# AIRCRAFT CONTROL ALLOCATION METHODS CONSIDERING PARAMETRIC MODEL UNCERTAINTY

by

Luke Joseph Miller

May 2025

A dissertation submitted to the
Faculty of the Graduate School of
the University at Buffalo, State University of New York
in partial fulfilment of the requirements for the
degree of

Doctor of Philosophy

Department of Mechanical & Aerospace Engineering

Copyright by

Luke Joseph Miller

2025

# Dedication

To my family who has selflessly helped me pursue my dreams.

# Acknowledgments

I would firstly like to express my deep gratitude to my committee, Dr. John Crassidis, Dr. Eleonora Botta, Dr. D. Joseph Mook, and Dr. Kasey Ackerman for their support, belief in me, and care during the difficult times of my Ph.D. studies.

I would also like to thank the Advanced degree program at NASA Langley Research center for providing funding to help make this dissertation possible.

This dissertation also would not have been possible without my colleagues at NASA Langley Research Center and their guidance and support, particularly, Jing Pei, Jared Grauer, Michael Acheson, Irene Gregory, Tenavi Nakamura-Zimmerer, and Rose Weinstein.

Lastly, I would like to thank my family for their unwavering love and support, which helped make this dissertation possible.

# Table of Contents

**Acknowledgments** **iv**

**List of Tables** **ix**

**List of Figures** **xi**

**Abstract** **xv**

**Chapter 1**

**Introduction** **1**

1.1 Overview of the Modeling and Control for Agile Aircraft Development Project . . . . . 3

1.2 Control Principles . . . . . 4

1.3 Contributions and Organization . . . . . 5

**Chapter 2**

**Preliminaries** **8**

2.1 Control Allocation . . . . . 8

2.1.1 Generalized Inverse Control Allocation . . . . . 9

2.1.2 Control Allocation by Linear Programming . . . . . 10

2.1.3 Control Allocation by Quadratic Programming . . . . . 13

2.2 Dual Norm . . . . . 16
2.3 Second-Order Cone Programs . . . . . 16
2.4 Summary . . . . . 17

**Chapter 3**

**Linear Parameter-Varying Control Compatible with Excitation for Real-Time Modeling 18**

3.1 Introduction . . . . . 19
3.2 Excitation for Real-Time Modeling . . . . . 20
3.3 LPV Model Derivation . . . . . 22
3.3.1 Affine Polytopic LPV Systems . . . . . 27
3.3.2 Tensor Product (TP) Transformation . . . . . 28
3.4 LPV Control Design . . . . . 33
3.4.1 NDI Review . . . . . 39
3.4.2 AutoThrottle . . . . . 41
3.4.3 Control Allocation . . . . . 42
3.5 Simulation . . . . . 43
3.5.1 Outer Loop Implementation . . . . . 44
3.5.2 Inner Loop Implementation . . . . . 45
3.6 Chapter Conclusions . . . . . 51

**Chapter 4**

**A Comparison of Control Allocation Methods in the Presence of Parametric Model Uncertainty 53**

4.1 Example . . . . . 56
4.1.1 Static Allocator Performance . . . . . 58
4.1.2 Closed-Loop Allocator Performance . . . . . 64

4.2 Chapter Conclusions . . . . . 74

**Chapter 5**

**Control Allocation by Robust Linear Programming 75**

5.1 Introduction . . . . . 75

5.2 Robust Linear Programming Allocation Design . . . . . 78

5.2.1 Box Uncertainty Sets . . . . . 80

5.2.2 Ellipsoidal Uncertainty Sets . . . . . 82

5.3 Example . . . . . 83

5.3.1 Allocator Performance . . . . . 84

5.3.2 Solver Performance . . . . . 91

5.4 Chapter Conclusions . . . . . 92

**Chapter 6**

**High-Efficiency Direction-Preserving Control Allocation 94**

6.1 Introduction . . . . . 94

6.2 Control Allocation by Linear Programming . . . . . 96

6.3 Linear Programming Direction-Preserving Control Allocation . . . . 98

6.3.1 Problem Formulation . . . . . 98

6.3.2 Worst-Case Direction Error Constraint . . . . . 100

6.4 Simulation Example . . . . . 101

6.4.1 Attainable Moment Sets and Allocation Efficiency . . . . . 102

6.4.2 Magnitude and Direction Errors . . . . . 103

6.5 Solver Analysis . . . . . 110

6.6 Chapter Conclusions . . . . . 111

**Chapter 7**

**Direction-Preserving Control Allocation by Robust Linear Programming** **113**

7.1 Introduction . . . . . 113

7.2 Robust Linear Programming Direction-Preserving Control Allocation 114

7.2.1 Problem Formulation . . . . . 115

7.3 Simulation Example . . . . . 120

7.3.1 Attainable Moment Sets and Allocation Efficiency . . . . . 123

7.3.2 Moment Direction Errors . . . . . 124

7.4 Chapter Conclusions . . . . . 126

**Chapter 8**

**Nonlinear Aircraft Example** **127**

8.1 T-2 Aircraft Model . . . . . 128

8.2 Aircraft Simulations . . . . . 130

8.2.1 Nominal Simulation Case . . . . . 131

8.2.2 Model Uncertainty Case . . . . . 134

8.2.3 Effector Failure Case . . . . . 135

8.3 Chapter Conclusions . . . . . 137

**Chapter 9**

**Conclusions** **138**

9.1 Future Research . . . . . 139

9.2 List of Publications . . . . . 140

# List of Tables


3.1 PTI frequencies for each effector (Hz) . . . . . 21

3.2 Parameter bounds and points . . . . . 32

3.3 Error Parameters . . . . . 32

3.4 Desired frequency calculations . . . . . 41

4.1 F-16 AFTI aircraft control surface position and rate limits . . . . . 57

4.2 Standard errors used in example . . . . . 58

4.3 Maximum allocation errors with nominal $\boldsymbol{B}$ matrix . . . . . 71

4.4 Cumulative allocation errors with nominal $\boldsymbol{B}$ matrix . . . . . 72

4.5 Mean maximum allocation errors . . . . . 72

4.6 Mean cumulative allocation errors . . . . . 72

5.1 Standard errors used in example . . . . . 84

5.2 Standard Deviations in Allocation Errors . . . . . 85

5.3 Allocation efficiency . . . . . 86

6.1 F-16 AFTI aircraft control surface position limits [1] . . . . . 102

6.2 Efficiency (area of PAMS / area of TAMS) of each allocator applied to the F-16 AFTI example problem . . . . . 103

6.3 Magnitude and Direction Errors for a Representative Generalized Moment Command . . . . . 105

7.1 F-18 HARV aircraft control surface nomenclature [2] . . . . . 122
7.2 F-18 HARV aircraft control surface position limits [1] . . . . . 122
7.3 Standard errors used in example . . . . . 122
7.4 Standard Deviations in generalized moment . . . . . 125

8.1 T-2 aircraft parametric model uncertainty . . . . . 129
8.2 T-2 control surface deflection limits . . . . . 129
8.3 T-2 Mass and Geometry Properties . . . . . 130

# List of Figures


1.1 Traditional vs. Real-Time Self-Learning Aircraft Development. . . . 3

1.2 Overview of control architecture used in L2F project. . . . . 4

3.1 NASA E1 Aircraft. . . . . 19

3.2 Weighting function: $w_{i,1}$. . . . . 32

3.3 Weighting function: $w_{o,2}$. . . . . 33

3.4 Detailed $H_\infty$ LPV Control Architecture. . . . . 34

3.5 Bode Plots of $W_{1,inner}$ and $W_{2,inner}$. . . . . 36

3.6 Bode Plots of $W_{1,outer}$ and $W_{prefilter}$. . . . . 37

3.7 Guidance tracking of the LPV controller. . . . . 46

3.8 Attitude angles using LPV controller. . . . . 46

3.9 Guidance tracking of the NDI controller. . . . . 47

3.10 Attitude angles using NDI controller. . . . . 47

3.11 Control Surface Deflections with NDI Control. . . . . 48

3.12 Control Surface Deflections with LPV Control. . . . . 49

3.13 Control Surface Frequency Content with NDI Control. . . . . 50

3.14 Control Surface Frequency Content with LPV Control. . . . . 51

4.1 F-16 AFTI aircraft in flight (credit: NASA). . . . . 57

4.2 Allocation efficiency. . . . . 59

4.3 Control allocation error comparison for a single moment command. 60
4.4 Mean error parameters. . . . . 62
4.5 Standard error parameters. . . . . 63
4.6 Block diagram of time-domain simulation. . . . . 64
4.7 Nominal tracking performance for all allocators. . . . . 65
4.8 Difference between NA. . . . . 66
4.9 Nominal control surface deflections. . . . . 66
4.10 Nominal control surface deflections. . . . . 67
4.11 Tracking performance for increased gain case. . . . . 68
4.12 Difference between NA for increased gain case. . . . . 69
4.13 Control surface deflections for increased gain case. . . . . 69
4.14 Control surface deflections for increased gain case. . . . . 70
4.15 Steady-state $\dot{p}$ errors (deg/s$^2$). . . . . 73
4.16 Steady-state $\dot{r}$ errors (deg/s$^2$). . . . . 73

5.1 1st axis moment error. . . . . 85
5.2 2nd axis moment error. . . . . 86
5.3 3rd axis moment error. . . . . 86
5.4 PI Allocation Efficiency. . . . . 87
5.5 MV Allocation Efficiency. . . . . 88
5.6 RLPb Allocation Efficiency. . . . . 89
5.7 RLPe Allocation Efficiency. . . . . 90
5.8 Solution times in seconds with `coneprog` command. . . . . 92

6.1 F-16 AFTI aircraft in flight (credit: NASA). . . . . 102
6.2 Comparison of realized generalized moments for a representative unachievable moment command. . . . . 104

6.3 Relative magnitude errors for the PI and NI allocators. The TAMS of the F-16 AFTI aircraft is shown as a solid red line, and the approximate PAMS of each allocator as a dotted red line. . . . . 106
6.4 Relative magnitude errors for the LP and LPDP allocators. The TAMS of the F-16 AFTI aircraft is shown as a solid red line, and the approximate PAMS of each allocator as a dotted red line. . . . 107
6.5 Direction errors for the PI and NI allocators. The TAMS of the F-16 AFTI aircraft is shown as a solid red line, and the approximate PAMS of each allocator as a dotted red line. . . . . 108
6.6 Direction errors for the LP and LPDP allocators. The TAMS of the F-16 AFTI aircraft is shown as a solid red line, and the approximate PAMS of each allocator as a dotted red line. . . . . 109
6.7 Boxplot of CPU time per call of the LP and LPDP allocators. Boxes show 1st and 3rd quartiles, whiskers cover the range 1.5 times the interquartile range above and below this range, and circles denote outliers. . . . . 111

7.1 NASA F-18 High Angle-of Attack Research Vehicle (HARV) in flight (credit: NASA Armstrong Flight Research Center). . . . . 121
7.2 Efficiency (area of PAMS / area of TAMS) of each allocator applied to the F-18 HARV example problem. . . . . 124
7.3 Magnitude Error PDF . . . . . 125
7.4 Direction Error PDF . . . . . 126

8.1 T-2 aircraft in flight (credit: NASA Langley Research Center). . . . 128
8.2 Aggressive maneuver. . . . . 131
8.3 Nominal maneuver. . . . . 132

8.4 High-level block diagram of closed-loop T-2 simulation. . . . . 132
8.5 Nominal simulation performance. . . . . 133
8.6 Agressive maneuver performance. . . . . 133
8.7 Nominal maneuver performance with uncertainty. . . . . 134
8.8 Nominal maneuver performance with uncertainty. . . . . 135
8.9 Nominal maneuver performance with aileron failure. . . . . 136
8.10 Aggressive maneuver performance with aileron failure. . . . . 136

# Abstract

This dissertation presents the development and simulation of real-time control allocation methods which were specifically designed considering parametric model uncertainty and was performed under the NASA Modeling and Control for Agile Aircraft Development project. Previously, the NASA Learn-to-Fly project successfully showed that combining real-time control design, real-time guidance, and real-time modeling is feasible and can help eliminate the need for ground-based aerodynamic modeling in favor of simultaneous in-flight modeling and control framework development.

The real-time modeling algorithm must obtain flight data from a wide range of explanitory variables. To obtain said data, the real-time modeling procedure uses programmed test inputs to command each of the control surfaces using wide-band frequency content. These programmed test inputs, which are necessary for real time modeling, produce aircraft responses which the control system sees as disturbances and must be suppressed. This creates a fundamental conflict, good control tends to mitigate disturbances, including those due to the programmed test inputs. The basis of these control allocation methods are to separate the control design problem from the control allocation design problem. In this dissertation, the in-flight control design approach is to use loop-shaping techniques based on the rigid-body aircraft equations of motion to penalize control commands within

the programmed test input frequency band. This is shown to successfully control aircraft with control commands more compatible with excitation for real-time modeling.

Due to high initial uncertainty levels, optimal control allocation methods considering parametric model uncertainty are developed and employed. Conventional control allocation methods are unable to guarantee bounds on realized moment error in the presence of bounded disturbances. Control allocation by Robust Linear Programming is developed and employed within this dissertation. The Robust Linear Programming control allocator is derived to realize a moment which is guaranteed to lie within a certain prescribed tolerance bounds in the presence of bounded disturbances.

This dissertation also considers a second optimal control allocation method, the Linear Program Direction-Preserving Allocator. Conventionally, high-efficiency control allocation does not preserve commanded moment direction, this dissertation extends control allocation by linear programming to generate the linear programming direction preserving allocator. The linear programming direction-preserving allocator is then extended using robust control theory to develop the robust linear programming direction-preserving allocator which is shown to best maintain aircraft dynamics in the event of an actuator fault.

The suite of control and control allocation methods is simulated on linear and nonlinear aircraft models vehicles considering parametric model uncertainty. The methods show the improvement on existing algorithms when considering parametric model uncertainty.

# Chapter 1

# Introduction

Control theory plays an instrumental role in engineering systems and is increasingly seen throughout the world. As vehicles mature and become more complex, control theory must advance in order to sufficiently control newer more complex aircraft and spacecraft. The main challenge in control theory lies in understanding what is unknown about the system. Mathematical models of aerospace vehicles are usually empirically-derived which always result in some uncertainty. Additionally, external disturbances on aircraft such as control surface interaction, damaged control effectors, or control surface degradation due to icing can never be fully-known, yet still effect aircraft dynamics.

Several methods exist to effectively control systems subject to parametric model uncertainty. Adaptive control combines an online parameter estimator and automatic control design [4, 5] and formal proofs of stability of adaptive controllers have also been developed [6, 7, 8]. Extensive research has been done in adaptive control and has become a popular method for handling parametric model uncertainties. Unknown information about the system can be quantified into random variables, in the form of parametric model uncertainty [9]. Control of systems

subject to random disturbances is called stochastic control [10]. However, systems could still be subject to rigid control objectives such as minimum-time or minimum-fuel and/or have instabilities while subject to stochastic disturbances. Extensive research has been done on how to effectively control stochastic systems while maintaining performance objectives [10, 11, 12, 13, 14].

Another tenet of flight control theory is control allocation theory. Classical aircraft controls care designed with a single controller for each rotational degree of freedom [15]. In three degrees of freedom this implies there will be one controller for roll, one controller for pitch, and one controller for yaw. However, many aircraft and spacecraft use redundant, or coupled control effectors to achieve different performance objectives. Control allocation is needed to distribute virtual control commands to individual actuators to achieve performance objectives. There has been extensive research and investigation into control allocation methods in previous years, resulting in many effective control allocation methods, such as generalized inverse methods [15], daisy chained control allocation methods [16], linear and quadratic programming control allocation methods, [17]. In recent years, adaptive control allocation has gained more attention as a method to handle actuator failures, as well as parametric model uncertainty [18, 19, 20, 21, 22]. This shows promising developments to handle parametric model uncertainty in the control allocator.

This dissertation utilizes control allocation algorithms which consider parametric model uncertainty. This allows the loop-shaping controller to be designed considering rigid performance objectives and designed based on rigid-body equations of motion and the allocator to account for parametric model uncertainty.

## 1.1 Overview of the Modeling and Control for Agile Aircraft Development Project

Traditionally, flight control laws for new aircraft have been developed by models generated through wind-tunnel testing or computational fluid dynamics, then flight testing a control law, followed by post-flight analysis of data. During post-flight analysis it is often discovered that the data is often insufficient. This results in an iterative process requiring more flight testing which results in a time consuming and costly process to the aircraft developer. The Learn-to-Fly (L2F) project at NASA Langley Research Center successfully combined state-of-the-art real-time aerodynamic modeling, real-time guidance, and learning adaptive control to enable real-time self-learning aircraft. An overview of the traditional versus the real-time self-learning aircraft development process is shown in Fig. 1.1.

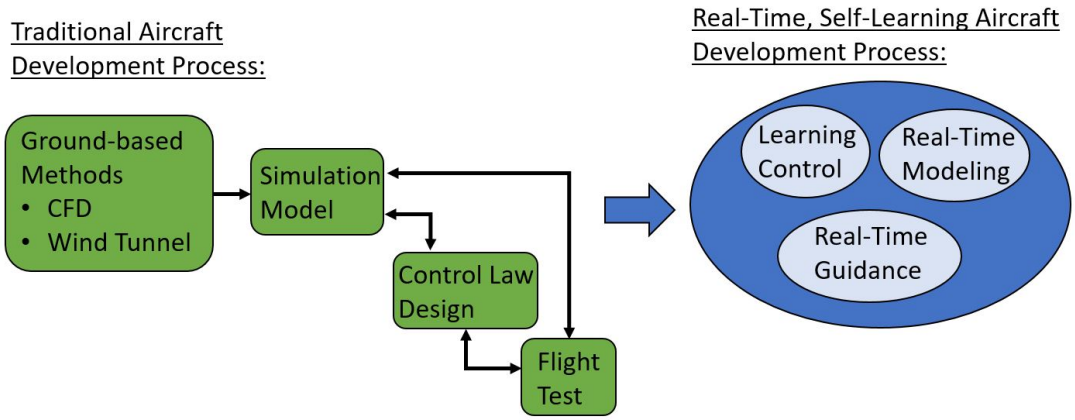



Figure 1.1: Traditional vs. Real-Time Self-Learning Aircraft Development.

The Modeling and Control for Agile Aircraft Development (MCAAD) project at NASA Langley Research Center seeks to improve upon the learning control framework developed under the L2F project. Initially, the only available model of the aircraft aerodynamics is a guess, which may be very inaccurate. The controller

must be able to ensure stability with this potentially poor model. The controller developed under the L2F project successfully ensured vehicle stability but was not specifically designed to be compatible with the real-time modeling procedure. MCAAD seeks to improve upon this control design by developing a controller specifically designed to be compatible with the real-time modeling procedure.

## 1.2 Control Principles

The L2F project successfully demonstrated the feasibility of simultaneous real-time modeling, real-time guidance, and learning control to develop mathematical models and control laws for aircraft with little or no ground-based testing [23]. This was done through usage of a baseline nonlinear dynamic inversion (NDI)-based baseline control law with adaptive $\mathcal{L}_1$ augmentation which actively assisted the NDI in regulating angular rates [23]. Figure 1.2 summarizes the control architecture

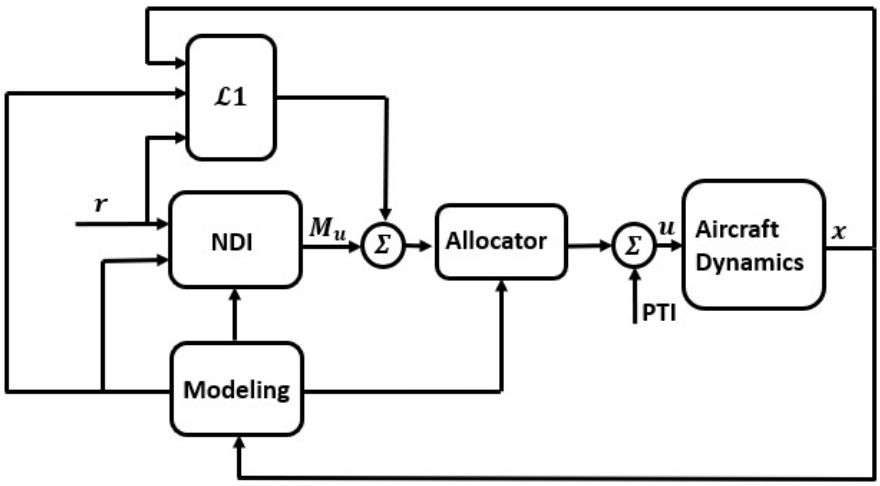



Figure 1.2: Overview of control architecture used in L2F project.

used in the L2F project. However, the L2F control allocator was not specifically designed considering parametric model uncertainty.

This dissertation develops an alternative flight control architecture shown to be specifically compatible with parametric model uncertainty. The principles of this architecture are based on the nonlinear aircraft dynamics expressed in general form:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(t, \boldsymbol{x}, \boldsymbol{\tau}) \tag{1.1}$$

$$\boldsymbol{\tau} = \boldsymbol{h}(t, \boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta}) \tag{1.2}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the state vector, $\boldsymbol{u} \in \mathbb{R}^r$ is the control surface deflection vector, $\boldsymbol{\tau} \in \mathbb{R}^p$ is the moment vector, and $t \geq 0$ is time. $\boldsymbol{\theta} \in \mathbb{R}^q$ is a vector of uncertain parameters. The real-time modeling procedure provides Gaussian estimates as shown by Eq. 1.3. More information about the real-time modeling procedure can be found in Ref. [9].

$$\hat{\boldsymbol{\theta}}_i \sim \mathcal{N}(\boldsymbol{\mu_i}|\boldsymbol{R}_i) \forall i = \{1, 2, ..., q\} \tag{1.3}$$

The fully-known rigid body aircraft dynamics are represented by Eq. (1.1), whereas the unknown moment model is represented by Eq. (1.2). In this dissertation, the baseline control law is an $H_\infty$-based linear parameter-varying (LPV) control law and is designed based on the fully-known rigid-body equations of motion. This allows the control designer to use loop-shaping techniques to shape moment commands into that more compatible with excitation due to real-time modeling. The control allocation in this dissertation is designed based on the moment model in Eq. (1.2), which is subject to uncertainty.

## 1.3 Contributions and Organization

The contributions of this dissertation are as follows:

1) **Show aircraft control based on rigid-body aircraft dynamics allows**

**the control designer to use loop-shaping techniques to shape control commands into excitation compatible with real-time modeling.**

2) **Show how control allocation by Robust Linear Programming can improve vehicle performance due to effects of parametric model uncertainty.**

3) **Extend control allocation by linear programming to preserve realized generalized moment direction which develops the linear programming direction-preserving allocator.**

4) **Extend the linear programming direction preserving allocator to be robust against parametric model uncertainties which develops the Robust Linear Programming Direction Preserving Allocator.**

The organization of this dissertation is detailed for convenience.

Chapter 2 provides an overview of mathematical preliminaries and control allocation methods used throughout the remainder of the dissertation.

Chapter 3 shows the derivation of an $H_\infty$-based LPV controller used to penalize control commands within the PTI frequency band, with the control allocator used in the L2F project. This control approach can also shape the moment commands into that more desirable for real-time modeling. However, this control approach introduces tracking performance worse than that of the L2F control architecture.

Chapter 4 shows a comparison of existing control allocation methods in the presence of parametric model uncertainty.

Chapter 5 shows the derivation and performance analysis of the Robust Linear Programming allocator.

Chapter 6 shows the derivation of the Linear Programming Direction-Preserving Allocator which can provide high allocation efficiency while preserving commanded moment direction.

Chapter 7 shows the derivation and performance benefits of the Robust Linear-Programming Direction-Preserving allocator.

Chapter 8 shows the efficacy of the control algorithms developed in chapters 5, 6, and 7 on a nonlinear, closed-loop T-2 aircraft example.

Lastly, chapter 9 provides concluding remarks along with recommendations for future research.

# Chapter 2

# Preliminaries

This chapter presents a brief overview of the mathematical principles and control allocation methods used in this dissertation.

## 2.1 Control Allocation

This section reviews the control allocation methods used in this dissertation. A convex representation of flight vehicle equations of motion is:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{\tau}(\boldsymbol{\theta}, \boldsymbol{u}) \tag{2.1}$$

where, $\boldsymbol{x} \in \mathbb{R}^n$ is the state vector, $\boldsymbol{\theta} \in \mathbb{R}^q$ is a vector of uncertain parameters, $\boldsymbol{u} \in \mathbb{R}^r$ is the control surface deflection vector, $\boldsymbol{f}$ and $\boldsymbol{\tau}$ are $\in \mathbb{R}^p$ and represent virtual controls resulting from vehicle states and vehicle control surfaces, respectively. In cases where the vehicle control surfaces are linear in the equations of motion the last term in Eq. (2.1) can be linearized to

$$\boldsymbol{\tau} = \boldsymbol{B}(\boldsymbol{\theta})\boldsymbol{u} \tag{2.2}$$

A control system commands a virtual control command, $\boldsymbol{\tau}_c$ to be imparted to the vehicle control surfaces. The mapping between the virtual control commands and the control surfaces is the control allocation problem.

### 2.1.1 Generalized Inverse Control Allocation

Generalized inverse control allocation poses several advantages, they are simple to implement and the control allocator can be solved offline and does not require the use of a convex optimization solver in real-time. However, generalized inverses cannot access all attainable moments, even if there is a set of admissible controls which satisfies the constraints. Generalized inverse control allocation is derived as the solution to a Lagrange optimization problem

$$\min_{\boldsymbol{u}} \mathcal{L} = J(\boldsymbol{u}) + \boldsymbol{\lambda}^{\boldsymbol{T}}(\boldsymbol{\tau}_c - \boldsymbol{B}(\boldsymbol{\theta})\boldsymbol{u}) \tag{2.3}$$

where $J(\boldsymbol{u})$ is a scalar cost function and $\boldsymbol{\lambda} \in \mathbb{R}^p$ is a Lagrange multiplier. In this chapter, the PI and MV allocators are used, each allocator is a solution to Eq. (2.3) with a respective cost function. Grauer and Pei found that the MV allocator is a specific form of pseudo-inverse with a different weighting term, $\boldsymbol{W}$ [11]. Equation (2.4) is the cost function used to derive the generalized inverse allocators.

$$J(\boldsymbol{u}) = \frac{1}{2}\boldsymbol{u}^T\boldsymbol{W}\boldsymbol{u} \tag{2.4}$$

the variable, $\boldsymbol{W}$ is a weighting function. The final form of the generalized inverse allocators are

$$\boldsymbol{u} = \boldsymbol{P}\boldsymbol{\tau}_c \tag{2.5}$$

where,

$$\boldsymbol{P} = \boldsymbol{W}^{-1}\boldsymbol{B}^T\left(\boldsymbol{B}\boldsymbol{W}^{-1}\boldsymbol{B}^T\right)^{-1} \tag{2.6}$$

The pseudo-inverse is the generalized inverse with $\boldsymbol{W} = \boldsymbol{I}_{r\times r}$. The minimum-variance is the generalized inverse with $\boldsymbol{W} = \mathbb{E}[\boldsymbol{B}^T(\hat{\boldsymbol{\theta}})\boldsymbol{B}(\hat{\boldsymbol{\theta}})]$. More information on the generalized inverse derivation can be found in Ref. [11].

### 2.1.2 Control Allocation by Linear Programming

A common formulation of the control allocation problem is to minimize the 1-norm of the allocation error [24, 25] and then transform the objective into a standard form linear-programming problem. This allows the problem to be solved in real-time using linear programming solvers which are commonly available in commercial software. The problem is set up as a mixed optimization problem where the designer can satisfy multiple optimization objectives. To begin, consider the following 1-norm mixed optimization problem [25].

$$\min_{\boldsymbol{u}} \quad \boldsymbol{J}(\boldsymbol{u}) = ||\boldsymbol{B}(\boldsymbol{\theta})\boldsymbol{u} - \boldsymbol{\tau}_c||_1 + \epsilon||\boldsymbol{u} - \boldsymbol{u}_0||_1 \tag{2.7a}$$

$$\text{s.t.} \quad \underline{\boldsymbol{u}} \leq \boldsymbol{u} \leq \overline{\boldsymbol{u}} \tag{2.7b}$$

The parameter, $\epsilon$ is a weighting parameter which allows the control designer to choose between a solution which is close to preferred control surface deflections, or prioritizing minimizing moment error. In this dissertation $\epsilon = 1 \times 10^{-6}$. To handle rate constraints, an additional term can be included to ensure the difference between the current and previous solution does not exceed a certain threshold. The

1-norm constraints in Eq. (2.7) can be written as,

$$||\boldsymbol{B}(\boldsymbol{\theta})\boldsymbol{u} - \boldsymbol{\tau}_c||_1 = \sum_{i=1}^{p} |\boldsymbol{B}(\boldsymbol{\theta})_i\boldsymbol{u} - \tau_{c,i}| \tag{2.8a}$$

$$||\boldsymbol{u} - \boldsymbol{u}_0||_1 = \sum_{j=1}^{r} |u_j - u_{p,j}| \tag{2.8b}$$

the subscript, $i$ denotes the $i^{th}$ row. The subscript, $j$ denotes the $j^{th}$ effector. Setting an upper bound, $s_i$ on the allocation error, $|\boldsymbol{B}(\boldsymbol{\theta})_i\boldsymbol{u} - \tau_{c,i}| \leq s_i$ and an upper bound, $q_j$ on the preferred deflection, $|u_j - u_{p,j}| \leq q_j$.

$$\min_{\boldsymbol{u},\boldsymbol{s},\boldsymbol{q}} \quad \boldsymbol{J} = \begin{bmatrix} 0\ldots0 & 1\ldots1 & \epsilon\ldots\epsilon \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{s} \\ \boldsymbol{q} \end{bmatrix} \tag{2.9a}$$

$$\text{s.t.} \quad \underline{\boldsymbol{u}} \leq \boldsymbol{u} \leq \overline{\boldsymbol{u}} \tag{2.9b}$$

$$|\boldsymbol{B}(\boldsymbol{\theta})_i\boldsymbol{u} - \tau_{c,i}| \leq s_i \quad \forall i \tag{2.9c}$$

$$-\boldsymbol{s} \leq \boldsymbol{0} \tag{2.9d}$$

$$|u_j - u_{p,j}| \leq q_j \quad \forall j \tag{2.9e}$$

The constraints in Eqs. (2.9b), (2.9c), and (2.9e) do not follow the standard form for linear programming solvers. The constraints are rewritten in the following form,

$$\min_{\boldsymbol{u},\boldsymbol{s},\boldsymbol{q}} \quad \boldsymbol{J} = \begin{bmatrix} 0\ldots0 & 1\ldots1 & \epsilon\ldots\epsilon \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{s} \\ \boldsymbol{q} \end{bmatrix} \tag{2.10a}$$

$$\text{s.t.} \quad -\boldsymbol{u} \leq -\underline{\boldsymbol{u}} \tag{2.10b}$$

$$\boldsymbol{u} \leq \overline{\boldsymbol{u}} \tag{2.10c}$$

$$\boldsymbol{B}(\boldsymbol{\theta})_i \boldsymbol{u} - \tau_{c,i} \leq s_i \quad \forall i \tag{2.10d}$$

$$\boldsymbol{B}(\boldsymbol{\theta})_i \boldsymbol{u} - \tau_{c,i} \geq -s_i \quad \forall i \tag{2.10e}$$

$$-\boldsymbol{s} \leq \boldsymbol{0} \tag{2.10f}$$

$$u_j - u_{p,j} \leq q_j \quad \forall j \tag{2.10g}$$

$$u_j - u_{p,j} \geq -q_j \quad \forall j \tag{2.10h}$$

and further refining the optimization to yield the final form of the linear programming problem:

$$\min_{\boldsymbol{u},\boldsymbol{s},\boldsymbol{q}} \quad \boldsymbol{J} = \begin{bmatrix} 0 \ldots 0 & 1 \ldots 1 & \epsilon \ldots \epsilon \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{s} \\ \boldsymbol{q} \end{bmatrix} \tag{2.11a}$$

$$\text{s.t.} \quad -\boldsymbol{u} \leq -\underline{\boldsymbol{u}} \tag{2.11b}$$

$$\boldsymbol{u} \leq \overline{\boldsymbol{u}} \tag{2.11c}$$

$$\boldsymbol{B}(\boldsymbol{\theta})_i \boldsymbol{u} - s_i \leq \tau_{c,i} \quad \forall i \tag{2.11d}$$

$$-\boldsymbol{B}(\boldsymbol{\theta})_i \boldsymbol{u} - s_i \leq -\tau_{c,i} \quad \forall i \tag{2.11e}$$

$$-\boldsymbol{s} \leq \boldsymbol{0} \tag{2.11f}$$

$$u_j - q_j \leq u_{p,j} \quad \forall j \tag{2.11g}$$

$$-u_j - q_j \leq -u_{p,j} \quad \forall j \tag{2.11h}$$

The vector of uncertain parameters, $\boldsymbol{\theta}$ belongs to a given set $\boldsymbol{\theta} \in \Theta$. In general, Eq. (2.11d) and Eq. (2.11e) have infinitely many constraints corresponding to all possible realizations of the uncertainty, $\boldsymbol{\theta}$. The problem can be solved considering a nominal value, $\hat{\boldsymbol{\theta}}$ as in chapter 4, but the solution would not be robust. Chapter 5

formulates the problem by robust control methods. The problem can solved in real-time using gradient-based, active-set algorithms or interior-point algorithms[26].

### 2.1.3 Control Allocation by Quadratic Programming

Another common formulation of the control allocation problem is to formulate the objective function as a mixed-objective 2-norm optimization problem [17]. The resultant optimization problem will transform into a quadratic programming problem, which similar to the previous section, could be solved with readily-available commercial software. To review control allocation by quadric programming begin by considering the following mixed-objective 2-norm optimization problem [17].

$$\min_{\boldsymbol{u}} \quad \boldsymbol{J}(\boldsymbol{u}) = ||\boldsymbol{B}(\boldsymbol{\theta})\boldsymbol{u} - \boldsymbol{\tau}_c||_2^2 + \epsilon||\boldsymbol{u} - \boldsymbol{u}_0||_2^2 \tag{2.12a}$$

$$\text{s.t.} \quad \underline{\boldsymbol{u}} \leq \boldsymbol{u} \leq \overline{\boldsymbol{u}} \tag{2.12b}$$

The cost function in Eq. (2.12) can be expanded to

$$\begin{aligned} \min_{\boldsymbol{u}} \quad \boldsymbol{J}(\boldsymbol{u}) &= (\boldsymbol{B}(\boldsymbol{\theta})\boldsymbol{u} - \boldsymbol{\tau}_c)^T(\boldsymbol{B}(\boldsymbol{\theta})\boldsymbol{u} - \boldsymbol{\tau}_c) \\ &\quad + \epsilon(\boldsymbol{u} - \boldsymbol{u}_0)^T(\boldsymbol{u} - \boldsymbol{u}_0) \end{aligned} \tag{2.13a}$$

$$\text{s.t.} \quad \underline{\boldsymbol{u}} \leq \boldsymbol{u} \leq \overline{\boldsymbol{u}} \tag{2.13b}$$

For brevity, define the following variables

$$\boldsymbol{H}(\boldsymbol{\theta}) = 2\Big(\boldsymbol{B}(\boldsymbol{\theta})^T\boldsymbol{B}(\boldsymbol{\theta}) + \epsilon\boldsymbol{I}_{r\times r}\Big) \tag{2.14a}$$

$$\boldsymbol{c}^T(\boldsymbol{\theta}) = -2\Big(\boldsymbol{\tau}_c^T\boldsymbol{B}(\boldsymbol{\theta}) + \epsilon\boldsymbol{u}_0^T\Big) \tag{2.14b}$$

$$k = \boldsymbol{\tau}_c^T\boldsymbol{\tau}_c + \epsilon\boldsymbol{u}_0^T\boldsymbol{u}_0 \tag{2.14c}$$

Since $k$ is constant in $\boldsymbol{u}$ it can be dropped from the objective function. The optimization problem in Eq. (2.13) now reduces to,

$$\min_{\boldsymbol{u}} \quad \boldsymbol{J}(\boldsymbol{u}) = \frac{1}{2}\boldsymbol{u}^T\boldsymbol{H}(\boldsymbol{\theta})\boldsymbol{u} + \boldsymbol{c}^T(\boldsymbol{\theta})\boldsymbol{u} \tag{2.15a}$$

$$\text{s.t.} \quad \underline{\boldsymbol{u}} \leq \boldsymbol{u} \leq \overline{\boldsymbol{u}} \tag{2.15b}$$

Similar to Eq. (2.15) is also solved considering a nominal value, $\hat{\boldsymbol{\theta}}$ and solved using the `quadprog` `MATLAB` command. Equation (2.15a) also has infinitely many constraints corresponding to all possible realizations of the uncertainty, $\boldsymbol{\theta}$. The problem can also be solved in real-time using gradient-based, active-set algorithms or interior-point algorithms [26].

**Example Application, Generalized Inverse:**

The following commanded generalized aircraft moment is provided:

$$\boldsymbol{\tau}_c = \begin{bmatrix} C_l \\ C_n \end{bmatrix} = \begin{bmatrix} 1.0 \times 10^{-2} \\ 0 \end{bmatrix} = \boldsymbol{B}\boldsymbol{u} \tag{2.16}$$

Where, $C_l$ and $C_n$ are the non-dimensional aircraft rolling and yawing moment coefficients, respectively. $\boldsymbol{B}$ represents the control derivatives matrix and is taken from the NASA Dryden "Controls Design Challenge" model, evaluated an $M = 0.5$, $h =$10 000 ft [27]. The control effectors considered are ailerons, $u_a$, a horizontal tail $u_{HT}$ and rudder, $u_r$ control effectiveness is expressed as per degree.

$$\begin{aligned} \boldsymbol{B} &= \begin{bmatrix} C_{l_{\delta_a}} & C_{l_{\delta HT}} & C_{l_{\delta_r}} \\ C_{n_{\delta_a}} & C_{n_{\delta HT}} & C_{n_{\delta_r}} \end{bmatrix} \\ &= \begin{bmatrix} 7.35 \times 10^{-4} & 7.55 \times 10^{-4} & -1.35 \times 10^{-4} \\ 8.56 \times 10^{-5} & 5.13 \times 10^{-4} & -1.37 \times 10^{-3} \end{bmatrix} \end{aligned} \tag{2.17}$$

$$\boldsymbol{u} = \begin{bmatrix} u_a \\ u_{HT} \\ u_r \end{bmatrix} \tag{2.18}$$

The control allocation task is to solve the system of nonlinear algebraic equations in Eq. (2.16) subject to $\mathbb{E}[\boldsymbol{\tau}] = \boldsymbol{\tau}_c$. The solution to the control allocation problem is non-unique and one introduces a cost function to determine a particular solution of the control allocation problem.

The control limits are $u_a$±20deg, $u_{HT}$± 20deg, and $u_r$±30deg. We use the pseudo-inverse derived in section 2.1.1 to solve for the control deflections, $\boldsymbol{u}$ to realize the requested moment $\boldsymbol{\tau}_c$.

$$\boldsymbol{P} = \boldsymbol{B}^T\left(\boldsymbol{B}\boldsymbol{B}^T\right)^{-1} \tag{2.19}$$

Notice that:

$$\begin{aligned} \boldsymbol{B}\boldsymbol{P}\boldsymbol{\tau}_c &= \boldsymbol{B}\boldsymbol{u} \\ \boldsymbol{\tau}_c &= \boldsymbol{B}\boldsymbol{u} \end{aligned} \tag{2.20}$$

$$\boldsymbol{u} = \boldsymbol{P}\boldsymbol{\tau}_c = \begin{bmatrix} 7.5450\,\text{deg} \\ 6.4136\,\text{deg} \\ 2.8730\,\text{deg} \end{bmatrix} \tag{2.21}$$

Notice that $\boldsymbol{u}$ does not exceed the predefined control deflection limits, thus the moment is realizable. Cases where the moment is unrealizable are reviewed in Ref. [15].

## 2.2 Dual Norm

A critical preliminary in this dissertation, which is discussed in chapter 5 and chapter 7 is the definition of the Dual Norm. A complete review of the Dual Norm can be found in Ref. [28, 29].

**Definition**: The dual norm of $||\boldsymbol{x}||$ is $||\boldsymbol{x}||_*$ where,

$$||\boldsymbol{x}||_* = \max_{||\boldsymbol{z}||\leq 1} \boldsymbol{z}^T\boldsymbol{x} \tag{2.22}$$

The dual norm obeys the relation where: the dual of the $p$ norm is the $q$ norm, where,

$$\frac{1}{p} + \frac{1}{q} = 1 \tag{2.23}$$

## 2.3 Second-Order Cone Programs

A subclass of convex-optimization problems are second-order cone programming problems (SOCP). SOCP problems can be solved in polynomial-time by existing interior-point methods [30, 31, 32]. SOCP problems take the form of

$$\min_{\boldsymbol{x}} \quad \boldsymbol{f}^T\boldsymbol{x} \tag{2.24a}$$

$$\text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} \tag{2.24b}$$

$$\boldsymbol{A}_{eq}\boldsymbol{x} = \boldsymbol{b}_{eq} \tag{2.24c}$$

$$||\boldsymbol{A}^i_{soc}\boldsymbol{x} - \boldsymbol{b}^i_{soc}||_2 \leq \boldsymbol{c}^{i,T}\boldsymbol{x} + d^i \tag{2.24d}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ represents the optimization variables and $\boldsymbol{f} \in \mathbb{R}^n$ represents a cost function. Inequality constraints are represented by the matrix $\boldsymbol{A} \in \mathbb{R}^{p\times n}$ and the vector $\boldsymbol{b} \in \mathbb{R}^p$. Equality constraints are represented by the matrix $\boldsymbol{A}_{eq} \in \mathbb{R}^{m\times n}$

and the vector $\boldsymbol{b}_{eq} \in \mathbb{R}^m$. The $i$-th second-order cone constraint is defined by the matrix $\boldsymbol{A}^i_{soc} \in \mathbb{R}^{k \times n}$ and the vectors, $\boldsymbol{b}^i_{soc} \in \mathbb{R}^k$ and $c^i \in \mathbb{R}^n$ and the scalar, $d^i$. More information about SOCP problems can be found in Ref. [33].

