A ROBUST CONTROLLER DESIGN FOR A TAILLESS UAV WITH EFFECTIVE CONTROL ALLOCATION

by

ETHEM HAKAN ORHAN

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN AEROSPACE ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2019

To Elif, Ali, and Banu

ACKNOWLEDGEMENTS

I would like to thank my supervising professor Dr.Kamesh Subbarao for his constant support and encouragement. His guidance always helped me find my way during my research. I also thank my committee members Dr.Atilla Doğan, Dr.Dudley E. Smith and Dr.Animesh Chakravarthy for their interest in my work and for taking time to be in my committee.

I would like to express my gratitude to my employer and sponsor, Turkish Aerospace Industries, Inc., for the generous financial support they provided.

Finally, I would like to thank my family for all their sacrifice and their understanding throughout this journey.

December 03, 2019

ABSTRACT

A ROBUST CONTROLLER DESIGN FOR A TAILLESS UAV WITH EFFECTIVE CONTROL ALLOCATION

ETHEM HAKAN ORHAN, M.S.

The University of Texas at Arlington, 2019

Supervising Professor: Dr.Kamesh Subbarao

Attempts to completely remove the tails from aircraft can be dated back to the early days of modern aviation. A number of stability and control problems arising from the unique characteristics of the configuration resulted in poor handling qualities and some dangerous flight characteristics in the early designs. Lately, this configuration is becoming widespread again and the current state-of-the-art of fly-by-wire technology and modern control design techniques enable design of tailless aircraft which are safe to fly.

In this thesis, a study on the application of modern robust control design techniques on a tailless UAV is presented. A nonlinear mathematical model for the aircraft is constructed and control laws are synthesized using μ-synthesis approach. Three different scheduling methods are investigated for the control laws: ad-hoc linear interpolation, synthesis using simplified linear parameter varying models and stability preserving interpolation. A control allocation module is implemented to distribute the controller commands into highly coupled control effectors in real time. Two different allocation approaches are investigated: Cascaded Generalized Inverses and Weighted

Least Squares. Effector limits and failure conditions are taken into account in an efficient way in allocation. A simulation study is performed using the nonlinear aircraft model, control laws, and control allocation models for various maneuvers and control effector failure cases.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Perfection is achieved, not when there is nothing more to add, but when there is nothing left to take away.

Antoine de Saint-Exupery

Ever since the early days of modern aviation history, there have been attempts to completely remove the tails from aircraft to reduce aerodynamic drag and weight [1]. In 1940's, the German Horten brothers designed a series of flying wing aircraft and also experimented with different approaches for achieving directional stability and control using mechanical control systems [2]. In the same period, Jack Northrop built a number of flying wing prototypes in the U.S. Until the late 1970's, all of these attempted aircraft designs were failures, primarily due to poor handling qualities and some dangerous flight characteristics [1][3][4].

The development of modern Fly-By-Wire (FBW) Flight Control System (FCS) technology finally made it possible to design tailless aircraft that are safe to fly. Also in the mid-1970's, the progress in stealth technology made the removal of vertical tails even more desirable, because it could reduce the side sector radar cross section of the aircraft [1][2][4]. Even though these two technologies have been available for over 40 years, only one manned aircraft without tails has been designed, developed, and put into operational service: the B-2 [1]. However, on the unmanned side, a number of tailless low observable (LO) vehicles have appeared in the last few decades. Some well-known examples are Lockheed-Martin/Boeing RQ-3 Darkstar, Lockheed-Martin Polecat, Lockheed-Martin RQ-170 Sentinel which have been designed for ISR (In-

telligence, Surveillance and Reconnaissance) missions, and Boeing X-45A, Northrop-Grumman X-47B, Boeing X-45C and BAE Systems Taranis, designed for combat missions. It is beyond doubt that the rapid improvements in technologies including payload, communications, processing and autonomy software made this leap possible [5].

In very broad terms, the problems attributed to tailless aircraft can be outlined as poor stability in longitudinal and directional axes, deficiency in directional control and high coupling between longitudinal, lateral and directional controls [6][7]. These made the airplane unsafe for human pilot who relied on simple conventional flight controls. Modern control design techniques, on the other hand, propose effective solutions for almost all these problems. The key point is taking the task of stabilizing the aircraft from the pilot/operator to a 'smart' flight control system. Consequently, the workload of the pilot/operator is limited to mission oriented tasks.

In this thesis, a study about the application of modern robust control design techniques on a tailless Unmanned Air Vehicle (UAV) is presented. The aircraft of concern is RQ-3 Darkstar, a LO UAV from 1990's, which is a high aspect ratio (AR) design aimed for long endurance ISR missions.

For the first step, a nonlinear mathematical model of the vehicle is constructed, based on the aircraft characteristics which are publicly available. Some missing parameters are selected in order to assure the model integrity. A control effector suite composed of three trailing edge flaps and two spoilers on each wing is implemented in the model, which is known to be different from the original RQ-3. Digital DATCOM [8] has been extensively used for extracting the aerodynamic characteristics of the vehicle. Once the nonlinear mathematical model is constructed, a number of linear-time invariant (LTI) models have been generated for selected operating points.

The flight control system is designed to be composed of two distinct blocks: a controller and a control allocation module. The controller block includes the LTI control laws synthesized for LTI aircraft models generated at different operating points. A multivariable robust control design approach, namely μ-synthesis has been used for this synthesis. This controller is designed in a way to generate 'virtual' controls which are perfectly decoupled in roll, pitch and yaw axes. The control allocation module is designed to distribute these virtual commands to highly coupled control effectors in real time. The limits and failure conditions of each individual effector are taken into account in an efficient way during allocation. Failure detection is assumed to be accomplished by some other mechanisms, which are not in the scope of this work, but failure isolation is a direct outcome of the implemented allocation approaches.

An evaluation of controller design techniques can be made based on: nominal and robust stability guarantees, nominal and robust performance, complexity, ease of implementation and considerations such as actuator limits, saturation, etc. In this sense μ-synthesis is considered to be one of the most powerful methodologies available for multivariable control design today. One of its main drawbacks is the scheduling difficulties due to big size and unmanageable structure of the controller [9]. In this work three different scheduling approaches are investigated. First, an ad-hoc linear interpolation of controllers which are synthesized for consecutive operating points is implemented. Then, a synthesis method using simplified LPV (Linear Parameter Varying) models [10] is applied. Finally, a stability preserving interpolation approach [11] is investigated.

Control allocation constitutes an important part of this work. Various online control allocation schemes have been investigated and two of them have been implemented: First, a comparably simple approach: Cascaded Generalized Inverses

(CGI) [12], then, a more complex but capable method: Weighted Least Squares (WLS) [13] is tested by simulation.

This thesis is organized as follows:

In Chapter 2, a general introduction for Robust Multivariable Control is presented. Basic definitons and theorems are given, concepts like interconnection model, linear fractional transformation are explained. Robustness measures for a closed loop system are discussed and the notion of structured singular value is defined. Then μ-synthesis design approach is explained with its theoretical background and practical implementation steps. Finally, gain scheduling problem for μ-synthesis is addressed and the two approaches, synthesis using simplified LPV model and stability preserving interpolation are discussed in detail.

Chapter 3 is dedicated to Control Allocation. First, the concept of control allocation and its role in a flight control system is discussed in detail. Actuator constraints and failures and ways for handling these using control allocation is explained. The control allocation problem is formulated in the mathematical sense and various approaches for solving it are defined. Two of these methods, Cascaded Generalized Inverses (CGI) and Weighted Least Squares (WLS), which are selected to be implemented in this work are discussed in more detail.

In Chapter 4, a case study on controller synthesis and control allocation implementation is presented with all its steps. First, main constituents of RQ-3 Darkstar nonlinear model are introduced, trim and linearization steps are explained. The components of the interconnection models, which are used in controller synthesis are defined. The maneuver scenarios and failure cases used in the simulations are explained. Finally, the results of simulation work are presented in a form of time history plots with brief discussion for each.

Chapter 5 is reserved for the discussion on the results and the future work.

CHAPTER 2

ROBUST MULTIVARIABLE CONTROL

2.1 Background

In simplest terms, control laws are algorithms that process the reference commands and sensor information and produce actuator commands to achieve the desired responses [9]. A control system is robust if it is insensitive to disturbances and to inaccuracies in the plant model, which are referred to as model uncertainty [14].

A challenge of flight control laws is their multivariable nature due to multiple control effectors, multiple sensors, multiple disturbances, changes in mass properties, sensor errors, multiple control objectives and multiple uncertainties associated with the models [9].

Until mid 1980's, classical control was the only practical approach for aircraft controller design. The essence of classical design was successive single input/single output (SISO) loop closures guided by a good deal of intuition and experience that assisted in selecting the control system structure. Although tools like root locus, Bode and Nyquist plots helped designers, the design procedure became increasingly difficult as more loops were added and did not guarantee success when the dynamics were multivariable [15].

Multivariable control techniques are considered to handle multiloop control problems in a formal, systematic and efficient manner [16]. Various multivariable control laws analysis and synthesis techniques have been proposed, many extensions and variations have been investigated, but only some of these could be implemented in aircraft controllers. The first significant applications of the multivariable control

methodologies in aircraft started in late 1970's, by the Boeing Company. The results demonstrated that multivariable control laws design techniques offered significant advantages over classical techniques in the solution of multiloop control problems. Motivated by the initial success, practical multivariable design methodologies have been further developed at Boeing and successfully applied to a wide range of control problems over years. A European action group involving industry and academia was established in 1990's to demonstrate application of advanced methodologies to design robust controllers for some realistic flight control benchmark problems. The results have shown that modern techniques could be used to design controllers for realistic applications and had much potential in terms of improved robustness, better performance, decoupled control, and simplification of the design process. However, some methods are concluded not yet to have the maturity required for industrialization [16].

Most multivariable design methods are variants of a few basic approaches for the solution of multivariable control problem. These approaches can simply be listed as [9]:

1. Formal optimization problems, consisting of linear quadratic Gaussian problem in its various manifestations;
2. Numerical optimization problems, utilizing the same optimization philosophy in a setting that do not necessarily yield analytic solutions or find global optima or even guarantee stabilizing answers;
3. Frequency domain methods, consisting of various adaptations of classical Bode and Nyquist SISO techniques to multivariable design, which close sequential loops around multivariable plants with singular value based loop shaping;

4. Eigenstructure assignment methods, concerned with locations of closed-loop eigenvalues and directions of closed-loop eigenvectors, as constrained by the limitations of linear feedback.

Numerous publications are available on robust multivariable control techniques. A selection from the literature, covering the gain scheduling problem or implementation of the methods in real aircraft problems are presented in the following.

Ref. [17] covers an application of H_∞ controller on a generic VSTOL aircraft model, inspired from Harrier. A gain scheduling approach using coprime factor formulation of robust stabilization problem is developed, a weighting selection procedure is proposed and a desaturation scheme against actuator saturation is applied. Ref. [16] takes a fly-by-wire small commercial aircraft as a benchmark problem. A model matching H_∞ control problem is solved via the μ-synthesis approach, and the designed flight control laws are evaluated by pilot-in-the-loop simulations using a ground-based simulator. Ref. [18] proposes a gain-scheduling flight controller design based on a blending/interpolating methodology using an optimal LFT based technique. A blending/interpolating scheduling controller is driven by using the fixed controllers in a robust setup. Nonlinear simulations are performed for a flexible aircraft. Ref. [10] presents a strategy for the design of controllers based on a simplified LPV model of a UAV. The dependence of the LPV model on the varying parameter is reformulated in terms of a μ-synthesis problem. Simulation results are compared for a straight μ design and a gain-scheduling μ control scheme. Ref. [19] presents a trajectory tracking controller design approach for a UAV using LPV methods. A two loop structure is proposed where the inner loop LPV controller is designed first using μ-synthesis. Then, the inner loop is approximated with a reference model and the outer loop is designed using loop-shaping techniques. Ref. [20] discusses a nonlinear robust control design procedure for a micro air vehicle that applies μ-synthesis tech-

nique, which overcomes structured uncertainty of the control plant and is valid over the entire flight envelope. Ref. [21] illustrates an approach to gain scheduling for H_∞ controllers in which so-called hidden coupling terms are removed. Potential performance improvement is demonstrated by simulation. Ref. [22] presents a method to develop a self-scheduled controller for a high performance aircraft using LPV techniques, combined with μ-synthesis. The ability of the controller to achieve specified handling qualities over a wide range of flight conditions is demonstrated by nonlinear simulations. Ref. [23] is concerned with the application of robust controller synthesis and analysis tools on Bell 205 teetering-rotor helicopter, mainly for H_∞ loop-shaping approach. A quantitative assessment based on in-flight tests and desktop simulations and a qualitative assessment based on the pilot comments are presented.

2.2 Preliminaries

The purpose of this section is to provide a short hand reference for the reader about some multivariable control related terms and concepts, which are referred within the thesis. Since, in depth information about these can be found in many textbooks on the subject, the content here is limited to brief reminders.

2.2.1 Definitions

Definition 1 (Analytic Functions):

Defining $\mathbb{D}$ as the union set of real and complex numbers, for any $x_0 \in \mathbb{D}$, if Taylor series expansion of a function $f(x)$ exists and converges to $f(x)$ in some nonzero interval $|x - x_0| < R$, then $f(x)$ is said to be analytic in $\mathbb{D}$. This implies that $f(x)$ is infinitely differentiable at $x_0 \in \mathbb{D}$. If the function is not analytic at x_0, it is singular there [24].

Definition 2 (Proper Transfer Function):

A proper transfer function $G(s)$ is a transfer function in which the degree of numerator $N(s)$ (number of zeroes) does not exceed the degree of the denominator $D(s)$ (number of poles).

$$G(s) = \frac{N(s)}{D(s)} \; ; \; \deg(N(s)) \leq \deg(D(s))$$

A proper transfer function will not grow unbounded and is definite as the frequency approaches infinity [25], i.e.:

$$\|G(j\infty)\| < \infty$$

Definition 3 (Well-Posedness):

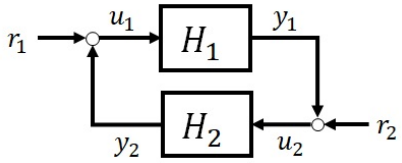

Figure 2.1: Two system interconnection

For the two linear systems $H_1$and H_2, where;

$$H_1 = \begin{bmatrix} A_1 & B_1 \\ C_1 & D_1 \end{bmatrix}; H_2 = \begin{bmatrix} A_2 & B_2 \\ C_2 & D_2 \end{bmatrix}$$

with respective states, inputs and outputs, (x_1, u_1, y_1), (x_2, y_2, u_2) and transfer functions $H_1(s)$ and $H_2(s)$ are proper, the interconnection shown in Figure 2.1 can be rewritten as:

$$\begin{bmatrix} I & -D_2 \\ -D_1 & I \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} 0 & C_2 \\ C_1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}$$

This interconnected system is called well-posed if the internal signals (u_1, u_2) of the feedback loop are uniquely defined for every choice of states (x_1, x_2) and external inputs (r_1, r_2). This implies the invertibility of the matrix:

$$\begin{bmatrix} I & -D_2 \\ -D_1 & I \end{bmatrix}$$

which also implies invertibility of $I - D_1 D_2$ or $I - D_2 D_1$ alike. The significance of well-posedness is that, once the interconnected system is solved for u_1, u_2 in terms of x_1, x_2 and r_1, r_2, u_1 and u_2 can be eliminated from the state-space representation of the closed-loop system with states x_1, x_2 [26].

Definition 4 (Structured and Unstructured Uncertainty):

Uncertain elements appearing in practical systems may be classified as structured or unstructured uncertainty. Uncertainty in model parameters is called structured as it models the uncertainty in a structured manner. An example of structured uncertainty is any parametric variation in poles and zeros of the plant transfer function. Analogously, lumped dynamics uncertainty is called unstructured. An example of unstructured uncertainty includes frequency dependent uncertainty due to usually neglected high-frequency modes in plant dynamic models [14][25].

Norms

The modern approach for characterizing closed-loop performance objectives is to measure the size of certain closed-loop transfer function matrices using various matrix norms. Matrix norms provide a measure of how large output signals can get for certain classes of input signals. Optimizing these types of performance objectives, over the set of stabilizing controllers is the main thrust of recent optimal control theory [27].

In time domain, finite dimensional linear systems can be represented as sets of linear ordinary differential equations (ODE) and signals as measurable functions of time. Using Laplace transform, both signals and systems can be represented as functions of a complex variable, 's' [28].

A linear dynamical system with a state space model:

$$\begin{bmatrix} \dot{x} \\ e \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x \\ d \end{bmatrix} \tag{2.1}$$

can be written in transfer function form as:

$$e(s) = T(s)d(s)$$

where;

$$T(s) = C(sI - A)^{-1}B + D$$

Definition 5 (L_2-norm):

2-norm (the energy) of a signal is defined as [27]:

$$\|e\|_2 = \left(\int_{-\infty}^{\infty} e(t)^2 dt \right)^{1/2}$$

Definition 6 (Frobenius norm):

Frobenius norm of a complex matrix M is defined as [27]:

$$\|M\|_F = \sqrt{\text{trace}(M^* M)}$$

Definition 7 (H_2-norm):

H_2-norm of the transfer matrix $T(s)$ in frequency domain is defined as [27]:

$$\|T\|_2 = \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} \|T(j\omega)\|_F^2 d\omega \right)^{1/2}$$

Definition 8 (H_∞-norm):

H_∞-norm of the transfer matrix $T(s)$ in frequency domain is defined as [27]:

$$\|T\|_\infty = \max_{\omega} \overline{\sigma} \left(T(j\omega) \right)$$

Definition 9 (L_2 gain):

The L_2 (or RMS) gain from $d \to e$ is defined as [27]:

$$\max_{d \neq 0} \frac{\|e\|_2}{\|d\|_2}$$

and it is equal to the H_∞-norm of the transfer matrix T [27]:

$$\max_{d \neq 0} \frac{\|e\|_2}{\|d\|_2} = \|T\|_\infty$$

Definition 10 (Sub-multiplicative property):

If any norms of two matrices A and B satisfies:

$$\|AB\| \leq \|A\|\|B\|$$

then the norm is called a sub-multiplicative norm.

Weighted Norms [27]:

In any performance criterion, relative magnitude of outside influences and relative importance of the magnitudes of regulated variables should also be accounted for. So, if the performance objectives are defined in the form of a matrix norm, it should actually be a weighted norm:

$$\|W_L T W_R\|$$

where the weighted function matrices W_L and W_R are frequency dependent, to account for bandwidth constraints and spectral content of exogenous signals. Assume that W_L and W_R are diagonal, stable transfer function matrices, with diagonal entries L_i and R_i such that:

$$W_L = \begin{bmatrix} L_1 & 0 & & 0 \\ & & \cdots & \\ 0 & L_2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & L_{n_e} \end{bmatrix}$$

$$W_R = \begin{bmatrix} R_1 & 0 & & 0 \\ & & \cdots & \\ 0 & R_2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & R_{n_e} \end{bmatrix}$$

then the weighted transfer function matrix has the form shown in Figure 2.2

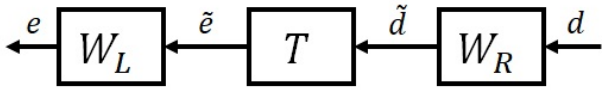

Figure 2.2: Weighted performance objectives

where $e = W_L\tilde{e} = (W_LT)\tilde{d} = (W_LTW_R)d$. Bounds on the quantity $\|W_LTW_R\|_\infty$ will imply bounds about the sinusoidal steady-state behavior of the signals $\tilde{d}$ and $\tilde{e}$. $\|W_LTW_R\|_\infty \leq 1$ if and only if for every fixed frequency ω and all sinusoidal disturbances $\tilde{d}$ satisfying $|\tilde{d}_i| \leq |W_{R_i}(j\omega)|$,the steady-state error components will satisfy:

$$|\tilde{e}_i| \leq \frac{1}{|W_{L_i}(j\omega)|}$$

The weighted H_∞ norm does not actually give element-by-element bounds on the components of $\tilde{e}$ based on element-by-element bounds on the components $\tilde{d}$. The precise bound it gives is in terms of Euclidean norms of the components of $\tilde{e}$ and $\tilde{d}$.

Small Gain Theorem:

Considering the simple closed-loop system shown in Figure 2.3, where $\Delta(s)$ and $M(s)$ are stable and proper transfer functions, the small gain theorem states that if the H_∞ norm of $\Delta(s)M(s)$ is less than unity the interconnected system is stable.

Theorem 1(Small Gain Theorem):

If $\|\Delta(s)M(s)\|_\infty < 1$, then the closed-loop system is stable [25].

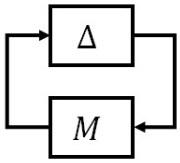

Figure 2.3: Simple closed-loop system

H_∞ Control:

Measuring the performance of a system in terms of the H_∞ norm has advantages in dealing with the uncertainties arising in control design. Consider the LTI systems M and Δ, where the H_∞ norm of these systems satisfy the sub-multiplicative property, i.e.:

$$\|M\Delta\|_\infty \leq \|M\|_\infty \|\Delta\|_\infty$$

The small gain theorem states that a plant M is robustly stable to perturbations Δ entering the system with $\|\Delta\|_\infty \leq \gamma$ if $\|M\|_\infty \leq \frac{1}{\gamma}$. The goal of the H_∞ design is to minimize the ∞ norm of the system M in order to increase the robustness of the system to the uncertainties represented in the Δ block. In H_∞ control the uncertainties in signal and system components are modeled as elements of a bounded set [28].

2.2.2 Parametric Uncertainty

Parameters in a state space or transfer function representation of a system are assumed to lie in a set given as:

$$P \in \{P_0 + W\Delta, \Delta \in [-k, k]\}$$

where P_0 is the nominal value of the parameter, Δ is allowed to take any value between $-k$ and k and W is the problem dependent scaling factor. It is common practice to scale the parametric uncertainty such as $k = 1$. Some common representations of models with uncertainties are:

Additive: $P \in \{P_0 + \Delta W,\ \|\Delta\| < 1\}$,

Multiplicative: $P \in \{P_0 + (I\Delta W),\ \|\Delta\| < 1\}$, and

Coprime factor: $P \in \{N_0 + \Delta_N)(M_0 + \Delta_M)^{-1},\ \|\Delta_M \Delta_N\| \leq \epsilon\}$

forms, where N_0 and M_0 are the coprime factors of the nominal plant $P_0 = N_0 {M_0}^{-1}$ and Δ_N, Δ_M denote the uncertainty on each factor [28].

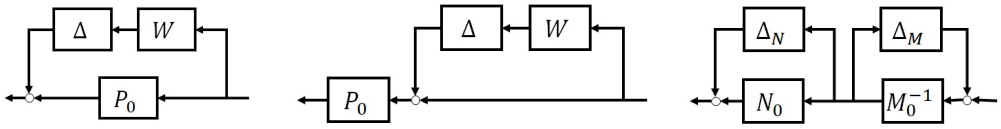

Figure 2.4: Uncertainty representations: additive, multiplicative, coprime factor

2.2.3 Interconnection Model

For a tracking problem with external disturbance, measurement noise and control input signal limitations, where K is some controller to be designed and P is the system to be controlled, the closed-loop performance objectives can be formulated as weighted closed-loop transfer functions which are to be made small through feedback.

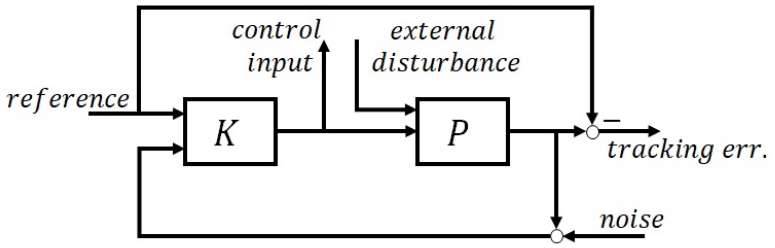

Figure 2.5: Basic interconnected system with plant P and controller K

An interconnection of nominal model together with uncertainty and performance weights as shown in Figure 2.6, can be used for robustness analysis and controller synthesis [27].

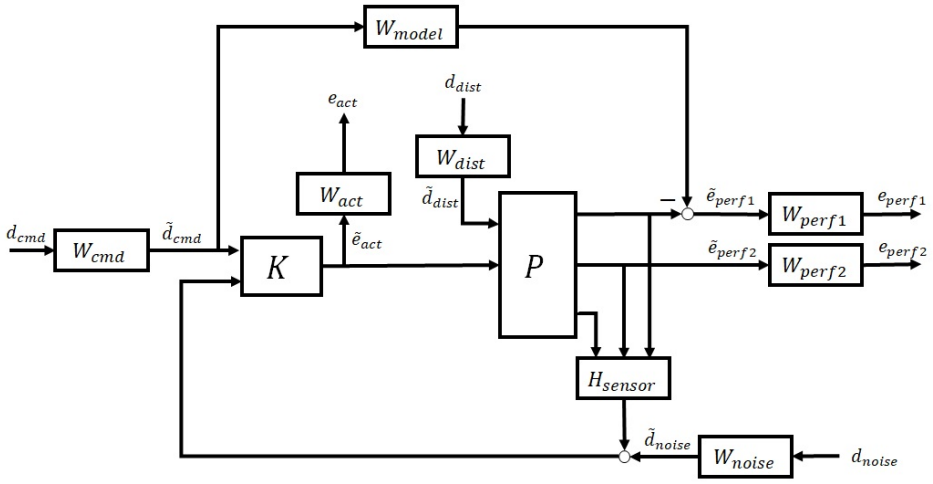

Figure 2.6: Interconnected system with weights

Here;

- W_{cmd} shapes the normalized reference command signals into the actual (or typical) reference signals that we expect to occur.
- W_{model} represents a desired ideal model for the closed-looped system, used for problems with tracking requirements.
- W_{dist} shapes the frequency content and magnitude of the exogenous disturbances affecting the plant.
- W_{perf1} weights the difference between the response of the plant and the response of the ideal model W_{model} .
- W_{perf2} penalizes variables internal to the process P, such as variables that are not part of the tracking objective.
- W_{act} is used to shape the penalty on control signal use.

- W_{noise} represents frequency domain models of sensor noise.
- H_{sensor} represents a model of sensor dynamics.

2.2.4 Linear Fractional Transformation (LFT)

An uncertain system can be represented in a linear fractional transformation (LFT) form, where M is the nominal model together with uncertainty and performance weights for analysis problems and Δ is a block diagonal matrix with different types of uncertainties as the block diagonal elements. Here, the controller is also absorbed in M.

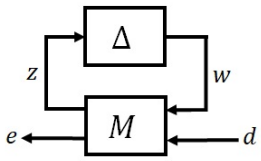

Figure 2.7: Upper connected model

Here;

d shows the exogenous disturbances,

e shows the regulated variables, i.e., errors.

The system can simply be written as:

$$\begin{bmatrix} z \\ e \end{bmatrix} = M \begin{bmatrix} w \\ d \end{bmatrix} ; \; w = \Delta z$$

and the transfer function M can be partitioned as:

$$M = \begin{bmatrix} M_{11}(s) & M_{12}(s) \\ M_{21}(s) & M_{22}(s) \end{bmatrix}$$

Selecting Δ such that the set of equations is well posed and the vectors e and d are related by:

$$e = F_u(M, \Delta)d$$

where;

$$F_u(M, \Delta) = M_{22} + M_{21}\Delta(I - M_{11})^{-1}M_{12} \tag{2.2}$$

F_u is called an upper linear fractional transformation (LFT) of M with Δ.

Likewise, a feedback loop can be represented in a linear fractional transformation (LFT) form, where P is the nominal model together with uncertainty and perfromance weights and K is an LTI system with its own states for output feedback control.

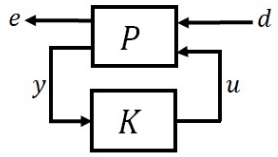

Figure 2.8: Lower connected model

Here;

u shows the manipulated variables, i.e., controls,

y shows the sensed variables, i.e., measurements.

The plant dynamics can be written as:

$$\begin{bmatrix} \dot{x} \\ e \\ y \end{bmatrix} = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} x \\ d \\ u \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \begin{bmatrix} x \\ d \\ u \end{bmatrix}$$

where;

$$P_{11} = [A] \; ; \; P_{12} = \begin{bmatrix} B_1 & B_2 \end{bmatrix} ; \; P_{21} = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} ; \; P_{22} = \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix}$$

and for the controller dynamics;

$$\begin{bmatrix} \dot{x}_K \\ u \end{bmatrix} = K \begin{bmatrix} x_K \\ y \end{bmatrix}$$

where;

$$K = \begin{bmatrix} A_K & B_K \\ C_K & D_K \end{bmatrix}$$

Similarly, the transfer function from inputs to outputs when lower loop is closed with K is:

$$F_l(P, K) = P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21} \tag{2.3}$$

This is called a lower linear fractional transformation (LFT) [28]. The overall system can be represented as the connection of lower and upper LFT models as shown in Figure 2.9.

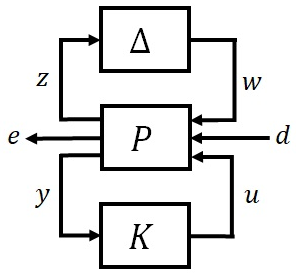

Figure 2.9: Overall system

2.2.5 Stability and Performance

Considering the system shown is Figure 2.9;

- Nominal stability refers to the property that the closed-loop system is stable for one model at the center of the model set, i.e. $\Delta(s) = 0$.
- Nominal performance refers that, in addition to stability, the nominal closed-loop system should satisfy the performance requirements.
- Robust stability refers the property that the closed-loop system is stable for all stable Δ where $\|\Delta\|_\infty \leq 1$, i.e. the controller must stabilize all plants defined by the uncertainty description $F_u(P, \Delta)$.
- Robust performance refers that, the performance specifications should be satisfied by the closed-loop system for all plants defined by the uncertain description [29].

2.2.6 Robustness Measures

In classical control, robustness is ensured by providing sufficient gain and phase margins to counteract the effects of inaccurate modeling or disturbances. In terms of the Bode magnitude plot, it is known that the loop gain should be high at low frequencies for performance robustness but low at high frequencies, where unmodeled dynamics may be present, for stability robustness. Classical control design techniques are generally in the frequency domain, so they afford a convenient approach to robust design for SISO systems. However, it is well known that the individual gain margins, phase margins, and sensitivities of all the SISO transfer functions in a multivariable system have little to do with its overall robustness [15].

Modern control techniques provide a direct way to design multiloop controllers for MIMO systems by closing all the loops simultaneously. As coupling generally exists between inputs and outputs of a MIMO system, approaches such as making

several individual SISO frequency plots for various combinations of the inputs and outputs and examining gain and phase margins may not always yield much insight into the true behavior of the system [15]. However, the classical frequency-domain robustness measures may easily be extended to MIMO systems in a rigorous fashion by using the notion of the singular value.

Theorem 2 (Singular Value Decomposition) [29]:

Defining $\mathbb{C}$ as the set of complex numbers, let $A \in \mathbb{C}^{m\times n}$, there exists unitary matrices $U \in \mathbb{C}^{m\times m}$ and $V \in \mathbb{C}^{n\times n}$ such that:

$$A = U\Sigma V^*$$

where;

$$\Sigma = diag(\sigma_1, \sigma_2, \ldots, \sigma_p, 0)$$

and;

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0, \quad p = min(m, n)$$

$$\overline{\sigma}(A) = \sigma_{max}(A) = \sigma_1; the\ largest\ singular\ value\ of\ A$$

$$\underline{\sigma}(A) = \sigma_{min}(A) = \sigma_p; the\ smallest\ singular\ value\ of\ A$$

The multivariable Bode magnitude plot, which is the plot of the singular values of the transfer function matrix versus frequency, allows the experience of classical control theory to be applied to MIMO systems. Analogous to SISO counterpart, in a coupled MIMO system the minimum singular value of the loop gain should be large at low frequencies for robust performance and the maximum singular value of the loop gain should be small at high frequencies for robust stability [15].

2.2.7 Structured Singular Value - μ

The structured singular value is a straightforward generalization of the singular values for constant matrices.

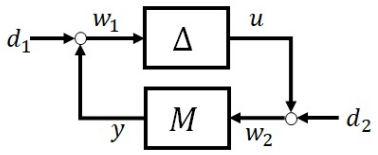

Figure 2.10: Standard feedback interconnection

Considering the standard feedback interconnection as shown in Figure 2.10, with stable $M(s)$ and $\Delta(s)$, the question is: how large Δ can be, in the sense of $\|\Delta\|_\infty$, without destabilizing the feedback system? It is known that the feedback system will become unstable if $\det(I - M\Delta) = 0$ for some $s \in \mathbb{C}$. Defining $\alpha > 0$ such that the closed-loop system is stable for all stable $\|\Delta\|_\infty < \alpha$, the maximum value of this α, α_{max} is called the robust stability margin. From small gain theorem:

$$\frac{1}{\alpha_{max}} = \|M\|_\infty = \max_{s\in\mathbb{C}} \overline{\sigma}(M(s)) = \max_{\omega} \overline{\sigma}(M(j\omega))$$

where;

$$\overline{\sigma}(M(s)) = \frac{1}{min\,\{\overline{\sigma}(\Delta) : det(I - M\Delta) = 0,\ \Delta\ is\ unstructured\}}$$

To quantify the smallest destabilizing structured Δ the concept of singular values is generalized. Defining structured singular value:

$$\mu_\Delta(M(s)) = \frac{1}{min\,\{\overline{\sigma}(\Delta) : det(I - M\Delta) = 0,\ \Delta\ is\ structured\}}$$

which is the largest structural singular value of $M(s)$ with respect to the structured Δ, and,

$$\frac{1}{\alpha_{max}} = \max_{s\in\mathbb{C}} \mu_\Delta(M(s)) \ = \max_{\omega} \mu_\Delta(M(j\omega))$$

for structured uncertainty [29]. An exact solution for μ does not exist. A solution can be approximated via upper and lower bounds on μ. The method of approximation depends on the structure of the Δ block [28]. The structured singular value μ

can be used to evaluate the robustness margins for a linear system with structured uncertainty as shown in Figure 2.7.

Theorem 3 (Robust stability) [28]:

Remembering the upper LFT definition in (2.2), robust stability can be defined as:

$$\forall \Delta, F_u(M,\Delta) \; is \; stable \; iff \; \sup_{\omega} \mu(M_{11}(j\omega)) \; \leq 1 \; , \; 0 \leq \omega \leq \infty$$

This theorem provides a test for the stability of the system for all allowable perturbations. Usually stability is not the only property of a feedback system that must be robust to perturbations. The effect of disturbances on the error signals can increase greatly and performance may degrade significantly when the nominal model is perturbed. A robust performance test is necessary to indicate the worst case level of performance associated with a given level of perturbations.

The robust performance problem can be formulated as a robust stability problem by associating a fictitious full block of uncertainty Δ_{perf}, with the performance inputs and outputs. The robust performance problem is equivalent to a robust stability problem but with respect to a different block structure. Consider a new $\underline{\Delta}$ structure defined as:

$$\underline{\Delta} = \left\{ \begin{bmatrix} \Delta & 0 \\ 0 & \Delta_{perf} \end{bmatrix} \; ; \; \Delta_{perf} \in \mathbb{C}^{n_v \times n_e} \right\}$$

Robust performance is defined by the following theorem.

