}_\mathrm{t}\|_\infty + \|\boldsymbol{\sigma}_\mathrm{t}\|_\infty$, and thus

Eq. (6.30) becomes

$$\begin{aligned}\|\boldsymbol{X}_{\text{ref,t}}\|_{\infty} \leqslant \|[G_1(s)]\|_1 \, \|[\Theta]\|_1 \, \|\boldsymbol{\vartheta}\|_1 \, \|\boldsymbol{X}_{\text{ref,t}}\|_{\infty} + \|[G_1(s)]\|_1 \, \|[\Theta]\|_1 \, \|\boldsymbol{\sigma}_{\text{t}}\|_{\infty} \\ + \|[H_2(s)][C_1(s)]\|_1 \, \|\dot{\boldsymbol{X}}_{\text{des,t}}\|_{\infty} + \|[H_2(s)][C_1(s)][K_\epsilon]\|_1 \, \|\boldsymbol{X}_{\text{des,t}}\|_{\infty} \\ + \|[H_3(s)]\|_1 \, \|\boldsymbol{X}_{\text{in,t}}(s)\|_{\infty} \, .\end{aligned}$$

Defining $L_1 \triangleq (\max \|[\Theta]\|_1) \, \|\boldsymbol{\vartheta}\|_1$ and $L_2 \triangleq (\max \|[\Theta]\|_1) \, \|\boldsymbol{\sigma}_{\text{t}}\|_{\infty}$, the norm of the reference system state can be written as

$$\|\boldsymbol{X}_{\text{ref,t}}\|_{\infty} \leqslant \frac{N}{1 - \|[G_1(s)]\|_1 \, L_1} \tag{6.31}$$

where

$$\begin{aligned}N = \|[H_2(s)][C_1(s)]\|_1 \, \|\dot{\boldsymbol{X}}_{\text{des,t}}\|_{\infty} + \|[H_2(s)][C_1(s)][K_\epsilon]\|_1 \, \|\boldsymbol{X}_{\text{des,t}}\|_{\infty} \\ + \|[G_1(s)]\|_1 \, L_2 + \|[H_3(s)]\|_1 \, \|\boldsymbol{X}_{\text{in,t}}(s)\|_{\infty} \, .\end{aligned}$$

The desired dynamics are assumed bounded and $[H(s)]$ is Hurwitz which makes $\boldsymbol{X}_{\text{in}}(s)$ bounded. As a result, Eq. (6.31) shows that $\boldsymbol{X}_{\text{ref}}$ is bounded provided that

$$\|[G_1(s)]\|_1 < \frac{1}{L_1} \, .$$

6.3.5 Modified Version

The $\mathcal{L}_1$ adaptive nonlinear dynamic inversion controller above can be modified to adapt to the error between the true dynamics and an onboard model with a slight modification. Consider the dynamics Eq. (6.14) and state predictor Eq. (6.15). Assume that $\boldsymbol{G}(\boldsymbol{X}, \boldsymbol{U}^{\star})$

can be parameterized instead as

$$\boldsymbol{G}(\boldsymbol{X},\boldsymbol{U}^{\star}) = \hat{\boldsymbol{F}}(\boldsymbol{X},\boldsymbol{U}^{\star}) + [\Theta]^{\mathsf{T}}\boldsymbol{\phi}(\boldsymbol{X}) - [A_{\mathrm{m}}]\boldsymbol{X}$$

where the true open-loop dynamics are now $\boldsymbol{F}(\boldsymbol{X},\boldsymbol{U}^{\star}) = \hat{\boldsymbol{F}}(\boldsymbol{X},\boldsymbol{U}^{\star}) + [\Theta]^{\mathsf{T}}\boldsymbol{\phi}(\boldsymbol{X})$. That is, the true open-loop dynamics can be represented as the known onboard model $\hat{\boldsymbol{F}}(\boldsymbol{X},\boldsymbol{U}^{\star})$ plus an unknown modeling error term that consists of an unknown constant parameter matrix $[\Theta]$ multiplied by known nonlinear basis functions $\boldsymbol{\phi}(\boldsymbol{X})$. The estimate $\hat{\boldsymbol{G}}(\boldsymbol{X},\boldsymbol{U}^{\star})$ is then

$$\hat{\boldsymbol{G}}(\boldsymbol{X},\boldsymbol{U}^{\star}) = \hat{\boldsymbol{F}}(\boldsymbol{X},\boldsymbol{U}^{\star}) + [\hat{\Theta}]^{\mathsf{T}}\boldsymbol{\phi}(\boldsymbol{X}) - [A_{\mathrm{m}}]\boldsymbol{X}\,,$$

which leads to the same error dynamics Eq. (6.16) as in the previous formulation, with the exception that the term $[\tilde{\Theta}]^{\mathsf{T}}\boldsymbol{\phi}$ now represents the open-loop modeling error instead of the estimate of the open-loop dynamics. The adaptive laws take the same form, and the control law becomes

$$\boldsymbol{\eta}(t) = \dot{\boldsymbol{X}}_{\mathrm{des}} - [K_{\epsilon}]\boldsymbol{\epsilon} - (\hat{\boldsymbol{F}}(\boldsymbol{X},\boldsymbol{U}^{\star}) + [\hat{\Theta}]^{\mathsf{T}}\boldsymbol{\phi}) - [\hat{\Lambda}]\boldsymbol{u}\,, \tag{6.32a}$$

$$\boldsymbol{u}(s) = [K][D(s)][\boldsymbol{\eta}(s)]\,, \tag{6.32b}$$

Note that $\hat{\boldsymbol{F}}(\boldsymbol{X},\boldsymbol{U}^{\star})$ is not an adaptive estimate but is the result of the modeling process as described in Section 2. The adaptive controller Eq. (6.32) is a more direct $\mathcal{L}_1$ analogue of the controller Eq. (6.4).

6.4 Simulation Results

This section presents simulation results for the MRAC and $\mathcal{L}_1$ adaptive dynamic inversion controllers presented in this section. Both controllers are used to implement the CAS design presented at the end of Section 3. This section first presents “ideal” results without noise or disturbances, and then presents results in a realistic environment.

6.4.1 MRAC Dynamic Inversion CAS

The MRAC NDI control law is tuned with the parameters $[K]$, $[\Gamma_\Theta]$, $[\Gamma_\Lambda]$, and $[Q]$. The tracking error feedback gain $[K]$ is designed using an LQR approach with the system $\dot{\boldsymbol{e}} = \boldsymbol{\nu}$, similar to the approach used for the “robust NDI” controller presented in Section 4. To avoid confusion with the Lyapunov matrix $[Q]$, the LQR design parameters are denoted with $[\bar{Q}]$ and $[\bar{R}]$. Table 6.3 lists the design parameters for the MRAC DI controller. The adaptive parameters $\hat{\Theta}_{ij}$ are limited to ± 5, and the adaptive parameters $\hat{\Lambda}_{ij}$ are limited to ± 0.95 from their initial values. The initial conditions are $[\hat{\theta}(0)] = [0_{16\times 3}]$ and $[\hat{\Lambda}(0)] = [I_3]$. The threshold for the dead zone modification is $e_0 = 1\,°/\text{s}$.

Table 6.3: MRAC DI controller design parameters.

Design Parameter	**Value**
$[Q]$	$15[I_3]$
$[\bar{Q}]$	$\text{diag}(2500, 10000, 500)$
$[\bar{R}]$	$\text{diag}(1, 0.1, 10)$
$[K]$	$\text{diag}(-50, -100, -7.0711)$
$[\Gamma_\Lambda]$	$30[I_3]$
$[\Gamma_\Theta]$	$30[I_{12}]$

The MRAC NDI control law uses the GNA model Eqs. (2.49–2.51) with the F-16C parameters found in Ref. [43] as the assumed onboard model. The adaptive controller uses the following vector of basis functions to correct for errors in this model:

$$\boldsymbol{\phi}(\boldsymbol{X}) = [1, \alpha, \beta, \hat{p}, \hat{q}, \hat{r}, \alpha\hat{q}, \beta^2, \alpha^2\hat{q}, \beta^3, \alpha^3\hat{q}, \alpha^4]^\mathsf{T} \tag{6.33}$$

where following previous convention in this thesis an over-hat indicates a non-dimensional angular rate. Feedback of nonlinear terms from the Euler equations Eq. (2.17) were considered but were found to result in degraded performance when the inertial terms were well-known.

6.4.1.1 Baseline Results

These commands are the same as used in the previous simulation examples featured in the thesis. The resulting desired dynamics and the achieved angular rates are shown in Fig. 6.1a. The control law achieves excellent tracking in all three axes, albeit with initial transients in the pitch axis that are difficult to observe in the figure. The tracking error plots show a tight pitch axis response with errors less than $0.31\,°/\mathrm{s}$. The control law tracks the desired yaw rate with errors less than $0.33\,°/\mathrm{s}$, while the desired roll rate is achieved with less than $2\,°/\mathrm{s}$ error. This roll axis response is an order of magnitude improvement over the robust NDI method used in Section 4.5.2. The figure additionally plots the airspeed and incidence angle responses for the MRAC NDI controller, which are comparable to previous controllers.

The commanded and achieved control effector deflections are plotting in Fig. 6.1b. The overall control effort is similar to that of the other controllers. Initially a high frequency demand is made on the elevator; the elevator remains within the rate limits for the entire simulation. All control effectors meet position and rate limits.

Finally, Fig. 6.2 plots the time histories of the adaptive parameter estimates. The parameters $\hat{\Theta}_{i1}$ correspond to the roll axis OBM error correction terms, while the $\hat{\Theta}_{i2}$ parameters correspond to the pitch axis OBM error and the parameters $\hat{\Theta}_{i3}$ correspond to the yaw axis OBM error. Note that Fig. 6.2 plots the *incremental inverse* of the control effectiveness estimate $[\hat{\Lambda}]$, defined as

$$[\delta\hat{\Lambda}]^{-1} \triangleq [\hat{\Lambda}]^{-1} - [I_3]\,.$$

The inverse is plotted as it is used in the control law in the term $[\hat{\Lambda}]^{-1}[B]^{-1}$. Note that the estimate is diagonally dominant with smaller off-diagonal terms that adapt for cross-axis control effectiveness.

6.4.1.2 Disturbance Results

This section presents simulation results for the MRAC NDI CAS in the presence of atmospheric disturbances. At the flight condition (M0.72 @ 20k ft) three sources of disturbances are modeled: horizontal winds, discrete gusts, and Dryden turbulence. These models are discussed in App. C. Horizontal winds are constant at $30\,\mathrm{ft/s}$ from $50°$ clockwise from the north. Discrete gusts begin five seconds into the simulation, with gusts lengths of $120\,\mathrm{ft}$, $120\,\mathrm{ft}$, and $80\,\mathrm{ft}$ in the $\hat{\boldsymbol{x}}_\mathrm{n}$, $\hat{\boldsymbol{y}}_\mathrm{n}$, and $\hat{\boldsymbol{z}}_\mathrm{n}$ directions, respectively. The corresponding gust amplitudes are $7\,\mathrm{ft/s}$, $7\,\mathrm{ft/s}$, and $5\,\mathrm{ft/s}$, respectively. Dryden gust turbulence is modeled with a wind speed $u_{20} = 30\,\mathrm{ft/s}$ and direction of $240°$ clockwise from the north. The turbulence intensities are obtained corresponding to a probability of exceedance of 1×10^{-4}, which lies between “moderate” and “severe” turbulence. Initially this simulation was specified to use severe turbulence (1×10^{-5}), however the simulation did not converge indicating failure of the controller to stabilize the system.

Simulation results are shown as state time histories in Fig. 6.3a. The MRAC NDI controller retains good tracking performance in the presence of the disturbances, following the roll, pitch, and yaw rate desired dynamics. The yaw axis performance suffers slightly as seen by the larger tracking errors in yaw rate and the reduction of sideslip regulation performance to within bounds of $2.5°$ as opposed to $1°$. While not part of the adaptive DI controller, note that the autothrottle performance degrades in the presence of the disturbances, with velocity errors on the order of $6\,\mathrm{ft/s}$.

Control effector time histories are shown in Fig. 6.3b. Compared to the disturbance-free case, higher demands are made in terms of both positions and rates. Commanded effector positions continue to satisfy position limits, but fail to meet rate limits.

Adaptive parameter time histories are shown in Fig. 6.4. Compared to the disturbance-free case, Fig. 6.2, the values of the adaptive parameters are larger and show more oscillatory behavior due to the turbulence. Note that despite the disturbances the adaptive parameters remain bounded without needing to resort to projection.

6.4.2 $\mathcal{L}_1$ Adaptive Dynamic Inversion CAS

Parameters for the $\mathcal{L}_1$ design are presented in Table 6.4. The vector of basis functions $\boldsymbol{\phi}(\boldsymbol{X})$ is defined as

$$\boldsymbol{\phi}(\boldsymbol{X}) = \bar{q}S[1, \alpha, \beta, \hat{p}, \hat{q}, \hat{r}, \alpha\hat{q}, \beta^2, \alpha^2\hat{q}, \beta^3, \alpha^3\hat{q}, \alpha^4]^{\mathsf{T}} . \tag{6.34}$$

Equation (6.34) differs from the MRAC DI controller basis functions, Eq. (6.33), by a factor of $\bar{q}S$ which effectively acts as a “scheduling” term based on the flight condition. The $\bar{q}S$ factor was found to result in better adaptive estimates with the high-gain $\mathcal{L}_1$ adaptive laws. The corresponding adaptive parameter matrix is initialized to zero, indicating no *a priori* knowledge of the dynamics, while the control effectiveness estimate $[\hat{\Lambda}]$ is initialized to identity.

6.4.2.1 Baseline Results

Figure 6.5a plots the true and desired attitude rates. Although not shown, the predicted states converge almost instantaneously on the true values. The desired attitude rates result from integrating the desired dynamics. The figure shows good tracking in all three axes. Pitch rate and yaw rate track very well, with absolute errors less than $1\,^\circ/\mathrm{s}$, while roll

Table 6.4: $\mathcal{L}_1$-AC controller design parameters.

Design Parameter	**Value**
$[A_\mathrm{m}]$	$-14[I_3]$
$[Q]$	$15[I_3]$
$[K]$	30
$[K_\epsilon]$	$\mathrm{diag}(5, 5, 1)$
$D(s)$	$1/s$
$[\Gamma_\Lambda]$	$1 \times 10^4[I_3]$
$[\Gamma_\Theta]$	$1 \times 10^4[I_{16}]$

rate has a maximum tracking error of approximately $20\,^\circ/\mathrm{s}$. However, since this error occurs at a command of approximately $180\,^\circ/\mathrm{s}$ it is deemed acceptable. The airspeed is held approximately constant during the diving maneuver. The angle-of-attack is not directly controlled, and it is simply desired that it stays at a "reasonable" value during the maneuver. The sideslip angle control objective is regulation to maintain coordinated flight. Observing the plot, the sideslip is kept to within 1° which is deemed acceptable.

The demanded actuator positions and the actual positions are plotted in Fig. 6.5b. The commanded actuator positions are well within limits, and the actuators do a good job of meeting the demands. Some attenuation of higher frequency effects is seen in the plot; the $\mathcal{L}_1$ gain parameter $[K]$ can be tuned to control the filter bandwidth to reduce high frequency demands on the actuators.

Finally, Fig. 6.6 plots the time histories of the adaptive parameters $[\hat{\Theta}]$ and $[\hat{\Lambda}]$. Note that the figure plots the deviations of the parameter estimates $[\hat{\Lambda}]$ from identity such that all elements are of a similar order of magnitude. All of the adaptive parameters are relatively small in magnitude since the elements of the basis function vector $\boldsymbol{\phi}(\boldsymbol{X})$ are large.

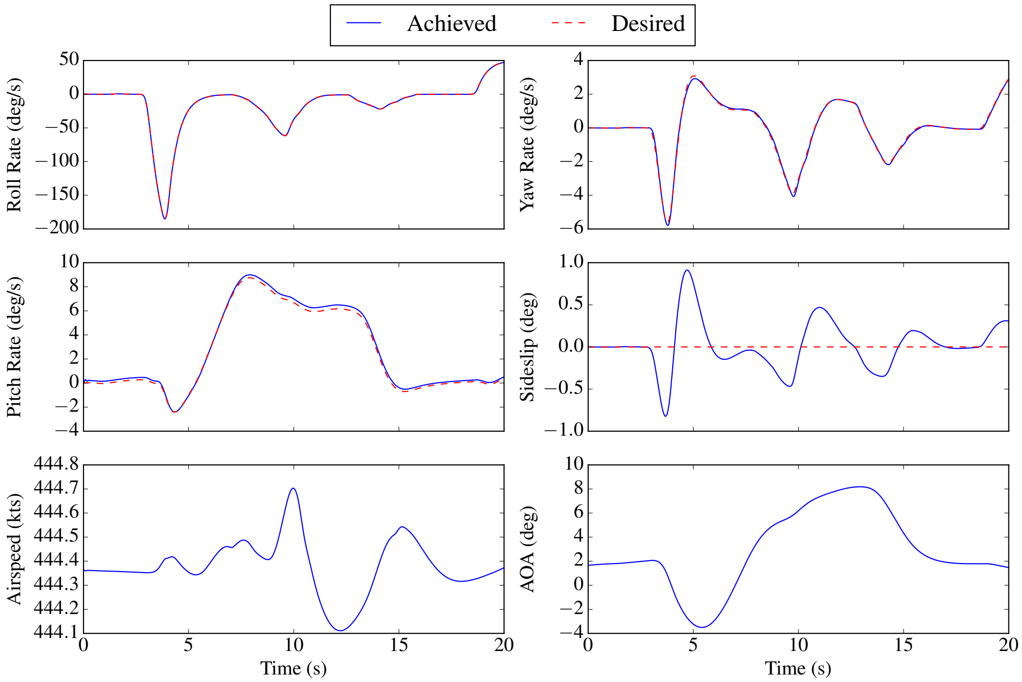
6.4.2.2 Disturbance Results

Results for the $\mathcal{L}_1$ adaptive dynamic inversion controller in windy and turbulent conditions are presented. The simulation is set up with the same constant wind, discrete gusts, and Dryden gusts as the MRAC NDI controller result in Section 6.4.1.2, with the exception that the turbulence intensity is set to the severe level corresponding to a probability of exceedance of 1×10^{-5}.

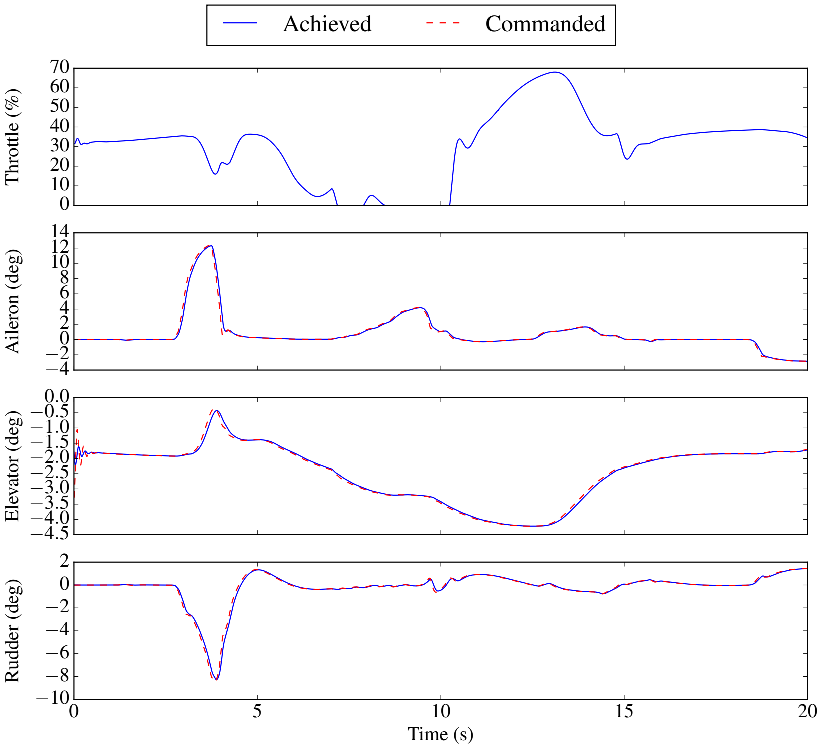
Figure 6.7a presents the state time histories, plotting the angular rates, airspeed, and incidence angles. Good tracking of the desired roll and pitch rate dynamics is observed, although the severe turbulence makes the signals oscillatory around the desired values. As with the MRAC NDI approach the yaw axis performance suffers. In this case the severe turbulence decreases the sideslip regulation performance to limiting sideslip angle to approximately $3°$. Compared to the disturbance-free case, the maximum AOA observed increases to $10°$. Overall, the tracking performance of the $\mathcal{L}_1$ controller is maintained with significant atmospheric disturbances.

Control effector time histories are provided in Fig. 6.7b. In the presence of disturbances higher demands on positions and rates are required to meet the control objective. The $\mathcal{L}_1$ controller makes similar demands on the effectors as the MRAC NDI controller; it makes slightly less demand on elevator and slightly more demand on rudder.

The adaptive parameters are plotted in Fig. 6.8. Compared to the nonadaptive case, the adaptive parameters exhibit more oscillatory behavior, a result of the severe turbulence. The magnitudes of the parameters remain similar to the case without turbulence, unlike with the MRAC NDI control law.



(a) Actual and desired state time histories.



(b) Aerodynamic control effector time histories.

Figure 6.1: MRAC NDI controller state and control time histories.

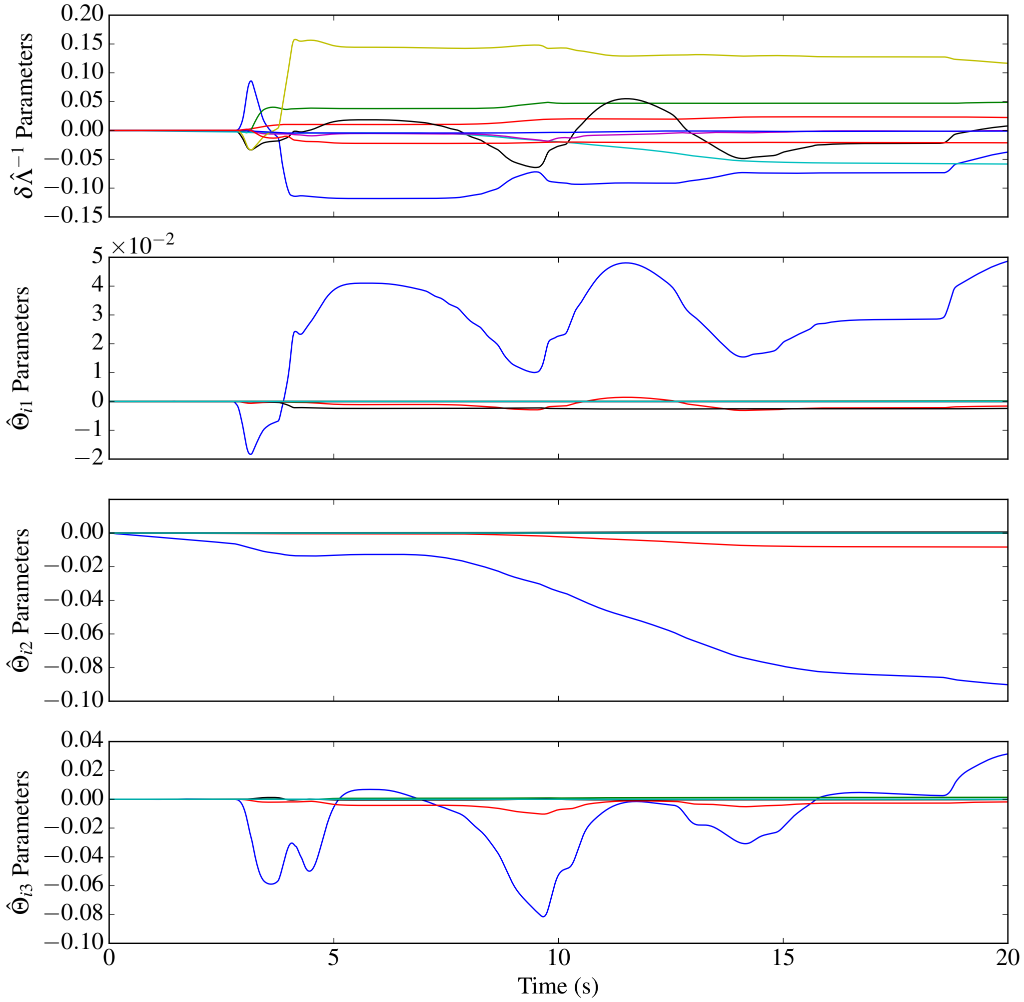
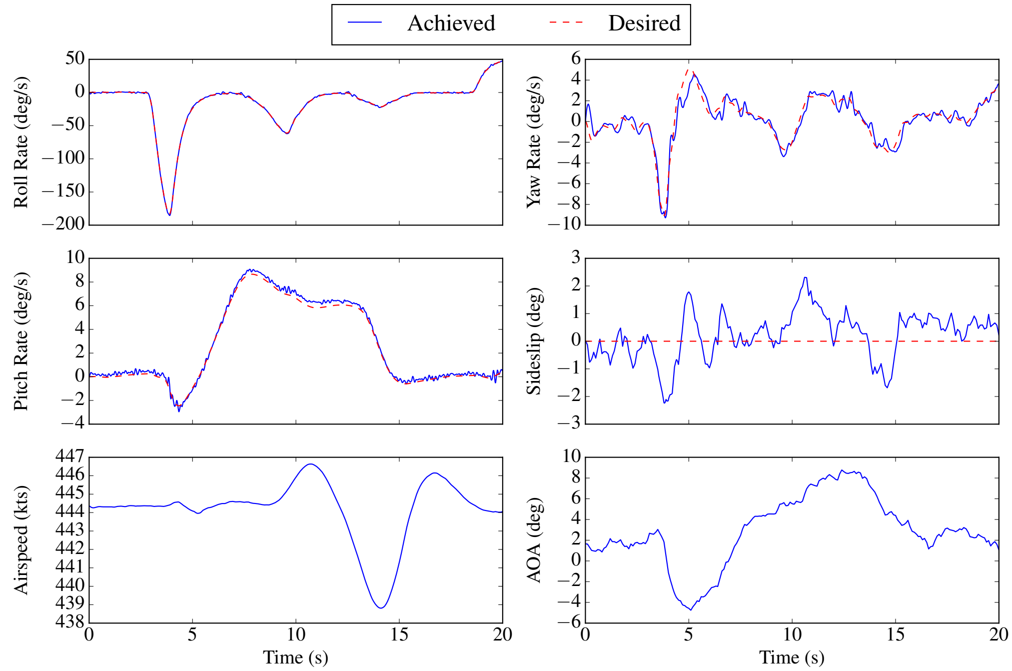
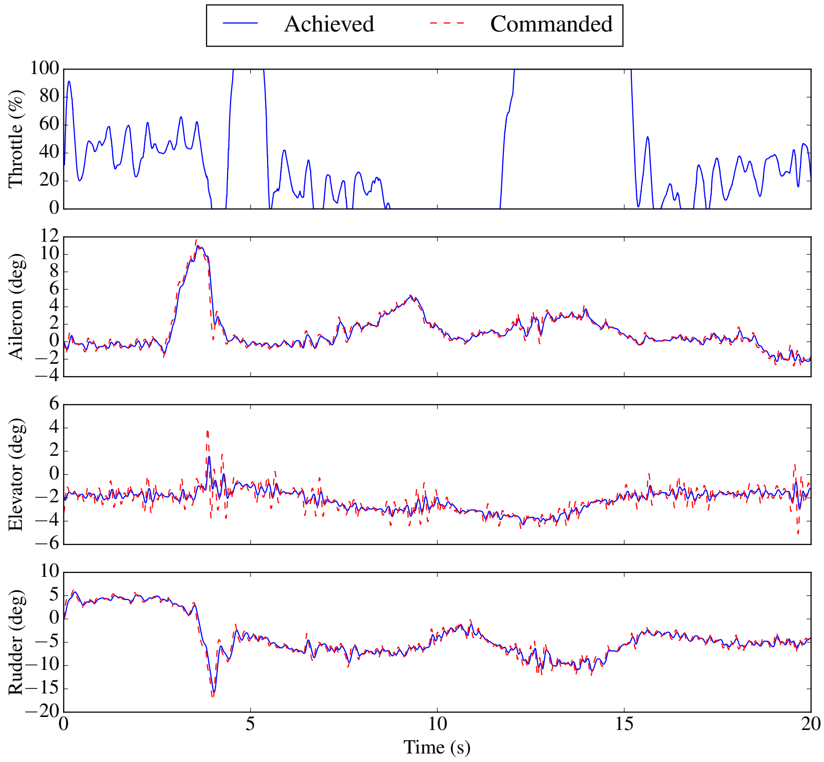


Figure 6.2: MRAC NDI adaptive parameter time histories.



(a) Actual and desired state time histories.



(b) Aerodynamic control effector time histories.

Figure 6.3: MRAC NDI controller state and control time histories with disturbances.