## 2.4 Summary

This chapter detailed the fundamental governing equations used throughout the remainder of this dissertation. The main results have been presented via discussion of Control Allocation, the Dual Norm, and Second-Order Cone Programs.

# Chapter 3

# Linear Parameter-Varying Control Compatible with Excitation for Real-Time Modeling

The efficacy of this work is demonstrated on a simulation of the NASA E1 aircraft which is depicted in Fig. 3.1. The E1 aircraft is a commercially-available 40-percent scale Edge 330 SC remotely-piloted fixed-wing airplane. Control surfaces are conventional ailerons and trailing-edge flaps on the wings, along with a conventional rudder and a split elevator. E1 is powered by an electric motor driving a fixed-pitch tractor propeller. More information about E1 can be found in [34, 35, 36].




Figure 3.1: NASA E1 Aircraft.

## 3.1 Introduction

This chapter presents an $H_\infty$-based Linear Parameter-Varying (LPV) controller which can serve to penalize control commands within the PTI frequency band and be more compatible with excitation due to real-time modeling. Secondly, the controller in this chapter presents can serve as the exponentially stable trajectory-tracking controller necessary for the development of optimal nonlinear control allocators [37] and adaptive control allocation algorithms [18].

In this chapter an LPV control design methodology is presented based on rigid-body equations of motion and is shown to be more compatible with real-time modeling methods than the NDI control developed under the L2F project.

This technique could also be applied to aircraft models subject to more complex dynamic effects, such as slosh or flex modes. The LPV technique is an alternative to the traditional gain scheduling process. The dynamic model and controller are still linear, but they are dependent on time-varying parameters whose values can

be measured, but are unknown in advance. A traditional gain-scheduling approach involves linearizing dynamic models at many trim points and then designing linear time invariant (LTI) controllers at each trim point. The global controller is then estimated by interpolating the gains of the local controllers. This approach has been successfully implemented in many engineering applications but there is no guarantee of exponential stability [38]. The LPV control strategy can show exponential stability for all possible values of the parameters and can be implemented like a traditionally gain-scheduled control system [39] [40].

The LPV control strategy requires an affine dynamic system that must be converted into convex polytopic form [39]. The rigid-body aircraft model is not affine and will result in an infinite number of Linear Matrix Inequalities (LMIs) to determine the controller [39]. This difficulty is overcome using the Tensor Product (TP) model transformation to convert the nonaffine system into convex polytopic form. There is then a linear control allocator generated from a linearized system model produced by the real-time modeling procedure. The control allocator commands control surface deflections such that the moment commanded by the controller is realized. The newly developed LPV control law was tested on a nonlinear simulation of the NASA E1 aircraft, depicted in Fig. 3.1.

## 3.2 Excitation for Real-Time Modeling

In order for the real-time modeling procedure to obtain a nonlinear global aircraft model, the flight data must include a wide range of values for the explanatory variables. The explanatory variables include, angle-of-attack, sideslip angle, nondimensional body-axis angular rates, and control surface positions. To excite motion across a wide range of explanitory variables, Programmed Test Inputs (PTI)

are developed and applied as described in [34] which simultaneously command each of the aircraft control surfaces using inputs with wide-band frequency content. Each perturbation $u_j$ applied to the $j^{th}$ individual control surface is a sum of selected harmonic sinusoids with individual phase shifts, $\phi_k$. The frequencies of these harmonic sinusoids are listed in Table 3.1.

$$u_j = \sum_{k \in \{1,2,\ldots,M\}} A_k \sin\left(\frac{2\pi k t}{T} + \phi_k\right) \tag{3.1}$$

The integer $M$ is the number of available harmonically-related frequencies. $T$ is the time length of excitation, $A_k$ are the amplitudes for the sinusoidal components, and $t$ is time. The design of each multisine input is determined by methods shown in [34]. The real-time modeling procedure is also described in detail in [34].

Table 3.1: PTI frequencies for each effector (Hz)

| L. Flap | R. Flap | L. Ail. | R. Ail. | Elev. | Rud. |
|---|---|---|---|---|---|
| .1250 | .1750 | .1000 | .1500 | .0750 | .0500 |
| .2750 | .3250 | .2500 | .3000 | .2250 | .2000 |
| .4250 | .4750 | .4000 | .4500 | .3750 | .3500 |
| .5750 | .6250 | .5500 | .6000 | .5250 | .5000 |
| .7250 | .7750 | .7000 | .7500 | .6750 | .6500 |
| .8750 | .9250 | .8500 | .9000 | .8250 | .8000 |
| 1.0250 | 1.0750 | 1.0000 | 1.0500 | .9750 | .9500 |
| 1.1750 | 1.2250 | 1.1500 | 1.2000 | 1.1250 | 1.1000 |
| 1.3250 | 1.3750 | 1.3000 | 1.3500 | 1.2750 | 1.2500 |
| 1.4750 | 1.5250 | 1.4500 | 1.5000 | 1.4250 | 1.4000 |
| 1.6250 | 1.6750 | 1.6000 | 1.6500 | 1.5750 | 1.5500 |

The L2F control system was designed without regard to the PTI excitations, which are necessary for real-time modeling. Therefore the aircraft control and the PTI are in conflict, the control might command a control surface deflection which could distort a PTI command.

## 3.3 LPV Model Derivation

The MCAAD project presents a challenge in real-time control design: the aircraft must be stabilized and controlled without a predefined model. However, a real-time modeling procedure can be performed to generate a global nonlinear aerodynamic model. The real-time modeling procedure uses PTI to excite the aircraft response to produce informative data. Thus the controller must be effective enough to stabilize and ensure safety of the aircraft, but not suppress the response of the aircraft due to the PTI. This is accomplished through a multi-loop control architecture design. The outer loop receives a guidance command $\boldsymbol{r}_o = \begin{bmatrix} \alpha_{cmd} & \beta_{cmd} & \chi_{cmd} \end{bmatrix}^T$ and calculates an attitude command state $\boldsymbol{r}_i = \begin{bmatrix} \phi_{cmd} & \omega_{2,cmd} & \omega_{3,cmd} \end{bmatrix}^T$ by the outer loop of the LPV control method described later in this chapter, where $\boldsymbol{\omega} = \begin{bmatrix} \omega_1 & \omega_2 & \omega_3 \end{bmatrix}^T$ are the body-axis angular rates. Additionally, $\theta_{cmd} = \alpha_{cmd}$ and $\omega_{1,cmd} = 0$ are added to the attitude command state. The inner loop then receives the attitude command state and calculates a moment command ($\boldsymbol{\tau}_c$) to be passed to the control allocator. In this section the LPV models used for inner- and outer- loop control design are derived.

A state-space representation of an LTI system is:

$$\begin{aligned} \dot{\boldsymbol{x}} &= A\boldsymbol{x} + B\boldsymbol{u} \\ \boldsymbol{y} &= C\boldsymbol{x} + D\boldsymbol{u} \end{aligned} \tag{3.2}$$

$\boldsymbol{x} \in \mathbb{R}^n$ is the state vector, $\boldsymbol{u} \in \mathbb{R}^q$ is the input vector, and $\boldsymbol{y} \in \mathbb{R}^m$ is the output vector. Where A, B, C, and D denote the system, input, output, and direct-transmission matrices respectively. In general, aircraft dynamics are nonlinear and the LTI representation in Eq. (3.2) is obtained by linearization of a nonlinear model at a specific point in the operating range. Linear control methods can

be applied to stabilize an LTI system, but the performance of the linear controller will deteriorate as the system deviates from the design point. To capture nonlinear effects in the linear system, matrices $A$, $B$, $C$, and $D$ can be described in terms of a parameter vector $\mathbf{p}$ which is unknown in advance, but can be measured in real-time. An example of an element in the parameter vector might be angle of attack, attitude angles, or airspeed. This yields the LPV model of the system as [40]:

$$\begin{aligned} \dot{\boldsymbol{x}} &= A(\mathbf{p})\boldsymbol{x} + B(\mathbf{p})\boldsymbol{u} \\ \boldsymbol{y} &= C(\mathbf{p})\boldsymbol{x} + D(\mathbf{p})\boldsymbol{u} \end{aligned} \tag{3.3}$$

The parameter-dependent LPV controller is defined by:

$$\begin{aligned} \dot{\boldsymbol{x}}_K &= A_K(\mathbf{p})\boldsymbol{x}_K + B_K(\mathbf{p})\boldsymbol{e} \\ \boldsymbol{u} &= C_K(\mathbf{p})\boldsymbol{x}_K + D_K(\mathbf{p})\boldsymbol{e} \end{aligned} \tag{3.4}$$

The value, $\boldsymbol{e}$, is the error signal that is the difference between the command and the output. The LPV dynamic model and the controller are parameter dependent. Therefore, both the LPV dynamic model and the controller are automatically updated based on the measured values for $\mathbf{p}$.

The LPV controllers are derived using Bounded Real Lemma with quadratic $H_\infty$ performance $\Gamma$ [40]. With closed-loop state-space matrices, $A_{cl}(\mathbf{p})$, $B_{cl}(\mathbf{p})$, $C_{cl}(\mathbf{p})$, and $D_{cl}(\mathbf{p})$ the systems have quadratic $H_\infty$ performance $\Gamma$ if and only if there exists a single positive definite matrix, $P$ such that,

$$\begin{bmatrix} A_{cl}^T(\mathbf{p})P + PA_{cl}(\mathbf{p}) & PB_{cl}(\mathbf{p}) & C_{cl}^T(\mathbf{p}) \\ B_{cl}^T(\mathbf{p})P & -\Gamma I & D_{cl}^T(\mathbf{p}) \\ C_{cl}(\mathbf{p}) & D_{cl}(\mathbf{p}) & -\Gamma I \end{bmatrix} < 0 \tag{3.5}$$

for all possible values of $\mathbf{p}$. The $\mathcal{L}_2$ gain between the input and the output is

bounded by $||\boldsymbol{y}||_2 < \Gamma||\boldsymbol{u}||_2$. Quadratic $H_\infty$ performance requires a single fixed quadratic Lyapunov function for the entire operating range [40]. Despite the controllers being designed by solving the linear matrix inequality (LMIs) given in Eq. (3.5) there are an infinite number of solutions, corresponding to the infinite possible trajectories of $\mathbf{p}$. The concept of an Affine Polytopic LPV model has been introduced to overcome this difficulty [38] and will be discussed in section 3.3.1.

The aircraft model used in both LPV control loops is a rigid-body dynamics model which are fully-known across all flight conditions. Use of this rigid-body model renders the controller independent of the real-time modeling algorithm which will help mitigate interactions between the real-time modeling and control. The control allocation method, described later, uses the results of the real-time aerodynamic modeling algorithm to realize the commanded moment by the LPV control loops.

To discuss the development of the LPV models, it is assumed that the aircraft dynamics are governed by the standard rigid-body equations of motion for gliding flight:

## Inner Loop Equations of Motion:

$$\dot{\phi} = \omega_1 + \tan(\theta)(\omega_2 \sin(\phi) + \omega_3 \cos(\phi)) \tag{3.6a}$$

$$\dot{\theta} = \omega_2 \cos(\theta) - \omega_3 \sin(\theta) \tag{3.6b}$$

$$\dot{\boldsymbol{\omega}} = I^{-1}(M + M_u) - I^{-1}(\boldsymbol{\omega} \times I\boldsymbol{\omega}) \tag{3.6c}$$

## Outer Loop Equations of Motion:

$$\dot{\alpha} = \omega_2 - \tan(\beta)(\cos(\alpha)\omega_1 + \sin(\alpha)\omega_1) + \frac{1}{mV\cos(\beta)}(-L + mg\cos(\gamma)\cos(\mu)) \tag{3.7a}$$

$$\dot{\beta} = -\cos(\alpha)\omega_3 + \sin(\alpha)\omega_2 + \frac{1}{mV}(Y\cos(\beta) + mg\cos(\gamma)\sin(\mu)) \tag{3.7b}$$

$$\dot{\gamma} = \frac{1}{mV}(L\cos\mu - Y\sin(\mu)\cos(\beta) - mg\cos(\gamma)) \tag{3.7c}$$

$$\dot{\chi} = \frac{1}{mV\cos(\gamma)}(L\sin(\mu) + Y\cos(\mu)\cos(\beta)) \tag{3.7d}$$

Under the following simplifying assumptions:

- $\mu \approx \phi$,
- $\cos(\alpha) \approx \cos(\gamma) \approx 1$,
- $Y \approx 0$,
- $\beta \approx 0$,
- $L\cos(\phi) \approx mg$,
- $\sin(\alpha)\omega_1 \approx 0$,

The outer loop equations of motion become:

$$\dot{\alpha} = \omega_2 - \frac{g}{V}\sin(\phi)\tan(\phi) \tag{3.8a}$$

$$\dot{\beta} = -\omega_3 + \frac{g}{V}\sin(\phi) \tag{3.8b}$$

$$\dot{\gamma} = 0 \tag{3.8c}$$

$$\dot{\chi} = \frac{g}{V}\tan(\phi) \tag{3.8d}$$

The simplifying assumptions, albeit highly restrictive, simplify the controls problem to a tractable problem for demonstration of the efficacy of the LPV control

law. Additional control laws based on the simplified equations of motion have been successfully demonstrated in flight [41].

Now separate the LPV model into inner and outer control loops. Although the control problem could be solved in a single loop, the two-loop architecture is used to eliminate the need of a post filter which will introduce unnecessary transients into the system [40]. The outer loop receives a guidance command, $\boldsymbol{r}_o = \begin{bmatrix} \alpha_{cmd} & \beta_{cmd} & \chi_{cmd} \end{bmatrix}^T$ and calculates an attitude state command, $\boldsymbol{r}_i = \begin{bmatrix} \phi_{cmd} & \omega_{2,cmd} & \omega_{3,cmd} \end{bmatrix}$. Equations (3.6) and (3.8) are now linearized about an equilibrium point in flight, $\begin{bmatrix} \omega_1 & \omega_2 & \omega_3 \end{bmatrix}^T = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$ and put into LPV form for both the inner and outer loops:

$$\begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\chi} \end{bmatrix} = A_o \begin{bmatrix} \alpha \\ \beta \\ \chi \end{bmatrix} + B_o \begin{bmatrix} \phi \\ \omega_2 \\ \omega_3 \end{bmatrix} \tag{3.9a}$$

$$\begin{bmatrix} \alpha \\ \beta \\ \chi \end{bmatrix} = C_o \begin{bmatrix} \alpha \\ \beta \\ \chi \end{bmatrix} + D_o \begin{bmatrix} \phi \\ \omega_2 \\ \omega_3 \end{bmatrix} \tag{3.9b}$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} = A_i \begin{bmatrix} \phi \\ \theta \\ \boldsymbol{\omega} \end{bmatrix} + B_i M \tag{3.10a}$$

$$\begin{bmatrix} \phi \\ \theta \\ \omega \end{bmatrix} = C_i \begin{bmatrix} \phi \\ \theta \\ \boldsymbol{\omega} \end{bmatrix} + D_i M \tag{3.10b}$$

$$\tag{3.10c}$$

where,

$$A_o = 0_{3\times3} \quad B_o = \begin{bmatrix} -\frac{g}{V}\Big(\cos(\phi)\tan(\phi) + \sin(\phi)\sec^2(\phi)\Big) & 1 & 0 \\ \frac{g}{V}\cos(\phi) & 0 & -1 \\ \frac{g}{V}\sec^2(\phi) & 0 & 0 \end{bmatrix}$$

$$C_o = I_{3\times3} \quad D_o = 0_{3\times3}$$

Assuming $I_{xz} = 0$:

$$A_i = \begin{bmatrix} 0 & 0 & 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & 0 & 0 & \cos(\phi) & -\sin(\phi) \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad B_i = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{I_{xx}} & 0 & 0 \\ 0 & \frac{1}{I_{yy}} & 0 \\ 0 & 0 & \frac{1}{I_{zz}} \end{bmatrix}$$

$$C_i = I_{5\times5} \quad D_i = 0_{5\times3}$$

Defining the parameter vector as the variables that remain in the linearized system, the parameter vectors for the outer loop and inner loop LPV controllers are:

$$\mathbf{p}_o = \begin{bmatrix} \phi & V \end{bmatrix}^T \tag{3.11a}$$

$$\mathbf{p}_i = \begin{bmatrix} \phi & \theta \end{bmatrix}^T \tag{3.11b}$$

### 3.3.1 Affine Polytopic LPV Systems

Assume that each element of the parameter vector, $\mathbf{p} = \begin{bmatrix} p_1 & p_2 & \dots & p_N \end{bmatrix} \in \mathbb{R}^N$ lies between extreme values $\underline{p_i}$ and $\overline{p_i}$ and $\mathbf{p}$ lies in a polytope $\mathfrak{T}$ with vertices $\mathfrak{T}_1$,

$\mathfrak{T}_2, \ldots \mathfrak{T}_m$. With $p \in \mathfrak{T}$ and $m = 2^N$ [42]. In an affine LPV model the system matrices, $S(\mathbf{p})$ are linear functions of the time-varying parameter vector and can be written as a convex combination of the vertex matrices of the parameter polytope, $\mathfrak{T}$ [40].

$$S(\mathbf{p}) = \begin{bmatrix} A(\mathbf{p}) & B(\mathbf{p}) \\ C(\mathbf{p}) & D(\mathbf{p}) \end{bmatrix} = \sum_{i=1}^{m} w_i(\mathbf{p}) \begin{bmatrix} A(\mathfrak{T}_i) & B(\mathfrak{T}_i) \\ C(\mathfrak{T}_i) & D(\mathfrak{T}_i) \end{bmatrix} \tag{3.12}$$

$$\sum_{i=1}^{m} w_i(\mathbf{p}) = 1$$

It can be shown that Eq. (3.5) holds for all possible trajectories of $\mathbf{p}$ if, and only if, it holds at the vertices, $\mathfrak{T}_i$ for all $i = 1, \ldots m$ [40]. This result allows for an infinite number of LMIs to reduce to a finite number of LMIs due to the convexity of the affine LPV system. However, Eqs. (3.9) and (3.10) are not linear functions of the parameter vector, thus the LPV model is not affine. The Tensor Product model transformation is used to generate an affine polytopic LPV system [39].

### 3.3.2 Tensor Product (TP) Transformation

Many control problems can be formulated into the affine LPV form, however the aircraft model in the previous section is in nonaffine form and has to be converted into affine form to obtain a LPV controller with a finite number of LMIs. This conversion process is done through the Tensor Product model transformation based on the Higher Order Singular Value Decomposition of tensors (HOSVD). The TP transformation is given in [42] and reviewed here. Begin by writing the LPV model

as follows:

$$S(\mathbf{p}(t)) = \begin{bmatrix} A(\mathbf{p}) & B(\mathbf{p}) \\ C(\mathbf{p}) & D(\mathbf{p}) \end{bmatrix} \in \mathbb{R}^{(n+q)\times(n+m)} \tag{3.13}$$

The integers $n$, $q$, and $m$ are the number of states, inputs, and outputs respectively. Given bounds on the limits of parameter variations, the parameter space, $P$ is,

$$P = [\underline{p}_1, \bar{p}_1] \times [\underline{p}_2, \bar{p}_2] \times \cdots \times [\underline{p}_N, \bar{p}_N] \tag{3.14}$$

The LPV model is evaluated at grid points in the parameter space and is stored as a tensor $\mathbb{S}^D \in \mathbb{R}^{M_1 \times M_2 \times \cdots \times M_N \times (n+q) \times (n+m)}$ where $M_1$, $M_2$, ..., $M_N$ are the number of grid lines defined in each dimension of the grid. The next step is to apply the HOSVD on the first $N$ dimensions of the of the tensor $S^D$ and discard the zero or small singular values, $\sigma_k$ and their corresponding singular vectors in all dimensions 1, 2, ..., $N$. This is performed using the tool given in [43]. The HOSVD results in a core tensor $\mathbb{S}$ and set of matrices $U_1$, $U_2$, ..., $U_N$ such that $\mathbb{S}^D \approx \mathbb{S} \times U_1 \times U_2 \times U_N$. The matrices $U_1$, $U_2$, ... $U_N$ contain the continuous weighting functions for each bounded variable. For example the matrix $U_n \in \mathbb{R}$ is made of weighting functions for the $n^{th}$ variable, $w_{n,j}(p_n) \in \mathbb{R}^{M_n}$, $j = 1, \ldots, I_n$ and $I_n$ is the number of weighting functions used in the $n^{th}$ dimension of the parameter vector $p_n$. The weighting functions at each vertex can now be defined as,

$$w_r(\mathbf{p}) = \prod_{n=1}^{N} w_{n,j}(p_n) \tag{3.15}$$

The purpose of the TP model transformation is to find a LTI vertex system $S_r$ and

weighting functions, $w_r(\mathbf{p})$ such that the system can be written by the following:

$$S(\mathbf{p}) = \sum_{r=1}^{R} w_r(\mathbf{p}) S_r \tag{3.16}$$

Convex combinations of the $S_r$-LTI vertex system are guaranteed by the following conditions [42]:

$$\forall n \in [1, N], j \in [1, I_n], p_n : w_{n,j}(p_n) \in [0, 1] \tag{3.17}$$

$$\forall n \in [1, N], p_n : \sum_{j=1}^{I_n} w_{n,j}(p_n) = 1 \tag{3.18}$$

$$\forall r \in [1, R], p : w_r(\mathbf{p}) \in [0, 1] \tag{3.19}$$

$$\forall \mathbf{p} : \sum_{r=1}^{R} w_r(\mathbf{p}) = 1 \tag{3.20}$$

The TP method will transform an LPV system, $S(\mathbf{p}(t))$ into a convex hull of LTI vertex systems, $S_r$. This is performed for both the inner and outer loops such that, $S_i(\mathbf{p}(t))$ and $S_o(\mathbf{p}(t))$ are transformed into $S_{i,r}$ and $S_{o,r}$ respectively. The TP convex-polytopic system representation is an approximation and the error, $\mathfrak{e}$ is given as [42]:

$$\left\| S(\mathbf{p}) - \sum_{r=1}^{R} w_r(\mathbf{p}) S_r \right\| = \mathfrak{e} \tag{3.21}$$

During the HOSVD of the inner loop, three singular values were found along the $\phi$ dimension of $S_i(\mathbf{p}(t))$. Two singular values were found along the $\theta$ dimension of $S_i(\mathbf{p}(t))$. This results in six combinations of vertices in the inner loop. The HOSVD of the outer loop found four singular values along the $\phi$ dimension of $S_o(\mathbf{p}(t))$ and four singular values along the V dimension of $S_o(\mathbf{p}(t))$. This results in eight

combinations of vertices the the outer loop. The original non-affine LPV model can now be summed across all vertices and be rewritten in TP-convex polytopic form as:

$$\begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\chi} \end{bmatrix} = \sum_{r=1}^{8} w_{o,r}(\mathbf{p}) B_{o,r} \begin{bmatrix} \phi \\ \omega_2 \\ \omega_3 \end{bmatrix} \tag{3.22a}$$

$$\begin{bmatrix} \alpha \\ \beta \\ \chi \end{bmatrix} = \sum_{r=1}^{8} w_{o,r}(\mathbf{p}) \Big( C_{o,r} \begin{bmatrix} \alpha \\ \beta \\ \chi \end{bmatrix} + D_{o,r} \begin{bmatrix} \phi \\ \omega_2 \\ \omega_3 \end{bmatrix} \Big) \tag{3.22b}$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \sum_{r=1}^{6} w_{i,r}(\mathbf{p}) \Big( A_{i,r} \begin{bmatrix} \phi \\ \theta \\ \omega \end{bmatrix} + B_{i,r} M \Big) \tag{3.23a}$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \sum_{r=1}^{6} w_{i,r}(\mathbf{p}) \Big( C_{i,r} \begin{bmatrix} \phi \\ \theta \\ \omega \end{bmatrix} + D_{i,r} M \Big) \tag{3.23b}$$

Where $A_{i,r}$, $B_{i,r}$, $C_{i,r}$, and $D_{i,r}$ are extracted from the LTI vertices from the inner loop, $S_{i,r}$. $B_{o,r}$, $C_{o,r}$, $D_{o,r}$ is extracted from the LTI vertices of the outer loop, $S_{o,r}$. Several weighting functions, $w_{i,1}$ and $w_{o,2}$ are depicted in Fig. 3.2 and Fig. 3.3. The boundaries of the parameter space are set to the maximum expected values during flight and they are listed in Table 3.2. Lastly, both TP-convex polytopic models were compared with the original LPV models over 1,000 randomly selected points in the parameter space the mean and maximum error are calculated by

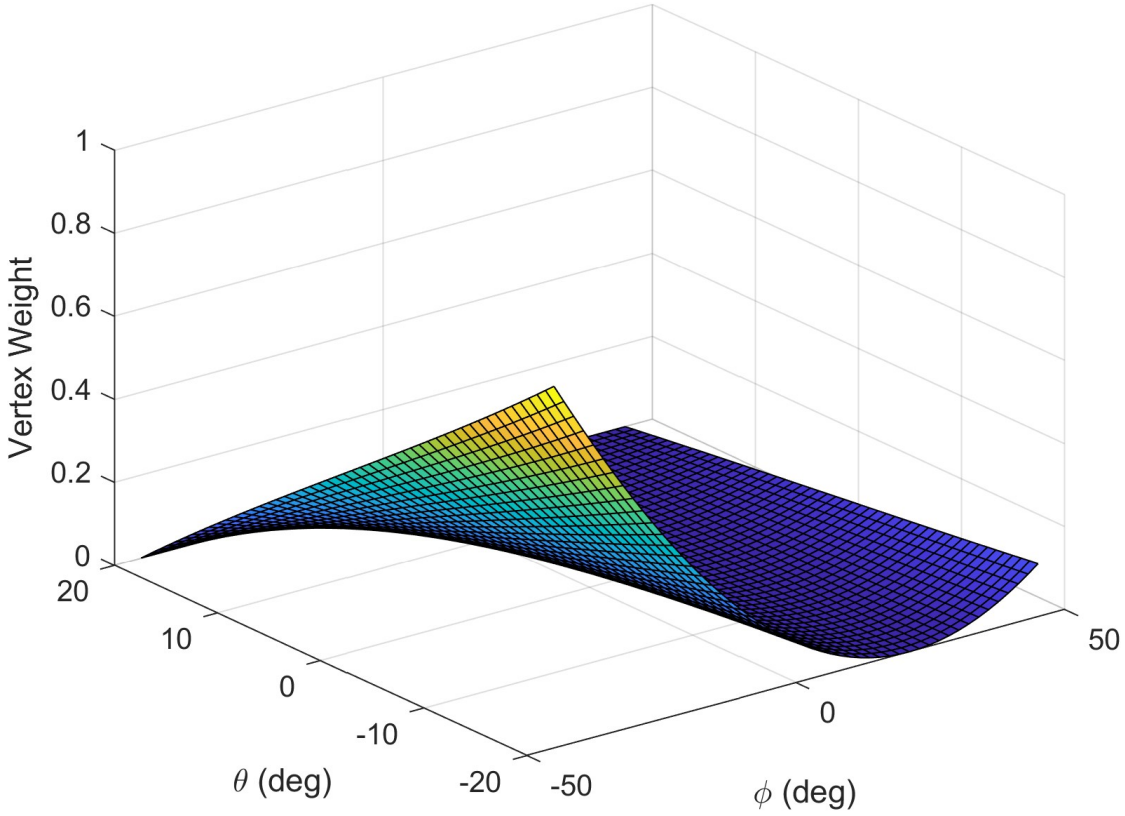



Figure 3.2: Weighting function: $w_{i,1}$.

Table 3.2: Parameter bounds and points

| Parameter | Min. | Max. | Unit | Grid Points |
|---|---|---|---|---|
| $\phi$ | -45 | 45 | deg | 50 |
| $\theta$ | -65 | 65 | deg | 50 |
| $V$ | 65 | 105 | $\mathrm{ft\,s^{-1}}$ | 10 |

Table 3.3: Error Parameters

| Model | Mean Error (%) | Max. Error (%) |
|---|---|---|
| Inner Loop | $3.2055 \times 10^{-5}$ | $6.7593 \times 10^{-5}$ |
| Outer Loop | $5.3495 \times 10^{-5}$ | $9.4605 \times 10^{-5}$ |

Eq. (3.21) and tabulated in Table 3.3. It is shown that the error is small and thus both the TP-convex polytopic models are good representations of the LPV models.

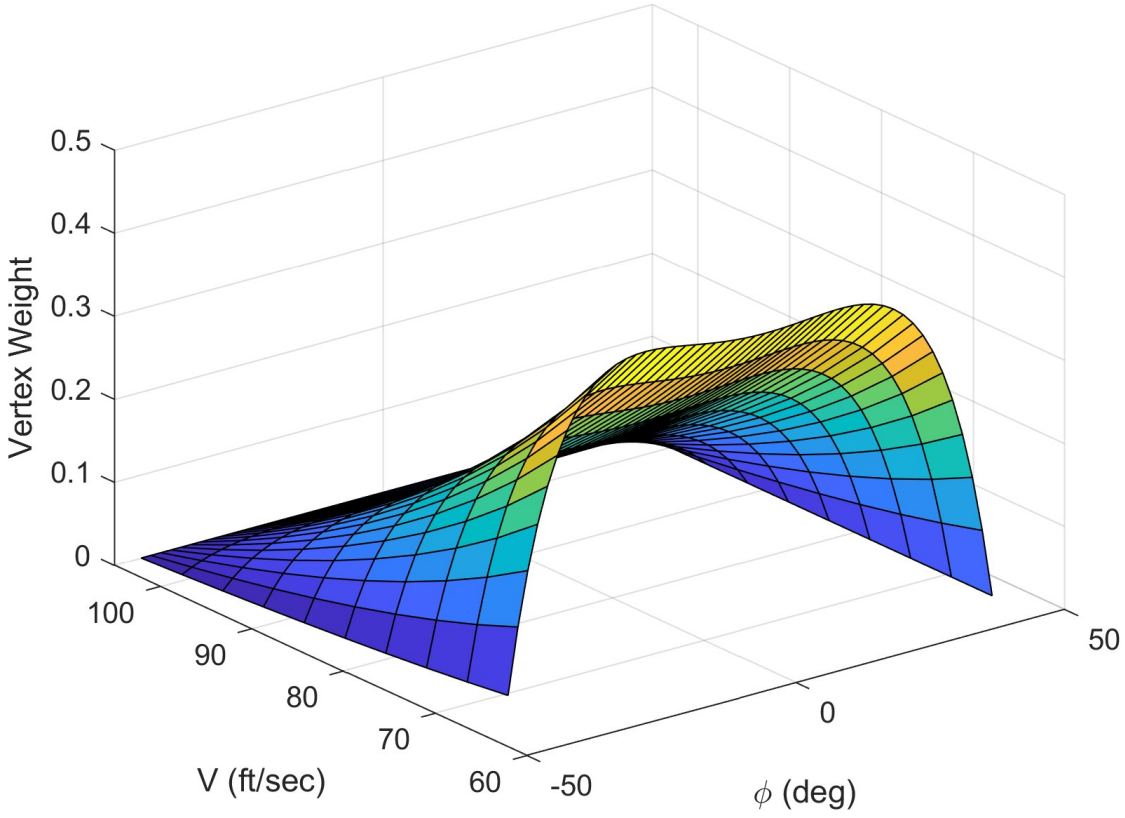



Figure 3.3: Weighting function: $w_{o,2}$.

## 3.4 LPV Control Design

It is desired that the controller track the guidance commands, but also not suppress the response of the aircraft to the PTI excitations. The guidance algorithms are described in detail in [44]. To design the LPV control loops to achieve adequate tracking performance, there are two penalty filters in the inner loop, $W_{1,inner}$ and $W_{2,inner}$, and one penalty filter, $W_{1,outer}$ in the outer loop. The control architecture is built using the `hinfgs` MATLAB routine and shown in Fig. 3.4.

The inner loop controller is designed to penalize moment commands inside the PTI bandwidth, through minimization of the output of $W_{2,inner}$, $Z_{2,inner}$. $W_{2,inner}$ is chosen as a first-order high-pass Bessel filter with cutoff frequency, $\omega_{c,2}$ to penalize control signals within the PTI bandwidth and force the control signals above the

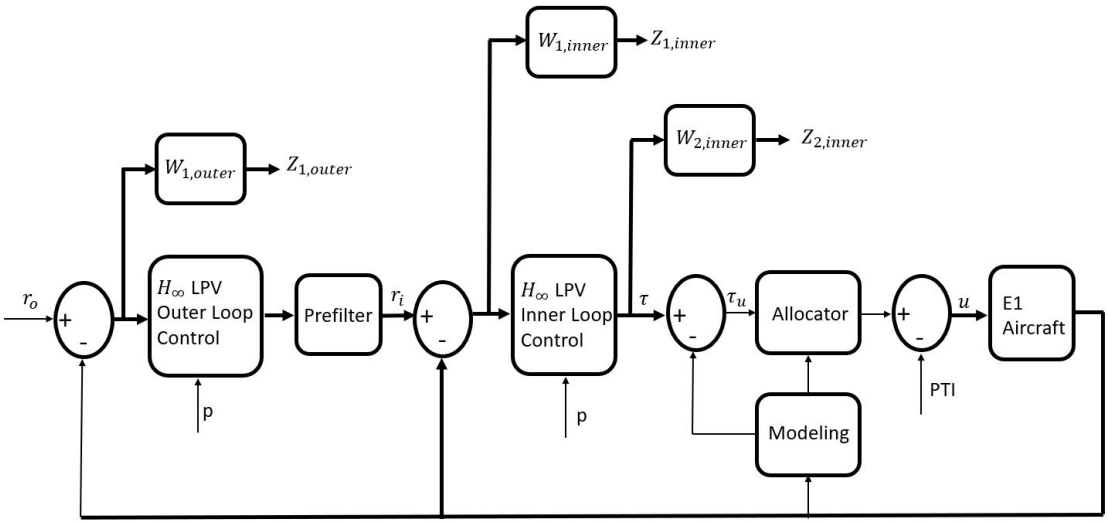



Figure 3.4: Detailed $H_\infty$ LPV Control Architecture.

PTI bandwidth.

$$\omega_{c,2} = 1.6\text{Hz} \tag{3.24}$$

The state-space of the first order, low-pass, Bessel filter is:

$$(A_{2,i}, B_{2,i}, C_{2,i}, D_{2,i}) = (-10.0531, 1, -10.0531, 1) \tag{3.25}$$

The moment vector is a three-dimensional vector and thus the first-order, high-pass, Bessel filter is concatenated into a three-input, three-output filter to form $W_{2,inner}$.

$$W_{2,inner} \begin{cases} \dot{\boldsymbol{x}}_{2,i} = \begin{bmatrix} A_{2,i} & 0 & 0 \\ 0 & A_{2,i} & 0 \\ 0 & 0 & A_{2,i} \end{bmatrix} \boldsymbol{x}_{2,i} + \begin{bmatrix} B_{2,i} & 0 & 0 \\ 0 & B_{2,i} & 0 \\ 0 & 0 & B_{2,i} \end{bmatrix} \boldsymbol{\tau}_c \\ Z_{2,inner} = \begin{bmatrix} C_{2,i} & 0 & 0 \\ 0 & C_{2,i} & 0 \\ 0 & 0 & C_{2,i} \end{bmatrix} \boldsymbol{x}_{2,i} + \begin{bmatrix} D_{2,i} & 0 & 0 \\ 0 & D_{2,i} & 0 \\ 0 & 0 & D_{2,i} \end{bmatrix} \boldsymbol{\tau}_c \end{cases} \tag{3.26}$$

To ensure adequate response speed, a first-order, low-pass, Bessel filter is put on the tracking commands in the inner-loop. The cutoff frequency, $\omega_{c,1}$ is used and the filter is concatenated into a five-input, five-output filter to form, $W_{1,inner}$ .

$$\omega_{c,1} = 1.6\text{Hz} \tag{3.27}$$

The state space of the first-order, high-pass, Bessel filter is:

$$(A_{1,i}, B_{1,i}, C_{1,i}, D_{1,i}) = (10.0531, 1, -10.0531, 0) \tag{3.28}$$

The vector $\boldsymbol{e}_i = \begin{bmatrix} \phi_{err} & \theta_{err} & \boldsymbol{\omega}_{err}^T \end{bmatrix}^T$ is five-dimensional and thus the filter is concatenated into a a five-input, five-output filter to form $W_{1,inner}$.

$$W_{1,inner} \begin{cases} \dot{\boldsymbol{x}}_{1,i} = \begin{bmatrix} A_{1,i} & 0 & \dots & 0 \\ 0 & A_{1,i} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & A_{1,i} \end{bmatrix} \boldsymbol{x}_{1,i} + \begin{bmatrix} B_{1,i} & 0 & \dots & 0 \\ 0 & B_{1,i} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & B_{1,i} \end{bmatrix} \boldsymbol{e}_i \\ Z_{1,inner} = \begin{bmatrix} C_{1,i} & 0 & \dots & 0 \\ 0 & C_{1,i} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & C_{1,i} \end{bmatrix} \boldsymbol{x}_{1,i} + \begin{bmatrix} D_{1,i} & 0 & \dots & 0 \\ 0 & D_{1,i} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & D_{1,i} \end{bmatrix} \boldsymbol{e}_i \end{cases} \tag{3.29}$$

To ensure adequate response speed in the outer-loop controller, a penalty filter on the guidance tracking error, $W_{1,outer}$ is added; which is also a first-order, low-pass, Bessel filter with cutoff frequency $\omega_{c,outer}$.

$$\omega_{c,outer} = 0.0045\text{Hz} \tag{3.30}$$

The state space of the first-order, low-pass, Bessel filter is:

$$(A_{1,o}, B_{1,o}, C_{1,o}, D_{1,o}) = (-.0283, 1, .0283, 0) \tag{3.31}$$

The vector $\boldsymbol{e}_o = \begin{bmatrix} \alpha_{err} & \beta_{err} & \chi_{err} \end{bmatrix}^T$ is three-dimensional and thus the filter is concatenated into a three-input, three-output filter.

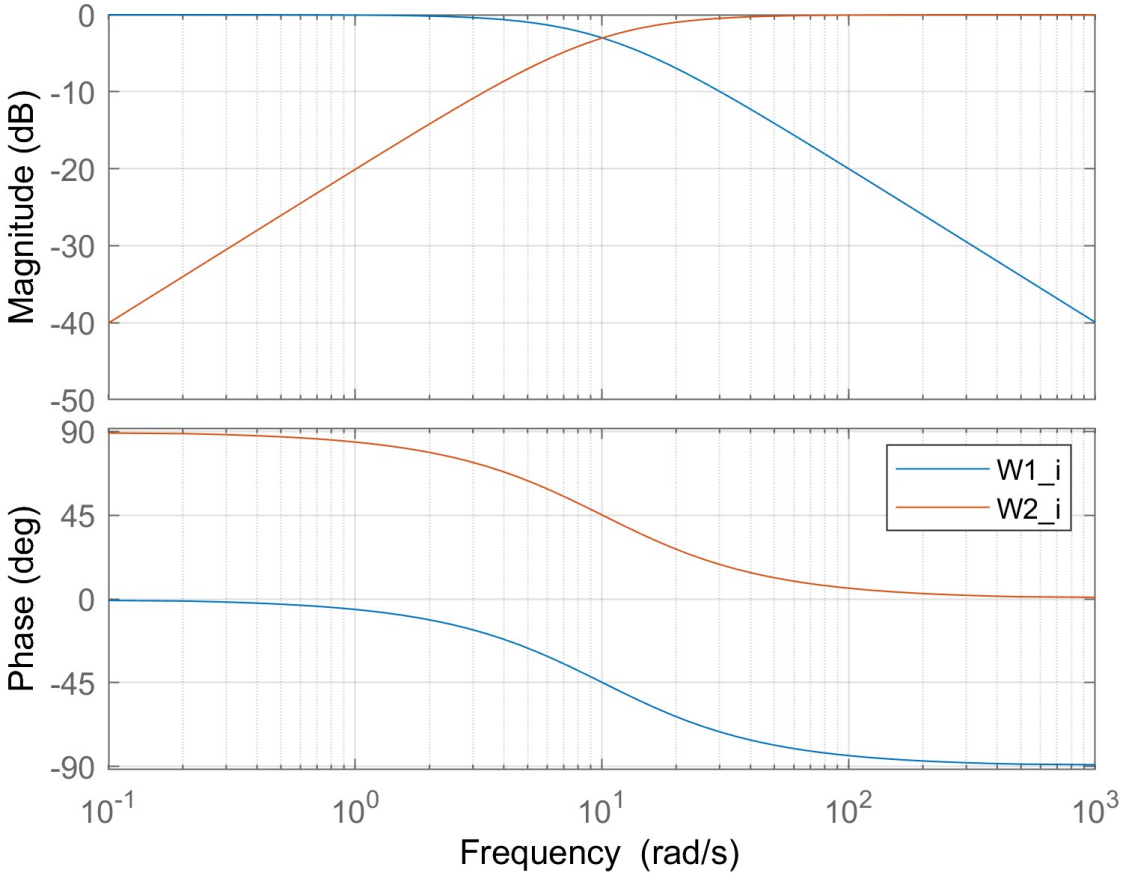



Figure 3.5: Bode Plots of $W_{1,inner}$ and $W_{2,inner}$.