Theorem 4 (Robust performance) [28]:

Robust performance is defined by:

$$\forall \Delta, F_u(M,\Delta) \; is \; stable \; and \; \|F_u(M,\Delta)\|_\infty < 1 \; iff \; \sup_{\omega} \mu_{\underline{\Delta}}(M(j\omega)) \leq 1 \; , \; 0 \leq \omega \leq \infty$$

This means that performance robustness of a closed-loop system can be evaluated by a μ test across all frequencies. The peak value on the μ plot determines the robustness properties.

2.3 μ-Synthesis

Structured singular value (μ) synthesis is a multivariable design method that can be used to directly optimize robust performance [9]. Performance specifications are defined as weighted transfer functions that describe magnitude and frequency content of external disturbances, pilot commands, atmospheric gusts, and sensor noise, as well as allowable magnitude and frequency content of generalized tracking errors, handling qualities, ride qualities, and actuator activity [9][27]. Specifications of performance and definition of the model set over which performance must be achieved are all incorporated into a single standard interconnection structure, upon which existing design algorithms can operate [9].

The μ-synthesis design technique combines the H_∞ control design with μ-analysis. Considering the standard robust performance μ-analysis framework, the overall system structure (Figure 2.9) and Theorem 4, it is desired to find a controller K achieving:

$$\inf_{stabilizing\ K} \sup_{\omega} \{F_l(P(j\omega), K(j\omega))\}$$

This minimization does not have a closed form solution. Upper bounds defined for complex μ can be used to approximate the solution. Using this fact the problem can be put in the following form:

$$\min_{stabilizing\ K} \left\{ \min_{D_\omega} \left\{ \max_{\omega} \overline{\sigma}\ \left[D_\omega F_l(P,K)(j\omega) D_\omega^{-1} \right] \right\} \right\}$$

or equivalently;

$$\min_{stabilizing\ K} \left\{ \min_{D_\omega} \left\| D_\omega F_l(P,K)(j\omega) D_\omega^{-1} \right\|_\infty \right\}$$

The approach used in solving this problem is to minimize the above expression for either K or D while holding the other constant. For fixed D it becomes an H_∞ optimal control problem. For K fixed, D can be formulated as a convex optimization problem. This process is applied iteratively until a satisfactory controller is achieved [28]. Main steps in μ-synthesis are as follows:

1. Interconnection Structure Definition
2. H_∞ Synthesis
3. μ Analysis
4. Rational Approximation of D-Scales
5. $D - K$ Iteration
6. Changing Weights

Interconnection Structure [9][27]:

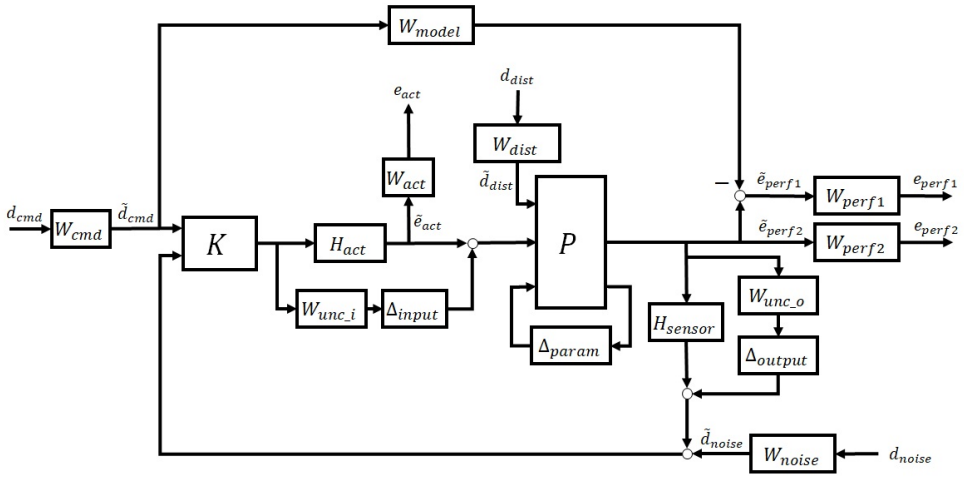

Figure 2.11: Generic interconnection structure

Interconnection structure is a state-space realization of the aircraft dynamics, augmented with handling qualities models and weighting functions with various inputs and outputs that specify control design goals. This structure is later used to define

the synthesis problem in the simple LFT form as shown in Figure 2.9. Some of the constituents of an interconnection structure are as follows:

Aircraft Model

The linear dynamic models of the bare airframe, actuators and the sensors are the central components of the interconnection structure. As appropriate, some of the dynamics such as low-frequency states that are not important to the design or high-frequency states, whose frequencies exceed intended bandwidth of the control loops may be omitted, truncated or residualized. A time delay in Pade-approximation form may be included for specific systems.

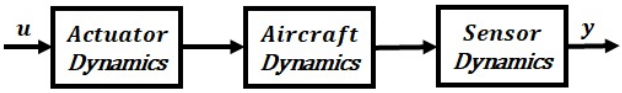

Figure 2.12: Plant dynamics

Performance Model

This model defines generalized errors e and generalized disturbances d of the model shown in Figure 2.11. These are the signals to be used to judge the quality of closed-loop performance. The most common generalized errors are tracking errors between the outputs of closed-loop system and some reference models, actuator deflections and actuator rates. Generalized disturbances are the various external inputs that excite the feedback loop and drive errors such as external disturbances (gusts, store drops, gun fire transients, etc.) and commands from pilots or outer loops.

Uncertainty Model

This model defines the set of plants over which performance objectives must be satisfied. It is represented by signals w and z which connect uncertain components Δ into the feedback loop. Uncertainties may be defined for inputs and outputs, and

for the system parameters. Design models used for flight control typically exhibit good fidelity at lower frequencies but they degrade rapidly at higher frequencies due to poorly modeled or neglected effects as aeroelasticity, actuator compliance, servo dynamics, computational/digital effects. Similar high frequency uncertainties are also associated with sensor hardware for various measured aircraft outputs. Another important type of uncertainty is internal to the design model, associated with such things as mass, inertia and/or aerodynamic coefficients.

Weights

The role of weights is to scale the interconnection structure (i.e., normalize it) such that control objectives in the unscaled structure are satisfied whenever the closed-loop gain from d to e in the scaled structure is less than unity for all unit-size perturbations Δ. The functions associated with signals d and e are called performance weights whereas those associated with w and z are called uncertainty weights. The weights are the specifications that drive the control design. They can either be used to determine achievable performance against fixed specifications or to trade off some specifications. In this manner, all of the classical control design knobs are embodied in the size and shape of the weights.

The weights chosen for the tracking errors can be thought of as penalty functions. That is, weights should be large in frequency ranges where small errors are desired and small where larger errors can be tolerated. Normally it is flat at low frequency and rolls off at high frequency. Often, accurate matching of the ideal model at low frequency is desired and less accurate matching at higher frequency is required, which turns out a weighting function which is flat at low frequency and rolls off and flattens out at a small, nonzero value at high frequency.

Actuator deflection and rate weights are used to penalize larger and faster deflections and thereby minimize control activity. Their size could be chosen to

make the normalized deflection and rate nearly flat and unit-size. Because system bandwidth is directly related to system response speed, weights on actuator rates can be used to modulate final bandwidth of the closed-loop system.

The role of weights for commands, disturbances and noise is basically the opposite of the role of weights for error. Rather than taking unscaled signals and normalizing them, these weights take flat unit-size signals and scale them to produce a specified range of magnitudes and frequencies over which the design must insure good performance.

Uncertainty weights transform the normalized unit-size perturbations into perturbations whose magnitudes and frequency content match uncertainty levels in the design model. For most aircraft flight control design work, models are reasonably well known out to the short-period and dutch-roll frequencies. Beyond that, they become progressively less reliable. Uncertainty weights must reflect this.

Given a normalized interconnection structure, the design problem is to find a stabilizing compensator K that makes the maximum singular value of the closed-loop frequency-response matrix from d to e less than unity for all possible unit size perturbations Δ with the defined block structure.

$$\Delta = \begin{bmatrix} \Delta_{input} & 0 & 0 \\ 0 & \Delta_{output} & 0 \\ 0 & 0 & \Delta_{param} \end{bmatrix}$$

H_∞ Synthesis [9]:

This step results in a control compensator K. The closed-loop system is then formed by connecting the sensors and actuators of P to K to produce the closed-loop interconnection structure, M.

The problem of finding compensators that make the structured singular value of M less than unity can be solved by repeated H_∞ solutions alternated with rescaling

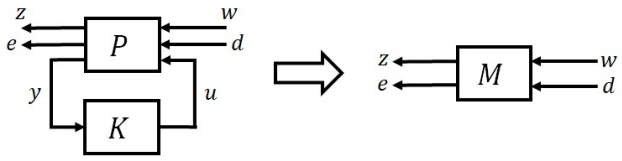

Figure 2.13: Closed-loop interconnection

of the signal sets. This theory provides compensators that minimize the H_∞ norm of M (i.e., minimize the maximum singular value 'γ' rather than the structured singular value 'μ', of M over frequency). After the γ search is completed, the final compensator K is supplied in state-space form. This controller system then used to construct the closed-loop system M and to compute its frequency response matrix over a frequency range of interest for subsequent μ-analysis.

μ-Analysis and D-Scales [9]:

This step involves calculation of the structured singular value, $\mu(M)$ and its associated frequency dependent D-scales. The structured singular value provides a measure of how close the compensator is to meeting its robust performance goals. Here, a complete point-by-point μ-analysis of closed-loop frequency response $M(jw)$ is performed. This involves calculating structured singular values (μ) at each frequency point and comparing those values against unity.

$$\Delta_M = \begin{bmatrix} \Delta_{input} & 0 & 0 & 0 \\ 0 & \Delta_{output} & 0 & 0 \\ 0 & 0 & \Delta_{param} & 0 \\ 0 & 0 & 0 & \Delta_p \end{bmatrix}$$

The block Δ_p is the so-called fictitious perturbation representing the performance requirements, with input/output signals e and d. For this uncertainty structure, the condition, $\mu(M) < 1 \ ; \ \forall\omega$ guarantees that closed-loop system M remains

stable when Δ is connected from z to d and Δ_p is connected from e to d, simultaneously. The latter condition ensures that performance is robust.

The exact value of μ at each frequency is not, in general, easy to calculate, so, upper and lower bounds are used to bracket the true value. The upper bound, in particular, is based on a computationally tractable search over the class of scaling matrices D that commute with perturbation Δ_M.

$$D = \begin{bmatrix} d_{in}I_{in} & 0 & 0 & 0 \\ 0 & D_{output} & 0 & 0 \\ 0 & 0 & D_{param} & 0 \\ 0 & 0 & 0 & I_p \end{bmatrix}$$

The specific D's that achieve the infimum in this equation are called D-scales.

$$\mu(M) \leq \inf_D \overline{\sigma}(DMD^{-1})$$

$D - K$ Iteration [9]:

Once the μ-analysis calculations are complete and if the condition $\mu(M) < 1$ is satisfied throughout the selected frequency band, the current H_∞ compensator, K, meets all robust performance goals. If not, another design iteration is performed until either an approximately flat μ function across frequency is achieved and/or current D-scales differ insignificantly from the previous iteration.

If another iteration is appropriate, it differs from the current one only in the sense that a modified optimization problem, which is a rescaled version of the original one using the current D-scales as scaling factors, is solved in the H_∞ step.

Changing Weights [9]:

If D and K have converged, but the compensator does not meet its goals (i.e., $\mu(M) > 1$), then the weights must be changed, trading off some goals against others,

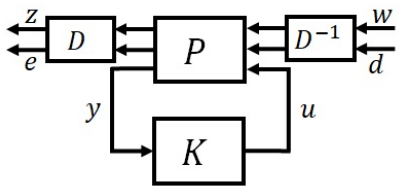

Figure 2.14: Feedback system with D-Scales

and then $D - K$ iteration is repeated. M-analysis can be used to determine which input/output paths are driving the problem.

2.4 Gain Scheduling

Traditionally, satisfactory performance across the flight envelope can be attained by scheduling gains of local autopilot controllers on a slow variable to yield a global controller [30]. While gain scheduling is well established as a practical tool for the design of controllers for nonlinear plants and a number of ad-hoc approaches [31] to interpolation have been reported, very few publications on detailed analyses of gain scheduled multivariable controllers appear in the literature. Some reported ad-hoc approaches include [11][17]:

- linearly interpolating poles, zeros, and gains of controller transfer functions;
- linearly interpolating the solutions of Riccati equations in H_∞ controller synthesis;
- linearly interpolating the balanced controller realizations of state-space matrices;
- implementing the controllers in parallel and linearly interpolating their output signals.

All these methods usually give satisfactory results in some particular applications, however, can generate non stabilizing controllers in other cases [11][19][32]. In

addition to ad-hoc interpolation, some theoretically justified methods have also been presented, such as stability preserving interpolation for feedback gains [11] and linear parameter varying (LPV) control design methods [11][31][32]. Various synthesis methods for gain scheduling using linear fractional transformations (LFTs) [33][34][35] have also been developed. The idea behind this is to let the controller have access to some of the uncertainties of the system to be controlled [34]. Here, the interpolation problem is addressed implicitly in that the controllers are also parameter varying [11].

The complexity of the gain scheduling is highly dependent upon the structure of the LTI controllers at each fixed operating point. With classical control techniques, the structure of the LTI controller dynamics may be fixed for each operating point with only few gains varying with changing parameters. However, for multivariable state space design procedures each LTI controller may have a different state order and feedback topology. In this latter case, the gain scheduling implementation considerations may become a serious drawback to the use of these multivariable design approaches [17][36].

In this study two different scheduling approaches will be investigated in detail. But before that, a prelude on uncertain model representation for a collection of LTI systems will be given.

Uncertain Model Representation:

In aerospace applications, it is common practice to represent plant dynamics as a collection of LTI systems, where, each one of the systems correspond to aircraft dynamics in the neighborhood of a specific operating point. Following this approach, we define the set of operating points as $\Lambda = \{\rho_1, \ldots, \rho_n\}$, where ρ_i represents a vector of parameters to be used for scheduling. Then, LTI system matrices can be

parameterized by ρ_i, such that: $\hat{A}_i = \hat{A}(\rho_i)$, $\hat{B}_i = \hat{B}(\rho_i)$, $\hat{C}_i = \hat{C}(\rho_i)$ and $\hat{D}_i = \hat{D}(\rho_i)$ The nonlinear plant dynamics for each operating point becomes:

$$\begin{aligned} \dot{x} &= \hat{A}_i x + \hat{B}_i u \\ y &= \hat{C}_i x + \hat{D}_i u \end{aligned} \quad for\ each\ \rho_i \in \Lambda$$

We define the matrix polynomials:

$$P_A(\delta) = \sum_{i=0}^{n-1} A_i \delta^i;\ P_B(\delta) = \sum_{i=0}^{n-1} B_i \delta^i;\ P_C(\delta) = \sum_{i=0}^{n-1} C_i \delta^i;\ P_D(\delta) = \sum_{i=0}^{n-1} D_i \delta^i$$

with proper transformation from $\hat{A}_i, \hat{B}_i, \hat{C}_i, \hat{D}_i$ to A_i, B_i, C_i, D_i, where, $\delta \in [-1,\ 1]$, such that at the points $\{\delta_j\}_{j=1}^{n}$ we have $P_A(\delta_j) = A_j$; $P_B(\delta_j) = B_j$; $P_C(\delta_j) = C_j$ and $P_D(\delta_j) = D_j$; $j = 1, \ldots, n$.

At this point it is possible to define the matrix M, which represents the system dynamics using matrices A_i, B_i, C_i, D_i such that:

$$M = \begin{bmatrix} A_0 & B_0 & A_1 & B_1 & \cdots & A_n & B_n \\ C_0 & D_0 & C_1 & D_1 & \cdots & C_n & D_n \\ I & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & I & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & I & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & I & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}$$

with δI as the structured uncertainty associated with the gain scheduled plant. Now it is possible to treat the system in a standard LFT form as displayed in Figure 2.15. A quiet similar approach for obtaining LPV models can be found in Ref. [18].

Synthesis using Simplified LPV Model [10]:

Once the plant is scheduled using the uncertainty loop as explained in the previous chapter, the very same uncertain parameters can also be used for controller

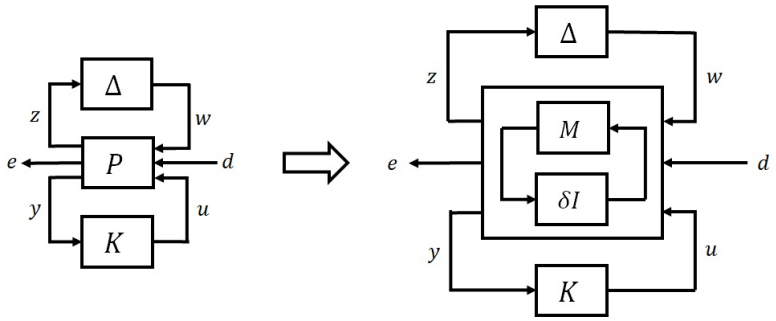

Figure 2.15: LFT form with scheduled plant

synthesis. For a linear parameter varying (LPV) plant, which is generated using a number of LTI systems, a single controller can be synthesized by $D-K$ iteration provided that the controller input signals is arranged in a proper fashion in interconnection structure.

For a plant composed of two LTI systems, the controller is also desired to act as a twofold system compatible with the plant. For this, the input channels of the controller is modified in order that the two input channels are multiplied with uncertain gains k_1 and k_2 to provide two linearly interpolated signals, e.g.:

$$for -1 \leq \delta \leq 1;\ k_1 = (1-\delta)/2;\ k_2 = (1+\delta)/2$$

A graphical representation for linear interpolated simplified LPV model is given in Figure 2.16. A detailed discussion on this method can be found in Ref. [10].

The method can be extended to higher order LPV plants. In Figure 2.17, a second order LPV model composed of three LTI systems is taken into consideration. In this case, three input channels are multiplied with uncertain gains k_1, k_2 and k_3 where;

$$for\ \ -1 \leq \delta \leq 1;\ k_1 = \delta(\delta-1)/2;\ k_2 = (1-\delta^2);\ k_3 = \delta(\delta+1)/2$$

Once the system interconnection is built, standard μ-synthesis steps can be followed for controller synthesis. The resulting LTI controller will have higher di-

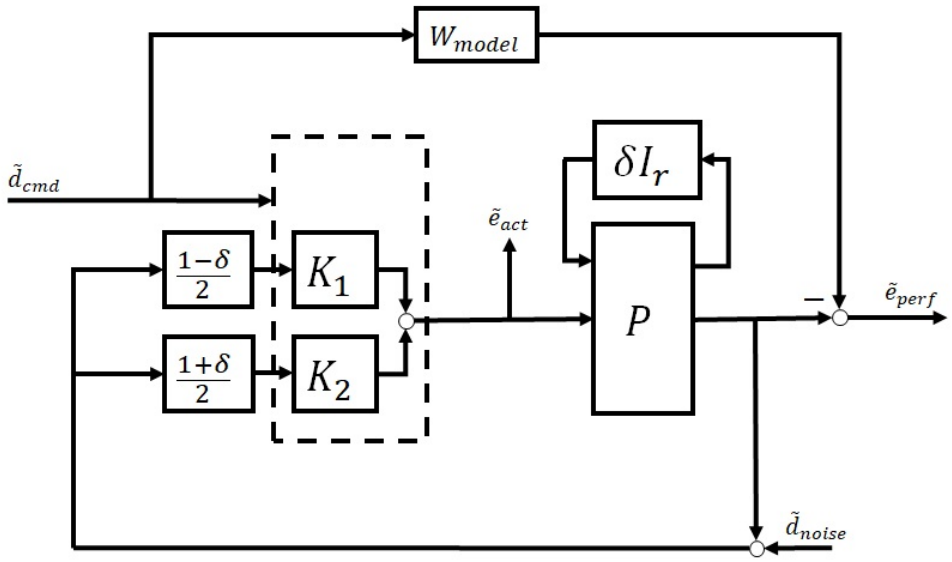

Figure 2.16: 1st order simplified LPV interconnection

mensions due to increased number of inputs and most likely will be of higher order, compared to LTI controllers designed for single operating points.

Stability Preserving Interpolation [11][37]:

Synthesis of gain scheduled controllers for nonlinear plants often requires that a parameter-varying controller be generated from a finite set of linear time-invariant controllers. The scheduling variable can be a function of the state, input, and exogenous signal. For the generic LTI system representation of the plant Σ:

$$\begin{bmatrix} \dot{x} \\ e \\ y \end{bmatrix} = \begin{bmatrix} F & G_1 & G_2 \\ H_1 & J_{11} & J_{12} \\ H_2 & J_{21} & J_{22} \end{bmatrix} \begin{bmatrix} x \\ d \\ u \end{bmatrix} \tag{2.4}$$

if the components of this system can be formulated as matrix functions of a parameter ρ, then the plant $\Sigma(\rho)$ is also a function of ρ. So the system equations become:

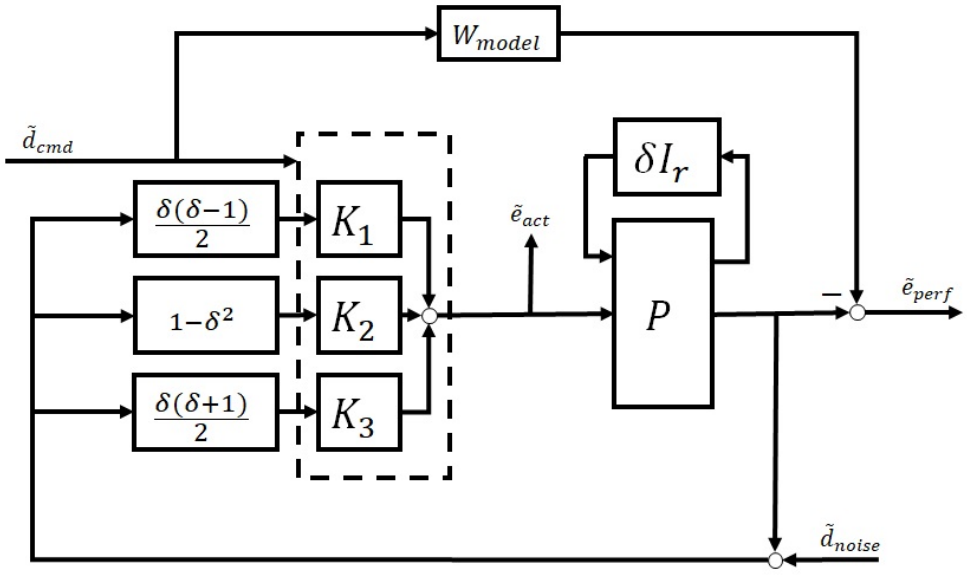

Figure 2.17: 2nd order simplified LPV interconnection

$$\Sigma(\rho) = \begin{cases} \dot{x}(t) = F(\rho)x(t) + G_1(\rho)w(t) + G_2(\rho)u(t) \\ e(t) = H_1(\rho)x(t) + J_{11}(\rho)w(t) + J_{12}(\rho)u(t) \\ y(t) = H_2(\rho)x(t) + J_{21}(\rho)w(t) \end{cases} \tag{2.5}$$

Assume that controllers in LTI form, Λ_i, have been synthesized for selected n operating points which are also parameterized by ρ, e.g.: $\Lambda_i = \Lambda(\rho_i), for\ i = 1, \ldots, n$. The equations of the LTI controllers to be interpolated can be written as:

$$\Lambda_i = \begin{cases} \dot{z}(t) = A_i z(t) + B_i y(t) \\ u(t) = C_i z(t) + D_i y(t) \end{cases} \tag{2.6}$$

The target in stability preserving interpolation is to generate a parametric controller $\Lambda(\rho)$ which can be used in conjunction with the parametric plant $\Sigma(\rho)$, and the closed-loop system is guaranteed to be stable for any ρ. For this to be possible, stability covering condition [11] should be satisfied for all Λ_i and $\Sigma(\rho)$. The steps followed in this process is briefly displayed in Figure 2.18.

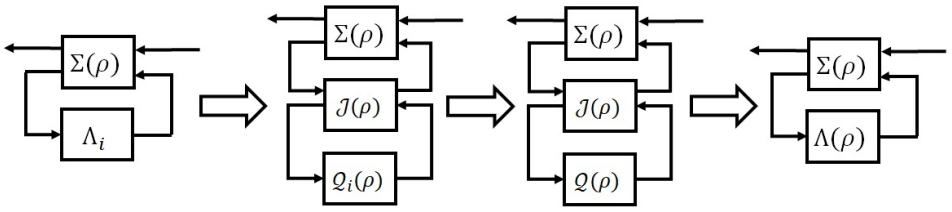

Figure 2.18: Steps of stability preserving interpolation

The process is simply based on an intermediate layer $\mathcal{J}(\rho)$, and parameterization of pointwise controllers Λ_i by use of this new layer. In the following, a linear interpolation between two operating points ρ_1 and ρ_2 is assumed with $\rho_1 = 0$, $\rho_2 = 1$ and $\rho \in (0, 1)$. Defining the closed-loop system using lower LFT of the plant $\Sigma(\rho)$ and the controller Λ_i:

$$\mathcal{F}_l(\Sigma(\rho), \Lambda_i) = \begin{cases} \dot{x}(t) = \hat{A}_i(\rho)x(t) + \hat{B}_i(\rho)u(t) \\ y(t) = \hat{C}_i(\rho)x(t) + \hat{D}_i(\rho)u(t) \end{cases} \tag{2.7}$$

Lemma: Given the LTI controller Λ_i and the LPV system $\Sigma(\rho)$ with ρ constant, suppose there exists K and L such that $F(\rho)+G_2(\rho)K$ and $F(\rho)+LH_2(\rho)$ are stable. Then Λ_i has the same transfer function as $\mathcal{F}_l(\mathcal{J}(\rho), \mathcal{Q}_i(\rho))$ where;

$$\mathcal{J}(\rho) = \begin{cases} \dot{x}(t) = (F(\rho) + G_2(\rho)K + LH_2(\rho))x(t) - Lw(t) + G_2(\rho)u(t) \\ e(t) = K(\rho)x(t) + u(t) \\ y(t) = H_2(\rho)x(t) - w(t) \end{cases} \tag{2.8}$$

$$\mathcal{Q}_i(\rho) = \begin{cases} \dot{z}(t) = \tilde{A}_i(\rho)z(t) + \tilde{B}_i(\rho)y(t) \\ u(t) = \tilde{C}_i(\rho)z(t) \end{cases} \tag{2.9}$$

with;

$$\tilde{A}_i(\rho) = \begin{bmatrix} F(\rho) + G_2(\rho)D_iH_2(\rho) & G_2(\rho)C_i \\ B_iH_2(\rho) & A_i \end{bmatrix} \tag{2.10}$$

$$\tilde{B}_i(\rho) = \begin{bmatrix} L(\rho) - G_2(\rho)D_i \\ B_i \end{bmatrix} \tag{2.11}$$

$$\tilde{C}_i(\rho) = \begin{bmatrix} -K(\rho) + D_iH_2(\rho) & C_i \end{bmatrix} \tag{2.12}$$

$$\tilde{D}_i(\rho) = -D_i \tag{2.13}$$

where $\tilde{A}_1(\rho)$ and $\tilde{A}_2(\rho)$ both stable for $\rho \in (0,1)$. Proof of this Lemma can be found in Ref. [37]. Let $W_1(\rho)$ and $W_2(\rho)$ be symmetric positive definite matrices such that:

$$\tilde{A}_1^T(\rho)W_1(\rho) + W_1(\rho)\tilde{A}_1(\rho) < -I for\ \rho \in [a,1) \tag{2.14}$$

$$\tilde{A}_2^T(\rho)W_2(\rho) + W_2(\rho)\tilde{A}_2(\rho) < -I for\ \rho \in (0,b] \tag{2.15}$$

This yields,

$$(1-\rho)\left(\tilde{A}_1^T(0)W_1(0) + W_1(0)\tilde{A}_1(0)\right) + \rho\left(\tilde{A}_2^T(1)W_2(1) + W_2(1)\tilde{A}_2(1)\right) < -I;\ for\ \rho \in (0,1)$$

Defining:

$$\tilde{W}(\rho) = (1-\rho)W_1(0) + \rho W_2(1) \tag{2.16}$$

$$\tilde{A}_W(\rho) = \tilde{W}^{-1}(\rho)((1-\rho)W_1(0)\tilde{A}_1(0) + \rho W_2(1)\tilde{A}_2(1))$$

is stable for each $\rho \in (0,1)$.

Defining:

$$\hat{W}(\rho) = \begin{cases} W_1(\rho), \ \rho \in [a,0] \\ \tilde{W}(\rho), \ \rho \in (0,1) \\ W_2(\rho), \ \rho \in [1,b] \end{cases} \tag{2.17}$$

$$\tilde{A}(\rho) = \begin{cases} \tilde{A}_1(\rho), \ \rho \in [a,0] \\ \tilde{A}_W(\rho), \ \rho \in (0,1) \\ \tilde{A}_2(\rho), \ \rho \in [1,b] \end{cases} \tag{2.18}$$

$$\tilde{B}(\rho) = \begin{cases} \tilde{B}_1(\rho), \ \rho \in [a,0] \\ (1-\rho)\tilde{B}_1(0) + \rho\tilde{B}_2(1), \ \rho \in (0,1) \\ \tilde{B}_2(\rho), \ \rho \in [1,b] \end{cases} \tag{2.19}$$

$$\tilde{C}(\rho) = \begin{cases} \tilde{C}_1(\rho), \ \rho \in [a,0] \\ (1-\rho)\tilde{C}_1(0) + \rho\tilde{C}_2(1), \ \rho \in (0,1) \\ \tilde{C}_2(\rho), \ \rho \in [1,b] \end{cases} \tag{2.20}$$

$$\tilde{D}(\rho) = \begin{cases} \tilde{D}_1(\rho), \ \rho \in [a,0] \\ (1-\rho)\tilde{D}_1(0) + \rho\tilde{D}_2(1), \ \rho \in (0,1) \\ \tilde{D}_2(\rho), \ \rho \in [1,b] \end{cases} \tag{2.21}$$

then

$$\mathcal{Q}(\rho) = \begin{cases} \dot{z}(t) = \tilde{A}(\rho)z(t) + \tilde{B}(\rho)y(t) \\ u(t) = \tilde{C}(\rho)z(t) + \tilde{D}(\rho)y(t) \end{cases} \tag{2.22}$$

is a stable system for each $\rho \in [a,b]$; and $\Lambda(\rho) = \mathcal{F}_l(\mathcal{J}(\rho), \mathcal{Q}(\rho)$ stabilizes $\Sigma(\rho)$ for each $\rho \in [a,b]$. For $W_i(\rho)$ and $W_i^{-1}(\rho)$, partitioned as follows:

$$W_i(\rho) = \begin{bmatrix} S_i(\rho) & N_i(\rho) \\ N_i^T(\rho) & P_i(\rho) \end{bmatrix}; \ W_i^{-1}(\rho) = \begin{bmatrix} R_i(\rho) & M_i(\rho) \\ M_i^T(\rho) & Q_i(\rho) \end{bmatrix} \tag{2.23}$$

Defining:

$$L_i(\rho) = G_2(\rho)D_i - S_i^{-1}(\rho)N_i(\rho)B_i \tag{2.24}$$

$$K_i(\rho) = D_iH_2(\rho) + C_iM_i^T(\rho)R_i^{-1}(\rho) \tag{2.25}$$

and;

$$L(\rho) = \begin{cases} L_1(\rho),\ \rho \in [a,0) \\ S^{-1}(\rho)((1-\rho)S_1(\rho)L_1(\rho) + \rho S_2(\rho)L_2(\rho)),\ \rho \in [0,1] \\ L_2(\rho),\ \rho \in (0,b] \end{cases} \tag{2.26}$$

$$K(\rho) = \begin{cases} K_1(\rho),\ \rho \in [a,0) \\ ((1-\rho)K_1(\rho)R_1(\rho) + \rho K_2(\rho)R_2(\rho))R^{-1}(\rho),\ \rho \in [0,1] \\ K_2(\rho),\ \rho \in (0,b] \end{cases} \tag{2.27}$$

where

$$S(\rho) = (1-\rho)S_1(\rho) + \rho S_2(\rho) \tag{2.28}$$

$$R(\rho) = (1-\rho)R_1(\rho) + \rho R_2(\rho) \tag{2.29}$$

A more detailed explanation of the method can be found in Ref. [11] and Ref. [37]. A step by step implementation will be discussed in Chapter 4.

CHAPTER 3

CONTROL ALLOCATION

3.1 Background

The control algorithm hierarchy for over actuated systems with a redundant set of effectors commonly includes three levels. First, a high level control algorithm commands a vector of virtual control efforts (i.e. forces and moments) in order to meet the overall control objectives. Second, a control allocation algorithm coordinates the different effectors such that they together produce the desired virtual control efforts, if possible. Third, low-level control algorithms may be used to control each individual effector via its actuators [38]. So to say, control allocation deals with the problem of distributing a given control demand among an available set of actuators [39]. Control allocation offers the advantage of a modular design where the high-level control algorithm can be designed without detailed knowledge about the effectors and actuators. Important issues such as input saturation and rate constraints, actuator and effector fault tolerance, and meeting secondary objectives such as power efficiency are handled within the control allocation algorithm [38].

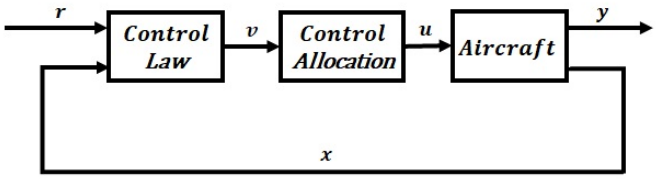

Figure 3.1: Control hierarchy for over-actuated systems

A typical control hierarchy diagram for over actuated systems is shown in Figure 3.1. Here;

r shows the reference commands for control law,

y shows the measurable output parameters,

x shows the controller feedback signals,

v shows the virtual control command generated by the control laws, and

u shows the true control input to the control effectors.

The control allocation problem is defined as finding an exact or approximate solution for a system of linear equations subject to constraints for redundant actuator control variable commands. The constraints usually arise from actuation rate and position limits [40]. This is often posed as a constrained least squares problem to incorporate the actuator position and rate limits. Most proposed methods for real-time implementation only deliver approximate, and sometimes unreliable solutions [13]. An axis priority weighting can be introduced when the equations cannot be solved exactly because of the constraints. Likewise, an actuator command preference weighting and preferred values can be introduced to uniquely solve the equations where there are more unknowns than equations [40]. The cost functions can be optimized in real time, or alternatively, pre-computed effector solutions that are optimal for specific conditions can be used as preferred solutions [12]. Concerns about maximum effector-induced loads, as well as repeated load applications causing fatigue failures may be decisive in determination of this cost function. When some effectors are slower than others, it may be desirable to use them less often by biasing the weighting matrix used in the cost function [12]. A variety of approaches for on-line control allocation have been reviewed in Refs. [38][41][42].