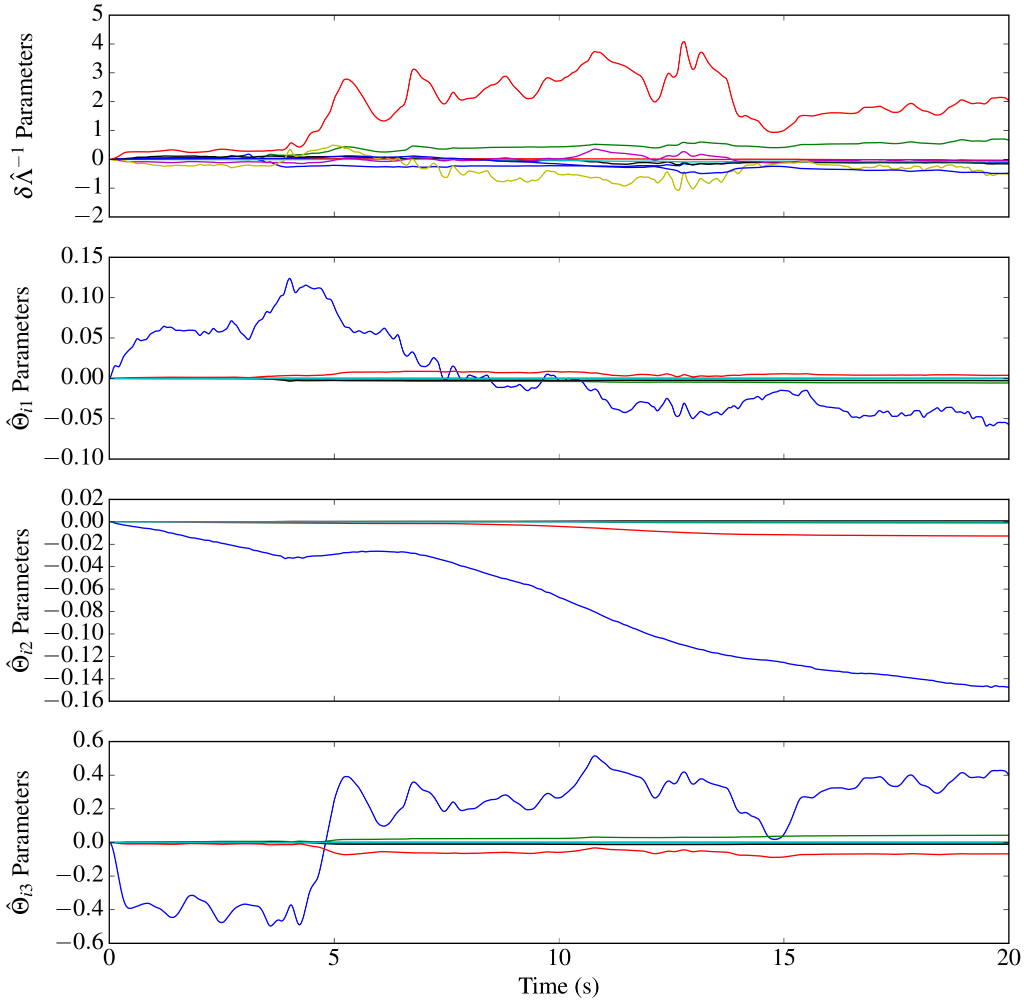
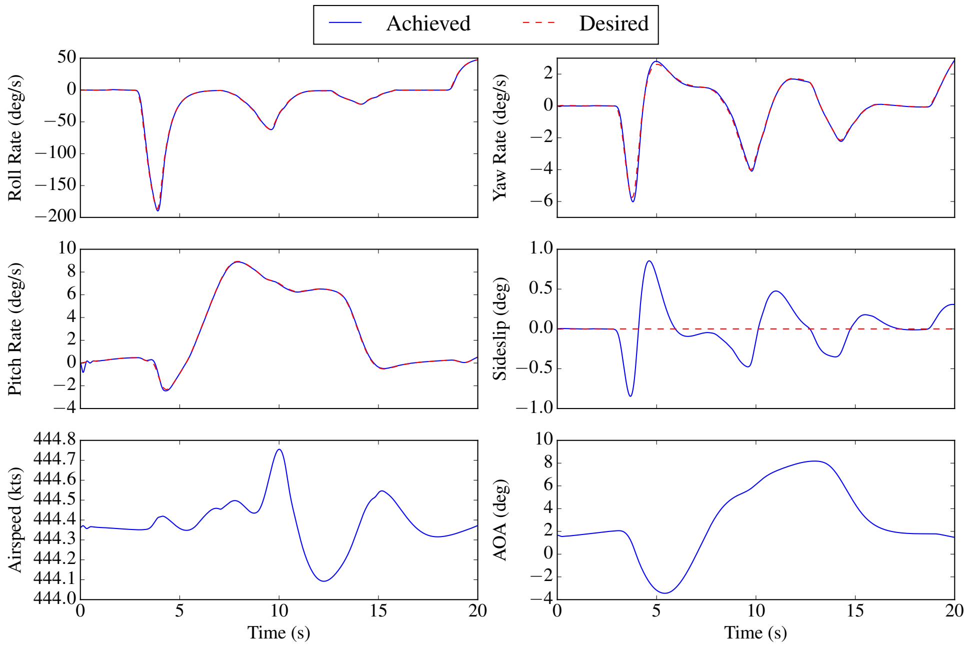
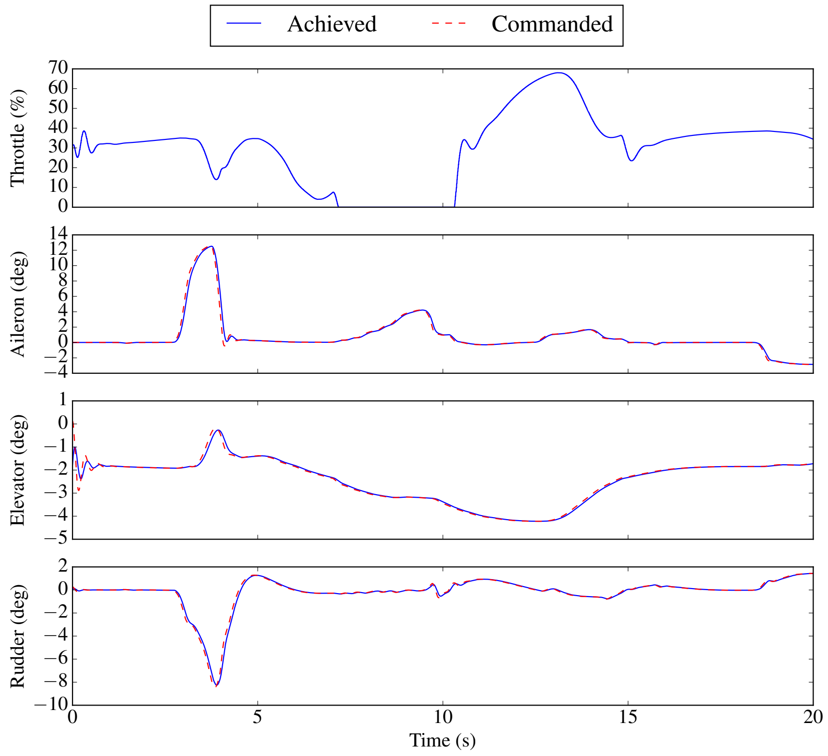


Figure 6.4: MRAC NDI adaptive parameter time histories with disturbances.



(a) Actual and desired state time histories.



(b) Aerodynamic control effector time histories.

Figure 6.5: $\mathcal{L}_1$ Adaptive NDI controller state and control time histories.

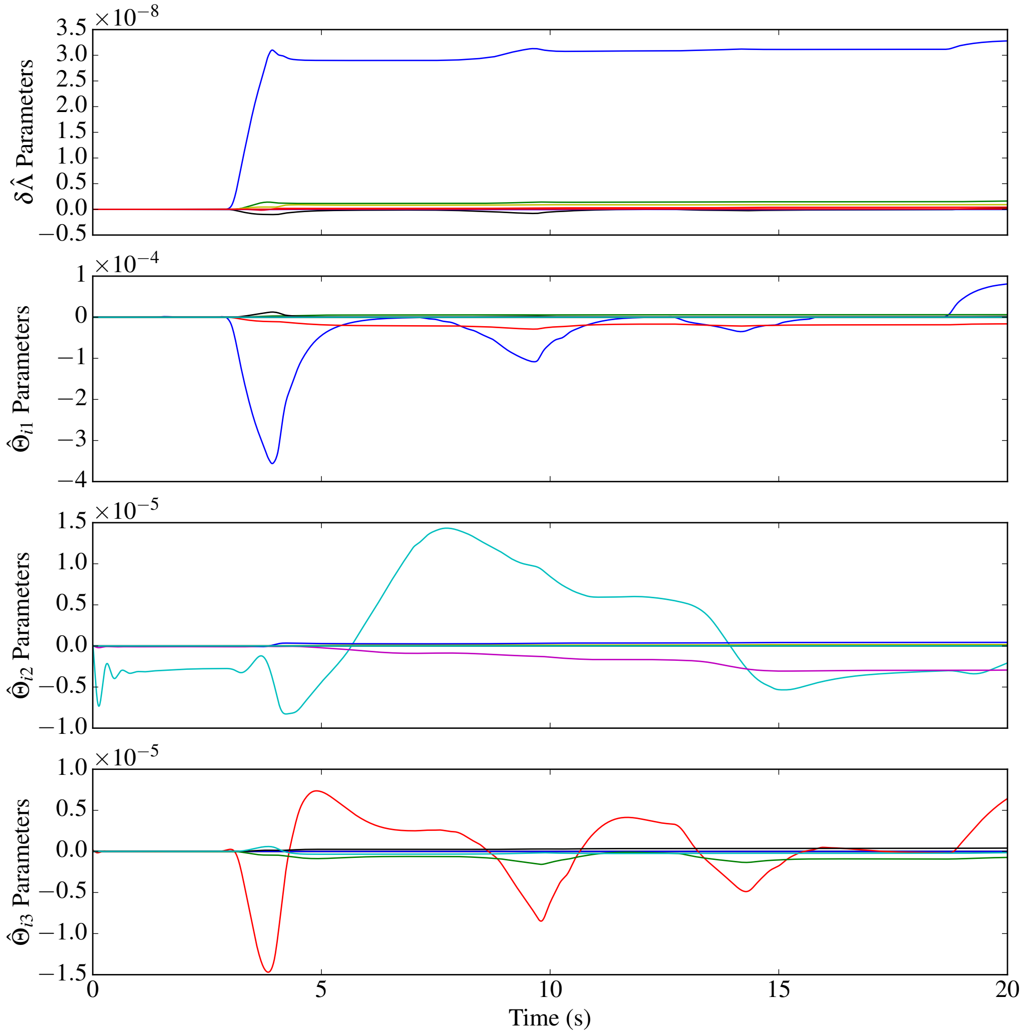
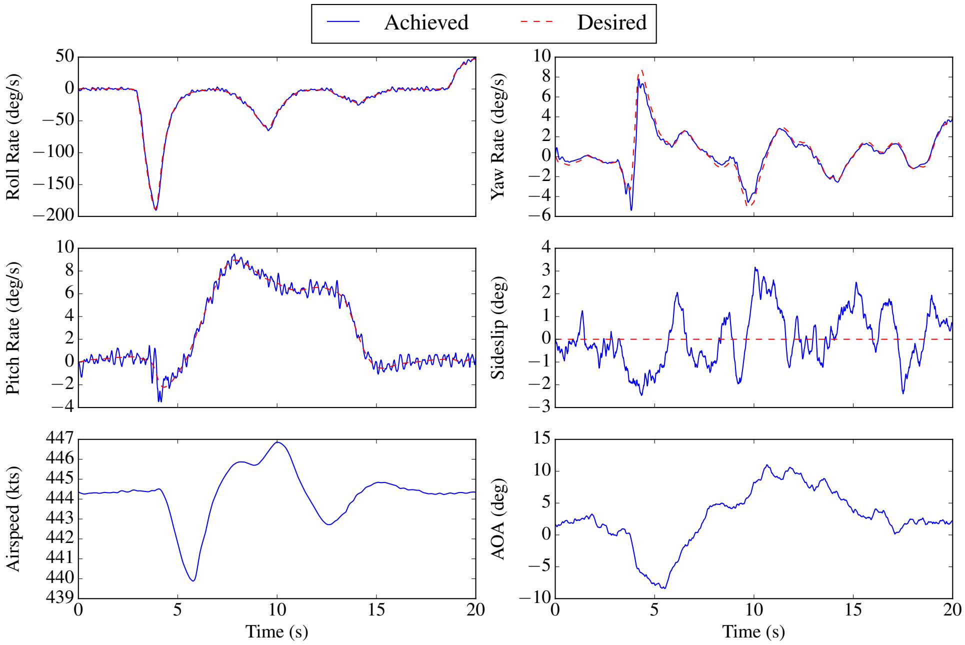
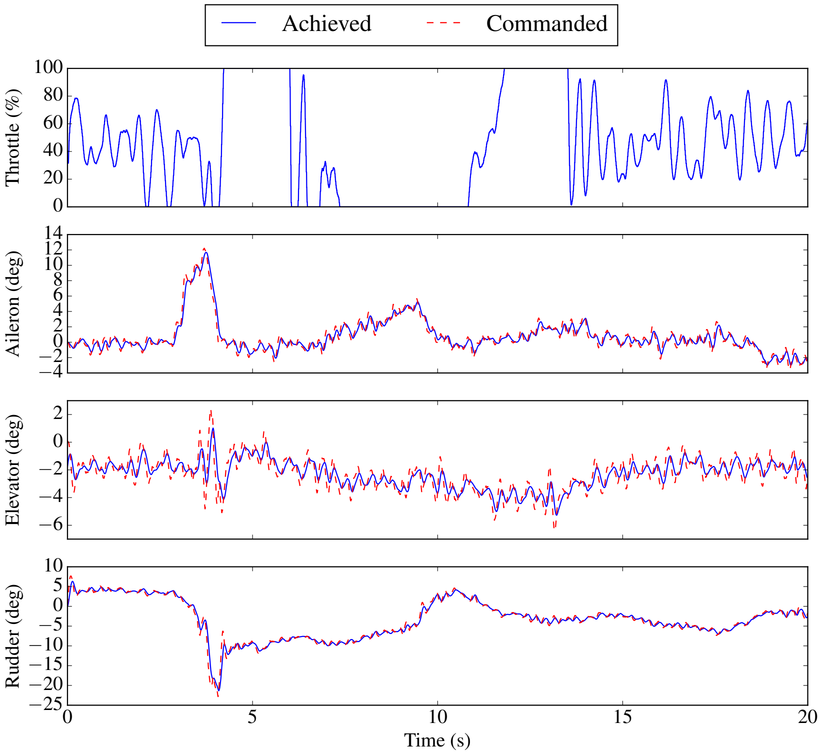


Figure 6.6: $\mathcal{L}_1$ Adaptive NDI adaptive parameter time histories.



(a) Actual and desired state time histories.



(b) Aerodynamic control effector time histories.

Figure 6.7: $\mathcal{L}_1$ Adaptive NDI controller state and control time histories with disturbances.

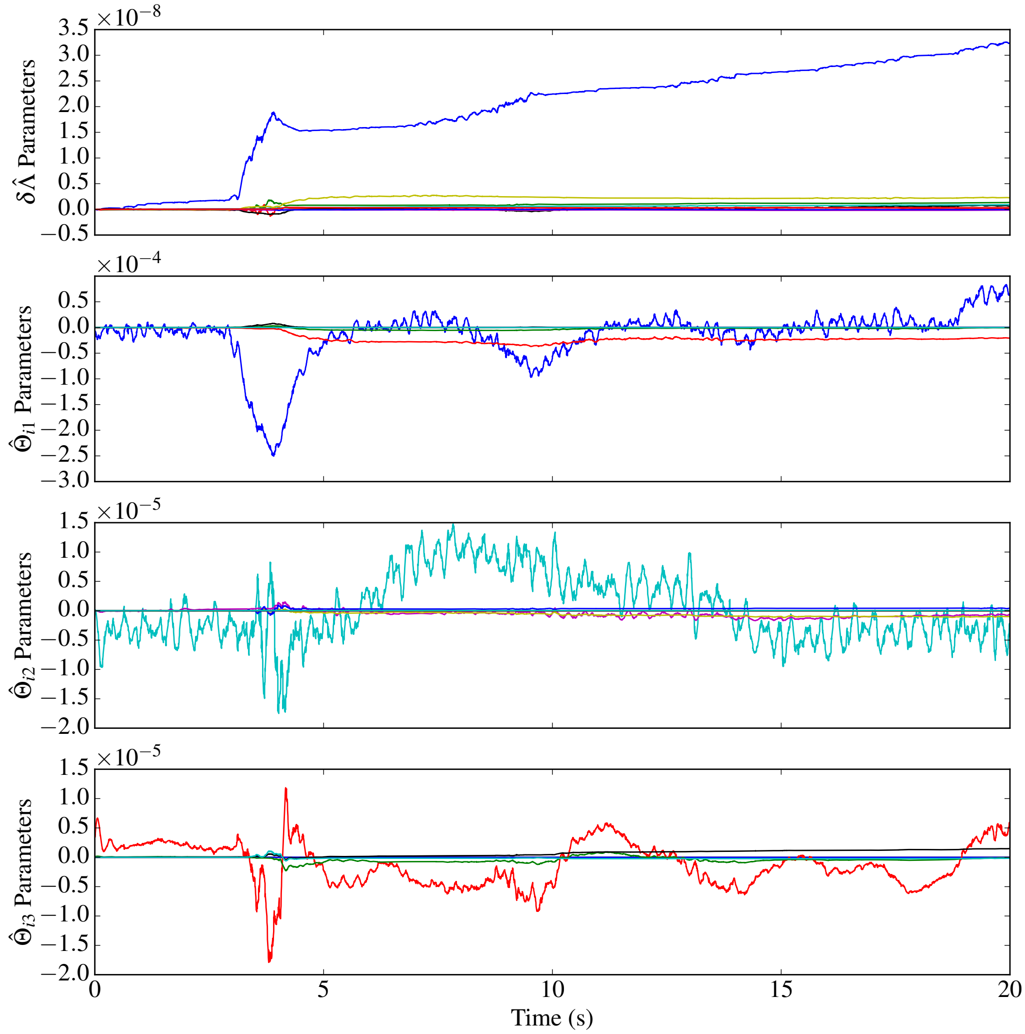


Figure 6.8: $\mathcal{L}_1$ Adaptive NDI adaptive parameter time histories with disturbances.

7. APPROACHES TO IN-FLIGHT SIMULATION AND VARIABLE STABILITY

7.1 Introduction

Variable stability (VS) and in-flight simulator (IFS) aircraft are able to change their dynamical characteristics in flight. In this thesis, a somewhat artificial distinction between VS and IFS capabilities is introduced as follows: VS aircraft can tailor their dynamics such as modifying levels of damping, while IFS aircraft actively try to replicate the dynamics of a different aircraft. Traditional VS and IFS aircraft achieved these capabilities through a response feedback (RFB) method, which uses the mechanism of feedback to modify the aircraft's inherent stability derivatives, or alternatively, to change the aircraft's inherent poles [7, 8]. The RFB technique is similar to the concept of equivalent stability derivatives, where a stability augmentation system (SAS) feedback gain is used to change a stability derivative to a desired value.

For example, consider a pitch damper, which modifies the pitch damping derivative $C_{\mathrm{m_q}}$, shown in Eq. (7.1):

$$C_{\mathrm{m_{q_{des}}}} = C_{\mathrm{m_{q_{inherent}}}} + \Delta C_{\mathrm{m_{q_{SAS}}}} \,. \tag{7.1}$$

The SAS term resulting from the pitching moment dynamics is

$$\Delta C_{\mathrm{m_{q_{SAS}}}} \left(\frac{q\bar{c}}{2U_1} \right) = C_{\mathrm{m_{\delta_E}}} \Delta \delta_{\mathrm{E_{SAS}}} \,, \tag{7.2}$$

where

$$\Delta \delta_{\mathrm{E_{SAS}}} = K_{\mathrm{q}} q \,. \tag{7.3}$$

Here, K_{q} is the feedback gain. Solving for the gain using Eqs. (7.1–7.3), the feedback gain is then

$$K_{\mathrm{q}} = \frac{\left(C_{\mathrm{m}_{\mathrm{q}_{\mathrm{des}}}} - C_{\mathrm{m}_{\mathrm{q}_{\mathrm{inherent}}}}\right)\bar{c}}{2U_1 C_{\mathrm{m}_{\delta_{\mathrm{E}}}}}\,.$$

The approach above is based on closing SISO feedback loops; alternatively MIMO pole placement techniques can be used to close all the loops simultaneously.

This section introduces alternative methods for VS and IFS based on modern flight control synthesis techniques. First, the different degrees-of-freedom that can be simulated are discussed, and the type of system is chosen. Second, an approach for variable stability inspired by flying qualities control design is introduced. Third is a model following IFS technique for linear state-space models, a standard control objective for model reference controllers. Fourth, model following is used for in-flight simulation of 6-DOF *nonlinear* aircraft models. This approach allows the adaptive NDI controllers to simulate an aircraft over the entire modeled flight envelope. Finally, simulation results are presented for IFS controllers simulating a variety of aircraft types. The IFS objective is prioritized as the primary function of a variable stability SUAS is envisioned as simulating other UAS.

7.2 Degrees-of-Freedom

Historically, different variable stability aircraft have had different capability levels. The first variable stability aircraft had only a single DOF, which allowed it to change airplane dihedral effect [4]. Later aircraft had improved variable stability systems capable of changing up to 3-DOF for aircraft attitude (yaw, pitch, and roll). Ability to change throttle response led to 4-DOF systems [12]. The Total In-Flight Simulator aircraft was able to simulate full 6-DOF motion by using flaps for direct lift control and special vertical surfaces mounted on the wings for direct sideforce control [9].

To perfectly simulate an aircraft it is obvious that a 6-DOF system is desirable. However, this requires the addition of non-traditional control effectors to an aircraft to allow direct sideforce generation. This is expensive for manned aircraft; the current NF-16D

VISTA simulator was originally intended to simulate 6-DOF but was reduced to a 4-DOF system instead [11]. While this is less of a concern for SUAS, it is still desirable to reduce the modifications needed to the base airframe. Consequently, this thesis limits itself to 4-DOF systems. The system implemented is an "approximate" 4-DOF system where a NADI control law is used to simulate the 3-DOF of the aircraft attitude rates and an autothrottle is commanded to match airspeed changes obtained from the onboard reference model but does not allow the VSS pilot to command throttle. Note that while the results presented in this thesis are limited to 3/4-DOF systems (depending on whether the autothrottle is enabled), the basic approach generalizes to a full 6-DOF system.

7.3 Variable Stability Using Flying Qualities Based Desired Dynamics

Modern fighter aircraft with dynamic inversion based control laws often use desired dynamics based on military flying qualities specifications, which provide guidelines for selecting time constants, damping ratios, and natural frequencies for acceptable performance [47, 50]. Reference [48] presents forms of desired dynamics specifying flying qualities and ride qualities. The latter takes a form

$$\dot{X}_{\text{des}} = \frac{K_{\text{RQ}}}{s+b}\left(X_{\text{cmd}} - X\right) \tag{7.4}$$

where the constants K_{RQ} and b are chosen by the designer as shown below. The closed-loop transfer function for Eq. (7.4) is

$$\frac{X(s)}{X_{\text{cmd}}(s)} = \frac{K_{\text{RQ}}}{s^2 + bs + K_{\text{RQ}}},$$

which by comparison to a standard second-order system

$$\frac{Y(s)}{U(s)} = \frac{\omega_{\text{n}}^2}{s^2 + 2\zeta\omega_{\text{n}}s + \omega_{\text{n}}^2}$$

leads to the following definitions for K_{RQ} and b:

$$K_{\mathrm{RQ}} = \omega_{\mathrm{n,des}}^2 \, ,$$

$$b = 2\zeta_{\mathrm{des}}\omega_{\mathrm{n,des}} \, ,$$

where ζ_{des} and $\omega_{\mathrm{n,des}}$ are the desired damping ratio and undamped natural frequency, respectively.

As previously mentioned, values for these parameters are usually chosen to obtain desirable flying qualities. However, for the variable stability use-case the designer may simply pick desired values of ζ_{des} and $\omega_{\mathrm{n,des}}$. If first-order responses are desired, time constants may be specified instead using a proportional desired dynamics formulation:

$$\dot{X}_{\mathrm{des}} = \frac{1}{\tau_{\mathrm{des}}} \left(X_{\mathrm{cmd}} - X \right) \, . \tag{7.5}$$

For the 3-DOF system, Eqs. (7.4–7.5) are applied separately for each axis. This allows specification of the desired damping and frequencies for the roll, pitch, and yaw responses. For cases where finer control over the responses is desired, it may be preferable to use a model following approach as described below with modified stability derivatives to obtain the desired responses and level of stability. Note that since this approach is essentially identical to the "design-to-flying-qualities" approach of the example CAS utilized throughout this thesis, differing only in the manner of specifying the desired flying qualities, simulation results are not presented as this method has effectively been demonstrated in Section 6.

7.4 In-Flight Simulation Using Reference Models

Model following controllers are the best option for in-flight simulation of an exact aircraft as the IFS control objective can be met simply by specifying the desired aircraft dynamics as the reference model. The use of linear and nonlinear reference models for a 4-DOF IFS system is described below.

7.4.1 Linear Reference Models

This approach to an IFS controller is similar to that found in Ref. [16]. Assume that a linear state-space model of the form

$$\begin{aligned} \dot{\boldsymbol{x}}_{\mathrm{ref}} &= [A]\boldsymbol{x}_{\mathrm{ref}} + [B]\boldsymbol{\nu}\,, \\ \boldsymbol{y}_{\mathrm{des}} &= [C]\boldsymbol{x}_{\mathrm{ref}}\,, \end{aligned} \tag{7.6}$$

is available for the aircraft to simulate. In this section the class of models is limited to those that are strictly-proper (i.e. $[D] = [0]$) although this is not strictly necessary. In Eq. (7.6) the vector $\boldsymbol{\nu}$ is a pseudo-control that is either the pilot's stick commands or an experimental control law that is applied to the reference aircraft model. Note that an alternative in the latter case is to simply specify the closed-loop reference model to simplify implementation. The output $\boldsymbol{y}_{\mathrm{des}}$ is chosen as the body axis angular rates p, q, and r of the linear model for a 3-DOF IFS design, and the desired dynamics are then

$$\dot{\boldsymbol{y}}_{\mathrm{des}} = [C][A]\boldsymbol{x}_{\mathrm{ref}} + [C][B]\boldsymbol{\nu}\,. \tag{7.7}$$

Equation (7.7) is then used as the desired dynamics in one of Eq. (6.4) or Eq. (6.20). It is required that the matrix $[A]$ be Hurwitz or the pseudo-control $\boldsymbol{\nu}$ stabilize the system if $[A]$ is unstable for the control laws to successfully replicate the simulated aircraft's dynamics. Since aircraft linear models are often decoupled into longitudinal (u/V_{T}, w/α, q, θ) and lateral/directional (v/β, p, r, ϕ, ψ) sets, Eq. (7.7) can be repeated for each axis, selecting q for longitudinal desired dynamics and p and r for lat/d desired dynamics. Alternatively, an augmented system can be formed as

$$\begin{Bmatrix} \dot{\boldsymbol{x}}_{\text{long}} \\ \dot{\boldsymbol{x}}_{\text{latd}} \end{Bmatrix} = \begin{bmatrix} [A_{\text{long}}] & [0] \\ [0] & [A_{\text{latd}}] \end{bmatrix} \begin{Bmatrix} \boldsymbol{x}_{\text{long}} \\ \boldsymbol{x}_{\text{latd}} \end{Bmatrix} + \begin{bmatrix} [B_{\text{long}}] & [0] \\ [0] & [B_{\text{latd}}] \end{bmatrix} \begin{Bmatrix} \boldsymbol{u}_{\text{long}} \\ \boldsymbol{u}_{\text{latd}} \end{Bmatrix}$$

and a new output equation chosen appropriately.

For a 4-DOF system, the reference aircraft airspeed is added as an output and is used as an input to the autothrottle. Since the linear model velocity state is a perturbation to the trimmed reference model value, the velocity term can be interpreted as a delta-velocity command to the current aircraft velocity. This allows tracking the simulated aircraft's airspeed changes resulting from the given aerodynamic control effector inputs but does not allow the VS pilot full control of the simulated aircraft's throttle. The lack of throttle control is a consequence of the fact that the linear models used neglect throttle dynamics and do not include throttle as a control.

Note that since the desired dynamics are based off of linear models the range of validity is limited to that of the linear model. Determining the range of validity is in general a difficult problem although models are usually valid for small perturbations (e.g. angles less than $\pm 15°$). A possible solution is to gain-schedule the linear reference models based on flight condition. If available, using a full nonlinear model as described below is a superior solution.

7.4.2 Nonlinear Reference Models

This section takes the same approach as the previous, but instead of the linear reference model Eq. (7.6), a nonlinear model is used:

$$\begin{aligned} \dot{\boldsymbol{X}}_{\text{ref}} &= \boldsymbol{F}(\boldsymbol{X}_{\text{ref}}, \boldsymbol{\nu}) \,, \\ \boldsymbol{Y}_{\text{des}} &= \boldsymbol{H}(\boldsymbol{X}_{\text{ref}}) \,. \end{aligned} \tag{7.8}$$

As before the selected outputs $\boldsymbol{Y}_{\text{des}}$ correspond to the DOF the controller is attempting to simulate, which for this thesis are again the body axis angular rates. The desired dynamics are then specified by $\dot{\boldsymbol{Y}}_{\text{des}}$. Note that in general the nonlinear model should be stable (or have a stabilizing controller applied via $\boldsymbol{\nu}$) for the IFS control law to successfully work.

For implementation, an onboard model of the vehicle aerodynamics and propulsion is required similarly to an NDI controller. The VSS then uses the reference aircraft OBM to compute the desired aircraft state vector derivative. The control variables are extracted from this vector and are used as desired dynamics for the NADI controller for the 3-DOF attitude simulation. For the fourth DOF of the system, throttle, a similar approach is used as with the linear models with slight modification. Here, the airspeed is the “full” nonlinear value of the reference aircraft. To generate a “delta” command, a perturbation quantity is calculated from the trimmed velocity of the reference aircraft. This value is then passed to the autothrottle as before.

This approach is more computationally expensive as a full nonlinear model must be simulated in real time, however it allows simulation over to entire modeled flight envelope. In this thesis the approach taken is to use the generic nonlinear parametric aerodynamic model found in Ref. [43], which is reproduced in Eqs. (2.46–2.51). This is chosen for two reasons: first, the model is compact with only 45 parameters, and second, the paper provides model parameters for several different aircraft, which enables multiple simulation studies. A model of the engine dynamics is generated using a generic turbofan model provided with the Simulink Aerospace Blockset. This block is configured by specifying the max thrust of the simulated aircraft, and a one second time constant is assumed for the engine dynamics.

7.5 Flight Control System Integration

The previous material discusses the subset of VSS design particular to the control laws, but does not address the larger picture of how the variable stability system integrates with the larger whole of the flight control system. Manned IFS aircraft such as the VISTA generally are dual-seat configurations, with an evaluation pilot and a safety pilot [74]. The evaluation pilot flies using the VSS system, while the safety pilot has direct control of the aircraft's baseline flight control system and can override the VSS system if necessary. The evaluation pilot can also access the baseline flight control system if necessary. A set of safety monitors can also revert to the baseline control laws if failures in the VSS are detected.

An SUAS VSS would likely use a flight controller following a design similar to that used by the NASA AirSTAR flight test infrastructure [75]. The AirSTAR architecture features a flight control system with three main modes:

- Mode 1: Stick-to-Surface reversion control law
- Mode 2: Baseline control law
- Mode 3: Research control law

The Mode 1 stick-to-surface control law has the pilot command the control effectors directly, and is also used by an RC safety pilot. This mode is a backup, "get-home" mode. Mode 2 is a baseline control law that is designed as a Stability Augmentation System (SAS) to reduce pilot workload. Mode 3 is an arbitrary research control that is being tested, and is enabled only during flight tests for specific test points. For the VSS design in this thesis, Mode 2 would be implemented using a Command Augmentation System similar to those developed in Section 6, while Mode 3 would be the VSS control law designs presented in this section. The three modes would be implemented in Simulink, and could easily be switched between using a standard three position switch found on RC controllers. Additionally, an outer loop safety monitor could be used to revert to either Mode 2 or Mode 1 control laws if the aircraft began to leave its nominal flight envelope. Such a safety monitor would likely be based on or operating in conjunction with an envelope protection system, the design of which lies outside the scope of this thesis.

If a software solution is deemed insufficiently reliable, hardware solutions exist. Current open-source SUAS flight control hardware is sufficiently inexpensive that, payload capacity and power permitting, it is possible to place two flight controllers on a vehicle. One would implement the VSS experimental control law, while the other would feature the stock autopilot software. Both would run concurrently and the pilot would be able to switch between the VSS autopilot and the baseline autopilot via standard RC multiplexers. An onboard safety monitor could also trigger the switch to the baseline controller (for safety reasons the reverse is prohibited). This approach is similar to that currently used by the VSCL system identification flight test system, which uses a mux to switch between an autopilot configured for autoexcitation and standard RC hardware for direct stick-to-surface control [76]. An advantage of having an unmodified autopilot as a fallback is the ability to use higher-level autopilot modes such as LOITER, WAYPOINT, and AUTOLAND to have the SUAS automatically fly back to a predefined waypoint or land if a fault in the VSS occurs.

7.6 Linear IFS Simulation Results

Two aircraft are simulated using the linear IFS approach: the McDonnell-Douglas F-4 Phantom II jet fighter and the NASA Generic Transport Model (GTM) T-2, a 5.5% scaled Boeing 757 Remotely Piloted Aircraft (RPA). These aircraft are chosen as both linear and nonlinear aircraft models are easily available. Additionally, the two aircraft are very dissimilar: one is an air combat fighter and the other is an RPA with the handling characteristics of a commercial air transport. The GTM Simulink model additionally allows the generation of coupled linear models, allowing comparison with the decoupled approach of the F-4 linear models. Data for the F-4 is taken from the flight dynamics textbook by Roskam, where DATCOM-based linear stability and control derivatives are available as “Airplane I” [36]. The GTM linear model is obtained from the NASA open-source model available on GitHub[1] [77]. Results for additional aircraft models are included in Appendix A.

7.6.1 MRAC NDI Control Law

This section presents results for linear in-flight simulation accomplished using the MRAC NDI control law described in Section 6.2. This is the same controller as used for the P-Q-β CAS in Section 6 with controller parameters listed in Table 6.3.

7.6.1.1 McDonnell-Douglas F-4 Phantom II

Figure 7.1a plots the state time histories from a simulation of a linearized F-4 model. Observing the plots, the controller is able to achieve good tracking of the roll, pitch, and yaw rates, which are directly controlled by the aerodynamic control effectors. The angle-of-attack and sideslip angle show acceptable tracking as well; this is a result of the angular rate tracking. Note that it is a well-known result that the number of outputs tracked must be less than or equal to the number of independent control effectors; accordingly this limits tracking to the angular rates and airspeed. A slight bias is observed in the AOA, but the dynamics are replicated well. Finally, the autothrottle tracks the reference airspeed relatively accurately.

The control time histories are shown in Fig. 7.1b. The “reference” controls are doublets that are used to perturb the linear reference model, while the “commanded” controls are calculated by the MRAC NDI controller. Finally, the “achieved” controls are the ac-

---

[1] GTM Design Sim: `https://github.com/nasa/GTM_DesignSim`

tual effector displacements obtained after the position and rate limited first-order actuator dynamics. Note that the controller produces some high-frequency commands near the “sharp” points resulting from the linear doublet responses. Finally, the adaptive parameters are shown in Fig. 7.2.