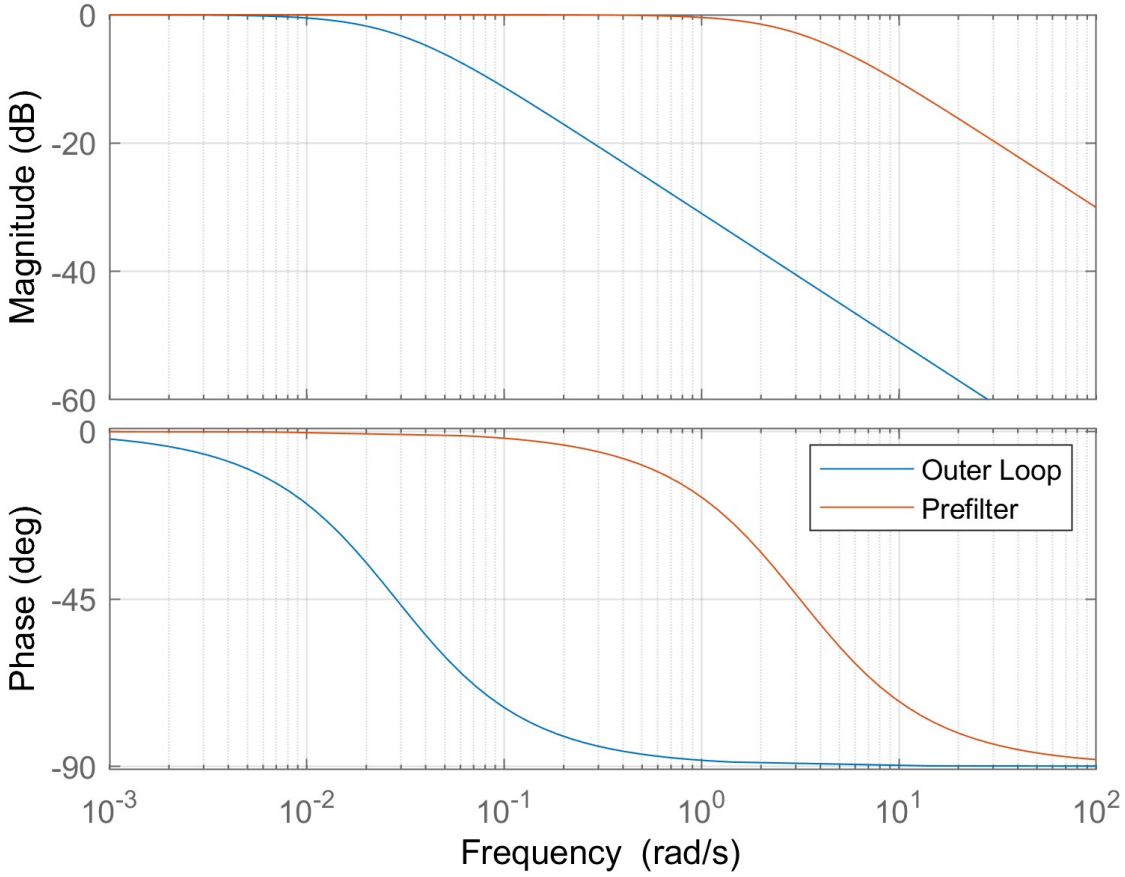



Figure 3.6: Bode Plots of $W_{1,outer}$ and $W_{prefilter}$.

$$W_{1,outer}\begin{cases} \dot{\boldsymbol{x}}_{1,o} = \begin{bmatrix} A_{1,o} & 0 & 0 \\ 0 & A_{1,o} & 0 \\ 0 & 0 & A_{1,o} \end{bmatrix} \boldsymbol{x}_{1,o} + \begin{bmatrix} B_{1,o} & 0 & 0 \\ 0 & B_{1,o} & 0 \\ 0 & 0 & B_{1,o} \end{bmatrix} \boldsymbol{e}_o \\ Z_{1,inner} = \begin{bmatrix} C_{1,o} & 0 & 0 \\ 0 & C_{1,o} & 0 \\ 0 & 0 & C_{1,o} \end{bmatrix} \boldsymbol{x}_{1,o} + \begin{bmatrix} D_{1,o} & 0 & 0 \\ 0 & D_{1,o} & 0 \\ 0 & 0 & D_{1,o} \end{bmatrix} \boldsymbol{e}_o \end{cases} \tag{3.32}$$

Lastly, a prefilter has been added to remove the parameter dependency on the matrix, $B_o$ [40]. This prefilter is also chosen as a first-order Bessel filter with high cutoff frequency as to minimize its effects on the aircraft dynamics.

$$\omega_{c,p} = 0.5\text{Hz} \tag{3.33}$$

$$(A_{1,p}, B_{1,p}, C_{1,p}, D_{1,p}) = (-3.1416, 1, 3.1416, 0) \tag{3.34}$$

The tracking error command is three-dimensional, thus the prefilter is concatenated into a three-input, three-output filter.

$$W_p \begin{cases} \dot{\boldsymbol{x}}_{1,p} = \begin{bmatrix} A_{1,p} & 0 & 0 \\ 0 & A_{1,p} & 0 \\ 0 & 0 & A_{1,p} \end{bmatrix} \boldsymbol{x}_{1,p} + \begin{bmatrix} B_{1,p} & 0 & 0 \\ 0 & B_{1,p} & 0 \\ 0 & 0 & B_{1,} \end{bmatrix} \begin{bmatrix} \alpha_{err} \\ \beta_{err} \\ \chi_{err} \end{bmatrix} \\ \begin{bmatrix} \alpha_{err,p} \\ \beta_{err,p} \\ \chi_{err,p} \end{bmatrix} = \begin{bmatrix} C_{1,p} & 0 & 0 \\ 0 & C_{1,p} & 0 \\ 0 & 0 & C_{1,p} \end{bmatrix} \boldsymbol{x}_{1,p} + \begin{bmatrix} D_{1,p} & 0 & 0 \\ 0 & D_{1,p} & 0 \\ 0 & 0 & D_{1,p} \end{bmatrix} \begin{bmatrix} \alpha_{err} \\ \beta_{err} \\ \chi_{err} \end{bmatrix} \end{cases} \tag{3.35}$$

Bode plots of the inner-loop and outer loop filters are given in Fig. 3.5 and 3.6 respectively. TP-convex polytopic models can now be concatenated with the state space models of the filters and put into $H_\infty$ control form to minimize the output of the weighting filters, $\mathbf{z}_o$ and $\mathbf{z}_i$ for both inner and outer loops.

$$\begin{aligned} \dot{\mathbf{x}}_o &= \mathcal{A}_o(\mathbf{p})\mathbf{x}_o + B_{1,o}(\mathbf{p})\boldsymbol{r} + B_{2,o}\boldsymbol{u}_o \\ \mathbf{z}_o &= C_{1,o}(\mathbf{p})\mathbf{x}_o + D_{11,o}(\mathbf{p})\boldsymbol{r} + D_{12,o}\boldsymbol{u}_o \\ \boldsymbol{y}_o &= C_{2,o}\mathbf{x}_o + D_{21,o}\boldsymbol{r} + D_{22,o}\boldsymbol{u}_o \end{aligned} \tag{3.36}$$

$$\begin{aligned} \dot{\mathbf{x}}_i &= \mathcal{A}_i(\mathbf{p})\mathbf{x}_i + B_{1,i}(\mathbf{p})\boldsymbol{u}_o + B_{2,i}\boldsymbol{tau} \\ \mathbf{z}_i &= C_{1,i}(\mathbf{p})\mathbf{x}_i + D_{11,i}(\mathbf{p})\boldsymbol{u}_o + D_{12,i}\boldsymbol{tau} \end{aligned} \tag{3.37}$$

$$\boldsymbol{y}_i = C_{2,i}\mathbf{x}_i + D_{21,i}\boldsymbol{u}_o + D_{22,i}\boldsymbol{tau}$$

The values $\mathbf{x}_o$ and $\mathbf{x}_i$ represent the concatenated state vector for both loops. $x$ represents the states of the TP-convex polytopic models. $\mathbf{z}_o$ and $\mathbf{z}_i$ represent the concatenated output of the weighting filters for both loops. $y$ represents the values of the TP-convex polytopic models. $\boldsymbol{r}$ represents the guidance command $\boldsymbol{r} = \begin{bmatrix} \alpha_{cmd} & \beta_{cmd} & \chi_{cmd} \end{bmatrix}^T$ and $u_o$ represents the control command $\boldsymbol{u}_o = \begin{bmatrix} \phi_{cmd} & \omega_{2,cmd} & \omega_{3,cmd} \end{bmatrix}^T$.

$$\mathbf{x}_o = \begin{bmatrix} x_{p,o}^T & x_{1,o}^T & x_{prefilter}^T \end{bmatrix}^T \tag{3.38}$$

$$\mathbf{x}_i = \begin{bmatrix} x_{p,i}^T & x_{1,i}^T & x_{2,i}^T \end{bmatrix}^T \tag{3.39}$$

$$\mathbf{z}_o = Z_{1,outer} \tag{3.40}$$

$$\mathbf{z}_i = \begin{bmatrix} Z_{1,inner}^T & Z_{2,inner}^T \end{bmatrix}^T \tag{3.41}$$

The inner and outer loop controllers were built in MATLAB using the `hinfgs` MATLAB routine. The quadratic $H_\infty$ performance was found to be $\Gamma_i = 1.002005$ for the inner loop and $\Gamma_o = 1.000012$ for the outer loop.

### 3.4.1 NDI Review

The LPV control algorithm is compared with an existing Nonlinear Dynamic Inversion control law and the control surface deflection commands from each controller are compared. The NDI control law is given in [41] and reviewed here. To start,

the outer-most loop of the NDI converts the $\chi$ guidance commands to attitude commands, $\phi$:

$$\phi_{cmd} = \mathrm{Sat}_{-45^\circ}^{45^\circ}\left(\arctan\left(\frac{V}{g}K_\chi(\chi_{cmd}-\chi)\right)\right) \tag{3.42}$$

The next inversion step treats $\omega_1$, $\omega_2$, and $\omega_3$ as inputs to the dynamics governing $\phi$, $\alpha$, and $\beta$ respectively. Based on the simplified dynamics in Eq. (3.8):

$$\omega_{1,c} = K_\phi(\phi_c - \phi) - \tan(\theta)(q\sin(\phi) + r\cos(\phi)) \tag{3.43a}$$

$$\omega_{2,c} = K_\alpha(\alpha_c - \alpha) + \frac{g}{V}\sin(\phi)\tan(\phi) \tag{3.43b}$$

$$\omega_{3,c} = -K_\beta(\beta_c - \beta) - \frac{g}{V}\sin(\phi) \tag{3.43c}$$

The fastest inner loop inverts the angular rate commands to produce a desired angular acceleration proportional to the angular rate error.

$$\boldsymbol{\tau}_{u,c} = IK_\omega(\omega_c - \omega) - (\hat{\boldsymbol{\tau}} - \omega \times I\omega) \tag{3.44}$$

$\hat{\boldsymbol{\tau}}$ is the estimate of the current moment on the aircraft, which is determined from real-time modeling results. In order to determine the proportionality constants it is assumed the dynamic inversion commands were able to remove the nonlinear effects and only the desired linear portions remain. As an example, examine the linearized short-period dynamics:

$$\begin{bmatrix}\dot{\alpha}\\ \dot{\beta}\end{bmatrix} = \begin{bmatrix}0 & 1\\ -K_qK_\alpha & -K_q\end{bmatrix}\begin{bmatrix}\alpha\\ q\end{bmatrix} + \begin{bmatrix}0\\ K_qK_\alpha\end{bmatrix}\alpha_{cmd} \tag{3.45}$$

Table 3.4: Desired frequency calculations

| Channel | Desired Frequency |
|---|---|
| Roll | $\omega_n = \sqrt{\lvert \frac{\bar{q}Sb}{2I_{xx}} C_{l_{\delta_r}} \rvert}$ |
| Pitch | $\omega_n = \sqrt{\lvert \frac{\bar{q}S\bar{c}}{I_{yy}} C_{m_\alpha} \rvert}$ |
| Yaw | $\omega_n = \sqrt{\lvert \frac{\bar{q}Sb}{I_{zz}} C_{n_\beta} \rvert}$ |

The control parameters are written in terms of the natural frequency and damping of the second order system.

$$K_q = 2\zeta\omega_n \tag{3.46}$$

$$K_\alpha = \omega_n/(2\zeta) \tag{3.47}$$

The desired natural frequency was determined based on the modeling results using the relationship in Table 3.4 and the desired damping ratio was fixed at $\zeta = 0.8$. The control parameters for the roll channel $(K_\phi, K_p)$ and the yaw channel $(K_\beta, K_r)$ are determined by the same procedure. The control parameter for the tracking angle, $\chi$ was fixed at $K_\chi = 0.375$.

### 3.4.2 AutoThrottle

Both the LPV and the NDI controllers use the same auto throttle controller to regulate the calibrated airspeed of the aircraft. A proportional-integral (PI) controller was taken from [41] and used here.

$$Th = K_p\Big(V_{CAS,c} - V_{CAS}\Big) + K_I \int_0^t \Big(V_{CAS,c} - V_{CAS}\Big) d\tau \tag{3.48}$$

### 3.4.3 Control Allocation

Control allocation algorithms are tasked with determining aircraft control surface deflections to achieve the desired moments commanded by the controller. Control allocation is generally required for aircraft where the number of control effectors exceeds the number of moments (over-actuated). Many researchers have formulated the control allocation problem as a constrained-optimization problem [17] [45, 46, 24, 47, 25, 48]. The allocator used to generate the results in this chapter is taken from the L2F project, [41] and is used for both the LPV and NDI control laws.

The real-time modeling procedure gives a matrix $\boldsymbol{B}(\boldsymbol{\theta}) \in \mathbb{R}^{3\times 6}$ with dimensional control derivatives defining the partial derivatives with respect to the 6 control surface deflections. The control allocation used in this chapter is the linear programming method described in chapter 2. In MCAAD the modeling determines $\boldsymbol{B}(\boldsymbol{\theta})$, whereas the online control specifies $\boldsymbol{\tau}_u$. This formulation of the control allocation problem is a mixed optimization problem that combines the error and control minimization problems into a single problem with parameter, $\epsilon$. Minimizing the moment error $||\boldsymbol{B}(\boldsymbol{\theta})\boldsymbol{u} - \boldsymbol{\tau}_c||_1$ is the primary objective of the control allocation designer, but for an over-actuated aircraft configuration there are infinitely many solutions. Thus the second term, $\epsilon||\boldsymbol{u} - \boldsymbol{u}_0||$ is added to the optimization problem to determine a particular solution. In this research the metric used to define $J$ is the $l_1$ norm which is the sum of the absolute values within the norm:

$$||\boldsymbol{x}||_1 = \sum_{i=1}^{n} |x_i| \tag{3.49}$$

The control allocation problem in this chapter is solved in real time using an efficient algorithm formulated by Bodson using linear programming approaches,

providing guaranteed convergence in an acceptable period of time [49]. The algorithm was based on the revised simplex method [50] with additional refinements, such as anti-cycling, as described in detail in [49].

## 3.5 Simulation

As preivously noted, the architecture shown in Fig. 3.4 is built in `Simulink` ® and the LPV controllers are designed using the `hinfgs` `MATLAB` routine. The gains from the two-loops LPV controller gains are now determined by:

$$K_{p,o} = \sum_{r=1}^{8} w_{r,o}(\mathbf{p}) K_{r,o} \tag{3.50}$$

$$K_{p,i} = \sum_{r=1}^{6} w_{r,i}(\mathbf{p}) K_{r,i} \tag{3.51}$$

Where:

$$K_r = \begin{bmatrix} A_{K,r} & B_{K,r} \\ C_{K,r} & D_{K,r} \end{bmatrix} \tag{3.52}$$

The implementation of the two-loop LPV control architecture is described in the following subsections.

### 3.5.1 Outer Loop Implementation

$$\dot{\boldsymbol{x}}_{K,o} = \sum_{r=1}^{8} w_{r,o}(\mathbf{p}) \left( A_{K,r,o}\boldsymbol{x}_{K,o} + B_{K,r,o} \begin{bmatrix} \alpha_{cmd} \\ \beta_{cmd} \\ \chi_{cmd} \end{bmatrix} \right)$$

$$\begin{bmatrix} \phi_{cmd} \\ \omega_{2,cmd} \\ \omega_{3,cmd} \end{bmatrix} = \sum_{r=1}^{8} w_{r,o}(\mathbf{p}) \left( C_{K,r,o}\boldsymbol{x}_{K,o} + D_{K,r,o} \begin{bmatrix} \alpha_{cmd} \\ \beta_{cmd} \\ \chi_{cmd} \end{bmatrix} \right) \tag{3.53}$$

The commands calculated by the outer loop controller $(\phi_{cmd}, \omega_{2,cmd}, \omega_{3,cmd})$ are then passed through the prefilter described in section 3.4 to produce $(\phi_{cmd,p}, \omega_{2,cmd,p}, \omega_{3,cmd,p})$. There is also a saturation function to prevent the outer loop from commanding excessive roll angle.

$$\phi_{cmd,sat} = \mathrm{Sat}_{-45^\circ}^{45^\circ} \left( \phi_{cmd,p} \right) \tag{3.54}$$

In order to command a zero flight path angle, pitch angle command is set to angle of attack command. Body-axis roll rate command is set to zero $\omega_{1,cmd} = 0$.

$$\theta_{cmd} = \alpha_{cmd} \tag{3.55}$$

### 3.5.2 Inner Loop Implementation

$$\dot{\boldsymbol{x}}_{K,i} = \sum_{r=1}^{6} w_{r,i}(\mathbf{p}) \left( A_{K,r,i}\boldsymbol{x}_{K,i} + B_{K,r,i} \begin{bmatrix} \phi_{cmd,sat} \\ \theta_{cmd} \\ \omega_{1,cmd} \\ \omega_{2,cmd} \\ \omega_{3,cmd} \end{bmatrix} \right) \tag{3.56}$$

$$\boldsymbol{\tau}_{u,cmd} = \sum_{r=1}^{6} w_{r,i}(\mathbf{p}) \left( C_{K,r,i}\boldsymbol{x}_{K,i} + D_{K,r,i} \begin{bmatrix} \phi_{cmd,sat} \\ \theta_{cmd} \\ \omega_{1,cmd} \\ \omega_{2,cmd,p} \\ \omega_{3,cmd,p} \end{bmatrix} \right) - \hat{\boldsymbol{\tau}}$$

The two-loop LPV controller has been tested on a SIMULINK simulation of the NASA E1 aircraft and compared with the results of the NDI controller.

It can be seen from Fig. 3.7 that there is a large initial transient until 5s and then the LPV controller is able to track $(\alpha, \beta)$ commands well, with an error about $\pm$ 5°. The initial transient is caused by the controller's response to the large initial error in $\chi$. When tracking $\chi$ commands, the LPV controller exhibits worse tracking performance than the NDI controller with a tracking error about $\pm$ 15°.

The NDI controller exhibits a better response to the initial $\chi$ error, causing less of an initial transient than the LPV controller, as shown in Fig. 3.9. Additionally the guidance commands a step in $\alpha$ command at 23s which is not seen in the LPV simulation. This step is commanded because of a switch in the guidance block being triggered by high altitude. After the initial transient is settled, around 5s, the NDI and LPV exhibit similar tracking characteristics when with respect to $\alpha$

and $\beta$. The NDI consistently tracks $\chi$ better than the LPV controller, but the NDI has a very large $\alpha$ excursion after the $\chi$ step change at 17s while the LPV does not. Figure 3.8 and Fig. 3.10 show the attitudes realized by both control laws. The LPV controller exhibits a large attitude response in roll angle, $\phi$ which is caused by the large initial $\chi$ error. Around 17s a large change in $\chi$ is commanded with results in a large change in roll angle, as realized by both the NDI and LPV controllers. Both controllers exhibit similar pitch angle excursions, with a maximum pitch angle about $+20°$. The yaw angle, $\psi$ follows the relationship $\psi = \chi - \beta$ and the resulting change in $\psi$ stems from the commanded change in $\chi$.

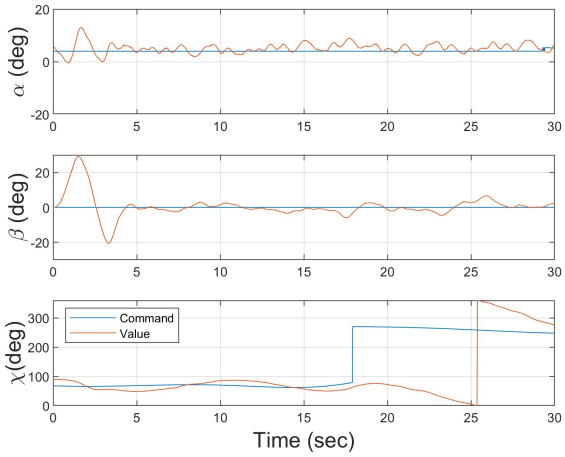



Figure 3.7: Guidance tracking of the LPV controller.

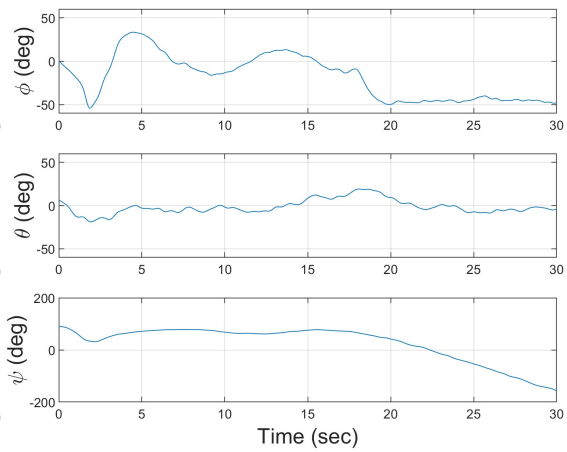



Figure 3.8: Attitude angles using LPV controller.

Additionally, the control surface deflections were recorded and they were compared to the commanded PTI control surface deflections in both the time domain and the frequency domain.

Figure 3.11 shows the control surface deflections commanded by the NDI controller compared to that of the PTI. The deflections correlate well for both left and right flaps. Additionally the deflections correlate well for both left and right ailerons until about 17s when there is an aggressive change in $\chi_{cmd}$ which results

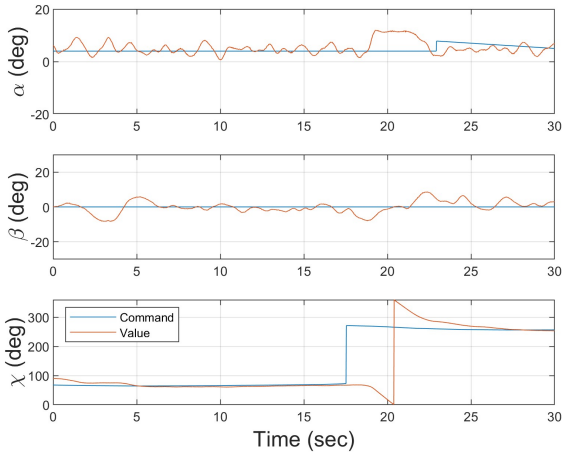



Figure 3.9: Guidance tracking of the NDI controller.

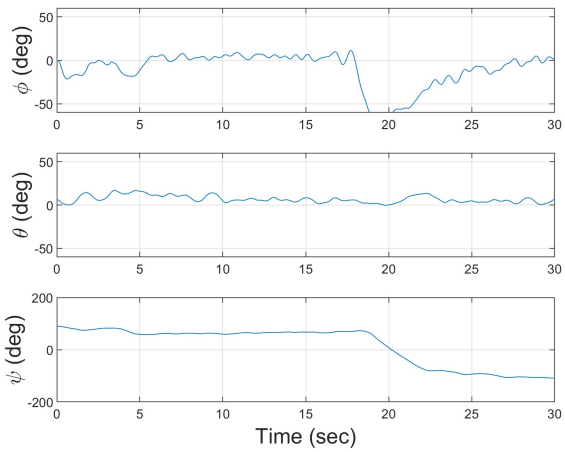



Figure 3.10: Attitude angles using NDI controller.

in an aggressive change in aileron command. The elevator and rudder are offset by about 5° from the PTI throughout the simulation.

Figure 3.12 shows the control surface deflections commanded by the LPV controller compared to that of the PTI. The deflections correlate well for both left and right flaps and the left and right ailerons. Around 15s high frequency control surface content can be seen, which is expected for the LPV controller. Similar to the NDI controller, the LPV commands for elevator and rudder deflection offset about 5°.

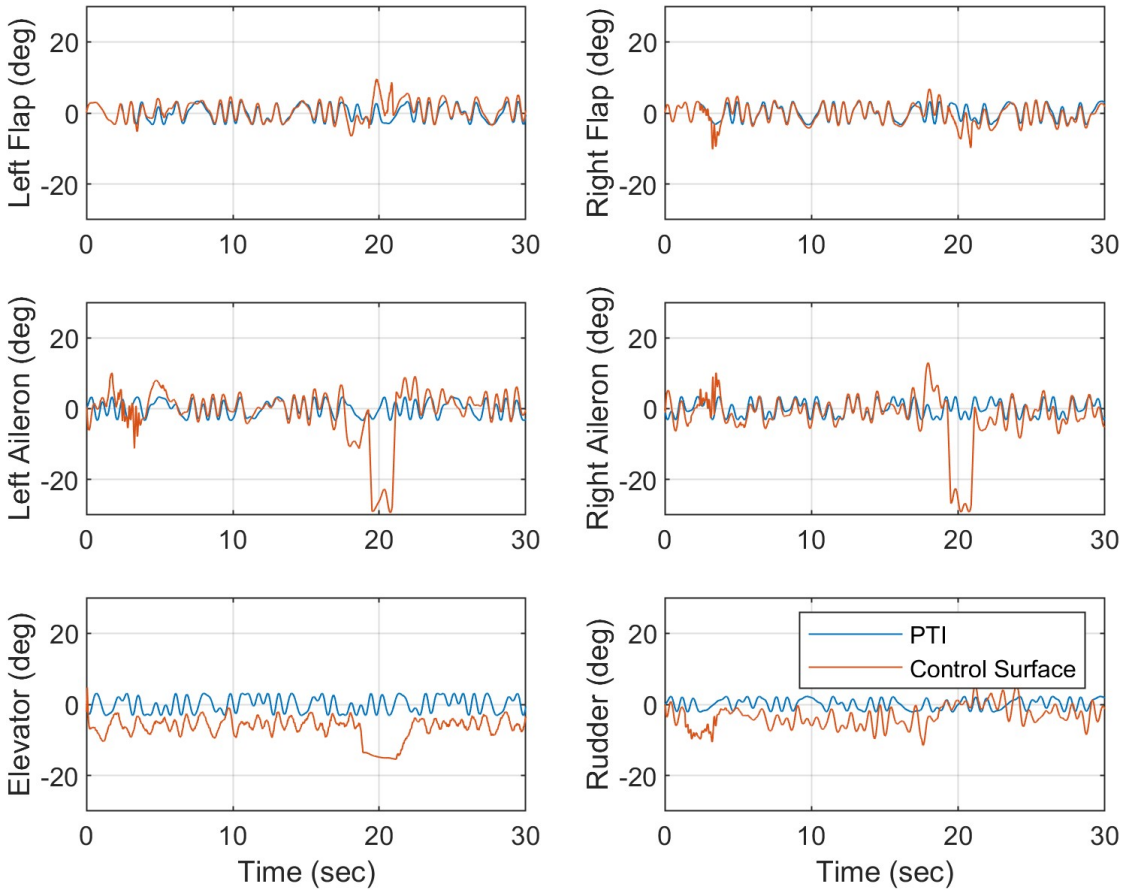



Figure 3.11: Control Surface Deflections with NDI Control.

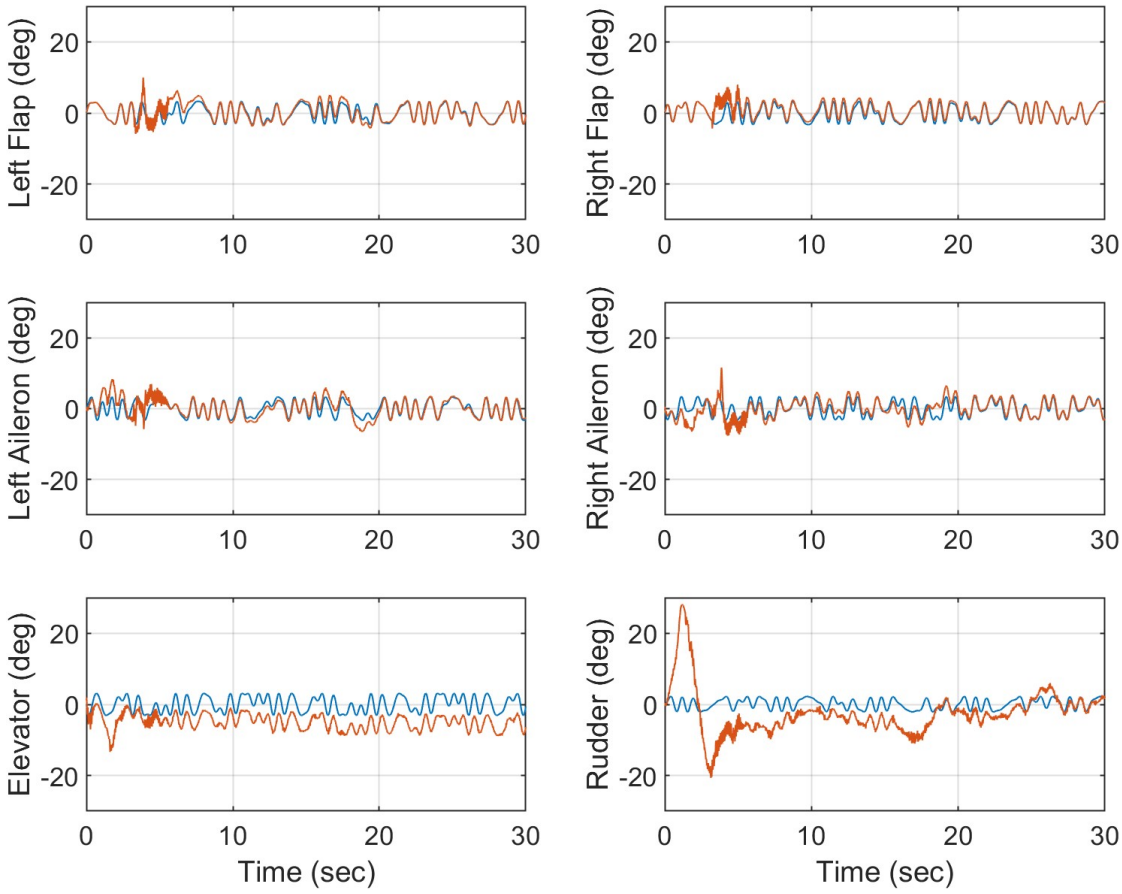



Figure 3.12: Control Surface Deflections with LPV Control.

Figure 3.13 shows the frequency domain content of the PTI and the control surface deflections commanded by the NDI control law. The left and right flap correlate well with the peaks of the PTI. The left and right aileron show poor correlation at frequencies below 0.5Hz, however at frequencies above 1Hz the control surface deflections start to align with the PTI. Similarly, the elevator and aileron do not align with the PTI well at frequencies below 0.5Hz but align better at frequencies above 1Hz.

Figure 3.14 shows the frequency domain content of the PTI and the control surface deflections commanded by the LPV control law. For all six effectors, there is little distortion in magnitude at frequencies below 0.5Hz and almost none at

frequencies above 1Hz. It is important to note that the LPV control law commands control surface deflections at a higher frequency than the PTI, which allows for minimal distortion.

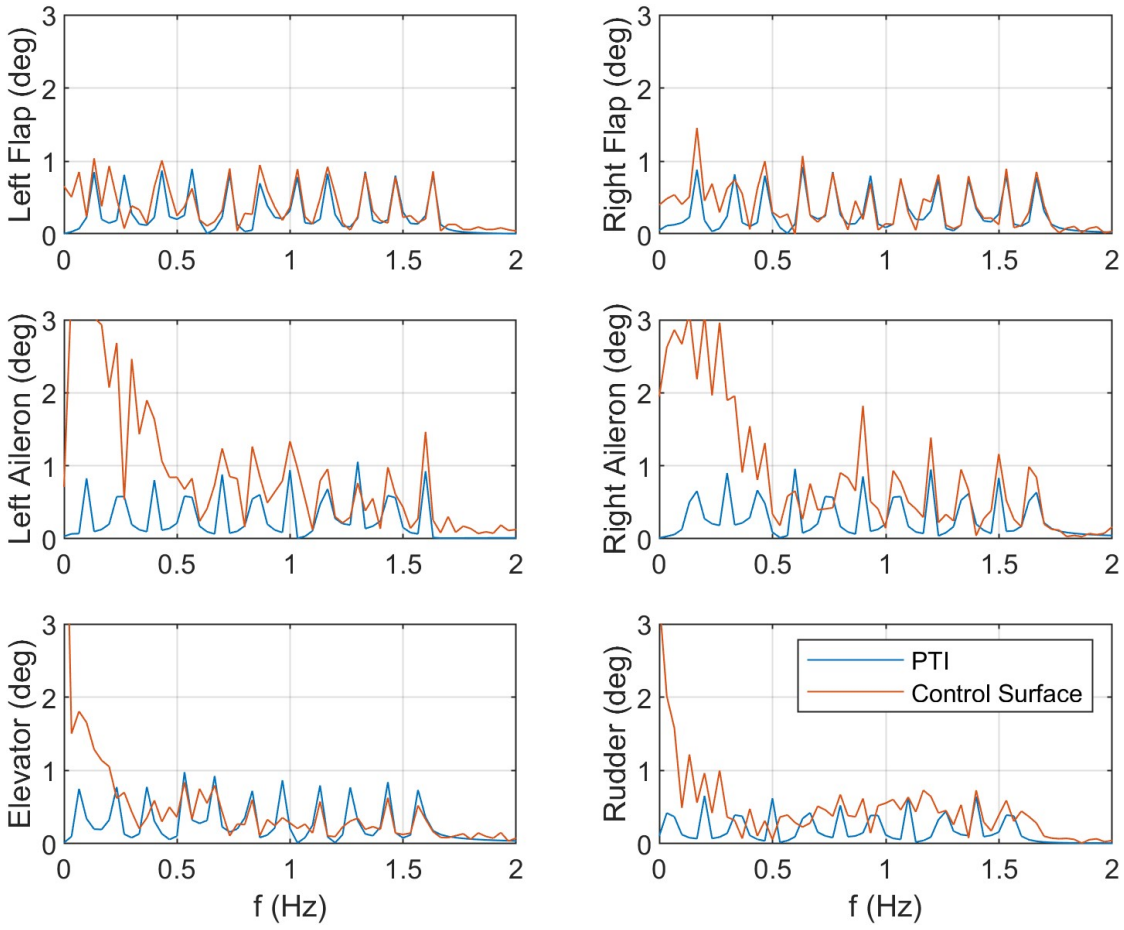



Figure 3.13: Control Surface Frequency Content with NDI Control.

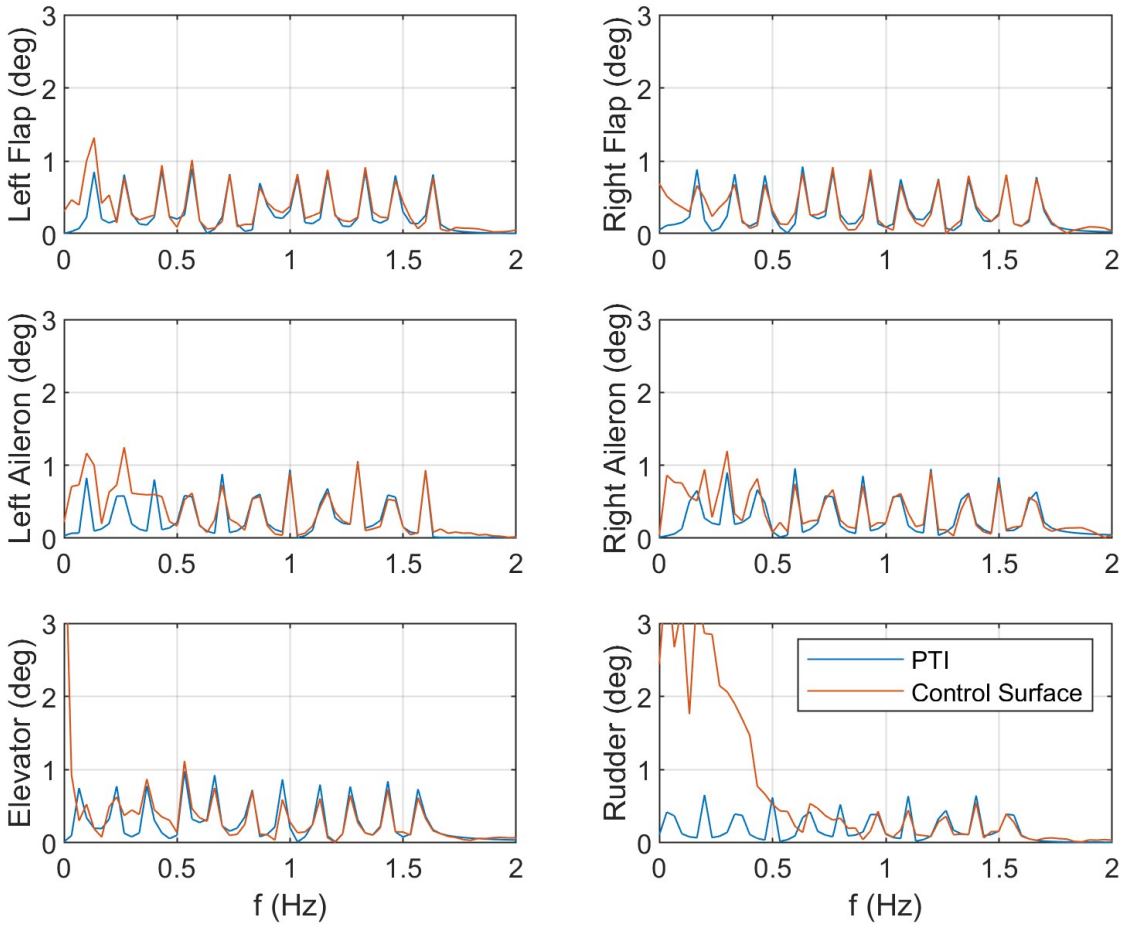



Figure 3.14: Control Surface Frequency Content with LPV Control.

## 3.6 Chapter Conclusions

In this chapter a two-loop LPV control law was designed for the NASA E1 aircraft. The two-loop control architecture was shown to penalize control commands within the PTI frequency band and be more compatible with excitation due to real-time modeling. The synthesis of both control loops requires an affine LPV model, which was generated from the non-affine kinematic-based E1 model using the TP method. The derived two-loop LPV controller is tested on a SIMULINK® simulation of the NASA E1 aircraft, which compared the performance of the LPV controller to a previously designed NDI controller. It was found that the LPV

controller is able to track guidance commands, but could exhibit slightly worse tracking performance than the NDI controller. Despite the slightly worse tracking performance, the LPV controller exhibits desirable characteristics such as its invariance of the real-time modeling parameters. The remainder of this dissertation entails control allocation methods to better realize commanded moment and thus improve tracking performance.

# Chapter 4

# A Comparison of Control Allocation Methods in the Presence of Parametric Model Uncertainty

Optimal control allocation is a well-studied aspect of flight control design. Typically control laws are designed to produce virtual moment commands to be imparted on a vehicle. These virtual moment commands could be forces, torques, or impulses. They are realized on vehicles through manipulation of available control effectors. Common vehicle control effectors include aerodynamic control surfaces and propulsors on aircraft. Reaction control systems, thrust vectoring control systems, and control moment gyros are common control effectors on spacecraft. Control allocation is the method that distributes the virtual moment commands to the individual effectors.

Generalized inverses are the most common control allocation approach which

utilizes a local linearization of control effectiveness. The subsequent mapping between virtual and physical commands is derived in a closed-form solution to a Lagrange optimization problem [51, 46]; the simplest generalized inverse is a pseudo-inverse (PI) allocator [15]. PI involves only a matrix multiplication, hence making it attractive for real-time implementation, especially on human-rated vehicles where safety is paramount. A major drawback to single matrix generalized inverses is that effector position constraints can only be satisfied by limiting the achievable moments.

The control allocation problem can also be formulated as a convex optimization problem and solved on-line using linear programming (LP) or quadratic programming (QP) techniques [25, 52]. The major benefit is that the LP and QP techniques can account for effector position constraints [53, 54, 49, 24]. Furthermore, it has been demonstrated in Ref. [55] that additional convex objectives such as minimizing the $l_\infty$ norm of the structural loads on the control surfaces can be incorporated into the optimization problem. Common solution techniques include interior-point methods, simplex algorithms, and more recently first-order methods such as Alternate Direction Method of Multiplier (ADMM) [56, 57, 58]. Historically, due to lack of computational power and solver maturity, the aerospace controls community has generally avoided on-line optimization as part of the flight controls architecture. Rigorous flight certification process requires convergence guarantees and fast solution times. The past 10–15 years has seen a sharp rise in solver technology [59, 60, 61, 62, 63]. The SpaceX Falcon F9 booster landing uses model predictive control to solve the real-time trajectory optimization problem with hundreds of decision variables in fractions of a second [64]. This shift in paradigm suggests that the formulation of the optimal control allocation problem as a convex program could supplant generalized inverses as the new industry standard.

The aforementioned literature does not account for uncertainty in the elements of the controls matrix, $\boldsymbol{B}$. Typically robustness to modeling errors is evaluated by performing Monte Carlo simulations of the truth model. Grauer and Pei derived the minimum-variance (MV) control allocator, which is a specific form of generalized inverse and considers the effects of parametric model uncertainty by minimizing the variance of the generalized moment error [11]. MV was shown to outperform PI in the presence of known parametric uncertainty in elements of $\boldsymbol{B}$. Robust control techniques have also gained attention to help minimize the effects of bounded disturbances in the allocation problem [65, 66, 67, 68]. Proper compensation of the effects of parametric model uncertainty is critical, as a realized virtual control error will create unintended motion and additional burden on the pilot and/or feedback control loop [69].