3.2 Actuator Constraints

Linear control design is typically carried out without regard to what the system behavior will be when an actuator is saturated or rate-limited. The gains of the control loop are selected so that actuators are sufficiently utilized to meet normal closed-loop performance requirements. Selecting the gains to avoid saturating the actuators when large disturbances or commands act on the closed-loop would give poor performance under normal operating conditions. Multivariable systems present much more of a problem when actuators saturate because the loop-gain has both magnitude and direction which are both affected by saturating actuators. The loss in directionality can mean loss of decoupling between the controlled outputs [17]. A control allocation scheme, in this context, eases handling of saturated actuators in an over actuated system. As the control law design relies on the limits of virtual controls, not the individual actuators, the position or rate limit requirements of the system can simply be embedded in this virtual controls, having a much reduced impact on controller performance. So, in case of a saturation, the virtual commands generated by the control laws can be realized by some combination of control surfaces different than the unsaturated case.

Actuator failures have damaging effect on the performance of control systems, leading to undesired system behavior or even instability. For system safety and reliability, the compensation of actuator failures is of both theoretical and practical significance. Several design approaches have been studied for this purpose, such as: multiple-model designs (switching and tuning designs), fault-diagnosis-based designs, robust control designs, and adaptive designs [43].

In multiple-model designs employed for reconfigurable flight control, it is assumed that the plant belongs to a set of plant models with corresponding set of controllers. When some actuators fail, a switching mechanism will find the best

matched model and switch to the appropriate controller. For fault-diagnosis-based designs, in addition to a controller, there is an FDI (fault detection and isolation) unit. When actuator failures appear, a reconfigurable controller will change its structure and parameters according to the fault diagnosis result to accommodate actuator failures. In adaptive designs, as the controller is implemented with unknown parameters, these parameters are updated on-line by some adaptive laws in the case of actuator failures. Robust control designs for systems with actuator failures use robust control techniques to accommodate some classes of failures by treating them as uncertainties. Most existing control designs for systems with actuator failures either are based on some knowledge of actuator failures or depend on some extra detection techniques [43].

For aircraft with redundant control effectors, it is much easier to overcome the effects of an actuator failure by implementing a control allocation based control method. A control allocation algorithm will simply use whatever controls are functional to attempt to satisfy the demands [12].

3.3 Control Allocation Problem

In most aerospace applications, control effectors are used to change the aerodynamic moments - and less frequently the aerodynamic forces - acting on the body. As discussed above, all control effectors have certain position limits, usually specified as positive and negative deflections from a specified position that is taken as zero deflection, and limits on the rate they can travel. Normally the actuator dynamics are very fast compared to the responses of the airplane [12]. The basic control allocation problem can be formulated as:

For given $B, \upsilon, \underline{u}, \overline{u}$, find u such that:

$$Bu = \upsilon, \quad \underline{u} \leq u \leq \overline{u}$$

where;

υ is the vector of desired moments or virtual commands and $\upsilon \in \mathbb{R}^n$. This vector is not always defined as actual moments, however this does not affect the geometry of the problem or methods of its solution.

u is vector of control effectors, $u \in \mathbb{R}^m$, where $m > n$ and B is the control effectiveness matrix, $B \in \mathbb{R}^{n \times m}$

The set of equations is under-determined and mathematically has an infinite number of solutions. When control limitations are introduced the equations may have no solutions. If there is an admissible control vector for a particular moment, then that moment is said to be attainable [12]. There are several published methods for handling the control allocation problem.

Ganging [12]:

Individual control effectors can be interconnected so that the movement of one depends on that of another. This interconnection reduces the number of independent solutions in control allocation problem, e.g. if $m - n$ interconnections are applied, the problem becomes determined and can be solved by simply inverting B.

Generalized Inverses [12]:

A generalized inverse is taken to be a matrix P that, without regard to the limits on control effectors, finds:

$$u_p = P\upsilon$$

such that $Bu_p = \upsilon$, and $BP = I_n$. All such generalized inverses may be represented as:

$$P = N(BN)^{-1} \tag{3.1}$$

where N is arbitrary with $N \in \mathbb{R}^{m \times n}$ and BN is invertible.

Daisy Chaining [12]:

The method works by dividing the controls into two or more groups based on the frequency of use of effectors. The first group, typically the more conventional control effectors, are to be used most frequently. The subsequent groupings are to be used only if the set of previous groupings is unable to satisfy the moment demand.

Direct Allocation [12][38][44]:

Direct Allocation is a constrained control allocation approach based on some scaling of the unconstrained optimal control allocation, such that the resulting control allocation is projected onto the boundary of the set of attainable moments (AMS). In simple terms, AMS is defined as the domain of all attainable virtual controls (moments) using every admissible control (effector) for a given control effectiveness matrix B. The Direct Allocation method finds a one-to-one relation between an actuator command and the controller (virtual) command using the boundary of the AMS. The information from AMS boundary can be converted to a set of look-up tables, so that, the computational burden is reduced considerably, which makes it suitable for on-line applications.

Optimization Based Allocation [38]:

The presence of physical constraints, operational constraints and secondary objectives makes optimization-based design a powerful approach. The simplest formulations allow explicit solutions to be computed using numerical linear algebra, while the more challenging formulations with complex constraints and objectives call for iterative numerical optimization procedures.

A powerful approach is to explicitly minimize the cost J defined as the n-norm of the weighted error between the allocated control input, u, and the desired one, u_p, i.e.: find $\min_u J$ where $J = \|W(u - u_p)\|_n$ with constraints $\underline{u} \leq u \leq \overline{u}$.

With the cost function defined using either 1-norm or ∞-norm, this resulting problem is a linear program (LP) that can be solved using iterative numerical LP algorithms. With the common choice of 2-norm, the control allocation problems leads to a quadratic program (QP) that can be solved using numerical QP methods.

Dynamic Allocation [39]:

Dynamic control allocation is a method proposed for distributing the control effect in the frequency domain and thus using the redundancy to have different actuators operate in different parts of the frequency spectrum. This requires the mapping from v to u to depend also on earlier values of u and v. A frequency-dependent control distribution can be designed to account for different actuator bandwidths by extending regular control allocation by also penalizing the actuator rates.

In this study two different control allocation approaches will be investigated in more detail. First, a simple method, Cascaded Generalized Inverses, then a more complicated, Weighted Least Squares based algorithm will be implemented. Both will be tested in an integrated simulation using a number of scenarios with several maneuvers and malfunctions and their performance will be compared.

Cascaded Generalized Inverse (CGI) [12]:

In its simplest form, CGI method uses a single generalized inverse exclusively until one or more of the control effectors is commanded to a position beyond saturation. When this happens, the saturated controls are left at whatever limit they were when saturated and removed from the problem. The control allocation process is repeated with the reduced problem until desired moments are satisfied or no control effectors are left to allocate.

Weighted Least Squares (WLS) [13]:

In a constrained optimization framework, the basic control allocation problem can be stated as finding the control that minimizes the weighted error between the

preferred and actual solution which is in a set of feasible solutions that minimize the weighted deviation from the virtual control, i.e.:

$$\min_{u \in \mathrm{K}} \|W_u(u - u_p)\|$$

where

$$\mathrm{K} = \min_{\underline{u} \le u \le \overline{u}} \|W_v(Bu - v)\|$$

An approximate way to reformulate the problem is to merge the two optimization criteria into one by summation and form a weighted least squares problem:

$$\min_{\underline{u} \le u \le \overline{u}} \left(\|W_u \left(u - u_p\right) \|^2 + \gamma \|W_v(Bu - v)\|^2 \right) \tag{3.2}$$

The most frequently encountered approaches for solving this problem are based on constrained quadratic programming. Active set methods are used in many of today's solvers for constrained quadratic programming, and can be shown to find the optimal solution in a finite number of iterations. We will implement an active set algorithm based solver for this weighted least squares problem.

CHAPTER 4

CASE STUDY

4.1 Background

A tailless configuration presents challenges in various disciplines, stability and control being the most prominent [4]. Major discrepancies related to this configuration are: light short-period damping, sensitive pitch control, inefficient use of high-lift devices, difficulty in takeoff rotation, lack of directional stability, poor directional control and strong coupling between control effectors [6][7]. Some of the proposed solutions for these were using wings with reflexed airfoil shapes, selecting proper combination of sweepback, taper and twist, inserting flow separators on wings, ganging controls in a proper way, etc. [6]. These, by no doubt, helped improving the designs, but frequently, they also created new problems or caused degradation in flight performance, so tailless design is understood not to be a simple task.

Today, one may be puzzled by the fact that we see so few tailless aircraft. This is not simply a reflection of aircraft manufacturers' conservatism, but an indication of the fact that tails are an efficient means of satisfying the requirements for trim and control. It is very easy to design a very bad tailless aircraft while conventional design wisdom leads to a quite good conventional design [45].

Available literature on tailless configurations may be categorized into three major groups: aircraft design oriented, control system design oriented and aerodynamic analysis oriented works. A selection from the literature, which are considered to be interesting by the author, will be presented here:

Ref. [46] is one of the earliest attempts to compile and describe all the special problems of tailless aircraft with an engineering perspective. The main focus is on design considerations and flight characteristics. Only the aerodynamic aspects of tailless aircraft are considered with a special attention to Horten gliders. Ref. [47] intends to assess the handling qualities of a tailless configuration, using piloted-handling trials in a moving-base simulator and to determine the effect of simulator motion on handling-quality ratings. Ref. [6] is concerned with identifying and then evaluating the flight dynamics, stability, flight controls and handling qualities of a generic Blended Wing Body (BWB) large transport aircraft concept. The possibility to use the conventional flying and handling qualities criteria and requirements for FCS design for BWB configuration is investigated. Ref. [48] deals with a very special kind of tailless aircraft: hang gliders. Some analytical methods and tools are developed for a quantitative understanding of hang glider aerodynamics. Effects of flexibility is investigated by combined aerodynamic and structural analysis. Wind tunnel tests are performed using elastically scaled model, for verification. Ref. [3] investigates the basic problems related to the development of advanced large-capacity aircraft of a flying-wing (FW) configuration from a TsAGI perspective. Ref. [1] discusses the proper incorporation of yaw control power effects - their magnitude and bandwidth - within the iterative conceptual design process, which is considered to be critical to a successful aircraft design from a dynamics standpoint. It also covers other factors that must be accounted for in designing tailless aircraft. Ref. [49] presents a numerical approach for determining optimum lift distribution for tailless aircraft using multiple trailing-edge flaps such that drag is minimized with a pitching-moment constraint.

The purpose of Ref. [50] is to determine the tailless aircraft performance improvements gained from relaxed static stability, to quantify this potential in terms of range-payload improvements, and to identify other possible operational and handling

benefits or problems. Two configurations are studied: a high aspect ratio, short-chord wing proposed as a high-altitude long endurance class UAV, and a lower aspect ratio, high volume wing proposed for a manned long-range reconnaissance mission. Ref. [42] presents a direct adaptive reconfigurable flight control approach and demonstrates its effectiveness via an application to an advanced tailless fighter aircraft. The reconfigurable control law is based on a dynamic inversion controller in an explicit model following architecture and an on-line neural network is used to adaptively regulate the error between the desired response model and the actual vehicle response. Ref. [41] investigates an eigenstructure assignment based flight control system for a wind tunnel model of a tailless aircraft and proposes a novel method capable of maintaining stability, desired performance, and disk gain/phase margins under control effector failures. In Ref. [4], a reconfigurable sliding mode flight controller is designed for a tailless jet fighter. The controller achieves robust, high accuracy command angle tracking both before and after damage. An optimal control allocation algorithm is employed using nominal mathematical model of an aircraft. Ref. [51] describes a neural network based direct adaptive control approach to the problem of reconfigurable flight control. A tailless fighter aircraft configuration with multiple and redundant control actuators is used to illustrate the level to which handling qualities can be maintained in the presence of large scale failures in the actuation channels.

Ref. [52] is focused on the use of active flow control on a tailless aircraft model. Wind tunnel experiments are performed to test the effectivity of sweeping jet flow control for the longitudinal stability problems at higher incidence angles. Ref. [53] intends to use computational fluid dynamics to study the tumbling characteristics of a tailless aircraft and then determine dynamic stability information from the simulations. The effects of various parameters are investigated and lumped pitch damping derivatives are determined from the simulations. Ref. [54] evaluates the stability

and control characteristics of the tailless aircraft using a hierarchy of variable fidelity aerodynamic analysis methods, including linear potential flow, Euler and Reynolds averaged Navier-Stokes solvers as low, medium and high fidelity analysis, respectively. Validation of variable fidelity computation results was made with wind tunnel test data. Ref. [55] investigates the dynamics and control of a tailless aircraft model in wind tunnel. Both steady and unsteady aerodynamic characteristics are studied using different test techniques. A method to control the aircraft for moderate sideslip angles is derived, and a recovery strategy for yaw departure is investigated.

In the following, a case study on the application of a modern control design technique for a tailless UAV, RQ-3 Darkstar, is presented in detail.

RQ-3 Darkstar UAV [56]:

RQ-3 DarkStar is an unmanned aircraft designed for high-altitude missions and incorporated stealth aircraft technology to make it difficult to detect, which allowed it to operate within heavily defended airspace. It is a tailless design with high aspect ratio wings and powered by a single air breathing turbofan engine (claimed to be a Williams-Rolls-Royce FJ44-1A).

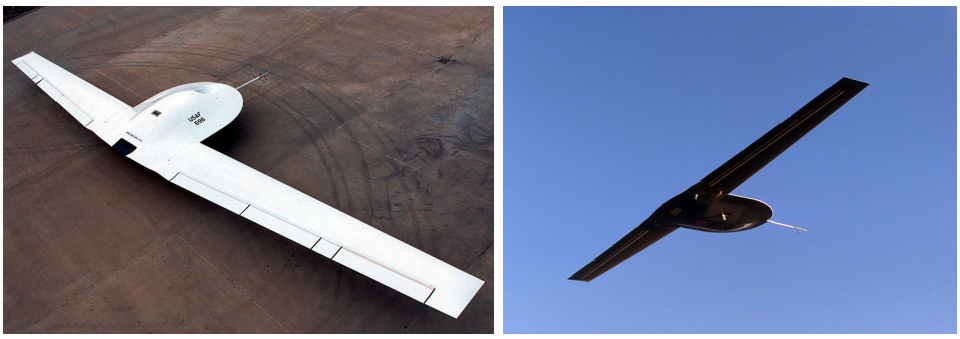

Figure 4.1: RQ-3 Darkstar UAV

The first prototype made its first flight on March, 1996. Its second flight ended in a crash shortly after takeoff. A modified, more stable design (the RQ-3A) first flew on June, 1998, and made a total of five flights before the program was canceled.

General characteristics of the vehicle are summarized in Table 4.1.

Table 4.1: RQ-3 basic characteristics [56]

Characteristic	Imperial	Metric
Length	15 ft	4.57 m
Wingspan	69 ft	21.03 m
Height	3.5 ft	1.07 m
Empty weight	4360 lb	1980 kg
Gross weight	8500 lb	3860 kg
Thrust	1900 lbf	8.5 kN
Cruise speed	250 kts	464 km/h
Range	500 nmi	925 km
Service ceiling	45000 ft	13700 m

Since very limited public data is available for the aircraft, some design parameters such as wing twist, incidence angle and wing airfoil have been selected by the author in order to introduce a reasonable design. Instead of the existing control effectors of RQ-3, in this study, a different set of redundant control effectors is assumed to be implemented.

4.2 Nonlinear Model

A nonlinear model for RQ-3 has been developed using MATLAB-Simulink®, to be used in tasks such as generation of LTI plants for various operating points and validation of the controllers and control allocation algorithms by simulation. The aircraft model consists of several sub-models: Atmosphere, flight controls, aerodynamics, engine, equations of motion, and mass properties [57]. All sub-models are

connected to a main data bus, which manages the communication between each other and input/outputs of the aircraft model. A top level overview of the model is shown in Figure 4.2.

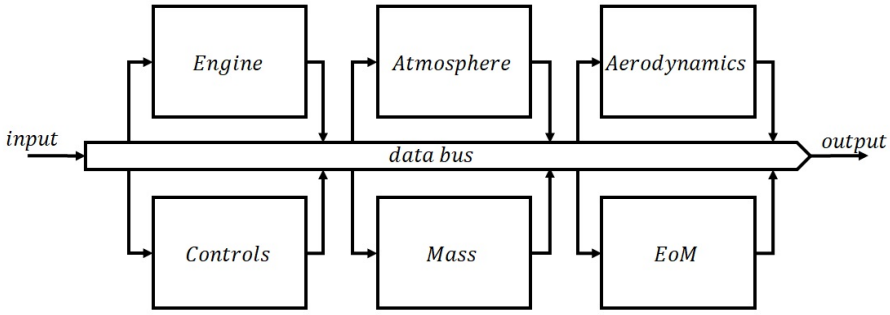

Figure 4.2: Simulink model overview

Aerodynamic Model [8]:

In conceptual design of an aircraft, the definition of aerodynamic database can be performed using either analytical, semi-empirical or numerical aerodynamic calculation techniques. When semi-empirical methods are used within their applicability limits, remarkable levels of accuracy and short turnaround times are achieved, what is ideal for conceptual work [58].

Digital DATCOM is an aerodynamic prediction software developed at McDonnell Douglas based on the database and systematic methods of USAF Stability and Control Datcom [59]. It is capable of calculating static and dynamic derivatives as well as a wide variety of high lift and control device contributions through subsonic, transonic, supersonic, and hypersonic speed regimes. Its primary purpose is to provide a rapid estimation of aerodynamic stability and control characteristics. Consequently, Digital DATCOM is considered to be a convenient tool for the objectives of this study

and used extensively for the aerodynamic model of RQ-3. All DATCOM input files generated for RQ-3 are listed in Appendix A.

Parameters such as Mach number and Reynolds number are required to define the flow regime in Digital DATCOM. In this study a subsonic regime with Mach 0.4 and Re 5.0 million is considered to be a fair approximation for RQ-3.

Tapered wings including effects of sweep, taper and incidence can be treated by Digital DATCOM. A trapezoidal wing planform in accordance with the geometry displayed in Figure 4.3 has been defined in the program.

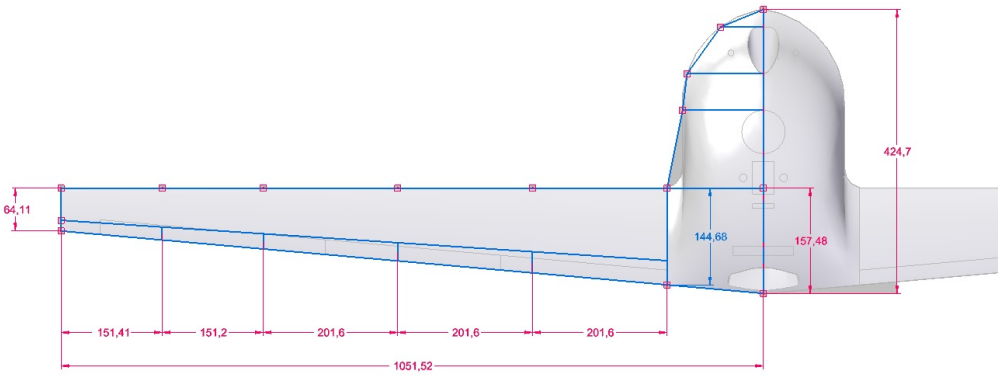

Figure 4.3: Wing geometry (measurements in cm)

In accordance with the published speed and mission characteristics, a thick laminar airfoil, NACA 63-416, is assumed to be used for the wing. Wing is assumed to have no twist and attached to fuselage with an incidence angle of $+1°$.

Digital DATCOM can provide subsonic longitudinal data for cambered bodies of arbitrary cross section. The fuselage geometry is defined using the geometrical parameters of the fuselage sections displayed in Figure 4.4.

Based on the configuration definition and flight regime selection discussed above, the longitudinal coefficients C_D, C_L, C_M, static derivatives C_{Y_β}, C_{N_β}, C_{R_β} and dy-

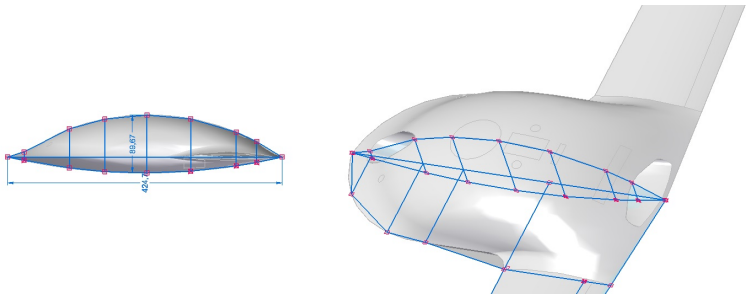

Figure 4.4: Fuselage sections (measurements in cm)

namic derivatives C_{L_q}, C_{M_q}, C_{R_p}, C_{Y_p}, C_{N_p}, C_{N_r} for wing-body combination (wb) are provided as output by the program. Aerodynamic coefficients for our RQ-3 model, which are obtained using Digital DATCOM are displayed in Figure 4.5, Figure 4.6 and Table 4.2.

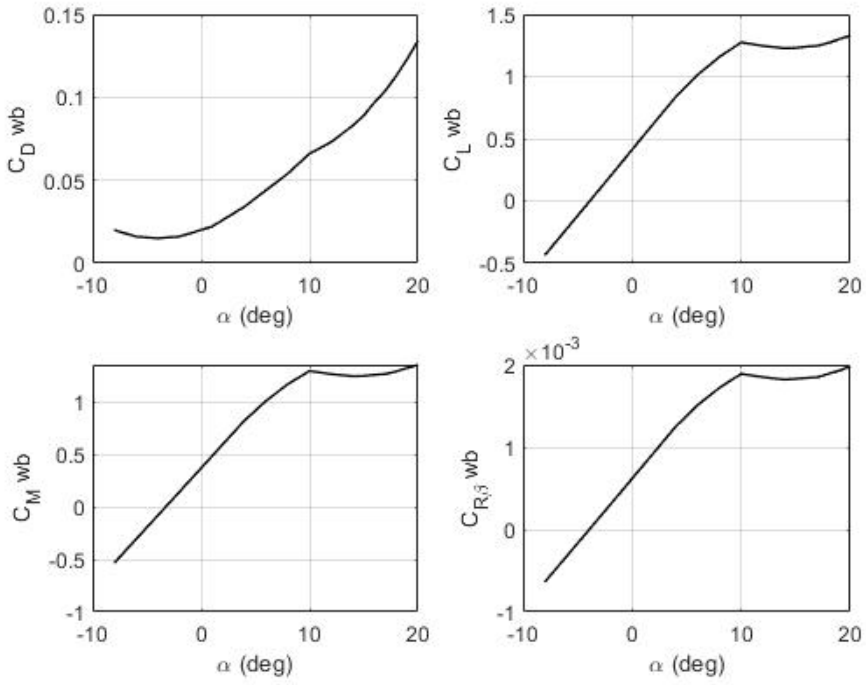

Figure 4.5: Polars for static aerodynamic coefficients

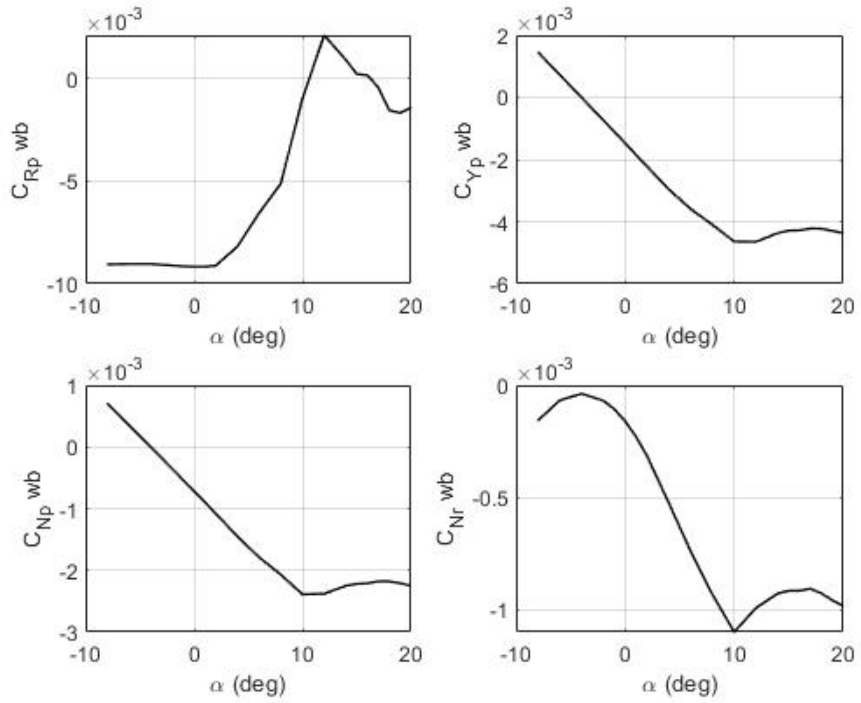

Figure 4.6: Polars for aerodynamic damping coefficients

Table 4.2: Fixed aerodynamic coefficients

C_{Y_β}	-6.08E-03
C_{N_β}	-4.62E-06
C_{L_q}	-1.66E-01
C_{M_q}	-1.36E-01

High-lift devices such as jet flaps, split, plain, single-slotted, double-slotted, fowler, and leading edge flaps and slats, control devices, such as trailing-edge flap-type controls and spoilers, can be analyzed in Digital DATCOM. The program provides the incremental effects of control device deflections at zero angle of attack. Asymmetrical deflection of wing control devices can be analyzed for rolling and yawing effectiveness.

In this study, a suite of multiple control effectors consisting of three trailing edge flaps and two spoilers are assumed to be implemented on each wing. The designation used for each effector is shown in Figure 4.7.

Table 4.3: Control effector designation

No.	Effector
1	Inboard flap
2	Middle flap
3	Outboard flap
4	Inboard spoiler
5	Outboard spoiler

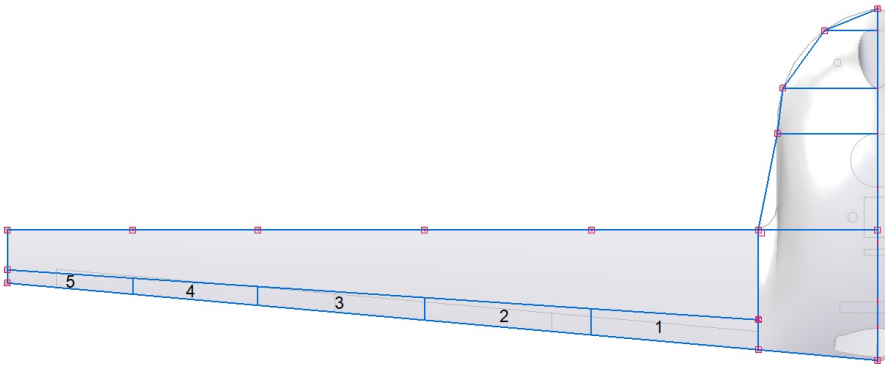

Figure 4.7: Control effector designation

Effectors 1, 2 and 3 are plain flaps with 30% chord ratio. The deflection limits are taken as ±25. Effectors 4 and 5 are spoilers located between 65-80% chord on top surface and maximum projected height of the spoiler is 15% chord, which corresponds to a 90° spoiler angle. Trailing edge down deflection is taken as positive for flaps and spoiler deflection is taken as negative for convention. Maximum spoiler deflection is taken as -25 in the model to be in same order of magnitude with maximum flap deflection.

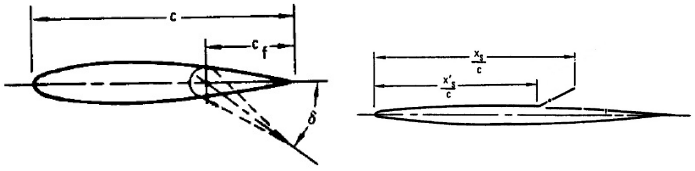

Figure 4.8: Flap and spoiler configurations

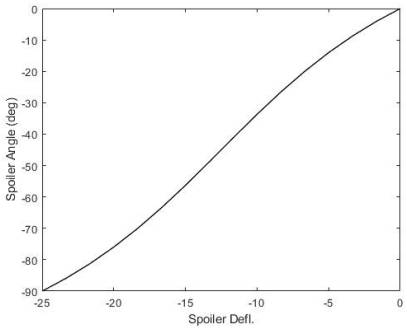

Figure 4.9: Change in spoiler angle with spoiler deflection

Table 4.4: Control effector limits

	Flap defl.	Flap angle	Spoiler defl.	Spoiler angle
Position (min, max)	-25,+25	-25°,+25°	-25,0	-90°,0°
Rate	60/sec	60°/sec	100/sec	360°/sec

Control effectiveness for each of the effectors obtained by Digital DATCOM are shown in Figure 4.10 and Figure 4.11.

The build-up of the aerodynamic model is briefly explained in the following. Non-dimensional rotational rates are defined as:

$$\hat{p} = p\frac{b}{2V};\ \hat{q} = q\frac{c}{2V};\ \hat{r} = r\frac{b}{2V}$$

The total value for each single aerodynamic coefficient is the sum of clean configuration data and contributions from control surfaces and dynamic derivatives.

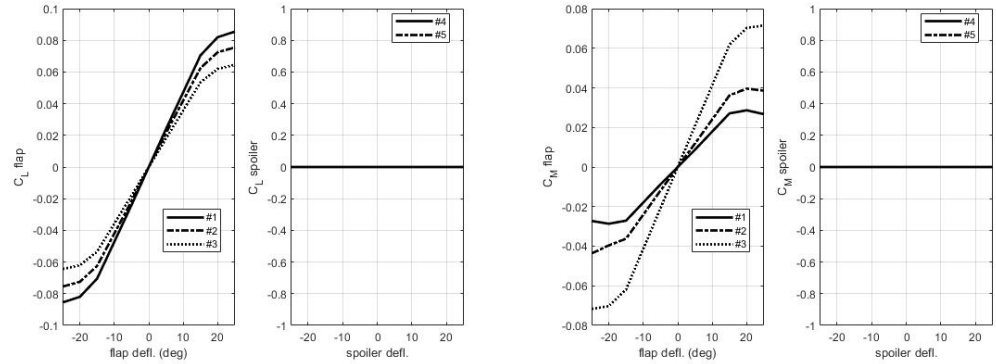

Figure 4.10: Control effectiveness for flaps and spoilers, C_L and C_M

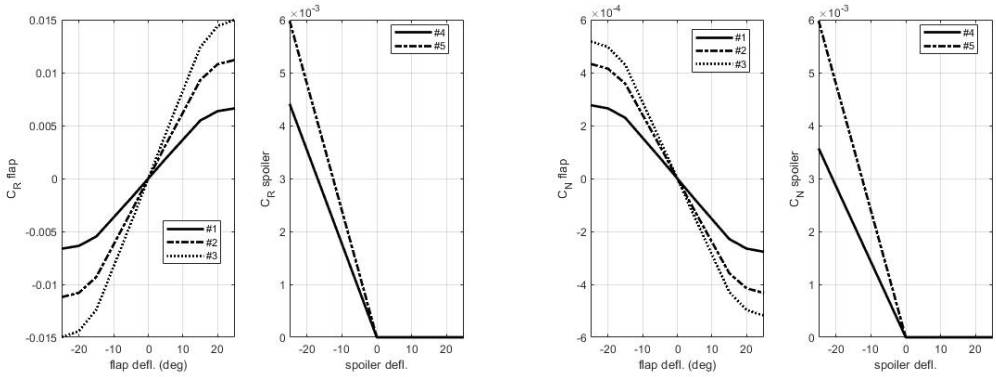

Figure 4.11: Control effectiveness for flaps and spoilers, C_R and C_N

$$C_L = C_{L_{wb}}(\alpha) + C_{L_{q\_wb}}\hat{q} + \sum_{i=1}^{10} C_{L_{eff}}(\delta_i) \tag{4.1}$$

$$C_Y = C_{Y_{\beta\_wb}}\beta + C_{Y_{p\_wb}}(\alpha)\hat{p} + \sum_{i=1}^{10} C_{Y_{eff}}(\delta_i) \tag{4.2}$$

$$C_D = C_{D_{wb}}(\alpha) + \sum_{i=1}^{10} C_{D_{eff}}(\delta_i) \tag{4.3}$$

$$C_R = C_{R_{\beta\_wb}}(\alpha)\beta + C_{R_{p\_wb}}(\alpha)\hat{p} + \sum_{i=1}^{10} C_{R_{eff}}(\delta_i) \tag{4.4}$$

$$C_N = C_{N_{\beta\_wb}}\beta + C_{N_{p\_wb}}(\alpha)\hat{p} + C_{N_{r\_wb}}(\alpha)\hat{r} + \sum_{i=1}^{10} C_{N_{eff}}(\delta_i) \tag{4.5}$$

$$C_M = C_{M_{wb}}(\alpha) + C_{M_{q\_wb}}\hat{q} + \sum_{i=1}^{10} C_{M_{eff}}(\delta_i) \tag{4.6}$$

Dynamic pressure, angle of attack and angle of sideslip are defined as:

$$\overline{q} = \frac{1}{2}\rho V^2;\ \alpha = \tan^{-1}(w/u)\ ;\ \beta = \sin^{-1}(v/V)$$

Dimensional forces and moments are:

$$L = \overline{q}SC_L;\ D = \overline{q}SC_D;\ Y = \overline{q}SC_Y;\ M = \overline{q}ScC_M;\ N = \overline{q}SbC_N;\ R = \overline{q}SbC_R$$

and body axis aerodynamic forces are:

$$X = Lsin\,(\alpha) - Dcos\,(\alpha)$$

$$Z = -Lcos\,(\alpha) - Dsin\,(\alpha)$$

DATCOM methods are known to have limitations in drag calculations and low accuracy for thick airfoils. Besides, it is seen that parameters such as effects of spoiler use in wing lift are totally ignored in the program output. As this study is not focused on the accuracy of the mathematical model, such deficiencies are disregarded as long as they are considered to have a secondary impact on the derivatives of concern.

Engine Model:

A simplified 2^{nd} order dynamic model based on the more complex engine dynamic response characteristics defined in Ref. [60] is used for RQ-3 engine simulation. A thrust between idle and maximum throttle, T_{max}, is assumed to be changing linearly with throttle position, δ_{thr}. The simplified model is:

$$T = T_{max}\frac{1.96}{s^2 + 2.8s + 1.96}(\delta_{thr}/100)$$

where $0 \leq \delta_{thr} \leq 100$ and $T_{max} = 10kN$

Flight Controls:

A flight control model composed of a control allocation block and servo dynamics is implemented. Two different control allocation approaches have been tested.

A detailed discussion on these have been given in preceding chapters. A 2^{nd} order servo dynamics is implemented for each one of control effectors. The linear models for actuators are:

$$\delta_{out} = \frac{2500}{s^2 + 100s + 2500}\delta_{in}$$

Malfunctions on each effector is implemented using position or rate limit parameters. For instance, if an actuator is stuck on a time instant, its rate limits are set to zero until the malfunction is removed. Or, in some other cases, the position limits of the actuator is set to a fixed (failure) position.