7.6.1.2 NASA Generic Transport Model

The GTM presents an interesting case given the difference in scale and flight envelope from the F-16 plant used as the in-flight simulator. State time histories for the angular rates, airspeed, and incidence angles are presented in Fig. 7.3a. The responses show good tracking of the angular rates. Note in particular the tracking of the high-frequency response at the 10 second mark. Good tracking of sideslip results from the tracking of the lateral/directional rates. The angle-of-attack captures the general dynamics, but does not track the actual response from the linear model.

The aerodynamic control effector time histories are shown in Fig. 7.3b. Note the large magnitudes of the inputs given to the linear GTM model, which result in the oscillatory responses as seen in Fig. 7.3a. In general the position and rate limits are met, although the elevator rate limits are saturated while attempting to track the reference GTM elevator doublet, which results in approximately $2\,{}^\circ/\mathrm{s}$ tracking error. Adaptive parameters are plotted in Fig. 7.4.

7.6.2 $\mathcal{L}_1$ NDI Control Law

This section presents results for the same aircraft, but using the $\mathcal{L}_1$ adaptive dynamic inversion controller derived in Section 6.3. The controller parameters are the same as those designed for the CAS example in Section 6; these parameters are listed in Table 6.4.

7.6.2.1 McDonnell-Douglas F-4 Phantom II

Figure 7.5a plots the state time histories using the $\mathcal{L}_1$ adaptive NDI control law for linear in-flight simulation of the F-4. As with the MRAC NDI control law, excellent tracking of all the states is achieved. The $\mathcal{L}_1$ control law shows slightly improved tracking performance; for instance, the angle-of-attack bias in the MRAC NDI response is not present. Figure 7.5b plots the F-4 control time histories. The $\mathcal{L}_1$ control law produces less demand on the actuators. Rate limits are saturated briefly while trying to track the doublet responses.

7.6.2.2 NASA Generic Transport Model

The $\mathcal{L}_1$ controller results for linear IFS of the GTM aircraft are presented in Fig. 7.7a. The $\mathcal{L}_1$ controller achieves good tracking of all the states and has performance similar to that of the MRAC NDI control law. An overshoot of near $2\,^{\circ}/\mathrm{s}$ is seen in the pitch rate, resulting from rate saturation in the elevator command. The AOA and sideslip angle tracking is generally good and captures the dynamics. As with the F-4, the AOA tracking is improved over the MRAC NDI controller. Control effector commands and deflections are plotted in Fig. 7.7b, showing that the commanded controls meet position limits but at times saturate rate limits. The adaptive parameter time histories are shown in Fig. 7.6.

7.7 Nonlinear IFS Simulation Results

Nonlinear IFS simulation results are presented using nonlinear models of the F-4 and the GTM. Aerodynamic data for the models is taken from Ref. [43], while a generic turbofan is used as an engine model. Results are presented for both the MRAC and $\mathcal{L}_1$ adaptive dynamic inversion controllers.

7.7.1 MRAC NDI Control Law

This section presents nonlinear reference model in-flight simulation results for the MRAC-augmented nonlinear dynamic inversion controller.

7.7.1.1 McDonnell-Douglas F-4 Phantom II

Figure 7.9a plots the nonlinear IFS results for the F-4 Phantom II. The nonlinear model results in oscillations of greater magnitude and frequency that the linear model and thus poses a more challenging control problem. Note that because the reference model is initialized at the state of the F-16 IFS plant it is not at a trimmed flight condition and thus the reference model begins oscillating in pitch. Good tracking performance of all states is achieved, with the exception of the AOA. The AOA *dynamics* are replicated, but the MRAC NDI response's magnitude has an approximately 4° offset from the reference model. Commanded and actual control deflections are shown in Fig. 7.9b. As before, position limits are satisfied while rate limits are violated. Adaptive parameters are shown in Fig. 7.10.

7.7.1.2 NASA Generic Transport Model

State time histories for the GTM nonlinear IFS example using the MRAC NDI control law are presented in Fig. 7.11a. The nonlinear GTM model responses are larger in magnitude than the linear model, and because the reference model is not initially in trim the oscillations begin almost immediately. Note that oscillations in the roll and yaw rates are lower frequency than the linear model. Once again, the MRAC NDI controller demonstrates good tracking performance of the states, in particular the angular rates that are controlled directly. Biases are present in the AOA and sideslip tracking, which are controlled only indirectly. Control effector deflections are plotted in Fig. 7.11b, while adaptive parameters are plotted in Fig. 7.12.

7.7.2 $\mathcal{L}_1$ NDI Control Law

This section presents nonlinear reference model in-flight simulation results for the $\mathcal{L}_1$ adaptive dynamic inversion controller.

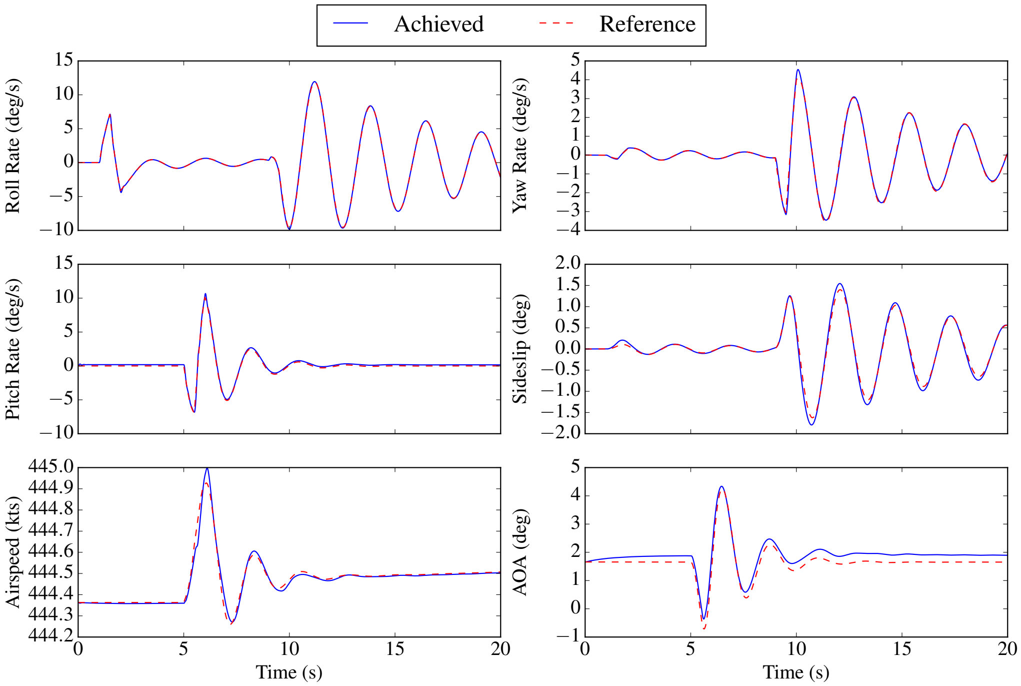
7.7.2.1 McDonnell-Douglas F-4 Phantom II

Time histories for the F-4 Phantom II simulation using the $\mathcal{L}_1$ controller are presented in Fig. 7.13a. The $\mathcal{L}_1$ controller achieves the IFS tracking objectives for the angular rates. Good tracking of sideslip is achieved indirectly. In contrast to the linear simulation results, the $\mathcal{L}_1$ controller is unable to correct for the bias in the AOA and shows similar tracking performance to the MRAC NDI control law. Like the MRAC augmented dynamic inversion controller, the $\mathcal{L}_1$ controller has a bias of approximately $4°$.

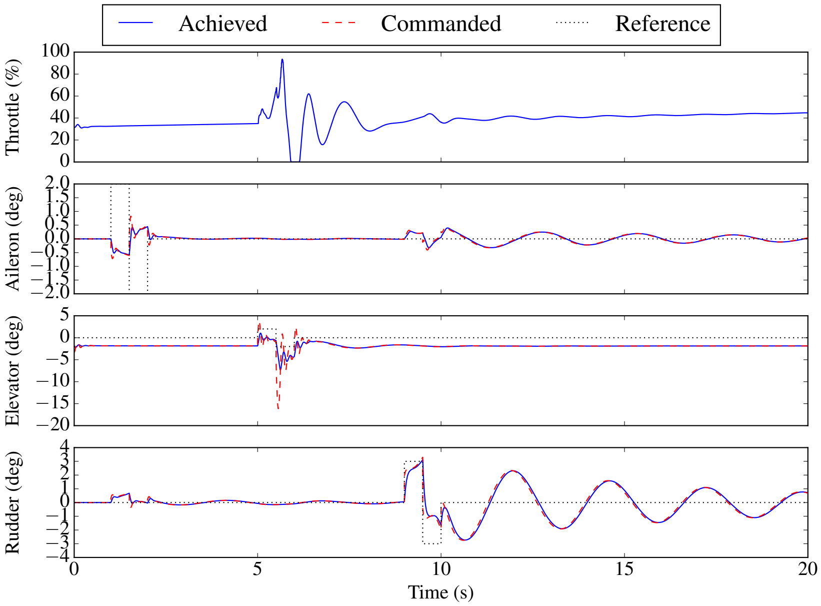
Figure 7.13b plots the control effector time histories. Compared to the MRAC NDI controller, the $\mathcal{L}_1$ control law demands less of the actuators for roll and pitch control, but places higher demands on the rudder for yaw control. Consistent with previous simulations, position limits are met while rate limits are saturated at times. Adaptive parameters are shown in Fig. 7.14.

7.7.2.2 NASA Generic Transport Model

Finally, nonlinear simulation results are presented for the NASA GTM research aircraft using the $\mathcal{L}_1$ adaptive dynamic inversion controller. State time histories are shown in Fig. 7.15a. The IFS tracking objective is again met, with good tracking of all states except AOA. The controller is able to recreate the character of each state's dynamic response. Very slight improvement over the performance of the MRAC NDI control law is observed. As with the F-4 simulation, Fig. 7.15b shows that the $\mathcal{L}_1$ controller meets actuator position and rate limits and produces less demand for aileron and elevator while demanding more rudder than the MRAC NDI control law. Figure 7.16 presents adaptive parameters.



(a) Actual and desired state time histories.



(b) Aerodynamic control effector time histories.

Figure 7.1: Linear F-4 Phantom II MRAC NDI controller state and control time histories.

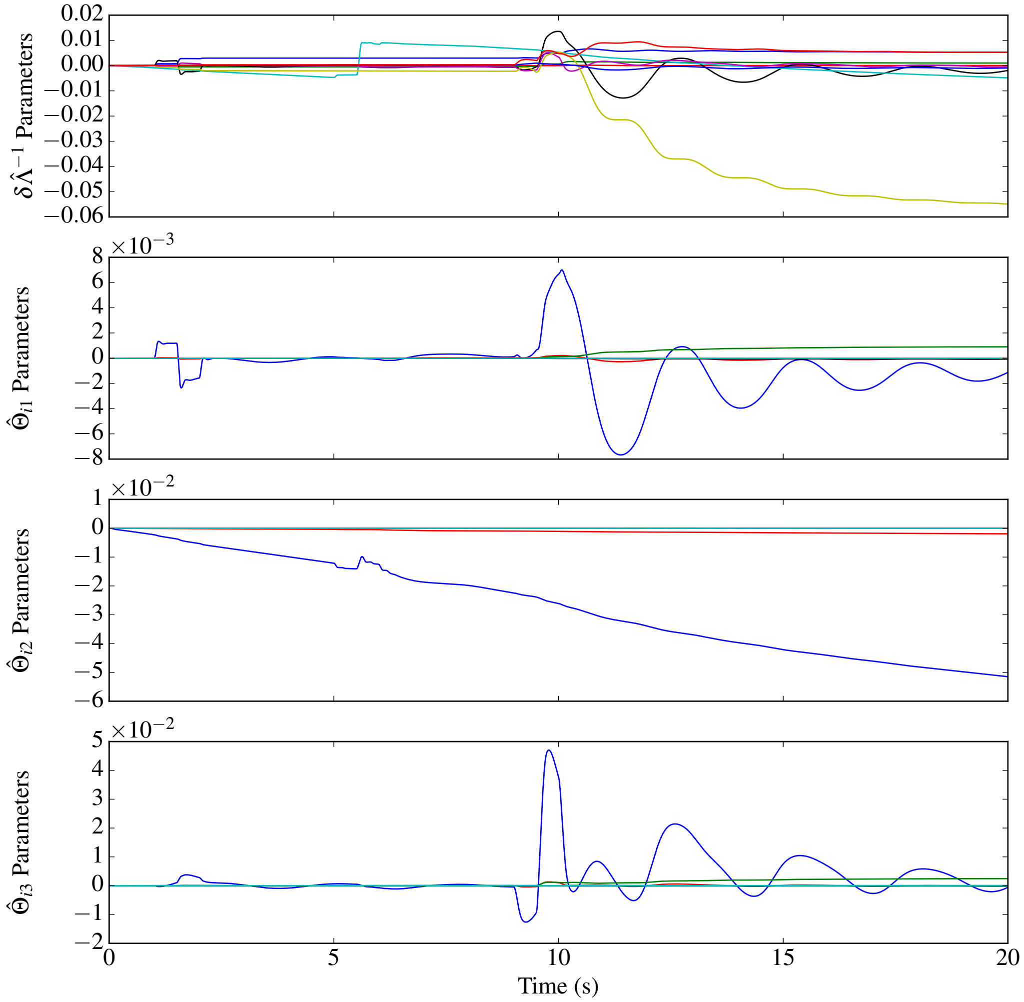
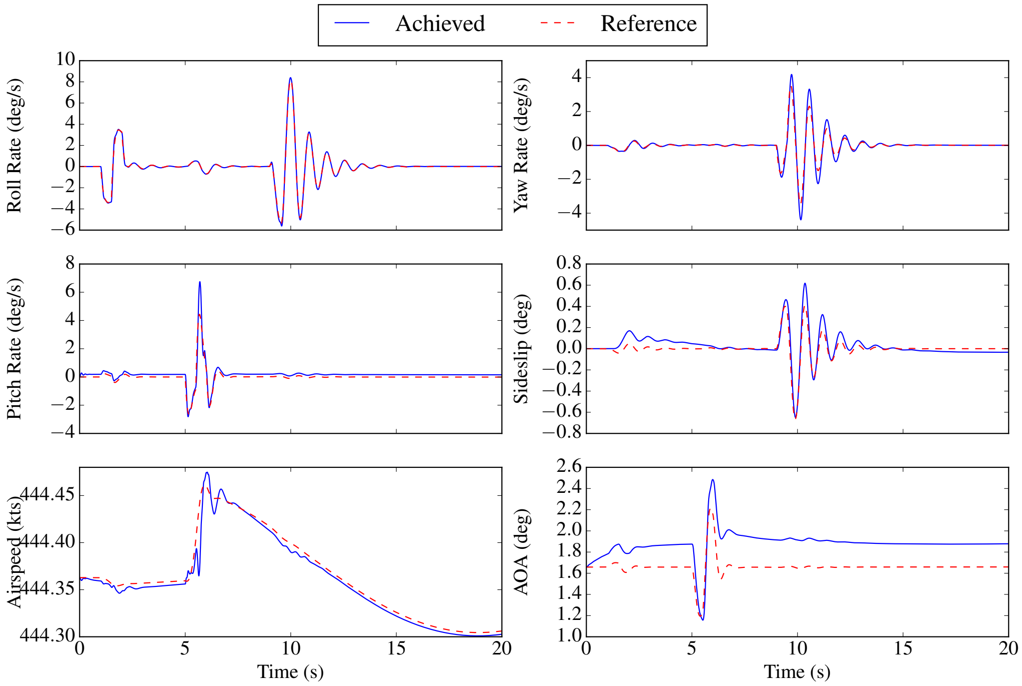
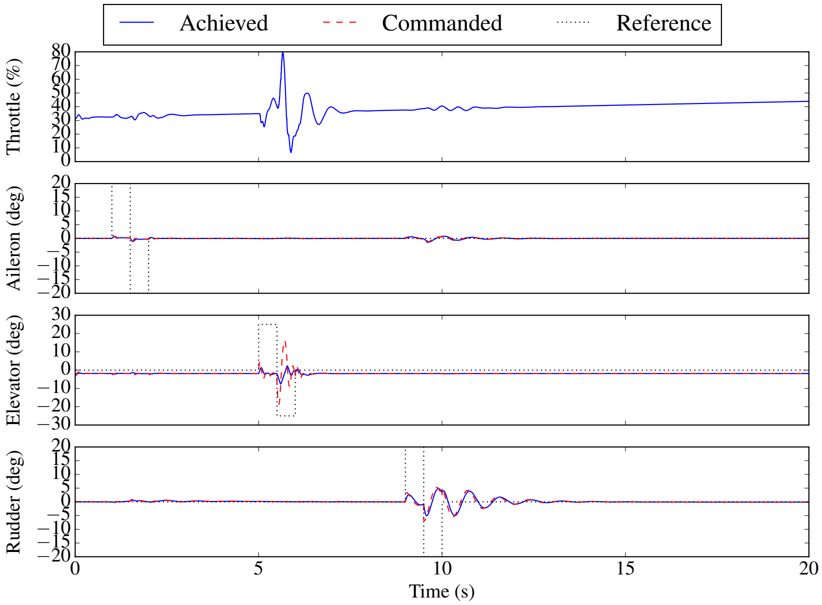


Figure 7.2: Linear F-4 Phantom II MRAC NDI adaptive parameter time histories.



(a) Actual and desired state time histories.



(b) Aerodynamic control effector time histories.

Figure 7.3: Linear Generic Transport Model MRAC NDI controller state and control time histories.

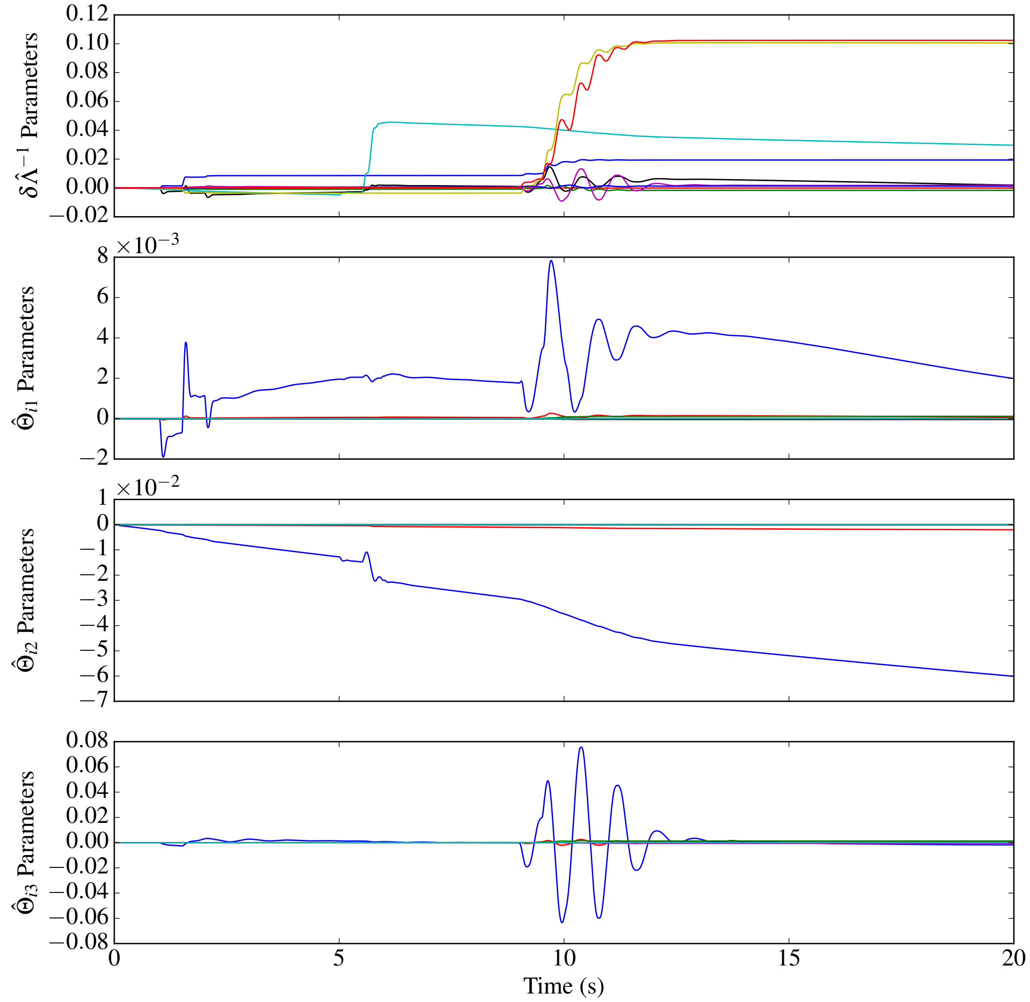
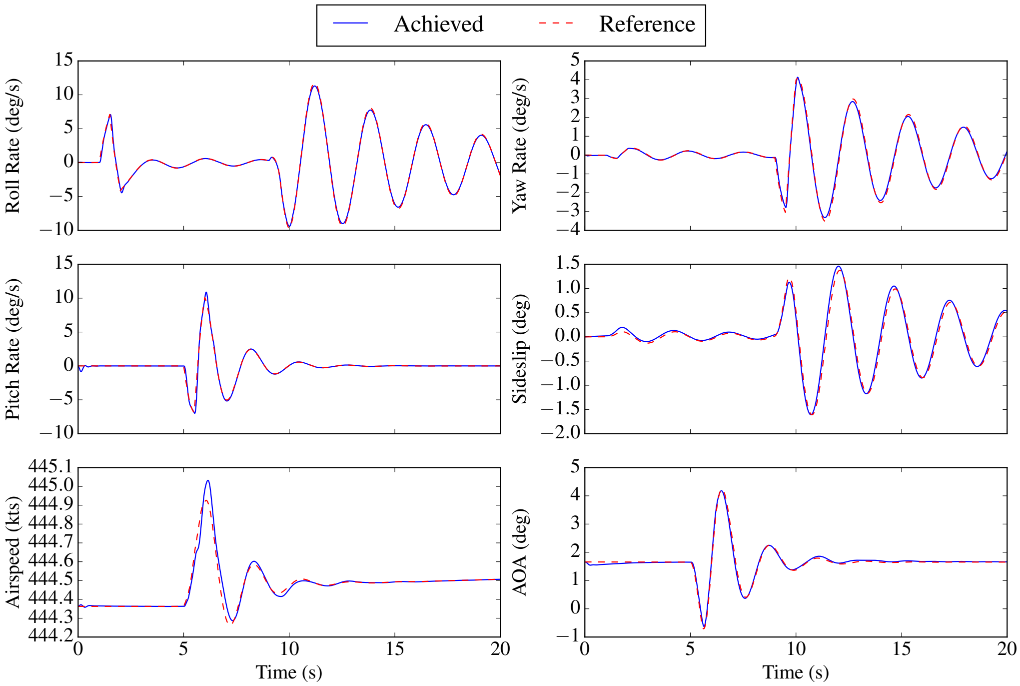
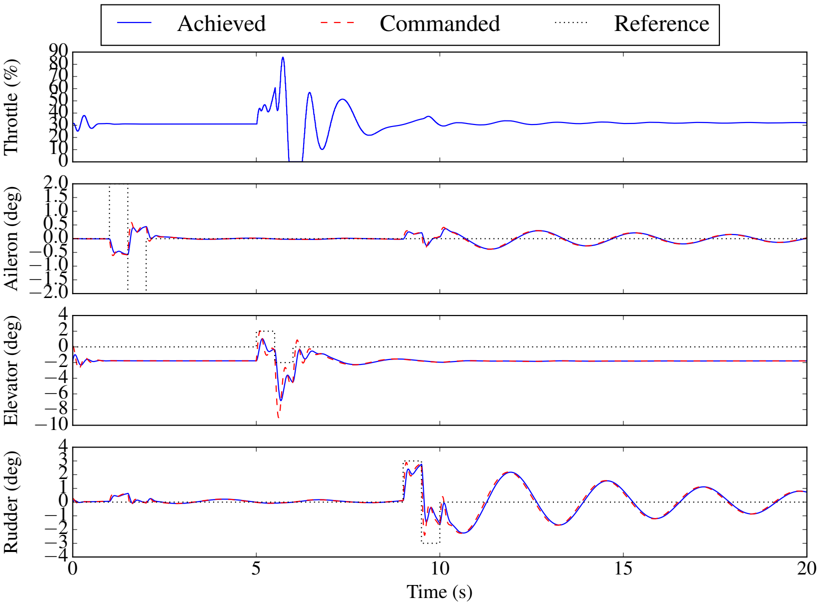


Figure 7.4: Linear Generic Transport Model MRAC NDI adaptive parameter time histories.



(a) Actual and desired state time histories.



(b) Aerodynamic control effector time histories.

Figure 7.5: Linear F-4 Phantom II $\mathcal{L}_1$ NDI controller state and control time histories.

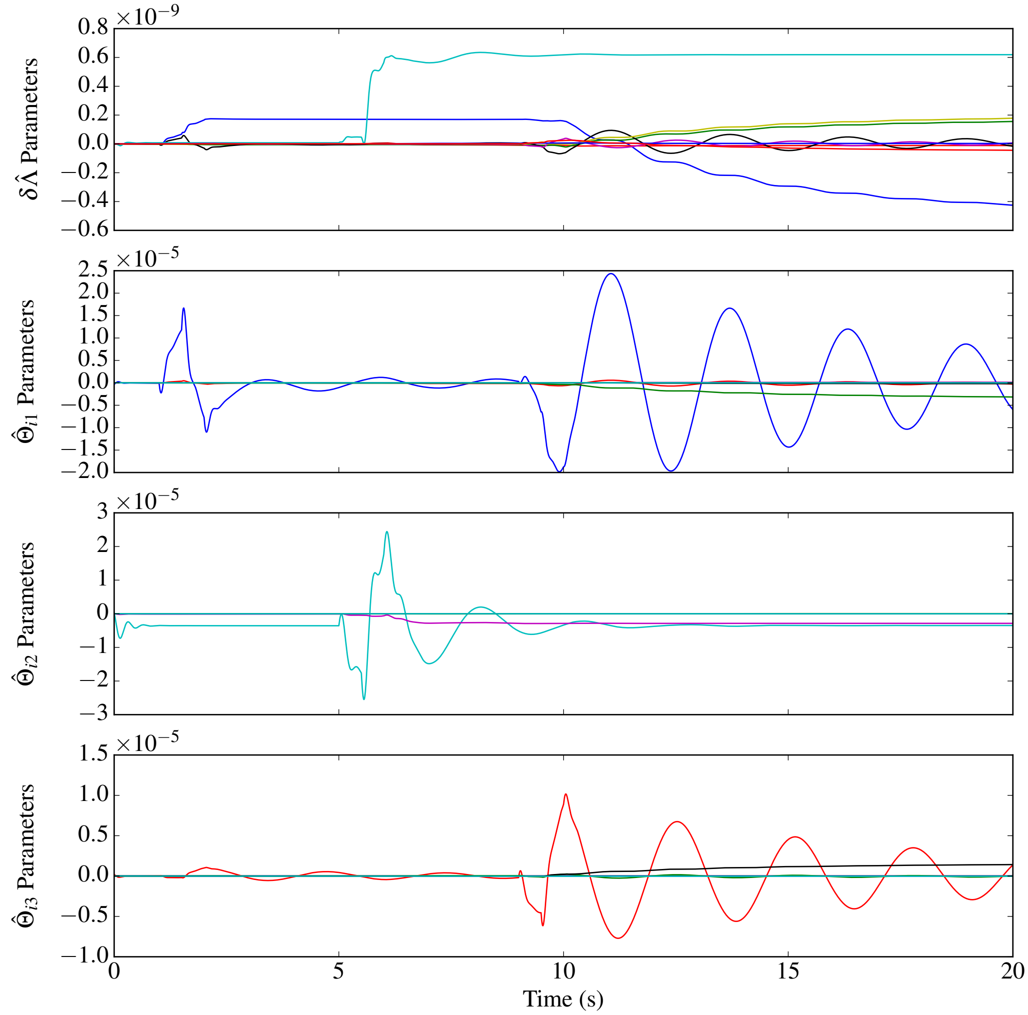
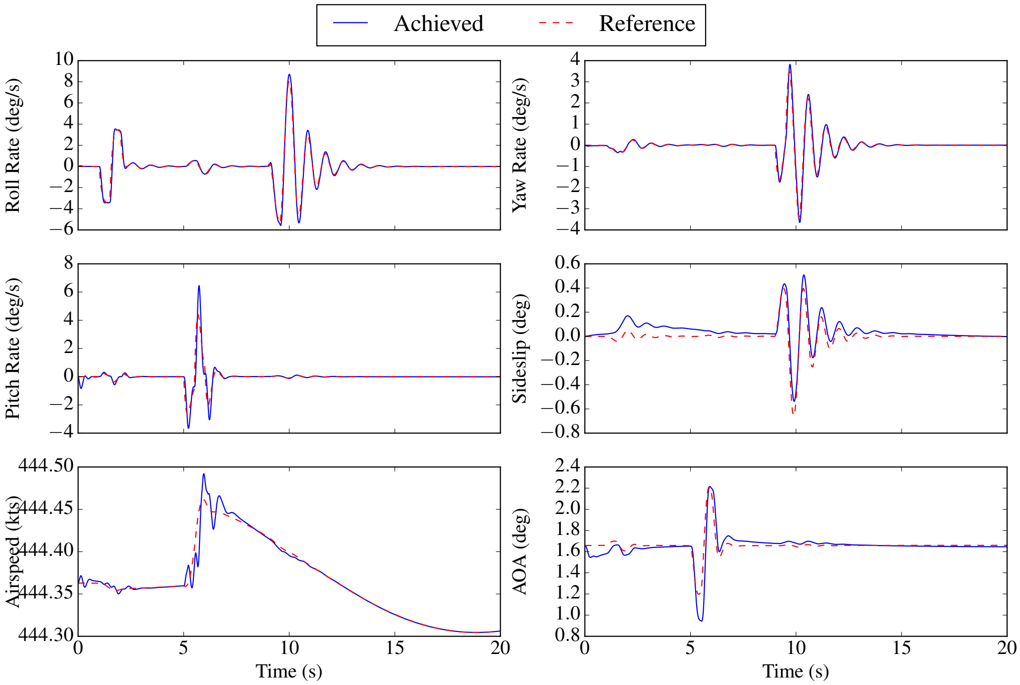
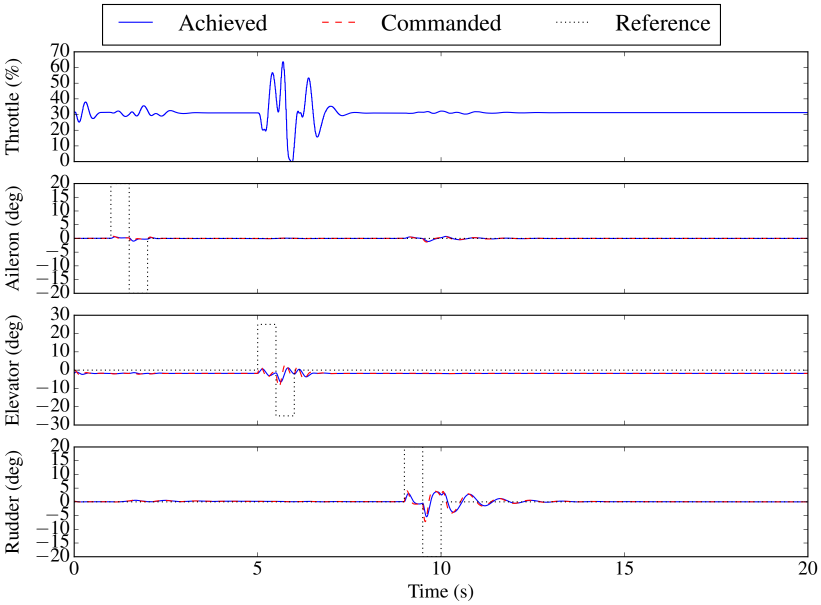


Figure 7.6: Linear F-4 Phantom II $\mathcal{L}_1$ NDI adaptive parameter time histories.



(a) Actual and desired state time histories.



(b) Aerodynamic control effector time histories.

Figure 7.7: Linear Generic Transport Model $\mathcal{L}_1$ NDI controller state and control time histories.