To the knowledge of the authors, there appears to be little documentation that discusses the effects of convex optimization-based control allocation considering parametric model uncertainty. De Almeida applied robust convex optimization to offline control allocation design [70]. Duan, Zhang, Zhao, and Kong studied on-line robust control allocation under ellipsoidal uncertainty [26]. In this chapter, an aircraft model was used to compare the PI, MV, LP, and QP allocators in the presence of parametric model uncertainty. The parametric model uncertainty was taken from previously derived examples [11], in practice, covariance matrix of aerodynamic effectors can be derived from system identification techniques [9] either off-line or in-real time [41, 23].

The remainder of this chapter is organized as follows. Section 4.1 provides an example which compares the efficacy of the PI, MV, LP, and QP approaches which were introduced in chapter 2. Lastly, section 4.2 provides concluding remarks and recommendations for future research.

## 4.1 Example

For a fixed-wing aircraft, several reasons for significant uncertainty in the control effectiveness matrix include: high angle-of-attack flight conditions where unsteady flow and vortex shedding can occur, control effectiveness degradation due to icing, control surface interaction, damaged or failed control surfaces, or in real-time modeling research where the initial control effectiveness is unknown, but learned in-flight [23]. The example in this chapter considers the F-16 Advanced Fighter Technology Integration (AFTI) model which is depicted in Fig. 4.1. The control surfaces used to control the lateral-directional dynamics are: anti-symmetric ailerons $u_a$, differential stabilator $u_{ds}$, rudder $u_r$, and vertical canards $u_{vc}$ mounted underneath the engine intake. The virtual controls are body-axis roll acceleration $\dot{p}$ and body-axis yaw acceleration $\dot{r}$. The model for the lateral-directional dynamics was evaluated in straight and level flight at Mach 0.6 and $30,000$ ft altitude [1].

$$\begin{bmatrix} \dot{\beta} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} -0.154 & 0.0824 & -0.998 & 0.0521 \\ -19.2 & -0.894 & 0.319 & 0 \\ 2.30 & -0.0009 & -0.279 & 0 \\ 0 & 1 & 0.0823 & 0 \end{bmatrix} \begin{bmatrix} \beta \\ p \\ r \\ \phi \end{bmatrix} + \begin{bmatrix} 0.0004 & 0.0144 & 0.0212 & 0.0074 \\ -17.5 & -13.6 & 3.92 & 0.415 \\ -0.268 & -1.51 & -1.97 & 1.51 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_a \\ u_{ds} \\ u_r \\ u_{vc} \end{bmatrix} \tag{4.1}$$

The control surface deflection limits and control surface rate limits in this example are also taken from Ref. [1] and are given in Table 4.1. The control surface




Figure 4.1: F-16 AFTI aircraft in flight (credit: NASA).

dynamics are modeled by the first order lag in Eq. (4.2).

Table 4.1: F-16 AFTI aircraft control surface position and rate limits

| Control surface | Position limit, deg | Rate limit, deg/s |
|---|---|---|
| Ailerons | ±20 | 52 |
| Stabilators | ±25 | 60 |
| Rudder | ±30 | 120 |
| Vertical canards | ±27 | 100 |

$$TF(s) = \frac{20}{s + 20} \tag{4.2}$$

The uncertainty in the $\boldsymbol{B}$ matrix is given in Table 4.2 and is taken from Ref. [11]. This uncertainty is typical for that of fixed-wing aircraft and expressed in percentages of control derivatives [11]. The PI, MV, LP, and QP allocators are designed based on the information in Eq. (4.1) and Table 4.1.

Table 4.2: Standard errors used in example

| | $u_a$ | $u_{ds}$ | $u_r$ | $u_{vc}$ |
|---|---|---|---|---|
| $L'$ | 5% | 10% | 20% | 25% |
| $N'$ | 50% | 50% | 5% | 10% |

### 4.1.1 Static Allocator Performance

This section compares the static performance of all four allocators with and without uncertainty in the $\boldsymbol{B}$ matrix. Begin by defining the theoretical attainable moment set (TAMS) as the set of moments due to all possible combinations of control effector positions. Similarly, define the particular attainable moment set (PAMS) is the subset of the TAMS which can be realized by the allocator. One performance metric of an allocation scheme is allocation efficiency and is defined as the ratio of the size of the PAMS to the TAMS, $\boldsymbol{\eta}$. If the allocator can realize the entire space in the TAMS then $\boldsymbol{\eta} = 1$ and the allocator is perfectly efficient. Perfect knowledge of the control effectiveness matrix is necessary to characterize the nominal TAMS, however, the control effectiveness matrix is subject to parametric model uncertainty which will change the TAMS. Figure 4.2 shows the TAMS of the F-16 AFTI aircraft based on a nominal control effectiveness matrix (TAMS $B_{des}$), the TAMS computed with an example dispersed control effectiveness matrix (TAMS $B_{disR}$), and PAMS for each allocator. There are several regions where the TAMS $B_{des}$ and TAMS $B_{disR}$ do not overlap. The regions occupied by TAMS $B_{des}$ but not TAMS $B_{disR}$ mean that moments predicted to be attainable by the model are unattainable in practice.

Convex optimization-based approaches in LP and QP provide superior allocation efficiency as compared to the generalized inverses. The efficiencies are provided in Fig. 4.2 in terms of percentages, due to its conservative nature, MV has

the smallest value of $\boldsymbol{\eta}$, as expected.

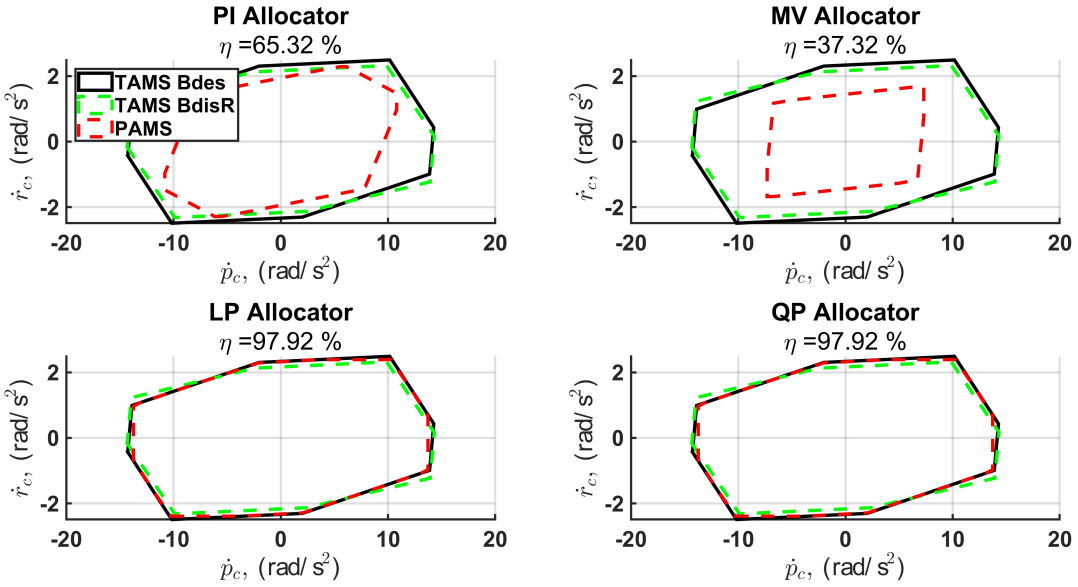



Figure 4.2: Allocation efficiency.

While it is necessary to characterize the allocation efficiency when comparing allocator static performance, it is equally important to quantify the allocation error. The allocation error is defined as the error between the realized and commanded moment. The sources of this error might be due to aforementioned dispersions in the control effectiveness matrix or a moment command outside the PAMS of the respective allocator. Fig. 4.3 compares the static control allocation performance of all four approaches for a single moment command that is outside of the TAMS and considers dispersions in the control effectiveness matrix. To show the performance of the allocators in the presence of effector saturation, a moment command outside the TAMS, $\boldsymbol{\tau}_c = \begin{bmatrix} 13 & 2 \end{bmatrix}^T$ rad/s$^2$ is chosen and is represented by the blue arrow.

Known for direction-preserving in the case of control saturation, if no uncertainties are considered, the expected result is that the PI and MV allocators will be co-linear with $\boldsymbol{\tau}_c$ and reach the edge of their respective PAMS. With convex optimization-based approaches the resultant moment is not guaranteed to be

co-linear with $\boldsymbol{\tau}_c$. The LP and QP allocators will also reach the edges of their respective PAMS, with a non-zero direction error. The direction error of the LP and QP allocators is attributed to the formulations of their respective objective functions which are to minimize the 1-norm and the 2-norm of the moment error without regard to moment direction.

In the presence of uncertainty, it is expected that neither the direction nor the magnitude will be preserved. In this example the MV allocator shows the smallest direction error and the largest magnitude error. Conversely, LP and QP allocators show the largest direction errors, but the least magnitude errors.

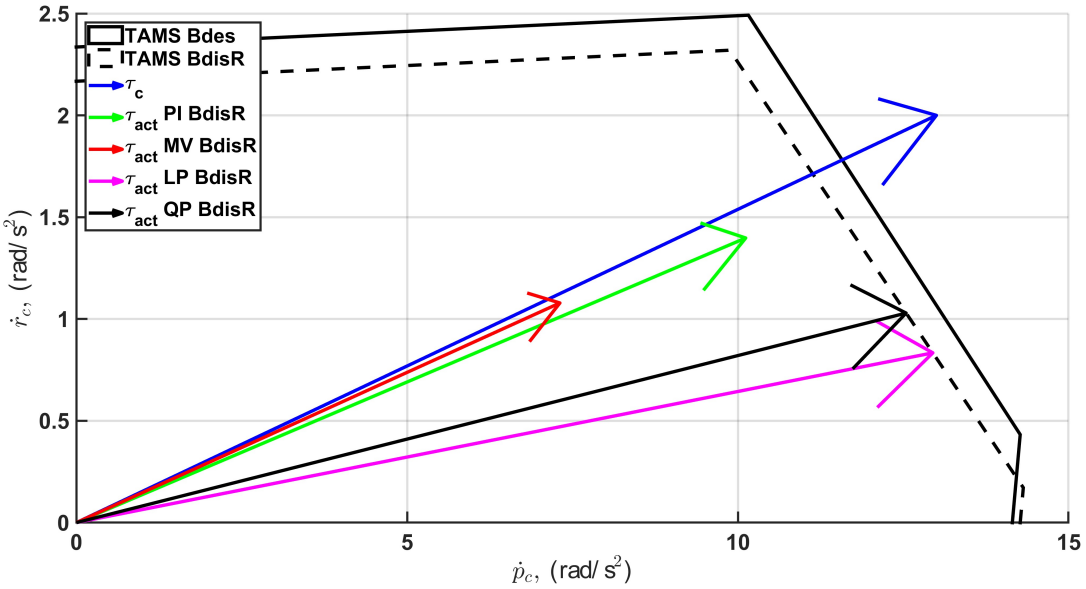



Figure 4.3: Control allocation error comparison for a single moment command.

While it is important compare allocator performances for a single $\boldsymbol{\tau}_c$, it is equally important to characterize performances across the entire moment space and dispersions in the $\boldsymbol{B}$ matrix. Figure 4.4 shows the magnitude error normalized by the moment command magnitude, alongside the direction error. Figure 4.5 shows the standard error in magnitude expressed as a percentage of moment command, alongside the standard error in direction. To generate Fig. 4.4 and 4.5 a

range of moment commands which spans and encompasses the nominal TAMS is commanded to the allocators. At each moment command $1,000$ Monte Carlo simulations were run where the control effectiveness matrix was varied according to the parameters provided in Table 4.2. The outline of the nominal TAMS is shown by the solid black line and the outline of the respective PAMS is shown by the dotted black line. The mean magnitude error for the $k$-th dispersion, $(\bar{M}_{e,k})$, the mean direction error for the $k$-th dispersion, $(\bar{D}_{e,k})$, the magnitude standard error, $(SE_M)$, and the direction standard error, $(SE_D)$ were computed by Eqs. (4.3)-(4.6), respectively. Since there are $1,000$ Monte Carlo simulations, $N = 1,000$.

$$\bar{M}_{e,k} = \frac{\sum_{k=1}^{N} ||\boldsymbol{B}_{dis}\boldsymbol{u}_{c,k} - \boldsymbol{\tau}_{c,k}||_2}{N \times ||\boldsymbol{\tau}_{c,k}||_2} \tag{4.3}$$

$$\bar{D}_{e,k} = \frac{\sum_{k=1}^{N} \operatorname{acos}\left(\dfrac{\boldsymbol{B}_{dis}\boldsymbol{u}_{c,k}}{||\boldsymbol{B}_{dis}\boldsymbol{u}_{c,k}||_2} - \dfrac{\boldsymbol{\tau}_{c,k}}{||\boldsymbol{\tau}_{c,k}||_2}\right)}{N} \tag{4.4}$$

$$SE_M = \sqrt{\frac{\sum_{k=1}^{N} \left(||\boldsymbol{B}_{dis}\boldsymbol{u}_{c,k} - \boldsymbol{\tau}_{c,k}||_2 - \bar{M}_{e,k}\right)^2}{N^2 \times ||\boldsymbol{\tau}_{c,k}||_2}} \tag{4.5}$$

$$SE_D = \sqrt{\frac{\sum_{k=1}^{N} \left( \operatorname{acos}\left(\dfrac{\boldsymbol{B}_{dis}\boldsymbol{u}_{c,k}}{||\boldsymbol{B}_{dis}\boldsymbol{u}_{c,k}||_2} \times \dfrac{\boldsymbol{\tau}_{c,k}}{||\boldsymbol{\tau}_{c,k}||_2}\right) - \bar{D}_{e,k}\right)^2}{N^2}} \tag{4.6}$$

In Fig. 4.4 all allocators show small percentage of allocation error for commands within the respective PAMS of the allocator. For all allocators the percentage of moment error rapidly increases near the origin, this is because the magnitude of the moment command is small near the origin and exacerbates the percent of magnitude error. Outside the PAMS the mean moment error magnitude increases radially outward, the error increases most with the MV allocator, and the least with the LP and QP allocators.

Mean direction errors are also shown in Fig. 4.4 for all allocators. It is apparent that all allocators exhibit significantly larger direction errors when $\boldsymbol{\tau}_c$ is a pure yaw command. These areas of higher mean direction error are an artifact of the higher control effectiveness in the roll channel. Consequently, a small magnitude error in the yaw channel will result in a large direction error. The PI, LP, and QP allocators show a mean direction error of about 10 deg at the two peaks. The MV allocator shows a higher mean direction error, about 14 deg. The larger mean direction error for the MV allocator is attributed to the reduced PAMS, as a large $\boldsymbol{\tau}_c$ will lead to a greater direction error. Additionally, large mean magnitude in the control allocation errors could lead to degraded tracking performance or handling qualities.

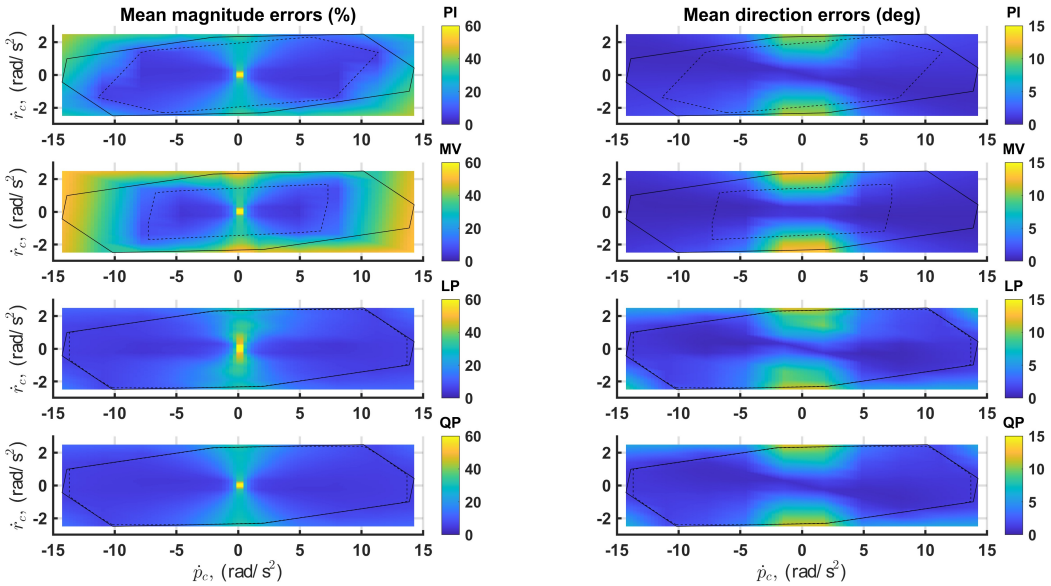



Figure 4.4: Mean error parameters.

Figure 4.5 shows comparisons of the magnitude standard errors expressed as a percentage of moment command. Near the center of the TAMS all four allocators show higher standard error which decreases as the moment command increases in roll magnitude. The LP, and QP allocators show the fastest rate of decrease

and the MV allocator shows the slowest rate of decrease. This decrease is due to the moment error being normalized by a larger moment command. As the moment command increases in the yaw channel the standard error increases, this is attributed to the high uncertainty high control effectiveness in the roll channel. The standard direction error shown in Fig. 4.5 follows a similar trend as the mean direction error shown in Fig. 4.4 for the same rationale; due to the lesser control effectiveness in the yaw channel a small magnitude error in the yaw channel will result in a large direction error.

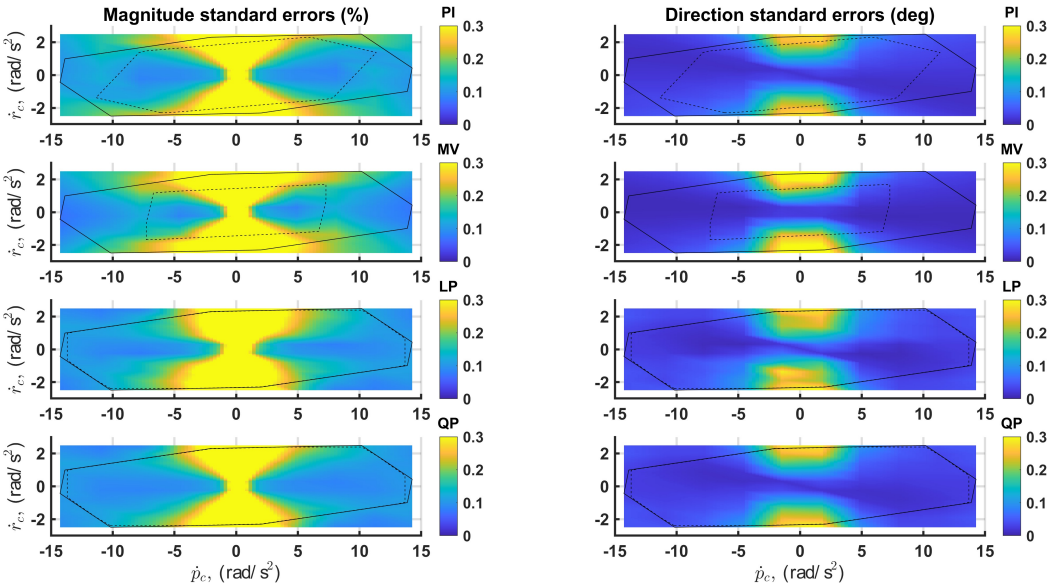



Figure 4.5: Standard error parameters.

### 4.1.2 Closed-Loop Allocator Performance

Subsection 4.1.1 investigated the static performance of the PI, MV, LP, and QP allocators. The purpose of this section is to compare the closed-loop regulation of each allocator. To generate this simulation, a proportional-derivative (PD) feedback controller was designed to regulate the aircraft states. Figure 4.6 shows a block diagram of the F-16 AFTI time-domain simulation.

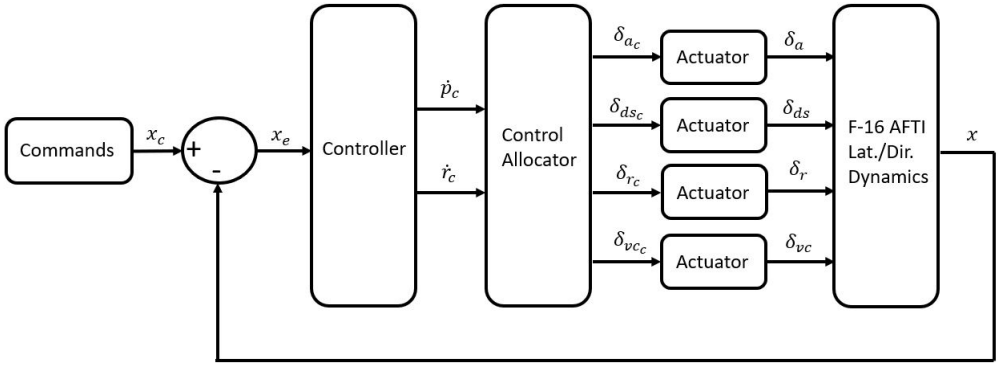



Figure 4.6: Block diagram of time-domain simulation.

To highlight the differences between the four allocation schemes an aggressive initial condition was chosen; $\beta = 30$ deg, $p = 0$ deg/s, $r = 0$ deg/s, $\phi = 0$ deg. This simulation uses a linear model, however in practice, these initial conditions may exceed the validity of the linearization point. With the nominal set of control gains, the closed-loop system returned to the equilibrium within 6 seconds. The state time history is shown in Fig. 4.7 with all four allocators and a no allocator (NA) case. The NA case was generated by feeding the virtual control commands directly into the F-16 AFTI aircraft model; this case represents the perfect allocator where all commanded moments are realized instantly. Fig. 4.7 shows that all five cases yield similar performance, the differences between the four allocators and the NA case is shown in Fig. 4.8.

Figure. 4.9 shows the control surface deflections with all four allocators. During the nominal simulation, no allocator realizes effector saturation which can be seen in Fig. 4.9. Figure. 4.10 represents a "zoomed in" view of Fig. 4.9. The subtle differences are caused by each allocator commanding a unique control deflection; due to the modeled actuator lag, the control deflection commanded by each allocator will be realized in a different time, thus resulting in subtle differences in tracking performance.

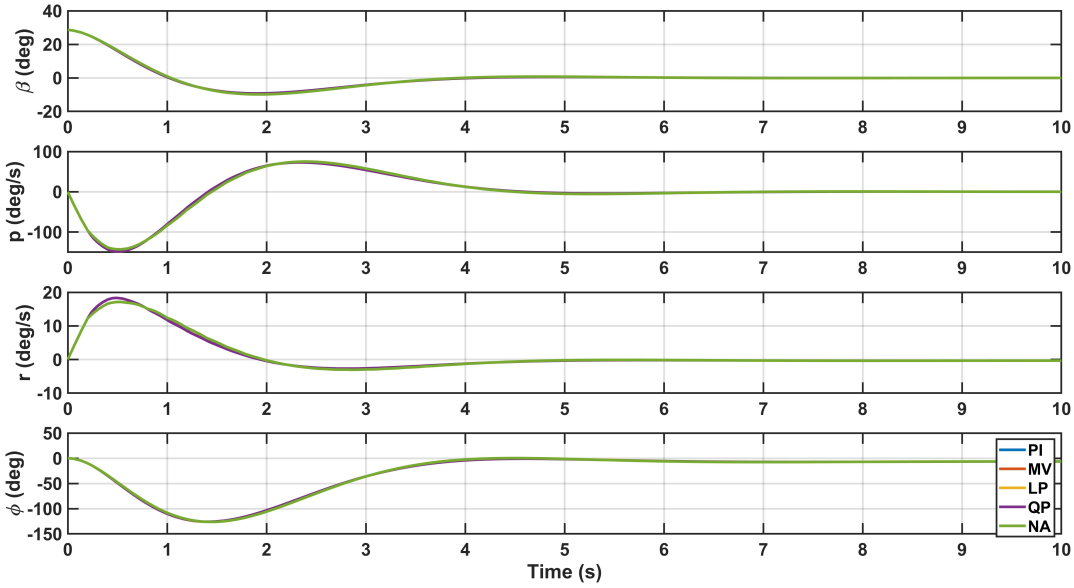



Figure 4.7: Nominal tracking performance for all allocators.

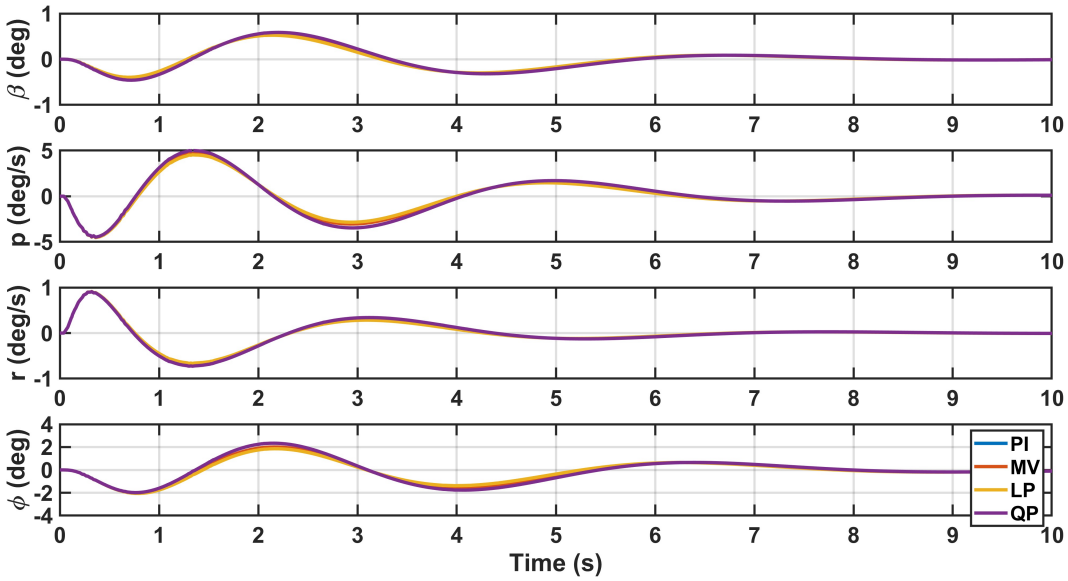



Figure 4.8: Difference between NA.

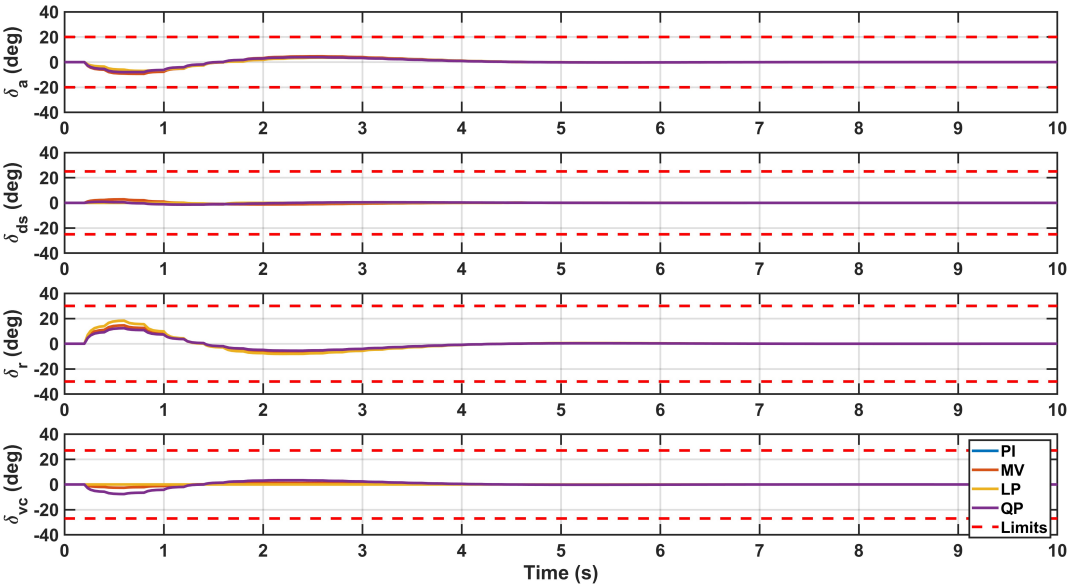



Figure 4.9: Nominal control surface deflections.

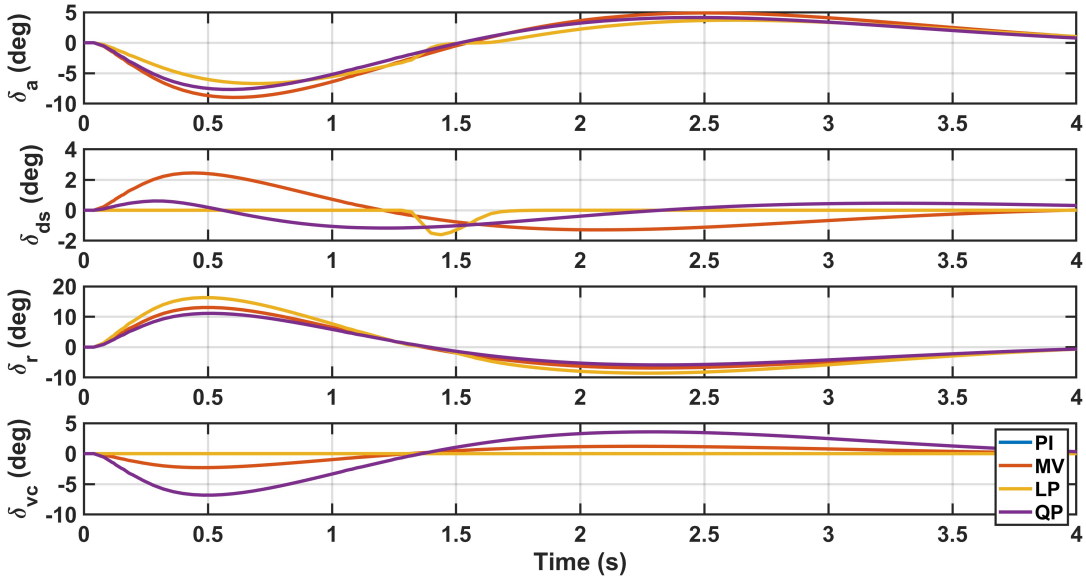



Figure 4.10: Nominal control surface deflections.

To highlight the disparity in performance in the presence of effector saturation, the control system gains were increased until effector saturation was observed. The tracking performance and control surface deflections are shown in Fig. 4.11 and Fig. 4.13. The differences between each allocator and the NA case are shown in Fig. 4.12. Figure 4.14 represents a "zoomed in" view of Fig. 4.13.

Initially there are large transient errors in the roll channel, which can be seen by $p$ and $\phi$ in Fig. 4.12. This is expected because the aircraft ailerons which have a large contribution to rolling control effectiveness saturate, which limits the aircraft's ability to regulate the roll channel. The yawing channel is most effected by the rudder, which also saturates leading to tracking errors in $r$. The tracking errors in $r$ are less significant than the tracking errors in $p$ and $\phi$ because the rudder control effectiveness is not as large as the aileron control effectiveness.

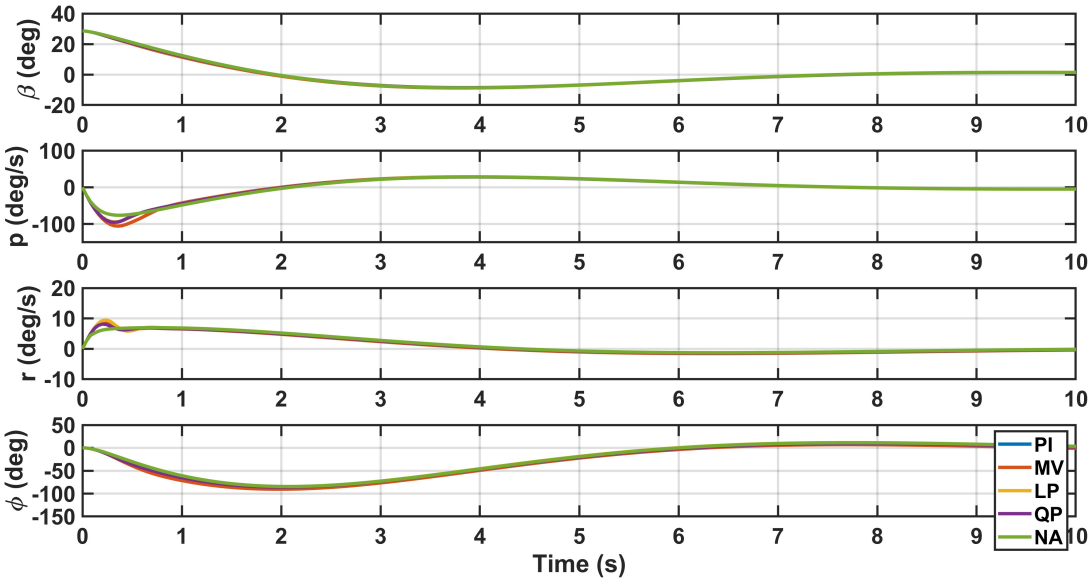



Figure 4.11: Tracking performance for increased gain case.

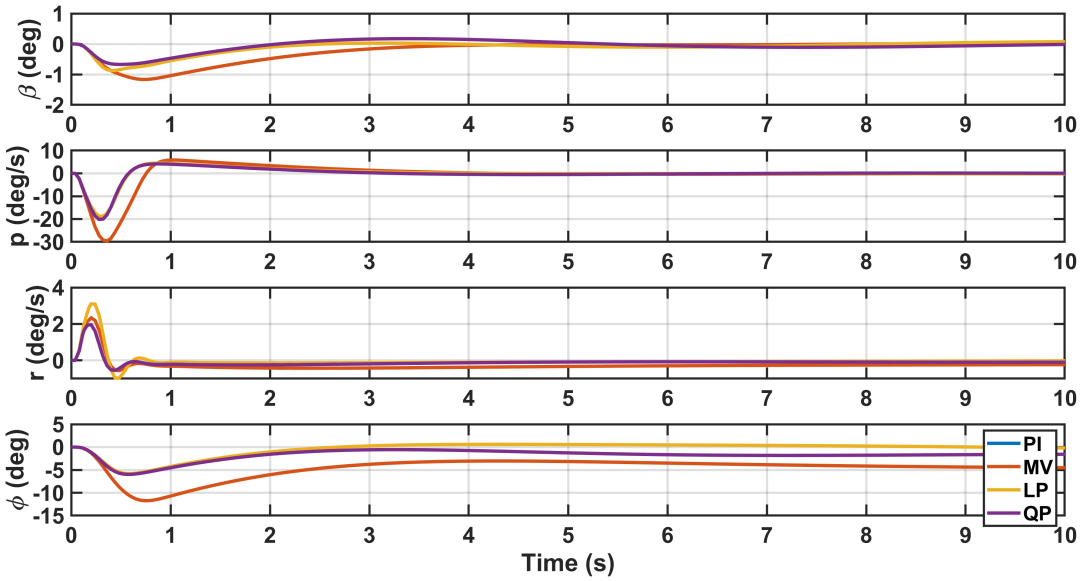



Figure 4.12: Difference between NA for increased gain case.

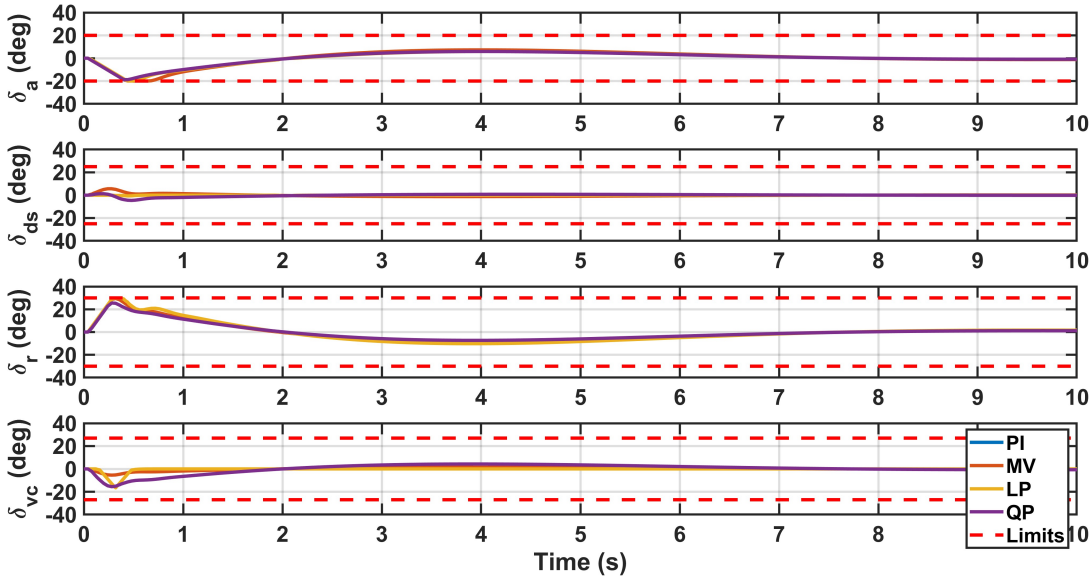



Figure 4.13: Control surface deflections for increased gain case.

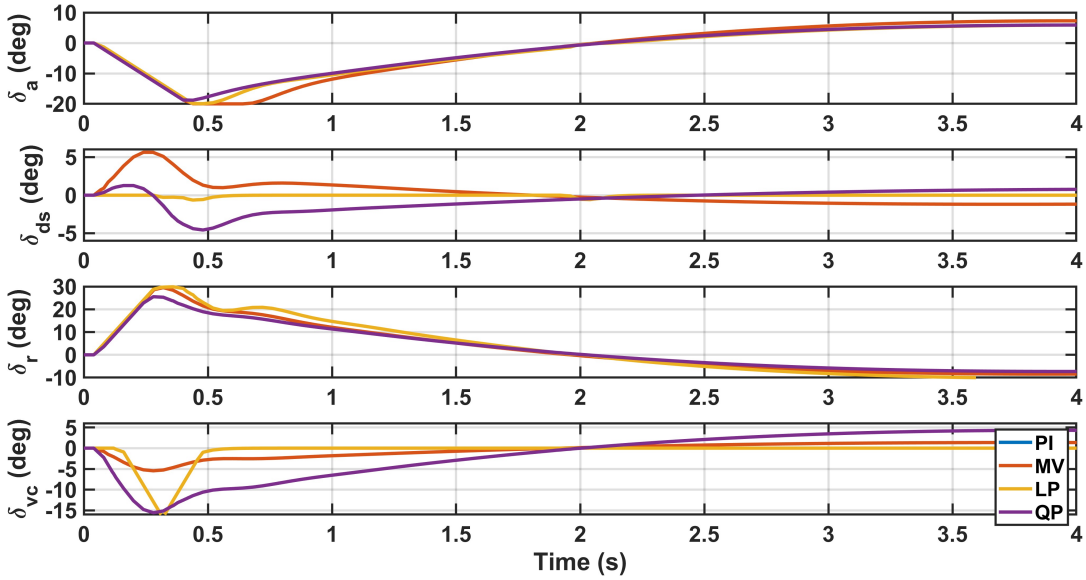



Figure 4.14: Control surface deflections for increased gain case.

The performance of the increased gain case is quantified by Table 4.3 and Table 4.4, where the allocation error is defined as the difference in states between runs with an allocator and the NA case. Both the maximum error and the integrated (cumulative) error over the 10 s simulation window are reported. The larger the maximum allocation error for a given allocator, the larger the unexpected transient response. In the case of piloting, this could lead to degraded handling qualities. The larger cumulative allocation error for a given allocator the more control effort is needed. A feedback controller would need to apply more effort to eliminate control allocation errors. Or in a piloted case, this could result in degraded handling qualities. In general, the PI, LP, and QP allocators show better performance in both maximum allocation error and cumulative allocation error as compared to the MV allocator. This is expected because the MV design is more conservative, therefore the allocator realizes a smaller PAMS. The LP and QP allocator accumulate slightly less error in $\beta$ and $\phi$, due to the brief period when the controller commands a moment which is realizable by the LP and QP allocators but not the PI allocator.

Table 4.3: Maximum allocation errors with nominal $\boldsymbol{B}$ matrix

| State | PI | MV | LP | QP |
|---|---|---|---|---|
| $\beta$ (deg) | 0.79 | 1.29 | 0.95 | 0.79 |
| p (deg/s) | 22.70 | 32.26 | 21.55 | 22.68 |
| r (deg/s) | 2.27 | 2.71 | 3.45 | 2.27 |
| $\phi$ (deg) | 6.90 | 13.00 | 6.77 | 6.84 |

To illustrate the effects of varying control effectiveness on allocator performance, 100 Monte Carlo simulations were performed with variations in the control effectiveness matrix while the control gains were set to the nominal values. To quantify the performance of the allocators across the Monte Carlo simulations, the

Table 4.4: Cumulative allocation errors with nominal $\boldsymbol{B}$ matrix

| State | PI | MV | LP | QP |
|---|---|---|---|---|
| $\beta$ (deg −s) | 2.84 | 3.71 | 2.78 | 2.83 |
| p (deg) | 19.17 | 28.89 | 19.63 | 19.08 |
| r (deg) | 2.14 | 3.97 | 2.12 | 2.13 |
| $\phi$ (deg −s) | 26.13 | 51.74 | 21.20 | 25.93 |

mean maximum allocation error and the mean cumulative allocation error were taken for each allocator. The results are shown in Table 4.5 and Table 4.6. The PI, LP, and QP allocators in this case show similar performance against variations in the control effectiveness matrix. The MV allocator, while designed to be robust to variations in the control effectiveness matrix, cannot realize large moment commands therefore it performs worse than the other allocators.