Mass Model:

A single weight, CG and moment of inertia combination is used in this study. No change is assumed during flight. The parameters used are given in Table 4.5.

Table 4.5: Inertial parameters

mass	3800 kg
CG pos.	-60% mac
I_x	9104 kg.m^2
I_y	13124 kg.m^2
I_z	21490 kg.m^2
I_{xz}	872 kg.m^2
I_{xy}	0 kg.m^2
I_{yz}	0 kg.m^2

Atmosphere Model [15]:

U.S. Standard Atmosphere model (1976) is implemented in the model.

Equations of Motion [15]:

Aircraft motion is calculated using the positive body axis directions as displayed in Figure 4.12.

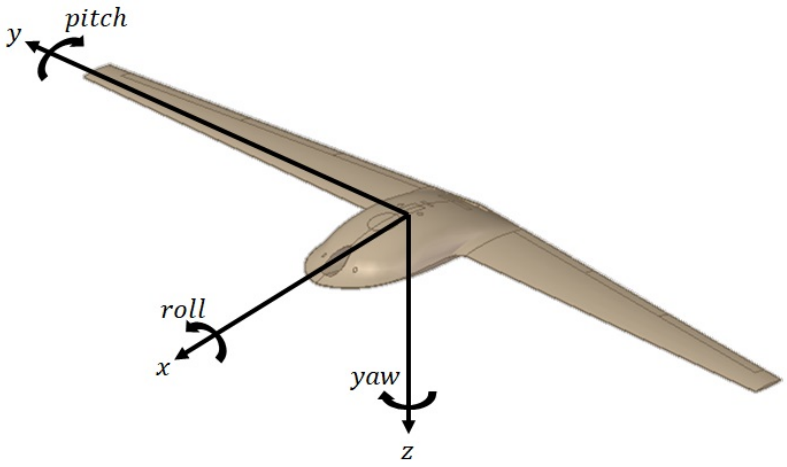

Figure 4.12: Axis convention

The set of equations of motion in 6 DoF used in the nonlinear simulation model is given in the following. The state vector $\overline{x}$ is defined as:

$$\overline{x} = [p_N, p_E, p_D, \phi, \theta, \psi, u, v, w, p, q, r]$$

Force Equations:

$$\dot{u} = rv - qw - gsin(\theta) + X_{total}/mass \tag{4.7}$$

$$\dot{v} = -ru + pw - gsin(\phi)cos(\theta) + Y_{total}/mass \tag{4.8}$$

$$\dot{w} = qu - pv + gcos(\phi)cos(\theta) + Z_{total}/mass \tag{4.9}$$

Moment Equations:

$$\Gamma\dot{p} = I_{xz}[I_x - I_y + I_z]pq - [I_z(I_z - I_y) + I_{xz}^2]qr + I_z R + I_{xz}N \tag{4.10}$$

$$I_y\dot{q} = (I_z - I_x)pr - I_{xz}(p^2 - r^2) + m \tag{4.11}$$

$$\Gamma\dot{r} = [(I_x - I_y)I_x + I_{xz}^2]pq - I_{xz}[I_x - I_y + I_z]qr + I_{xz}R + I_x N \tag{4.12}$$

$$\Gamma = I_x I_x - I_{xz}^2 \tag{4.13}$$

Kinematic Equations

$$\dot{\phi} = p + tan(\theta)(qsin(\phi) + rcos(\phi)) \tag{4.14}$$

$$\dot{\theta} = qcos(\phi) - rsin(\phi) \tag{4.15}$$

$$\dot{\psi} = (qsin(\phi) + rcos(\phi))/cos(\theta) \tag{4.16}$$

$$\tag{4.17}$$

Navigation Equations

$$\dot{p}_N = u(c\theta c\phi) + v(-c\phi s\psi + s\phi s\theta c\psi) + w(s\phi s\psi + c\phi s\theta c\psi) \tag{4.18}$$

$$\dot{p}_E = u(c\theta s\psi) + v(c\phi c\psi + s\phi s\theta s\psi) + w(-s\phi c\psi + c\phi s\theta s\psi) \tag{4.19}$$

$$\dot{h} = u(s\theta) - v(s\phi c\theta) - w(c\phi c\theta) \tag{4.20}$$

4.3 Trim and Linearization

When engineers speak of trim, they refer to the process of manipulating Euler angles and pilot controls within constraints to cause the sum of the forces and moments on each of the body axes to go to zero. In other words, trimming a vehicle is the process of finding equilibrium. Several methods are available to trim the aircraft model or dynamic subsystems. Some popular methods are sequential-correction, fly-to-trim, Jacobian and Periodic Shooting methods [61]. In this work, the preferred method for finding a trim is the Jacobian method.

The Jacobian method, in the simplest expression, finds the trim by iteratively solving the equations of motions for the given trim inputs and unknowns with zero accelerations. It runs Newton-Raphson iterations using the Jacobian, until the error between consecutive iterations converges below a defined minimum. A thorough description of the method is provided in Ref. [61].

The nonlinear simulation model, which is covered in the previous chapter, is trimmed for level flight at various operating points between 130-270 KIAS (Indicated Airspeed - Knots). The variation of angle of attack and control effectors with airspeed is shown in Figure 4.13.

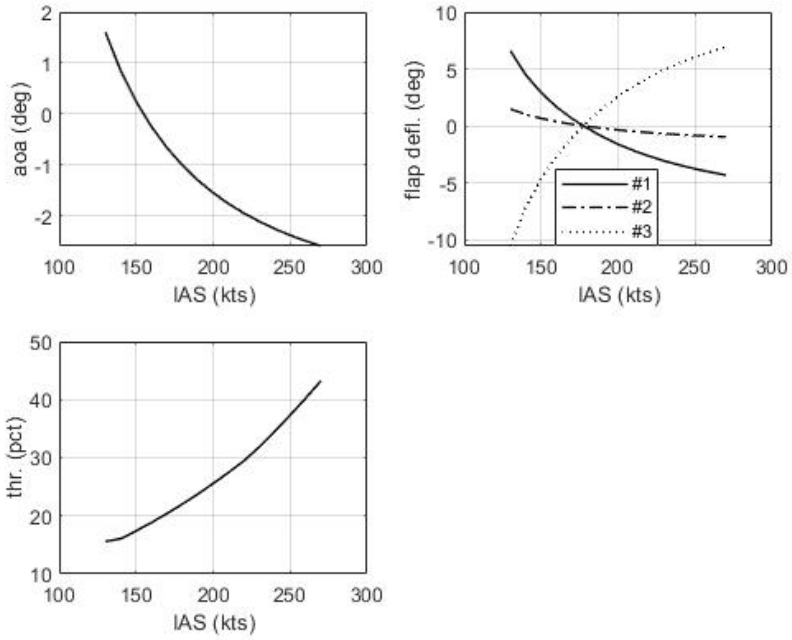

Figure 4.13: Level flight trim points

The figures show that, the aircraft can be stabilized with zero control surface deflection at an airspeed of about 170 KIAS. The change of parameters are smooth, which is an important point for controller gain scheduling.

After trimming, linear models corresponding to each operating points are generated by Jacobian linearization. Since DATCOM based aerodynamic model is decoupled in itself and so the inertial characteristics, separate linear models are generated for longitudinal and lateral-directional dynamics. The inputs, outputs and states for longitudinal and lateral-directional models are given in Table 4.6.

Table 4.6: Inputs, outputs and states for linear systems

	Long.	Lat.-Dir.
Inputs	$\upsilon_{pitch}, \upsilon_{thr}$	$\upsilon_{roll}, \upsilon_{yaw}$
Outputs	V, q, N_z, θ	β, p, r, N_y, ϕ
States	u, w, θ, q	v, r, p, ϕ

Variation of poles, natural frequencies and damping parameters with trim airspeed for each of the modes are shown in Figure 4.14 and Figure 4.15.

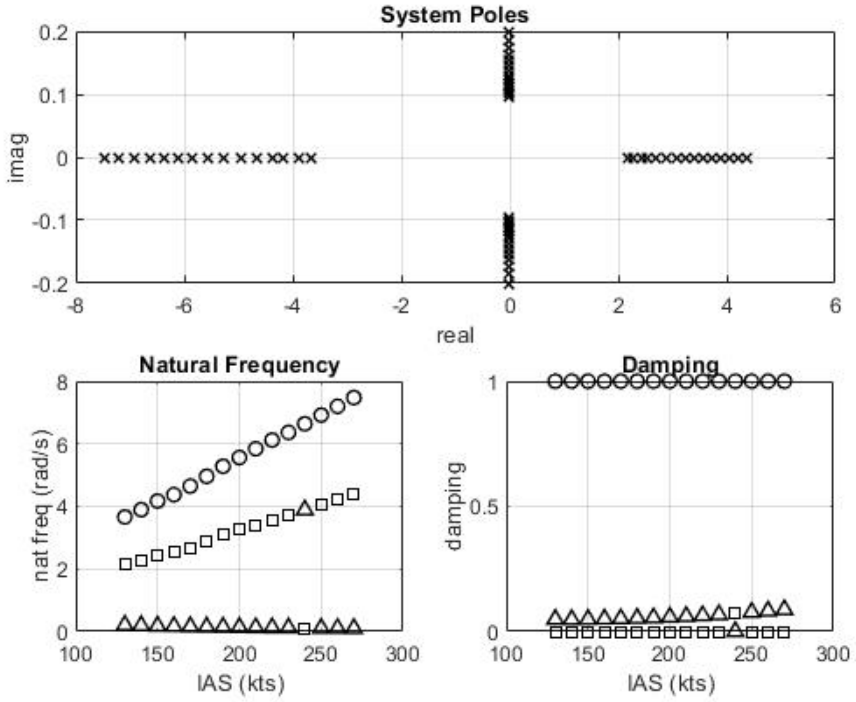

Figure 4.14: Longitudinal poles, natural frequencies and damping

The longitudinal poles show one unstable and two stable modes. One of the stable modes is a lightly damped low frequency motion resembling Phugoid. Two higher frequency modes, one stable with 3-8 rad/s and the other unstable with 2-5 rad/s frequency together form the -fast- longitudinal dynamics.

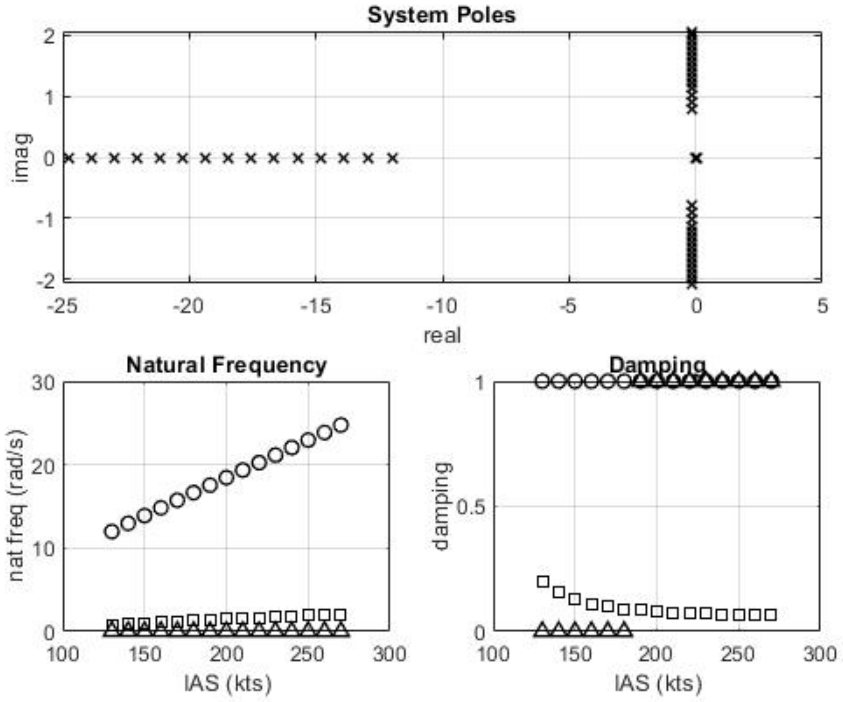

Figure 4.15: Lateral-Directional poles, natural frequencies and damping

Lateral-directional poles indicates a more familiar dynamics, similar with conventional wing-body-tail configurations. Three modes, one of them a critically damped high frequency Roll mode, the other a stable, lightly damped low frequency Dutch-Roll mode, and the last, a very low frequency Spiral mode which is unstable below 180 kts can be distinguished in the plots.

4.4 Controller

In this chapter μ-synthesis for the longitudinal and lateral-directional controllers will be covered in detail. As mentioned before, the flight control system is composed of two major parts: a controller and a control allocation module. The controller block includes the LTI control laws synthesized for different operating points and generates 'virtual' controls which are perfectly decoupled in roll, pitch and yaw axes,

and throttle commands. In the final section, application details for the scheduling approaches will be presented.

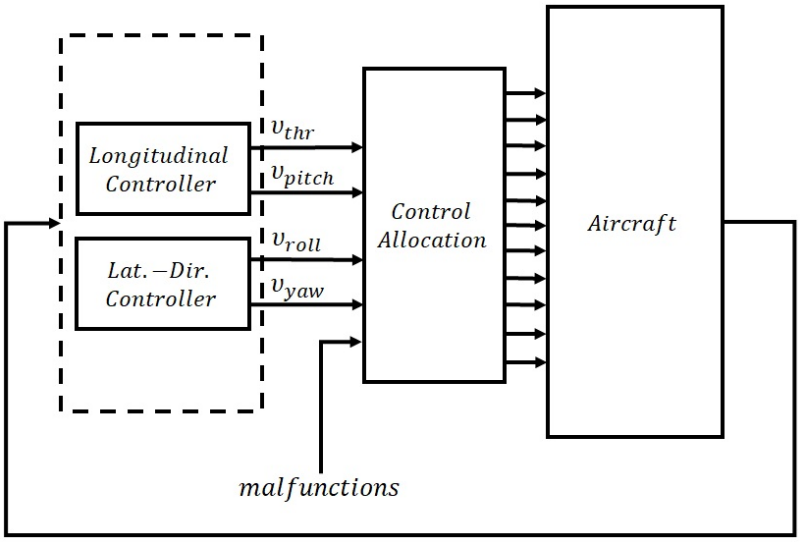

Figure 4.16: Interaction of controller, control allocation module and aircraft

4.4.1 μ-Synthesis

MIL HDBK 1797 [62] provides a set of requirement specifications on handling qualities and dynamic responses for piloted aircraft. Although UAVs are not within the intended scope of this specification, some of the requirements are used as reference in this study. An interconnection structure as shown in Figure 4.17 is used for the synthesis of longitudinal controller.

The longitudinal controller is designed to follow some reference models defined for pitch attitude and airspeed. Reference dynamics for θ is a second order model with a gain of 3, natural frequency of 1.1rad/s and damping ratio of 0.6. This results in a step response with a rise time less than 2 sec. The selected frequency and damping

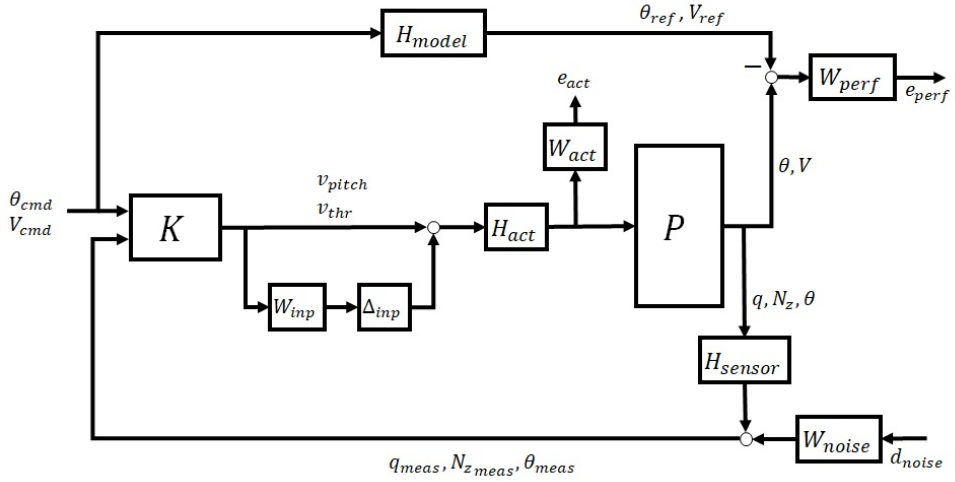

Figure 4.17: Longitudinal interconnection model for synthesis

values corresponds to Level I Flying Qualities, for gradual maneuvers of light aircraft in MIL HDBK 1797.

$$H_{model_\theta} = 3\frac{(1.1)^2}{s^2 + (2 * 0.6 * 1.1)s + (1.1)^2}$$

Reference dynamics for airspeed is way slower, with about 10secs of rise time. This is due to relatively slow engine dynamics and slow speed response of the air vehicle. A gain of 9 is applied, meaning that unit input will result in an airspeed change of 9 kts.

$$H_{model_V} = 9\frac{(0.3)^2}{s^2 + (2 * 1.0 * 0.3)s + (0.3)^2}$$

The Bode gain plots of the reference models is shown in Figure 4.18.

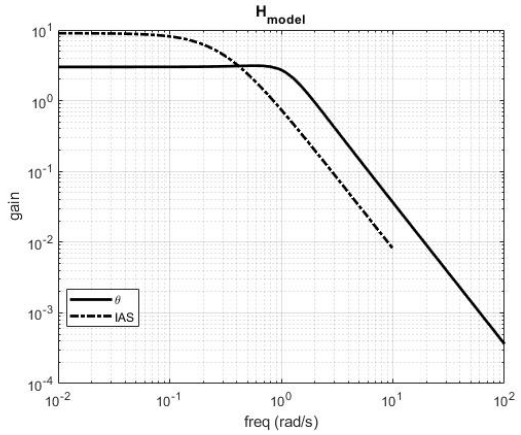

Figure 4.18: Bode gain plots for reference dynamicss

Sensor dynamics for pitch rate, pitch attitude and normal acceleration are implemented as a combination of anti-aliasing filters and delays. The cutoff frequency is assumed to be 4.1Hz for q and θ filters, and 12.5Hz for N_z filter. A delay of 20ms is assumed for all sensors.

$$H_{sensor_q} = H_{sensor_\theta} = \frac{(4.1 * 2\pi)^2}{s^2 + (2 * 0.7 * 4.1 * 2\pi)s + (4.1 * 2\pi)^2} D_{Pade}$$

$$H_{sensor_{N_z}} = \frac{(12.5 * 2\pi)^2}{s^2 + (2 * 0.5 * 12.5 * 2\pi)s + (12.5 * 2\pi)^2} D_{Pade}$$

2^{nd} order Pade approximation for 20ms sensor delay is implemented.

$$D_{Pade} = \frac{s^2 - 300s + 30000}{s^2 + 300s + 30000}$$

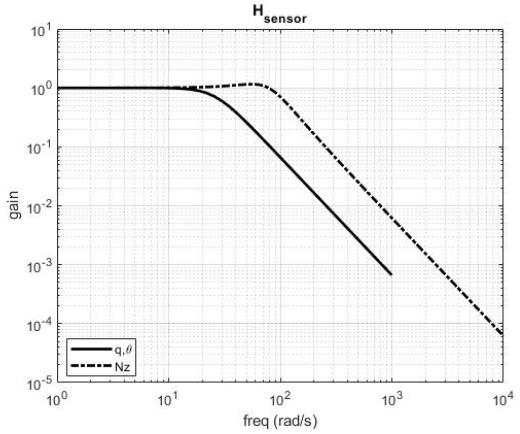

Figure 4.19: Bode magnitude for anti-aliasing filters

Electro-mechanical servo actuators with 50rad/s cutoff and 0.7 damping is assumed to be used for the flaps. As flaps are the primary effectors for pitching motion, these values are used for the dynamic model of the virtual pitch command as well.

$$H_{act_{pitch}} = \frac{(50)^2}{s^2 + (2 * 0.7 * 50)s + (50)^2}$$

Normally, actual throttle actuator dynamics is expected to be implemented here for the virtual throttle model. However, as the linearized aircraft model does not contain any states for engine dynamics, which is not fast enough to be totally neglected, we insert engine dynamics into the synthesis model at this point. Since throttle actuator dynamics is expected to be much faster than engine dynamics, it is neglected. Therefore, the virtual throttle actuator model becomes:

$$H_{act_{thr}} = \frac{(1.4)^2}{s^2 + (2 * 1.0 * 1.4)s + (1.4)^2}$$

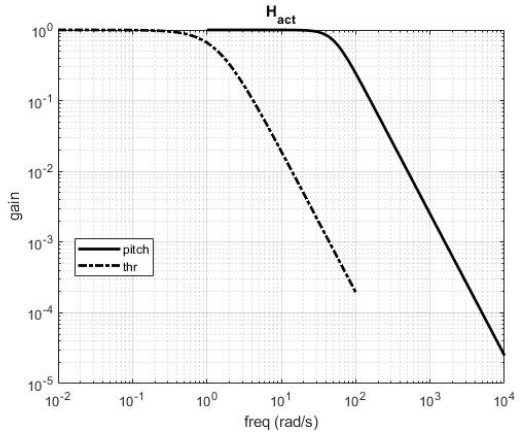

Figure 4.20: Bode magnitude for actuators

The weighting filters for the tracking performance of longitudinal controller are selected after some trial and errors. As a result, a first order filter with gain 6 and cutoff 50rad/s is used for pitch attitude, and a first order filter with gain 2 and cutoff 3rad/s for airspeed.

$$W_{perf_\theta} = 6\frac{50}{s+50}, \quad W_{perf_V} = 2\frac{3}{s+3}$$

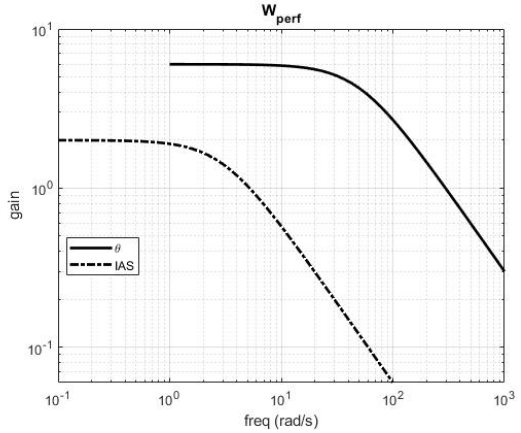

Figure 4.21: Bode magnitude for tracking performance error filters

Rate and attitude signals are assumed to be 2.5% noisy. Acceleration signals are assumed to be pretty accurate for frequencies below 1rad/s and having a noise ratio of 1.25% at high frequency.

$$W_{noise} = diag(0.025, \frac{0.0125s + 0.0125}{s + 100}, \ 0.025)$$

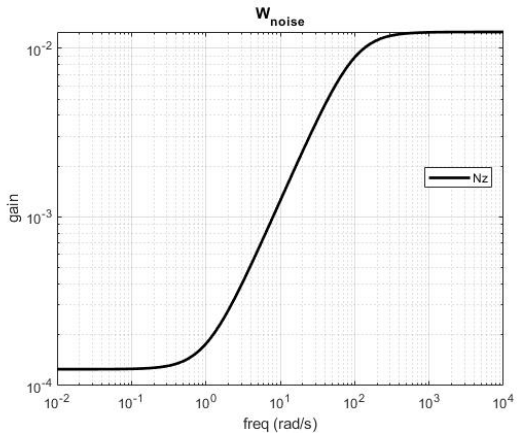

Figure 4.22: Bode magnitude for N_z noise filter

Virtual pitch commands are allowed to change between -25° and +25° with a rate up to 100°/sec. Similarly virtual throttle command is assumed to change between -25% and +25% with a rate up to 100%/sec. The maximum allowed rates may look little bit too high at the first glance, however, they have been chosen so in order to be able satisfy the robust performance requirements. Besides, the controllers are observed to generate commands with much lower rates than the ones specified here.

$$W_{act} = diag(\frac{1}{100}, \frac{1}{50}, \frac{1}{100}, \frac{1}{50})$$

The multiplicative process uncertainty for pitch command is chosen to be about 1% below 1 rad/s and 100 % above 100 rad/s. As for the throttle command, uncertainty is taken 0.1% below 0.01 rad/s and 100% above 10 rad/s.

$$W_{inp_{pitch}} = \frac{s+1}{s+100}, \quad W_{inp_{thr}} = \frac{s+0.01}{s+10}$$

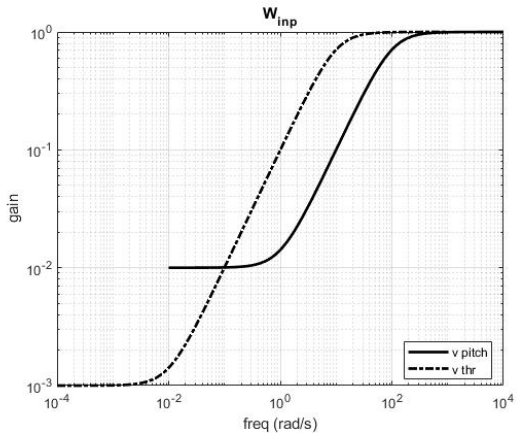

Figure 4.23: Bode magnitude for process uncertainty

An interconnection structure as shown in Figure 4.24 is used for the synthesis of lateral-directional controller.

The lateral-directional controller is designed to follow some reference models defined for roll attitude and sideslip. Reference dynamics for ϕ is a second order model with a gain of 30, natural frequency of 0.7rad/s and damping ratio of 0.5. So, maximum roll attitude with unit input is chosen to be 30°, which is considered to be quite sufficient for the intended missions of RQ-3. The military specification given in Ref. [62], defines Level I performance as 60° bank in 1.7sec for light aircraft in gradual maneuvers. The reference model gives 30° bank in 3.4sec. Although this value is much

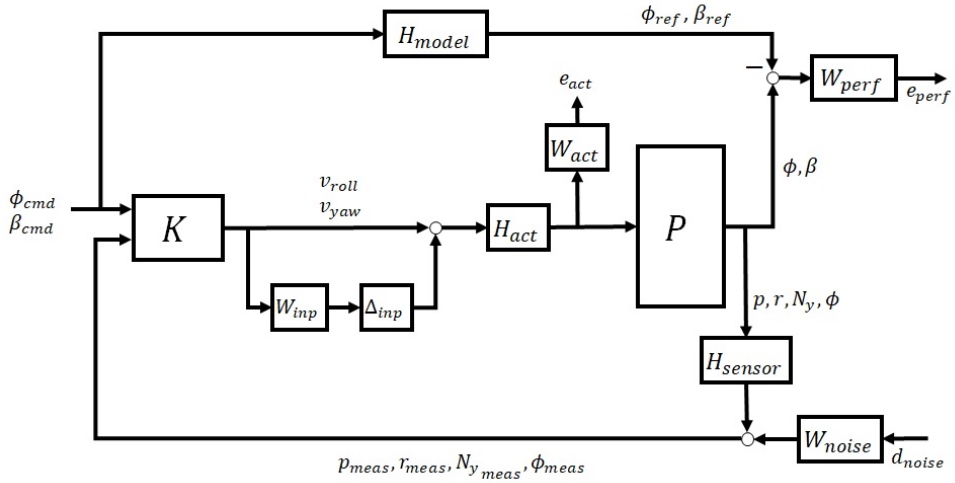

Figure 4.24: Lateral-directional interconnection model for synthesis

slower than the specification, it is known that the intended use of the UAV is also substantially different from the manned aircraft considered in the specification.

$$H_{model_\phi} = 30\frac{(0.7)^2}{s^2 + (2 * 0.5 * 0.7)s + (0.7)^2}$$

Although the primary role of the sideslip controller is to provide turn coordination, a tracker is also designed for flight conditions like approach in sidewind. The controller provides a 5° slip to unit input with a rise time less than 2sec.

$$H_{model_\beta} = -5\frac{(1.25)^2}{s^2 + (2 * 1.0 * 1.25)s + (1.25)^2}$$

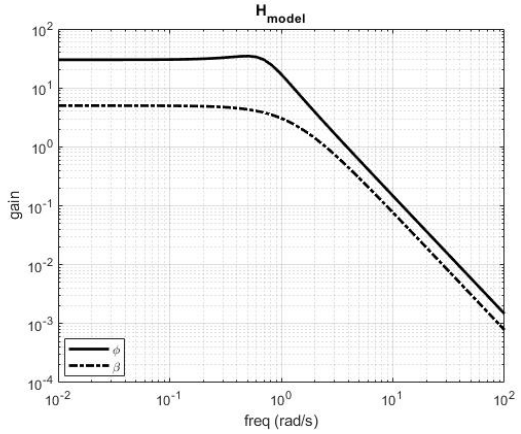

Figure 4.25: Bode magnitude for reference dynamics

Sensor dynamics for roll rate, yaw rate, roll attitude and lateral acceleration are implemented as a combination of anti-aliasing filters and delays. The cutoff frequency is assumed to be 4.1Hz for p and ϕ filters, and 12.5Hz for r and N_y filters. A delay of 20ms is assumed for all sensors.

$$H_{sensor_p} = H_{sensor_\phi} = \frac{(4.1 * 2\pi)^2}{s^2 + (2 * 0.7 * 4.1 * 2\pi)s + (4.1 * 2\pi)^2} D_{Pade}$$

$$H_{sensor_r} = H_{sensor_{N_y}} = \frac{(12.5 * 2\pi)^2}{s^2 + (2 * 0.5 * 12.5 * 2\pi)s + (12.5 * 2\pi)^2} D_{Pade}$$

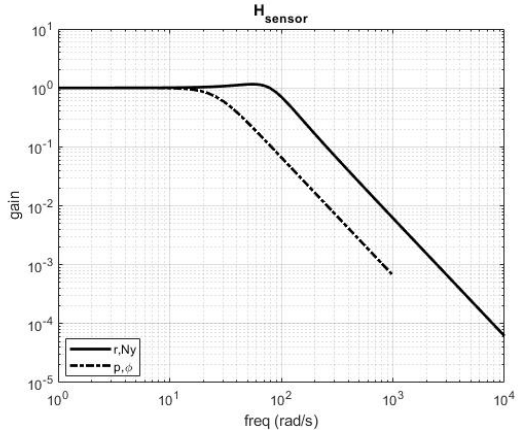

Figure 4.26: Bode magnitude for anti-aliasing filters

Flaps are the primary effectors for rolling motion as well, so the flap actuator dynamics is also used for the virtual roll command. Spoilers are assumed to be driven by slightly different actuators, with 50rad/s cutoff and 0.8 damping. Primary yaw effectors are spoilers, so virtual yaw actuator dynamics is assumed to be same with spoiler servo.

$$H_{act_{roll}} = \frac{(50)^2}{s^2 + (2 * 0.7 * 50)s + (50)^2}$$

$$H_{act_{yaw}} = \frac{(50)^2}{s^2 + (2 * 0.8 * 50)s + (50)^2}$$

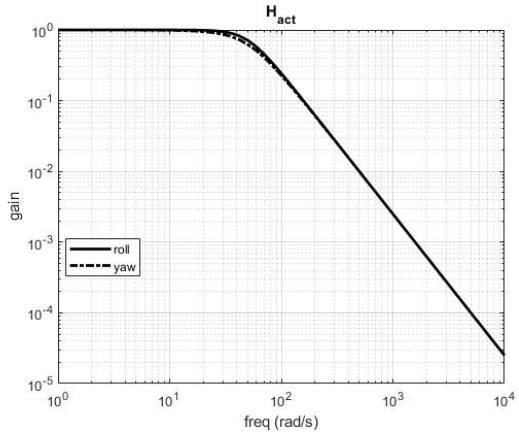

Figure 4.27: Bode magnitude for roll and yaw virtual commands

A first order filter with gain 0.5 and cutoff 7rad/s is used for roll attitude, and a first order filter with gain 2.5 and cutoff 10rad/s for sideslip.

$$W_{perf_\phi} = 0.5\frac{7}{s+7}, \quad W_{perf_\beta} = 2.5\frac{10}{s+10}$$

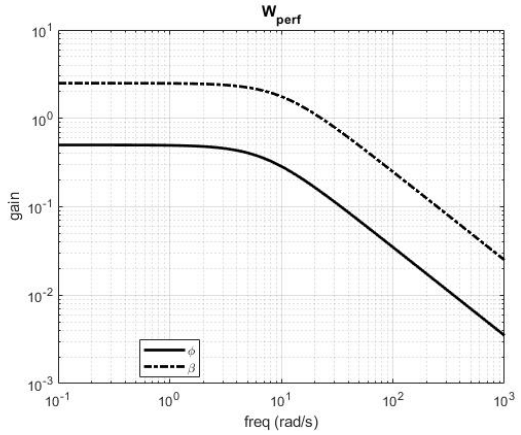

Figure 4.28: Bode magnitude for tracking performance error filters

Rate and attitude signals are assumed to be 2.5% noisy. Acceleration signals are assumed to be pretty accurate for frequencies below 1rad/s and having a noise ratio of 1.25% at high frequency.

$$W_{noise} = diag(0.025,\ 0.025, \frac{0.0125s + 0.0125}{s + 100}, 0.025)$$

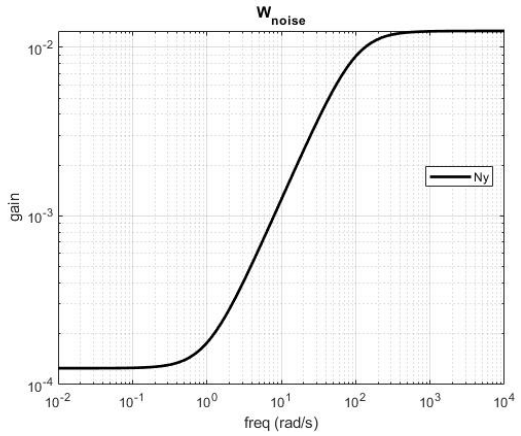

Figure 4.29: Bode magnitude for noise

Both virtual roll and way commands are allowed to change between -25° and +25° with a rate up to 100°/sec.

$$W_{act} = diag(\frac{1}{100}, \frac{1}{50}, \frac{1}{100}, \frac{1}{50})$$

The multiplicative process uncertainty for roll command is chosen to be about 5% below 4 rad/s and 200% above 160 rad/s. For virtual yaw command, uncertainty is taken 15% below 20 rad/s and 150% above 200 rad/s.

$$W_{inp_{roll}} = 2\frac{s+4}{s+160}, \quad W_{inp_{yaw}} = 1.5\frac{s+20}{s+200}$$

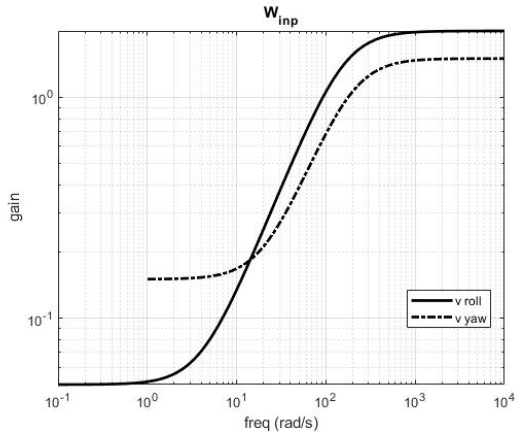

Figure 4.30: Bode magnitude for process uncertainty

The controller synthesis is repeated for each operating point (or design point) selected between 130-270 KIAS. As a consequence of the synthesis method selected, each controller is an LTI system with an order much higher than the order of linear aircraft model. The robust performance of the closed-loop system is guaranteed by the synthesis method for the design points but this does not assure stability at any flight conditions in between.