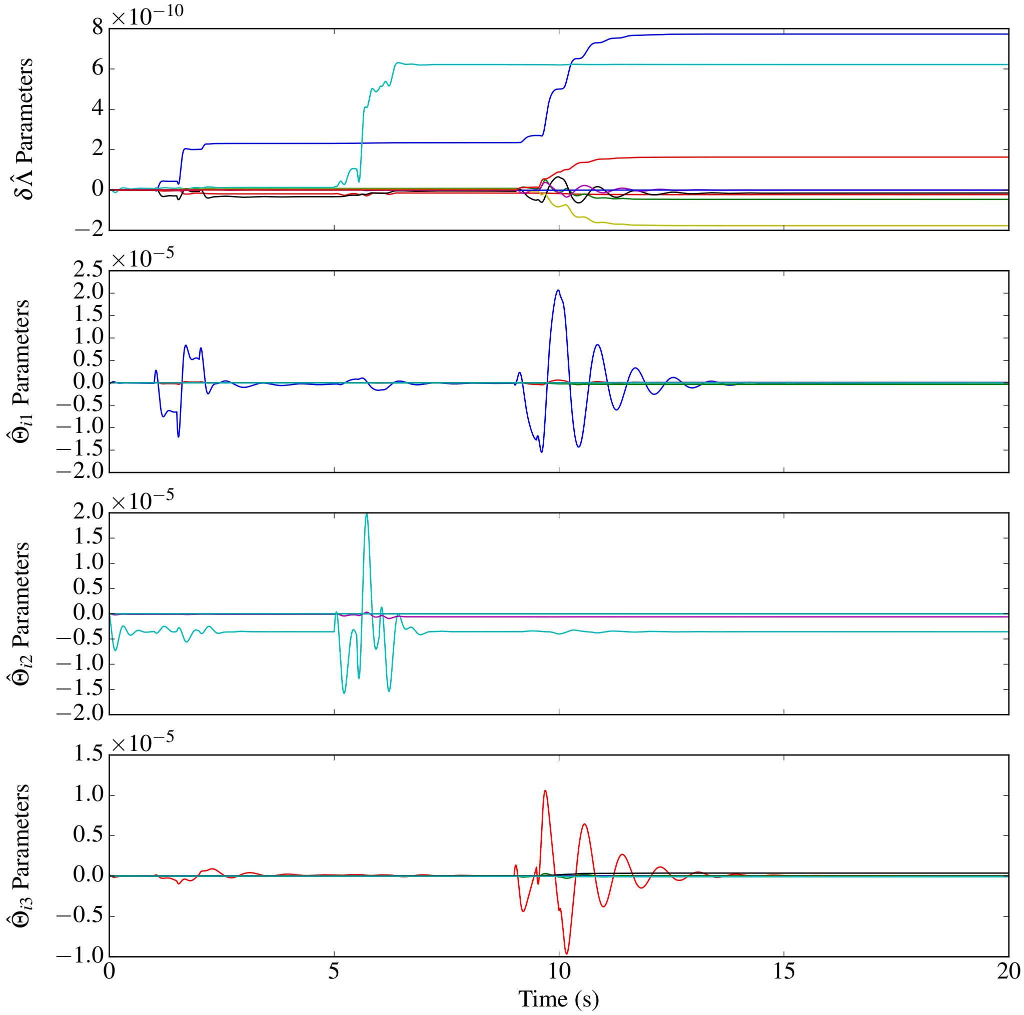
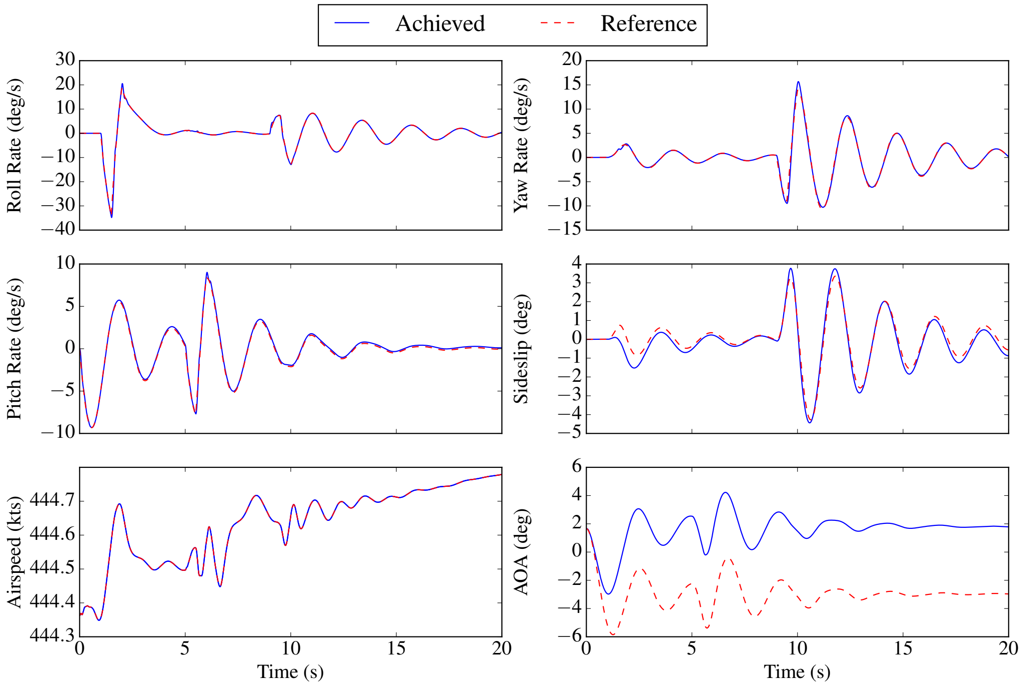
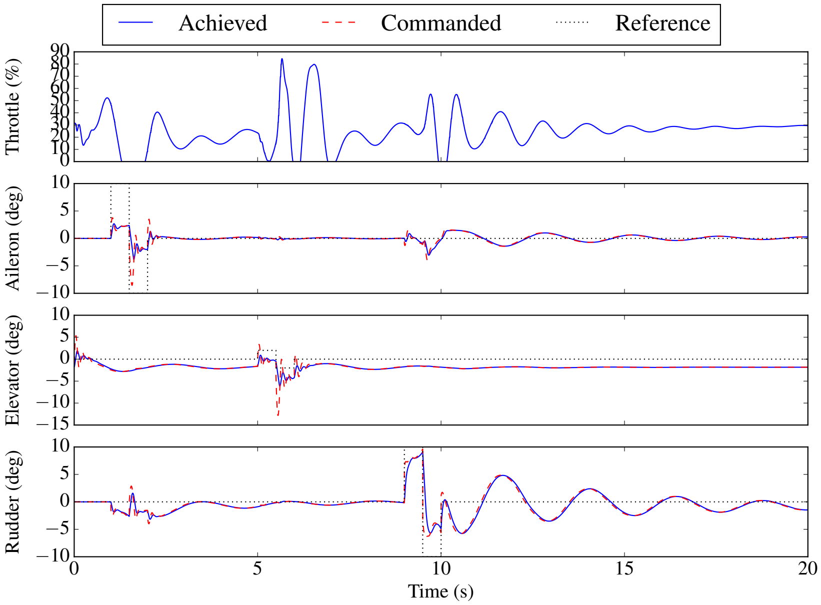


Figure 7.8: Linear Generic Transport Model $\mathcal{L}_1$ NDI adaptive parameter time histories.



(a) Actual and desired state time histories.



(b) Aerodynamic control effector time histories.

Figure 7.9: Nonlinear F-4 Phantom II MRAC NDI controller state and control time histories.

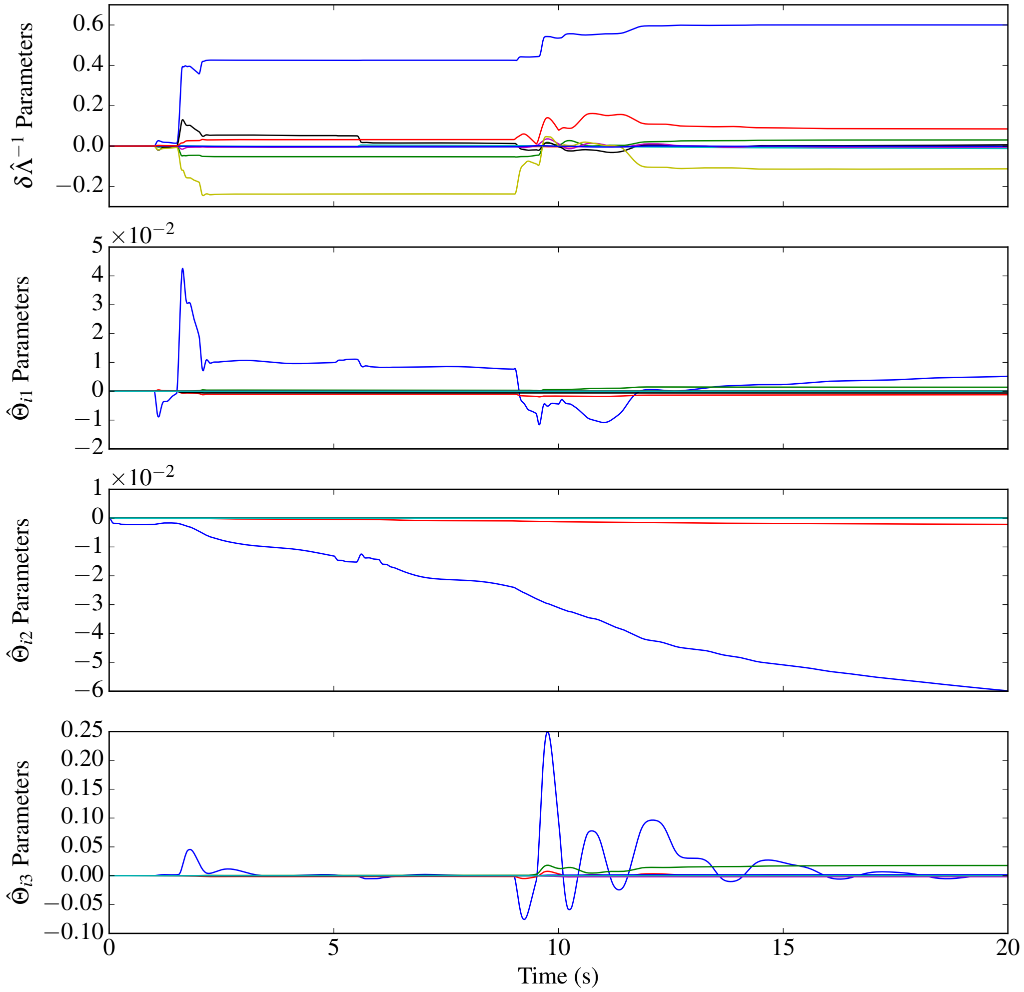
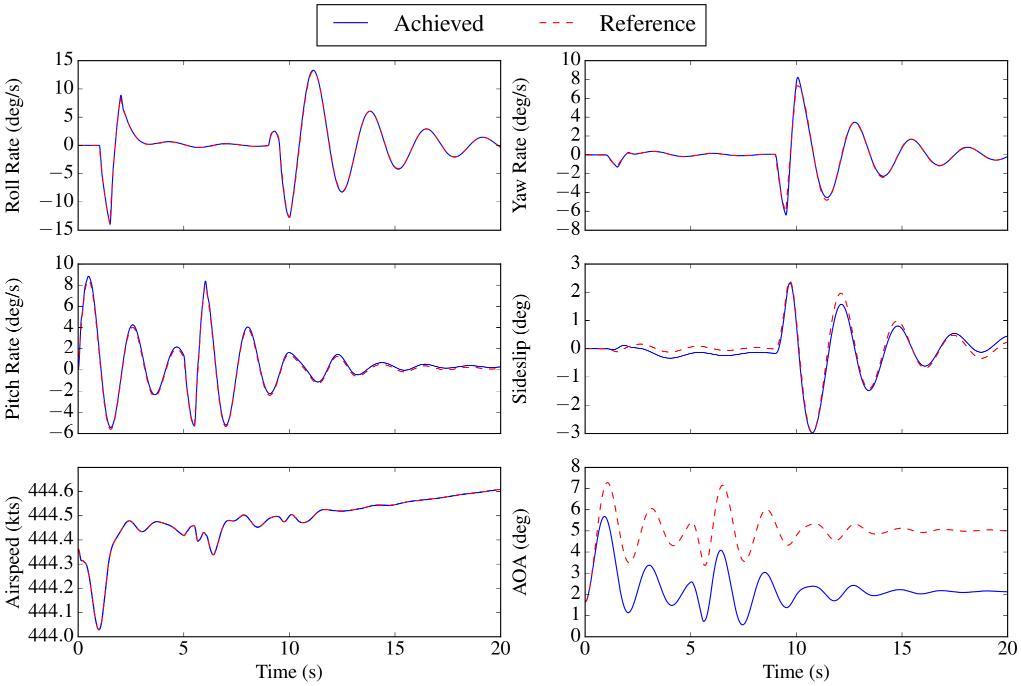
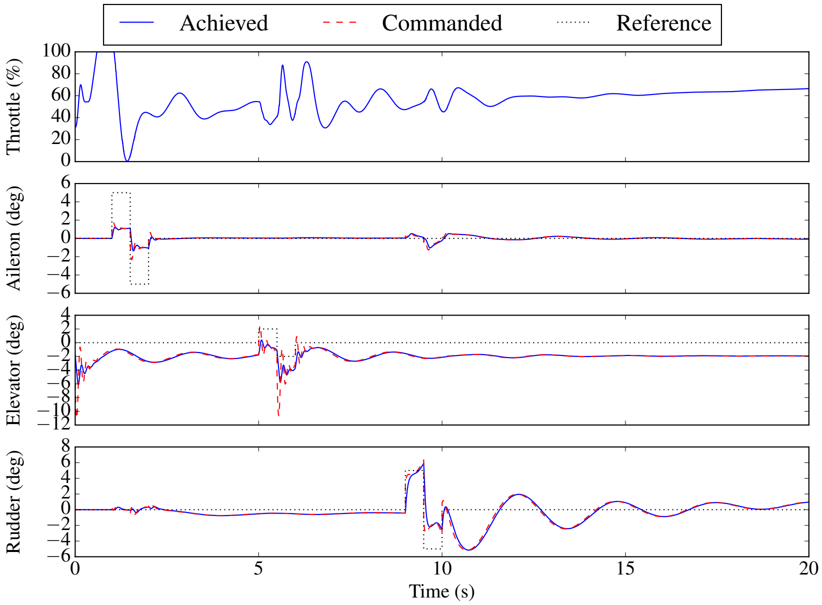


Figure 7.10: Nonlinear F-4 Phantom II MRAC NDI adaptive parameter time histories.



(a) Actual and desired state time histories.



(b) Aerodynamic control effector time histories.

Figure 7.11: Nonlinear Generic Transport Model MRAC NDI controller state and control time histories.

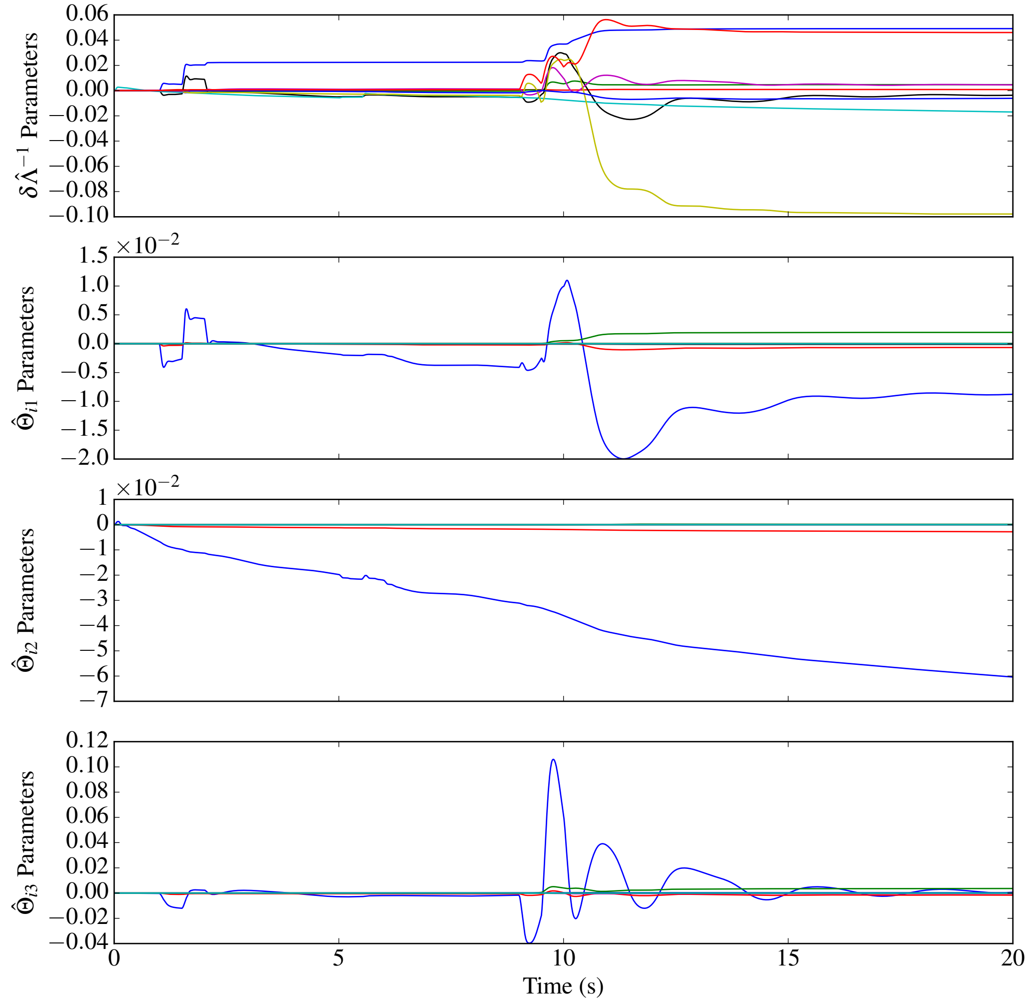
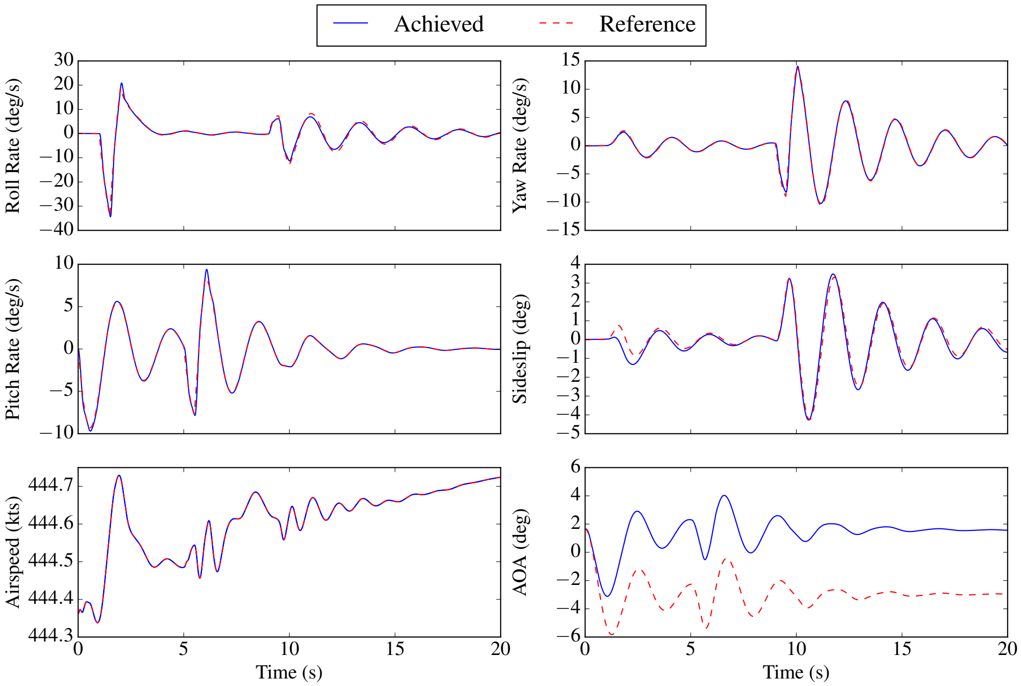
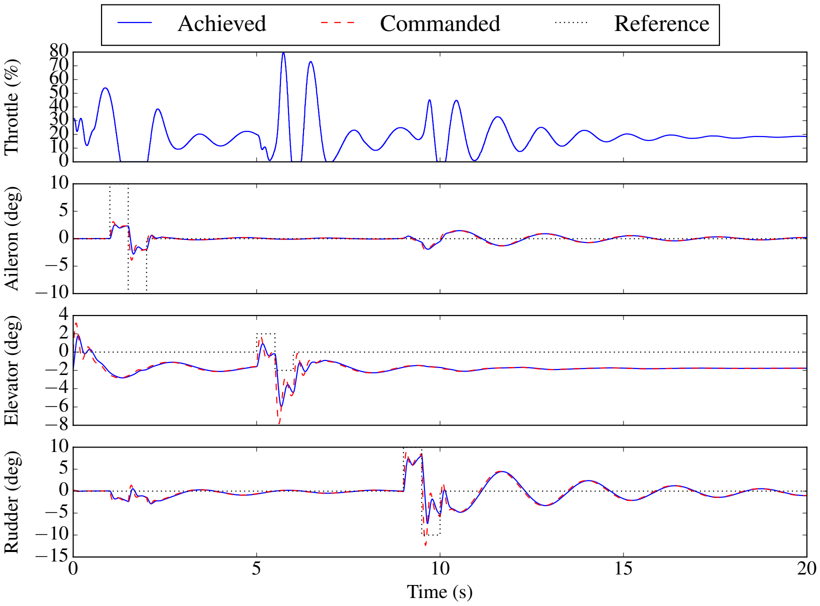


Figure 7.12: Nonlinear Generic Transport Model MRAC NDI adaptive parameter time histories.



(a) Actual and desired state time histories.



(b) Aerodynamic control effector time histories.

Figure 7.13: Nonlinear F-4 Phantom II $\mathcal{L}_1$ NDI controller state and control time histories.

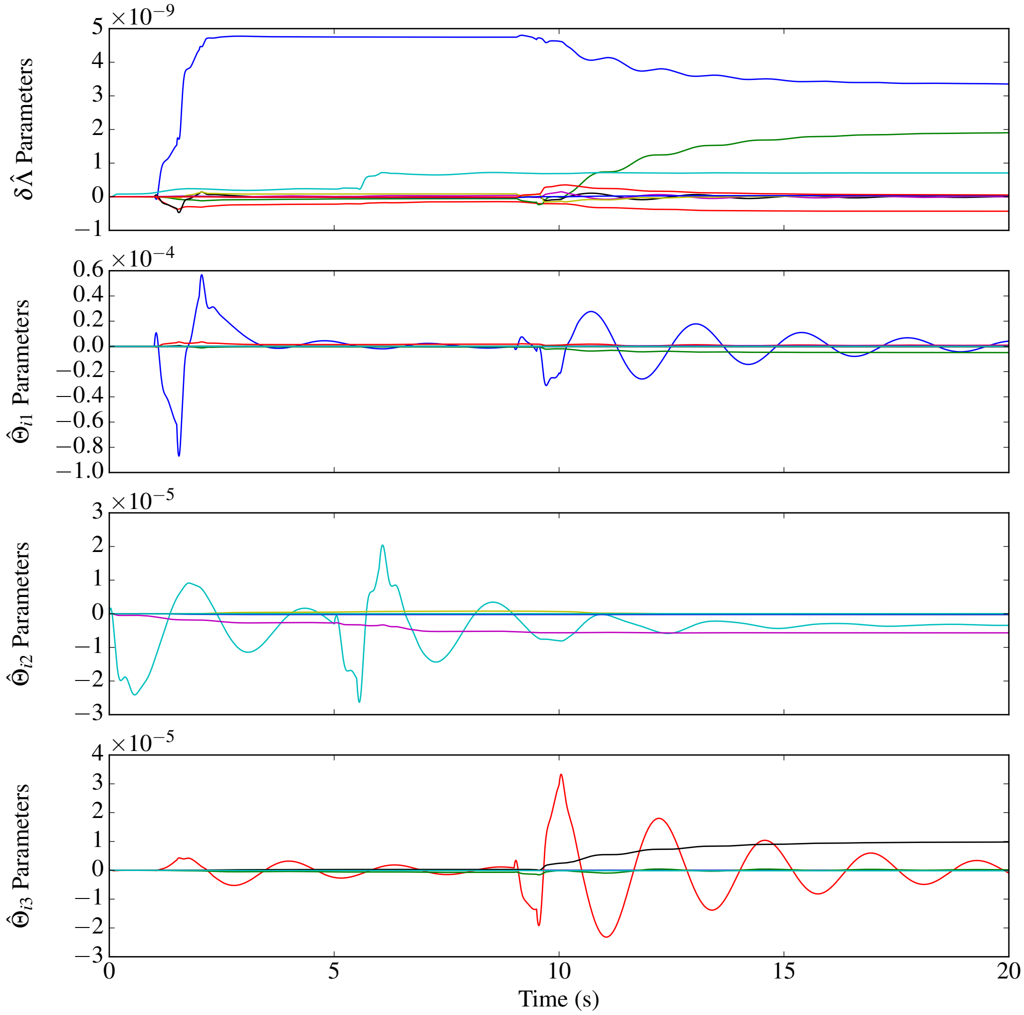
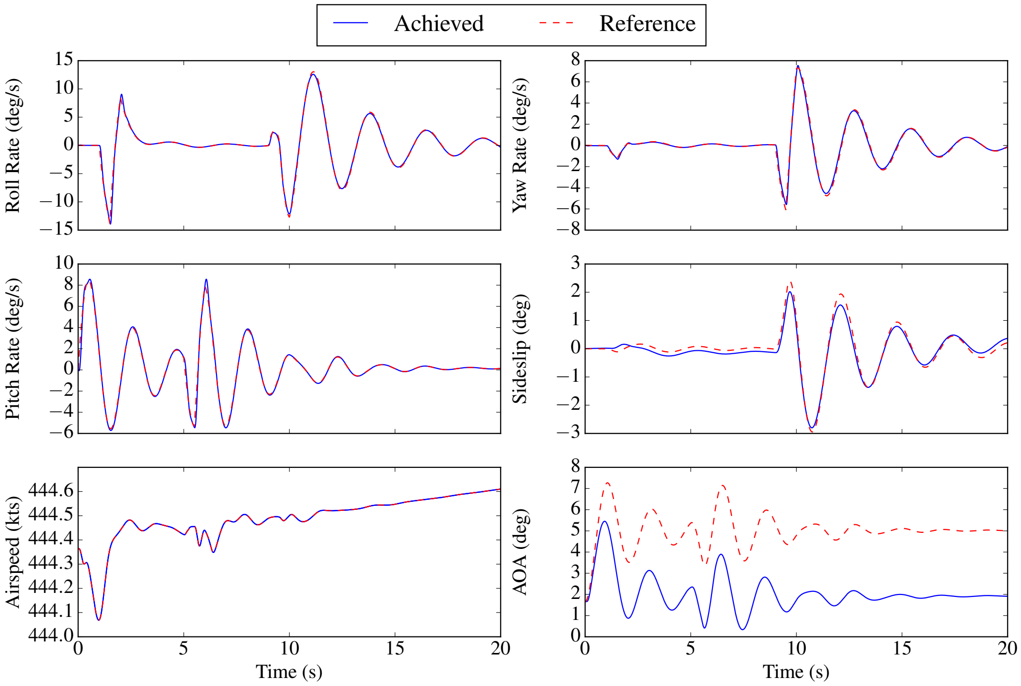
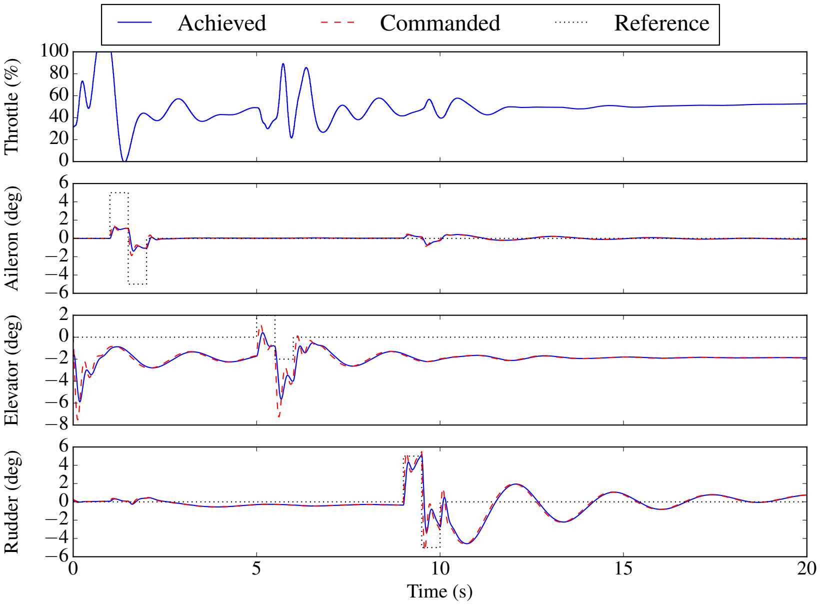


Figure 7.14: Nonlinear F-4 Phantom II $\mathcal{L}_1$ NDI adaptive parameter time histories.



(a) Actual and desired state time histories.



(b) Aerodynamic control effector time histories.

Figure 7.15: Nonlinear Generic Transport Model $\mathcal{L}_1$ NDI controller state and control time histories.

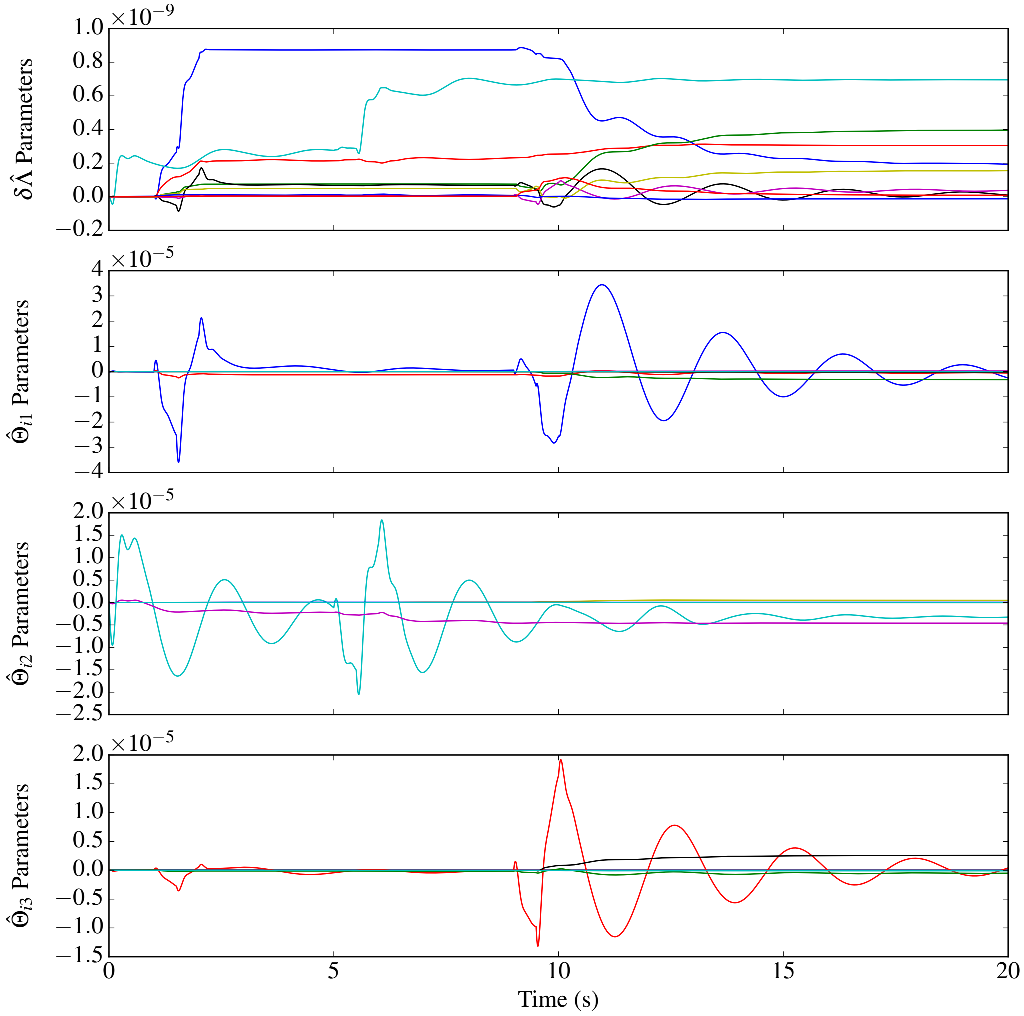


Figure 7.16: Nonlinear Generic Transport Model $\mathcal{L}_1$ NDI adaptive parameter time histories.