Table 4.5: Mean maximum allocation errors

| State | PI | MV | LP | QP |
|---|---|---|---|---|
| $\beta$ (deg) | 0.76 | 0.76 | 0.70 | 0.75 |
| p (deg/s) | 7.61 | 8.14 | 7.95 | 7.94 |
| r (deg/s) | 1.22 | 1.24 | 1.18 | 1.22 |
| $\phi$ (deg) | 5.45 | 5.88 | 5.64 | 5.59 |

The steady-state allocation error during the Monte Carlo simulations was also examined. To show the control effort of each allocator relative to the NA allocator case, the difference between realized moment and commanded moment at steady-state conditions is measured and the results are shown as a non-Gaussian

Table 4.6: Mean cumulative allocation errors

| State | PI | MV | LP | QP |
|---|---|---|---|---|
| $\beta$ (deg −s) | 2.21 | 2.27 | 2.07 | 2.21 |
| p (deg) | 19.43 | 20.61 | 19.13 | 19.84 |
| r (deg) | 2.17 | 2.18 | 2.02 | 2.18 |
| $\phi$ (deg −s) | 15.02 | 15.13 | 14.39 | 15.27 |

probability density function (PDF). These PDFs are shown in Figs. 4.15- 4.16. The MV allocator is characterized by a taller and thinner PDF shows less variance in control allocation error. The PI, LP, and QP allocators are able to show less error in Table 4.5 and Table 4.6, but higher variance in control allocation error. This is because the MV allocator will cause the effectors to saturate earlier, thus resulting in greater tracking error but less error at steady-state conditions.

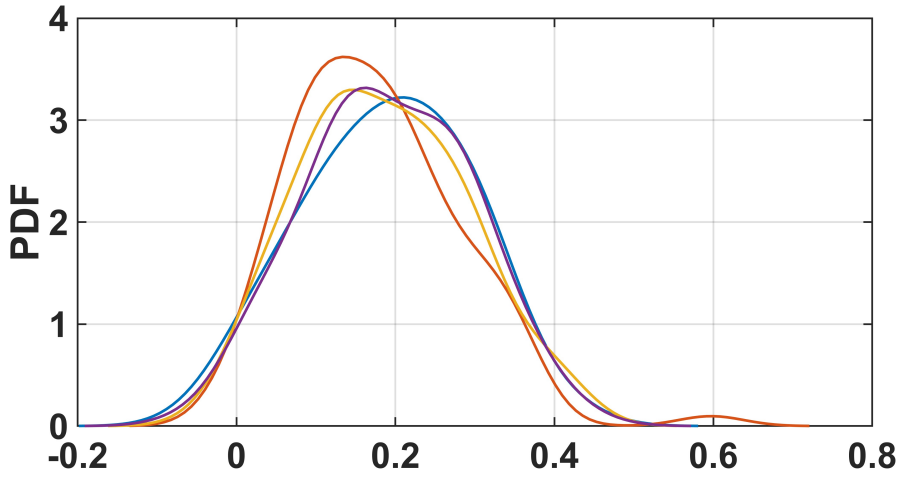



Figure 4.15: Steady-state $\dot{p}$ errors (deg/s$^2$).

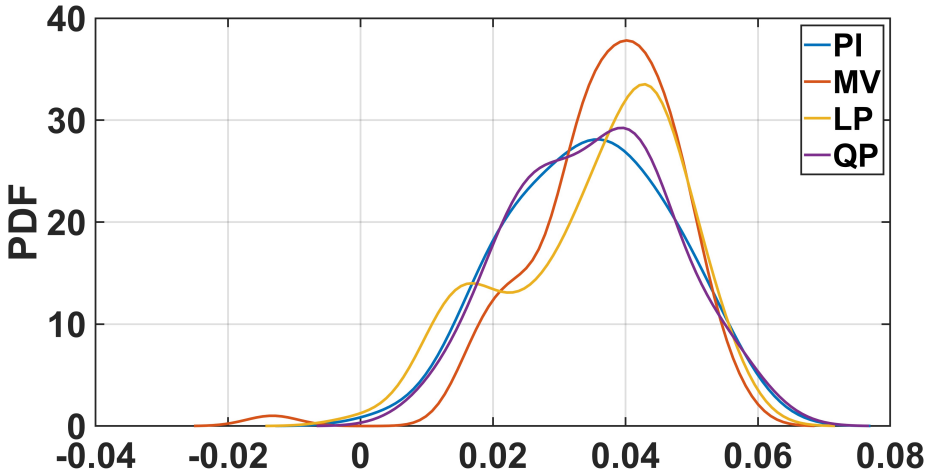



Figure 4.16: Steady-state $\dot{r}$ errors (deg/s$^2$).

## 4.2 Chapter Conclusions

In this study, several control allocation approaches were compared for systems with parametric model uncertainty in the control effectiveness matrix. Two generalized inverse approaches, the psuedo-inverse and the minimum-variance were considered. Additionally, two on-line optimization-based approaches were considered, linear and quadratic programming.

First, the static performance of the allocators were examined. In the absence of dispersions the LP and QP allocators yield the largest PAMS. The MV allocator which was originally designed for robustness to model uncertainties yielded the smallest PAMS, as expected. This implies that the MV allocator is well-suited to handle model dispersions, but might limit performance during aggressive maneuvers. When dispersions are considered the LP and QP allocators show the smallest mean magnitude errors near the edges of the TAMS. The four allocation approaches exhibited similar direction errors.

Second, the closed-loop response of each allocator was examined. Under nominal conditions it was shown that all allocators have similar performance. The control system gains were subsequently increased such that the commanded moments would exceed the boundaries of the PAMS of each allocator. For the example shown in this chapter, the LP and QP allocators showed smaller cumulative errors. This implies that the LP and QP allocators are more suitable when the aircraft is operating near the limits of the control surfaces. Additionally, the robustness to model dispersions was examined. It was found that the MV showed the least variance in allocation error, but had higher tracking errors than the PI, LP, and QP allocators.

# Chapter 5

# Control Allocation by Robust Linear Programming

## 5.1 Introduction

Optimal control allocation is a well-studied aspect of flight control design. Typically, control laws are designed to produce virtual moment commands to be imparted on a vehicle. These virtual moments commands encapsulate either forces, torques, or impulses. However, the vehicle must realize these commands and can do so through manipulation of its control effectors. Common vehicle control effectors include aerodynamic control surfaces on aircraft. Reaction control systems, thrust vectoring control systems, control moment gyros are common control effectors on spacecraft. Control allocation is the method which distributes the virtual moment commands to the individual effectors.

Generalized inverses is the most common control allocation approach which the design is based on a local linearization of the effector control effectiveness. The subsequent mapping between virtual and physical commands is derived in a

closed-form solution to a Lagrange optimization problem [51, 46]. The simplest generalized inverse is a weighted pseudo-inverse (WPI) allocator [15]. WPI involves only a matrix multiplication, hence making it attractive for real-time implementation, especially on human-rate vehicles which safety is paramount. A major drawback to generalized inverses is that effector position and rate limits are not accounted for.

The control allocation problem can also be formulated as a convex optimization problem and solved on-line using linear programming or quadratic programming techniques [25, 52]. The major benefit is that the effector position and rate limits can be accounted for [53, 54, 49, 24]. Furthermore, it has been demonstrated in Ref. [55] additional convex objectives such as minimizing the $\ell_\infty$ norm of the structural loads on the control surfaces can be incorporated into the optimization problem. Common solution techniques include interior-point methods, simplex algorithms, and more recently first-order methods such as Alternate Direction Method of Multiplier (ADMM) [56, 57, 58]. Historically, due to lack of computational power and solver maturity, the aerospace controls community has generally avoided on-line optimization as part of the flight controls architecture. Rigorous flight certification process requires convergence guarantees and solve times. The past 10–15 years has seen a sharp rise in solver technology [59, 60, 61, 62, 63]. SpaceX Falcon F9 booster landing uses model predictive control to solve the real-time trajectory optimization problem with hundreds of decision variables in fractions of a second [64]. This shift in paradigm suggests that the formulation of the optimal control allocation problem as a convex program could supplant generalized inverses as the new industry standard.

The aforementioned literature assumes no uncertainty in the elements of the controls matrix ($B_m$). Robustness to modeling errors is evaluated by perform-

ing Monte Carlo simulations of the truth model. Grauer and Pei derived the minimum-variance (MV) control allocator, which is based on generalized inverse and considers the effects of parametric model uncertainty by minimizing the variance of the generalized moment error [11]. MV was shown to outperform WLS in the presence of known parametric uncertainty in elements of $B_m$. Robust control techniques have also gained attention to help minimize the effects of bounded disturbances in the allocation problem [65, 66, 67, 68]. Proper compensation of the effects of parametric model uncertainty is critical. A realized virtual control error will create unintended motion on the vehicle and additional burden on the pilot and/or feedback control loop [69].

To the knowledge of the authors, there appears to be little documentation that discusses the effects of robust control allocation with either box of ellipsoidal uncertainty. De Almeida applied robust convex optimization to offline control allocation design [70]. Duan, Zhang, Zhao, and Kong studied on-line robust control allocation under ellipsoidal uncertainty [26]. Robust convex optimization is widely used across numerous fields such as biology, epidemiology, finance, or engineering [71, 72]. In this work, the control allocation scheme of a model aircraft is formulated as a robust optimization problem. Similar to the LP formulation as described in [25], the objective in this study is to minimize the $\ell_1$ norm of the moment error and effector usage, while satisfying effector position limits and accounting for uncertainties in elements of $B_m$. To illustrate the efficacy of the robust linear allocator, a sample control allocation problem with parametric uncertainties in its control surfaces is considered. $3\sigma$ bounds on control surface deflections are taken from previously derived examples [11]. In practice, covariance matrix of aerodynamic effectors can be derived from system identification techniques [9] either off-line or in-real time [41, 23].

The remainder of this chapter is organized as follows. Section 2.1.2 derives the linear programming allocator (LP). Section 5.2 using a box model of the uncertainty. Section 5.3 provides two examples which comparing the efficacy of the RLP allocator to WLS and MV. Lastly, section 5.4 provides concluding remarks and recommendations for future research.

## 5.2 Robust Linear Programming Allocation Design

This section derives the control allocator using robust optimization theory and ensures that a generalized moment is achieved within a prescribed tolerance given its parametric model uncertainty [71, 72]. The robust linear programming allocator is derived considering both box uncertainty sets (RLPb) and ellipsoidal uncertainty sets (RLPe).

The derivation begins with the linear programming allocator from chapter 2:

$$\min_{\boldsymbol{u},\boldsymbol{s},\boldsymbol{q}} \quad \boldsymbol{J} = \begin{bmatrix} 0\ldots0 & 1\ldots1 & \epsilon\ldots\epsilon \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{s} \\ \boldsymbol{q} \end{bmatrix} \tag{5.1a}$$

$$\text{s.t.} \quad -\boldsymbol{u} \leq -\underline{\boldsymbol{u}} \tag{5.1b}$$

$$\boldsymbol{u} \leq \overline{\boldsymbol{u}} \tag{5.1c}$$

$$\boldsymbol{B}(\boldsymbol{\theta})_i \boldsymbol{u} - s_i \leq \tau_{c,i} \quad \forall i \tag{5.1d}$$

$$-\boldsymbol{B}(\boldsymbol{\theta})_i \boldsymbol{u} - s_i \leq -\tau_{c,i} \quad \forall i \tag{5.1e}$$

$$-\boldsymbol{s} \leq \boldsymbol{0} \tag{5.1f}$$

$$u_j - q_j \leq u_{0,j} \quad \forall j \tag{5.1g}$$

$$-u_j - q_j \leq -u_{0,j} \quad \forall j \tag{5.1h}$$

**Definition 1:** The vector, $\boldsymbol{x} = \begin{bmatrix} \boldsymbol{u}^T & \boldsymbol{s}^T & \boldsymbol{q}^T \end{bmatrix}^T$ is called a *robust feasible solution* if it remains feasible for all realizations of the constraints from the uncertainty set [71]. That is,

$$\boldsymbol{B}(\boldsymbol{\theta})_i \boldsymbol{u} - \boldsymbol{s}_i \leq \tau_{c,i} \quad \forall i \tag{5.2a}$$

$$-\boldsymbol{B}(\boldsymbol{\theta})_i \boldsymbol{u} - \boldsymbol{s}_i \leq -\tau_{c,i} \quad \forall i \tag{5.2b}$$

$\forall \boldsymbol{\theta} \in \Theta$.

**Definition 2:** Assume a candidate solution, $\boldsymbol{x}$ the *robust value of the objective function* is the largest value of the objective function at $\mathbf{x}$ over all realizations from the uncertainty set [73]:

$$\boldsymbol{J} = \sup_{\boldsymbol{B}(\boldsymbol{\theta})_i \in \Theta_i} \begin{bmatrix} 0 \ldots 0 & 1 \ldots 1 & \epsilon \ldots \epsilon \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{s} \\ \boldsymbol{q} \end{bmatrix} \tag{5.3}$$

**Definition 3:** The *robust counterpart* is the problem of minimizing the *robust value of the objective function* across all *robust feasible solutions* [73]:

$$\min_{\boldsymbol{u},\boldsymbol{s},\boldsymbol{q}} \quad \boldsymbol{J} = \sup_{\boldsymbol{B}(\boldsymbol{\theta})_i \in \Theta_i} \begin{bmatrix} 0 \ldots 0 & 1 \ldots 1 & \epsilon \ldots \epsilon \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{s} \\ \boldsymbol{q} \end{bmatrix} \tag{5.4a}$$

$$\text{s.t.} \quad -\boldsymbol{u} \leq -\underline{\boldsymbol{u}} \tag{5.4b}$$

$$\boldsymbol{u} \leq \overline{\boldsymbol{u}} \tag{5.4c}$$

$$\boldsymbol{B}(\boldsymbol{\theta})_i\boldsymbol{u} - s_i \leq \tau_{c,i} \quad \forall i \tag{5.4d}$$

$$-\boldsymbol{B}(\boldsymbol{\theta})_i\boldsymbol{u} - s_i \leq -\tau_{c,i} \quad \forall i \tag{5.4e}$$

$$-\boldsymbol{s} \leq \boldsymbol{0} \tag{5.4f}$$

$$u_i - q_i \leq u_{0,i} \quad \forall i \tag{5.4g}$$

$$-u_i - q_i \leq -u_{0,i} \quad \forall i \tag{5.4h}$$

Since the objective function in (5.4a) is certain, the *robust counterpart* of the constraints in (5.4d) and (5.4e) are derived using methodology presented in [29]:

$$\max_{\boldsymbol{B}(\boldsymbol{\theta})_i \in \Theta_i} \boldsymbol{B}(\boldsymbol{\theta})_i\boldsymbol{u} - s_i \leq \tau_{c,i} \quad \forall i \tag{5.5a}$$

$$\max_{\boldsymbol{B}(\boldsymbol{\theta})_i \in \Theta_i} -\boldsymbol{B}(\boldsymbol{\theta})_i\boldsymbol{u} - s_i \leq -\tau_{c,i} \quad \forall i \tag{5.5b}$$

### 5.2.1 Box Uncertainty Sets

This section derives the robust allocator considering box uncertainty sets.

**Assumption:** $\Theta_i$ is an box uncertainty set. That is,

$$\Theta_i = \{\boldsymbol{B}(\boldsymbol{\theta})_i : ||\boldsymbol{B}(\boldsymbol{\theta})_i - \boldsymbol{B}(\hat{\boldsymbol{\theta}})_i||_\infty \leq \rho_i\} \quad \forall i \tag{5.6}$$

where the parameters $\hat{\boldsymbol{\theta}}$ and $\rho_i$ are the nominal data and size of the uncertainty, respectively [29]. In this chapter, the size of the uncertainty $\rho_i$ is chosen as 3 standard deviations of the parametric model uncertainty. This is a popular choice because it encapsulates 99.73% of possible perturbations.

$$\rho_i = 3\left[\mathbb{E}[\boldsymbol{B}(\hat{\boldsymbol{\theta}})_i\boldsymbol{B}(\hat{\boldsymbol{\theta}})_i^T]\right]^{\frac{1}{2}} \tag{5.7}$$

$\rho_i$ is determined by (5.7), where chol represents the Cholesky factorization. The maximization problem in (5.5) can be reformulated as the following constraints [29].

$$\boldsymbol{B}(\hat{\boldsymbol{\theta}})_i\boldsymbol{u} + \rho_i||\boldsymbol{u}||_1 - s_i \leq \tau_{c,i} \quad \forall i \tag{5.8a}$$

$$-\boldsymbol{B}(\hat{\boldsymbol{\theta}})_i\boldsymbol{u} + \rho_i||\boldsymbol{u}||_1 - s_i \leq -\tau_{c,i} \quad \forall i \tag{5.8b}$$

Equation (5.9) is created by replacing the constraints (5.4d) and (5.4e) with the constraints in (5.8a) and (5.8b).

$$\min_{\boldsymbol{u},\boldsymbol{s},\boldsymbol{q}} \quad \boldsymbol{J} = \begin{bmatrix} 0\ldots0 & 1\ldots1 & \epsilon\ldots\epsilon \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{s} \\ \boldsymbol{q} \end{bmatrix} \tag{5.9a}$$

$$\text{s.t.} \quad -\boldsymbol{u} \leq -\underline{\boldsymbol{u}} \tag{5.9b}$$

$$\boldsymbol{u} \leq \overline{\boldsymbol{u}} \tag{5.9c}$$

$$\boldsymbol{B}(\hat{\boldsymbol{\theta}})_i\boldsymbol{u} + \rho_i||\boldsymbol{u}||_1 - s_i \leq \tau_{c,i} \quad \forall i \tag{5.9d}$$

$$-\boldsymbol{B}(\hat{\boldsymbol{\theta}})_i\boldsymbol{u} + \rho_i||\boldsymbol{u}||_1 - s_i \leq -\tau_{c,i} \quad \forall i \tag{5.9e}$$

$$-\boldsymbol{s} \leq \boldsymbol{0} \tag{5.9f}$$

$$u_i - q_i \leq u_{0,i} \quad \forall i \tag{5.9g}$$

$$-u_i - q_i \leq -u_{0,i} \quad \forall i \tag{5.9h}$$

The robust linear programming problem in (5.9) is then solved using the `linprog` `MATLAB` command. The robust linear programming problem can also be solved on aircraft or spacecraft in real-time using simplex algorithms or interior-point algorithms[26].

### 5.2.2 Ellipsoidal Uncertainty Sets

This section derives the robust control allocator considering ellipsoidal uncertainty sets.

**Assumption:** $\Theta_i$ is an ellipsoidal uncertainty set. That is,

$$\Theta_i = \{\boldsymbol{B}(\boldsymbol{\theta})_i = \boldsymbol{B}(\hat{\boldsymbol{\theta}})_i + \boldsymbol{k}\boldsymbol{R}_i : ||\boldsymbol{k}||_2 \leq 1\} \quad \forall i \tag{5.10a}$$

where $\boldsymbol{R}_i$ is a symmetric positive-definite matrix which describes the shape of the ellipsoid about its center [29]. The matrix $\boldsymbol{R}_i$ is chosen by manipulation of the parameter covariance matrix

$$\boldsymbol{R}_i = 3\,\text{chol}\left[\mathbb{E}[\boldsymbol{B}(\hat{\boldsymbol{\theta}})_i^T \boldsymbol{B}(\hat{\boldsymbol{\theta}})_i]\right] \tag{5.11}$$

The solution to the constraints in (5.5) considering an ellipsoidal uncertainty set are [29]:

$$\boldsymbol{B}(\hat{\boldsymbol{\theta}})_i\boldsymbol{u} + ||\boldsymbol{R}_i\boldsymbol{u}||_2 - s_i \leq \tau_{c,i} \quad \forall i \tag{5.12a}$$

$$-\boldsymbol{B}(\hat{\boldsymbol{\theta}})_i\boldsymbol{u} + ||\boldsymbol{R}_i\boldsymbol{u}||_2 - s_i \leq -\tau_{c,i} \quad \forall i \tag{5.12b}$$

Equation (5.13) is created by replacing the constraints (5.4d) and (5.4e) with the constraints in (5.12a) and (5.12b).

$$\min_{\boldsymbol{u},\boldsymbol{s},\boldsymbol{q}} \quad \boldsymbol{J} = \begin{bmatrix} 0\ldots0 & 1\ldots1 & \epsilon\ldots\epsilon \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{s} \\ \boldsymbol{q} \end{bmatrix} \tag{5.13a}$$

$$\text{s.t.} \quad -\boldsymbol{u} \leq -\underline{\boldsymbol{u}} \tag{5.13b}$$

$$\boldsymbol{u} \leq \overline{\boldsymbol{u}} \tag{5.13c}$$

$$\boldsymbol{B}(\hat{\boldsymbol{\theta}})_i \boldsymbol{u} + ||\boldsymbol{R}_i \boldsymbol{u}||_2 - s_i \leq \tau_{c,i} \quad \forall i \tag{5.13d}$$

$$-\boldsymbol{B}(\hat{\boldsymbol{\theta}})_i \boldsymbol{u} + ||\boldsymbol{R}_i \boldsymbol{u}||_2 - s_i \leq -\tau_{c,i} \quad \forall i \tag{5.13e}$$

$$-\boldsymbol{s} \leq \boldsymbol{0} \tag{5.13f}$$

$$u_i - q_i \leq u_{0,i} \quad \forall i \tag{5.13g}$$

$$-u_i - q_i \leq -u_{0,i} \quad \forall i \tag{5.13h}$$

Note that (5.13) is equivalent to a SOCP. This can be solved using the `coneprog` MATLAB command, or by interior-point methods on aircraft or spacecraft in real-time.

## 5.3 Example

The example in this chapter is taken from Ref. [45] where the example is used to design the PI, MV, RLPb, and RLPe allocators. This example entails three moments and four control effectors. The $\boldsymbol{B}$ matrix used in this problem is,

$$\boldsymbol{B} = \begin{bmatrix} 0.2 & -0.2 & 0.8 & 0.1 \\ -0.5 & -0.5 & -1.0 & 0.0 \\ -0.3 & 0.3 & -0.4 & -0.5 \end{bmatrix} \tag{5.14}$$

where the control deflection limits in this problem are,

$$\underline{\boldsymbol{u}} = \begin{bmatrix} -1 & -1 & -1 & -1 \end{bmatrix} \tag{5.15a}$$

$$\overline{\boldsymbol{u}} = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} \tag{5.15b}$$

The uncertainty in the B matrix is given by the following Table 5.1 and is taken from Ref. [11]. This uncertainty is typical for that of fixed-wing aircraft [11]. The PI, MV, RLPb, and RLPe are designed based on the information in (5.14),

Table 5.1: Standard errors used in example

| | Column 1 | Column 2 | Column 3 | Column 4 |
|---|---|---|---|---|
| Row1 | 50% | 50% | 5% | 10% |
| Row2 | 5% | 10% | 20% | 25% |
| Row3 | 50% | 50% | 5% | 10% |

and (5.15). This chapter evaluates the performance of the allocators, as well as the performance of the solvers used to perform the allocation.

### 5.3.1 Allocator Performance

The allocators are evaluated according to two different criteria. The first criterion investigates the effects of allocator designs on the accuracy of the steady-state yawing moment produced. To evaluate this criterion a $\boldsymbol{\tau}_c = \begin{bmatrix} 0.05 & 0.05 & 0.05 \end{bmatrix}^T$ moment is commanded and then the allocation error is computed across 10,000 variations of the system model, where the variations are determined by the standard errors listed in Table 5.1. The second criterion evaluates how efficient the allocator design can be. The theoretical attainable moment set (TAMS) is the set of all possible moments due to all possible combinations of control effectors. The attainable moment set (AMS) is the subset of the TAMS which can be realized by the allocator. The performance of the allocator can be quantified by the ratio of the size of these two sets, $\boldsymbol{\eta}$. If the allocator can realize the entire space in the TAMS then $\boldsymbol{\eta} = 1$ and the allocator is perfectly efficient. To investigate this criterion $10,000$ points are commanded across the TAMS and the size of the AMS is compared for each allocator. The allocation efficiency for each allocator is then

tabulated in Table 5.2.

The performance of the PI, MV, RLPb, and RLPe allocators based on the two criteria are shown in Figures 5.1, 5.2, and 5.3 and also shown graphically in Tables 5.2 and 5.3.

Table 5.2: Standard Deviations in Allocation Errors

| Parameter | 1st axis | 2nd axis | 3rd axis |
|---|---|---|---|
| $\sigma(\tilde{\tau}_{PI})$ | $1.93 \times 10^{-2}$ | $1.83 \times 10^{-2}$ | $2.96 \times 10^{-2}$ |
| $\sigma(\tilde{\tau}_{MV})$ | $1.78 \times 10^{-2}$ | $1.51 \times 10^{-2}$ | $2.82 \times 10^{-2}$ |
| $\sigma(\tilde{\tau}_{RLPb})$ | $10.54 \times 10^{-2}$ | $6.97 \times 10^{-2}$ | $15.73 \times 10^{-2}$ |
| $\sigma(\tilde{\tau}_{RLPe})$ | $1.78 \times 10^{-2}$ | $1.53 \times 10^{-2}$ | $2.81 \times 10^{-2}$ |

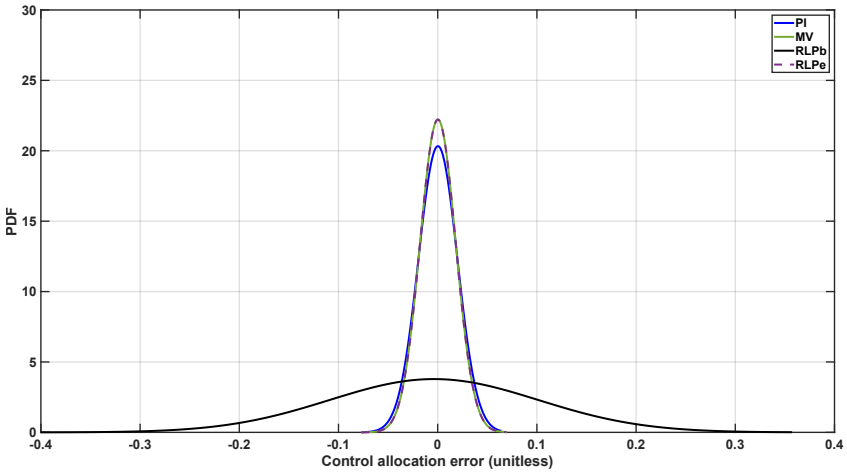



Figure 5.1: 1st axis moment error.

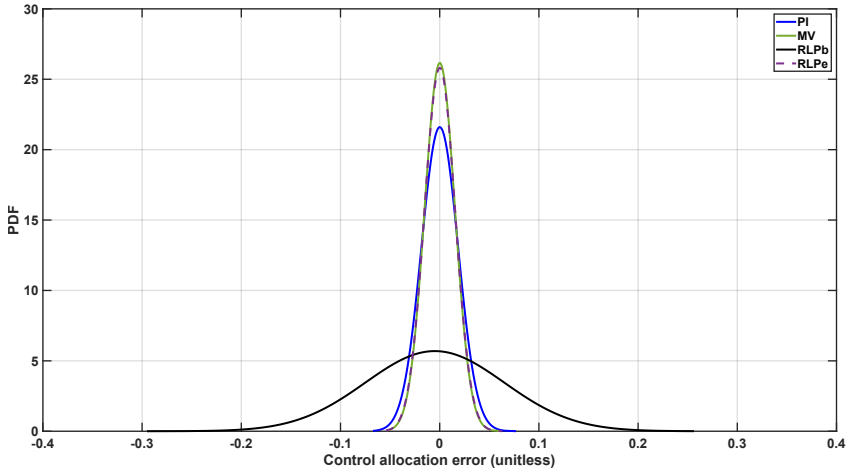



Figure 5.2: 2nd axis moment error.

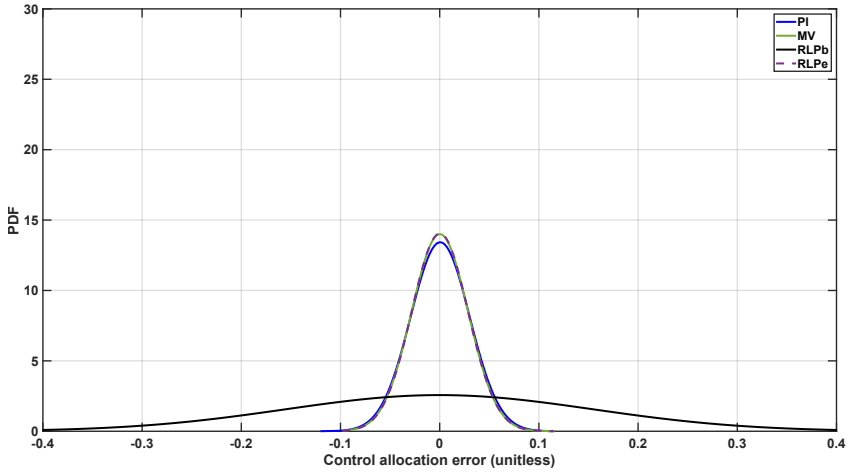



Figure 5.3: 3rd axis moment error.

Table 5.3: Allocation efficiency

| Parameter | Efficiency |
|---|---|
| $\boldsymbol{\eta}_{PI}$ | 75.38% |
| $\boldsymbol{\eta}_{MV}$ | 61.20% |
| $\boldsymbol{\eta}_{RLPb}$ | 33.51% |
| $\boldsymbol{\eta}_{RLPe}$ | 92.34% |

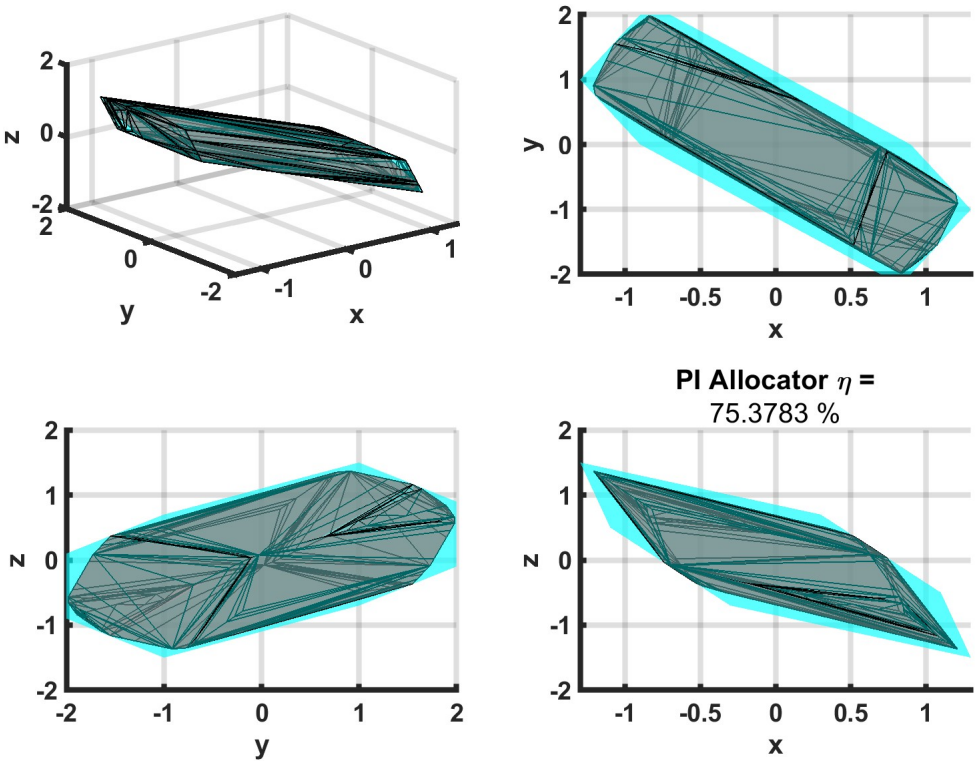



Figure 5.4: PI Allocation Efficiency.

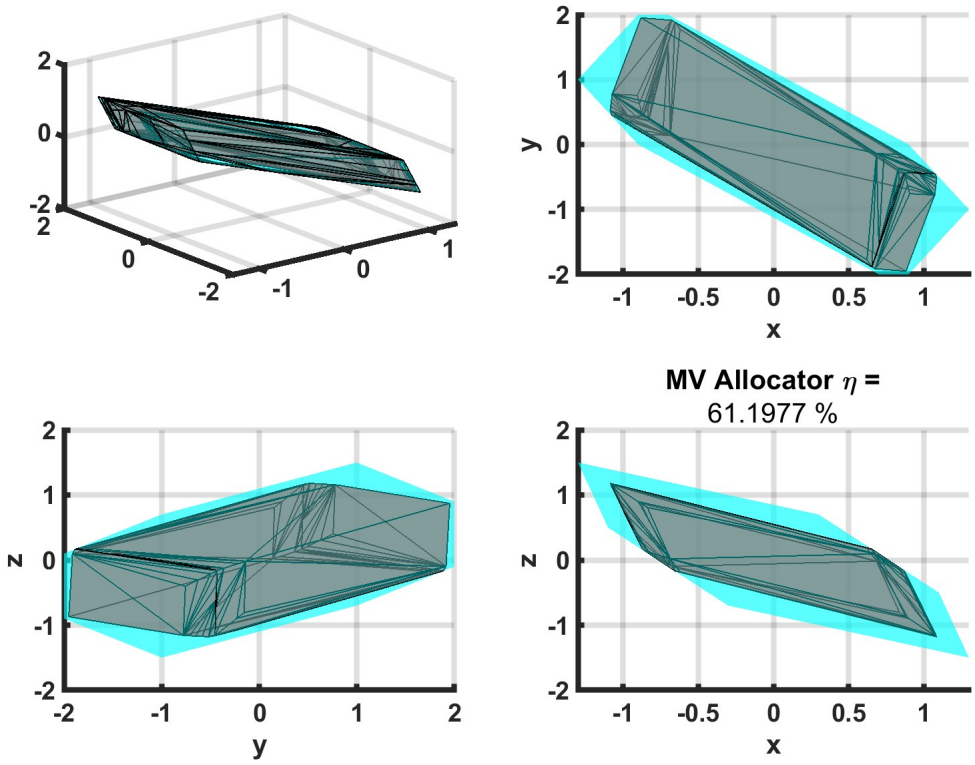



Figure 5.5: MV Allocation Efficiency.

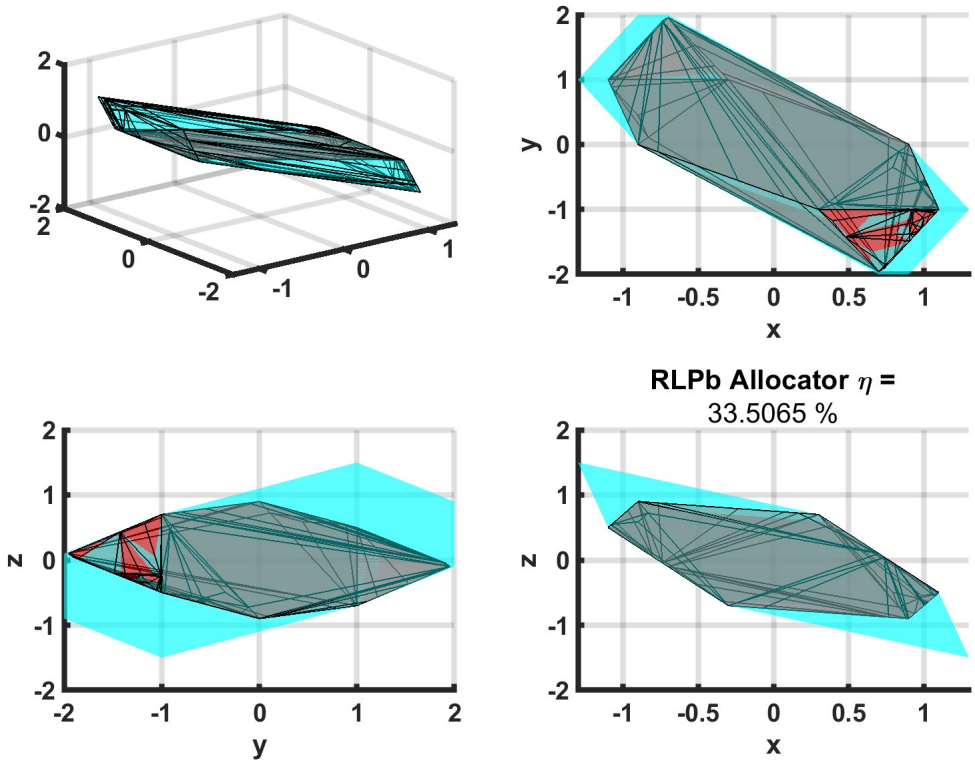



Figure 5.6: RLPb Allocation Efficiency.

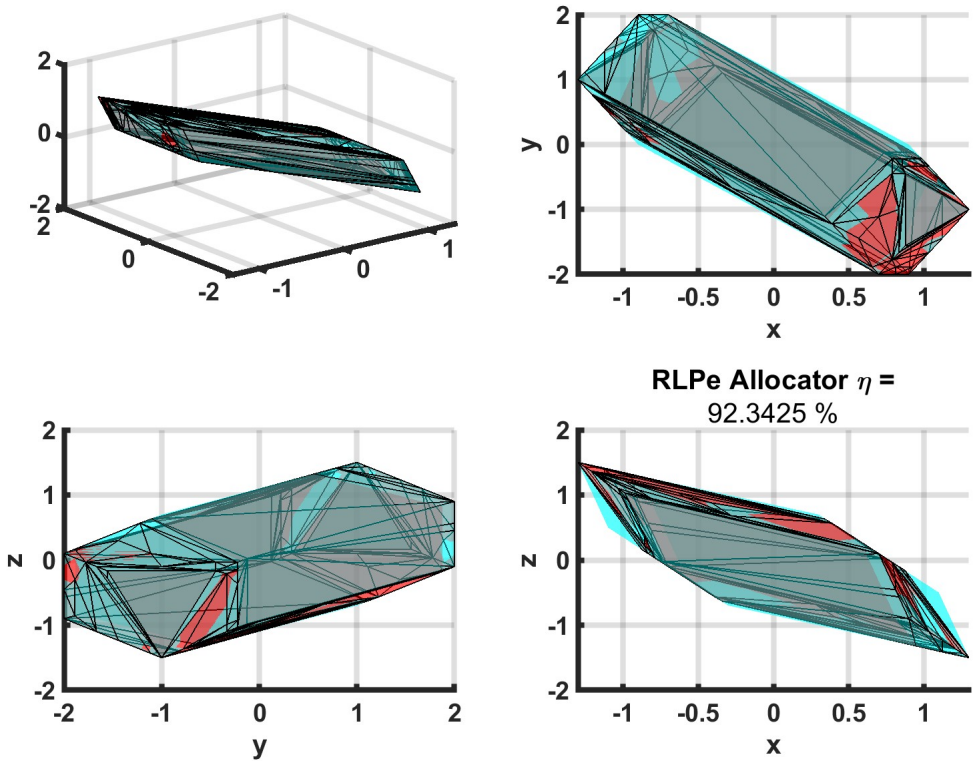



Figure 5.7: RLPe Allocation Efficiency.

The MV and RLPe allocation both had the lowest standard deviations in allocation error across all perturbations to model errors and were roughly equivalent performance. The PI allocator showed slightly higher standard deviations 8% larger in the first axis, 19% larger in in the second axis, and 5% larger in the third axis. The RLPb showed notably higher standard deviation than the MV and RLPe allocators. The RLPb is roughly 592% higher in Moment 1, 455% higher in Moment 2 and 557% higher in Moment 3. This higher standard deviation is due to the conservative nature of the box uncertainty set.

The RLPe allocator has more allocation efficiency than the MV allocator, while being just as robust as the MV allocator. The MV allocator attains 61.20% of the TAMS whereas the RLPe allocator obtains 92.34% of the TAMS. The PI allocator is 75.38% efficient, meaning is it more efficient than the MV allocator while more susceptible to model uncertainties. The RLPb allocator is 33.51% efficient, less efficient than the PI, MV, or RLPe allocators.

### 5.3.2 Solver Performance

This section evaluates the performance of the solvers when performing the RLPe allocation problem. The second order cone program in (5.13) is solved using the `coneprog` command in MATLAB and the solution algorithms are compared. The options available are: Augmented step form solver (augmented), Normal step form solver (normal), Normal step form solver using dense linear algebra (dense), Product form Cholesky step solver (prodchol), and Schur compliment method solver. More information about the solvers can be found in Refs. [74, 75, 76, 77]. The RLPe problem is solved 10,000 times for each solver and the solution times are shown in Figure 5.8.

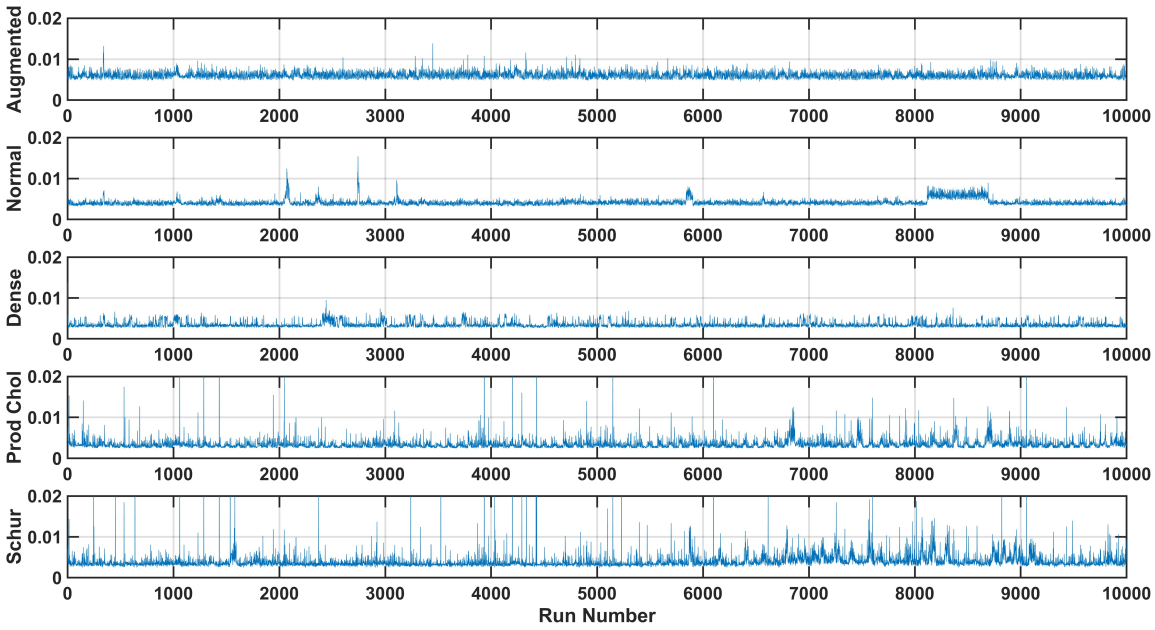



Figure 5.8: Solution times in seconds with `coneprog` command.