MATLAB-Robust Control Toolbox [63] is used as the synthesis tool. The frequency band is selected between 10^{-2} rad/s and 10^{2} rad/s. The maximum value of the scaling matrix order is limited in order to obtain a similar controller order for most of the design points. Also, in order to help the controller parameters be close on neighboring design points, the initial controller used in D-K iteration is selected as the controller synthesized in the previous design point.

Variation of closed-loop gain vs frequency at low, medium and high speed design points are shown in Figure 4.31 for longitudinal closed-loop system and in Figure 4.32 for lateral directional closed-loop system. As can be followed from the figures, lateral-directional system has much higher robustness margin compared to the longitudinal one. The airspeed does not seem to have a considerable effect on the longitudinal system, however, the robustness margin seems to increase with airspeed in the lateral-directional.

4.4.2 Gain Scheduling

Ad-hoc Linear Interpolation:

A simple ad-hoc linear interpolation between the two controllers corresponding to two consecutive design points is performed and the performance characteristics of the resultant closed-loop system is investigated.

For two consecutive design points a, b where P_a, P_b are the bare (with no filters etc.) linear plants and K_a, K_b are the controllers synthesized for the corresponding interconnections, we define: $P = \frac{P_a+P_b}{2}$, $\Delta_p = \frac{P_b-P_a}{2}$ and $K = \frac{K_a+K_b}{2}$, $\Delta_k = \frac{K_b-K_a}{2}$ Following the approach explained in Chapter 2.5, the interpolated system can be represented as the closed-loop system composed of plant P with an uncertainty Δ_p and controller K with an uncertainty Δ_k as shown in Figure 4.33.

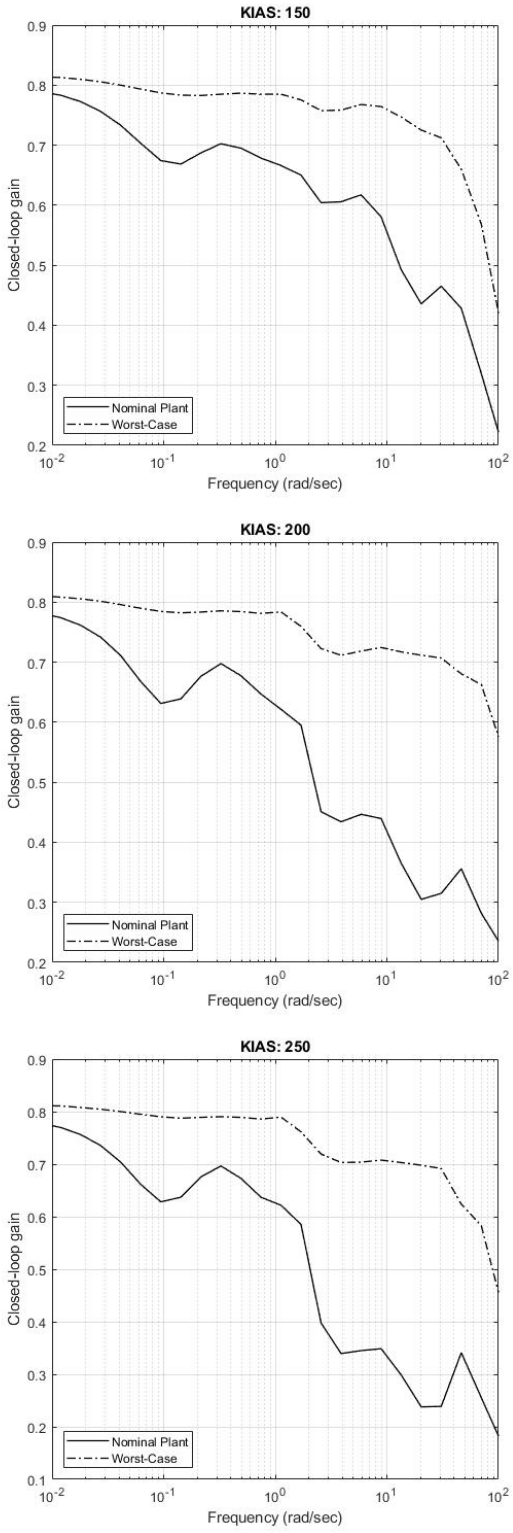

Figure 4.31: Performance of longitudinal controller

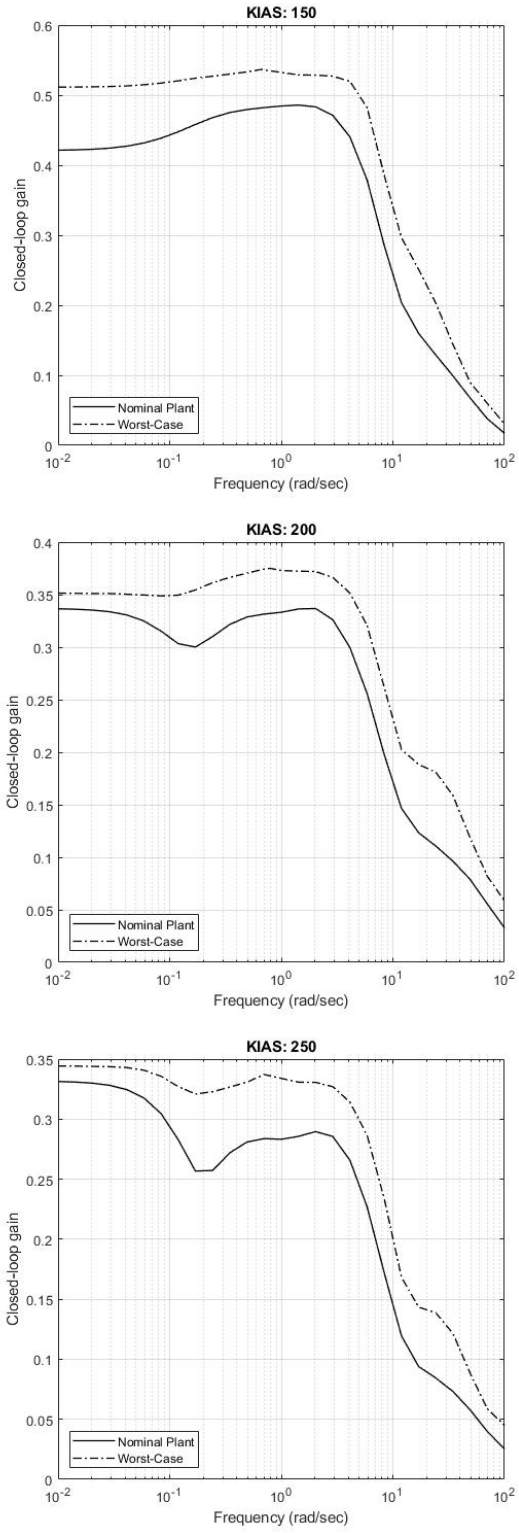

Figure 4.32: Performance of lateral-directional controller

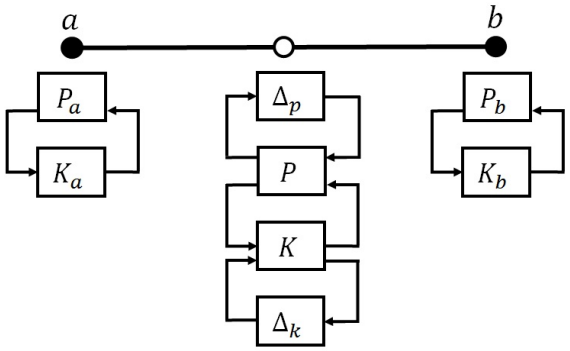

Figure 4.33: Linearly interpolated system

μ-Analysis results obtained for the interpolated systems are shown in Figure 4.34. As can be followed, there is no significant change in the nominal performance of the interpolated system, but the robust performance is deteriorated significantly. Despite the loss in performance, the interpolated system is still stable and the interpolation will work fine for nominal system. However, this may not always be the case. Especially when the two subsequent controllers differ significantly from each other, the interpolation may end up with an unstable closed-loop system [64].

Synthesis using Simplified LPV Model:

A single controller for two consecutive design points can simply be synthesized using linear Simplified LPV Model (Figure 2.15) as explained in Chapter 2.5. The controller is assumed to be linear in parameters, and an interconnection shown in Figure 2.16 is used. All components of the interconnection model are implemented as defined in Chapter 4.4.1. The design points are the two LTI models linearized at 190 kts and 200 kts airspeeds. μ-Analysis results for the simplified LPV model synthesis are shown in Figure 4.35.

A comparison of Figure 4.34 and Figure 4.35 reveals a difference between nominal and robust (worst-case) performance between the linearly interpolated systems and simplified LPV synthesis. The former has a low nominal gain, but very high

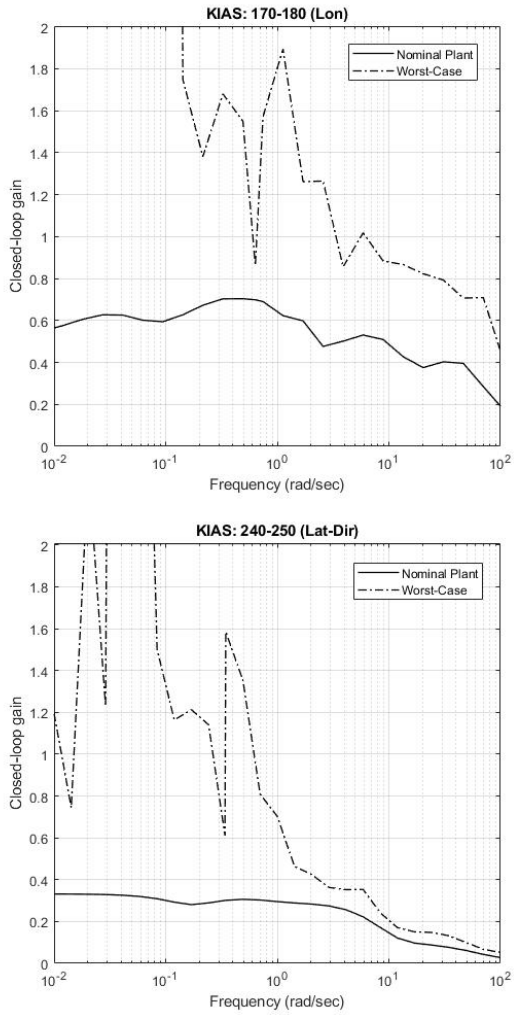

Figure 4.34: Performance for linearly interpolated systems

worst-case gain. However, for the latter, a far better worst case performance is evident with some sacrifice from the nominal performance.

Stability Preserving Interpolation:

The procedure applied for the implementation of Stability Preserving Interpolation is described in Table 4.7. All the steps in this procedure is based on the formulation given in Chapter 2.5.

Table 4.7: Procedure for Stability Preserving Interpolation

Step	Eqn.	
1	(2.5),(2.6)	For $\rho \in [0,1]$ and $i = 1,2$ Generate Linear State Space IC models for end points, Λ_1, Λ_2, $\Sigma(0)$ and $\Sigma(1)$
2	(2.7)	Calculate $\mathcal{F}_l(\Sigma(0), \Lambda_2)$ and $\mathcal{F}_l(\Sigma(1), \Lambda_1)$
3		From $\Sigma(0)$ and $\Sigma(1)$, Extract $F(0)$,$G_1(0)$,$G_2(0)$,$H_1(0)$ $H_2(0)$,$J_{11}(0)$,$J_{12}(0)$,$J_{21}(0)$,$J_{22}(0)$ Extract $F(1)$,$G_1(1)$,$G_2(1)$,$H_1(1)$,$H_2(1)$,$J_{11}(1)$,$J_{12}(1)$,$J_{21}(1)$,$J_{22}(1)$
4	(2.10)	Extract $\tilde{A}_1(0)$ from $\mathcal{F}_l(\Sigma(0), \Lambda_1)$ Extract $\tilde{A}_2(1)$ from $\mathcal{F}_l(\Sigma(1), \Lambda_2)$
5		Solve LMIs $\tilde{A}_1^T(0)W_1(0) + W_1(0)\tilde{A}_1(0) < -I$ and $W_1(0) > 0$ for $W_1(0)$ Solve LMIs $\tilde{A}_2^T(1)W_2(1) + W_2(1)\tilde{A}_2(1) < -I$ and $W_2(1) > 0$ for $W_2(1)$
6	(2.23)	Extract $S_1(0)$, $N_1(0)$, $P_1(0)$ from $W_1(0)$ Extract $S_2(1), N_2(1), P_2(1)$ from $W_2(1)$ Extract $R_1(0), M_1(0), Q_1(0)$ from $\mathrm{W}_1^{-1}(0)$ Extract $R_2(1), M_2(1), Q_2(1)$ from $\mathrm{W}_2^{-1}(1)$
7	(2.24)	Calculate $L_1(0), L_2(1), K_1(0), K_2(1)$
8	(2.10)	Calculate $\tilde{B}_1(0), \tilde{C}_1(0), \tilde{D}_1(0)$ Calculate $\tilde{B}_2(1), \tilde{C}_2(1), \tilde{D}_2(1)$
9	(2.17)	Select a value for interpolation parameter ρ' where $0 \leq \rho' \leq 1$; Calculate $\tilde{W}(\rho'), \tilde{A}(\rho'), \tilde{B}(\rho'), \tilde{C}(\rho'), \tilde{D}(\rho')$ Calculate $F(\rho'), G_2(\rho'), H_2(\rho')$
10		Calculate $\tilde{A}_1(\rho')$, $\tilde{A}_2(\rho')$
11		Solve LMIs $\tilde{A}_1^T(\rho')W_1(\rho') + W_1(\rho')\tilde{A}_1(\rho') < -I$ and $W_1(\rho') > 0$ for $W_1(\rho')$ Solve LMIs $\tilde{A}_2^T(\rho')W_2(\rho') + W_2(\rho')\tilde{A}_2(\rho') < -I$ and $W_2(\rho') > 0$ for $W_2(\rho')$
12	(2.23)	Extract $S_1(\rho'), N_1(\rho')$ from $W_1(\rho')$ Extract $S_2(\rho'), N_2(\rho')$ from $W_2(\rho')$ Extract $R_1(\rho'), M_1(\rho')$ from $\mathrm{W}_1^{-1}(\rho')$ Extract $R_2(\rho'), M_2(\rho')$ from $\mathrm{W}_2^{-1}(\rho')$
13	(2.24)	Calculate $L_1(\rho'), L_2(\rho'), K_1(\rho'), K_2(\rho')$
14	(2.26)	Calculate $S(\rho'), R(\rho'), L(\rho'), K(\rho')$
15	(2.8)	Calculate $\mathcal{J}(\rho')$
16		Calculate $\Lambda(\rho')$ from $\mathcal{F}_l(\mathcal{J}(\rho'), \mathcal{Q}(\rho'))$

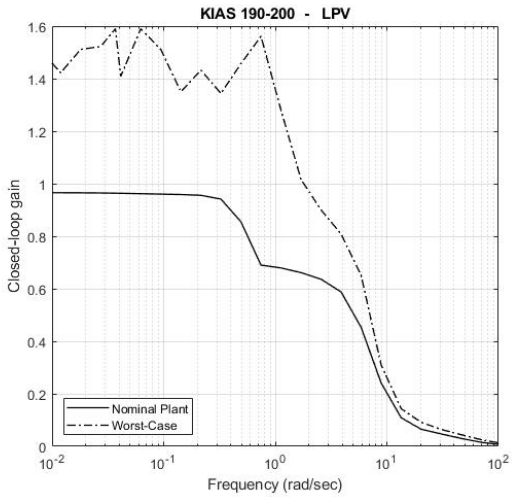

Figure 4.35: Performance of simplified LPV controller

The steps given in Table 4.7 have been implemented in MATLAB, and applied for some selected values of interpolants successfully. The procedure requires two LMI (Linear Matrix Inequalities) solutions at each time step, one at the end points of interpolation and the other within the interpolant itself. LMI solutions are computationally expensive with the existing level of technology, and the computation time is too high for real time operation, therefore this algorithm could not be implemented in the integrated simulation. However, standalone solutions are obtained for the same parameters with the simulation model using MATLAB-LMI Control Toolbox [65]. A comprehensive coverage on LMIs in control theory can be found in Ref. [66].

4.5 Control Allocation

Control allocation module distributes virtual commands to highly coupled control effectors in real time. The limits and failure conditions of the each individual effector are taken into account in an efficient way during allocation. Failure detection is assumed to be accomplished by other mechanisms, which are beyond the scope

of this work, but failure isolation is a direct outcome of the implemented allocation method.

All control allocation schemes need the B matrix, which is simply the relationship between virtual control commands and actual effector positions. In real applications this relationship may not be linear or time invariant, however except for the boundaries of the flight envelope, a fixed B is not a bad approximation. The B matrix used in this study is given in Table 4.8. It is calculated at the neutral position for each one of the effectors.

Table 4.8: B matrix used for control allocation

	1	2	3	4	5
	Left				
$\partial C_L/\partial\delta_i$	4.8E-03	4.3E-03	3.6E-03	0.0E+00	0.0E+00
$\partial C_R/\partial\delta_i$	3.7E-04	6.3E-04	8.4E-04	-1.8E-04	-2.4E-04
$\partial C_M/\partial\delta_i$	1.8E-03	2.4E-03	4.2E-03	0.0E+00	0.0E+00
$\partial C_N/\partial\delta_i$	-1.6E-05	-2.4E-05	-2.9E-05	-1.4E-04	-2.4E-04
	Right				
$\partial C_L/\partial\delta_i$	4.8E-03	4.3E-03	3.6E-03	0.0E+00	0.0E+00
$\partial C_R/\partial\delta_i$	-3.7E-04	-6.3E-04	-8.4E-04	1.8E-04	2.4E-04
$\partial C_M/\partial\delta_i$	1.8E-03	2.4E-03	4.2E-03	0.0E+00	0.0E+00
$\partial C_N/\partial\delta_i$	1.6E-05	2.4E-05	2.9E-05	1.4E-04	2.4E-04

Modern aircraft use digital control systems which operate in discrete time. Considering the actuator position and rate constraints which are defined as:

$$\delta_{min} \leq \delta \leq \delta_{max}, \left|\dot{\delta}\right| \leq \delta_{rate}$$

Then, these can simply be merged for the sampling period T, such that [39]:

$$\underline{u} \leq u \leq \overline{u}$$

where

$$\underline{u}(t) = \max[\delta_{min},\ u(t-T) - \delta_{rate}T] \tag{4.21}$$

$$\overline{u}(t) = \min[\delta_{max},\ u(t-T) + \delta_{rate}T] \tag{4.22}$$

Cascaded Generalized Inverse (CGI) [12][67]:

An algorithm using the CGI scheme has been implemented and tested in the integrated simulation. The steps of this procedure is explained in Table 4.9.

Table 4.9: Procedure for Cascaded Generalized Inverse

Step	
1	A single generalized inverse is used for calculating control effector command, using (3.1).
2	If any control effector command is beyond saturation: $u_i < \underline{u}$ or $u_i > \overline{u}$, then:
3	The saturated controls are left at limits, and removed from the problem. $u_i = \underline{u}$ *or* $u_i = \overline{u}$
4	The columns of B associated with the saturated controls are removed to obtain B'
5	Total virtual commands v_{sat} associated with the saturated controls u_i is calculated: $$v_{sat} = \sum_{i=1}^{n_{sat}} B_i u_i$$
6	The saturated virtual command is removed from the original. $\hat{v} = v - v_{sat}$
7	The control allocation process is repeated with the reduced problem: $B'\hat{u} = \hat{v}$
8	The process is repeated until v is satisfied or no control effectors are left to allocate.

Weighted Least Squares (WLS):

As a second approach, WLS formulation (3.2) for the control allocation problem has been solved using an Active Set Algorithm [13][67]. Active set algorithm uses the

set of active constraints which is called Working Set-$\mathcal{W}$. In each iteration step some of the inequality constraints are regarded as equality constraints, and are included in the working set, while the remaining inequality constraints are disregarded. The working set at the optimum is known as the active set of the solution. For a generic bounded least squares problem:

$$\min_{u} \|Au - b\|$$

with constraints;

$$Bu = v, \quad Cu \geq U$$

considering our control allocation problem, C and U becomes:

$$C = \begin{bmatrix} I \\ -I \end{bmatrix}, U = \begin{bmatrix} \underline{u} \\ -\overline{u} \end{bmatrix}$$

re-writing the cost function for WLS:

$$\|W_u(u - u_p)\|^2 + \gamma\|W_v(Bu - v)\|^2 = \left\| \begin{pmatrix} \gamma W_v B \\ W_u \end{pmatrix} u - \begin{pmatrix} \gamma W_v v \\ W_u u_p \end{pmatrix} \right\|^2$$

the solution for u can be found by applying the procedure given in Table 4.10.

For more comprehensive information on the algorithms presented here, on different variants of these and on other control allocation algorithms as well, the reader is suggested to refer Ref. [67].

4.6 Simulation Results

In this section the response of the integrated model, composed of the controller, control allocation module and the nonlinear plant, to selected inputs will be studied in detail. The performance of the implemented modules will be analyzed first for the nominal case where no malfunction occurs, and then with different malfunction

Table 4.10: Procedure for Weighted Least Squares

Step	
1	Let u^0 be a feasible starting point and let the working set $\mathcal{W}$ contain the active inequality constraints at u^0.
2	For each iteration k, for given u^k, solve: $$\min_p \left\| A\left(u^k + p\right) - b \right\| , \quad Bp = 0, \quad p_i = 0, \quad i \in \mathcal{W}$$
3a	If u^k+p is feasible, set u^{k+1}=u^k+p and compute the Lagrange multipliers, (μ, λ), where μ is associated with equality constraints, and λ is associated with active inequality constraints.
3b	If not, determine the maximum step length α such that $u^{k+1} = u^k + \alpha p$ is feasible. Add the primary bounding constraint to the working set.
4a	If all $\lambda \geq 0$, u^{k+1} is the optimal solution, so exit the loop.
4b	If not remove the constraint associated with the most negative λ with the working set.

scenarios. For all cases, a similar set of plots will be presented to make the comparison easier. The parameters displayed on each plot is given in Table 4.11.

All simulations are performed in MATLAB-Simulink with fixed 10ms time step using ODE5 Dormand-Prince integrator. The time history plots start five seconds later than the simulation start, as the first few seconds are allocated for stabilization. A single LTI controller model corresponding to the selected design point is used in each run. No relative weighting is applied among control effectors.

Nominal Case:

The response of the aircraft to doublet shaped commands is investigated in this part. The total simulation time is set to 100 sec. The airspeeds are selected as 150, 200, 250 KIAS corresponding to low, medium and high speed flight. In each run, the doublets are applied in only single input channels, others left neutral. WLS method is used for control allocation. The timing of the doublet is shown in Figure 4.36. Four different maneuvers are executed as listed in Table 4.12.

Table 4.11: Parameters plotted

Symbol	Unit	Parameter
IAS	kts	Indicated airspeed
θ, α	deg	Pitch attitude, Angle of attack
ϕ	deg	Roll attitude
β	deg	Angle of sideslip
p, q, r	deg/s	Body axis rotational rates for roll, pitch, yaw
N_y, N_z	g	Body axis accelerations for lateral, vertical
$\upsilon_{pitch}, \upsilon_{roll}, \upsilon_{yaw}$	deg	Virtual controller commands for pitch, roll, yaw
υ_{thr}	pct	Virtual controller command for throttle
$\delta_{L1}, \delta_{L2}, \delta_{L3}$	deg	Control effector angles for left flaps
$\delta_{R1}, \delta_{R2}, \delta_{R3}$	deg	Control effector angles for right flaps
δ_{L4}, δ_{L5}	deg	Control effector angles for left spoilers
δ_{R4}, δ_{R5}	deg	Control effector angles for right spoilers

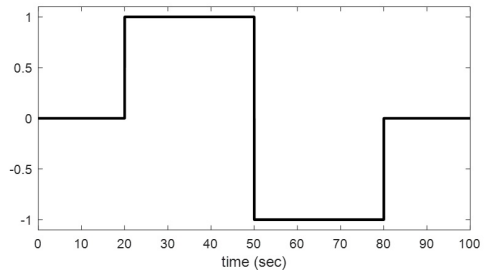

Figure 4.36: Doublet command

Table 4.12: Maneuver codes

No.	Maneuver	Amplitude
1	Pitch doublet	3°
2	Roll doublet	30°
3	IAS doublet	9 kts
4	Sideslip doublet	3°

Only mid speed results will be presented in the main body of this thesis and other plots are going to be left for Appendix B.

200 KIAS - Pitch Doublet:

Time history plots for pitch doublet input are given in Figures 4.37 and 4.38.

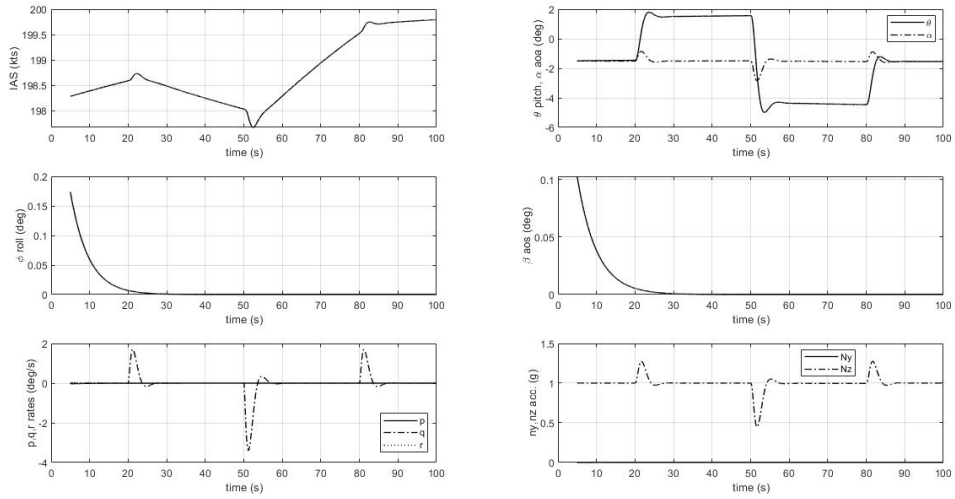

Figure 4.37: Nominal, 200 KIAS, pitch doublet, plot 1

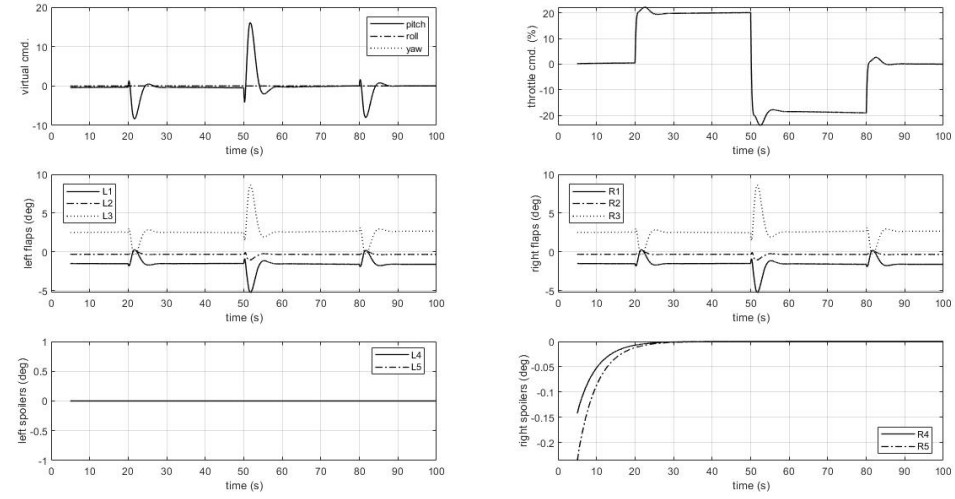

Figure 4.38: Nominal, 200 KIAS, pitch doublet, plot 2

Pitch attitude response is smooth and shows no objectionable discrepancy with respect to the reference model. Airspeed change is within 3 kts throughout the maneuver. Angle of attack is kept constant, therefore pitch movement is expected to end up with a change in glide slope angle. No change is observed in off axis. Control activity is smooth, a maximum deflection of about 7° is observed in outboard flaps at the midpoint of doublet. ±20% throttle activity is used for keeping airspeed constant during the maneuver.

200 KIAS - Roll Doublet:

Time history plots for roll doublet input are given in Figures 4.39 and 4.40.

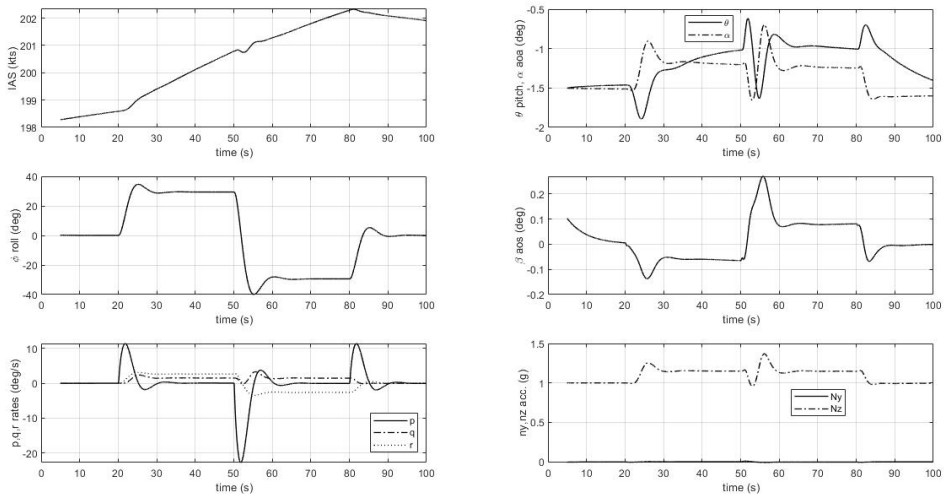

Figure 4.39: Nominal, 200 KIAS, roll doublet, plot 1

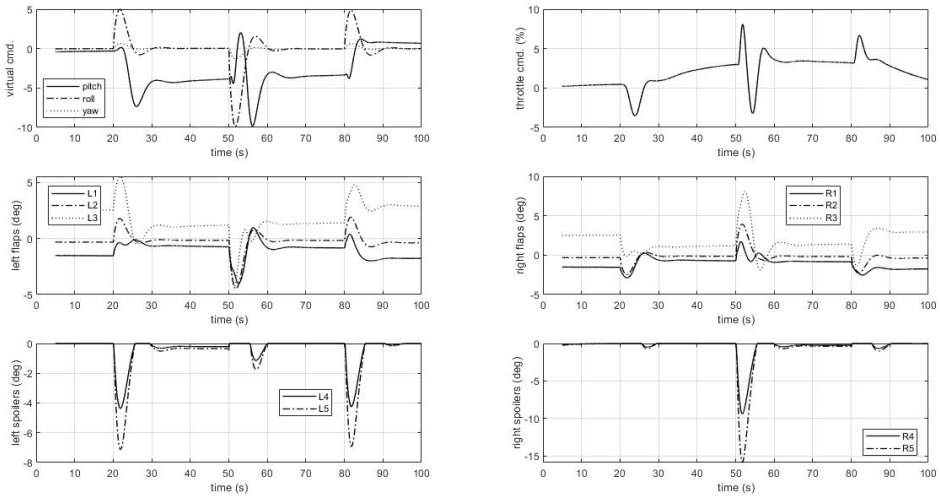

Figure 4.40: Nominal, 200 KIAS, roll doublet, plot 2

Roll attitude response is smooth and shows no objectionable discrepancy with respect to the reference model. Angle of sideslip and lateral acceleration is kept close to zero, so the turn is coordinated. Airspeed change is within 3 kts throughout the maneuver. Change in pitch attitude and angle of attack is within 1° each. Control activity is smooth, a maximum deflection of about 6° is observed in right outboard flap at the midpoint of doublet. Spoiler deflections reaching up to 15° is observed. Throttle activity is within 10% during the maneuver.

200 KIAS - Airspeed Doublet:

Time history plots for airspeed doublet input are given in Figures 4.41 and 4.42.

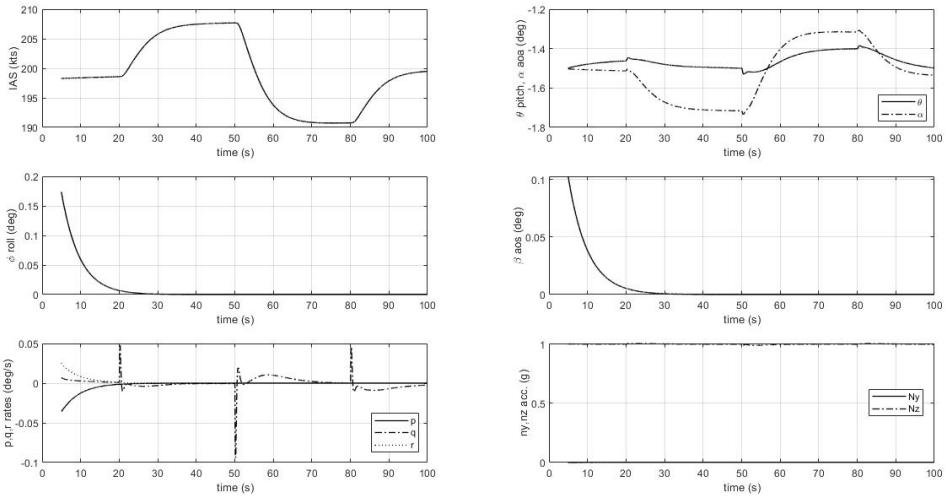

Figure 4.41: Nominal, 200 KIAS, speed doublet, plot 1

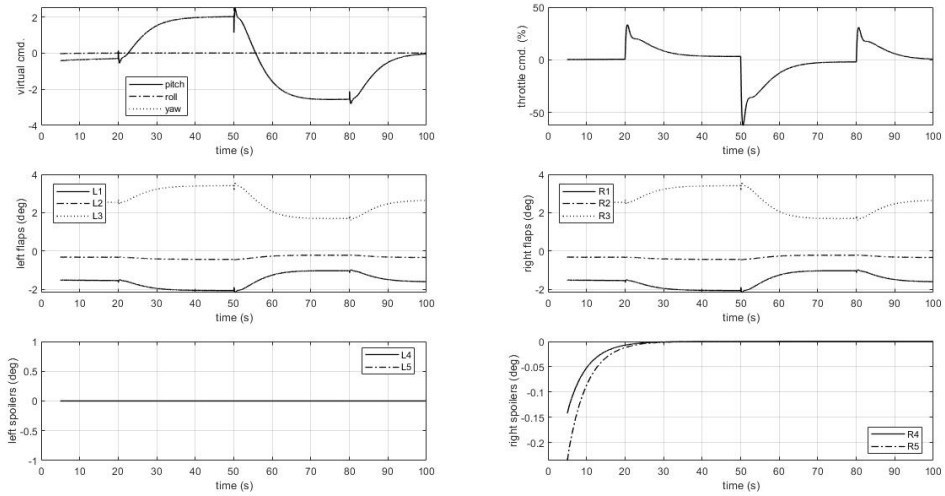

Figure 4.42: Nominal, 200 KIAS, speed doublet, plot 2

Airspeed response shows no objectionable discrepancy with respect to the reference model. Both pitch attitude and angle of attack change is within 0.2°, therefore

no significant change in glide slope. No change is observed in off axis, as well. Control activity is smooth, a maximum deflection of about 2° is observed in outboard flap. +20%, -60% maximum throttle commands are observed.