8. CONCLUSIONS

Two adaptive dynamic inversion control architectures are proposed in this thesis, one based on traditional model reference adaptive control and the other on the $\mathcal{L}_1$ adaptive control theory. Lyapunov stability theory is used with the projection operator and dead zone modifications to develop robust adaptive laws and prove stability of both schemes. These control architectures are applied to implement a basic command augmentation system similar to that in nature to those found on modern fighter aircraft, using a traditional design-to-flying qualities approach to generate desired dynamics based on military standards and an in-flight simulation system for linear and nonlinear reference models. The command augmentation system design is used as a baseline for comparison of the two architectures and evaluation of the robustness properties in the context of atmospheric disturbances, while the design of an IFS system suitable for implementation on a small uninhabited aerial system is the primary research objective of this thesis.

A high level design for such an IFS system is proposed, with focus on the inner loop control. This is accomplished using the two adaptive nonlinear dynamic inversion approaches. Schema for using both linear and nonlinear models as the reference model or desired dynamics for the in-flight simulation system are created. This approach is validated using numerous linear and nonlinear models spanning a broad range of configurations and aircraft types.

Controller designs are implemented in MATLAB/Simulink in continuous time. A nonlinear F-16 model serves as a surrogate unmanned system model; this model was chosen given the relative lack of high-fidelity models for small unmanned systems. A realistic environment model is created, and realistic actuator dynamics with position and (no-load) rate limits are used. Sensor noise and dynamics as well as computational delays are neglected and are considered outside the scope of this work.

The following conclusions are drawn based on the simulation studies conducted for this thesis:

1. Both the MRAC and $\mathcal{L}_1$ adaptive dynamic inversion approaches allow the use of dynamic inversion control with lower fidelity models. This is important for the small uninhabited aerial system application, as high-fidelity models are generally unavailable.

2. Both designs are robust in the presence of severe levels of turbulence and other atmospheric disturbances. The systems remain closed-loop stable and are able to track the desired dynamics, with maximum tracking errors of approximately $3\,^\circ/\mathrm{s}$ for the MRAC NDI approach and $12\,^\circ/\mathrm{s}$ for the $\mathcal{L}_1$ approach for a $180\,^\circ/\mathrm{s}$ roll rate command.

3. In-flight simulation of both linear and nonlinear models is achievable using these architectures. For both controllers the maximum tracking error is $8\,^\circ/\mathrm{s}$ while on average the errors are less than $3\,^\circ/\mathrm{s}$.

Finally, some remarks are presented:

- Successful in-flight simulation requires stable reference models, achievable either with the inherent dynamics of the reference plant or via a closed-loop reference. Additionally, reference commands must be reasonable in size for successful tracking, especially in the pitch axis. Practically, this translates to the need to limit the in-flight simulation envelope to a reduced subset of the simulator's flight envelope.

- For the $\mathcal{L}_1$ controller the higher tracking error is partially due to phase lag introduced by the $\mathcal{L}_1$ lowpass filtering structure. This lag can be reduced by increasing the bandwidth of the $\mathcal{L}_1$ lowpass filter subject to the available effector bandwidth.

While actuator demand is increased, control effector commands remain reasonable. High bandwidth increases performance while reducing the time delay margin of the $\mathcal{L}_1$ controller. The optimal design of the filter structure largely remains an open problem.

- Both methods (MRAC and $\mathcal{L}_1$) have advantages and disadvantages. Design of the MRAC controller is straightforward, while the $\mathcal{L}_1$ controller requirements more analysis that can be mathematically challenging.

- The $\mathcal{L}_1$ design offers greater robustness to disturbances than the MRAC design. Similar results are obtained for turbulence intensities corresponding to a probability of exceedance of 1×10^{-4}. For severe turbulence with a probability of exceedance of 1×10^{-5} the MRAC controller fails to meet the control objective while the $\mathcal{L}_1$ controller meets the objective with degraded performance. reduced robustness to disturbances and larger effector demands.

- An additional drawback of the continuous-time $\mathcal{L}_1$ approach results from the high-gain adaptation. This increases the requirements on the flight control hardware control hardware and can result in highly stiff differential equations posing problems for simulation.

Ultimately, the relative merits of both controllers depend on the actuation, choice of basis functions for the nonlinear adaptation, and tuning of controller parameters.

9. RECOMMENDATIONS

Significant work remains to take the controllers in this thesis from simulation to flight test. This section describes some of these issues and proposes future work to address shortcomings and transition to operational use. These are enumerated below:

1. The control law development and simulation studies conducted for this thesis assumes full-state feedback without sensor noise or dynamics. This is an unrealistic assumption, and evaluation of robustness of the controllers with sensed output measurements is required. In particular for the small unmanned aircraft application, critical states such as angle-of-attack and sideslip angle are generally not available and require estimation. The presence of sensor noise presents some simulation issues, as integration of noise is difficult numerically and results in greatly reduced simulation step sizes when using variable-step solvers, to the point of making simulations infeasible.

2. One way to address this issue is to move to discrete time implementations, whether by using approximation methods such as the Tustin transformation or pole-zero mapping or direct digital design methods for sampled data controllers. This is also a necessary step for implementation on flight hardware, as controllers are invariably implemented on digital computers. As part of this step some redesign of the controllers may be necessary to retain stability and robustness in the presence of sampling. The effect of digital sampling must be accounted for and the sampling time required for a stable discrete time implementation that meets the control objective must be determined. This required sampling time additionally drives hardware concerns such as the selection of sensors and flight control computers.

3. Implementation of the control laws on actual hardware also introduces delays due to computation and transit times between systems (flight control computer, sensors, actuators, etc.). The controller designs must be made robust to these delays, or hardware must be chosen such that the realized delay falls within the system time delay margin. For the in-flight simulation objective computational time for the reference models must also be considered. Linear state-space reference models are almost certainly fast enough for real-time simulation, but the nonlinear models may prove computationally intractable for the quality of flight control computers available for small unmanned aircraft. A candidate solution to this problem is the integration of a separate computer with more powerful hardware for implementation of the variable stability system, although this should be avoided if possible.

4. An envelope protection or run-time assurance system must be provided to ensure that the vehicle remains in a safe part of the flight envelope at all times. This should ensure that the aircraft is prevented from stalling or exceeding structural limits. A combination of an envelope protection system to scale and limit commands and a run-time assurance system to verify that the VSS functions correctly is the most obvious solution. The run-time assurance system should monitor the system for nominal operation and be able to automatically "trip" into a fallback mode if the VSS fails, effectors fail, limits are exceeded, or telemetry/command links are lost. Additionally, the system should ensure that the safety pilot is able to bypass the variable stability system if needed at any time.

5. Finally, the complete system design must be transitioned to flight. This requires selection of a SUAS platform for implementation, which should be low-cost and provide sufficient payload capacity for integration of the VSS systems. Implementation of the control laws required modeling of the simulator vehicle aerodynamics and

propulsion for the baseline dynamic inversion controller's onboard model. Adaptation reduces the effort required allowing for low-order parametric models obtained via methods such as vortex lattice codes or empirical methods such as DATCOM. However, higher fidelity modeling should be considered for creation of a "truth" model for flight simulation to support pilot training and analysis of potential control systems. If simulation of throttle and engine response is desired, modeling of the simulator's engine is required. The ultimate VSS design should be implemented in MATLAB/Simulink and flight software should be autocoded to the target platform. This allows high quality flight software without requiring extensive embedded software design skills. Lastly, modification of a small radio control aircraft to allow direct lift and sideforce generation would prove interesting for implementation of a full six-degree-of-freedom in-flight simulator.

REFERENCES

[1] Barry Jr., J. and Schelhorn, A. E., "VISTA – A Modest Proposal for a New Fighter In-Flight Simulator," *AIAA 22nd Aerospace Sciences Meeting*, AIAA, Reno, NV, Jan. 1984, doi:10.2514/6.1984-520.

[2] Weingarten, N. C., "History of In-Flight Simulation at General Dynamics," *Journal of Aircraft*, Vol. 42, No. 2, March–April 2005, pp. 290–298, doi:10.2514/1.4663.

[3] Valasek, J., "ConSenT – An Unmanned Air Vehicle (UAV)," Request for Proposal, Vehicle Systems & Control Laboratory, Department of Aerospace Engineering, Texas A&M University. Unpublished.

[4] Kauffman, W. M., Liddell Jr., C. J., Smith, A., and Dyke Jr., R. D. V., "An Apparatus for Varying Effective Dihedral in Flight with Application to a Study of Tolerable Dihedral on a Conventional Fighter Airplane," NACA Technical Report 948, Ames Aeronautical Laboratory, National Advisory Committee for Aeronautics, Moffett Field, CA, Jan. 1949.

[5] U.S. Air Force, "Lockheed NT-33A," URL `http://www.af.mil/News/Photos.aspx?igphoto=2001299068`, Online image; accessed 2016-12-29.

[6] Hall, G. W. and Huber, R. W., "System Description and Performance Data for the USAF/CAL Variable Stability T-33 Airplane," CAL Report No. BM-2821-F-2, Cornell Aeronautical Laboratory, Buffalo, NY, July 1970.

[7] Kidd, E. A., Bull, G., and Harper Jr., R. P., "In-Flight Simulation – Theory and Application," AGARD 368, NATO Advisory Group For Aeronautical Research and Development, Paris, France, April 1961.

[8] Szalai, K. J., "The Influence of Response Feedback Loops on the Lateral-Directional Dynamics of A Variable-Stability Transport Aircraft," NASA TN D-3966, NASA

Flight Research Center, Edwards, CA, May 1967.

[9] Reynolds, P. A., Wasserman, R., Fabian, G. J., and Motyka, P. R., "Capabilities of the Total In-Flight Simulator (TIFS)," AFFDL-TR 72-39, Cornell Aeronautical Laboratory, Buffalo, NY, July 1972.

[10] Zapka, B., "Real-world projects prepare students for future flight test," URL `http://www.edwards.af.mil/News/Photos/igphoto/2000087833`, Dec. 2012, Online image; accessed 2016-12-13.

[11] Hellmann, G. K., Frearson, D. E., and Jr., J. B., "VISTA/F16 – The Next High Performance In-flight Simulator," *Flight Simulation Technologies Conference*, AIAA, Atlanta, GA, Sept. 1988, pp. 198–205, doi:10.2514/6.1988-4610.

[12] Lindak, J. A. and McCarley, Z. A., "Variable Stability Learjets – First Flights of an Old Airplane with a New VSS," *AIAA Flight Testing Conference, AIAA AVIATION Forum*, Washington, D.C., June 2016, doi:10.2514/6.2016-3359.

[13] Bronder, T. J., Helms, N. J., Heinz, D. A., Lee, J., Mangini, M., and Keeble, M., "Initial testing and development of an organic remote pilot aircraft capability at the USAF Test Pilot School," *28th Aerodynamic Measurement Tecnology, Ground Testing, and Flight Testing Conference*, New Orleans, LA, June 2012, doi:10.2514/6.2012-3082.

[14] Cotting, M. C., Jeffrey, R. B., Blackstun, M. P., Fulkerson, T. P., Lau, T. R., Stephens, S. S., Ackerman, K. A., Xargay, E., Choe, R., and Hovakimyan, N., "'Can I Get $\mathcal{L}_1$ On?!' Providing Consistent Handling Qualities on Calspan's Variable-Stability Learjet," *AIAA Guidance, Navigation, and Control Conference, AIAA SciTech Forum*, San Diego, CA, June 2016, doi:10.2514/6.2016-0632.

[15] Ackerman, K. A., Xargay, E., Choe, R., Hovakimyan, N., Cotting, M. C., Jeffrey, R. B., Blackstun, M. P., Fulkerson, T. P., Lau, T. R., and Stephens, S. S., "$\mathcal{L}_1$ Stability Augmentation System for Calspan's Variable-Stability Learjet," *AIAA Guidance,*

Navigation, and Control Conference, AIAA SciTech Forum, San Diego, CA, Jan. 2016, doi:10.2514/6.2016-0631.

[16] Germann, K. P., *T-6A Texan II In-Flight Simulation and Variable Stability System Design*, Ph.D. dissertation, Mississippi State University, Starkville, MI, Dec. 2006.

[17] Cox, T. H. and Cotting, M. C., "A Generic Inner-Loop Control Law Structure for Six-Degree-of-Freedom Conceptual Aircraft Design," *AIAA Aerospace Sciences Meeting and Exhibit*, No. AIAA 2005-0031, Reno, NV, Jan. 2005, doi:10.2514/6.2005-31.

[18] Åström, K. J. and Wittenmark, B., *Adaptive Control*, Dover Publications, Inc., Mineola, NY, 2nd ed., 2008.

[19] Ioannou, P. A. and Sun, J., *Robust Adaptive Control*, Dover, Dover Reprint ed., 2012.

[20] Ito, D., Georgie, J., Valasek, J., and Ward, D. T., "Reentry Vehicle Flight Controls Design Guidelines: Dynamic Inversion," NASA TP 2002–210771, NASA, March 2002.

[21] Reiner, J., Balas, G. J., and Garrard, W. L., "Robust Dynamic Inversion for Control of Highly Maneuverable Aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 1, Jan.-Feb. 1995, pp. 18–24, doi:10.2514/3.56651.

[22] Lavretsky, E. and Wise, K. A., *Robust and Adaptive Control with Aerospace Applications*, Advanced Textbooks in Control and Signal Processing, Springer-Verlag, London, UK, 2013.

[23] Rollins, E., *Nonlinear Adaptive Dynamic Inversion Control for Hypersonic Vehicles*, Ph.D. dissertation, Texas A&M University, College Station, TX, 2013.

[24] Famularo, D., *Observer-based Nonlinear Dynamic Inversion Adaptive Control with State Constraints*, Ph.D. dissertation, Texas A&M University, College Station, TX, 2017.

[25] Subbarao, K., *Structured Adaptive Model Inversion: Theory and Applications to Trajectory Tracking for Non-Linear Dynamical Systems*, Ph.D. dissertation, Texas A&M

University, College Station, TX, 2001.

[26] Subbarao, K. and Junkins, J. L., "Structured Model Reference Adaptive Control for a Class of Nonlinear Systems," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 4, July–Aug. 2003, pp. 551–557, doi:10.2514/2.5104.

[27] Tandale, M. and Valasek, J., "Fault-Tolerant Structured Adaptive Model Inversion Control," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 3, May 2006, pp. 635–642, doi:10.2514/1.15244.

[28] Valasek, J., Famularo, D., and Marwaha, M., "Fault-Tolerant Adaptive Model Inversion Control for Vision-Based Autonomous Air Refueling," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 6, Feb. 2017, pp. 1336–1347, do:10.2514/1.G001888.

[29] Rysdyk, R. T. and Calise, A. J., "Fault Tolerant Flight Control via Adaptive Neural Network Augmentation," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Boston, MA, 1998, doi:10.2514/6.1998-4483.

[30] Johnson, E. N. and Calise, A. J., "A Six Degree-of-Freedom Adaptive Flight Control Architecture for Trajectory Following," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, No. AIAA-2002-4776, Monterey, CA, Aug. 2002, doi:10.2514/6.2002-4776.

[31] Stevens, B. L., Lewis, F. L., and Johnson, E. N., *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, John Wiley & Sons, Inc., Hoboken, NJ, 3rd ed., 2016.

[32] Leman, T., Xargay, E., Dullerud, G., and Hovakimyan, N., "$\mathcal{L}_1$ Adaptive Control Augmentation System for the X-48B Aircraft," *AIAA Guidance, Navigation, and Control Conference*, Chicago, IL, Aug. 2009, doi:10.2514/6.2009-5619.

[33] Peter, F., Holzapfel, F., Xargay, E., and Hovakimyan, N., "L1 Adaptive Augmentation of a Missile Autopilot," *AIAA Guidance, Navigation, and Control Conference*,

Minneapolis, MN, Aug. 2012, doi:10.2514/6.2012-4832.

[34] Campbell, S. F. and Kaneshige, J. T., "A Nonlinear Dynamic Inversion $\mathcal{L}_1$ Adaptive Controller for a Generic Transport Model," *Proceedings of the 2010 American Control Conference*, June 2010, pp. 862–867, 10.1109/ACC.2010.5530704.

[35] McRuer, D. T., Graham, D., and Ashkenas, I., *Aircraft Dynamics and Automatic Control*, Princeton Legacy Library, Princeton University Press, Princeton, NJ, 2014.

[36] Roskam, J., *Airplane Flight Dynamics & Automatic Flight Controls: Part I*, DARCorporation, Jan. 2001.

[37] Schmidt, D. K., *Modern Flight Dynamics*, McGraw-Hill, New York, NY, 1st ed., 2010.

[38] Schaub, H. and Junkins, J. L., *Analytical Mechanics of Space Systems*, American Institute of Aeronautics and Astronautics, Reston, VA, 2nd ed., 2009.

[39] Hurtado, J. E., *Kinematic and Kinetic Principles*, Lulu Press, Inc., Raleigh, NC, 2012.

[40] Klein, V. and Morelli, E. A., *Aircraft System Identification – Theory and Practice*, AIAA Education Series, AIAA, Reston, VA, 2006.

[41] Iliff, K. W. and Maine, R. E., "Practical Aspects of Using a Maximum Likelihood Estimation Method to Extract Stability and Control Derivatives from Flight Data," NASA TN D-8209, NASA Dryden Flight Research Center, April 1976.

[42] Iliff, K. W., "Parameter Estimation for Flight Vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 12, No. 5, Sept.–Oct. 1989, pp. 609–622, doi:10.2514/3.20454.

[43] Grauer, J. A. and Morelli, E. A., "A Generic Nonlinear Aerodynamic Model for Aircraft," *AIAA Atmospheric Flight Mechanics Conference*, No. AIAA 2014-0542, National Harbor, MD, 2014, doi:10.2514/6.2014-0542.

[44] Balmer, G. R., *Modelling and Control of a Fixed-wing UAV for Landings on Mobile*

Landing Platforms, Master's thesis, KTH Royal Institute of Technology, Stockholm, Sweden, Oct. 2015.

[45] Nguyen, L. T., Ogburn, M. E., Gilbert, W. P., Kibler, K. S., Brown, P. W., and Deal, P. L., "Simulator Study of a Stall/Post-Stall Characteristics of a Fighter Airplane With Relaxed Longitudinal Static Stability," TP 1538, NASA, Hampton, VA, Dec. 1979.

[46] Ioannou, P. and Fidan, B., *Adaptive Control Tutorial*, No. 11 in Advances in Design and Control, SIAM, Philadelphia, PA, 2006.

[47] "Military Specification: Flying Qualities of Piloted Airplanes," MIL-F-8785C, Nov. 1980.

[48] Georgie, J., *Selection of Desired Dynamics for Dynamic Inversion Controlled Re-Entry Vehicles*, MS thesis, Texas A&M University, College Station, TX, Aug. 2001.

[49] Enns, D. F. et al., "Application of Multivariable Control Theory to Aircraft Control Laws," Tech. Rep. WL-TR-96-3099, Flight Dynamics Directorate, USAF Wright Laboratory, Wright-Patterson AFB, OH, May 1996.

[50] "Military Standard: Flying Qualities of Piloted Aircraft," MIL-STD-1797B, Feb. 2006.

[51] Durham, W., Bordignon, K. A., and Beck, R., *Aircraft Control Allocation*, John Wiley & Sons, Ltd., Chichester, UK, 2017.

[52] Valasek, J., "Block 9: Flight Control Systems," AERO 445: Vehicle Management Systems class notes. Unpublished.

[53] Enns, D., Bugajski, D., Hendrick, R., and Stein, G., "Dynamic inversion: an evolving methodology for flight control design," *International Journal of Control*, Vol. 59, No. 1, Nov. 1994, pp. 71–91, doi:10.1080/00207179408923070.

[54] Bugajski, D. J. and Enns, D. F., "Nonlinear Control Law with Application to High Angle-of-Attack Flight," *Journal of Guidance, Control, and Dynamics*, Vol. 15,

No. 3, May–June 1992, pp. 761–767, doi:10.2514/3.20902.

[55] Bordignon, K. and Bessolo, J., "Control Allocation for the X-35B," *2002 Biennial International Powered Lift Conference and Exhibit*, AIAA, Williamsburg, VA, Nov. 2002, doi:10.2514/6.2002-6020.

[56] Balas, G. J., "Flight Control Law Design: An Industry Perspective," *European Journal of Control*, Vol. 9, No. 2, 2003, pp. 207–226, doi:10.3166/ejc.9.207-226.

[57] Khalil, H. K., *Nonlinear Systems*, Prentice Hall, Upper Saddle River, NJ, 3rd ed., 2002.

[58] Slotine, J.-J. E. and Li, W., *Applied Nonlinear Control*, Prentice Hall, Englewood Cliffs, NJ, 1991.

[59] Georgie, J. and Valasek, J., "Evaluation of Longitudinal Desired Dynamics for Dynamic-Inversion Controlled Generic Reentry Vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 5, Sept.-Oct. 2003, pp. 811–819, doi:10.2514/2.5116.

[60] Miller, C. J., "Nonlinear Dynamic Inversion Baseline Control Law: Architecture and Performance Predictions," *AIAA Guidance, Navigation, and Control Conference*, No. AIAA 2011-6467, Portland, OR, Aug. 2011, doi:10.2514/6.2011-6467.

[61] Lewis, F. L., *Applied Optimal Control and Estimation*, Prentice Hall, 1992.

[62] Dorato, P., Abdallah, C. T., and Cerone, V., *Linear Quadratic Control: An Introduction*, Krieger Publishing Company, 2000, Reprint edition with corrections.

[63] Boskovich, B. and Kaufmann, R. E., "Evolution of the Honeywell First-Generation Adaptive Autopilot and Its Applications to F-94, F-101, X-15, and X-20 Vehicles," *Journal of Aircraft*, Vol. 3, No. 4, July–Aug. 1966, pp. 296–304, doi:10.2514/3.43739.

[64] "Experience with the X-15 Adaptive Flight Control System," NASA TN D-6208, NASA Flight Research Center, Edwards, CA, March 1971.

[65] Rynaski, E., *Applications of Adaptive Control*, chap. Adaptive Control Application to Aircraft, Academic Press, Inc., New York, NY, 1980, pp. 245–267.

[66] Cao, C. and Hovakimyan, N., "Design and Analysis of a Novel $\mathcal{L}_1$ Adaptive Controller, Part I: Control Signal and Asymptotic Stability," *2006 American Control Conference*, Minneapolis, MN, June 2006, pp. 3397–3402, doi:10.1109/ACC.2006.1657243.

[67] Cao, C. and Hovakimyan, N., "Design and Analysis of a Novel $\mathcal{L}_1$ Adaptive Controller, Part II: Guaranteed Transient Performance," *2006 American Control Conference*, Minneapolis, MN, June 2006, pp. 3403–3408, doi:10.1109/ACC.2006.1657244.

[68] Akella, M. R., *Structured Adaptive Control: Theory and Applications to Trajectory Tracking in Aerospace Systems*, Ph.D. dissertation, Texas A&M University, College Station, TX, Dec. 1998.

[69] Hovakimyan, N. and Cao, C., *$\mathcal{L}_1$ Adaptive Control Theory – Guaranteed Robustness with Fast Adaptation*, No. 21 in Advances in Design and Control, SIAM, Philadelphia, PA, 2010.

[70] Lavretsky, E., Gibson, T. E., and Annaswamy, A. M., "Projection Operator in Adaptive Systems," Oct. 2012, Eprint: arXiv:1112.4232v6.

[71] Kautsky, J., Nichols, N., and Van Dooren, P., "Robust pole assignment in linear state feedback," *International Journal of Control*, Vol. 41, No. 5, 1985, pp. 1129–1155, doi:10.1080/0020718508961188.

[72] Player, J. and Gingras, D. R., "Rapid Simulation Development for Evaluation of Conceptual Unmanned Aerial Vehicles," *AIAA Modeling and Simulation Technologies Conference and Exhibit*, AIAA, Providence, RI, Aug. 2004, doi:10.2514/6.2004-5042.

[73] Famularo, D., Valasek, J., Muse, J. A., and Bolender, M. A., "Observer-Based

Feedback Adaptive Control for Nonlinear Hypersonic Vehicles," *AIAA Guidance, Navigation, and Control Conference, AIAA SciTech Forum*, No. AIAA-2017-1492, Grapevine, TX, Jan. 2017, doi:10.2514/6.2017-1492.

[74] Dunbar, M. V. and Dargan, J. L., "VISTA/F-16 Design Features," *Flight Simulation Technologies Conference and Exhibit*, No. AIAA-1989-3330, Boston, MA, 1989, doi:10.2514/6.1989-3330.

[75] Murch, A. M., "A Flight Control System Architecture for the NASA AirSTAR Flight Test Infrastructure," *AIAA Guidance, Navigation and Control Conference and Exhibit*, Honolulu, HI, Aug. 2008, doi:10.2514/6.2008-6990.

[76] Lu, H.-H., Rogers, C. T., Goecks, V. G., and Valasek, J., "Online Near Real Time System Identification on a Fixed-Wing Small Unmanned Air Vehicle," *AIAA Atmospheric Flight Mechanics Conference*, 2018, (Accepted; to appear.).

[77] Cunningham, K., Cox, D., Murri, D., and Riddick, S., "A Piloted Evaluation of Damage Accommodating Flight Control Using a Remotely Piloted Vehicle," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, No. AIAA 2011-6451, Portland, OR, Aug. 2011, doi:10.2514/6.2011-6451.

APPENDIX A

ADDITIONAL IN-FLIGHT SIMULATION RESULTS

This appendix provides some additional simulation results for linear and nonlinear in-flight simulation models. The majority of these results are presented without comment, with the exception of examples with features of particular interest. The primary motivation for this appendix is the demonstration of in-flight simulation controller applicability to a wide-variety of reference models. Model data is provided in App. B.

A.1 Linear In-Flight Simulation

Here some additional results for in-flight simulation of linear models are presented using both the MRAC and $\mathcal{L}_1$ adaptive dynamic inversion control architectures.

A.1.1 Rockwell Commander 700

The Commander 700 is a small, twin engined general aviation aircraft, and provides another dissimilar configuration and operating range for the baseline F-16 to simulate.

A.1.1.1 MRAC NDI Controller

Figure A.1a presents state time histories, Fig. A.1b presents control effector time histories, and Fig. A.2 plots the variation of the adaptive parameters. Note good tracking of all reference states. Commanded controls meet position limits but saturate rate limits, especially in the pitch channel when trying to match the open-loop reference model pitch rate response.

A.1.1.2 $\mathcal{L}_1$ NDI Controller

Results for the $\mathcal{L}_1$ controller are presented in Figs. A.3a–A.4, which present time histories of states, controls, and adaptive parameters, respectively. Performance is similar to the MRAC NDI controller, although the $\mathcal{L}_1$ controller design exhibits more phase lag in tracking roll rate with the current gain set. Additionally, the $\mathcal{L}_1$ controller demands lower effector rates.

A.1.2 Cessna T-37 Tweet Model

The Cessna T-37 is a small primary flight trainer powered by two turbojet engines.

A.1.2.1 MRAC NDI Controller

State time histories are shown in Fig. A.5a, demonstrating that the controller successfully meets the tracking objective. Note slight bias in the pitch axis, manifested in pitch rate and AOA. Control effector time histories are plotted in Fig. A.5b; note high frequency elevator demand corresponding to the pitch perturbations of the reference model. Variation of adaptive parameters is shown in Fig. A.6.

A.1.2.2 $\mathcal{L}_1$ NDI Controller

Figures A.7a–A.8 present time histories for the $\mathcal{L}_1$ adaptive dynamic inversion controller applied to the T-37 IFS example. Performance is comparable to the MRAC NDI controller. As with other examples, the $\mathcal{L}_1$ filtering structure results in both attentuation and phase lag compared to the MRAC NDI control law but also less demand placed on the actuators in terms of both position and rate.

A.1.3 Boeing 747 Model

The Boeing 747 is a well-known wide body commercial air transport, and serves as another dissimilar aircraft model for simulation.

A.1.3.1 MRAC NDI Controller

Figures A.9a–A.10 present time histories of states, controls, and adaptive parameters, respectively for the B747 using the MRAC NDI control law. Observing the figures, the IFS tracking objective is successfully met.

A.1.3.2 $\mathcal{L}_1$ NDI Controller

Results for the $\mathcal{L}_1$ adaptive dynamic inversion controller are plotted in Fig. A.11a for states, Fig. A.11b for effectors, and Fig. A.12 for adaptive parameters. The $\mathcal{L}_1$ controller successfully captures the reference dynamics, but exhibits slight attenuation and phase lag, particularly in the lateral/directional axes.

A.2 Nonlinear In-Flight Simulation

Two additional results for in-flight simulations using nonlinear models are presented. Controllers are implemented using both the MRAC and $\mathcal{L}_1$ adaptive dynamic inversion control architectures.

A.2.1 Convair F-106 Delta Dart

The F-106 is a delta-winged interceptor aircraft, approximately $10{,}000\,\mathrm{lbf}$ heavier than the F-16 and with larger inertias.

A.2.1.1 MRAC NDI Controller

Results for the MRAC adaptive dynamic inversion controller are plotted in Fig. A.13a for states, Fig. A.13b for effectors, and Fig. A.14 for adaptive parameters. The controller achieves good tracking of the angular rates and captures the behavior of the AOA and sideslip.

A.2.1.2 $\mathcal{L}_1$ NDI Controller

Figures A.13a–A.14 plots the results for the $\mathcal{L}_1$ adaptive dynamic inversion control system. Like the MRAC NDI controller, good tracking of the angular rates is achieved.

A.2.2 Rockwell/MBB X-31

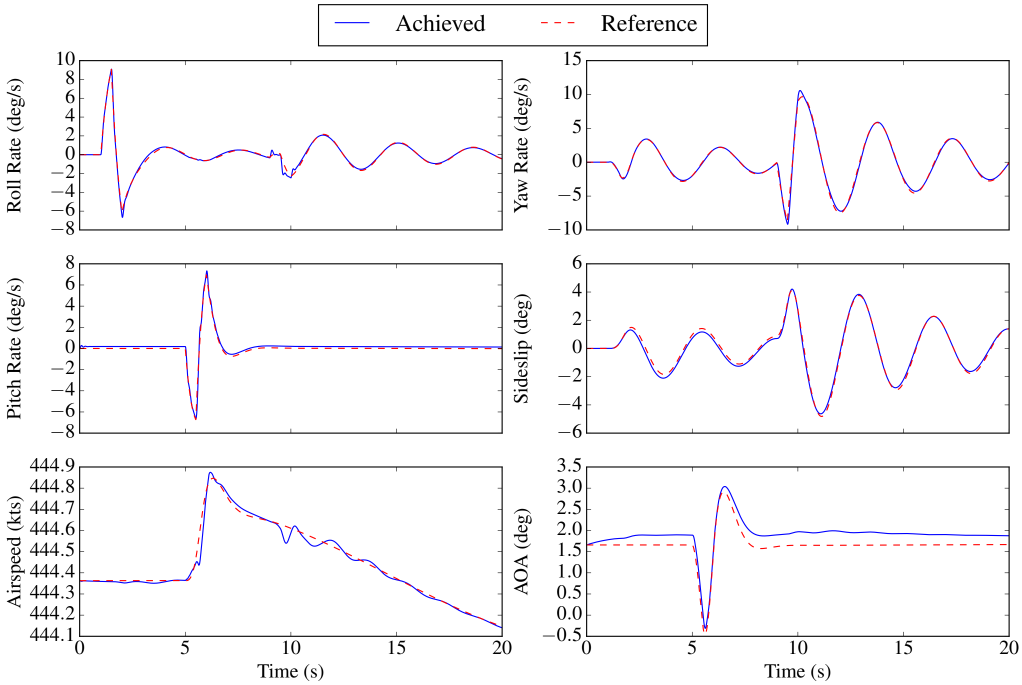
The Rockwell/MBB X-31 is an experimental test platform for supermaneuverability research.