Figure 5.8 shosw that all solution cases show solution times less than 0.02sec, which is suitable for real-time implementation on an aircraft, where the flight control system can be designed at 50Hz. Additionally the Augmented, Normal, and Normal-Dense solvers show less variation in solution times and thus would be more reliable for real-time implementation on hardware.

## 5.4 Chapter Conclusions

An on-line control allocation design approach was developed for systems which are linear in control inputs and are subject to parametric model uncertainty. Two formulations of parametric model uncertainty are were considered, a box formulation and an ellipsoidal formulation. The control allocation approach was evaluated on two criteria. To evaluate the first criterion, a simulation was run where the system dynamics were varied and the performance in the presence of uncertainty was

compared. To evaluate the second criterion additional simulations were run where the system considered all moments in the TAMS and the allocation efficiency was compared for each allocator.

The RLPb allocator showed notably lesser performance than the PI, MV, and RLPe allocators due to its conservative uncertainty design. It is important to still note that it could be used in cases high uncertainty and low performance requirements such as aircraft with no *a priori* dynamic model. Initially the RLPb allocator could be used, then as the model improves in flight the aircraft could switch to the RLPe allocator.

The RLPe allocator was found to have similar performance against model uncertainty as the legacy MV allocator while simultaneously having greater allocation efficiency. The RLPe allocator will help enable high-performance and high-maneuverability in vehicles while successfully handling the effects of parametric model uncertainty.

On-line control allocation design can greatly improve allocation efficiency, while also providing robustness to modeling errors. Additionally, with incorrect information about the model uncertainty the RLPe allocator can still be used, but it may not show the best performance in the presence of parametric model uncertainty. The RLPe allocator has similar performance against modeling errors than the legacy MV allocator but also has a better allocation efficiency than the MV allocator which will allow for greater vehicle maneuverability and performance.

# Chapter 6

# High-Efficiency Direction-Preserving Control Allocation

## 6.1 Introduction

In aerospace applications control laws have been traditionally designed to produce generalized moment commands. Common generalized moment commands are moments, forces, torques or impulses. These commands can be realized on vehicles through manipulation of available control effectors. Common vehicle control effectors are aerodynamic control surfaces on aircraft and reaction control surfaces on spacecraft. Control allocation is the mapping between generalized moments and the commands to the individual effectors.

Generalized inverse control allocators are the most common control allocation approach which utilize a local linearization of control effectiveness. The advantages of generalized inverse control allocation is that the control allocator is a closed-

form, linear transformation allocator, which is derived as a solution to a Lagrange optimization problem. The simplest generalized inverse is a pseudo-inverse (PI) allocator [51, 46, 15]. This is practical for real-time implementation, however generalized inverses cannot access all of the attainable generalized moment commands, even if there is a set of admissible control effector positions available. This can limit flight vehicle performance.

High-efficiency generalized inverse control allocation has also been achieved by null space intersection (NI) control allocation methods. NI methods utilize the intersection of the null space of the control effectiveness matrix with the set of admissible controls. These null space intersection methods are strictly linear and also closed-form, thus easy to implement, but determination of the null space intersection is computationally expensive and NI-based control allocation not direction-preserving [78, 79, 80].

Control allocation is also commonly formulated as a convex optimization-based linear programming (LP) problem and is solved on-line using convex optimization solvers [25]. The main advantage of the LP approach is that nearly all possible generalized moment commands are attainable, however the resulting generalized moment will not necessarily be co-linear with the command. This lack of co-linearity could degrade key performance metrics such as piloted handling qualities on aircraft, or increase fuel costs on spacecraft.

The contribution of this chapter is to derive the Linear Programming Direction-Preserving allocator (LPDP) which can achieve a large portion of attainable generalized moments while the resulting generalized moment is co-linear with the commanded generalized moment. Ensuring co-linearity between the resulting and commanded generalized moment will help improve performance metrics such as piloted handling qualities on aircraft or propellant savings on spacecraft. Simi-

lar results have been achieved by scaling the generalized moment command as in Ref. [80], however the SOCP formulation in this chapter shows how non-convex constraints can be implemented in the control allocation problem by introducing convex relaxations.

The resulting LPDP allocator is formulated as a second-order cone programming problem (SOCP). SOCP are a class of convex-optimization problems which have low complexity and can be solved by interior-point methods which can guarantee global polynomial convergence [30, 31, 32]. SOCP solution algorithms are promising for real-time on-board computations and have been applied to real-time trajectory optimization problems for Mars landing [81].

The remainder of this chapter is organized as follows. Section 6.2 reviews the derivation of the LP allocator. Section 6.3 shows the derivation of the LPDP allocator. Section 6.4 compares the performance of the PI, NI, LP, and LPDP allocators on an aircraft model. Section 6.5 shows computational time of SOCP algorithm used. Lastly, section 6.6 provides concluding remarks.

## 6.2 Control Allocation by Linear Programming

In general, flight vehicle equations of motion can be written as

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\tau}) \tag{6.1a}$$

$$\boldsymbol{\tau} = \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}) \tag{6.1b}$$

where, $\boldsymbol{x} \in \mathbb{R}^{n_x}$ is the state vector, $\boldsymbol{u} \in \mathbb{R}^{n_u}$ is the control effector deflection vector, $\boldsymbol{\tau} \in \mathbb{R}^{n_\tau}$ represent generalized moments resulting from vehicle states and effectors, and $\boldsymbol{f} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_\tau} \rightarrow \mathbb{R}^{n_x}$ are the vehicle dynamics. The control effectors are

required to satisfy box constraints,

$$\underline{\boldsymbol{u}} \leq \boldsymbol{u} \leq \overline{\boldsymbol{u}}, \tag{6.2}$$

for lower and upper limits $\underline{\boldsymbol{u}}, \overline{\boldsymbol{u}} \in \mathbb{R}^{n_u}$.

We consider the typical case where the vehicle control effectors appear linearly in the equations of motion (or the allocation design model is based on linearization), that is,

$$\boldsymbol{\tau} = \boldsymbol{B}\boldsymbol{u}, \qquad \boldsymbol{B} \in \mathbb{R}^{n_\tau \times n_u}. \tag{6.3}$$

For a control allocation solution to exist it is assumed that the $\boldsymbol{B}$ matrix has full row rank. A control system commands a generalized moment, $\boldsymbol{\tau}_c \in \mathbb{R}^{n_\tau}$, to be imparted to the vehicle control effectors. The control allocation problem is to find a mapping between the generalized moment commands and the control effectors, i.e., find $\boldsymbol{u}$ such that $\boldsymbol{B}\boldsymbol{u} \approx \boldsymbol{\tau}_c$.

A common formulation of the control allocation problem is to minimize the $\ell_1$ norm of the allocation error [25, 24] and then transform the objective into a standard form LP problem. This allows the problem to be solved in real-time using LP solvers which are commonly available in commercial software.

The LP problem is set up as a mixed optimization problem where the designer can satisfy multiple optimization objectives [25]:

$$\min_{\boldsymbol{u}} \quad \|\boldsymbol{B}\boldsymbol{u} - \boldsymbol{\tau}_c\|_1 + \epsilon\|\boldsymbol{u} - \boldsymbol{u}_0\|_1 \tag{6.4a}$$

$$\text{s.t.} \quad \underline{\boldsymbol{u}} \leq \boldsymbol{u} \leq \overline{\boldsymbol{u}} \tag{6.4b}$$

Here $\boldsymbol{u}_0$ represents the preferred control surface deflection, which is nominally zero but could be chosen as the current control surface deflection so the control allocator

could command a solution which requires only a small movement by the effector, thereby mitigating potential effector rate saturation. $\epsilon \geq 0$ is a small weighting parameter which allows the control designer to choose between a solution which is close to preferred control surface deflections, or prioritizing minimizing moment error.

Equation (2.7) can be parsed into a standard-form LP problem as described in [25]. This allows one to solve it in real-time using an interior-point-based method, of which mature implementations exist.

## 6.3 Linear Programming Direction-Preserving Control Allocation

The linear programming allocator can access large sets of attainable moments, but in general will not result in a realized generalized moment which is co-linear with the command. This section derives the LPDP allocator which is shown to be highly-efficient while resulting in a realized moment which is co-linear with the generalized moment command. The LPDP allocator is a form of the LP allocator augmented with an additional constraint to preserve co-linearity.

### 6.3.1 Problem Formulation

To derive the direction-preserving constraint, begin with the dot product rule between commanded moment, $\boldsymbol{\tau}_c$, and realized moment, $\boldsymbol{\tau}_r \coloneqq \boldsymbol{B}\boldsymbol{u}$:

$$\boldsymbol{\tau}_c^T \boldsymbol{\tau}_r = \|\boldsymbol{\tau}_c\|_2 \|\boldsymbol{\tau}_r\|_2 \cos\theta \tag{6.5}$$

where $\theta$ is the angle between $\boldsymbol{\tau}_c$ and $\boldsymbol{\tau}_r$. It is clear that when $\boldsymbol{\tau}_c$ and $\boldsymbol{\tau}_r$ are co-linear, $\cos\theta = 1$. Substituting $\cos\theta = 1$ and $\boldsymbol{\tau}_r = \boldsymbol{B}\boldsymbol{u}$ into eq. (6.5) leads to

$$\|\boldsymbol{\tau}_c\|_2 \|\boldsymbol{B}\boldsymbol{u}\|_2 = \boldsymbol{\tau}_c^T \boldsymbol{B}\boldsymbol{u}. \tag{6.6}$$

Equation (6.6) is a non-convex equality constraint which when satisfied will ensure co-linearity between $\boldsymbol{\tau_c}$ and $\boldsymbol{\tau_r}$. To implement eq. (6.6) in a real-time solver, we introduce the convex relaxation,

$$\|\boldsymbol{\tau}_c\|_2 \|\boldsymbol{B}\boldsymbol{u}\|_2 \leq \|\boldsymbol{\tau}_c\|_2^2 \gamma + \boldsymbol{\tau}_c^T \boldsymbol{B}\boldsymbol{u}. \tag{6.7}$$

where $\gamma \geq 0$ is a variable to be minimized. Notice that the direction error is zero if $\gamma = 0$, the non-convex equality constraint in eq. (6.6) is satisfied and the direction error is zero. If $\gamma > 0$, the direction error will be non-zero. Finally, the constant multiplier $\|\boldsymbol{\tau}_c\|_2^2$ ensures that $\gamma$ is unitless and improves problem scaling.

The LPDP allocator is now formed by augmenting the LP allocator in eq. (2.7) with the second-order cone constraint in eq. (6.7):

$$\min_{\boldsymbol{u},\gamma} \quad \|\boldsymbol{B}\boldsymbol{u} - \boldsymbol{\tau}_c\|_1 + \epsilon\|\boldsymbol{u} - \boldsymbol{u}_0\|_1 + \xi\left(h + \|\boldsymbol{\tau}_c\|_1\right)\gamma \tag{6.8a}$$

$$\text{s.t.} \quad \underline{\boldsymbol{u}} \leq \boldsymbol{u} \leq \overline{\boldsymbol{u}} \tag{6.8b}$$

$$\|\boldsymbol{\tau}_c\|_2 \|\boldsymbol{B}\boldsymbol{u}\|_2 \leq \|\boldsymbol{\tau}_c\|_2^2 \gamma + \boldsymbol{\tau}_c^T \boldsymbol{B}\boldsymbol{u} \tag{6.8c}$$

Here $\xi > 0$ is a parameter used to weight direction priorities in the cost function. In particular, as $\xi$ is increased, more emphasis is placed on reducing direction error compared to the LP allocator eq. (2.7), at the cost of somewhat greater moment error. $h > 0$ is a small constant which keeps $\gamma$ near zero when $\|\boldsymbol{\tau}_c\|_1$ is small

(the constant multiplier $\|\boldsymbol{\tau}_c\|_1$ in the cost function matches scaling between terms, improving computational performance for different command magnitudes).

It is clear to see that eq. (6.8) is a SOCP and can be solved in real-time by conic optimization solvers with guaranteed convergence properties [30, 31, 32]. In section 6.5 it is demonstrated that the SOCP problem eq. (6.8) can be solved efficiently, with only a small increase in computational cost compared to the LP problem eq. (2.7).

**Remark:** It is possible to derive an expression relating $\gamma$, $\cos\theta$, and the relative magnitudes of $\boldsymbol{\tau}_r$ and $\boldsymbol{\tau}_c$. To this end, we substitute eq. (6.5) into eq. (6.7) and solve for

$$\cos\theta \geq 1 - \gamma \frac{\|\boldsymbol{\tau}_c\|_2}{\|\boldsymbol{\tau}_r\|_2}, \tag{6.9}$$

where we assume that $\|\boldsymbol{\tau}_c\|_2, \|\boldsymbol{\tau}_r\|_2 > 0$. While the inequality eq. (6.9) alone does not provide a global bound on $\cos\theta$, it shows how $\gamma$ controls the tradeoff between direction and magnitude errors when $\boldsymbol{\tau}_c$ is not realizable. Futhermore, such constraints can be explicitly implemented as discussed in section 6.3.2, below.

### 6.3.2 Worst-Case Direction Error Constraint

While in practice we find that the relaxed constraint eq. (6.7) leads to very small direction error, if desired, it is also possible to directly enforce a constraint on the worst-case direction error. In particular, suppose we would like to bound $|\theta| \leq \theta_0$ for some $\theta_0 > 0$. Equivalently, $\cos\theta \geq \cos\theta_0$, which we substitute into eq. (6.5) to obtain a second-order cone constraint,

$$\|\boldsymbol{\tau}_c\|_2 \|\boldsymbol{B}\boldsymbol{u}\|_2 \cos\theta_0 \leq \boldsymbol{\tau}_c^T \boldsymbol{B}\boldsymbol{u}. \tag{6.10}$$

This additional constraint is easy to implement and does not conflict with the relaxed constraint eq. (6.7). However, we did not use it in the numerical simulations presented in section 6.4, and still observed very small direction error. Such a constraint could be useful in practice, and allow one to use a smaller direction error weight, $\xi$.

## 6.4 Simulation Example

The example of this chapter uses the F-16 Advanced Fighter Technology Integration (AFTI) aircraft model, shown in fig. 6.1. We focus on the linear lateral-directional dynamics with generalized moments $\boldsymbol{\tau} = \begin{bmatrix} \dot{p} & \dot{r} \end{bmatrix}^T$, where $\dot{p}$ is the body-axis roll acceleration and $\dot{r}$ is the body-axis yaw acceleration. The control deflection vector is $\boldsymbol{u} = \begin{bmatrix} u_a & u_{ds} & u_r & u_{vc} \end{bmatrix}^T$, where $u_a$ denotes anti-symmetric ailerons, $u_{ds}$ denotes differential stabilator, $u_r$ denotes rudder, and $u_{vc}$ denotes vertical canards. The control surface deflection limits are listed in table 6.1, and the control effectiveness matrix, $\boldsymbol{B}$, is evaluated at Mach 0.6 and 30,000 ft altitude [1]:

$$\boldsymbol{B} = \begin{bmatrix} -17.5 & -13.6 & 3.92 & 0.415 \\ -0.268 & -1.51 & -1.97 & 1.51 \end{bmatrix}. \tag{6.11}$$

In this chapter the parameters are set to $\epsilon = \|\boldsymbol{B}\|_1 10^{-6} = 1.75 \times 10^{-5}$, $\xi = 10^3$, and $h = 10^{-6}$. The direction error weight, $\xi$, was tuned to improve direction error while maintaining high-efficiency was observed.

Table 6.1: F-16 AFTI aircraft control surface position limits [1]

| Control Surface | Position Limit (deg) |
|---|---|
| Ailerons | ±20 |
| Stabilators | ±25 |
| Rudder | ±30 |
| Vertical Canards | ±27 |




Figure 6.1: F-16 AFTI aircraft in flight (credit: NASA).

### 6.4.1 Attainable Moment Sets and Allocation Efficiency

To evaluate the performance of the PI, NI, LP, and LPDP allocators, begin by defining the Theoretical Attainable Moment Set (TAMS) as the set of all possible achievable generalized moments due to all possible control effector deflections. Define the Particular Attainable Moment Set (PAMS) as the subset of the TAMS which can be realized by the specific allocator. This chapter compares the allocation efficiency of each allocator. The *allocation efficiency* is defined as the ratio

Table 6.2: Efficiency (area of PAMS / area of TAMS) of each allocator applied to the F-16 AFTI example problem

| Allocator | Efficiency (%) |
|---|---|
| PI | 64.56 |
| NI | 91.35 |
| LP | 96.93 |
| LPDP | 96.93 |

of the volume of the PAMS to the TAMS. If the allocator can realize the entire space in the TAMS then the allocator is perfectly efficient. The efficiency of each allocator is reported in table 6.2.

As expected, the PI allocator is the least efficient, while the NI, LP, and LPDP allocators can realize a much greater percentage of the TAMS. In this example, the efficiency of the LPDP allocator is virtually identical to the LP allocator, although as seen below, the error magnitude *outside* the PAMS is greater for LPDP than for LP, since LPDP seeks to improve the direction error when it cannot achieve a commanded moment.

### 6.4.2 Magnitude and Direction Errors

Next we quantify the magnitude and direction of the allocation error, which are defined as

$$\text{magnitude error} \coloneqq \frac{\|\boldsymbol{B}\boldsymbol{u} - \boldsymbol{\tau}_c\|_1}{\max\left(10^{-6}, \|\boldsymbol{\tau}_c\|_1\right)} \tag{6.12}$$

and

$$\text{direction error} \coloneqq \cos^{-1}\left(\frac{\boldsymbol{\tau}_c^T \boldsymbol{B}\boldsymbol{u}}{\max\left(10^{-6}, \|\boldsymbol{\tau}_c\|_2 \|\boldsymbol{B}u\|_2\right)}\right). \tag{6.13}$$

Of particular interest is the behavior *outside* each allocator's PAMS, as this can influence stability, performance, and handling qualities. To illustrate this idea, a sample generalized moment $\boldsymbol{\tau}_c = \begin{bmatrix} 13 & 2 \end{bmatrix}^T$ rad/s$^2$ is commanded to the

four allocators. Note that this command is outside the TAMS of the F-16 AFTI aircraft and is thus not realizable by any allocator, but could still be commanded by the aircraft control system. The realized moments are shown in fig. 6.2 and the specific magnitude and direction errors for this command are tabulated in table 6.3.

This example illustrates the significant improvement in direction error that LPDP allocation provides relative to LP allocation, at the cost of a small increase in relative magnitude error. A small remaining direction error is caused by a small but non-zero value of $\gamma$.

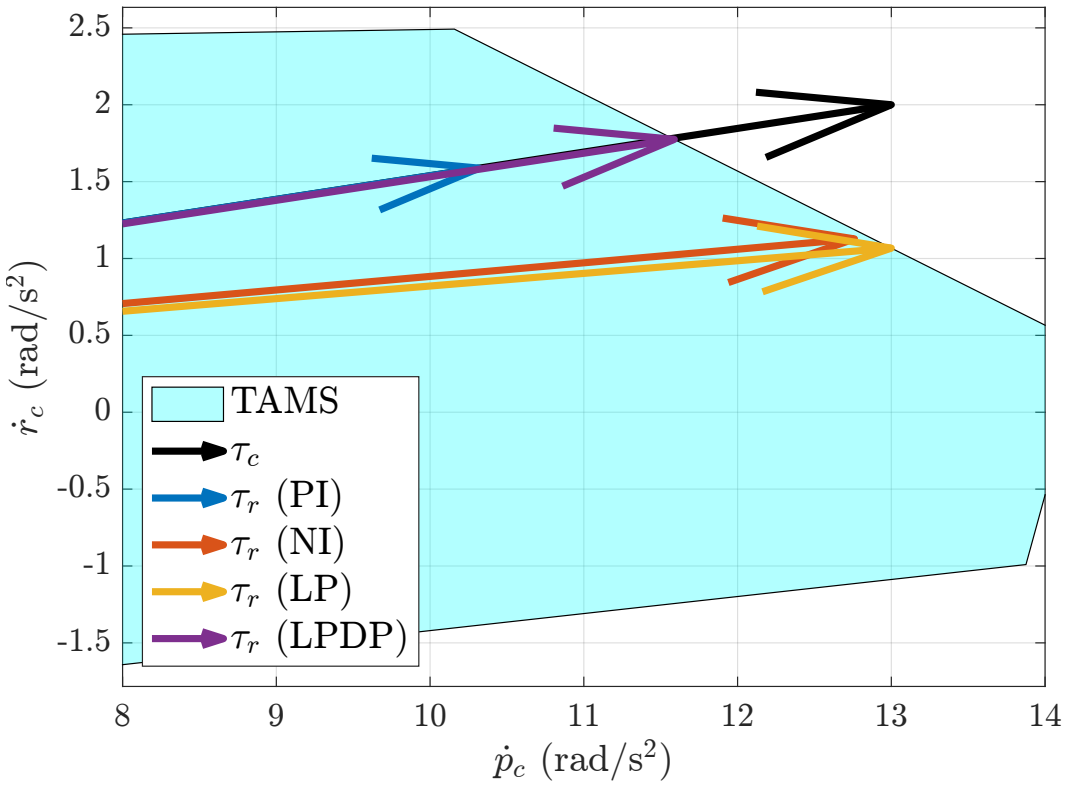



Figure 6.2: Comparison of realized generalized moments for a representative unachievable moment command.

Next, the magnitude and direction error were characterized across a range of generalized moment commands which encompasses the TAMS, $\boldsymbol{\tau}_c \in [-15, 15] \times [-3, 3]$ rad/s$^2$. The magnitude errors—expressed as a percentage of control command—

Table 6.3: Magnitude and Direction Errors for a Representative Generalized Moment Command

| Allocator | Magnitude Error (%) | Direction Error (deg) |
|---|---|---|
| PI | 20.6188 | 0 |
| NI | 7.4140 | 3.6971 |
| LP | 6.2179 | 4.0526 |
| LPDP | 10.9111 | 0.0387 |

are shown in fig. 6.3 and fig. 6.4. Direction errors are shown in fig. 6.5 and fig. 6.6. These plots also include the TAMS and the approximate PAMS of each allocator.

By definition, each allocator's magnitude and direction error are negligible within their respective PAMS. Outside the PAMS and TAMS, however, the behavior of each allocator is noticably different.

First, the PI allocator has zero direction error both inside and outside the TAMS. This is because the effectors are saturated proportionally to the most saturated effector, allowing generalized moment direction to be preserved. Next, while the NI allocator has a larger PAMS than the PI allocator, outside its PAMS, magnitude and direction errors increase dramatically and unpredictably—which is highly undesirable.

As expected, the LP allocator has the lowest magnitude error for all allocators, as it is specifically formulated to minimize the $\ell_1$ error. Outside its PAMS, the direction error of the LP allocator increases noticably. In comparison, the magnitude error of the LPDP allocator increases faster outside its PAMS due to the mixed-objective formulation. This tradeoff allows the LPDP allocator to maintain very small direction error inside and outside the PAMS. This property is expected to preserve vehicle dynamics and handling qualities.

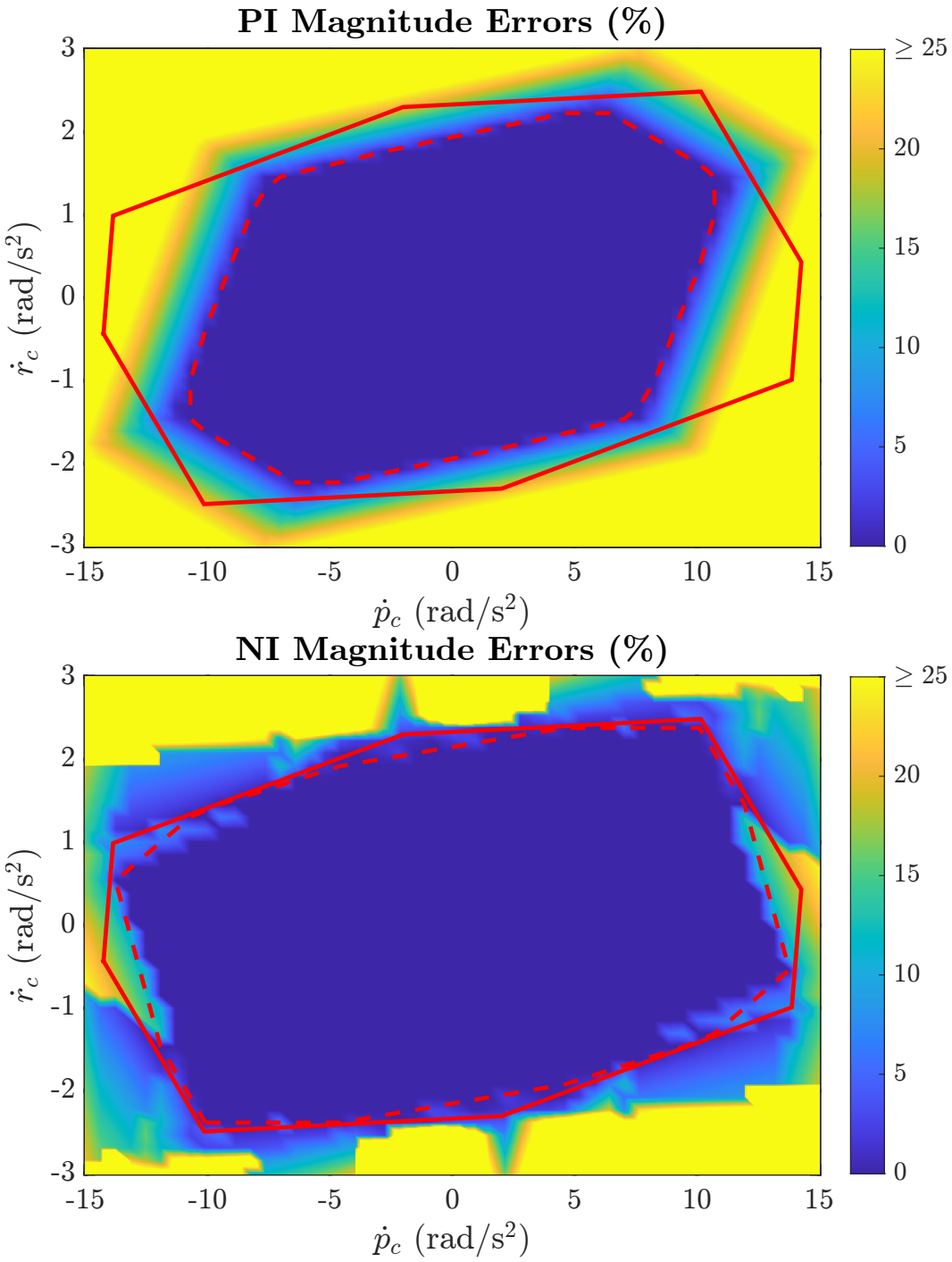



Figure 6.3: Relative magnitude errors for the PI and NI allocators. The TAMS of the F-16 AFTI aircraft is shown as a solid red line, and the approximate PAMS of each allocator as a dotted red line.

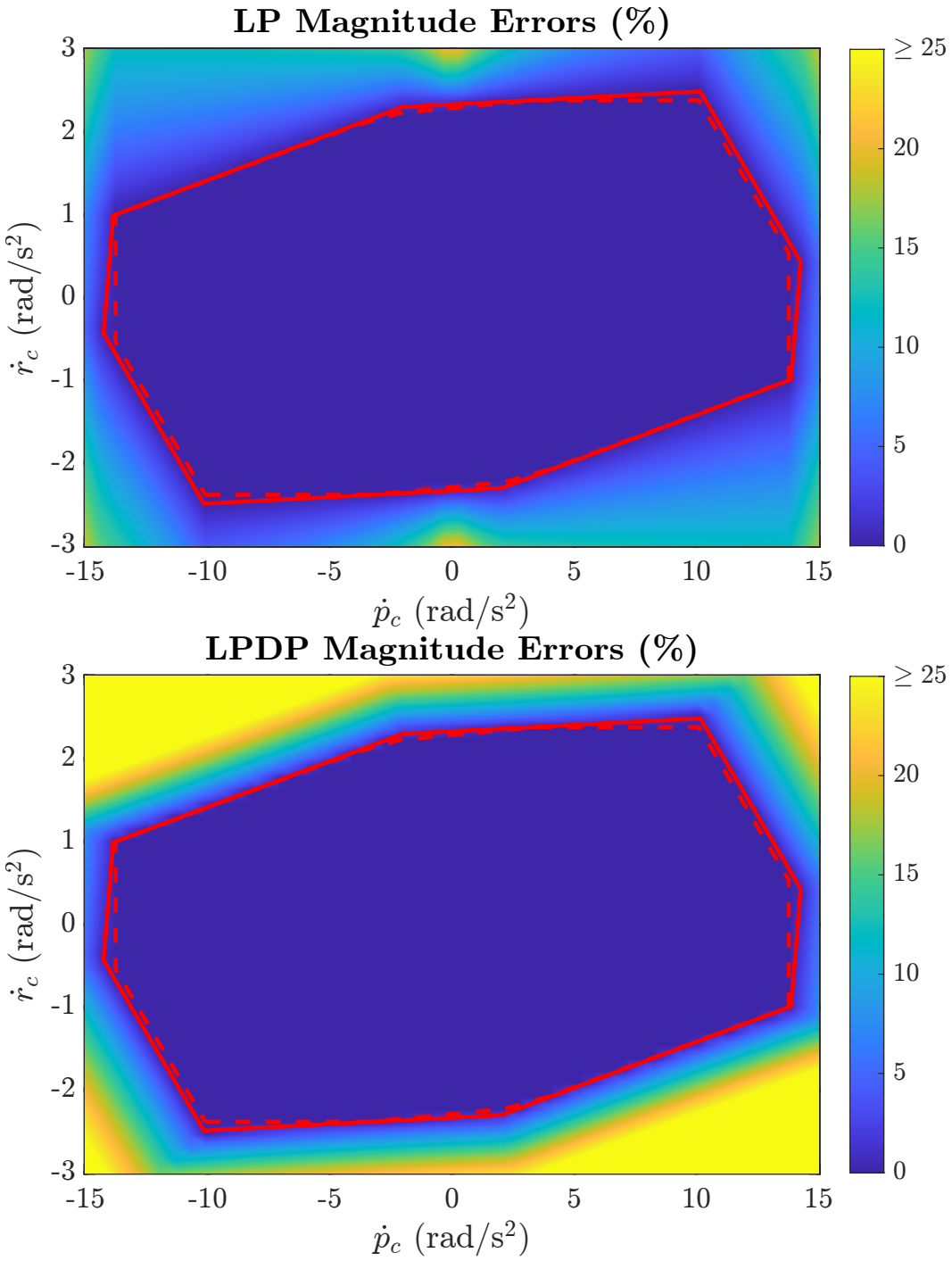



Figure 6.4: Relative magnitude errors for the LP and LPDP allocators. The TAMS of the F-16 AFTI aircraft is shown as a solid red line, and the approximate PAMS of each allocator as a dotted red line.

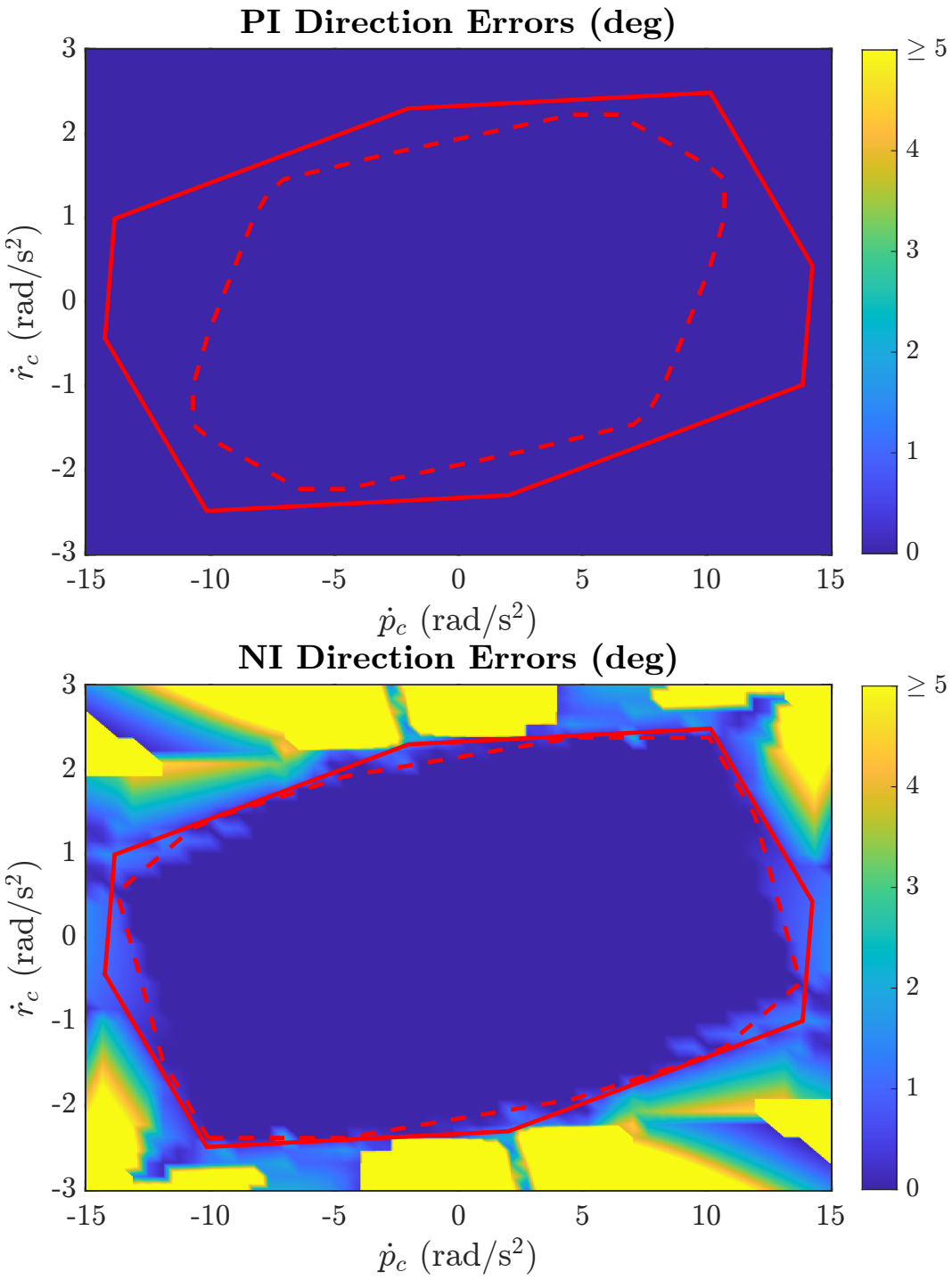



Figure 6.5: Direction errors for the PI and NI allocators. The TAMS of the F-16 AFTI aircraft is shown as a solid red line, and the approximate PAMS of each allocator as a dotted red line.

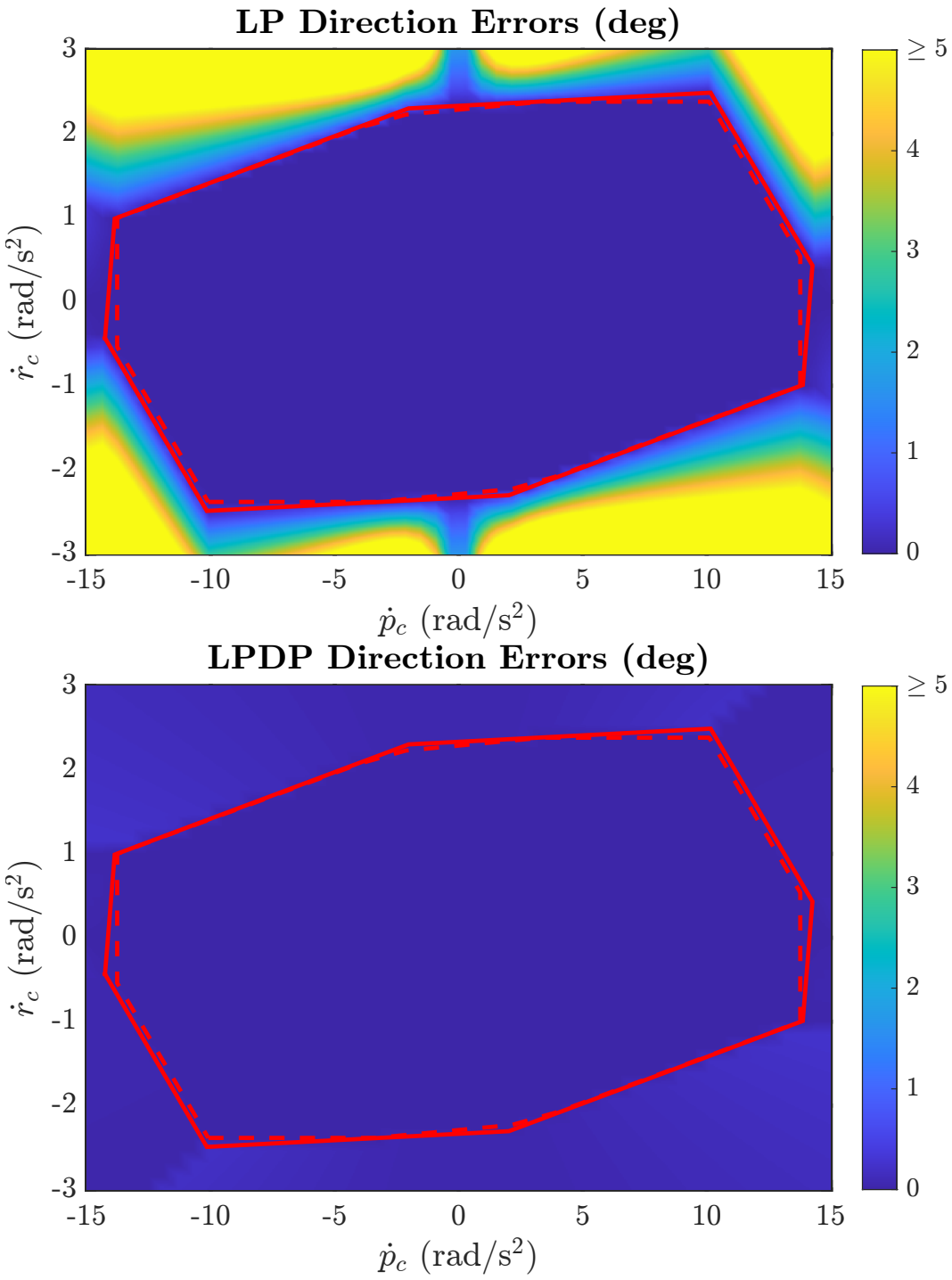



Figure 6.6: Direction errors for the LP and LPDP allocators. The TAMS of the F-16 AFTI aircraft is shown as a solid red line, and the approximate PAMS of each allocator as a dotted red line.

## 6.5 Solver Analysis

The proposed LPDP method eq. (6.8) is a convex optimization-based allocator in the form of a SOCP. Mature SOCP solvers are available, and can be shown to converge in a polynomial number of iterations [32], thus providing a promising indication that the LPDP allocator can be used for real-time application. In this section we confirm the feasiblity of this approach by evaluating the CPU time needed to generate allocation solutions for the F-16 AFTI model. We show that the solution times are consistently fast, and similarly efficient to the LP allocator eq. (2.7).

To solve the LPDP problem we use MATLAB's `coneprog` command, and to solve the LP problem we use `linprog` command [82]. Simulations were implemented in MATLAB R2023b and run on a 2023 MacBook Pro with M2 Pro chip and 16GB RAM.

Figure 6.7 shows the distribution CPU times for 2000 LPDP and LP allocation solutions for the grid of commands shown in Figs. 6.3 - 6.6. The first call to each solver is slower due to the initial parsing of the problem, and so are excluded from the plot. Every subsequent call takes less than $5 \times 10^{-3}$s. Consequently the LPDP allocator could be called at 200 Hz or slower, indicating sufficient speed for real-time application. Furthermore, the LPDP problem is only slightly slower to solve than the simpler LP problem on average and in outlier cases.