200 KIAS - Sideslip Doublet:

Time history plots for sideslip doublet input are given in Figures 4.43 and 4.44.

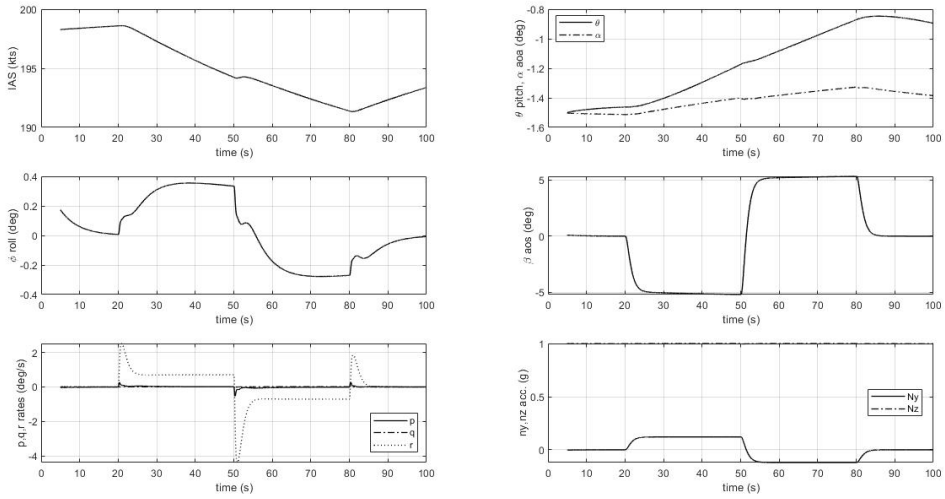

Figure 4.43: Nominal, 200 KIAS, sideslip doublet, plot 1

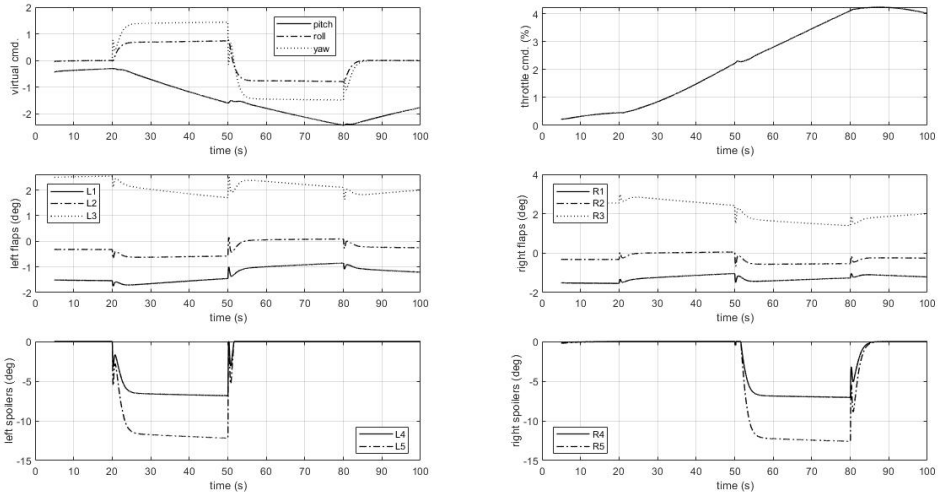

Figure 4.44: Nominal, 200 KIAS, sideslip doublet, plot 2

Angle of sideslip response is smooth and shows no objectionable discrepancy with respect to the reference model. A slight change in roll attitude is evident. Airspeed change is close to 7-8 kts throughout the maneuver. Change in pitch attitude and angle of attack is less than 0.6°. Flap activity is quite limited. The sideslip is generated by continuous deflection of spoilers by less than 15°.

Now, the response of the simulation model to a series of step inputs will be investigated. The main purpose is to compare the performance of the two control allocation methods - CGI and WLS - in case of failed control effectors. In order to generate a baseline, first no failure runs are performed for each of the control allocation methods. All cases are run at 200 KIAS and the total simulation time is set as 100 sec. The sequence of the step inputs in roll, pitch and sideslip is displayed in Figure 4.45.

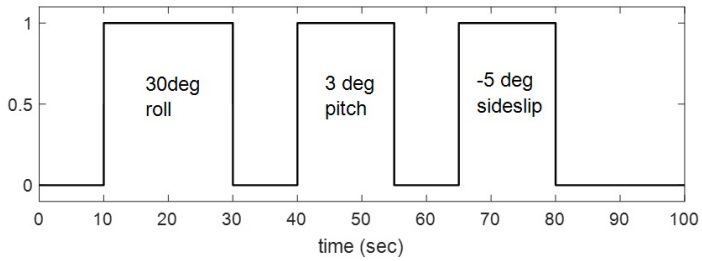

Figure 4.45: Sequence of step commands

CGI & WLS Baselines:

Time history plots for CGI and WLS baselines are given in Figures 4.46, 4.47 and Figures 4.48, 4.49.

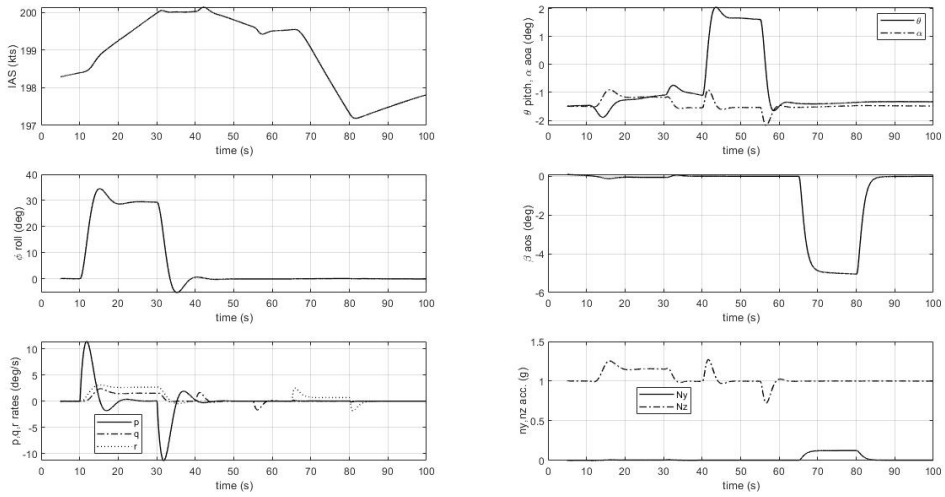

Figure 4.46: CGI no failure baseline, plot 1

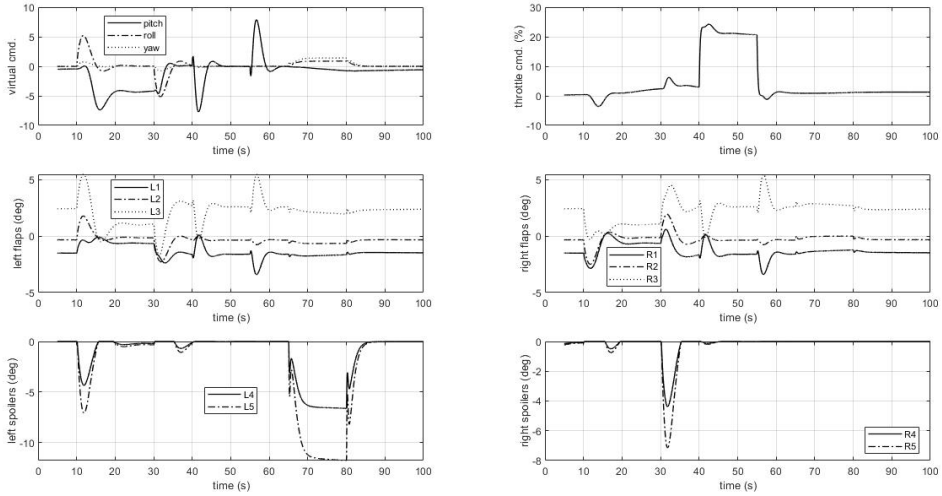

Figure 4.47: CGI no failure baseline, plot 2

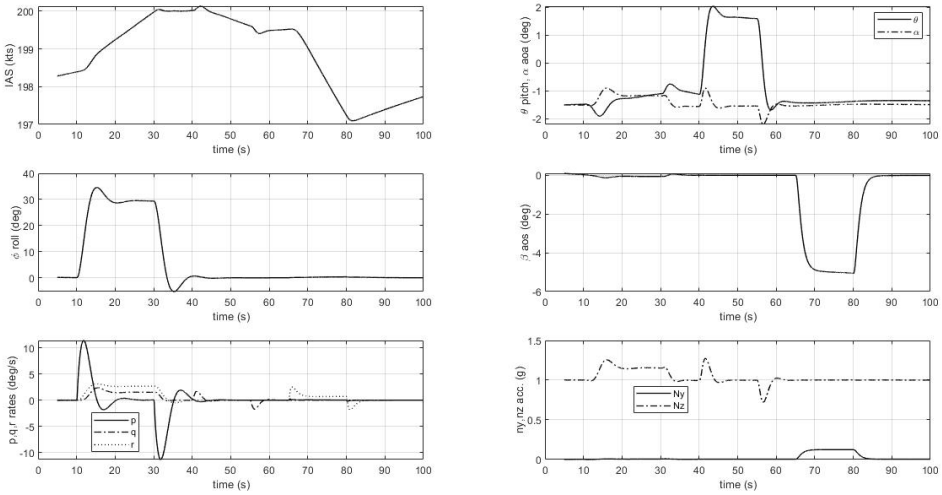

Figure 4.48: WLS no failure baseline, plot 1

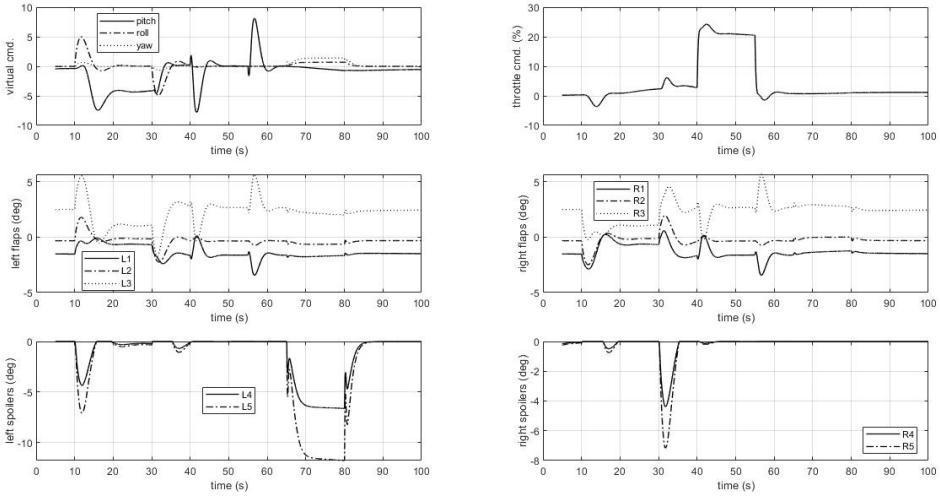

Figure 4.49: WLS no failure baseline, plot 2

A comparison of the Figure 4.46 with Figure 4.48 and Figure 4.47 with Figure 4.49 reveals no distinguishable difference. This means that, in nominal (no failure) case, the response and control activity using CGI and WLS control allocation methods are similar. As the same controller is used in both cases and both control allocation approaches tend to minimize control activity by keeping each effector angle close to zero as possible, and no position or rate saturation or failure occurs during the maneuver, two methods practically function similarly.

Malfunction Case:

The malfunctions applied in this simulation work are in two general types: jamming and hard over. Jamming malfunctions are implemented by injecting a zero rate limit in WLS allocation, and by injecting a fixed zero position in CGI allocation. For hard over, a fixed maximum deflection is injected in both CGI and WLS. The effected actuators are considered to be in their failure conditions throughout the

simulation time, except for the intermittent jamming case. The designation of the simulated malfunction scenarios are given in Table 4.13.

Table 4.13: Malfunction scenarios

	L					R					Notes
	5	4	3	2	1	1	2	3	4	5	
1					x						Jamming of flap with highest lift effectiveness
2			x		x						Jamming of flaps with highest lift and roll eff.
3	x		x		x						Jamming of effectors with highest lift, roll, and yaw eff.
4	x		x		x						Intermittent jamming of two flaps and one spoiler
5				x							Hard over of one flap
6		x									Hard over of one spoiler

The time history plots of all the malfunction runs - except for #3 - with both CGI and WLS control allocation are given in Appendix B. Plots of malfunction#3 will be presented here.

Malfunction#3 - CGI Control Allocation:

Time history plots for Malfunction#3 with CGI control allocation are given in Figures 4.50 and 4.51.

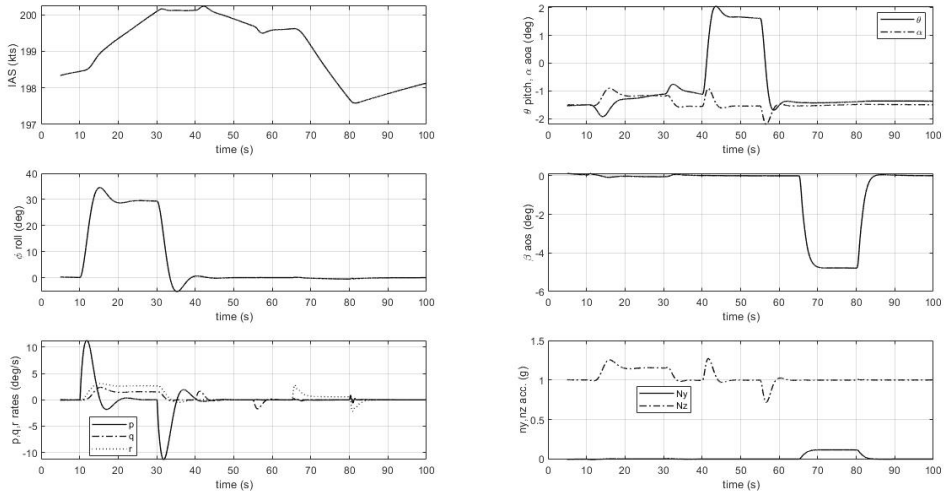

Figure 4.50: Malf#3, CGI control allocation, plot 1

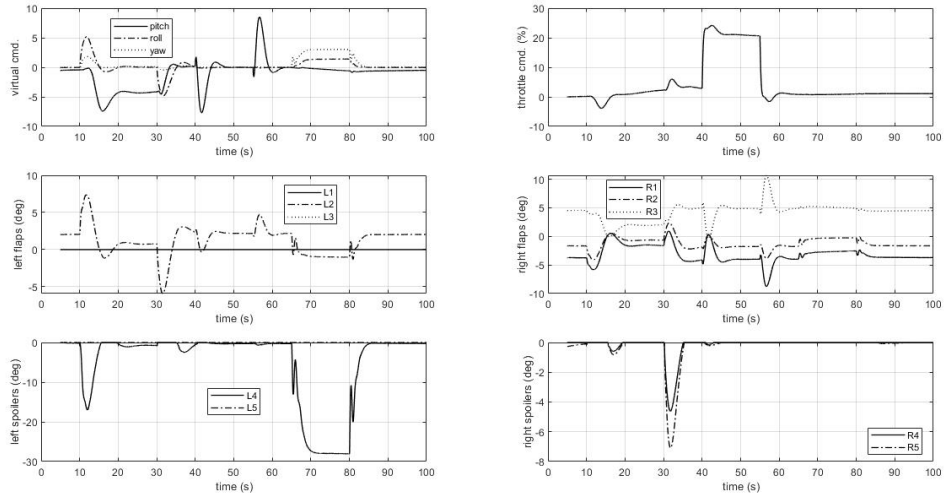

Figure 4.51: Malf#3, CGI control allocation, plot 2

Left inboard flap, left outboard flap and left outboard spoiler is jammed at neutral position. The malfunctions do not seem to cause a significant degradation in

system performance with CGI allocation. The system tracks the reference commands in a smooth fashion without excessive use of remaining control effectors.

Malfunction#3 - WLS Control Allocation:

Time history plots for Malfunction#3 with WLS control allocation are given in Figures 4.52 and 4.53.

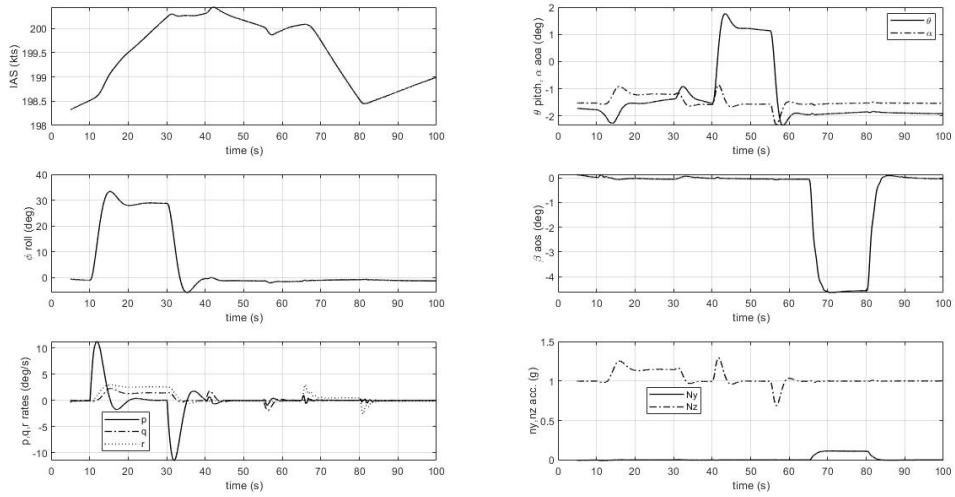

Figure 4.52: Malf#3, WLS control allocation, plot 1

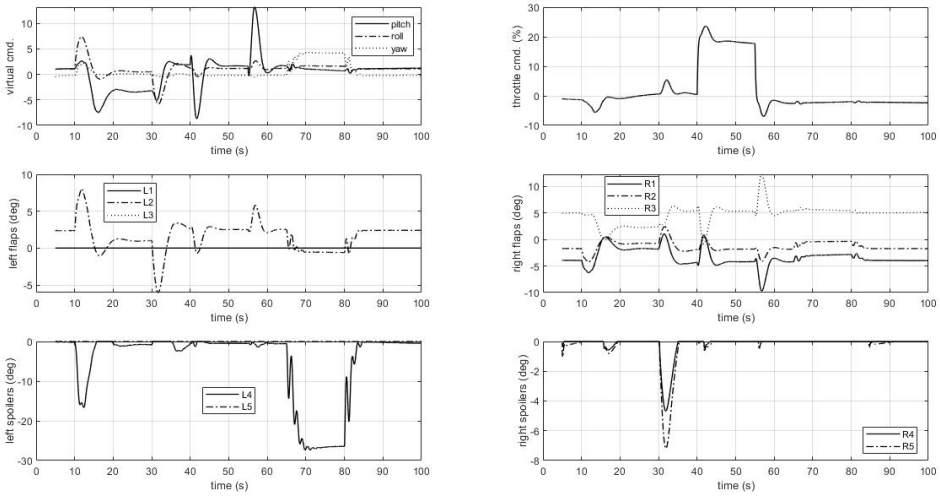

Figure 4.53: Malf#3, WLS control allocation, plot 2

The malfunctions do not seem to cause a significant degradation in system performance with WLS allocation as well. Use of control effectors looks slightly higher compared to CGI case.

None of the malfunction tests, even with the ones with three failed actuators, resulted with significant degradation in tracking performance of reference commands. However, in some of the tests, a small difference between the results of models using CGI and WLS is observed. At the first glance, the tracking performance seems smoother in CGI, and the effector usage is slightly less. But, when investigated deeper, this difference is associated with the different convergence time of the two control allocation modules in initialization phase. The control allocation modules need a number of seconds to stabilize when a new simulation run starts. In most scenarios, the malfunctions are effective from the very first instance of simulation. Especially in multi-effector failure cases, the stabilization time becomes considerably higher in WLS compared to CGI (Figure 4.54 and Figure 4.55) and so the system

initial states deviate more from trim. Time history plots of control effector angles in the first five seconds of the simulation are given in Figures 4.54, 4.55, 4.56 and 4.57.

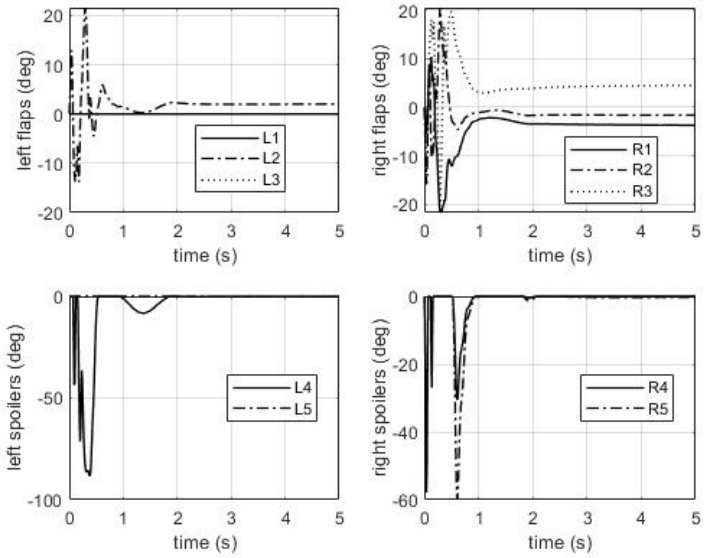

Figure 4.54: Effector angles, first 5 seconds, Malf#3 with CGI

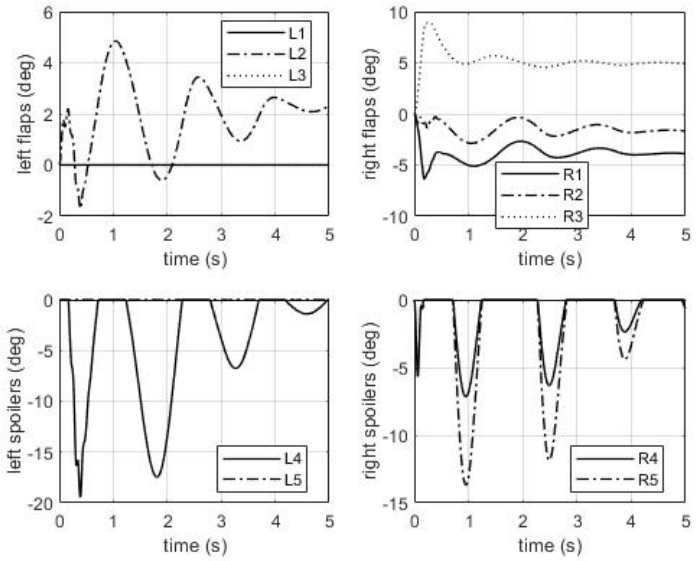

Figure 4.55: Effector angles, first 5 seconds, Malf#3 with WLS

In cases where the malfunctions are injected later in the simulation, as in the case of malfunction#4, the stabilization time is smaller (Figure 4.56 and Figure 4.57), and consequently the performance of the two modules are close throughout the maneuver.

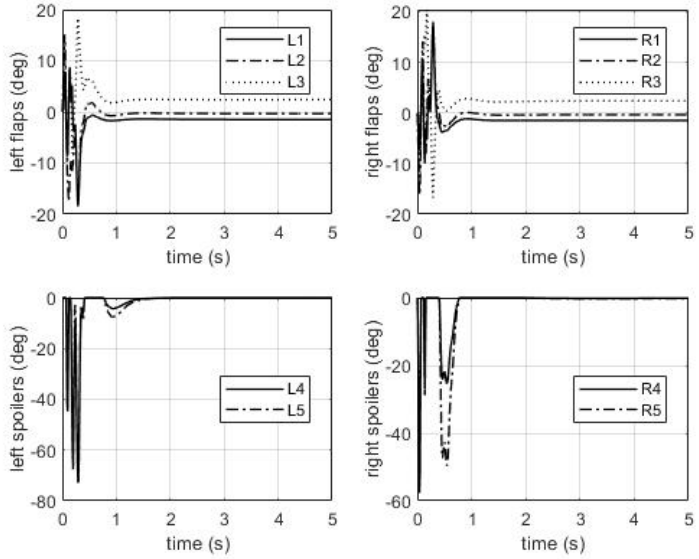

Figure 4.56: Effector angles, first 5 seconds, Malf#4 with CGI

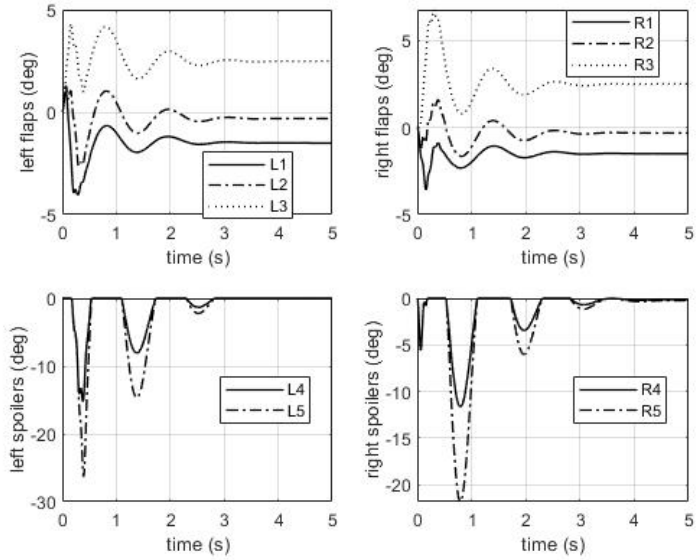

Figure 4.57: Effector angles, first 5 seconds, Malf#4 with WLS

Slower initial stabilization of WLS is considered to be related with its more complicated structure and its dependence on the parameters calculated in the previous time step, which imposes a small delay in the system.

Actuator position and rate limits can be merged (4.21) to form the effector limits, which was discussed in Chapter 4.5. This is the approach followed in this work for WLS allocation. However, as for CGI, this method did not perform well, so only position limits could be implemented. In all the simulation work presented, CGI allocation involves only position limits, whereas WLS involves both position and rate limits.

In this part a comparison of CGI and WLS, when rate limits are implemented in both is presented. An off-line simulation of two control allocation modules with a given virtual command input is performed. In this example the rate limit for the left outboard flap (L3) is set to 1°/sec. An 80 sec part of time history plots for left flap angles is presented in Figure 4.58.

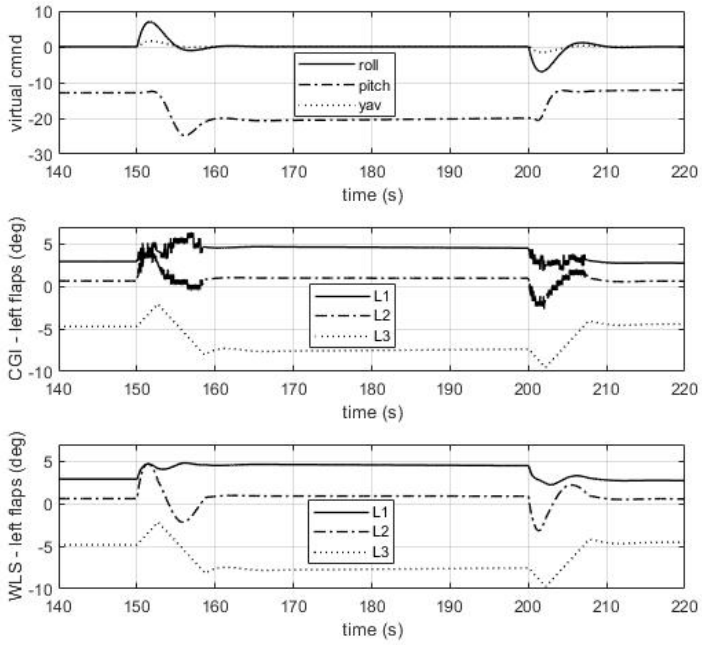

Figure 4.58: CGI & WLS performance in case of limited rate

As can be followed, rate limited change of L3 around 150 sec and 200 sec is evident for both CGI and WLS. In the same periods, a chatter occurs in L1 and L2 for CGI case. In WLS, limited rate in L3 does not seem to trigger any anomalies in other effectors.

Uncertain Plant:

The performance of the control system in the case of plant uncertainties is investigated in this part. Some aerodynamic parameters and servo transfer function gains are assumed to be deviated from their nominal values as defined in the following. The values of the deviation parameters are given in Table. 4.14.

$$(C_D)_{unc} = (C_D)_{nom} + \Delta C_D$$

$$(C_L)_{unc} = (C_L)_{nom} + k_L C_{L_\alpha} \alpha + \Delta C_L$$

$$(C_M)_{unc} = (C_M)_{nom} + k_M C_{M_\alpha} \alpha + \Delta C_M$$

$$(C_{Y_\beta})_{unc} = k_Y (C_{Y_\beta})_{nom}$$

$$(C_{N_\beta})_{unc} = k_N (C_{N_\beta})_{nom}$$

$$(TF_{servo})_{unc} = k_{servo} (TF_{servo})_{nom}$$

Table 4.14: Deviation from the nominal model

Parameter	Deviation
ΔC_D	+0.0050
ΔC_L	-0.02
ΔC_M	-0.02
k_L	-10%
k_M	-10%
k_Y	-5%
k_N	-5%
k_{servo}	-10%

A comparison for the tracking performance of the nominal plant with the deviated plant is given in Figure 4.59, and a comparison of the virtual commands generated is given in Figure 4.60.

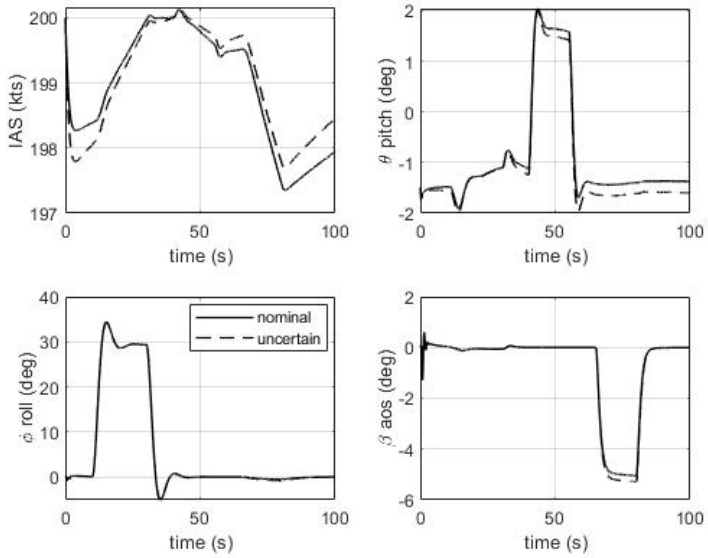

Figure 4.59: Tracking performance for nominal and uncertain models

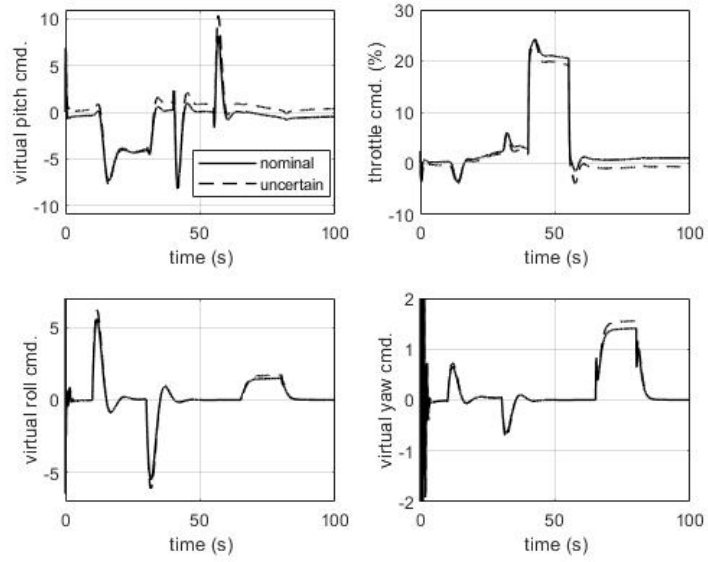

Figure 4.60: Virtual commands for nominal and uncertain models

This single simulation shows that, even in a case of major deviation from the anticipated model parameters, the system tracking performance is slightly effected. However, for a thorough investigation, a Monte-Carlo analysis should be performed by taking all the uncertainties into account.

CHAPTER 5

DISCUSSION

In this thesis, a study about the application of modern robust control design techniques on a tailless UAV, RQ-3 Darkstar, is presented. First, a 6-DoF nonlinear mathematical model for the vehicle is constructed using a semi empirical aerodynamic prediction software: Digital DATCOM. A number of LTI plant models have been generated for selected operating points, based on this nonlinear model. Secondly, control laws are synthesized for plant models generated at different operating points using μ-synthesis design method. Three different approaches are investigated to schedule the control laws: ad-hoc linear interpolation, synthesis using simplified LPV models and stability preserving interpolation. Then, a control allocation module is implemented to distribute the virtual commands, which are generated by the control laws, to highly coupled control effectors in real time. Effector limits and failure conditions are taken into account in an efficient way during allocation. Two different allocation approaches are investigated: CGI and WLS. Finally, a simulation study is performed using the nonlinear aircraft model, control laws and control allocation models for various maneuvers and control effector failure cases.

Time history plots show that, for almost all maneuvers, the on-axis responses are smooth and do not deviate from the reference model significantly. The only exception is a small beat in pitch virtual commands at the corners of the longitudinal doublets. This is associated with a single small RHS zero apparent in the closed-loop transfer function from pitch command. In the off-axis, a deviation in airspeed is evident, especially for sideslip commands. This is considered to be related with the

coupling between longitudinal and lateral directional axes in 6-DoF equations of motion. Coupling problem can be solved by using cross feed terms between longitudinal and lateral-directional controllers in more simple designs, but for our multivariable method this is not applicable. A coupled synthesis using a full-state plant is considered as a reasonable solution for this problem. However, as the system order will be much higher compared to decoupled case, the controller order and the level of difficulty for implementation will be even higher, in this solution. This work can be extended in the future by a full-state synthesis, taking aerodynamic and inertial coupling terms into account.