A.2.2.1 MRAC NDI Controller

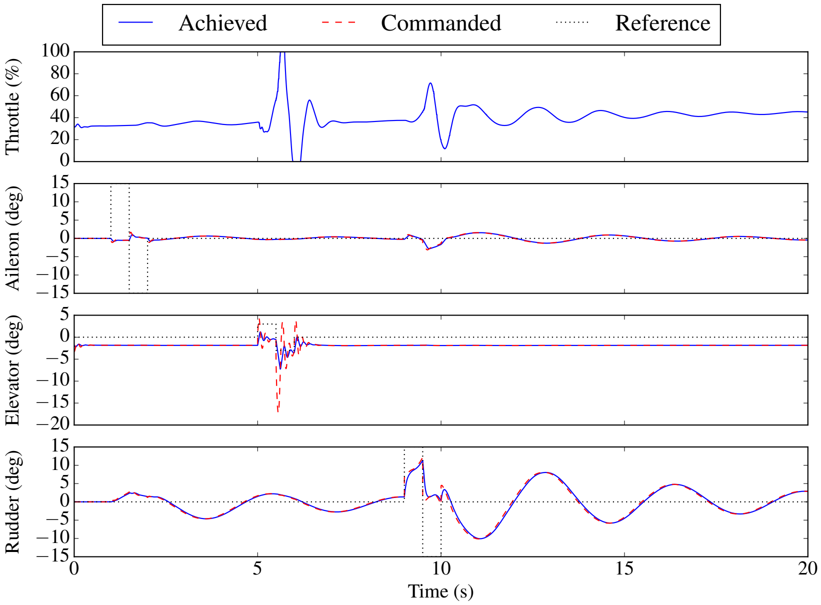
The X-31 simulation results for the MRAC NDI controller are presented in Figs. A.17a–A.18. Near perfect tracking of the states is achieved.

A.2.2.2 $\mathcal{L}_1$ NDI Controller

Figure A.19a plots states for the $\mathcal{L}_1$ controller simulation of the X-31, while Fig. A.19b plots effectors and Fig. A.20 plots adaptive parameters. Good tracking is achieved, and performance is comparable to the MRAC NDI approach.



(a) Actual and desired state time histories.



(b) Aerodynamic control effector time histories.

Figure A.1: Rockwell Commander 700 NRAC NDI controller state and control time histories.

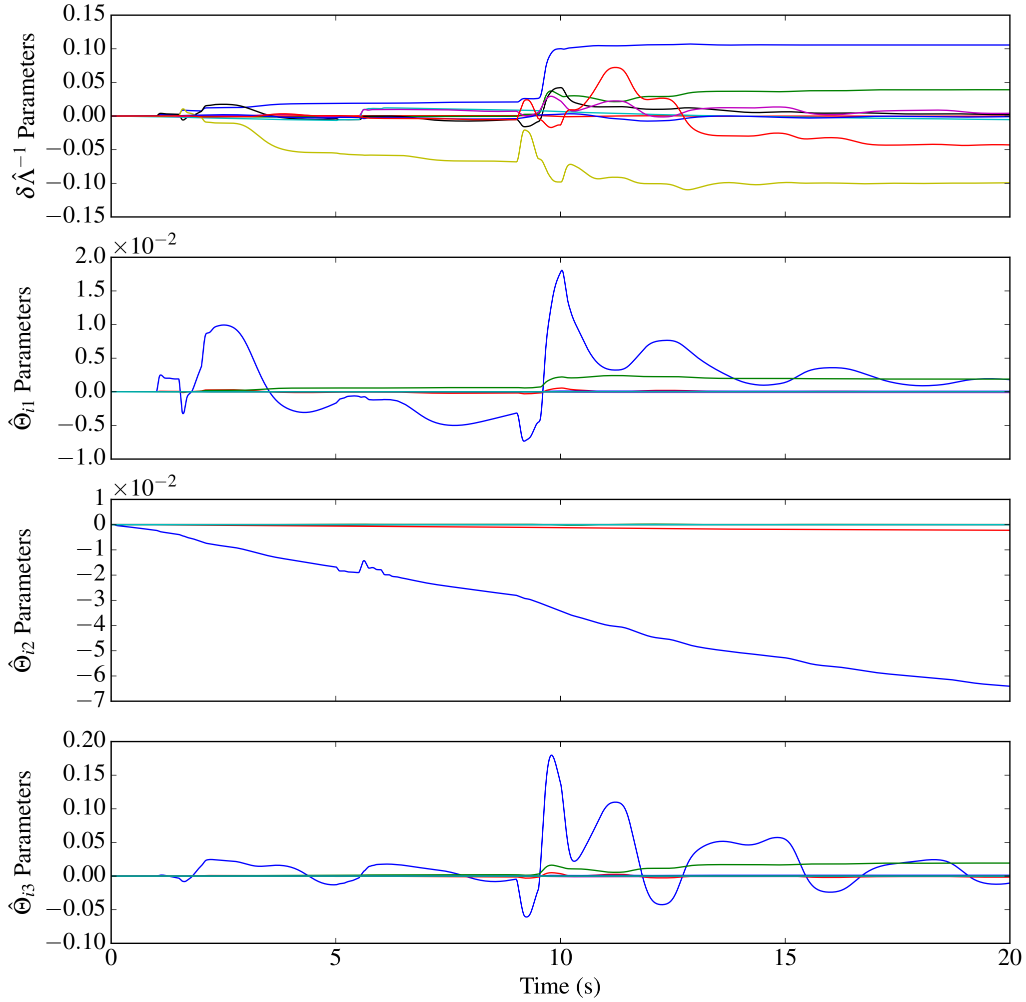
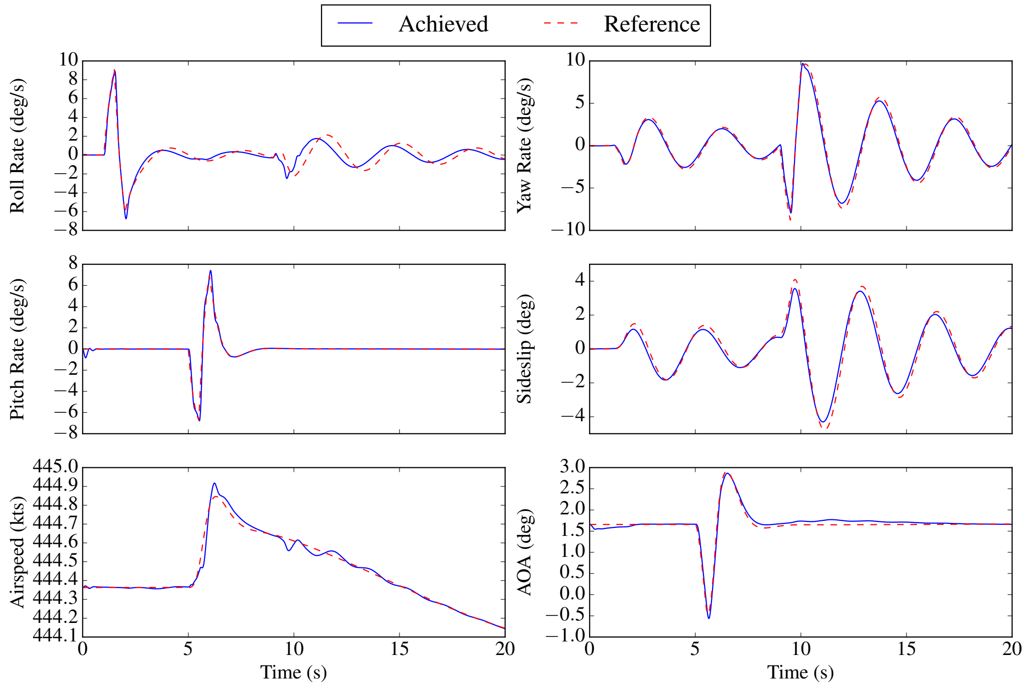
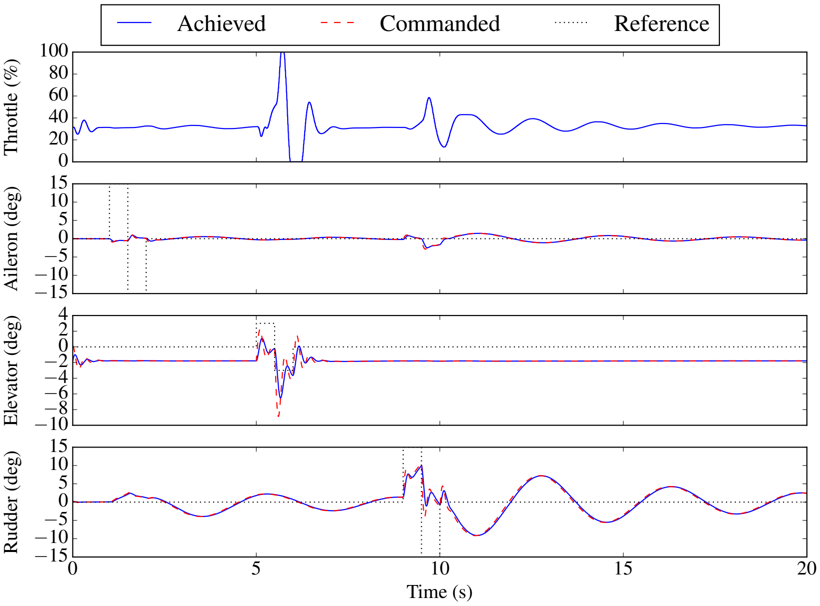


Figure A.2: Rockwell Commander 700 NRAC NDI adaptive parameter time histories.



(a) Actual and desired state time histories.



(b) Aerodynamic control effector time histories.

Figure A.3: Rockwell Commander 700 $\mathcal{L}_1$ NDI controller state and control time histories.

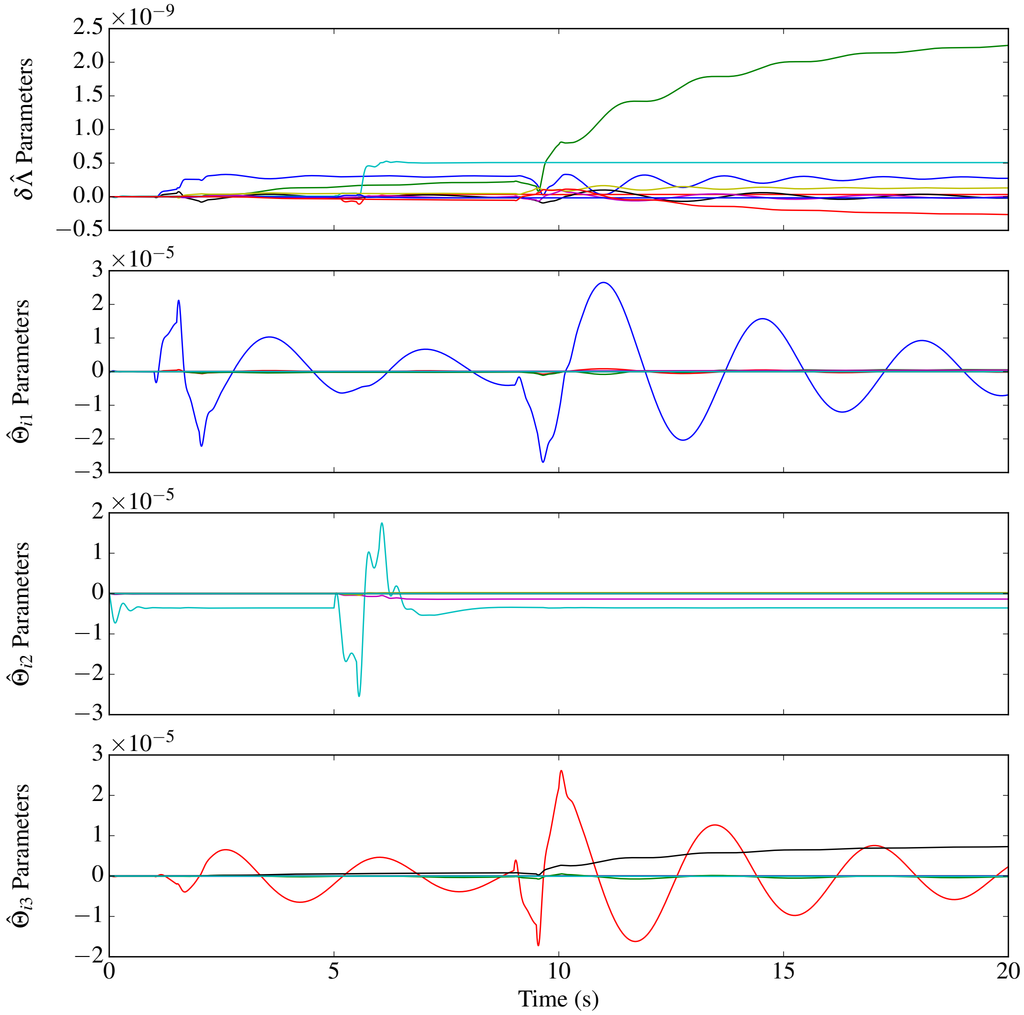
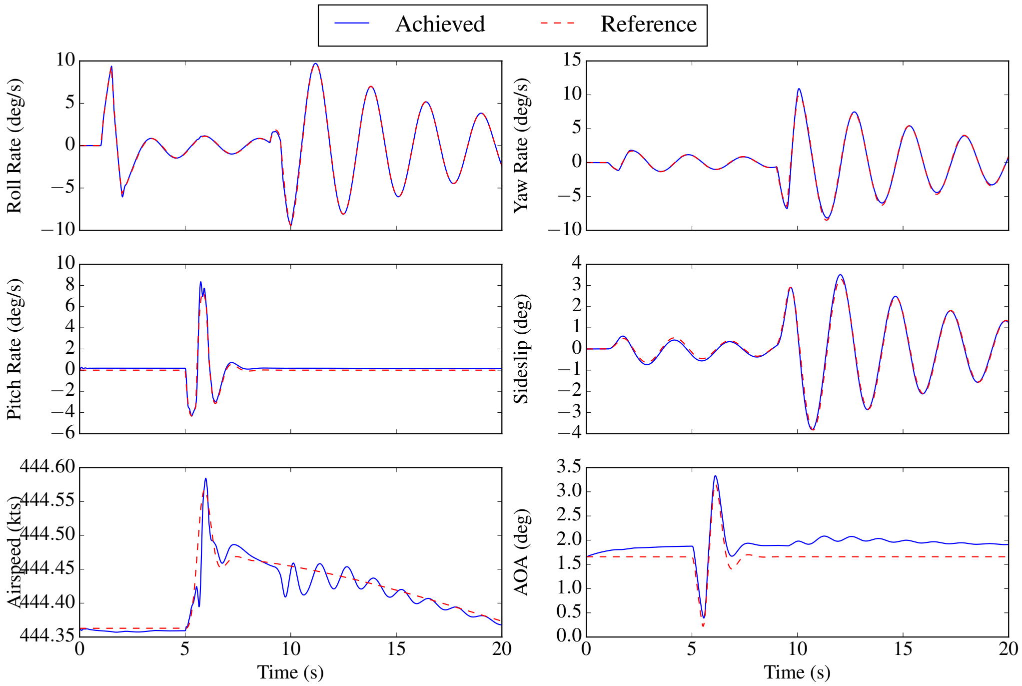
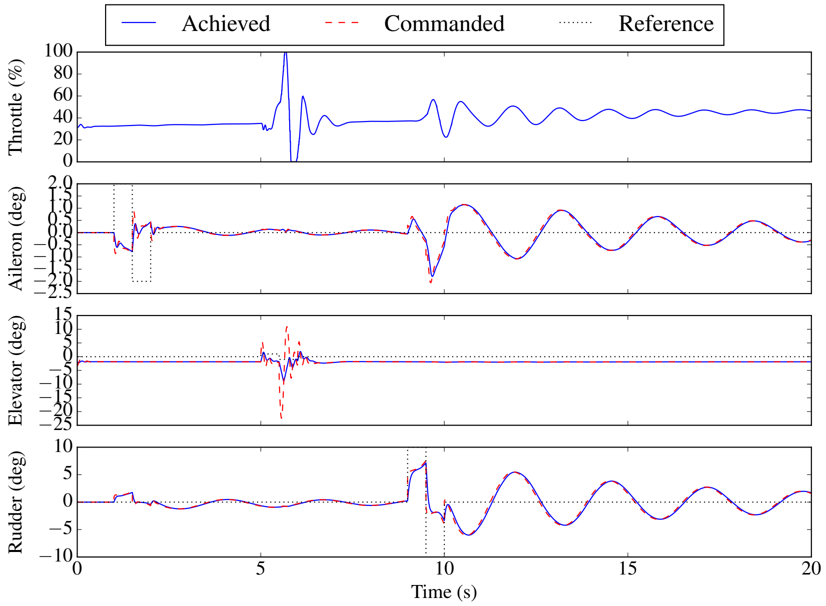


Figure A.4: Rockwell Commander 700 $\mathcal{L}_1$ NDI adaptive parameter time histories.



(a) Actual and desired state time histories.



(b) Aerodynamic control effector time histories.

Figure A.5: Cessna T-37 Tweet MRAC NDI controller state and control time histories.

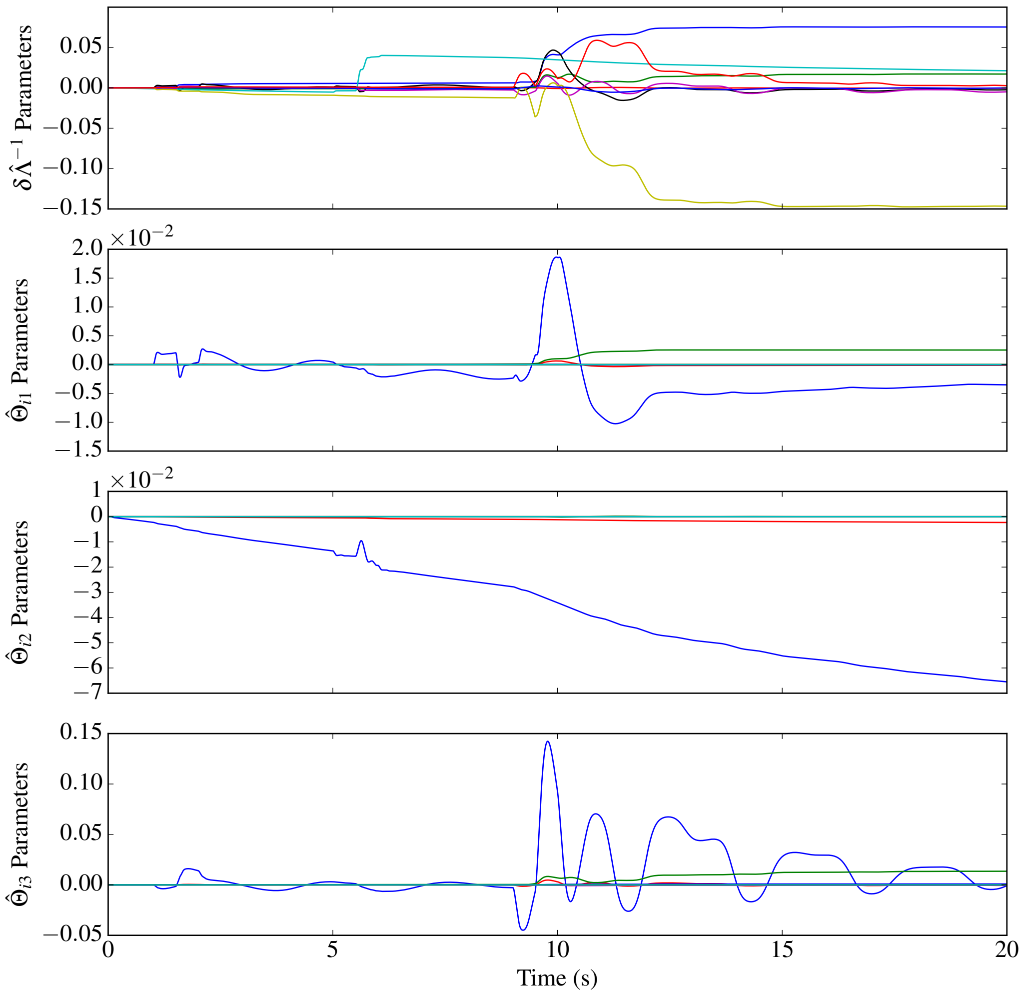
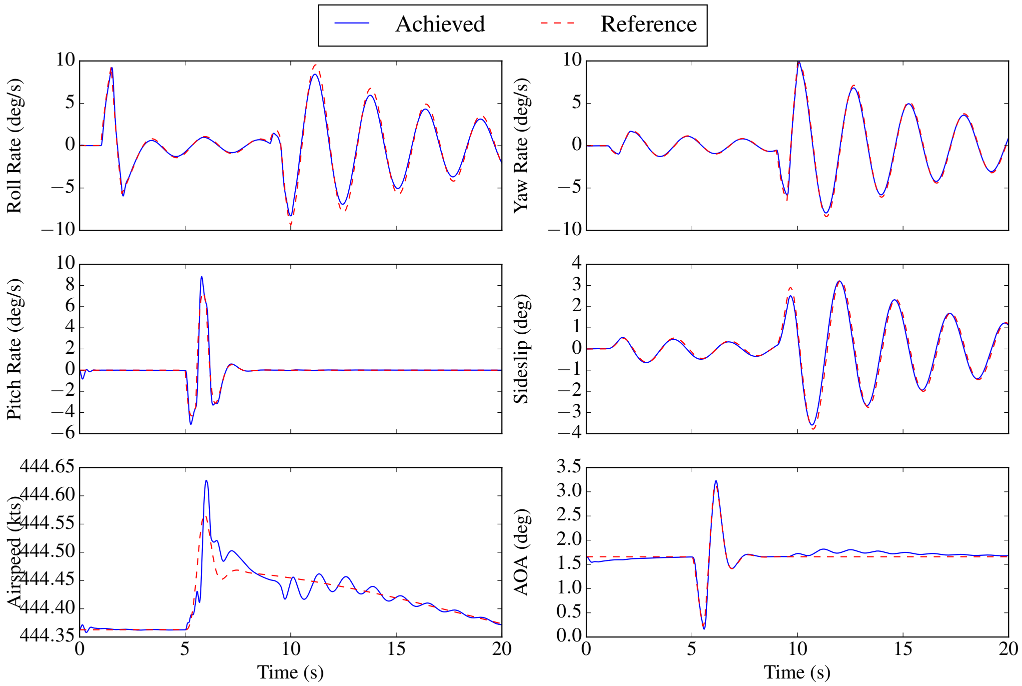
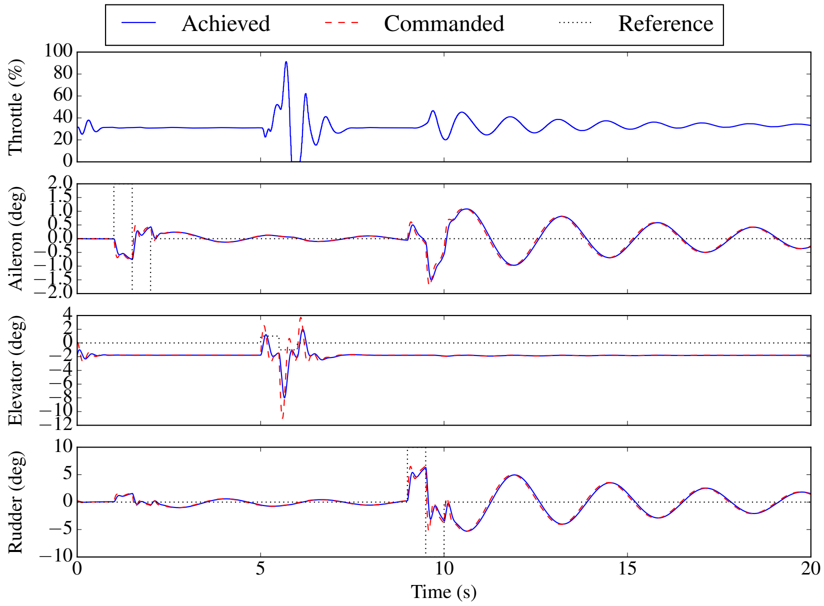


Figure A.6: Cessna T-37 Tweet MRAC NDI adaptive parameter time histories.



(a) Actual and desired state time histories.



(b) Aerodynamic control effector time histories.

Figure A.7: Cessna T-37 Tweet $\mathcal{L}_1$ NDI controller state and control time histories.

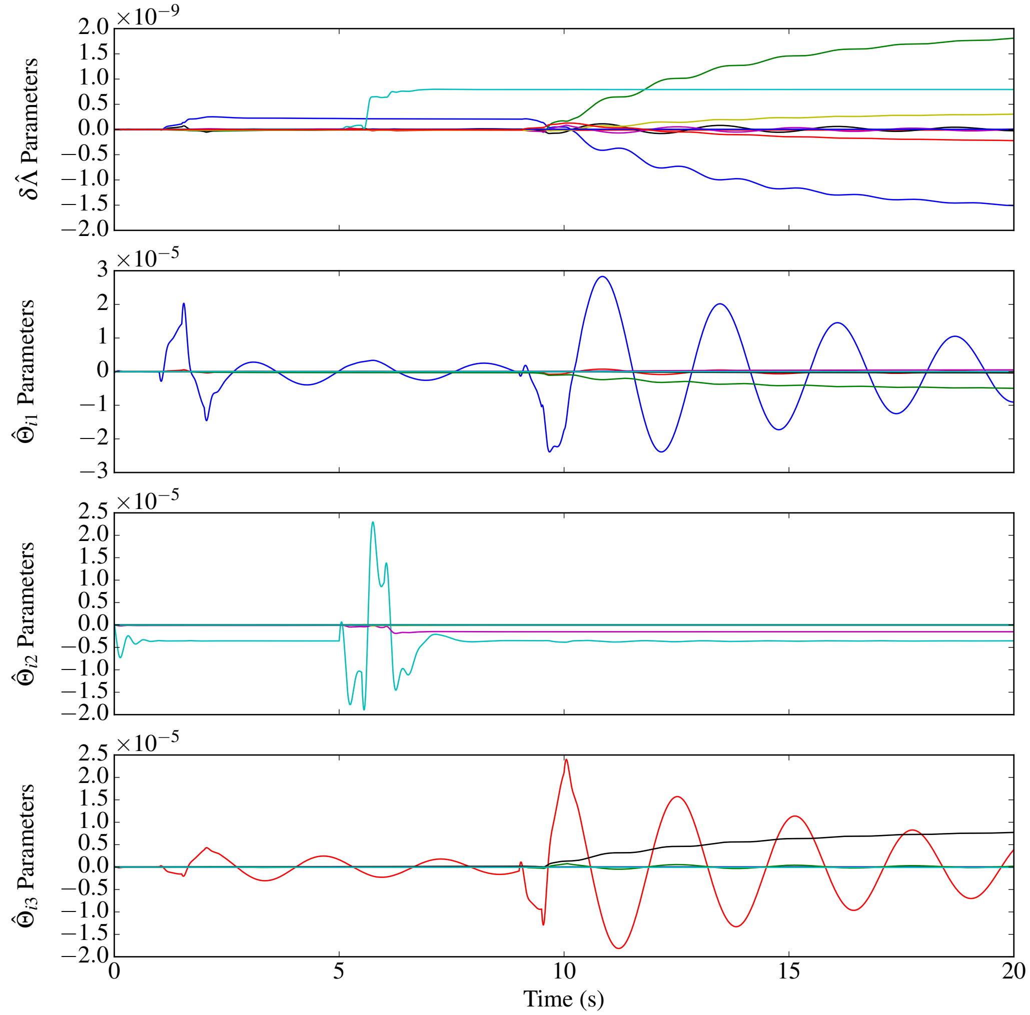
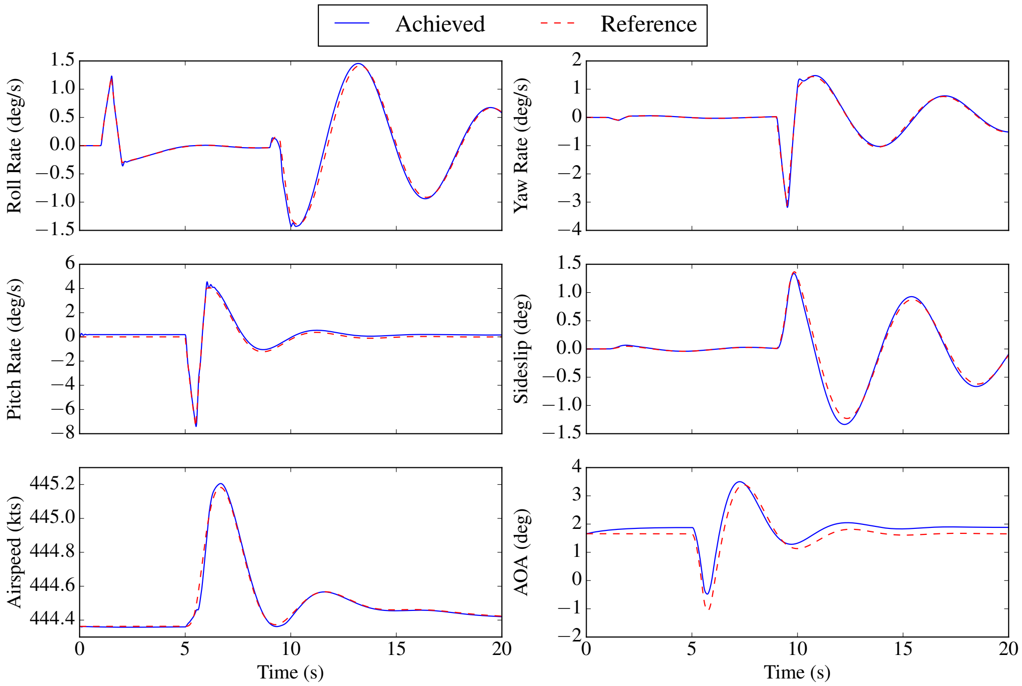
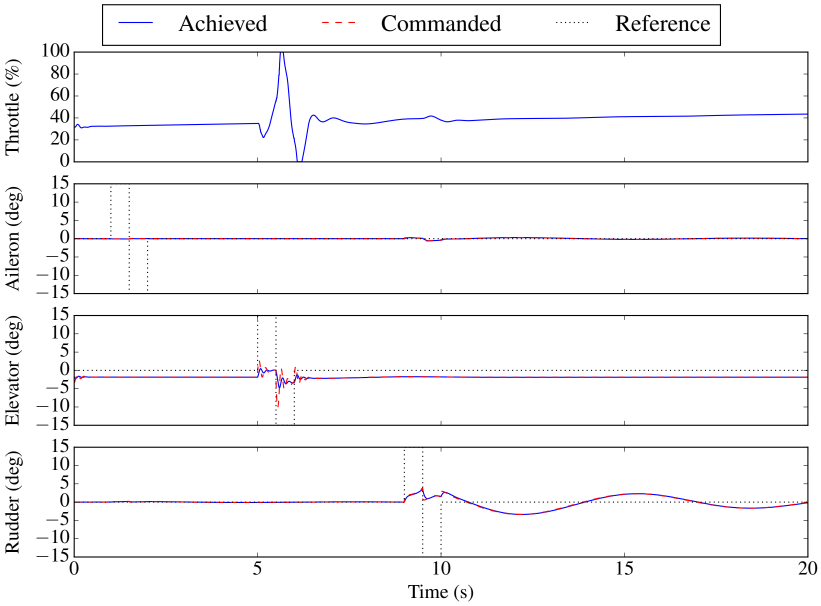


Figure A.8: Cessna T-37 Tweet $\mathcal{L}_1$ NDI adaptive parameter time histories.



(a) Actual and desired state time histories.



(b) Aerodynamic control effector time histories.

Figure A.9: Boeing 747 MRAC NDI controller state and control time histories.