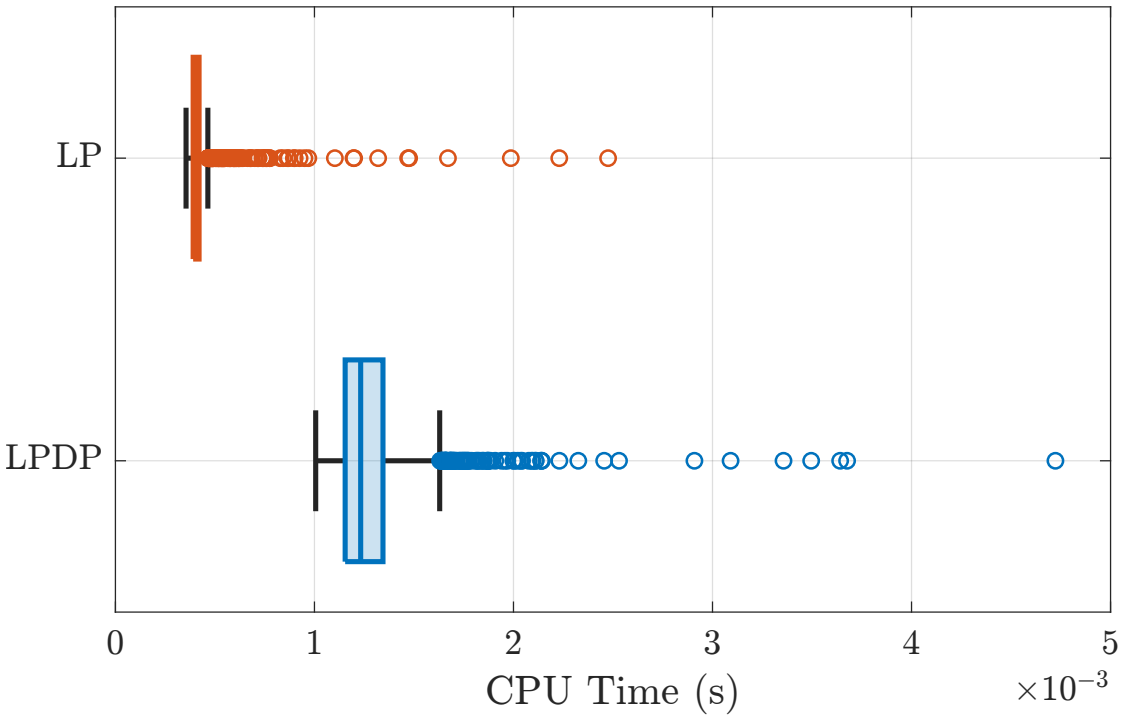



Figure 6.7: Boxplot of CPU time per call of the LP and LPDP allocators. Boxes show 1st and 3rd quartiles, whiskers cover the range 1.5 times the interquartile range above and below this range, and circles denote outliers.

## 6.6 Chapter Conclusions

Traditional generalized inverse control allocation can preserve resultant generalized moment direction, but it does not allow access to a large portion of attainable generalized moments, even if there is a set of admissible effector positions with satisfies the constraints. On the other hand, LP control allocation will allow access to a large portion of attainable generalized moments, but the resultant generalized moment vector may not be co-linear with the command.

This chapter derived the LPDP allocator to effectively balance the two priorities. The LPDP allocator takes the form of a SOCP and can be solved sufficiently fast for real-time implementation. Numerical experiments demonstrated that the LPDP allocator can access a large percentage of attainable generalized moments,

while yielding generalized moments which are nearly co-linear with the command, even when the command is far outside of the TAMS.

Future work will entail further studies of the proposed LPDP allocator's error properties in various problem contexts, eventually leading to studying piloted handling qualities with the LPDP allocator implemented on aircraft.

# Chapter 7

# Direction-Preserving Control Allocation by Robust Linear Programming

## 7.1 Introduction

As control laws in aerospace applications typically produce generalized moment commands and control allocators are used to map the generalized moment commands to individual effectors however the control allocation problem is typically solved without regard to parametric model uncertainty.

Previously, Grauer [11] derived the minimum-variance control allocator, however its generalized inverse formulation is not highly-efficient; also it minimizes the variance of the magnitude error in generalized moment. Chapter 5 showed how robust optimization theory could be used in control allocation problem. This chapter showed how the RLPe allocator could also show similar performance against uncertainty as the legacy MV allocator while having a higher allocation efficiency.

However, chapter 5 was limited to using robust optimization theory to preserve the magnitude error in generalized moment. Chapter 6 showed how the dot product rule could be modified to form a second-order cone constraint which results in the LPDP allocator, a highly efficient allocator with minimal direction error.

The contribution of this chapter is to derive the Robust Linear Programming Direction-Preserving allocator (RLPDP) which is a modified version of the LPDP allocator in Chapter 6 and uses robust optimization theory to preserve direction in the presence of parametric model uncertainty considering ellipsoidal uncertainty sets. The resulting RLPDP allocator is also formulated as a SOCP and shows a similar formulation as the RLPe allocator in Chapter 5. This chapter shows how the RLPDP allocator can preserve generalized moment direction in the case of an effector fault without knowledge of the fault.

The remainder of this chapter is organized as follows. Section 7.2 shows the derivation of the RLPDP allocator. Section 7.3 shows a simulation example of the RLPDP allocator. Lastly, section 7.4 provides concluding remarks.

## 7.2 Robust Linear Programming Direction-Preserving Control Allocation

The legacy linear programming allocator can access large sets of attainable moments, but in general will not result in a realized generalized moment which is co-linear with the command. To address this limitation the LPDP allocator was derived in chapter 6 which will access large portions of attainable moments while preserving generalized moment direction error.

This section derives the RLPDP allocator which is shown to be highly-efficient

while resulting in a realized moment which is co-linear with the generalized moment command. The RLPDP allocator is a form of the LPDP allocator where the second-order cone constraint is modified using robust control theory to account for parametric model uncertainties.

### 7.2.1 Problem Formulation

To consider the direction-preserving case in the presence of parametric model uncertainty, begin with the second-order cone direction preserving constraint, which was Eq. (6.7) in chapter 6.

$$\|\boldsymbol{\tau}_c\|_2\|\boldsymbol{B}\boldsymbol{u}\|_2 \leq \|\boldsymbol{\tau}_c\|_2^2\gamma + \boldsymbol{\tau}_c^T\boldsymbol{B}\boldsymbol{u} \tag{7.1}$$

When considering parametric model uncertainty Eq. (7.1) becomes,

$$\|\boldsymbol{\tau}_c\|_2\|\boldsymbol{B}(\boldsymbol{\theta})\boldsymbol{u}\|_2 \leq \|\boldsymbol{\tau}_c\|_2^2\gamma + \boldsymbol{\tau}_c^T\boldsymbol{B}(\boldsymbol{\theta})\boldsymbol{u} \tag{7.2}$$

Equation (7.1) was shown to sufficiently preserve generalized moment direction in chapter 6 however the effects of parametric model uncertainty were not considered. The remainder of this chapter is aimed at using robust optimization theory to preserve generalized moment direction in the presence of parametric model uncertainty. To begin, recall the parsed LP problem from chaper 2.

$$\min_{\boldsymbol{u},\boldsymbol{s},\boldsymbol{q}} \quad \boldsymbol{J} = \begin{bmatrix} 0\ldots0 & 1\ldots1 & \epsilon\ldots\epsilon \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{s} \\ \boldsymbol{q} \end{bmatrix} \tag{7.3a}$$

$$\text{s.t.} \quad -\boldsymbol{u} \leq -\underline{\boldsymbol{u}} \tag{7.3b}$$

$$\boldsymbol{u} \leq \overline{\boldsymbol{u}} \tag{7.3c}$$

$$\boldsymbol{B}(\hat{\boldsymbol{\theta}})_i \boldsymbol{u} - s_i \leq \tau_{c,i} \quad \forall i \tag{7.3d}$$

$$-\boldsymbol{B}(\hat{\boldsymbol{\theta}})_i \boldsymbol{u} - s_i \leq -\tau_{c,i} \quad \forall i \tag{7.3e}$$

$$-\boldsymbol{s} \leq \boldsymbol{0} \tag{7.3f}$$

$$u_i - q_i \leq u_{0,i} \quad \forall i \tag{7.3g}$$

$$-u_i - q_i \leq -u_{0,i} \quad \forall i \tag{7.3h}$$

Similarly to chapter 6 Eq. (7.3) can be augmented with the constraint from Eq. (7.2)

$$\min_{\boldsymbol{u},\boldsymbol{s},\boldsymbol{q},\gamma} \quad \boldsymbol{J} = \begin{bmatrix} 0 \ldots 0 & 1 \ldots 1 & \epsilon \ldots \epsilon & \xi\,(h + \|\boldsymbol{\tau}_c\|_1) \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{s} \\ \boldsymbol{q} \\ \gamma \end{bmatrix} \tag{7.4a}$$

$$\text{s.t.} \quad -\boldsymbol{u} \leq -\underline{\boldsymbol{u}} \tag{7.4b}$$

$$\boldsymbol{u} \leq \overline{\boldsymbol{u}} \tag{7.4c}$$

$$\boldsymbol{B}(\hat{\boldsymbol{\theta}})_i \boldsymbol{u} - s_i \leq \tau_{c,i} \quad \forall i \tag{7.4d}$$

$$-\boldsymbol{B}(\hat{\boldsymbol{\theta}})_i \boldsymbol{u} - s_i \leq -\tau_{c,i} \quad \forall i \tag{7.4e}$$

$$-\boldsymbol{s} \leq \boldsymbol{0} \tag{7.4f}$$

$$u_i - q_i \leq u_{0,i} \quad \forall i \tag{7.4g}$$

$$-u_i - q_i \leq -u_{0,i} \quad \forall i \tag{7.4h}$$

$$\|\boldsymbol{\tau}_c\|_2 \|\boldsymbol{B}(\boldsymbol{\theta})\boldsymbol{u}\|_2 \leq \|\boldsymbol{\tau}_c\|_2^2 \gamma + \boldsymbol{\tau}_c^T \boldsymbol{B}(\boldsymbol{\theta})\boldsymbol{u} \tag{7.4i}$$

The next step is to derive the *robust counterpart* of the constraint in Eq. (7.4i) this is performed by the methodology in Ref. [29]. The constraint should be modified

to maximize $\boldsymbol{B}(\boldsymbol{\theta})$.

$$\|\boldsymbol{\tau}_c\|_2 \max_{\boldsymbol{B}(\boldsymbol{\theta})_i \in \Theta_i} \|\boldsymbol{B}(\boldsymbol{\theta})\boldsymbol{u}\|_2 \leq \|\boldsymbol{\tau}_c\|_2^2 \gamma + \boldsymbol{\tau}_c^T \max_{\boldsymbol{B}(\boldsymbol{\theta})_i \in \Theta_i} \boldsymbol{B}(\boldsymbol{\theta})\boldsymbol{u} \tag{7.5}$$

Equation (7.5) involves two terms, both are maximization problems. Consider these two terms separately then plug them both into Eq. 7.5. Begin with the first maximization term by recalling the definition of the ellipsoidal uncertainty set in section 5.2.2 and plug into the term.

$$\max_{\boldsymbol{B}(\boldsymbol{\theta})_i \in \Theta_i} \|\boldsymbol{B}(\boldsymbol{\theta})\boldsymbol{u}\|_2 = \left\| \begin{bmatrix} \boldsymbol{B}(\hat{\boldsymbol{\theta}})_1 \\ \boldsymbol{B}(\hat{\boldsymbol{\theta}})_2 \\ \vdots \\ \boldsymbol{B}(\hat{\boldsymbol{\theta}})_i \end{bmatrix} \boldsymbol{u} + \begin{bmatrix} \boldsymbol{k}^T \boldsymbol{R}_1 \boldsymbol{u} \\ \boldsymbol{k}^T \boldsymbol{R}_2 \boldsymbol{u} \\ \vdots \\ \boldsymbol{k}^T \boldsymbol{R}_i \boldsymbol{u} \end{bmatrix} \right\|_2 \tag{7.6}$$

Using the identities presented in [26] Eq. (7.6) reduces to

$$\max_{\boldsymbol{B}(\boldsymbol{\theta})_i \in \Theta_i} \|\boldsymbol{B}(\boldsymbol{\theta})\boldsymbol{u}\|_2 = \left\| \begin{bmatrix} \boldsymbol{B}(\hat{\boldsymbol{\theta}})_1 \\ \boldsymbol{B}(\hat{\boldsymbol{\theta}})_2 \\ \vdots \\ \boldsymbol{B}(\hat{\boldsymbol{\theta}})_i \end{bmatrix} \boldsymbol{u} + \begin{bmatrix} \|\boldsymbol{R}_1 \boldsymbol{u}\|_2 \\ \|\boldsymbol{R}_2 \boldsymbol{u}\|_2 \\ \vdots \\ \|\boldsymbol{R}_i \boldsymbol{u}\|_2 \end{bmatrix} \right\|_2 \tag{7.7}$$

Although Eq. (7.7) is a solution to the maximization term in Eq. (7.5) it is a non-convex constraint. To convexify this constraint, linearize the $\|\boldsymbol{R}_i \boldsymbol{u}\|_2$ in Eq. (7.7) about the preferred control surface deflection, $\boldsymbol{u}_0$.

$$\|\boldsymbol{R}_i \boldsymbol{u}\|_2 \approx \left(\boldsymbol{u}_0^T \boldsymbol{R}_i^T \boldsymbol{R}_i \boldsymbol{u}_0\right)^{\frac{-1}{2}} \left(\boldsymbol{u}_0^T \boldsymbol{R}_i^T \boldsymbol{R}_i\right) \boldsymbol{u} \tag{7.8}$$

A convex expression in the form of a second-order cone constraint can be now

shown as,

$$\max_{\boldsymbol{B}(\boldsymbol{\theta})_i \in \Theta_i} \|\boldsymbol{B}(\boldsymbol{\theta})\boldsymbol{u}\|_2 = \left\| \begin{bmatrix} \boldsymbol{B}(\hat{\boldsymbol{\theta}})_1 + \left(\boldsymbol{u}_0^T \boldsymbol{R}_1^T \boldsymbol{R}_1 \boldsymbol{u}_0\right)^{\frac{-1}{2}} \left(\boldsymbol{u}_0^T \boldsymbol{R}_1^T \boldsymbol{R}_1\right) \\ \boldsymbol{B}(\hat{\boldsymbol{\theta}})_2 + \left(\boldsymbol{u}_0^T \boldsymbol{R}_2^T \boldsymbol{R}_2 \boldsymbol{u}_0\right)^{\frac{-1}{2}} \left(\boldsymbol{u}_0^T \boldsymbol{R}_2^T \boldsymbol{R}_2\right) \\ \vdots \\ \boldsymbol{B}(\hat{\boldsymbol{\theta}})_i + \left(\boldsymbol{u}_0^T \boldsymbol{R}_i^T \boldsymbol{R}_i \boldsymbol{u}_0\right)^{\frac{-1}{2}} \left(\boldsymbol{u}_0^T \boldsymbol{R}_i^T \boldsymbol{R}_i\right) \end{bmatrix} \boldsymbol{u} \right\|_2 \tag{7.9}$$

which is a second-order cone constraint! To prevent singularity at $\boldsymbol{u}_0 = \boldsymbol{0}$ a anti-deadzone function is introduced:

$$\|\boldsymbol{u}_0\|_1 \geq 1 \times 10^{-9} \tag{7.10}$$

Similar to the methods in chapter 5 and Ref. [26], the solution to the second term is given in Eq. (7.5) can be shown as,

$$\max_{\boldsymbol{B}(\boldsymbol{\theta})_i \in \Theta_i} \boldsymbol{B}(\boldsymbol{\theta})\boldsymbol{u} = \boldsymbol{B}(\hat{\boldsymbol{\theta}})_i \boldsymbol{u} + \|\boldsymbol{R_i u}\|_2 \quad \forall i \tag{7.11}$$

Inserting Eqs. (7.9) and (7.11) into the constraint given in eq. (7.5).

$$\begin{aligned} \|\boldsymbol{\tau}_c\|_2 \left\| \boldsymbol{B}(\hat{\boldsymbol{\theta}})_i \boldsymbol{u} + \left(\boldsymbol{u}_0^T \boldsymbol{R}_i^T \boldsymbol{R}_i \boldsymbol{u}_0\right)^{\frac{-1}{2}} \left(\boldsymbol{u}_0^T \boldsymbol{R}_i^T \boldsymbol{R}_i\right) \boldsymbol{u} \right\|_2 \\ \leq \|\boldsymbol{\tau}_c\|_2^2 \gamma + \tau_{c,i} \Big( \boldsymbol{B}(\hat{\boldsymbol{\theta}})_i \boldsymbol{u} + \|\boldsymbol{R_i u}\|_2 \Big) \quad \forall i \end{aligned} \tag{7.12}$$

To represent Eq. (7.12) as a second-order cone constraint the $\tau_{c,i}\|\boldsymbol{R}_i\boldsymbol{u}\|_2$ term is linearized about $\boldsymbol{u}_0$ using the same procedure as Eq. (7.8). Equation (7.12) can be

re-written into a standard form second-order cone constraint.

$$\begin{aligned} &\|\boldsymbol{\tau}_c\|_2 \left\| \boldsymbol{B}(\hat{\boldsymbol{\theta}})_i \boldsymbol{u} + \left(\boldsymbol{u}_0^T \boldsymbol{R}_i^T \boldsymbol{R}_i \boldsymbol{u}_0\right)^{\frac{-1}{2}} \left(\boldsymbol{u}_0^T \boldsymbol{R}_i^T \boldsymbol{R}_i\right) \boldsymbol{u} \right\|_2 \\ &\leq \|\boldsymbol{\tau}_c\|_2^2 \gamma + \tau_{c,i}^T \left( \boldsymbol{B}(\hat{\boldsymbol{\theta}})_i + \left(\boldsymbol{u}_0^T \boldsymbol{R}_i^T \boldsymbol{R}_i \boldsymbol{u}_0\right)^{\frac{-1}{2}} \left(\boldsymbol{u}_0^T \boldsymbol{R}_i^T \boldsymbol{R}_i\right) \right) \boldsymbol{u} \quad \forall i \end{aligned} \tag{7.13}$$

The RLPDP allocator is now formed by augmenting the legacy LP allocator in chapter 2 with the second-order cone constraint in eq. (7.13):

$$\min_{\boldsymbol{u},\boldsymbol{s},\boldsymbol{q},\gamma} \quad \boldsymbol{J} = \begin{bmatrix} 0 \ldots 0 & 1 \ldots 1 & \epsilon \ldots \epsilon & \xi \left(h + \|\boldsymbol{\tau}_c\|_1\right) \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{s} \\ \boldsymbol{q} \\ \gamma \end{bmatrix} \tag{7.14a}$$

$$\text{s.t.} \quad -\boldsymbol{u} \leq -\underline{\boldsymbol{u}} \tag{7.14b}$$

$$\boldsymbol{u} \leq \overline{\boldsymbol{u}} \tag{7.14c}$$

$$\boldsymbol{B}(\hat{\boldsymbol{\theta}})_i \boldsymbol{u} - s_i \leq \tau_{c,i} \quad \forall i \tag{7.14d}$$

$$-\boldsymbol{B}(\hat{\boldsymbol{\theta}})_i \boldsymbol{u} - s_i \leq -\tau_{c,i} \quad \forall i \tag{7.14e}$$

$$-\boldsymbol{s} \leq \boldsymbol{0} \tag{7.14f}$$

$$u_i - q_i \leq u_{0,i} \quad \forall i \tag{7.14g}$$

$$-u_i - q_i \leq -u_{0,i} \quad \forall i \tag{7.14h}$$

$$\begin{aligned} &\|\boldsymbol{\tau}_c\|_2 \left\| \boldsymbol{B}(\hat{\boldsymbol{\theta}})_i \boldsymbol{u} + \left(\boldsymbol{u}_0^T \boldsymbol{R}_i^T \boldsymbol{R}_i \boldsymbol{u}_0\right)^{\frac{-1}{2}} \left(\boldsymbol{u}_0^T \boldsymbol{R}_i^T \boldsymbol{R}_i\right) \boldsymbol{u} \right\|_2 \\ &\leq \|\boldsymbol{\tau}_c\|_2^2 \gamma + \tau_{c,i}^T \left( \boldsymbol{B}(\hat{\boldsymbol{\theta}})_i + \left(\boldsymbol{u}_0^T \boldsymbol{R}_i^T \boldsymbol{R}_i \boldsymbol{u}_0\right)^{\frac{-1}{2}} \left(\boldsymbol{u}_0^T \boldsymbol{R}_i^T \boldsymbol{R}_i\right) \right) \boldsymbol{u} \quad \forall i \end{aligned} \tag{7.14i}$$

Here $\xi > 0$ is a parameter used to weight direction priorities in the cost function. In particular, as $\xi$ is increased, more emphasis is placed on reducing direction error

compared to the LP allocator eq. (2.7), at the cost of somewhat greater moment error. $h > 0$ is a small constant which keeps $\gamma$ near zero when $\|\boldsymbol{\tau}_c\|_1$ is small (the constant multiplier $\|\boldsymbol{\tau}_c\|_1$ in the cost function matches scaling between terms, improving computational performance for different command magnitudes).

It is clear to see that eq. (7.14) is a SOCP and can be solved in real-time by conic optimization solvers with guaranteed convergence properties [30, 31, 32]. In section 6.5 it is demonstrated that the SOCP problem eq. (7.14) can be solved efficiently, with only a small increase in computational cost compared to the LP problem eq. (2.7).

## 7.3 Simulation Example

The example of this chapter uses the NASA F-18 High Angle-of Attack Research Vehicle (HARV) shown in Fig. 7.1. The efficacy of the RLPDP allocator is shown when a fault in the aircraft ailerons is considered and no knowledge of the fault is assumed.



Figure 7.1: NASA F-18 High Angle-of Attack Research Vehicle (HARV) in flight (credit: NASA Armstrong Flight Research Center).

This example focuses on two generalized moments $\boldsymbol{\tau} = \begin{bmatrix} \dot{p} & \dot{r} \end{bmatrix}^T$, where $\dot{p}$ is the body-axis roll acceleration and $\dot{r}$ is the body-axis yaw acceleration. The control deflection vector is

$$\boldsymbol{u} = \begin{bmatrix} u_{DT} & u_{AI} & u_{RU} & u_{RTV} & u_{YTV} \end{bmatrix}^T \tag{7.15}$$

where the nomenclature for each effector is defined in table 7.1.

The control surface deflection limits are listed in table 7.2, and the control effectiveness matrix, $\boldsymbol{B}$, is evaluated at Mach 0.2 and 10,000 ft altitude [2]:

Table 7.1: F-18 HARV aircraft control surface nomenclature [2]

| Symbol | Control Surface |
|---|---|
| $u_{DT}$ | Differential Tail |
| $u_{AI}$ | Ailerons |
| $u_{RU}$ | Rudder |
| $u_{RTV}$ | Roll Thrust Vectoring |
| $u_{YTV}$ | Yaw Thrust Vectoring |

$$\boldsymbol{B} = \begin{bmatrix} 0.006 & 0.006 & 0.004 & 0 & 0.090 \\ 1.879 & 1.328 & 0.029 & 0.675 & 0.217 \\ -0.109 & -0.096 & -0.084 & 0.007 & -2.974 \end{bmatrix} \tag{7.16}$$

In the control allocation design the first row of the $\boldsymbol{B}$ matrix is neglected. In this chapter, the parameters are set to $\epsilon = \|\boldsymbol{B}\|_1 10^{-6} = 1.75 \times 10^{-5}$, $\xi = 10^3$, and $h = 10^{-6}$. The direction error weight, $\xi$, was tuned to improve direction error while maintaining high-efficiency was observed.

Table 7.2: F-18 HARV aircraft control surface position limits [1]

| Control Surface | Position Limit (deg) |
|---|---|
| Differential Tail | $\pm 17.5$ |
| Ailerons | $\pm 17.5$ |
| Rudder | $\pm 17.5$ |
| Roll Thrust Vectoring | $\pm 17.5$ |
| Yaw Thrust Vectoring | $\pm 17.5$ |

Table 7.3: Standard errors used in example

| | $u_{DT}$ | $u_{AI}$ | $u_{RU}$ | $u_{RTV}$ | $u_{YTV}$ |
|---|---|---|---|---|---|
| $L$ | 5% | 5% | 20% | 5% | 20% |
| $N$ | 50% | 50% | 5% | 20% | 5% |

### 7.3.1 Attainable Moment Sets and Allocation Efficiency

To evaluate the performance of the MV, RLPe, LPDP, and RLPDP allocators, begin by defining the Theoretical Attainable Moment Set (TAMS) as the set of all possible achievable generalized moments due to all possible control effector deflections. Define the Particular Attainable Moment Set (PAMS) as the subset of the TAMS which can be realized by the specific allocator. This chapter compares the allocation efficiency of each allocator. The *allocation efficiency* is defined as the ratio of the volume of the PAMS to the TAMS. If the allocator can realize the entire space in the TAMS then the allocator is perfectly efficient. The efficiency of each allocator is shown in Fig. 7.2.

As expected, the MV allocator is the least efficient. The RLPe allocator is slightly more efficient due to the convex optimization nature of the allocator, however, the robust constraints will limit its efficiency. The LPDP allocator is highly-efficient, and RLPDP allocator is 53% efficient. Its efficiency is limited by the robust LPDP constraint.

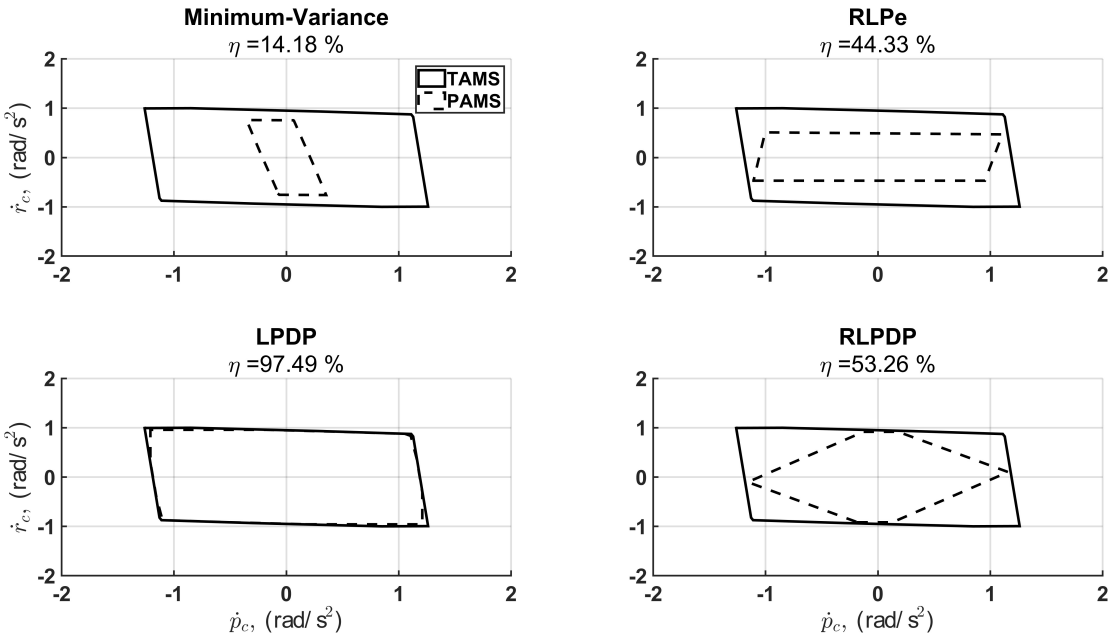



Figure 7.2: Efficiency (area of PAMS / area of TAMS) of each allocator applied to the F-18 HARV example problem.

### 7.3.2 Moment Direction Errors

The allocators are now evaluated by commanding a $\boldsymbol{\tau}_c = \begin{bmatrix} 0 & 0.5 \end{bmatrix}^T$ rad/s moment is commanded and then the allocation error is computed across 10,000 variations of the system model, where the variations are determined by the standard errors listed in Table 7.3. An aileron effector failure is simulated where aileron deflection, $u_{AI} = 0$ despite the command from the allocator. The allocator has no knowledge of the fault.

A probability density function of the generalized moment magnitude error and generalized moment direction error is plotted for both axes.

The performance of the MV, RLPe, LPDP, and RLPDP allocators based on the two criteria are shown in Figures 7.3, and 7.4 and also shown in Table 7.4. The RLPDP allocator shows both the smallest mean error in magnitude and direction when considering an aileron fault. The RLPDP allocator also shows the smallest

standard deviation in direction error when considering an effector fault. This small standard deviation in direction error can improve aircraft performance qualities without explicit knowledge of the aileron fault.

Table 7.4: Standard Deviations in generalized moment

| Allocator | Magnitude ($\times 10^{-3}$ rad/s) | Direction (deg) |
|---|---|---|
| MV | 11.4 | 1.2066 |
| RLPe | 9.4 | 1.0677 |
| LPDP | 13.8 | 0.7494 |
| RLPDP | 14.2 | 0.6602 |

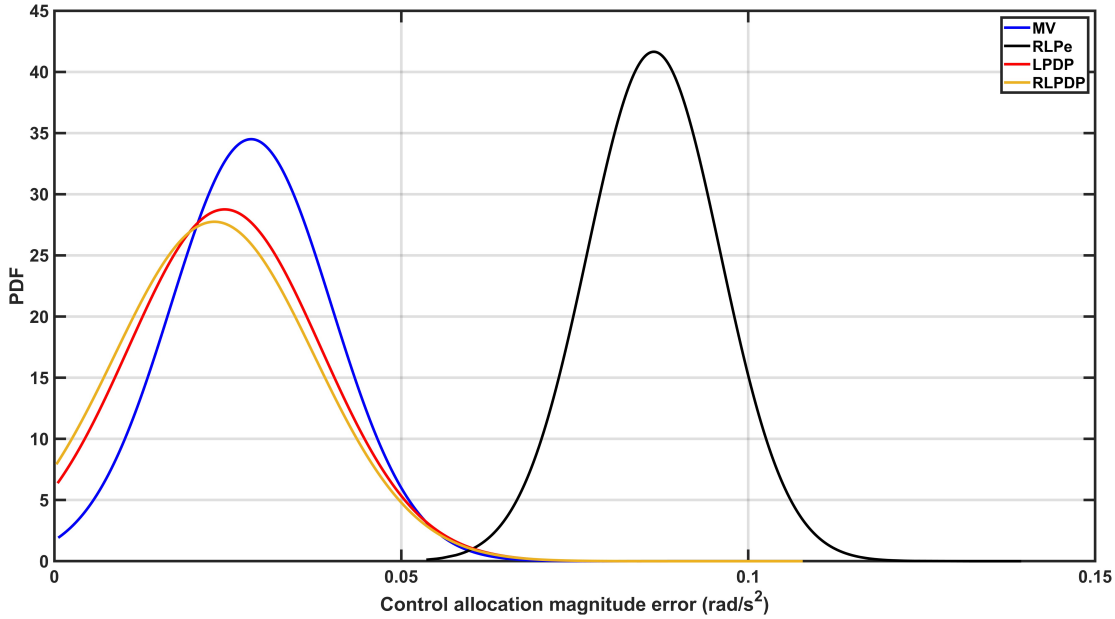



Figure 7.3: Magnitude Error PDF

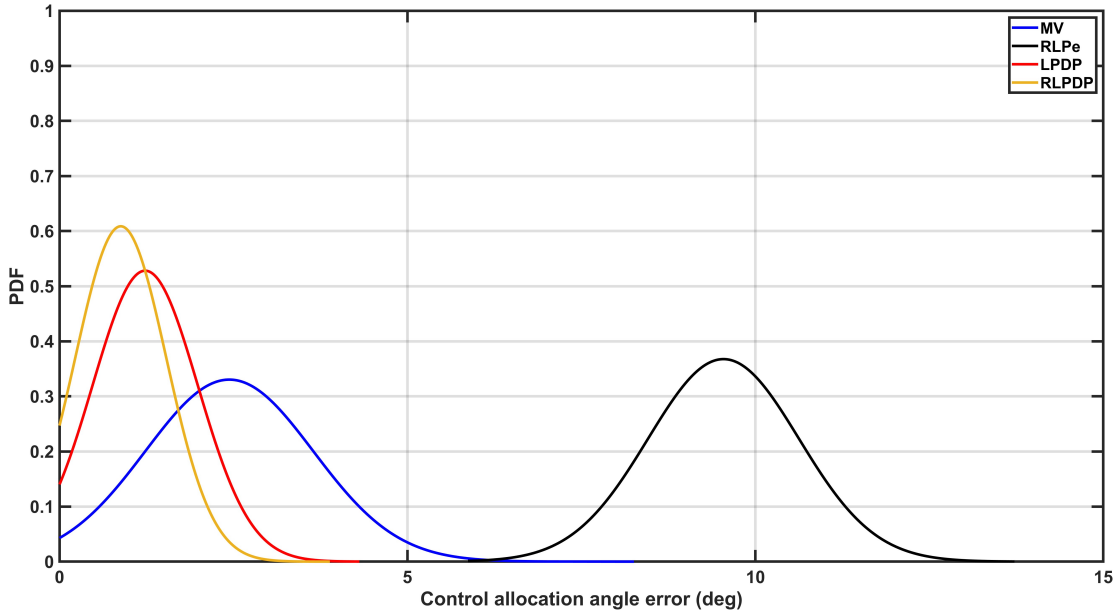



Figure 7.4: Direction Error PDF

## 7.4 Chapter Conclusions

In the previous chapter the LPDP allocator was derived to preserve generalized moment direction while simultaneously allowing access to a large portion of attainable generalized moments. This chapter focused on a extending the linear programming direction-preserving allocator to account for model uncertainties under ellipsoidal uncertainty sets.

It was found that the robust linear programming direction-preserving allocator shows the best preservation of generalized moment direction among the MV, RLPe, LPDP, and RLPDP allocators when considering a aileron fault. This preservation can help preserve piloted handling qualities and aircraft tracking performance. The next chapter will show the effects of the RLPDP allocator on a nonlinear aircraft simulation when an effector fault is considered.

# Chapter 8

# Nonlinear Aircraft Example

Previously this dissertation studied the effects of existing control allocation methods in the presence of parametric model uncertainty and derived the linear programming direction-preserving and robust linear programming direction-preserving control allocation methods. In this chapter the efficacy of this research is demonstrated on a nonlinear model of the NASA Langley T-2 subscale jet transport aircraft. More information about the T-2 and associated flight test operations can be found in Refs. [83, 84, 85, 86]. The T-2 aircraft was tested extensively in a wind tunnel and an extensive aerodyanamic database was built based on the wind tunnel test results [87, 88].

Three simulation cases are considered, a nominal simulation case with no uncertainty considered. A case with realistic uncertainty levels, and a case where a control effector has failed. The remainder of this chapter is organized as follows. Section 8.1 introduces the nonlinear T-2 aircraft model. Section 8.2 shows simulation results for the cases where no uncertainty is considered, cases with uncertainty, and cases with a failed control effector. Lastly, section 8.3 provides concluding remarks.



Figure 8.1: T-2 aircraft in flight (credit: NASA Langley Research Center).

## 8.1 T-2 Aircraft Model

The T-2 aircraft model was developed at NASA Langley Research Center and is a dynamically-scaled model of a twin-engine jet transport aircraft. The T-2 aircraft is depicted in Fig. 8.1. The nonlinear T-2 aircraft model was originally published in Ref. [89] These models were developed by force and moment data from wind tunnel testing and do not exist in closed-form. The coefficients of rolling, pitching, and yawing moment are denoted as $C_l$ and $C_m$ and $C_n$, respectively. The coefficients of force in the body-axis $x$, $y$, and $z$ directions are denoted as $C_x$, $C_y$, and $C_z$, respectively. The aircraft drag coefficient is denoted as $C_D$ and the aircraft lift coefficient is denoted as $C_L$. The T-2 has 16 independent control surfaces, left and right ailerons, left and right inboard and outboard elevators, upper and lower rudders, left and right inboard and outboard trailing edge flaps, left and right inboard and outboard spoilers. During flight only the elevators, rudders, and

Table 8.1: T-2 aircraft parametric model uncertainty

| Coefficient | Elevator | Aileron | Rudder |
|---|---|---|---|
| $C_l$ | 10% | 5% | 20% |
| $C_m$ | 5% | 50% | 50% |
| $C_n$ | 50% | 50% | 5% |

Table 8.2: T-2 control surface deflection limits

| Effector | Deflection Limit (deg) |
|---|---|
| Elevator | $[-30, 20]$ |
| Aileron | $\pm 20$ |
| Rudder | $\pm 30$ |

ailerons are deflected.

The non-dimensional aerodynamic coefficients vary nonlinearly with relative airflow quantities, angle-of-attack, $\alpha$, and sideslip angle, $\beta$. The non-dimensional aerodynamic moment coefficients also vary with non-dimensional body-axis angular rates ($p$, $q$, and $r$) and control surface deflections.

To generate body-axis force coefficients, the lift and drag coefficients are transformed.

$$\begin{bmatrix} C_x \\ C_z \end{bmatrix} = \begin{bmatrix} \sin\alpha & -\cos\alpha \\ -\cos\alpha & -\sin\alpha \end{bmatrix} \begin{bmatrix} C_L \\ C_D \end{bmatrix} \tag{8.1}$$

The aircraft body-axis force and moment coefficients were implemented in lookup table form to obtain the non-dimensional control derivatives a batch linearization routine was done where the `linearize` command in MATLAB was run across the entire flight envelope.

The data table for parametric model uncertainty is taken from Ref. [11] in Table 8.1. The control effector limits are given in Table 8.2.

The aircraft mass properties and geometry properties used in this simulation are listed in Table 8.3 and taken from Ref. [91].

Table 8.3: T-2 Mass and Geometry Properties

| Parameter | Value |
|---|---|
| Mass | 1.5850 (slug) |
| $I_{xx}$ | 1.1790 (slug $-$ ft$^2$) |
| $I_{yy}$ | 4.5200 (slug $-$ ft$^2$) |
| $I_{zz}$ | 5.5270 (slug $-$ ft$^2$) |
| $I_{xz}$ | 0.2110 (slug $-$ ft$^2$) |
| $b$ | 6.8490 (ft) |
| $\bar{c}$ | 0.9150 (ft) |
| $S$ | 5.9020 (ft$^2$) |

## 8.2 Aircraft Simulations

This section demonstrates the efficacy of the research in this dissertation on a nonlinear T-2 aircraft simulation. The maneuver considered is a 45 deg right turn followed by a 90 deg left turn, followed by another 45 deg right turn, while maintaining a constant altitude. Both an aggressive and a nominal turn maneuver is simulated. The aggressive turn maneuver is designed such that the aircraft turn radius generates $1g$ of lateral acceleration in an inertial reference frame. The nominal turn maneuver is generated to maintain $0.2g$ of lateral acceleration in an inertial reference frame. The aggressive maneuver is shown in Fig 8.2. The nominal maneuver is shown in Fig. 8.3.

To simulate turn maneuver a closed-loop simulation was built where PID controllers running at 50Hz were used to produce attitude and thrust commands to follow the path and maintain a constant airspeed of 135ft/s and 500ft altitude. A block diagram of the closed-loop simulation is shown in Fig. 8.4.

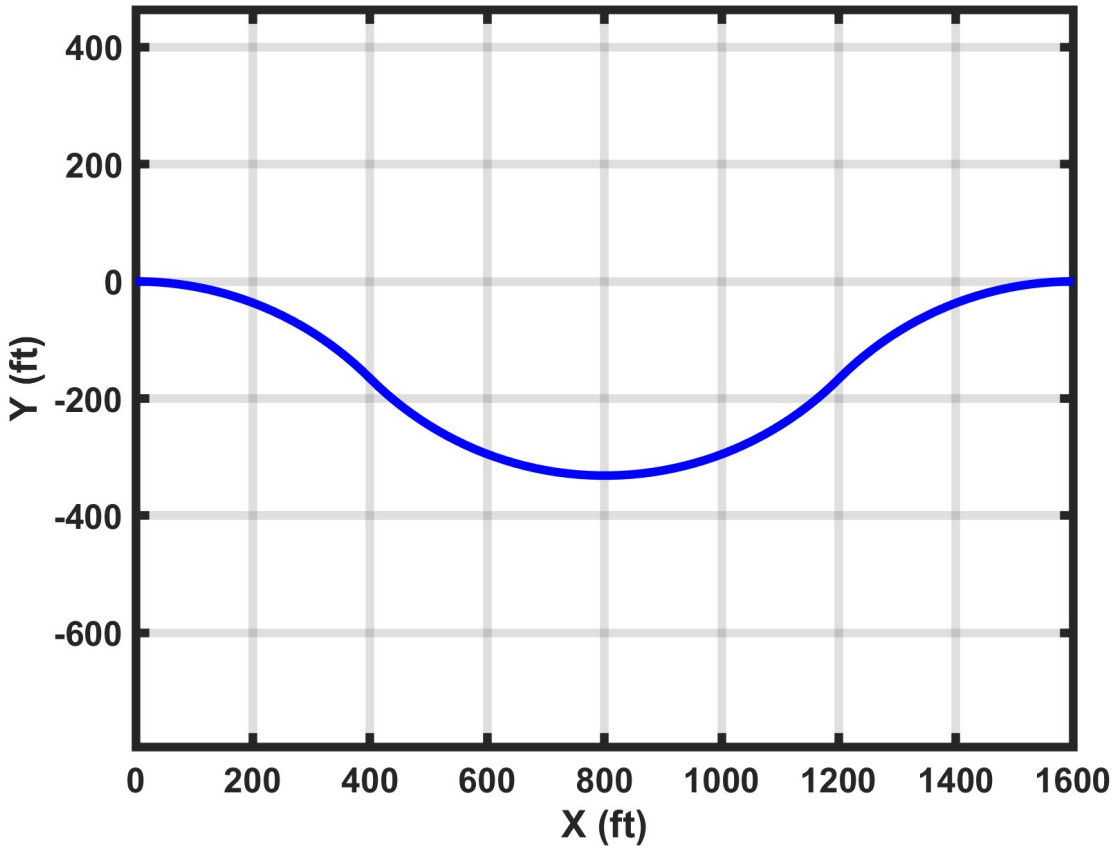



Figure 8.2: Aggressive maneuver.