Three interpolation methods are investigated for μ-synthesis. Ad-hoc linear interpolation is simple from implementation point, but it gives no stability guarantees for the interpolated systems. Even if the stability is preserved, robust performance is shown to be much lower. Synthesis using simplified LPV models method is again simple for implementation and gives much better robust performance, however, it is applicable only between each two operating points and the problem of stitching all the operating intervals still remains. Stability preserving interpolation is not simple but straightforward for implementation, however it requires solution of LMIs at each time step, which is considered not to be feasible with the available solvers. An extension to this work can be done by investigating LFT synthesis based interpolation techniques, in which a considerable amount of literature is available.

The malfunction simulations show no significant degradation in system performance for the two allocation methods implemented. Even for cases with three failed actuators, the system tracks the reference commands in a smooth fashion without excessive use of remaining control effectors. WLS is observed to need a longer initial stabilization period compared to CGI. However, this is considered to be a simulation

problem. In conditions where effector rate is limited, CGI is observed to result in a small amplitude chatter, whereas, WLS is observed to perform well.

In this study, the controller is designed to create virtual commands in three rotational axes: roll, pitch and yaw with no change in lift. However, for the flying wing configuration, it is quite possible to create direct lift by using the effectors in a proper sense. The structure of the control allocation module is designed compatible for direct lift commands. A future study which makes use of this extra degree of freedom is considered to be a very interesting extension to the current work.

The relationship between virtual control commands and actual effector positions, the B matrix, is assumed to be constant in control allocation algorithms, and the uncertainty in it is associated with the process uncertainty in controller synthesis. A method for implementing the change in B matrix can be studied as an extension to the current work. As a result, the robustness of the system can be improved due to the decrease in process uncertainty.

In μ-synthesis, further information about contributions of robust stability and nominal performance requirements on the overall robust performance goal can be extracted by decomposing closed-loop system matrix, M, into various constituents. A systematic investigation of these contributions is considered to be an interesting extension to the present work.

APPENDIX A

DATCOM INPUT CARDS

A.1 Datcom input card for symmetric flap

```
DIM M
DAMP
     $FLTCON
     STMACH=0.9, TSMACH=1.02,
     NMACH=1.0, MACH(1)=0.4,
     NALT=1.0, ALT=0.0,
     NALPHA=20.0,
     ALSCHD(1)=-8.0, -6.0, -4.0, -2.0, -1.0, 0.0, 1.0, 2.0, 4.0, 6.0, 8.0,
10.0, 12.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0,
     LOOP=1.0$
     $SYNTHS
     XCG=2.965, ZCG=0.00,
     XW=2.672, ZW=0.00, ALIW=1.00$
     $BODY NX=9.0,
     X(1)=0., 0.259, 0.958, 1.506, 2.157, 2.834, 3.539, 3.854, 4.247,
     ZU(1)=0., 0.082, 0.437, 0.587, 0.651, 0.591, 0.385, 0.239, 0.0,
     ZL(1)=0., -0.050, -0.168, -0.226, -0.246, -0.219, -0.138, -0.083, 0.0,
     P(1)=0., 2.594, 4.932, 5.396, 5.488, 5.396, 5.116, 4.963, 0.0,
     R(1)=0., 0.638, 1.143, 1.210, 1.210, 1.210, 1.210, 1.210, 0.0,
     S(1)=0., 0.133, 1.086, 1.544, 1.704, 1.540, 0.994, 0.613, 0.0,
     METHOD=1.0$
     $WGPLNF CHRDTP=0.641, CHRDR=1.575, SSPN=10.515, SSPNE=9.075, SAVSI=-13.1,
     DHDADI=0.0, TYPE=1.0$
     $SYMFLP FTYPE=1.0, NDELTA=9.0,
```

```
     DELTA(1)= -25., -20., -15., -5., 0., 5., 15., 20., 25.,
     PHETE=0.117, CHRDFI=0.272, CHRDFO=0.227,
     SPANFI=5.459, SPANFO=7.475$
     $WGSCHR DWASH=1.0$
NACA-W-6-63-416
CASEID DARKSTAR 00100 SYMFLP - FLAP
NAMELIST
BUILD
NEXT CASE
```

A.2 Datcom input card for asymetric flap

```
DIM M
DAMP
     $FLTCON
     STMACH=0.9, TSMACH=1.02,
     NMACH=1.0, MACH(1)=0.4,
     NALT=1.0, ALT=0.0,
     NALPHA=20.0,
     ALSCHD(1)=-8.0, -6.0, -4.0, -2.0, -1.0, 0.0, 1.0, 2.0, 4.0, 6.0, 8.0,
10.0, 12.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0,
     LOOP=1.0$
     $SYNTHS
     XCG=2.965, ZCG=0.00,
     XW=2.672, ZW=0.00, ALIW=1.00$
```

```
     $BODY NX=9.0,
     X(1)=0., 0.259, 0.958, 1.506, 2.157, 2.834, 3.539, 3.854, 4.247,
     ZU(1)=0., 0.082, 0.437, 0.587, 0.651, 0.591, 0.385, 0.239, 0.0,
     ZL(1)=0., -0.050, -0.168, -0.226, -0.246, -0.219, -0.138, -0.083, 0.0,
     P(1)=0., 2.594, 4.932, 5.396, 5.488, 5.396, 5.116, 4.963, 0.0,
     R(1)=0., 0.638, 1.143, 1.210, 1.210, 1.210, 1.210, 1.210, 0.0,
     S(1)=0., 0.133, 1.086, 1.544, 1.704, 1.540, 0.994, 0.613, 0.0,
     METHOD=1.0$
     $WGPLNF CHRDTP=0.641, CHRDR=1.575, SSPN=10.515, SSPNE=9.075, SAVSI=-13.1,
     DHDADI=0.0, TYPE=1.0$
     $ASYFLP STYPE=4.0, NDELTA=9.0,
     DELTAL(1)= -25., -20., -15., -5., 0., 5., 15., 20., 25.,
     DELTAR(1)= -0., -0., -0., -0., 0., 0., 0., 0., 0.,
     PHETE=0.117, CHRDFI=0.272, CHRDFO=0.227,
     SPANFI=5.459, SPANFO=7.475$
     $WGSCHR DWASH=1.0$
NACA-W-6-63-416
CASEID DARKSTAR 00100 ASYMFLP - FLAP
NAMELIST
BUILD
NEXT CASE
```

A.3 Datcom input card for spoiler

```
DIM M
```

```
DAMP
     $FLTCON
     STMACH=0.9, TSMACH=1.02,
     NMACH=1.0, MACH(1)=0.4,
     NALT=1.0, ALT=0.0,
     NALPHA=20.0,
     ALSCHD(1)=-8.0, -6.0, -4.0, -2.0, -1.0, 0.0, 1.0, 2.0, 4.0, 6.0, 8.0,
10.0, 12.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0,
     LOOP=1.0$
     $SYNTHS
     XCG=2.965, ZCG=0.00,
     XW=2.672, ZW=0.00, ALIW=1.00$
     $BODY NX=9.0,
     X(1)=0., 0.259, 0.958, 1.506, 2.157, 2.834, 3.539, 3.854, 4.247,
     ZU(1)=0., 0.082, 0.437, 0.587, 0.651, 0.591, 0.385, 0.239, 0.0,
     ZL(1)=0., -0.050, -0.168, -0.226, -0.246, -0.219, -0.138, -0.083, 0.0,
     P(1)=0., 2.594, 4.932, 5.396, 5.488, 5.396, 5.116, 4.963, 0.0,
     R(1)=0., 0.638, 1.143, 1.210, 1.210, 1.210, 1.210, 1.210, 0.0,
     S(1)=0., 0.133, 1.086, 1.544, 1.704, 1.540, 0.994, 0.613, 0.0,
     METHOD=1.0$
     $WGPLNF CHRDTP=0.641, CHRDR=1.575, SSPN=10.515, SSPNE=9.075, SAVSI=-13.1,
     DHDADI=0.0, TYPE=1.0$
     $ASYFLP STYPE=1.0, NDELTA=4.0,
     DELTAS(1)= 0., 0.05, 0.1, 0.15,
     XSOC(1)= 0.80, 0.75, 0.7, 0.65,
     HSOC(1)= 0.07, 0.095, 0.12, 0.145,
```

```
     PHETE=0.117, XSPRME=0.65,
     SPANFI=8.988, SPANFO=10.500$
     $WGSCHR DWASH=1.0$
NACA-W-6-63-416
CASEID DARKSTAR 00001 ASYFLP - SPOILER
NAMELIST
BUILD
NEXT CASE
```

APPENDIX B

SIMULATION - TIME HISTORY PLOTS

In this appendix, time history plots of the simulation results are presented.