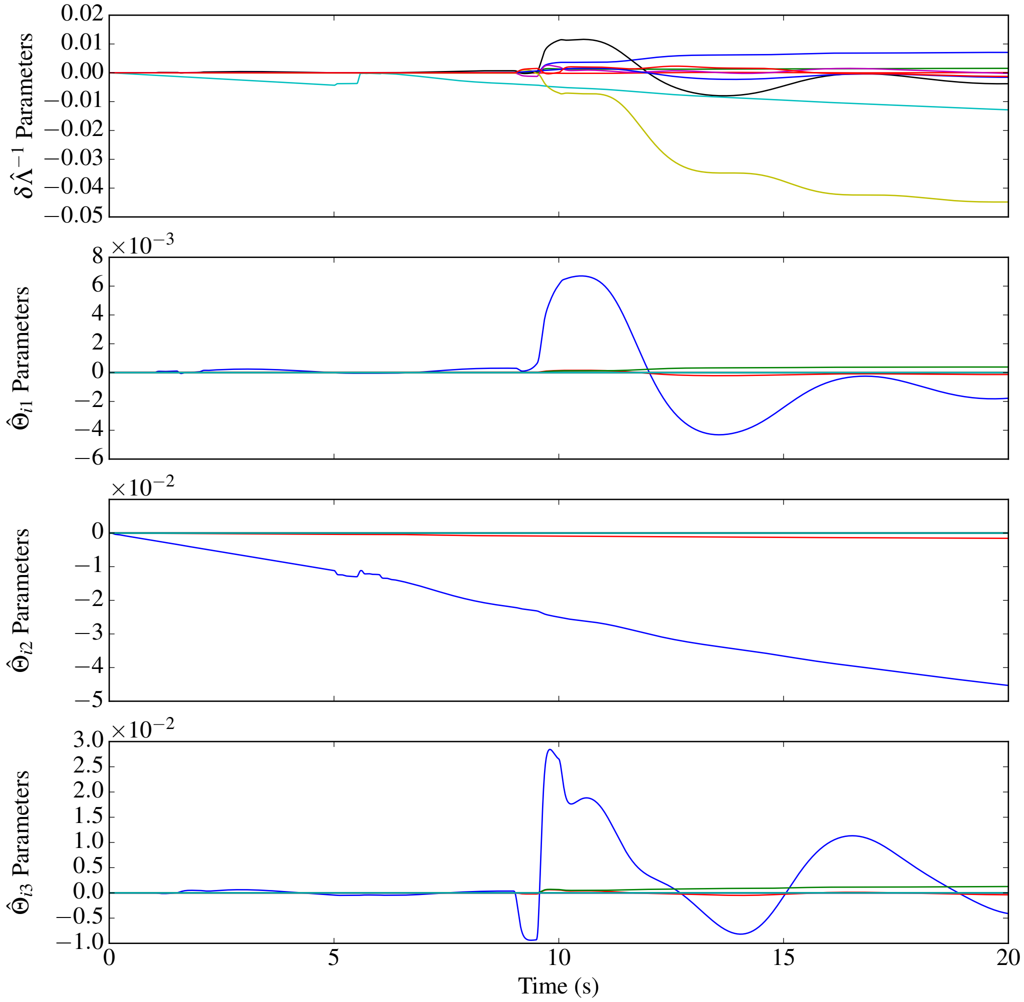
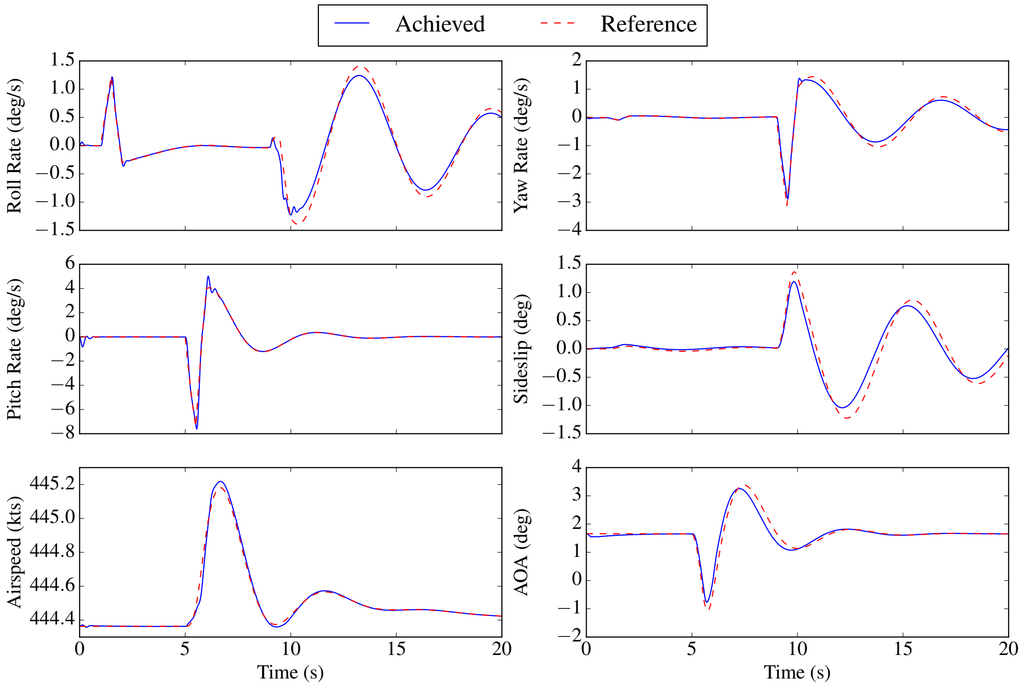
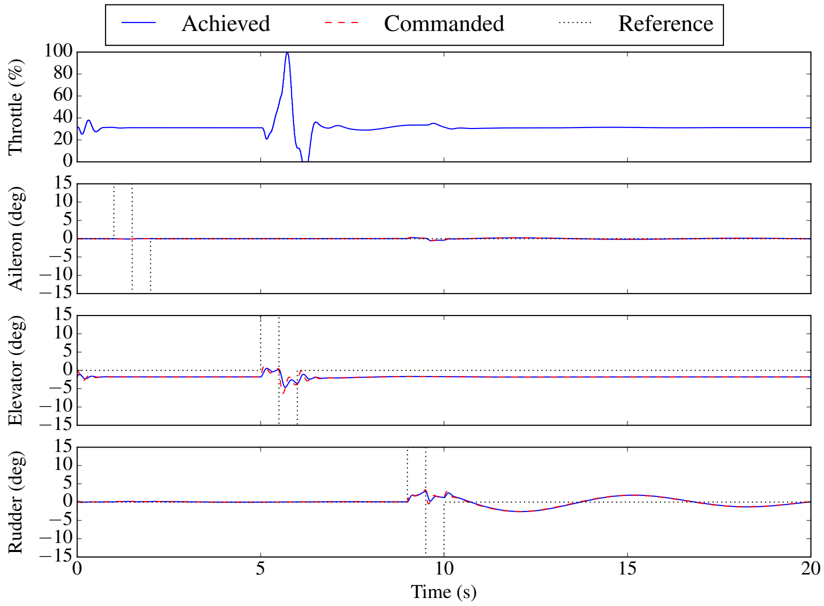


Figure A.10: Boeing 747 MRAC NDI adaptive parameter time histories.



(a) Actual and desired state time histories.



(b) Aerodynamic control effector time histories.

Figure A.11: Boeing 747 $\mathcal{L}_1$ NDI controller state and control time histories.

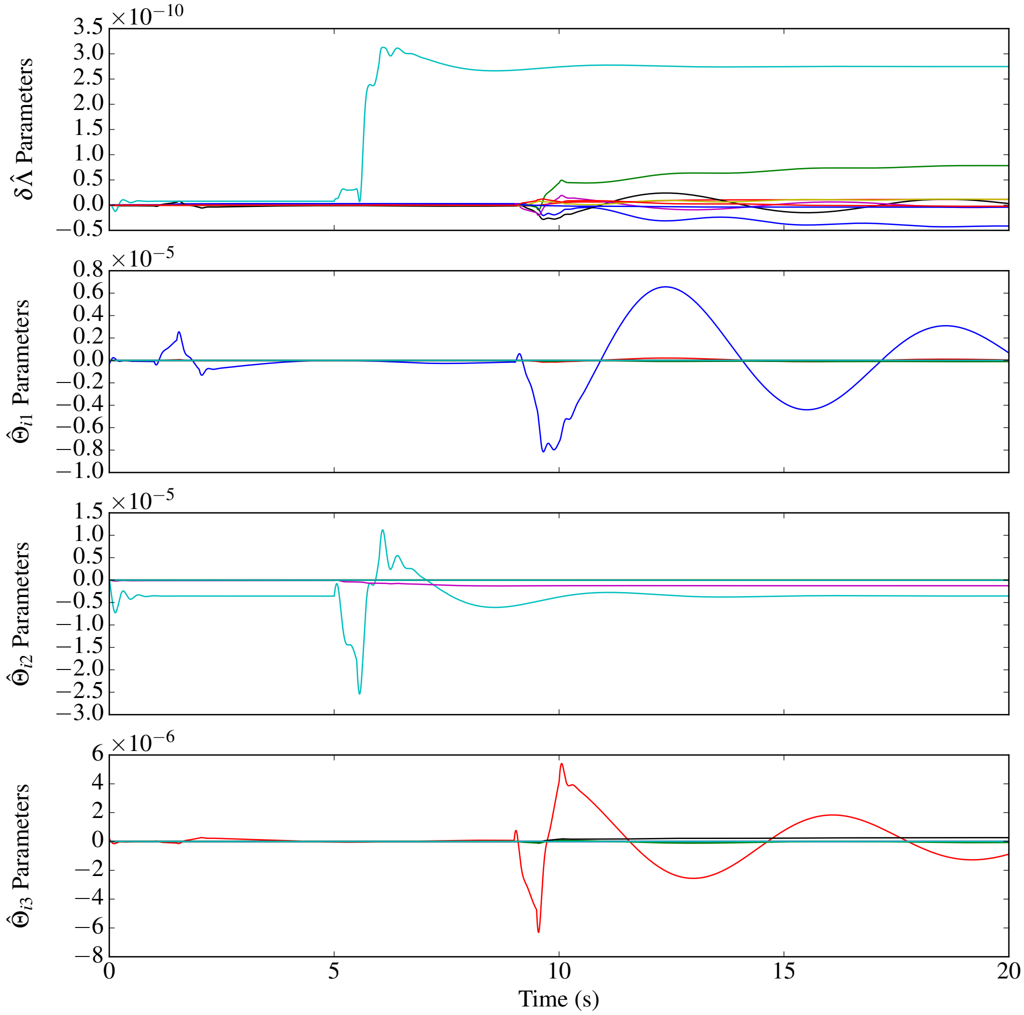
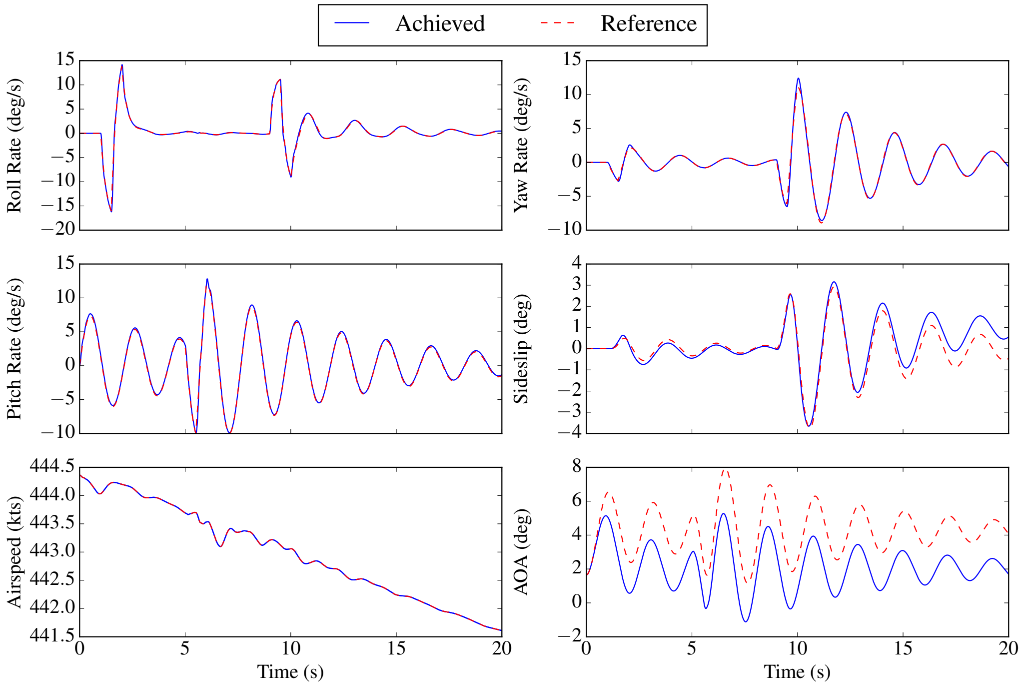
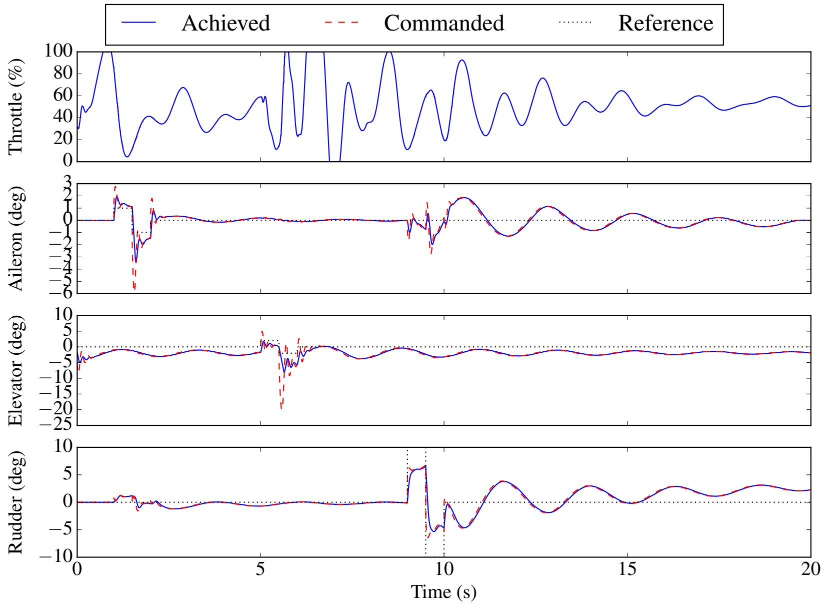


Figure A.12: Boeing 747 $\mathcal{L}_1$ NDI adaptive parameter time histories.



(a) Actual and desired state time histories.



(b) Aerodynamic control effector time histories.

Figure A.13: Convair F-106 Delta Dart MRAC NDI controller state and control time histories.

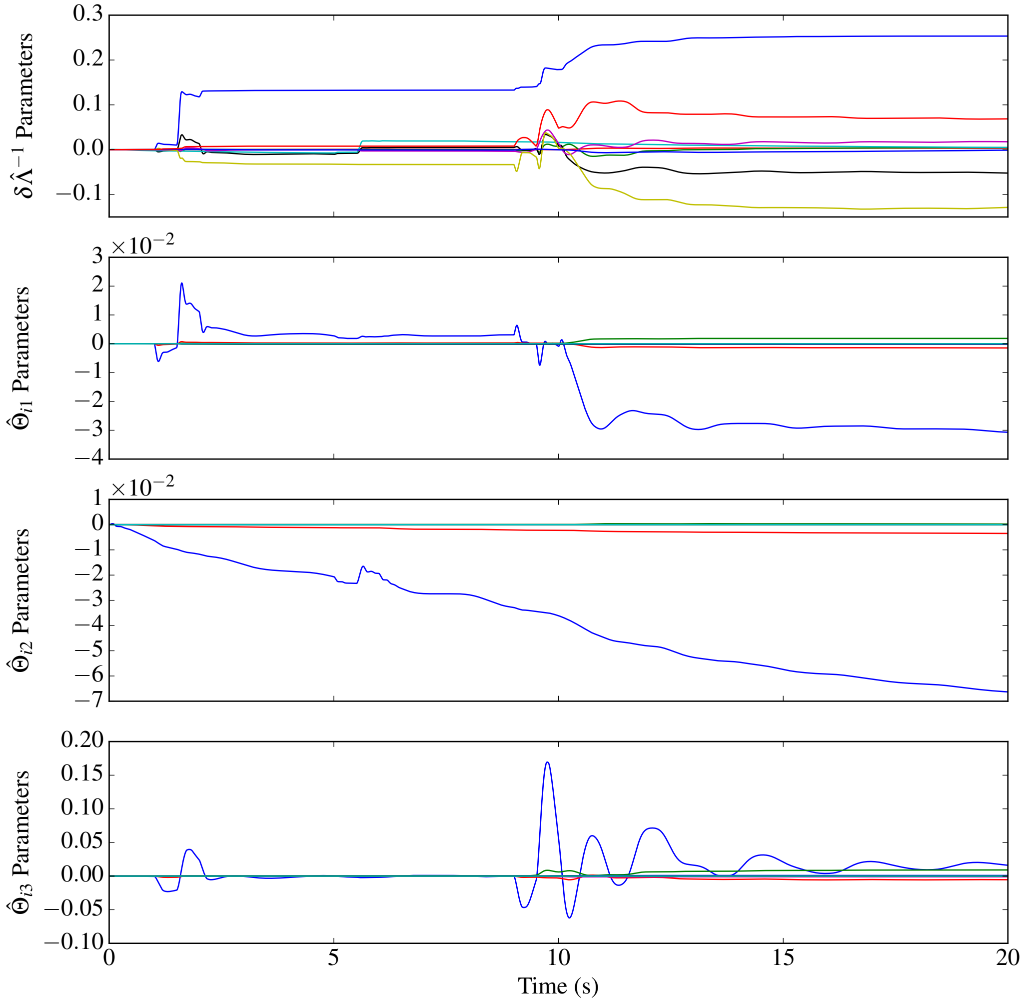
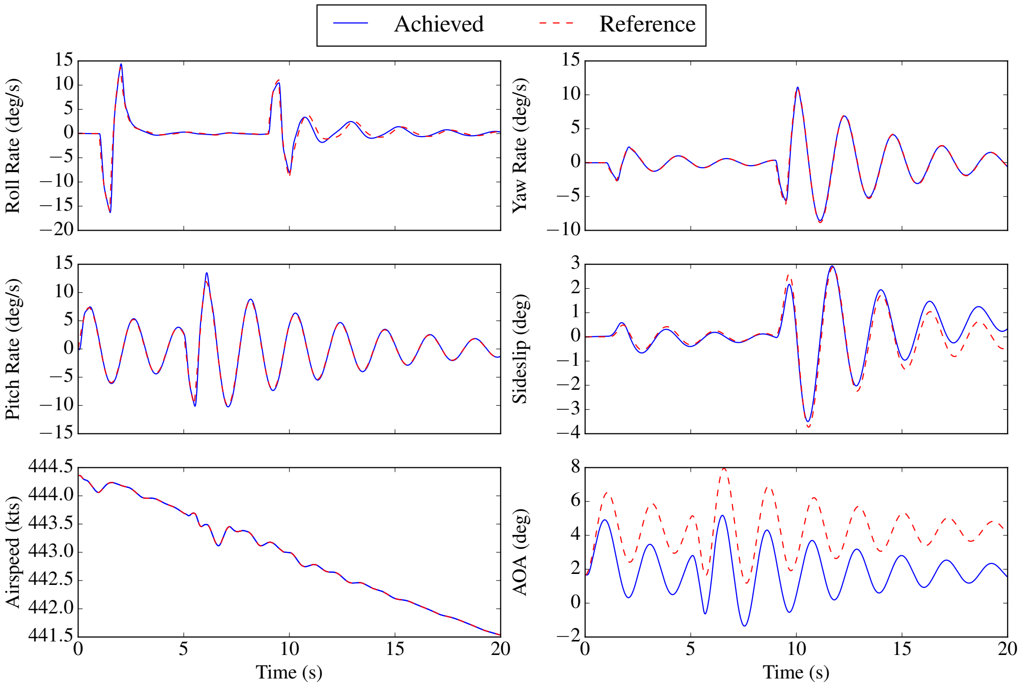
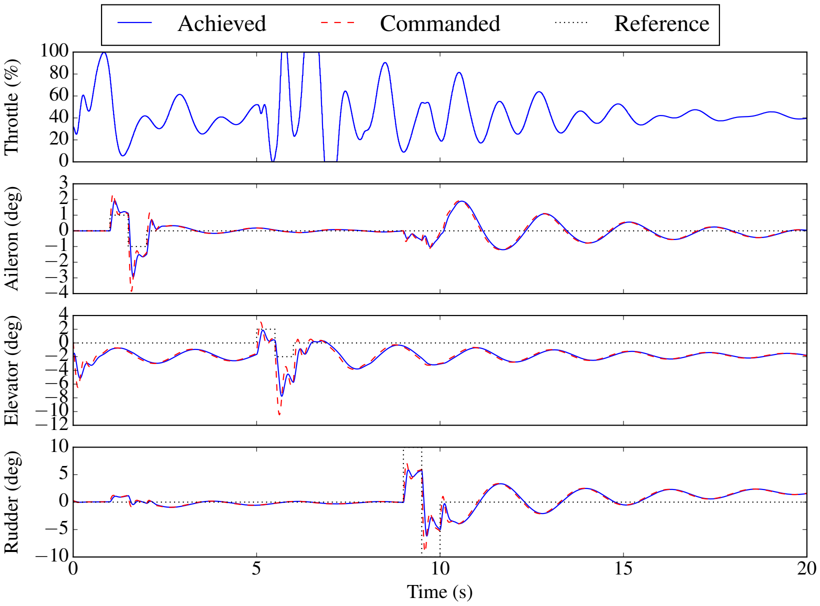


Figure A.14: Convair F-106 Delta Dart MRAC NDI adaptive parameter time histories.



(a) Actual and desired state time histories.



(b) Aerodynamic control effector time histories.

Figure A.15: Convair F-106 Delta Dart $\mathcal{L}_1$ NDI controller state and control time histories.

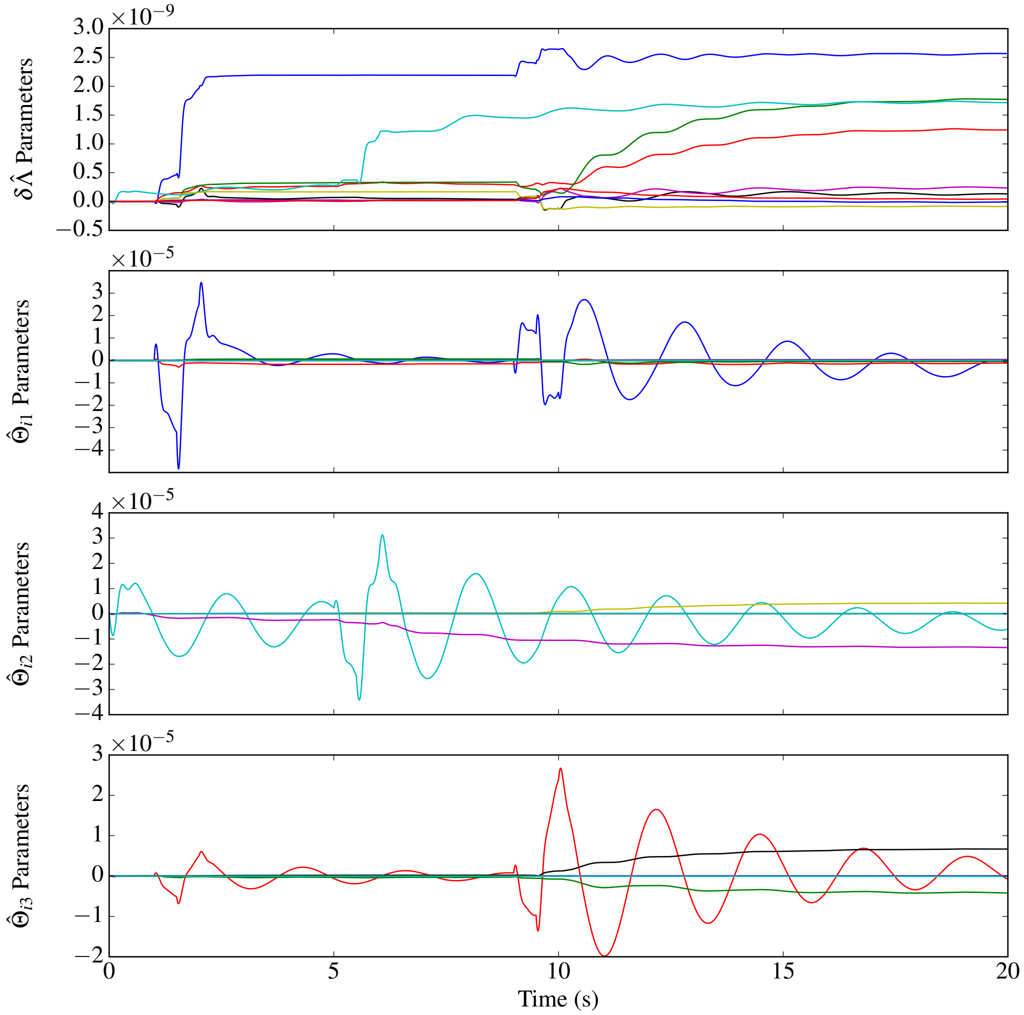
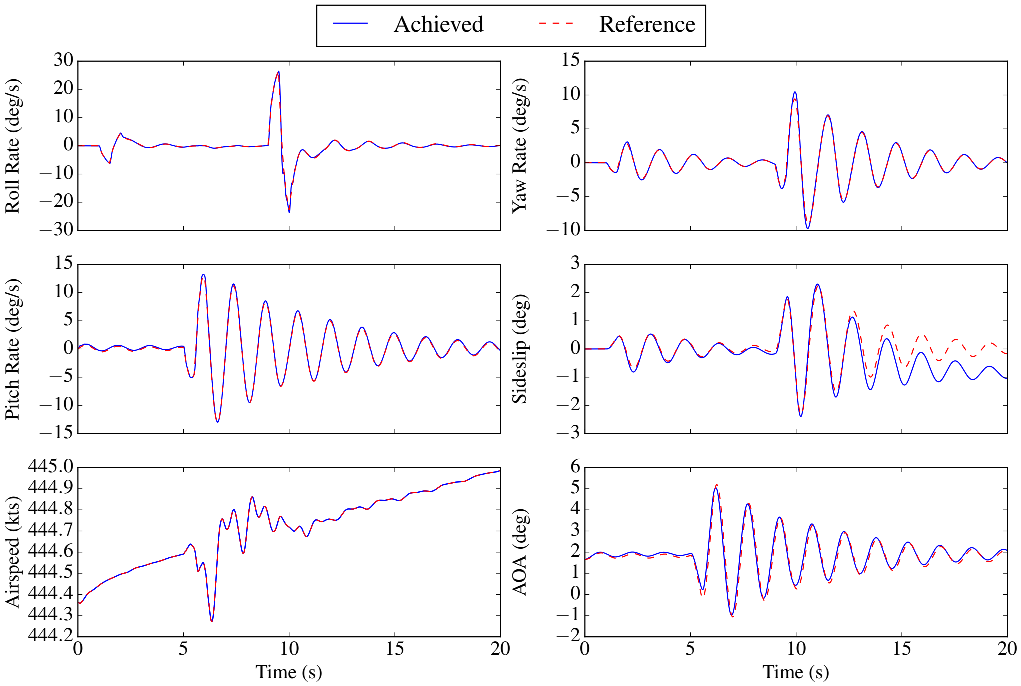
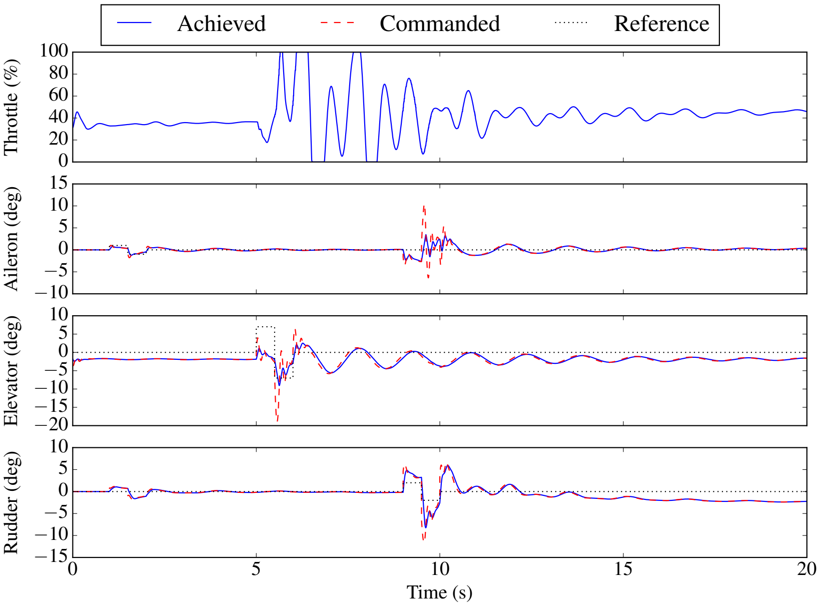


Figure A.16: Convair F-106 Delta Dart $\mathcal{L}_1$ NDI adaptive parameter time histories.



(a) Actual and desired state time histories.



(b) Aerodynamic control effector time histories.

Figure A.17: Rockwell/MBB X-31 MRAC NDI controller state and control time histories.