### 8.2.1 Nominal Simulation Case

The first case considered is a nominal simulation case where no uncertainty and no faults were considered. Because no uncertainty was considered the PI, LP, and LPDP allocators are considered. None of these allocators were designed to incorporate the effects of parametric model uncertainty. Figure 8.5 shows the performance of the allocators in this case. In this case all three allocators track similarly, this is because there is no saturation thus no performance benefit for one allocator over another. The results for the aggressive turn case are shown in Fig. 8.6 which was designed to saturate control effectors. Tracking performance is degraded due to saturation, In this case the LPDP allocation showed better

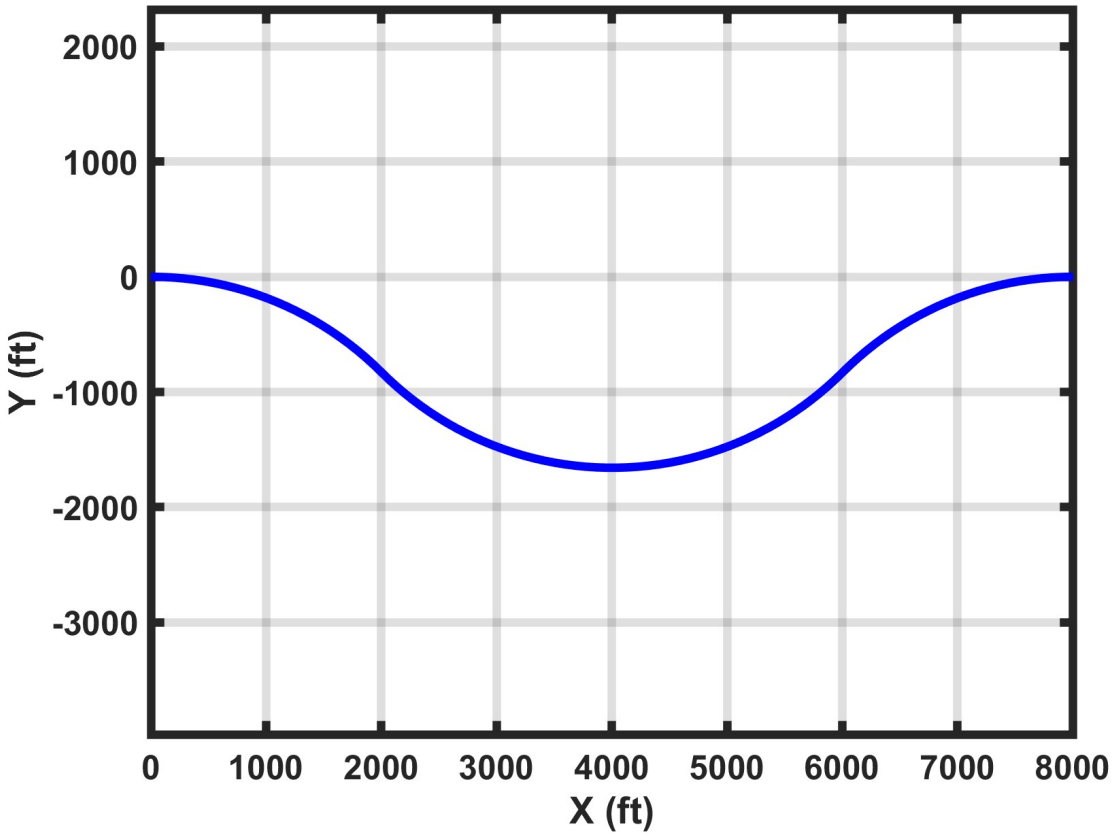



Figure 8.3: Nominal maneuver.

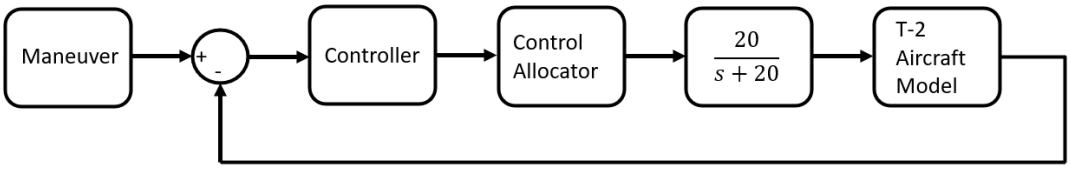



Figure 8.4: High-level block diagram of closed-loop T-2 simulation.

tracking due to its direction-preserving properties over the LP allocation.

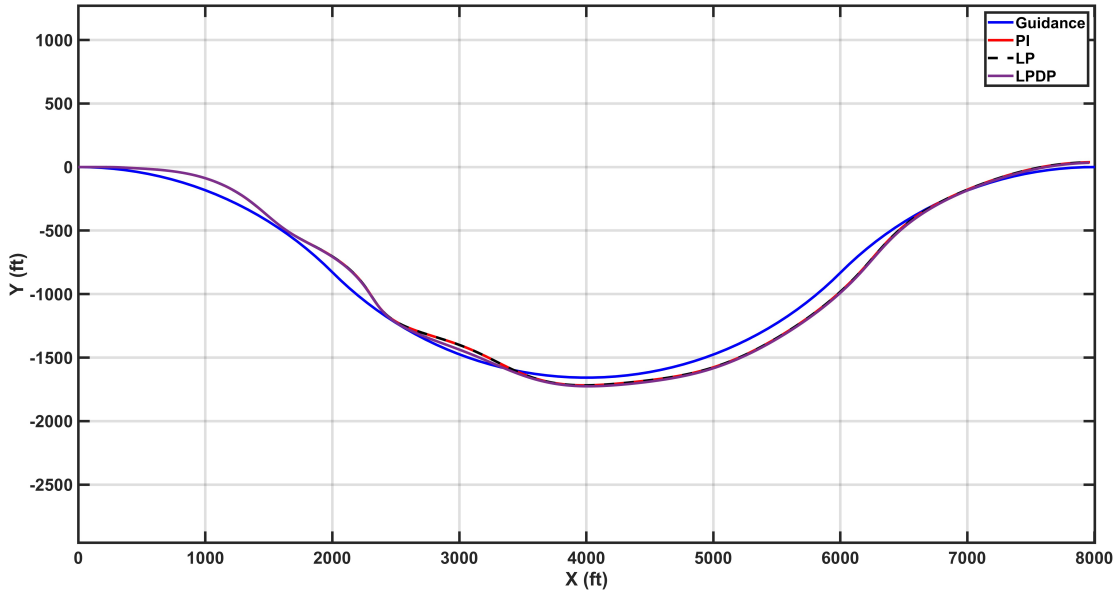



Figure 8.5: Nominal simulation performance.

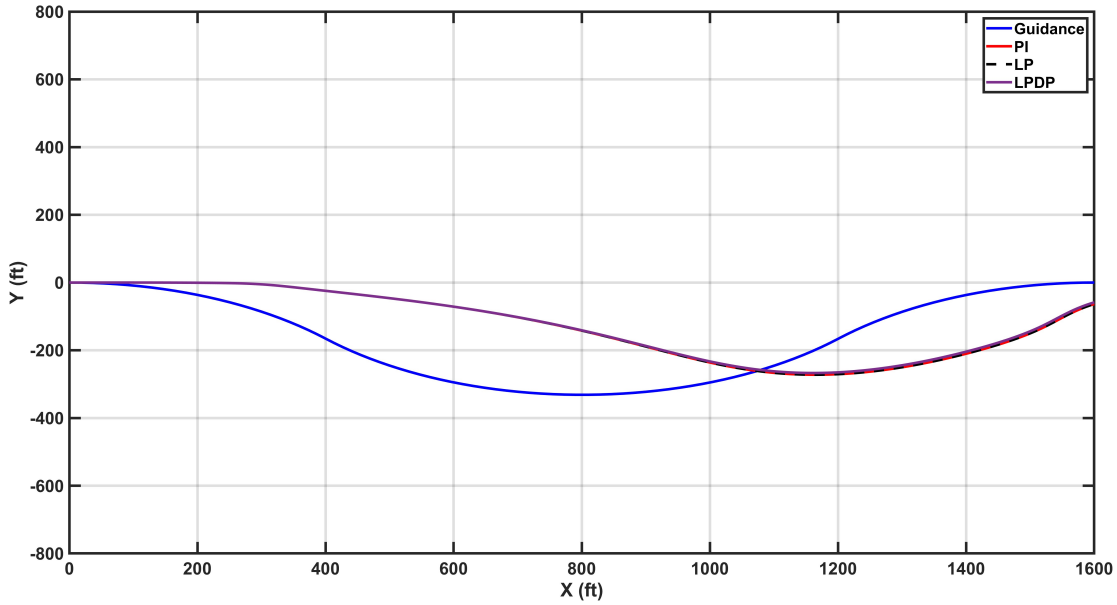



Figure 8.6: Agressive maneuver performance.

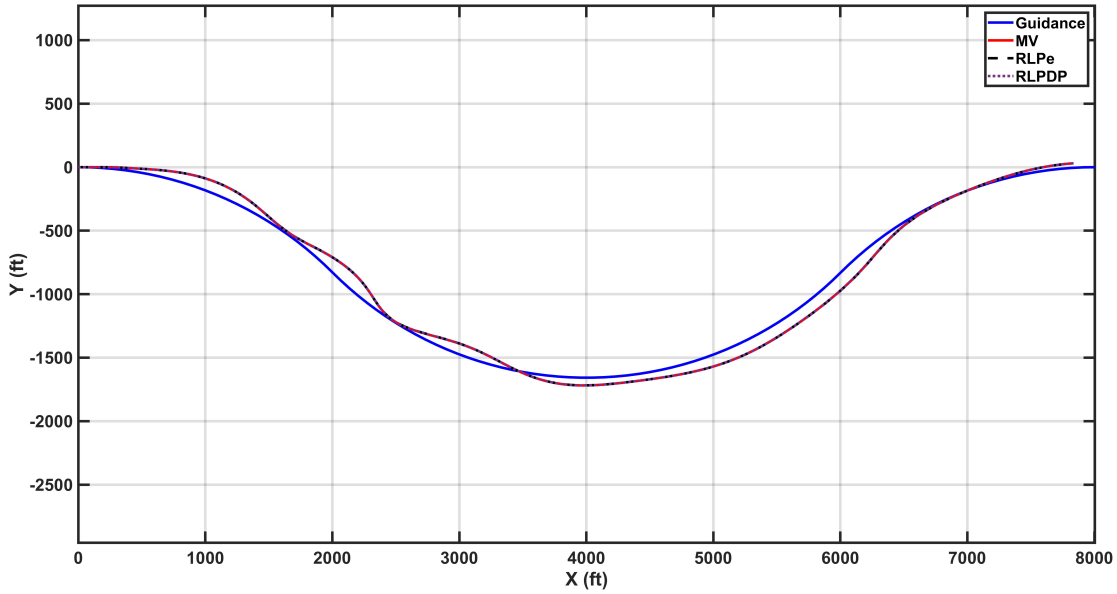



Figure 8.7: Nominal maneuver performance with uncertainty.

### 8.2.2 Model Uncertainty Case

In this case, realistic parametric model uncertainty was considered both the nominal and aggressive turn trajectories were simulated and the performance of the MV, RLPe and RLPDP allocation methods were compared. These three allocation methods were all designed to incorporate the effects of parametric model uncertainty and show how it can be mitigated in a closed-loop nonlinear simulation. The results for the nominal case are shown in Fig. 8.7 Figure 8.8 shows that the RLPe allocator has the best tracking performance in the presence of parametric model uncertainty while flying the aggressive trajectory. The RLPe allocator has the highest allocation efficiency over the MV, and RLPDP allocators and thus has the best tracking performance in the presence of parametric model uncertainty.

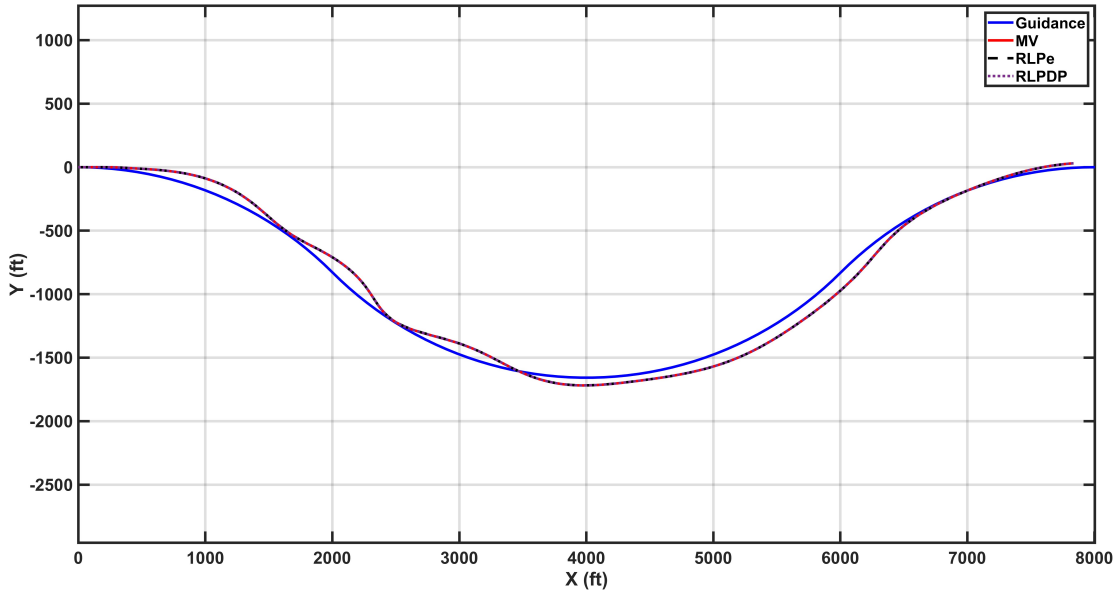



Figure 8.8: Nominal maneuver performance with uncertainty.

### 8.2.3 Effector Failure Case

In this case while flying the both the nominal and aggressive maneuvers, a failure is simulated in the aircraft ailerons where $\delta_{aL} = 0$. No knowledge of the failure is assumed. The performance of the MV, RLPe, and RLPDP allocators are considered and simulated. During the nominal trajectory shown in Fig. 8.10 it was found that all the allocators diverge from their commanded trajectory this is expected because of the aileron failure. However the RLPDP allocation method had the straightest trajectory, which highlights the direction-preserving method.

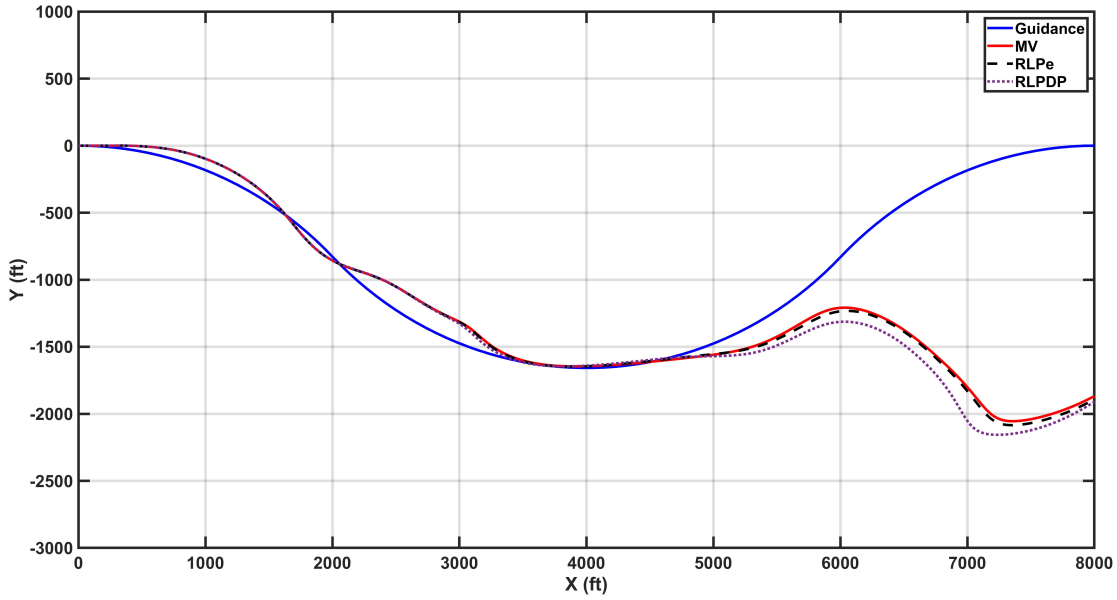



Figure 8.9: Nominal maneuver performance with aileron failure.

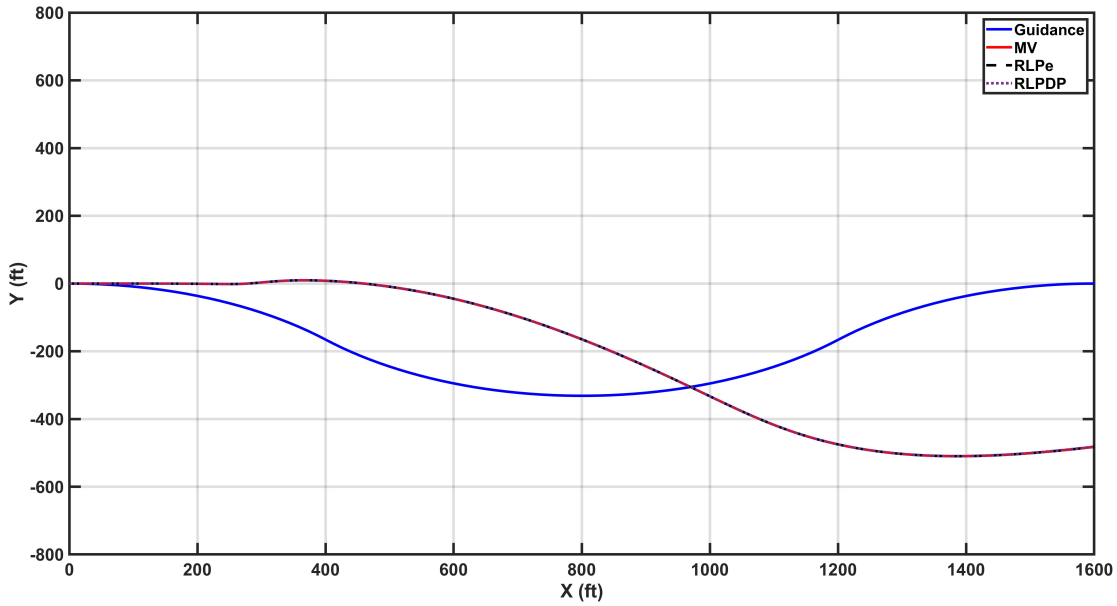



Figure 8.10: Aggressive maneuver performance with aileron failure.

## 8.3 Chapter Conclusions

This dissertation derived LPDP and RLPDP allocators. This chapter compared the efficacy of the control allocation methods considered in this dissertation on a closed-loop example of a T-2 aircraft considering the effects of parametric model uncertainty. The T-2 aircraft model was lookup table-based and controlled with several PID control loops. It was found that the effects of parametric model uncertainty were able to be effectively mitigated by the MV, RLPe, and RLPDP allocation methods. An allocation fault was also simulated, this showed that the RLPDP allocation method resulted in the straightest trajectory divergence from its commanded path.

# Chapter 9

# Conclusions

As flight vehicles advance, control algorithms must exist to effectively mitigate high levels of uncertainty and maintain sufficient flight performance requirements. This dissertation showed how control laws could be based on fully-known rigid-body state equations and the control allocation could be based on the parameters which are learned in real-time.

This dissertation also helped advance the study of the control allocation problem considering parametric model uncertainty and helped derive new methods to mitigate such uncertainty. Specifically, this dissertation considered the robust linear programming allocator derived using ellipsoidal uncertainty sets and box uncertainty sets. The RLPe allocator is shown to effectively incorporate model uncertainty into the control allocation problem, while maintaining high control allocation efficiency. One drawback is that control allocation as a convex-optimization problem is not inherently direction-preserving.

To develop direction-preserving control allocation, this dissertation derived the linear programming direction-preserving allocator which is an alternative method to access a large portion of attainable moments while the resulting moment is co-

linear with the commanded moment. This dissertation then extended the second-order cone constraint using robust control theory.

Lastly it was shown that these control allocation methods in the form of a second-order cone programming problem can be solved in real-time at 200Hz or slower which is suitable for many real-time aerospace applications.

## 9.1 Future Research

The control allocation methods in this dissertation were based on a local linearization of the aircraft control effectiveness. Several avenues exist to improve upon the existing suite of control methods. As more complex and harder-to-model vehicles are considered, control allocation based on local linearizations of control effectiveness could start to fail. Extensions to control allocation subject to nonlinear constraints, and nonlinear dynamics could be considered. This could take the form of a geometric programming problem.

Secondly, this dissertation considers convex-optimization based control allocation methods, commonly expressed as a second-order cone programming problem. These second-order cone programming-based control allocators do not exist in closed-form. When designing flight control systems, evaluation of stability is paramount. This means that formally showing stability is not simple, but critical. Further research could be done on formal stability methods when non closed-form control allocation methods are considered.

Convex optimization-based control allocation expressed as a second-order cone programming problem also requires a solution algorithm which must reliably converge in sufficient time. This dissertation utilized interior-point methods, however, more would could be done to aid in the understanding of solvers for second-order

cone programs.

Lastly, this dissertation does not include control effector rate limits into the control allocation design. There are terms to help select a control effector deflection close to its current deflection, this can help mitigate effects due to effector rate limits, however these terms may fail when there are significant differences between control effector bandwidths. Future research in model-predictive control allocation could consider robust control allocation specifically accounting for limited control allocator bandwidths.

## 9.2 List of Publications

Research included in this dissertation has been published in Ref. [92] and Ref. [93].

# Bibliography


[1] Barfield, A., *Multivariable Control Laws for the AFTI/F-16*, Master's thesis, Air Force Institute of Technology, 1983.

[2] Siwakosit, W. and Hess, R., "Multi-Input/Multi-Output Reconfigurable Flight Control Design," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 6, 2001, pp. 1079–1088, doi:10.2514/2.4841.

[3] Shaw, R. L., *Fighter Combat Tactics and Maneuvering*, United States Naval Institute, 1985.

[4] Åström, K. J. and Wittenmark, B., "On Self Tuning Regulators," *Automatica*, Vol. 9, No. 2, 1973, pp. 185–199, doi:10.1016/0005-1098(73)90073-3.

[5] Kalman, R. E., "Design of a Self-Optimizing Control System," *Transactions of the American Society of Mechanical Engineers*, Vol. 80, No. 2, 1958, pp. 468–477, doi:10.1115/1.4012407.

[6] Bellman, R. E., *Adaptive Control Processes - A Guided Tour*, Princeton, NJ: Princeton University Press, 1961.

[7] Y.D., L., *Adaptive Control: The Model Reference Approach*, New York, NY: Marcel Dekker, Inc., 1979.

[8] Monopoli, R. V., "Model Reference Adaptive Control with an Augmented Error Signal," *IEEE Transactions on Automatic Control*, Vol. 19, No. 4, Oct. 1974, pp. 474–484, doi:10.1109/TAC.1974.1100670.

[9] Morelli, E. A. and Klein, V., *Aircraft System Identification: Theory and Practice*, Sunflyte Enterprises, Williamsburg, VA, 2nd ed., 2016.

[10] Åström, K. J., *Introduction to Stochastic Control Theory*, Courier Corporation, 2012.

[11] Grauer, J. A. and Pei, J., "Minimum-Variance Control Allocation Considering Parametric Model Uncertainty," *AIAA SCITECH 2022 Forum*, AIAA-2022-0749, Jan. 2022, doi:10.2514/6.2022-0749.

[12] Grimble, M. J., "Controller Performance Benchmarking and Tuning Using Generalised Minimum Variance Control," *Automatica*, Vol. 38, No. 12, 2002, pp. 2111–2119, doi:10.1016/S0005-1098(02)00141-3.

[13] Nisio, M., *Stochastic Control Theory*, Vol. 9, Springer, 2015.

[14] Li, Z. and Evans, R. J., "Minimum-Variance Control of Linear Time-Varying Systems," *Automatica*, Vol. 33, No. 8, 1997, pp. 1531–1537, doi:10.1016/S0005-1098(97)00066-6.

[15] Durham, W. C., "Constrained Control Allocation," *Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 4, 1993, pp. 717–725, doi:10.2514/3.21072.

[16] Teel, A. and Buffington, J., "Anti-Windup for an F-16's Daisy Chain Control Allocator," *Guidance, Navigation, and Control Conference*, AIAA-97-3606, 1997, doi:10.2514/6.1997-3606.

[17] Petersen, J. A. and Bodson, M., "Constrained Quadratic Programming Techniques for Control Allocation," *IEEE Transactions on Control Systems Technology*, Vol. 14, No. 1, 2005, pp. 91–98, doi:10.1109/TCST.2005.860516.

[18] Tjønnås, J. and Johansen, T. A., "Adaptive Control Allocation," *Automatica*, Vol. 44, 2008, pp. 2754–2765, doi:10.1016/j.automatica.2008.03.031.

[19] Tohidi, S. S., Yildiz, Y., and Kolmanovsky, I., "Adaptive Control Allocation for Constrained Systems," *Automatica*, Vol. 121, No. 109161, 2020, doi:10.1016/j.automatica.2020.109161.

[20] Falconi, G. and Holzapfel, F., "Adaptive Fault Tolerant Control Allocation for a Hexacopter System," *American Control Conference*, June 2016, doi:10.1109/ACC.2016.7526736.

[21] Davidson, J. B., Lallman, F. J., and Bundick, W. T., "Real-Time Adaptive Control Allocation Applied to a High Performance Aircraft," *5th SIAM Conf. Control and Its Applications*, 2001.

[22] Schwager, M., Annaswamy, A., and E, L., "Adaptation-Based Reconfiguration in the Presence of Actuator Failures and Saturation," *American Control Conference*, July 2016, doi:10.1109/ACC.2005.1470366.

[23] Snyder, S., *L1 Adaptive Control Within Learn-To-Fly*, Ph.D. thesis, University of Illinois Urbana-Champaign, 2019.

[24] Buffington, J. M., "Modular Control Law Design for the Innovative Control Effectors (ICE) Tailless Fighter Aircraft Configuration 101-3," *AFRL-VA-WP-TR-1999-3057*, Air Force Research Labratory, Wright-Patterson AFB OH 45433-7542, June 1999, doi:10.21236/ADA374954.

[25] Petersen, J. A. and Bodson, M., "Interior-Point Algorithms for Control Allocation," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 3, 2005, pp. 471–480, doi:10.2514/1.5937.

[26] Duan, C., Zhang, S., Zhao, Y., and Kong, X., "Robust Control Allocation among Overactuated Spacecraft Thrusters under Ellipsoidal Uncertainty," *Abstract and Applied Analysis*, Vol. 2014, June 2014, pp. 1–12, doi:10.1155/2014/950127.

[27] Brumbaugh, R. W., "Aircraft Model for the AIAA Controls Design Challenge," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 4, 1994, pp. 747–752, doi:10.2514/3.21263.

[28] Tibshirani, R., "Convex Optimizaiton," *Carnegie Mellon University*, Spring 2015.

[29] Ghaoui, L. E., "Optimization Models and Applications," Spring 2021.

[30] Vandenberghe, L. and Boyd, S., "Semidefinite Programming," *SIAM Review*, Vol. 38, No. 1, 1996, pp. 49–95, doi:10.1137/1038003.

[31] Sturm, J. F., "Implementation of Interior Point Methods for Mixed Semidefinite and Second-Order Cone Optimization Problems," *Optimization Methods and Software*, Vol. 17, No. 6, 2002, pp. 1105–1154, doi:10.1080/1055678021000045123.

[32] Alizadeh, F. and Goldfarb, D., "Second-Order Cone Programming," *Mathematical Programming*, Vol. 95, No. 1, 2003, pp. 3–51, doi:10.1007/s10107-002-0339-5.

[33] Boyd, S. and Vandenberghe, L., *Convex Optimization*, Cambridge University Press, 2004.

[34] Morelli, E. A., "Practical Aspects of Real-Time Modeling for the Learn-To-Fly Concept," *2018 Atmospheric Flight Mechanics Conference*, AIAA Paper 2018-3309, June 2018, doi:10.2514/6.2018-3309.

[35] Heim, E. H., Viken, E., Brandon, J. M., and Croom, M. A., "NASA's Learn-to-Fly Project Overview," *2018 Atmospheric Flight Mechanics Conference*, AIAA-2018-3307, June 2018, doi:10.2514/6.2018-3307.

[36] Riddick, S. E., Busan, R. C., Cox, D. E., and Laughter, S. A., "Learn-to-Fly Test Setup and Concept of Operations," *2018 Atmospheric Flight Mechanics Conference*, AIAA-2018-3308, June 2018, doi:10.2514/6.2018-3308.

[37] Johansen, T. A., "Optimizing Nonlinear Control Allocation," *2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601)*, Vol. 4, IEEE, 2004, pp. 3435–3440, doi:10.1109/CDC.2004.1429240.

[38] Chumalee, S. and Whidborne, J. F., "Gain-Scheduled $H_\infty$ Control via Parameter-Dependent Lyapunov Functions," *International Journal of Systems Science*, Vol. 46, No. 1, 2015, pp. 125–138, doi:10.1080/00207721.2013.775386.

[39] Rangajeeva, S. L. and Whidborne, J. F., "Linear Parameter-Varying Control of a Quadrotor," *2011 6th International Conference on Industrial and Information Systems*, IEEE, 2011, pp. 483–488, doi:10.1109/ICIINFS.2011.6038118.

[40] Apkarian, P., Gahinet, P., and Becker, G., "Self-Scheduled $H_\infty$ Control of Linear Parameter-Varying Systems: a Design Example," *Automatica*, Vol. 31, No. 9, 1995, pp. 1251–1261, doi:10.1016/0005-1098(95)00038-X.

[41] Snyder, S., Bacon, B., Morelli, E., Frost, S., and Teubert, "Online Control Design for Learn-to-Fly," *AIAA Aviation Forum*, AIAA Paper 2018-3311, June 2018, doi:10.2514/6.2018-3311.

[42] Petres, Z., *Polytopic Decomposition of Linear Parameter-Varying Models by Tensor-Product Model Transformation*, Ph.D. thesis, Budapest University of Technology and Economics, May 2006.

[43] Nagy, S., Petres, Z., and Baranyi, P., "TP Tool - A Matlab Toolbox for TP Model Transformation," *Proc. of 8th International Symposium of Hungarian Researchers on Computational Intelligence and Informatics*, Budapest, Hungary, 2007, pp. 483–495.

[44] Foster, J. V., "Autonomous Guidance Algorithms for NASA Learn-to-Fly Technology Development," *2018 Atmospheric Flight Mechanics Conference*, AIAA-2018-3310, June 2018, doi:10.2514/6.2018-3310.

[45] Durham, W. C., "Attainable Moments for the Constrained Control Allocation Problem," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 6, 1994, pp. 1371–1373, doi:10.2514/3.21360.

[46] Enns, D., "Control Allocation Approaches," *Guidance, Navigation, and Control Conference and Exhibit*, AIAA Paper 98-4109, Aug. 1998, doi:10.2514/6.1998-4109.

[47] Burken, J. J., Lu, P., Wu, Z., and Bahm, C., "Two Reconfigurable Flight-Control Design Methods: Robust Servomechanism and Control Allocation," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 3, 2001, pp. 482–493, doi:10.2514/2.4769.

[48] Harkegard, O., "Efficient Active Set Algorithms for Solving Constrained Least Squares Problems in Aircraft Control Allocation," *Proceedings of the 41st IEEE Conference on Decision and Control*, 2002, pp. 1295–1300, doi:10.1109/CDC.2002.1184694.

[49] Bodson, M., "Evaluation of Optimization Methods for Control Allocation," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 4, 2002, pp. 703–711, doi:10.2514/2.4937.

[50] Luenberger, D. G. and Ye, Y., *Linear and Nonlinear Programming*, Vol. 2, Springer, 1984.

[51] Ivler, C. and Juhasz, O., "Evaluation of Control Allocation Techniques for a Medium Lift Tilt-Rotor," *American Helicopter Society*, Vol. 71, 2015, pp. 5–7.

[52] Härkegård, O., "Dynamic Control Allocation Using Constrained Quadratic Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 4, 2002, pp. 703–711, doi:10.2514/1.11607.

[53] Page, A. and Steinberg, M., "A Closed-Loop Comparison of Control Allocation Methods," *Guidance, Navigation, and Control Conference and Exhibit*, AIAA Paper 2000-4538, Aug. 2000, doi:10.2514/6.2000-4538.

[54] Page, A. and Steinberg, M., "High-Fidelity Simulation Testing of Control Allocation Methods," *Guidance, Navigation, and Control Conference and Exhibit*, AIAA Paper 2002-4547, Aug. 2002, doi:10.2514/6.2002-4547.

[55] Frost, S. A., Trinh, K. V., Taylor, B. R., Jutte, C. V., Burken, J., and Bodson, M., "Application of Structural Load Feedback in Flight Control," *AIAA Atmospheric Flight Mechanics Conference*, AIAA Paper 2011-6288, Aug. 2011, pp. 825–834, doi:10.2514/6.2011-6288.

[56] Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J., "Distributed Optimization and Stastical Learning via the Alternating Direction Method of Multipliers," *Foundation and Trends R© in Machine Learning*, Vol. 3, No. 1, 2010, pp. 1–122, doi:10.1561/2200000016.

[57] Dantzig, G. B., "Linear Programming Under Uncertainty," *Management Science*, Vol. 1, No. 3-4, 1955, pp. 197–281, doi:10.1287/mnsc.1.3-4.197.

[58] Dantzig, G. B. and Wolfe, P., "Decomposition Principle for Linear Programs," *Operations Research*, Vol. 8, No. 1, 1960, pp. 101–111, doi:10.1287/opre.8.1.101.

[59] Anand, R., Aggarwal, D., and Kumar, V., "A Comparative Analysis of Optimization Solvers," *Journal of Statistics and Management Systems*, Vol. 20, No. 4, 2017, pp. 623–635, doi:10.1080/09720510.2017.1395182.

[60] ApS, M., "MOSEK Optimization Toolbox for MATLAB," *Users Guide and Reference Manual*, Vol. 4, 2019.

[61] Domahidi, A., Chu, E., and Boyd, S., "ECOS: An SOCP Solver for Embedded Systems," *European Control Conference*, July 2013, pp. 3071–3076, doi:10.23919/ECC.2013.6669541.

[62] Stellato, B., Banjac, G., Goulart, P., Bemporad, A., and Boyd, S., "OSQP: An Operator Splitting Solver for Quadratic Programs," *Mathematical Programming Computation*, Vol. 12, February 2020, pp. 637–672, doi:10.1007/s12532-020-00179-2.

[63] Mattingley, J. and Boyd, S., "CVXGEN: A Code Generator For Embedded Convex Optimization," *Optimization Engineering*, Vol. 13, 2012, pp. 1–27, doi:10.1007/s11081-011-9176-9.

[64] Blackmore, L., Açikmeşe, B., and Scharf, D. P., "Minimum-Landing-Error Powered-Descent Guidance for Mars Landing Using Convex Optimization," *Journal of Guidance Control and Dynamics*, Vol. 33, No. 4, July-August 2010, pp. 1161–1171, doi:10.2514/1.47202.

[65] Cui, L. and Yang, Y., "Disturbance Rejection and Robust Least-Squares Control Allocation in Flight Control System," *Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 6, 2011, pp. 1632–1643, doi:10.2514/1.52234.

[66] Choi, B., Kim, H., and Kim, Y., "Robust Control Allocation with Adaptive Backstepping Flight Control," *Journal of Aerospace Engineering*, Vol. 228, No. 7, 2014, pp. 1033–1046, doi:10.1177/0954410013483687.

[67] Duan, M. and Okwudire, C., "Proxy-Based Optimal Dynamic Control Allocation for Multi-Input Multi-Output Over-Actuated Systems," *Dynamic Systems and Control Conference*, ASME Paper DSCC2017-5343, Oct. 2017, doi:10.1115/DSCC2017-5343.

[68] Duan, M. and Okwudire, C., "Proxy-Based Optimal Control Allocation for Dual-Input Over-Actuated Systems," *IEEE/ASME Transactions on Mechatronics*, Vol. 23, No. 2, 2018, pp. 895–905, doi:10.1109/TMECH.2018.2796500.

[69] Ma, J., Li, P., Li, W., and Zheng, Z., "Performance Comparison of Control Allocation for Aircraft with Control Effectiveness Uncertainties," *Asia Simulation Conference — 7th International Conference on System Simulation and Scientific Computing*, IEEE Paper, Oct. 2008, pp. 164–169, doi:10.1109/ASC-ICSC.2008.4675349.

[70] de Almeida, F. A., "Robust Off-Line Control Allocation," *Aerospace Science and Technology*, Vol. 52, May 2016, pp. 1–9, doi:10.1016/j.ast.2016.02.002.

[71] Ben-Tal, A. and Nemirovski, A., "Robust Solutions of Linear Programming Problems Contaminated with Uncertain Data," *Math. Programming*, Vol. 88, No. 3, 2000, pp. 411–424.

[72] Ben-Tal, A. and Nemirovski, A., *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*, SIAM, 2001.

[73] Nemirovski, A., *Lectures on Robust Convex Optimization*, H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta Georgia 30332-0205 USA, November 2012.

[74] Andersen, E., Roos, C., and Terlaky, R., "On Implimenting a Primal-Dual Interior Point Method for Conic Quadriatic Optimization," *Journal of Mathematical Programming*, Vol. 95, Fubruary 2003, pp. 249–277, doi:10.1007/s10107-002-0349-3.

[75] Andersen, K., "A Modified Schur-Complement Method for Handling Dense Columns in Interior-Point Methods for Linear Programming," *ACM Transactions on Mathematical Software*, Vol. 22, No. 3, 1996, pp. 348–356, doi:10.1145/232826.232937.

[76] Goldfarb, D. and Scheinbeg, K., "A Product-Form Cholesky Factorization Method for Handling Dense Columns in Interior Point Methods for Linear Programming," *Journal of Mathematical Programming*, Vol. 99, No. 1, 2004, pp. 1–34, doi:10.1007/s10107-003-0377-7.

[77] Goldfarb, D. and Scheinbeg, K., "Product-Form Cholesky Factorization In Interior Point Methods for Second-Order Cone Programming," *Journal of Mathematical Programming*, Vol. 103, No. 1, 2005, pp. 153–179, doi:10.1007/s10107-004-0556-1.

[78] Bordignon, K. A. and C., D. W., "Null-Space Augmented Solutions to Constrained Control Allocaiton Problems," *Guidance, Navigation, and Control Conference*, AIAA Paper 95-3209, 1995, pp. 328–333, doi:10.2514/6.1995-3209.

[79] Orr, J. S. and Slegers, N. J., "High-Efficiency Thrust Vector Control Allocation," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 2, 2014, pp. 374–382, doi:10.2514/1.61644.

[80] Durham, W. C. and Bordignon, K. A., *Aircraft Control Allocation*, John Wiley and Sons, 2017.

[81] Açıkmeşe, B. and Ploen, S. R., "Convex Programming Approach to Powered Descent Guidance for Mars Landing," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 5, 2007, pp. 1353–1366, doi:10.2514/1.27553.

[82] The MathWorks Inc., "MATLAB version 23.2.0 (R2023b)," 2023.

[83] Morelli, E. A. and Cunningham, K., "Aircraft Dynamic Modeling in Turbulence," *AIAA Atmospheric Flight Mechanics Conference*, AIAA-2012-4650, Aug. 2012, doi:10.2514/6.2012-4650.

[84] Jordan, T. L., M., L. W., and Hill, J. S., "Airborne Subscale Transport Aircraft Research Testbed - Aircraft Model Development," *AIAA Guidance Navigation and Control Conference and Exhibit*, AIAA-2005-6432, Aug. 2005, doi:10.2514/6.2005-6432.

[85] Jordan, T. L., Foster, J. V., Bailey, R. M., and Belcastro, C. M., "AirSTAR: A UAV Platform for Flight Dynamics and Control System Testing," *AIAA Aerodynamic Measurement Technology and Ground Testing Conference*, AIAA-2006-3307, Jun. 2006, doi:10.2514/6.2006-3307.

[86] Murch, A. M., "A Flight Control System Architecture for the NASA AirSTAR Flight Test Infrastructure," *AIAA Guidance Navigation and Control Conference and Exhibit*, AIAA-2008-6990, Aug. 2008, doi:10.2514/6.2008-6690.

[87] Shah, G. H., Cunningham, K., Foster, J. V., Fremaux, C. M., Stewart, E. C., Wilborn, J. E., Gato, W., and Pratt, D. W., "Wind-Tunnel Investigatin of Commercial Transport Aircraft Aerodynamics at Extreme Flight Conditions," *World Aviation Congress and Display*, SAE 2002-01-2912, Nov. 2002, doi:10.4271/2002-01-2912.

[88] Murch, A. M. and Foster, J. V., "Recent NASA Research on Aerodynamic Modeling of Post-Stall and Spin Dynamics of Large Transport Airplanes," *AIAA Aerospace Sciences Meeting and Exhibit*, AIAA 2007-463, Jan. 2007, doi:10.2514/6.2007-463.

[89] Hueschen, R. M., "Development of the Transport Class Model (TCM) Aircraft Simulation From a Sub-Scale Generic Transport Model (GTM) Simulation," *NASA TM-2011-217169*, August 2011.

[90] Grauer, J. A. and Morelli, E. A., "Generic Global Aerodynamic Model for Aircraft," *Journal of Aircraft*, Vol. 52, No. 1, 2015, pp. 13–20, doi:10.2514/1.C032888.

[91] Garza, F. R. and Morelli, E. A., "A Collection of Nonlinear Aircraft Simulations in MATLAB," *NASA TM-2003-212145*, January 2003.

[92] Miller, L. J., "Linear Parameter-Varying Control Compatible with Excitation for Real-Time Modeling," *AIAA SCITECH 2024 Forum*, AIAA-2024-2752, Jan. 2024, doi:10.2514/6.2024-2752.

[93] Miller, L. J. and Pei, J., "A Comparison of Control Allocation Methods in the Presence of Parametric Model Uncertainty," *AIAA SCITECH 2025 Forum*, AIAA-2025-2240, Jan. 2025, doi:10.2514/6.2025-2240.