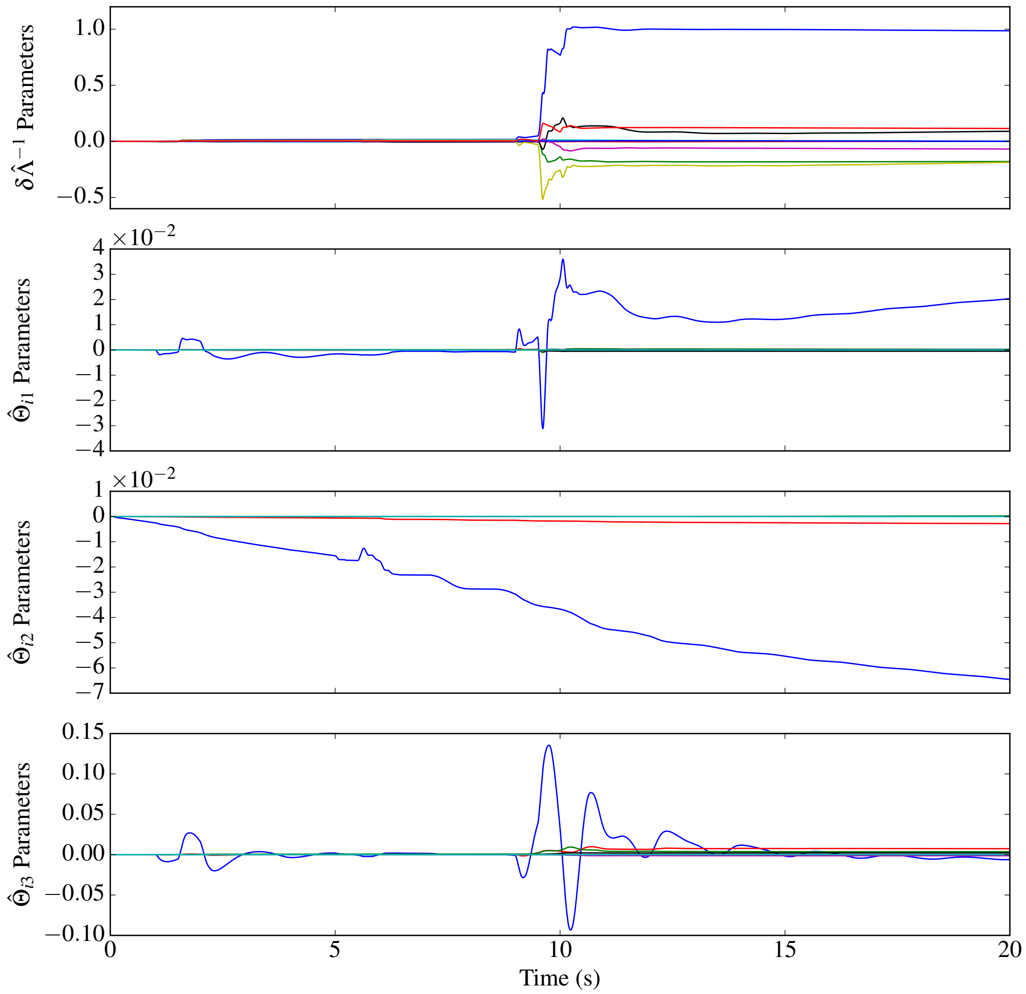
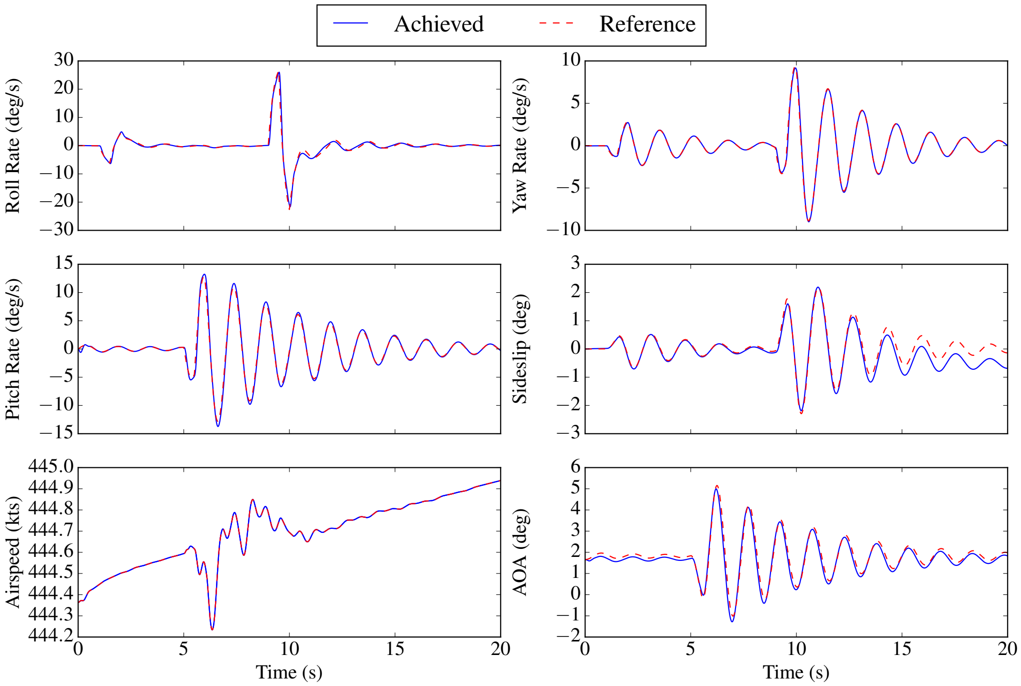
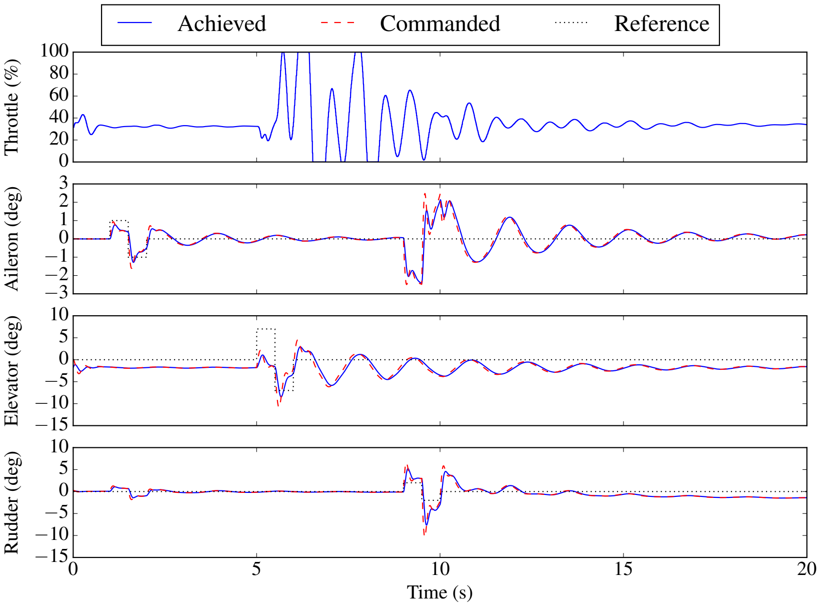


Figure A.18: Rockwell/MBB X-31 MRAC NDI adaptive parameter time histories.



(a) Actual and desired state time histories.



(b) Aerodynamic control effector time histories.

Figure A.19: Rockwell/MBB X-31 $\mathcal{L}_1$ NDI controller state and control time histories.

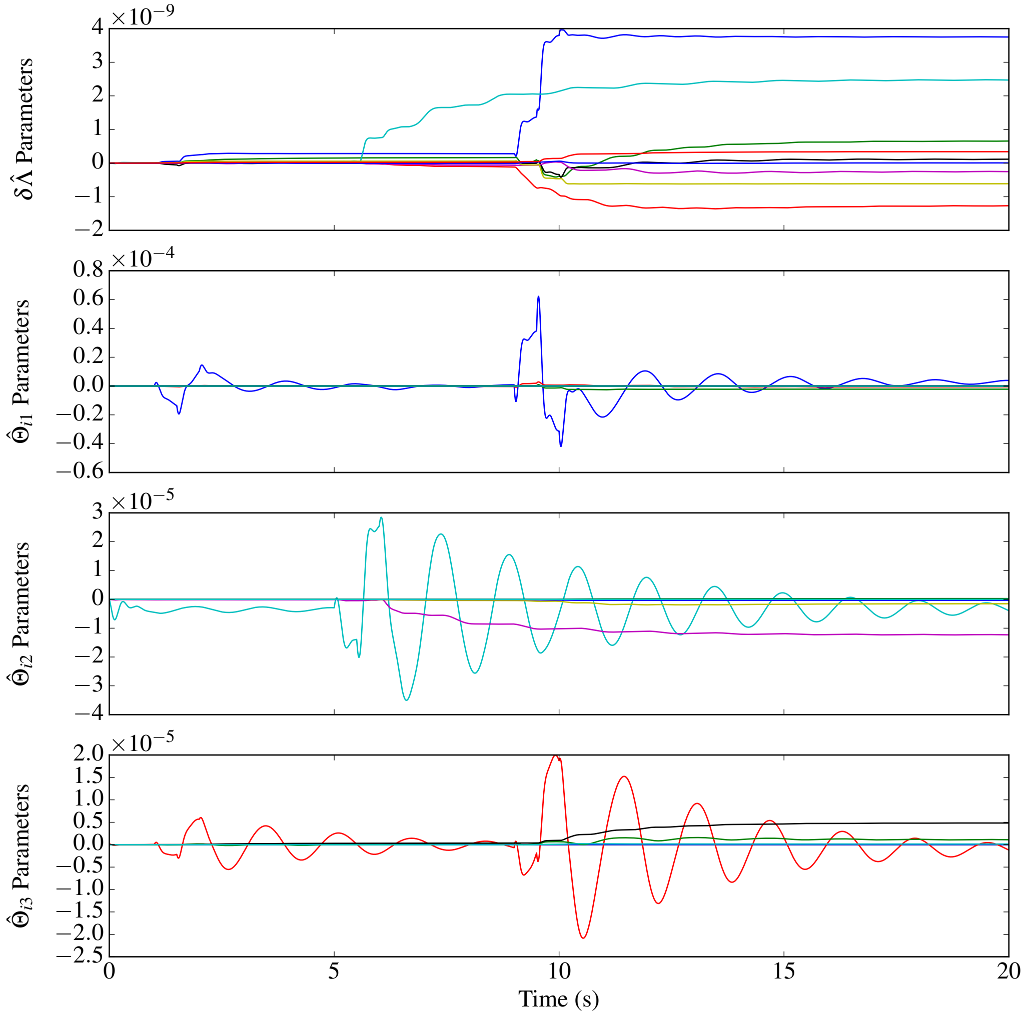


Figure A.20: Rockwell/MBB X-31 $\mathcal{L}_1$ NDI adaptive parameter time histories.

APPENDIX B

IN-FLIGHT SIMULATION AIRCRAFT MODELS

This appendix describes the various aircraft models used in this thesis as plants for the in-flight simulation controllers. These include both linear and nonlinear models. Model data is included for all of the models in this appendix.

B.1 Linear Models

This appendix provides the linear state-space models used for the in-flight simulation results. For all of the linear models except the Generic Transport Model (GTM) the longitudinal and lateral/directional axes are decoupled. Longitudinal states are u, α, q, and θ with control δ_{E}. Lateral/directional states are β, p, r, and ϕ with control vector $\boldsymbol{u}_{\mathrm{latd}} = [\delta_{\mathrm{A}}, \delta_{\mathrm{R}}]^{\mathsf{T}}$. All state-space models have $[D] = [0]$.

B.1.1 McDonnell-Douglas F-4 Phantom II

The McDonnell-Douglas F-4 Phantom II is a multi-role fighter during the Cold War era used by the US Navy, US Air Force, US Marine Corps, and other allied countries. The linear model for the F-4 is taken from the textbook by Roskam [36], which provides linear stability derivatives at several flight conditions. The selected flight condition is subsonic cruise, M0.9 at $35{,}000\,\mathrm{ft}$. State-space models are generated from the nondimensional stability and control derivatives. Equation (B.1) is the longitudinal state-space model and

Eq. (B.2) is the lateral/directional model.

$$[A_{\text{long}}] = \begin{bmatrix} -1.2863 \times 10^{-2} & -4.9530 & 0 & -3.2141 \times 10^{1} \\ -1.2733 \times 10^{-4} & -5.3372 \times 10^{-1} & 9.9657 \times 10^{-1} & -1.6643 \times 10^{-3} \\ -2.5951 \times 10^{-3} & -7.7365 & -7.1710 \times 10^{-1} & 3.8830 \times 10^{-4} \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

$$[B_{\text{long}}] = [12.3826 \quad -0.0565 \quad -11.3853 \quad 0]^{\mathsf{T}} \,. \tag{B.1}$$

$$[A_{\text{latd}}] = \begin{bmatrix} -9.6120 \times 10^{-2} & 0 & -1 & 3.6691 \times 10^{-2} \\ -1.9235 \times 10^{1} & -1.2292 & 3.8956 \times 10^{-1} & 0 \\ 5.6096 & -6.5658 \times 10^{-3} & -2.5627 \times 10^{-1} & 0 \\ 0 & 1 & 4.5410 \times 10^{-2} & 0 \end{bmatrix},$$

$$[B_{\text{latd}}] = \begin{bmatrix} -2.2617 \times 10^{-3} & 1.3429 \times 10^{-2} \\ 9.7746 & 1.7264 \\ -2.5217 \times 10^{-1} & -2.7802 \\ 0 & 0 \end{bmatrix}. \tag{B.2}$$

B.1.2 Generic Transport Model

The NASA Generic Transport Model is a scaled Boeing 757 RPA platform serving as a flight dynamics and flight control testbed at NASA Langley Research Center. The GTM model is a coupled linear model at at wings-level flight condition with $V_{\text{T}} = 100\,\text{kt}$ at $800\,\text{ft}$. The state vector is $\boldsymbol{x} = [V_{\text{T}}, \alpha, \beta, p, q, r, h, \phi, \theta]^{\mathsf{T}}$, the control vector is $\boldsymbol{u} = [\delta_{\text{E}}, \delta_{\text{A}}, \delta_{\text{R}}]^{\mathsf{T}}$, and the output vector is $\boldsymbol{y} = [p, q, r]^{\mathsf{T}}$. Equations (B.3–B.1.2) give the state-space model for the GTM. Note that there are more control effectors available in the model; the current effector suite is chosen as ganged elevators, ganged ailerons, and rudder.

$$[A] = \begin{bmatrix} -0.0675 & 18.8585 & -0.5614 & 0 & -0.0018 & 0 & 0.0004 & -32.1740 \\ -0.0022 & -3.2124 & 0.0018 & 0 & 0.9593 & 0 & 0.0001 & 0 \\ 0 & 0.0008 & -0.6475 & 0.0490 & 0 & -0.9877 & 0.1904 & 0 \\ 0.0003 & -10.5791 & -135.0587 & -8.1671 & -0.1575 & 1.8997 & 0 & 0 \\ -0.0091 & -65.9014 & 1.2658 & -0.0090 & -4.4979 & 0.0477 & 0 & 0 \\ 0.0002 & -0.8238 & 46.7588 & -0.5226 & -0.0467 & -1.7738 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0.0487 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0.0003 & 0 & 0 \end{bmatrix} \tag{B.3}$$

$$[B] = \begin{bmatrix} -0.04555 & -0.0247 & -0.0166 \\ -0.005323 & 0.0012 & -0.0007 \\ 0 & -0.0002 & 0.0038 \\ -0.02176 & -1.5396 & 0.3810 \\ -1.208 & 0.1325 & -0.0009 \\ -0.00106 & -0.0815 & -0.6978 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{B.4}$$

B.1.3 Rockwell Commander 700

The Rockwell Commander 700 is a small twin-engine general aviation aircraft. The linear models for the Commander 700 result from system identification flight test conducted at Texas A&M University. The flight condition is cruise with $V_\mathrm{T} = 206\,\mathrm{ft/s}$ at $8500\,\mathrm{ft}$. Equation (B.5) is the longitudinal state-space model and Eq. (B.6) is the lateral/directional model.

$$\begin{aligned} [A_\mathrm{long}] &= \begin{bmatrix} -0.0246 & 6.847 & 0 & -32.17 \\ -0.0012 & -1 & 1 & 0 \\ 0 & -3.535 & -2.245 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \\ [B_\mathrm{long}] &= [0 \quad -0.0915 \quad -8.574 \quad 0]^\mathsf{T} . \end{aligned} \tag{B.5}$$

$$
\begin{aligned}
[A_{\text{latd}}] &= \begin{bmatrix} -0.119 & -0.0013 & -0.993 & 0.159 \\ -1.22 & -2.00 & 0.0040 & 0 \\ 2.80 & -0.964 & -0.374 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \\
[B_{\text{latd}}] &= \begin{bmatrix} 0 & 0.0038 \\ 1.92 & 0.191 \\ 0.137 & -1.59 \\ 0 & 0 \end{bmatrix}.
\end{aligned}
\tag{B.6}
$$

B.1.4 Cessna T-37

The Cessna T-37 is a two-seat primary flight trainer. The linear model included below is assembled from S&C data found in Roskam. The data is for the flight condition M0.46 at $30{,}000\,\text{ft}$. Equation (B.7) is the longitudinal state-space model and Eq. (B.8) is the lateral/directional model.

$$
\begin{aligned}
[A_{\text{long}}] &= \begin{bmatrix} -1.3102\times10^{-2} & 1.0925\times10^{1} & 0 & -3.2154\times10^{1} \\ -3.0961\times10^{-4} & -9.6736\times10^{-1} & 9.9317\times10^{-1} & -2.4569\times10^{-3} \\ 3.5811\times10^{-4} & -1.8304\times10^{1} & -3.6284 & 2.8417\times10^{-3} \\ 0 & 0 & 1 & 0 \end{bmatrix}, \\
[B_{\text{long}}] &= [0 \quad -0.0934 \quad -30.9687 \quad 0]^{\mathsf{T}}\ .
\end{aligned}
\tag{B.7}
$$

$$
\begin{aligned}
[A_{\text{latd}}] &= \begin{bmatrix} -6.4761\times10^{-2} & -5.7367\times10^{-4} & -9.9792\times10^{-1} & 7.0514\times10^{-2} \\ -6.8181 & -1.1688 & 2.4867\times10^{-1} & 0 \\ 5.7098 & -3.4278\times10^{-2} & -2.6526\times10^{-1} & 0 \\ 0 & 1 & 3.4921\times10^{-2} & 0 \end{bmatrix}, \\
[B_{\text{latd}}] &= \begin{bmatrix} 0 & 0.03743 \\ 12.9398 & 1.0969 \\ -1.4248 & -1.8728 \\ 0 & 0 \end{bmatrix}.
\end{aligned}
\tag{B.8}
$$

B.1.5 Boeing 747

The Boeing 747 is a large wide-body commercial air transport. The linear model presented here derives from S&C data found in the textbook by Roskam. The data is for the flight condition M0.65 at $20{,}000\,\mathrm{ft}$. Equation (B.9) is the longitudinal state-space model and Eq. (B.10) is the lateral/directional model.

$$
\begin{aligned}
[A_{\text{long}}] &= \begin{bmatrix} -8.3188\times10^{-2} & 1.2386 & 0 & -3.2146\times10^{1} \\ -6.5537\times10^{-5} & -3.9077\times10^{-1} & 9.8255\times10^{-1} & -1.5332\times10^{-3} \\ -9.5090\times10^{-5} & -1.5614 & -5.4396\times10^{-1} & 2.1856\times10^{-4} \\ 0 & 0 & 1 & 0 \end{bmatrix}, \\
[B_{\text{long}}] &= [0 \quad -0.0211 \quad -1.2098 \quad 0]^{\mathsf{T}} .
\end{aligned}
\tag{B.9}
$$

$$
\begin{aligned}
[A_{\text{latd}}] &= \begin{bmatrix} -0.0640 & 0 & -1 & 0.0369 \\ -1.2752 & -0.4761 & 0.3009 & 0 \\ 1.0233 & 0.0142 & -0.1815 & 0 \\ 0 & 1 & 0.0419 & 0 \end{bmatrix}, \\
[B_{\text{latd}}] &= \begin{bmatrix} 0 & 0.0043 \\ 0.1853 & 0.0750 \\ -0.0148 & -0.4594 \\ 0 & 0 \end{bmatrix} .
\end{aligned}
\tag{B.10}
$$

B.2 Nonlinear Models

The nonlinear model data is sourced from Ref. [43] by Grauer and Morelli, and consists of estimated S&C derivatives for the model Eqs. (2.46–2.51). Data for five aircraft are used in this thesis: the F-16C, F-4, F-106, X-31, and GTM. These data are reproduced in Table B.1, which contains mass properties and geometry, and Table B.2, which containes aerodynamic model parameters. Note that the thrust field in Table B.1 is taken from various open sources and does not come from Ref. [43]. The thrust data in the table is used as the maximum thrust value for the generic turbofan Simulink block. Thrust data is not included for the F-16 as only the aerodynamic data is used in the approximated onboard model, which neglects effects due to thrust. Finally, note that the parameter $\bar{x}_{\text{cg}}$ is the nondimensionalized reference CG location about which the forces and moments act.

Table B.1: Selected mass properties and geometry for GNA aircraft models [43].

Parameter	**F-16C**	**F-4**	**F-106**	**X-31**	**GTM**
m (lbf)	20500	38294	29776	16000	49.6
I_{xx} (slug·ft^2)	9496	24970	18634	3553	1.327
I_{yy} (slug·ft^2)	55814	122190	177858	50645	4.254
I_{zz} (slug·ft^2)	63100	139800	191236	49367	5.454
I_{xz} (slug·ft^2)	982	1175	5539	156	0.120
S_{ref} (ft^2)	300	530	698	226.3	5.902
b_{ref} (ft)	30	38.67	38.12	22.83	6.849
$\bar{c}$ (ft)	11.32	16	23.75	12.35	0.915
$\bar{x}_{\mathrm{cg}}$ ($\%\bar{c}$)	25	29	25	30	30
Thrust (lbf)	—	23810	24500	16000	15

Table B.2: Selected aerodynamic data for GNA aircraft models [43].

Parameter	**F-16C**	**F-4**	**F-106**	**X-31**	**GTM**
C_{D_1}	0.034	0.031	0.052	0.015	0.019
C_{D_α}	-0.005	0.280	-0.202	-0.157	-0.078
$C_{\mathrm{D}_{\alpha\mathrm{q}}}$	20.77	-11.98	-9.298	1.587	-27.42
$C_{\mathrm{D}_{\alpha\delta_\mathrm{E}}}$	0.177	0	0.396	0.042	0.293
$C_{\mathrm{D}_{\alpha^2}}$	1.285	-1.818	-0.858	0.697	3.420
$C_{\mathrm{D}_{\alpha^2\mathrm{q}}}$	-19.97	209.4	-13.89	-3.684	288.2
$C_{\mathrm{D}_{\alpha^2\delta_\mathrm{E}}}$	0.756	0.515	0.911	0.302	-0.040
$C_{\mathrm{D}_{\alpha^3}}$	5.887	22.27	14.62	8.674	1.819
$C_{\mathrm{D}_{\alpha^3\mathrm{q}}}$	55.59	-287.4	70.04	21.74	-355.3
$C_{\mathrm{D}_{\alpha^4}}$	-5.155	-29.81	-15.41	-11.19	-6.563
C_{Y_β}	-1.146	-0.688	-0.573	-0.014	-1.003
$C_{\mathrm{Y}_\mathrm{p}}$	-0.188	0.129	-0.100	-0.122	0.033

continued . . .

. . . continued

Parameter	**F-16C**	**F-4**	**F-106**	**X-31**	**GTM**
$C_{\mathrm{Y_r}}$	0.876	0.670	0.500	0.710	0.952
$C_{\mathrm{Y}_{\delta_\mathrm{A}}}$	0.060	0	0	0.345	-0.009
$C_{\mathrm{Y}_{\delta_\mathrm{R}}}$	0.164	0.089	0	0.671	0.253
$C_{\mathrm{L_1}}$	0.074	0.105	-0.017	-0.020	0.016
C_{L_α}	4.458	1.519	1.888	3.023	5.343
$C_{\mathrm{L_q}}$	29.90	6.727	-9.226	3.697	30.78
$C_{\mathrm{L}_{\delta_\mathrm{E}}}$	0.412	0.265	0.774	0.237	0.396
$C_{\mathrm{L}_{\alpha\mathrm{q}}}$	-5.538	33.25	14.52	6.616	12.03
$C_{\mathrm{L}_{\alpha^2}}$	-2.477	9.9	-9.438	-3.330	0.506
$C_{\mathrm{L}_{\alpha^3}}$	-1.101	-12.71	48.71	11.60	-36.30
$C_{\mathrm{L}_{\alpha^4}}$	1.906	-12.91	-53.62	-16.94	46.13
C_{ℓ_β}	-0.071	-0.034	0	-0.002	-0.109
C_{ℓ_p}	-0.445	-0.236	-0.300	-0.395	-0.366
C_{ℓ_r}	0.058	0.025	0	-0.021	0.061
$C_{\ell_{\delta_\mathrm{A}}}$	-0.143	-0.035	-0.147	-0.048	-0.079
$C_{\ell_{\delta_\mathrm{R}}}$	0.023	0.013	0.009	0.101	0.021
$C_{\mathrm{m_1}}$	-0.024	-0.013	0.020	0.028	0.182
C_{m_α}	-0.288	-0.254	-0.267	-0.870	-1.782
$C_{\mathrm{m_q}}$	-8.267	-2.916	-0.497	-2.352	-44.34
$C_{\mathrm{m}_{\delta_\mathrm{E}}}$	-0.563	-0.403	-0.378	-0.166	-1.785
$C_{\mathrm{m}_{\alpha\mathrm{q}}}$	-5.513	-3.955	0.313	-0.488	374.0
$C_{\mathrm{m}_{\alpha^2\mathrm{q}}}$	9.793	-24	-11.72	-9.691	-1748

continued . . .

…continued

Parameter	**F-16C**	**F-4**	**F-106**	**X-31**	**GTM**
$C_{\mathrm{m}_{\alpha^2\delta_\mathrm{E}}}$	-1.065	-0.270	-0.644	-0.064	-2.439
$C_{\mathrm{m}_{\alpha^3\mathrm{q}}}$	-2.018	55.32	19.60	1.262	1949
$C_{\mathrm{m}_{\alpha^3\delta_\mathrm{E}}}$	1.987	1.479	1.443	0.361	-0.038
$C_{\mathrm{m}_{\alpha^4}}$	-0.094	-0.448	-0.048	1.795	0.803
C_{n_β}	0.234	0.142	0.152	0.406	0.183
$C_{\mathrm{n}_\mathrm{p}}$	0.056	-0.006	0.002	0.205	-0.022
$C_{\mathrm{n}_\mathrm{r}}$	-0.418	-0.358	-0.308	-0.875	-0.405
$C_{\mathrm{n}_{\delta_\mathrm{A}}}$	-0.034	0.001	-0.090	-0.128	-0.009
$C_{\mathrm{n}_{\delta_\mathrm{R}}}$	-0.085	-0.053	-0.044	-0.223	-0.129
$C_{\mathrm{n}_{\beta^2}}$	0.372	0	0	-2.575	0.184
$C_{\mathrm{n}_{\beta^3}}$	-0.725	0.337	0	4.329	-0.337

APPENDIX C

DISTURBANCE MODELS

Four types of wind and turbulence are considered in the modeling work done for this thesis:

- horizontal wind,
- wind shear,
- wind turbulence, and
- wind gusts.

The models for these phenomena are taken from military flying qualities specifications, particularly MIL-F-8785C [47]. An updated standard, MIL-STD-1797B, exists but is not available to the public [50]. These models are implemented in the F-16 simulation developed in Section 3 using blocks from the Simulink Aerospace Blockset.

C.1 Horizontal Wind

Horizontal wind is specified simply as a wind speed and direction. The horizontal wind can be written in the NED frame as

$$\boldsymbol{V}_{\text{wind}} = -V_{\text{wind}} \left(\cos(\psi_{\text{wind}}) \hat{\boldsymbol{x}}_{\text{n}} + \sin(\psi_{\text{wind}}) \hat{\boldsymbol{y}}_{\text{n}} \right) \tag{C.1}$$

where V_{wind} is the wind speed and ψ_{wind} is the direction from which the wind is coming, measured clockwise from the $\hat{\boldsymbol{x}}_{\text{n}}$ axis. Note that V_{wind} and ψ_{wind} are not necessarily constant.

C.2 Wind Shear

The wind shear magnitude u_{w} is given by

$$u_{\mathrm{w}} = u_{20} \frac{\ln\left(\dfrac{h}{z_0}\right)}{\ln\left(\dfrac{20}{z_0}\right)} \tag{C.2}$$

where u_{20} is the wind speed at an altitude of $20\,\mathrm{ft}$, h is the altitude above ground level, and $z_0 = 0.15\,\mathrm{ft}$ for Category C (i.e., terminal) flight phases or $2.0\,\mathrm{ft}$ for other flight phases. The wind shear vector in the NED frame is then

$$\boldsymbol{V}_{\mathrm{shear}} = -u_{\mathrm{w}} \left(\cos(\psi_{\mathrm{shear}})\hat{\boldsymbol{x}}_{\mathrm{n}} + \sin(\psi_{\mathrm{shear}})\hat{\boldsymbol{y}}_{\mathrm{n}}\right) \tag{C.3}$$

where u_{w} is defined in Eq. (C.2) and ψ_{shear} is the angle of the oncoming wind shear.

C.3 Wind Turbulence

The two most common turbulence models are those of Dryden and von Karman. The Dryden form is simpler to use and is thus used to provide turbulence spectra for this thesis. The Dryden models for the u, v, and w components are

$$F_{\mathrm{u}}(s) = \sigma_{\mathrm{u}} \sqrt{\frac{2V}{L_{\mathrm{u}}}} \frac{1}{s + \dfrac{V}{L_{\mathrm{u}}}}\,, \tag{C.4a}$$

$$F_{\mathrm{v}}(s) = \sigma_{\mathrm{v}} \sqrt{\frac{V}{L_{\mathrm{v}}}} \frac{\sqrt{3}s + \dfrac{V}{L_{\mathrm{v}}}}{s^2 + 2\dfrac{V}{L_{\mathrm{v}}}s + \left(\dfrac{V}{L_{\mathrm{v}}}\right)^2}\,, \text{ and} \tag{C.4b}$$

$$F_{\mathrm{w}}(s) = \sigma_{\mathrm{w}} \sqrt{\frac{V}{L_{\mathrm{w}}}} \frac{\sqrt{3}s + \dfrac{V}{L_{\mathrm{w}}}}{s^2 + 2\dfrac{V}{L_{\mathrm{w}}}s + \left(\dfrac{V}{L_{\mathrm{w}}}\right)^2}\,. \tag{C.4c}$$

In Eq. (C.4), L_u, L_v, and L_w are length scales, while σ_u, σ_v, and σ_w are intensities, which can be found in MIL-F-8785C or MIL-STD-1797A. The turbulence filters Eq. (C.4) are driven by independent, identically distributed unit-intensity white-noise signals [49].

Using MIL-F-8785C, the lengths scales are

$$\begin{aligned} L_\mathrm{u} &= L_\mathrm{v} = \frac{h}{(0.177 + 0.000823h)^{1.2}} \text{ and} \\ L_\mathrm{w} &= h \end{aligned}$$

for $10 < h < 1000\,\mathrm{ft}$. For $h > 2000\,\mathrm{ft}$, $L_\mathrm{u} = L_\mathrm{v} = L_\mathrm{w} = 1750\,\mathrm{ft}$. The turbulence scales are defined by

$$\begin{aligned} \sigma_\mathrm{u} &= \frac{\sigma_\mathrm{w}}{(0.177 + 0.000823h)^{0.4}}, \\ \sigma_\mathrm{v} &= \frac{\sigma_\mathrm{w}}{(0.177 + 0.000823h)^{0.4}} \text{ and} \\ \sigma_\mathrm{w} &= 0.1u_{20} \end{aligned}$$

for altitudes under $1000\,\mathrm{ft}$. For altitudes over $2000\,\mathrm{ft}$, the turbulence intensities are shown in Fig. C.1.

Between $1000\,\mathrm{ft}$ and $2000\,\mathrm{ft}$, the turbulence components are interpolated between the low altitude and medium-high altitude models.

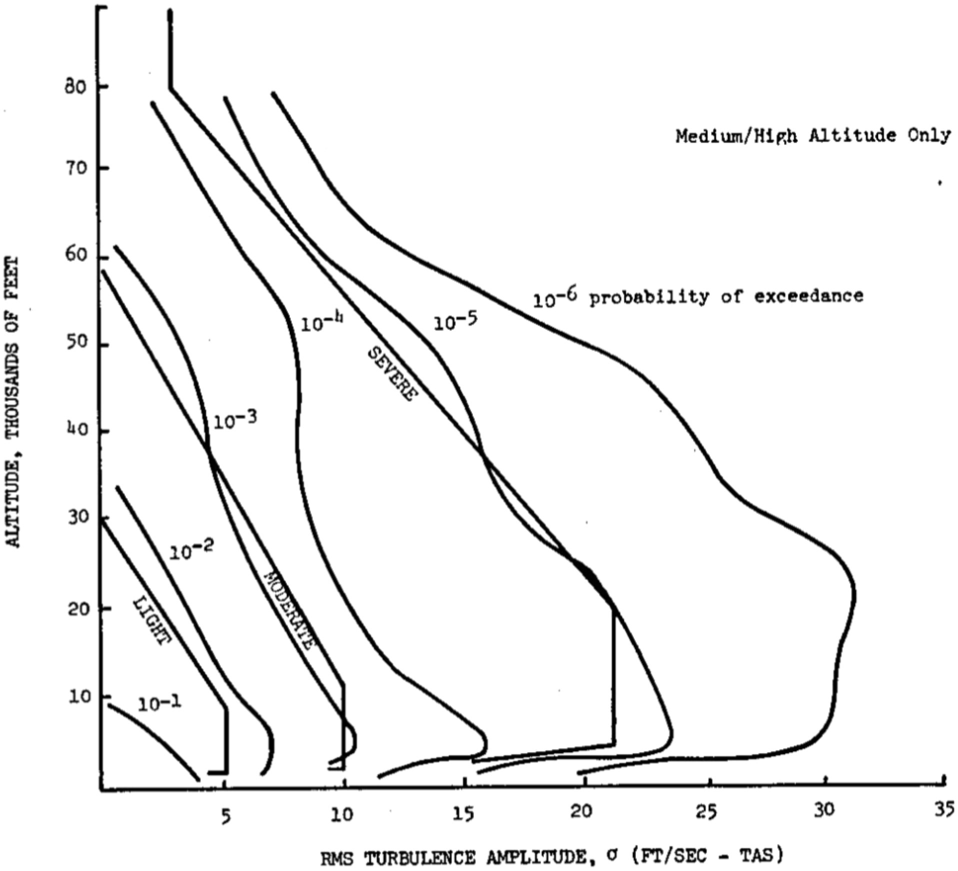


Figure C.1: Dryden turbulence intensities, medium-high altitude. (Reprinted from [47].)

C.4 Wind Gusts

The MIL-F-8785C discrete gust model takes the following form:

$$v = \begin{cases} 0 & x < 0 \\ \dfrac{v_{\mathrm{m}}}{2}\left(1 - \cos\left(\dfrac{\pi x}{d_{\mathrm{m}}}\right)\right) & 0 \leqslant x \leqslant d_{\mathrm{m}} \\ v_{\mathrm{m}} & x > d_{\mathrm{m}} \end{cases} \tag{C.5}$$

where d_{m} is the gust length and v_{m} is the gust amplitude. The values of d_{m} are chosen for each axis such that the gust is tuned to the natural frequencies of the aircraft. Gust magnitudes are chosen from the values of the gust length based on Figure 8 of MIL-F-8785C [47].