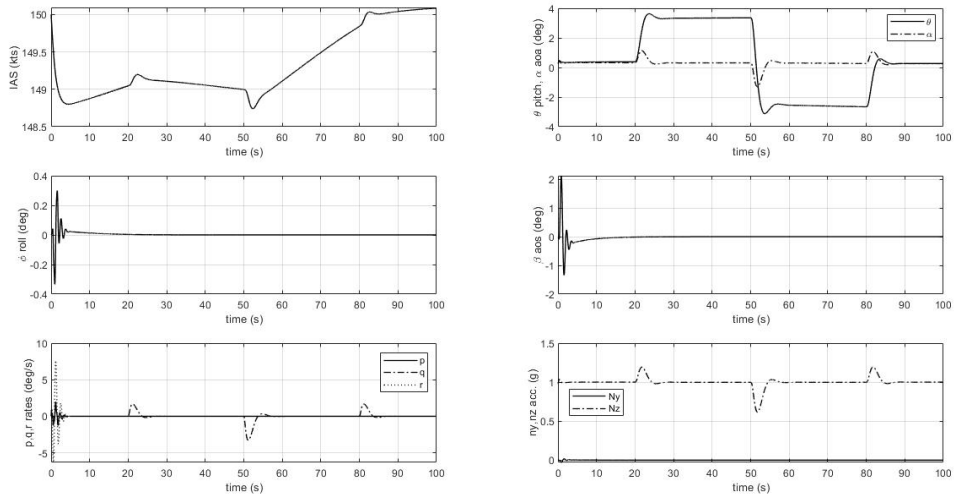

Figure B.1: Nominal, 150 KIAS, pitch doublet, plot 1

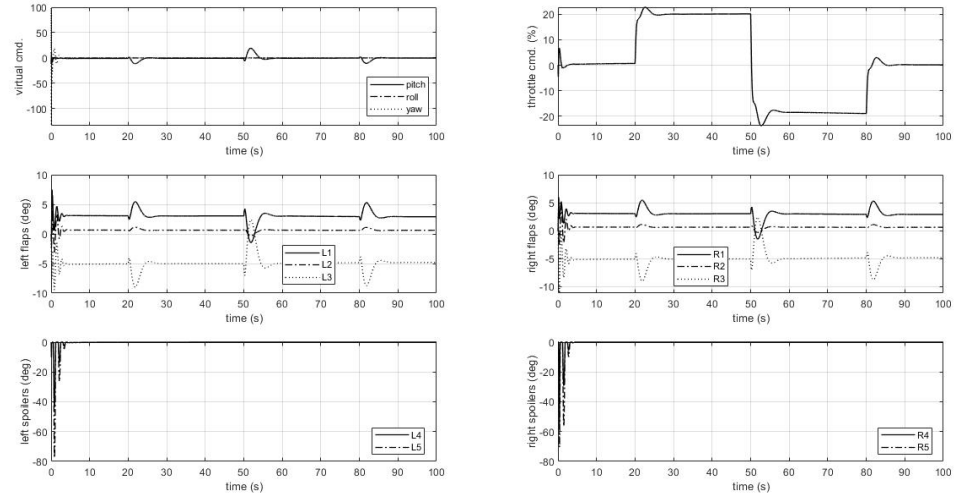

Figure B.2: Nominal, 150 KIAS, pitch doublet, plot 2

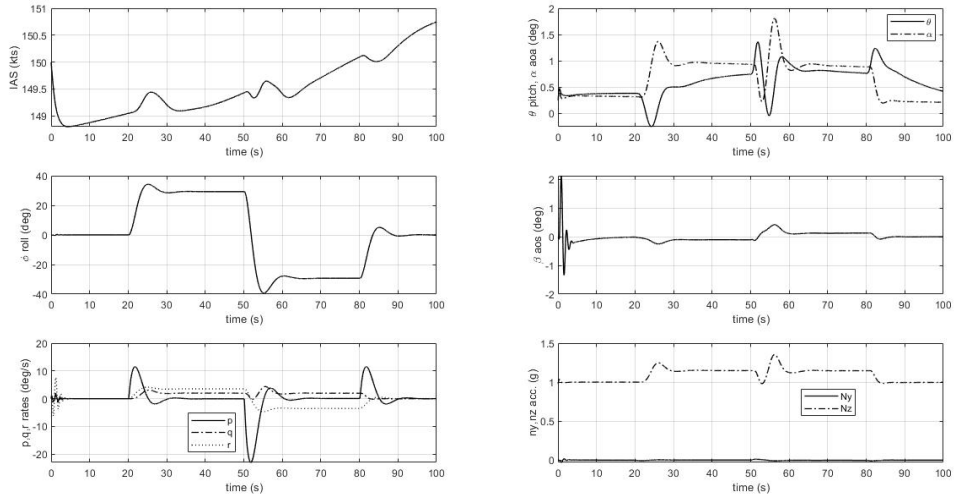

Figure B.3: Nominal, 150 KIAS, roll doublet, plot 1

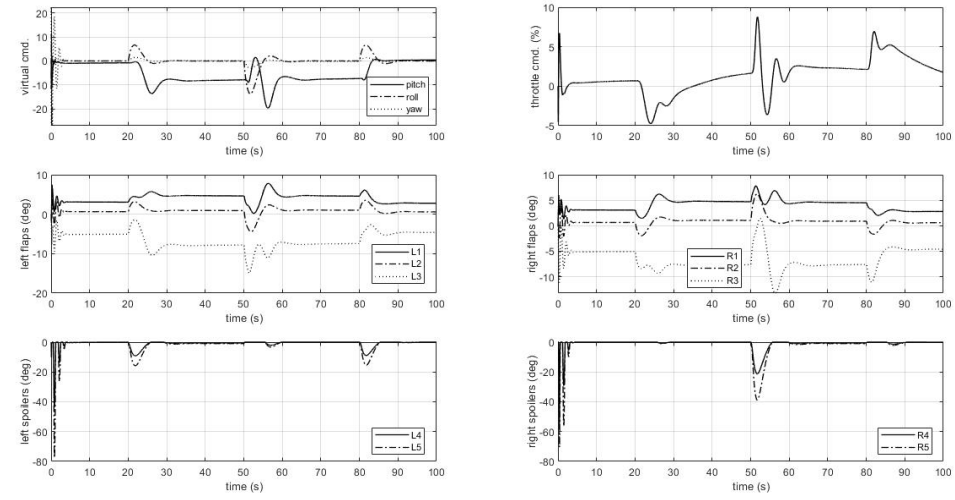

Figure B.4: Nominal, 150 KIAS, roll doublet, plot 2

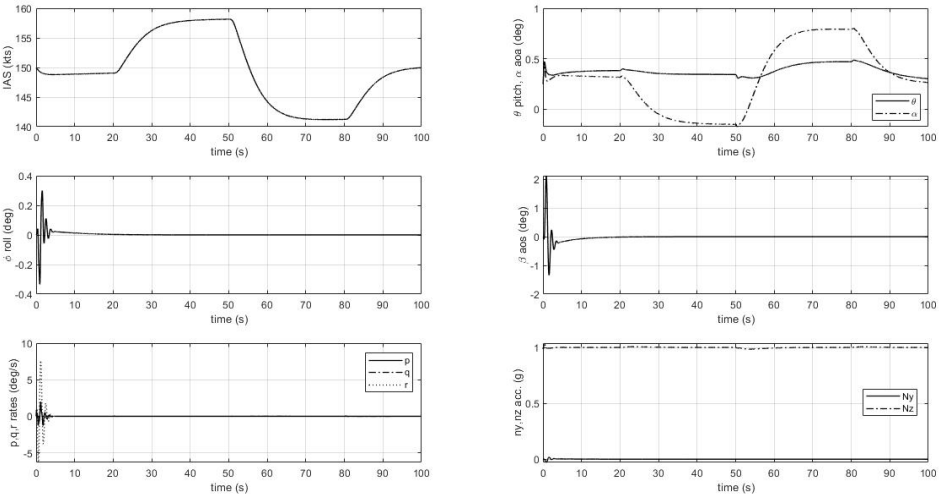

Figure B.5: Nominal, 150 KIAS, speed doublet, plot 1

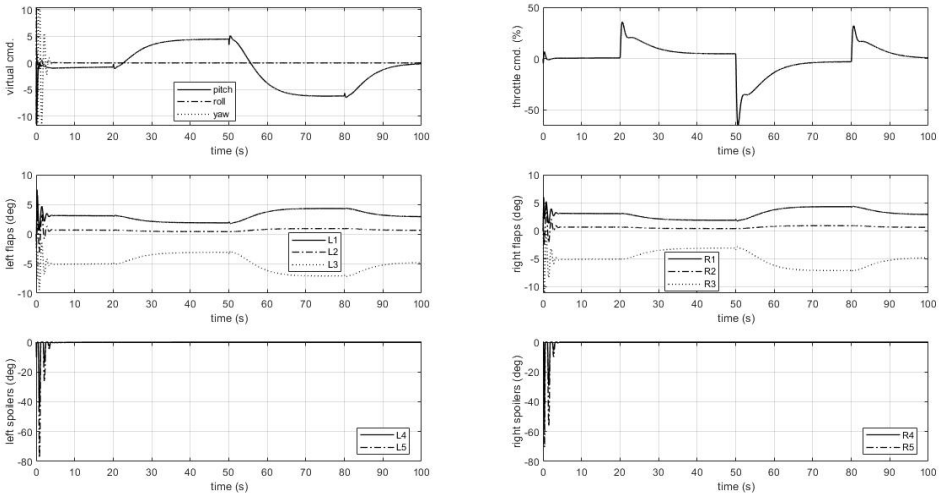

Figure B.6: Nominal, 150 KIAS, speed doublet, plot 2

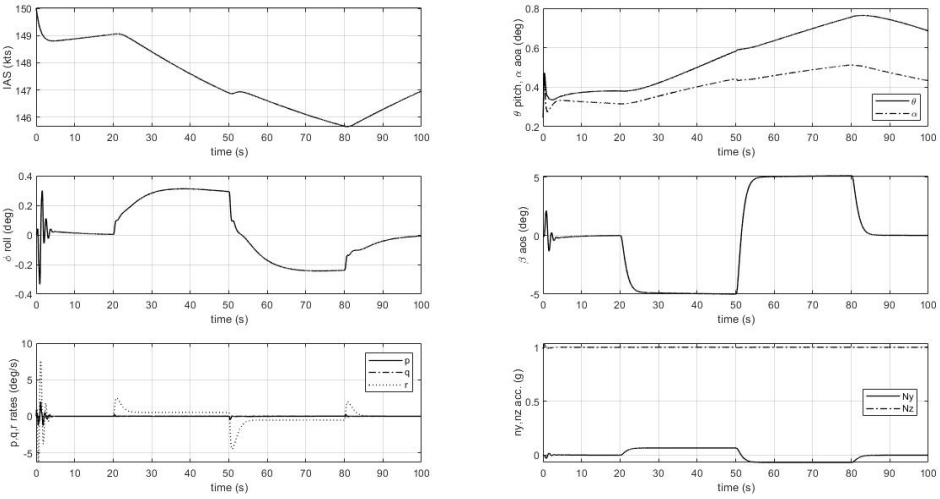

Figure B.7: Nominal, 150 KIAS, sideslip doublet, plot 1

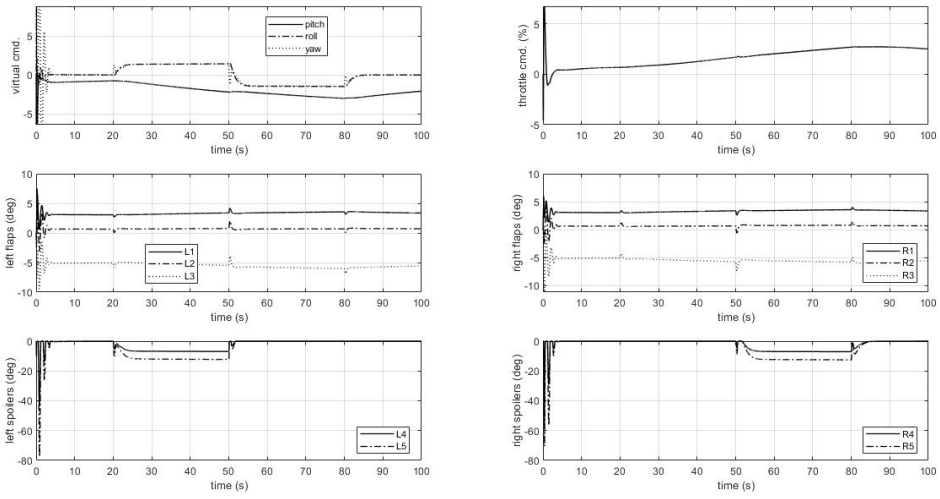

Figure B.8: Nominal, 150 KIAS, sideslip doublet, plot 2

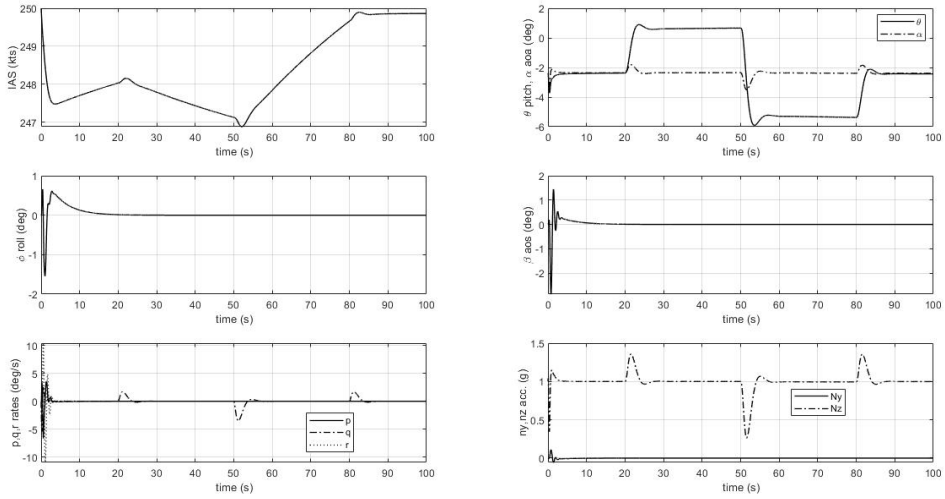

Figure B.9: Nominal, 250 KIAS, pitch doublet, plot 1

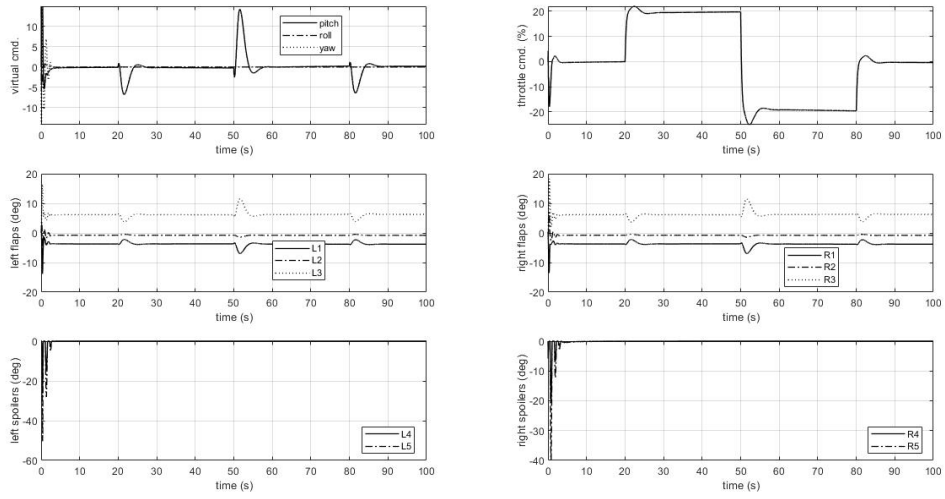

Figure B.10: Nominal, 250 KIAS, pitch doublet, plot 2

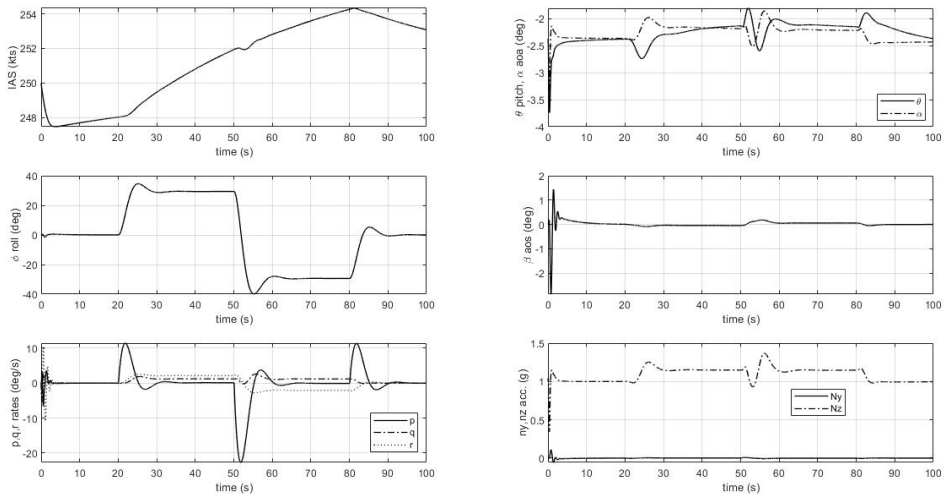

Figure B.11: Nominal, 250 KIAS, roll doublet, plot 1

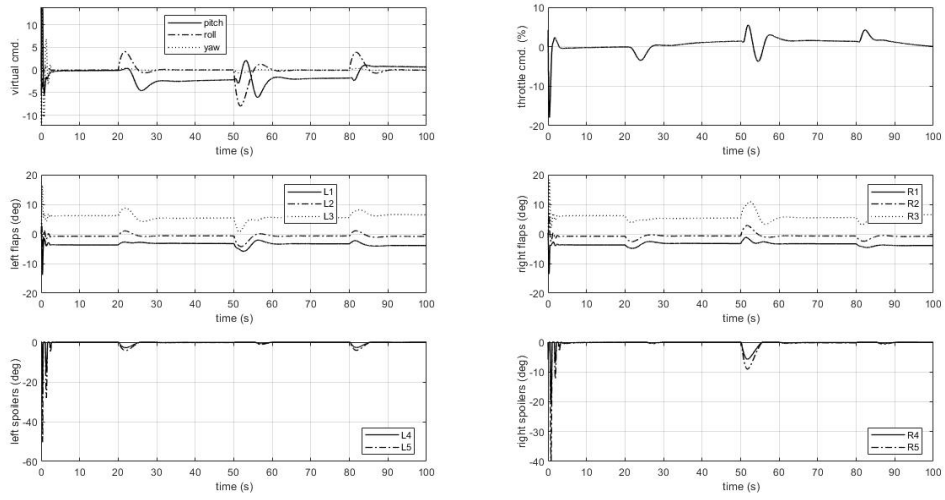

Figure B.12: Nominal, 250 KIAS, roll doublet, plot 2

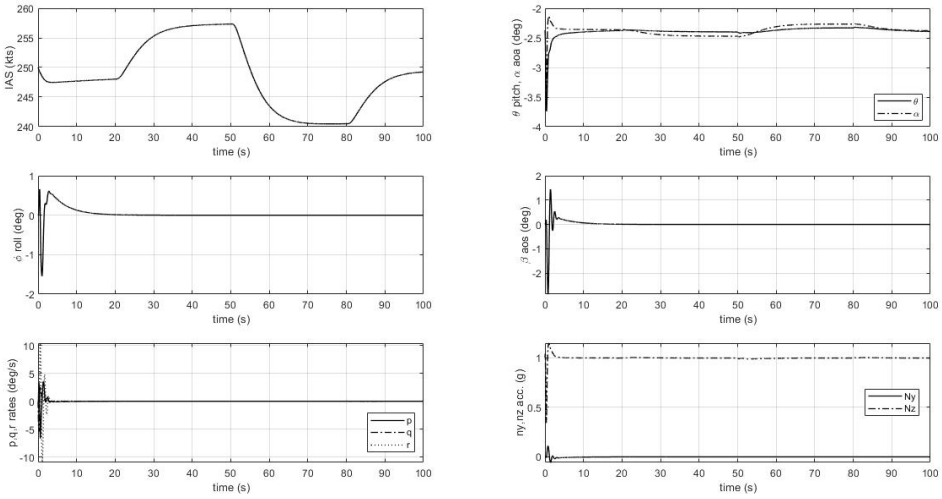

Figure B.13: Nominal, 250 KIAS, speed doublet, plot 1

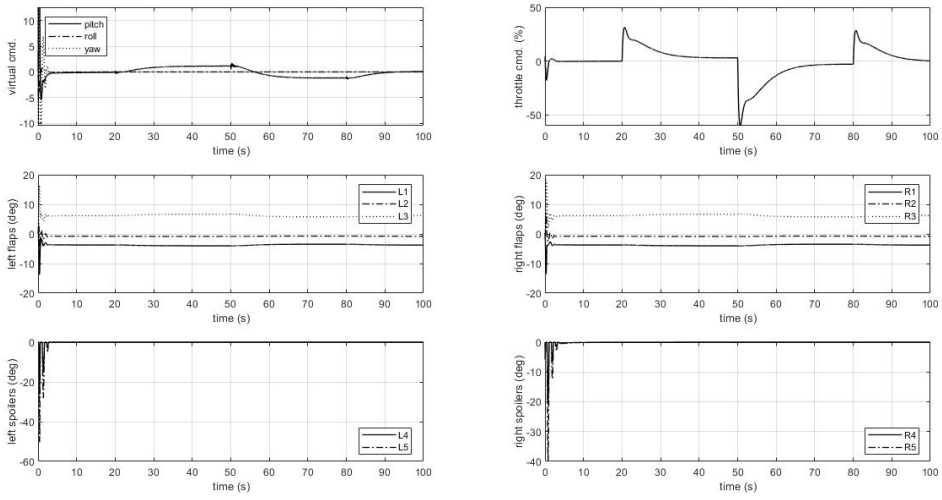

Figure B.14: Nominal, 250 KIAS, speed doublet, plot 2

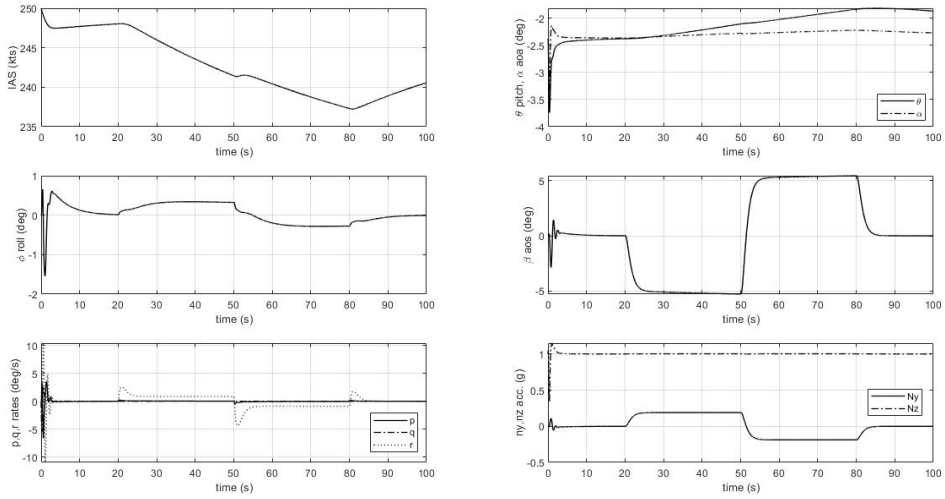

Figure B.15: Nominal, 250 KIAS, sideslip doublet, plot 1

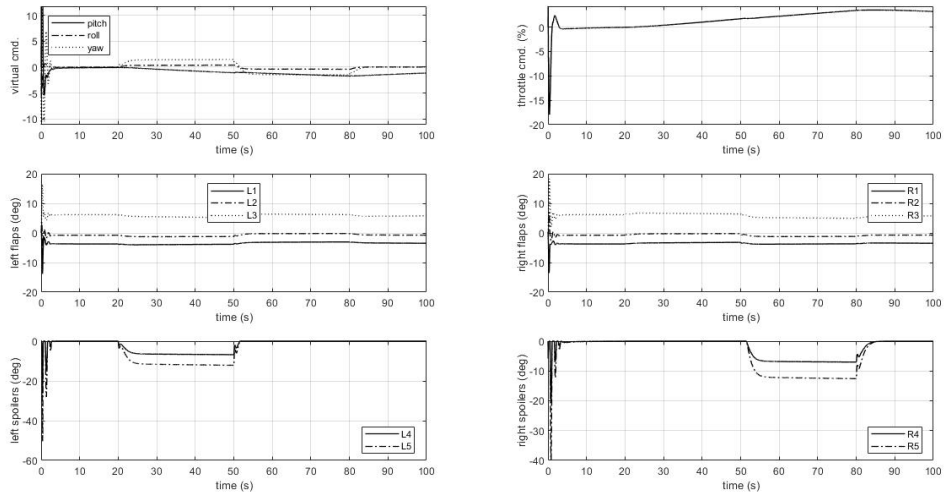

Figure B.16: Nominal, 250 KIAS, sideslip doublet, plot 2

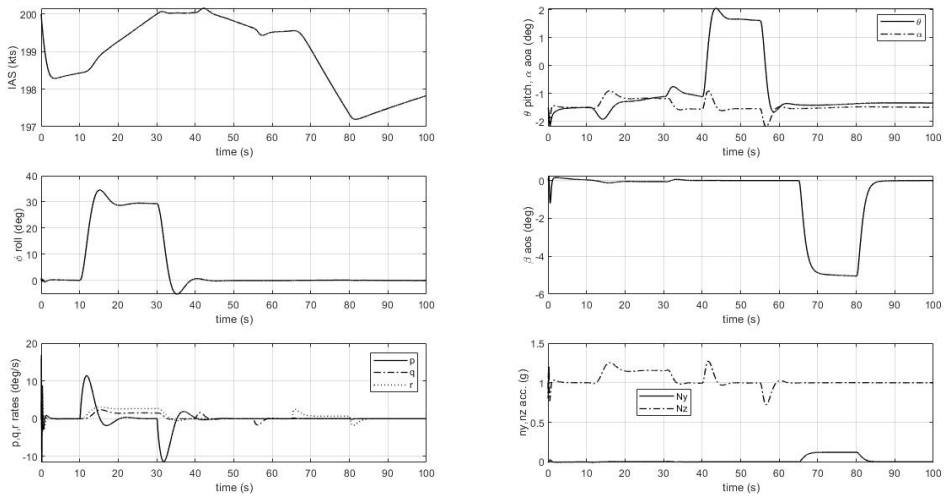

Figure B.17: Malf#1, CGI control allocation, plot 1

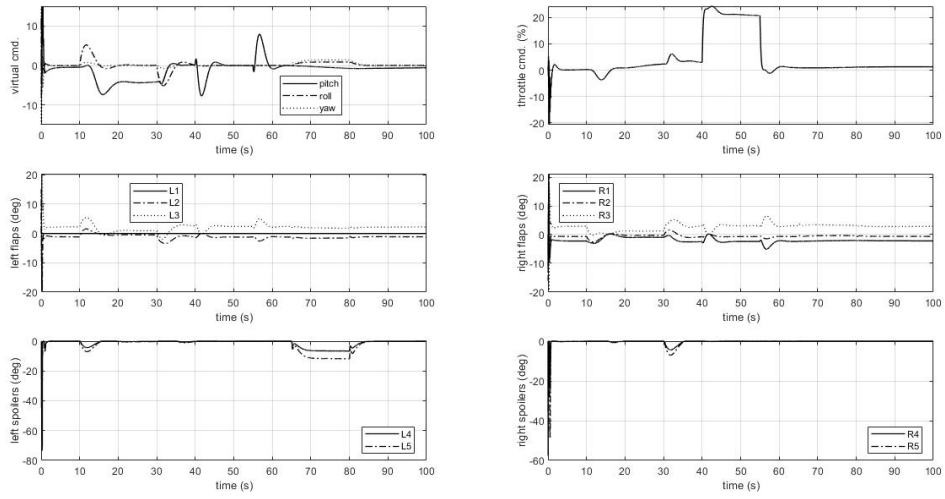

Figure B.18: Malf#1, CGI control allocation, plot 2

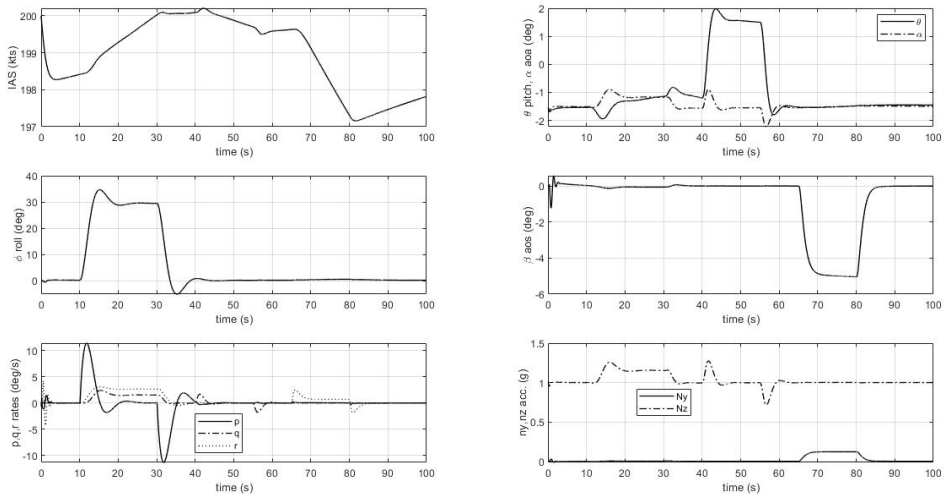

Figure B.19: Malf#1, WLS control allocation, plot 1

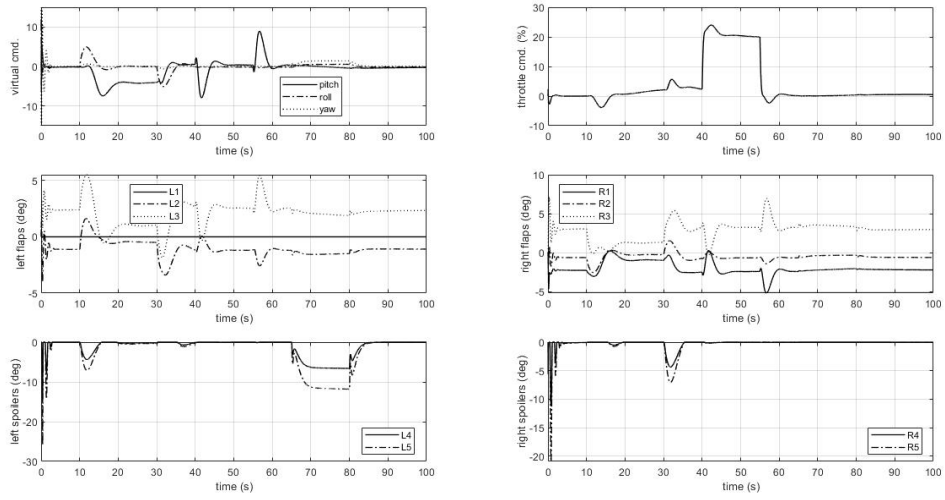

Figure B.20: Malf#1, WLS control allocation, plot 2

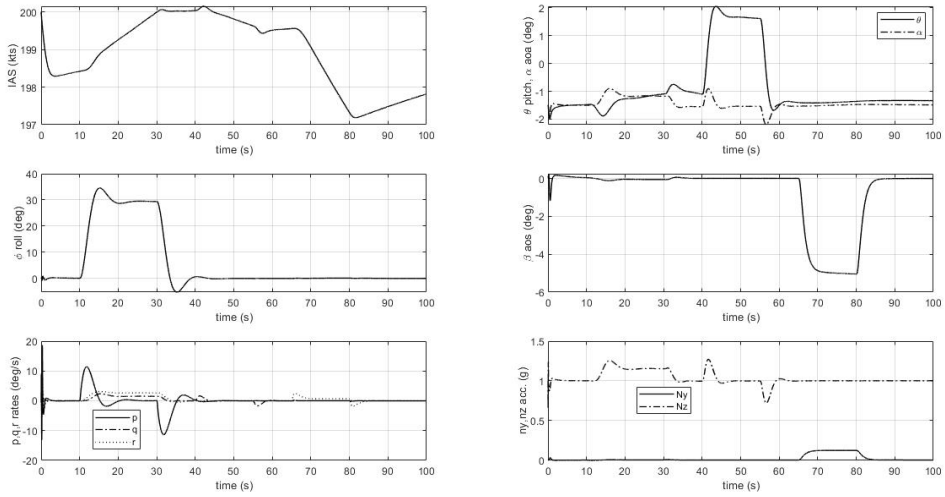

Figure B.21: Malf#2, CGI control allocation, plot 1

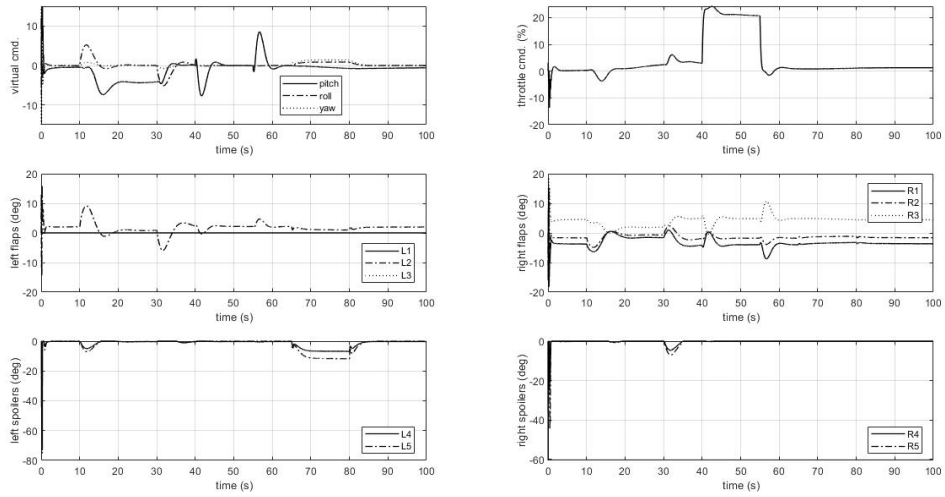

Figure B.22: Malf#2, CGI control allocation, plot 2

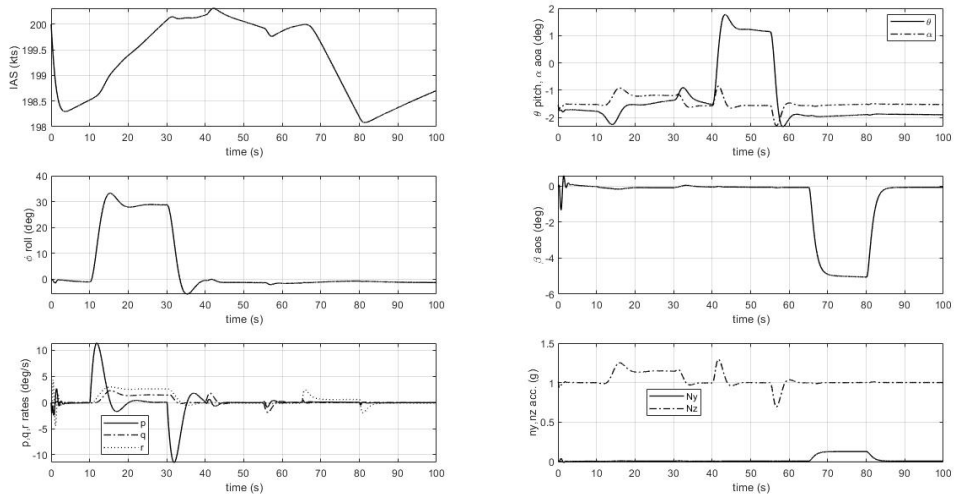

Figure B.23: Malf#2, WLS control allocation, plot 1

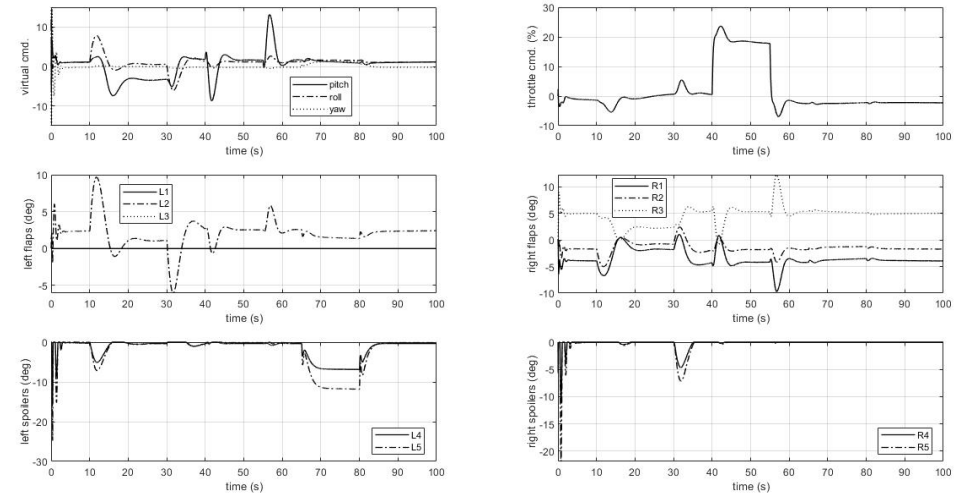

Figure B.24: Malf#2, WLS control allocation, plot 2

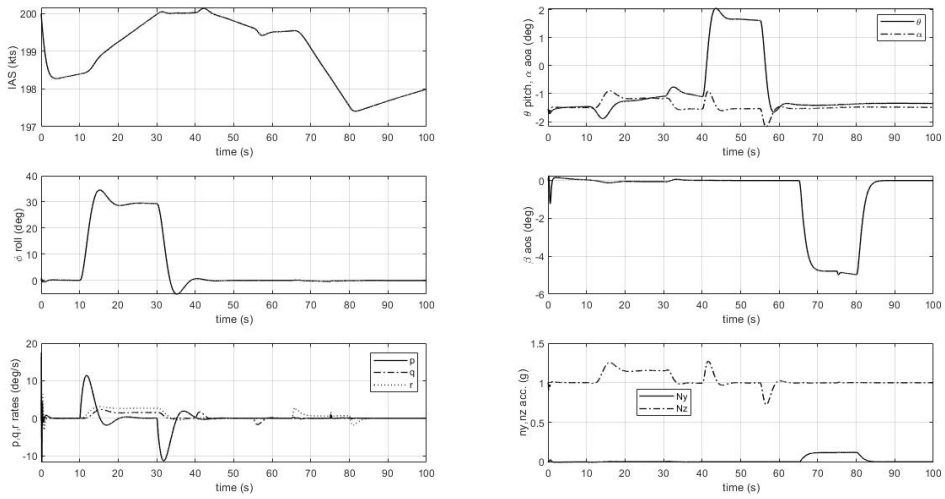

Figure B.25: Malf#4, CGI control allocation, plot 1

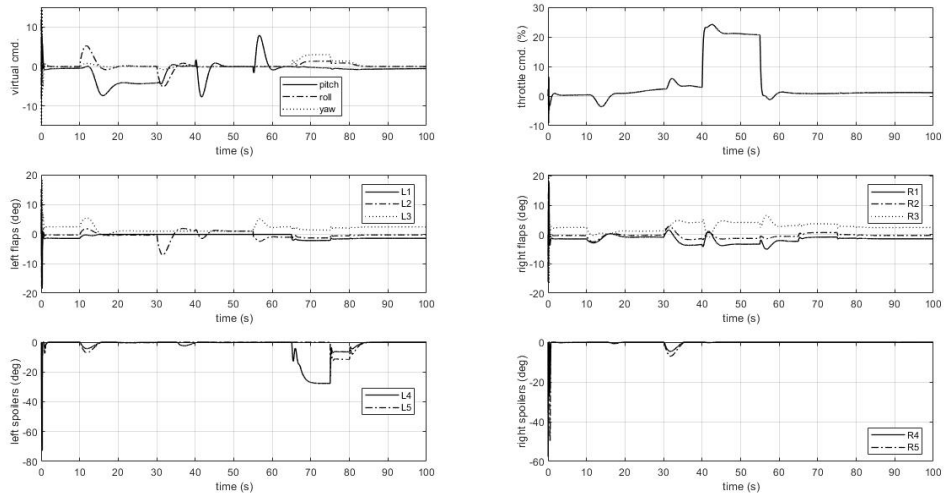

Figure B.26: Malf#4, CGI control allocation, plot 2

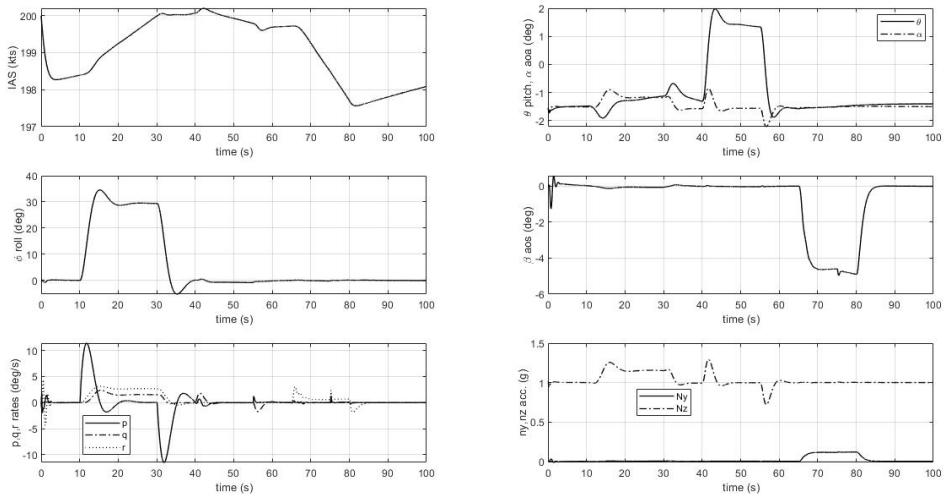

Figure B.27: Malf#4, WLS control allocation, plot 1

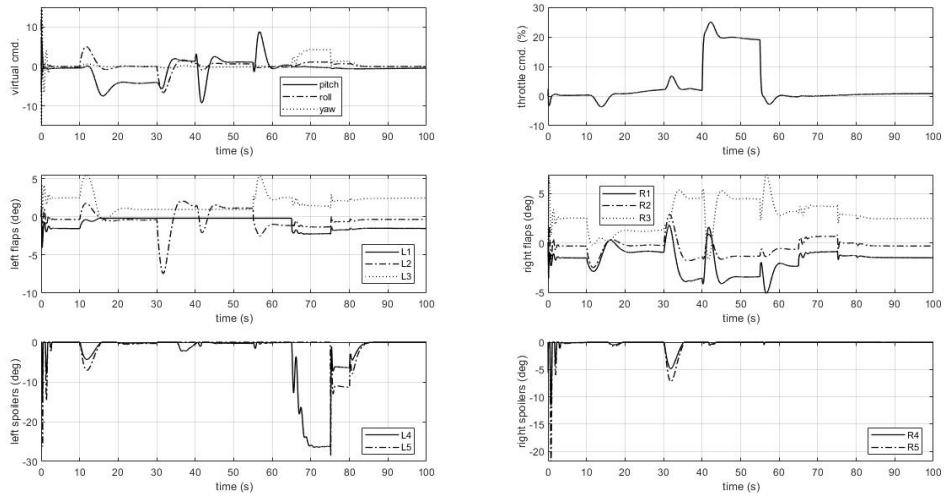

Figure B.28: Malf#4, WLS control allocation, plot 2

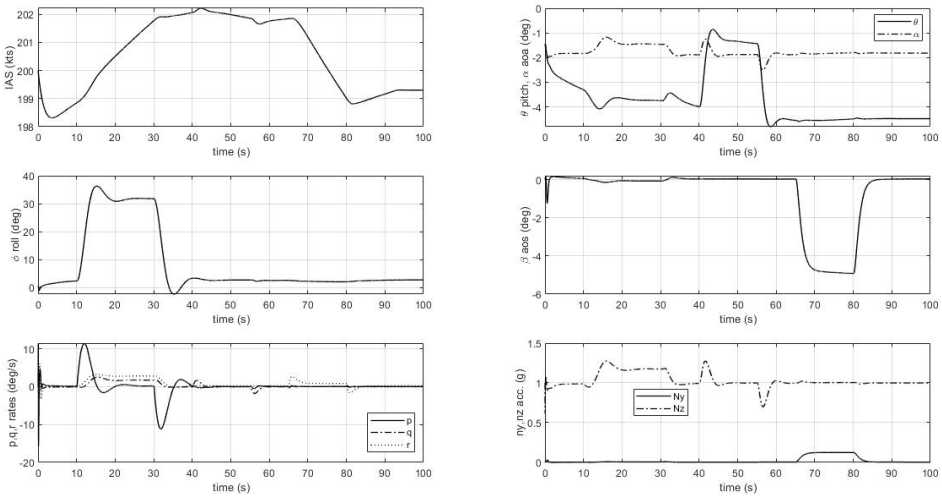

Figure B.29: Malf#5, CGI control allocation, plot 1

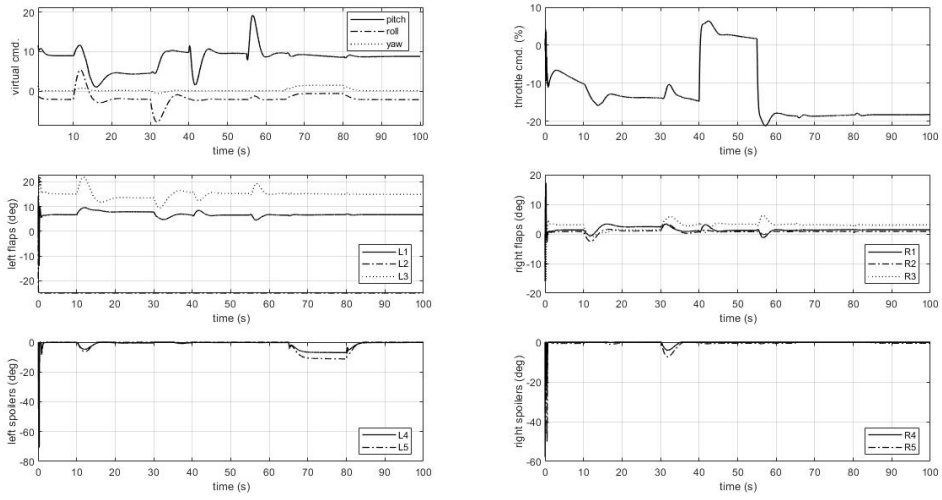

Figure B.30: Malf#5, CGI control allocation, plot 2

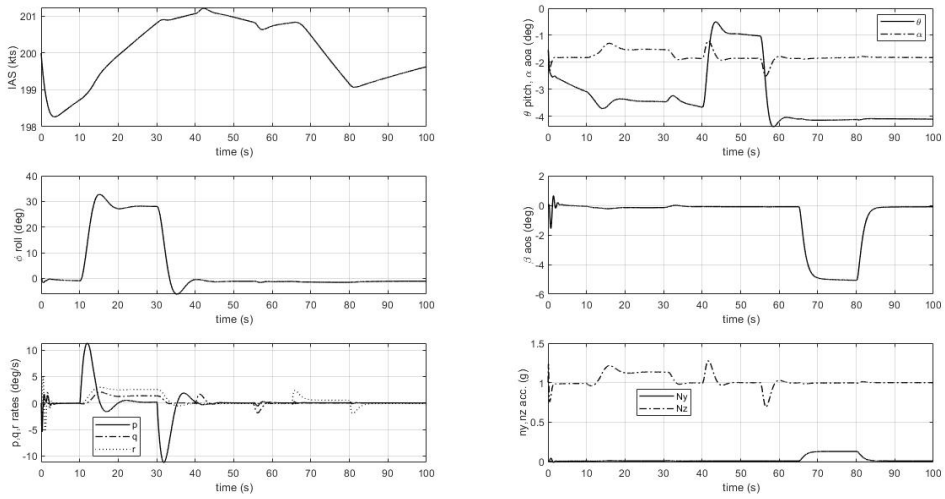

Figure B.31: Malf#5, WLS control allocation, plot 1

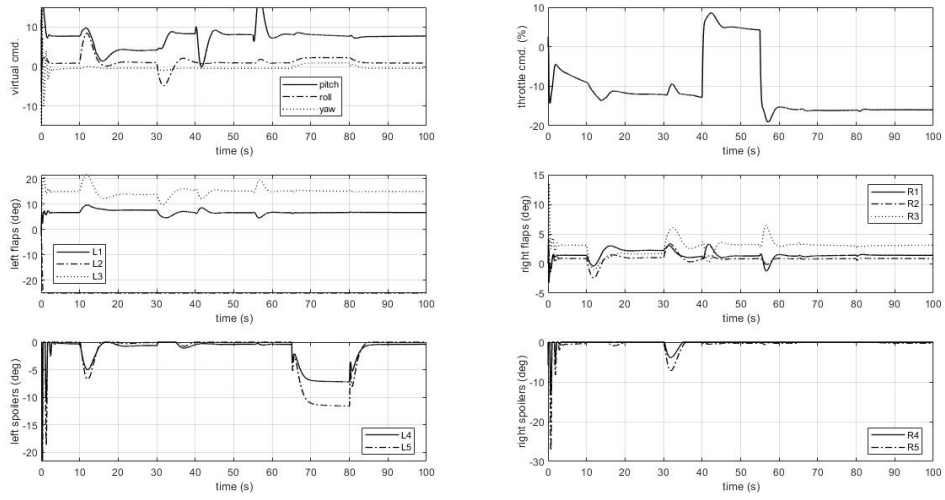

Figure B.32: Malf#5, WLS control allocation, plot 2

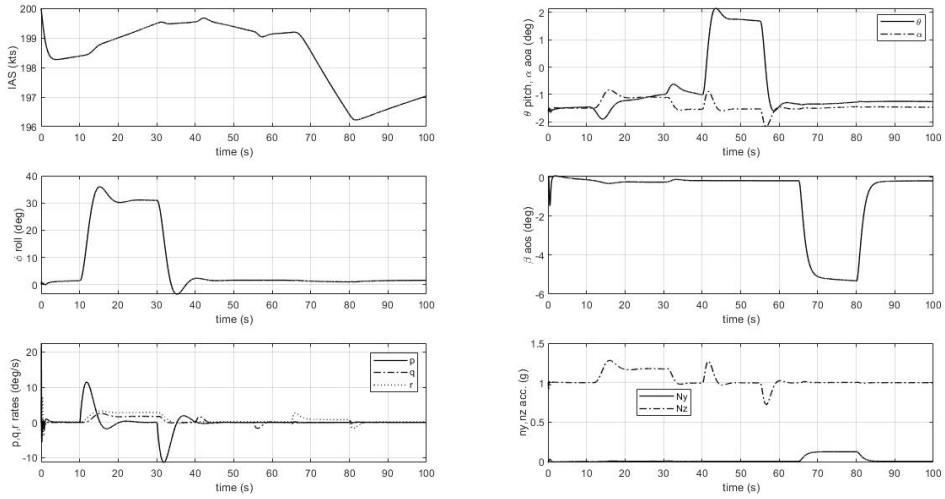

Figure B.33: Malf#6, CGI control allocation, plot 1

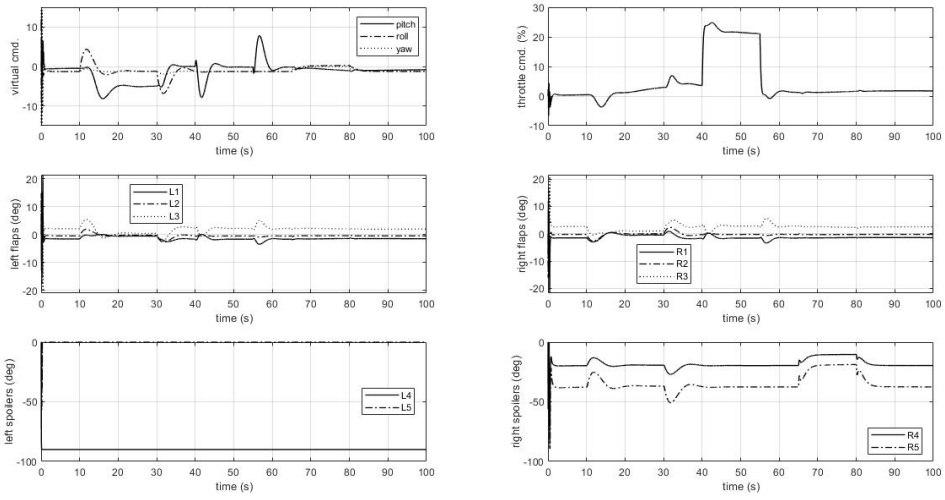

Figure B.34: Malf#6, CGI control allocation, plot 2

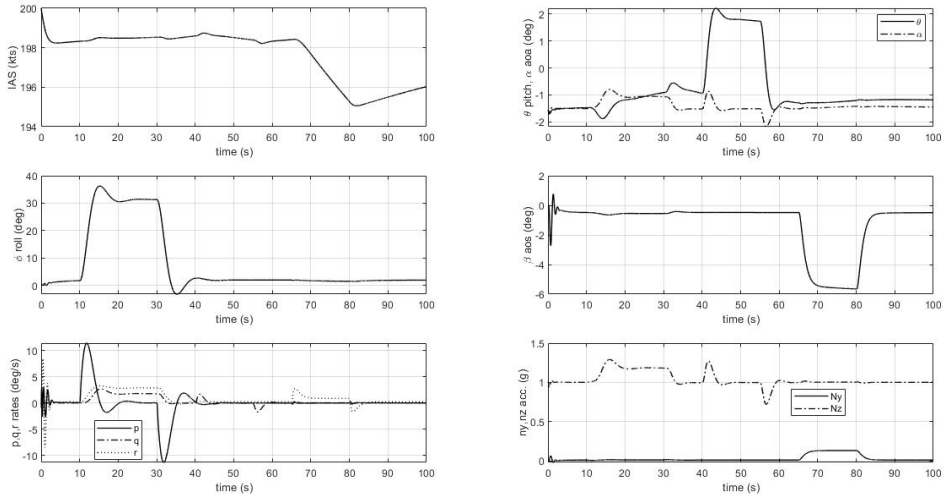

Figure B.35: Malf#6, WLS control allocation, plot 1

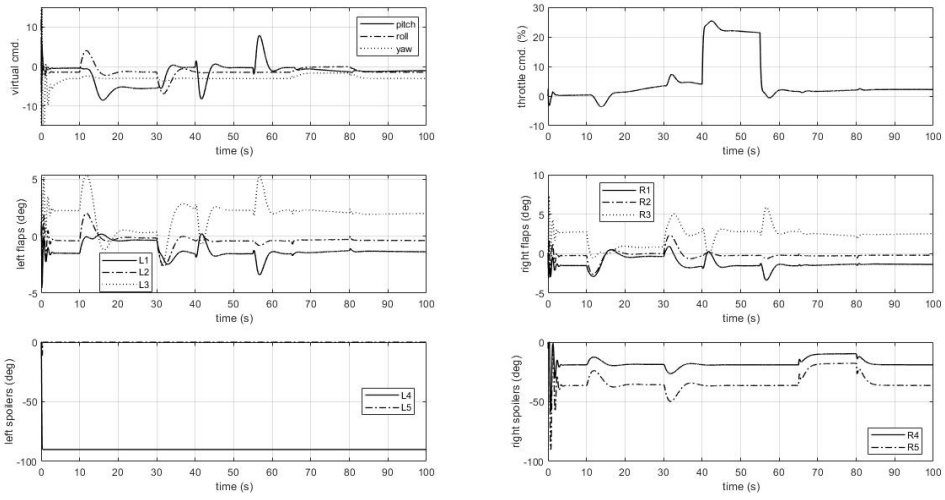

Figure B.36: Malf#6, WLS control allocation, plot 2

REFERENCES

[1] Colgren, R. and Loschke, R., "Effective Design of Highly Maneuverable Tailless Aircraft," *AIAA Journal of Aircraft*, Vol. 45, July 2008, pp. 1441 – 1449.

[2] Stenfelt, G. and Ringertz, U., "Lateral Stability and Control of a Tailless Aircraft Configuration," *AIAA Journal of Aircraft*, Vol. 46, Nov. 2009, pp. 2161 – 2164.

[3] Bolsunovsky, A., Buzoverya, N., Gurevich, B., Denisov, V., Dunaevsky, A., Shkadov, L., Sonin, O., Udzhuhu, A., and Zhurihin, J., "Flying Wing - Problems and Decisions," *Aircraft Design*, Vol. 4, Dec. 2001, pp. 193 – 219.

[4] Shtessel, Y., Buffington, J., and Banda, S., "Tailless Aircraft Flight Control Using Multiple Time Scale Reconfigurable Sliding Modes," *IEEE Transactions on Control Systems Technology*, Vol. 10, March 2002, pp. 288 – 296.

[5] Gundlach, J., *Designing Unmanned Aircraft Systems: A Comprehensive Approach*, AIAA Education Series, Reston, VA, 2012.

[6] Castro, H. D., *Flying and Handling Qualities of a Fly-by-Wire Blended Wing Body Civil Transport Aircraft*, Ph.D. thesis, Cranfield University, Cranfield, UK, 2003.

[7] Seckel, E., *Stability and Control of Aircraft and Helicopters*, Academic Press, New York, NY, 1964.

[8] "The USAF Stability and Control Digital DATCOM, Volume I, Users Manual," Tech. Rep. AFFDL-TR-79-3032, McDonnell Douglas Astronautics Company, St.Louis, MO, 1979.

[9] "Application of Multivariable Control Theory to Aircraft Control Laws, Final Report: Multivariable Control Design Guidelines," Tech. Rep. WL-TR-96-3099,

Honeywell Technology Center, Lockheed Martin Skunk Works, Lockheed Martin Tactical Aircraft Systems, 1996.

[10] Natesan, K., Da-Wei, G., Postlethwaite, I., and Jianchi, C., "Design of Flight Controllers based on Simplified LPV model of a UAV," *Proceedings of the 45th IEEE Conference on Decision and Control*, Dec. 2006.

[11] Stilwell, D. J. and Rugh, W. J., "Stability Preserving Interpolation Methods for the Synthesis of Gain Scheduled Controllers," *Automatica*, Vol. 36, 2000, pp. 665 – 671.

[12] Durham, W., Bordignon, K., and Beck, R., *Aircraft Control Allocation*, John Wiley & Sons, Chickester, West Sussex, UK, 2017.

[13] 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.

[14] Skogestad, S. and Postlethwaite, I., *Multivariable Feedback Control Analysis and Design*, John Wiley & Sons, Chickester, West Sussex, UK, 2005.

[15] Stevens, B., Lewis, F., and Johnson, E., *Aircraft Control and Simulation - Third ed.*, Wiley, Hoboken, NJ, 2016.

[16] Amato, F. and Iervolino, R., "μ-Synthesis for a Small Commercial Aircraft: Design and Simulator Validation," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 27, May 2004, pp. 479 – 490.

[17] Hyde, R. A. and Glover, K., "The Application of Scheduled H_∞ Controllers to a VSTOL Aircraft," *IEEE Transactions on Automatic Control*, Vol. 38, July 1993, pp. 1021.

[18] Aouf, N. and Boulet, B., "Linear Fractional Transformation Gain-Scheduling Flight Control Preserving Robust Performance," *Journal of Aerospace Engineering*, Vol. 226, July 2012, pp. 763 – 773.

[19] Natesan, K., Gu, D., and Postlethwaite, I., "Design of Linear Parameter Varying Trajectory Tracking Controllers for an Unmanned Air Vehicle," *Journal of Aerospace Engineering*, Vol. 224, April 2010, pp. 395 – 402.

[20] Mystkowski, A., "An Application of μ-Synthesis to Control of a Small Air Vehicle and Simulation Results," *Journal of Vibroengineering*, Vol. 14, March 2012.

[21] Nichols, R., Reichert, R., and Rugh, W., "Gain Scheduling for H_∞ Controllers: A Flight Control Example," *IEEE Transactions on Control Systems Technology*, Vol. 1, June 1993, pp. 69 – 79.

[22] Biannic, J.-M., Apkarian, P., and Garrard, W. L., "Parameter Varying Control of a High Performance Aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 20, March 1997, pp. 225 – 231.

[23] Turkoglu, E., *Application of Robust Control to a Rotary-Wing Aircraft*, Ph.D. thesis, University of Leicester, Leicester, UK, 2007.

[24] Greenberg, M., *Advanced Engineering Mathematics - 2nd Edition*, Prentice Hall, Upper Saddle River, NJ, 1998.

[25] Ogata, K., *Modern Control Engineering - 5th Edition*, Prentice Hall, Upper Saddle River, NJ, 2010.

[26] Frazzoli, E., "MIT Open Courseware, Lecture Notes," Internet draft, 2004.

[27] Balas, G., Doyle, J., Glover, K., Packard, A., and Smith, R., *μ-Analysis and Synthesis Toolbox - Users Guide - Version 3*, The Mathworks Inc., Natick, MA, 2001.

[28] Nalbantoglu, V., *Robust Control and System Identification for Flexible Structures*, Ph.D. thesis, University of Minnesota, MN, 1998.

[29] Zhou, K., Doyle, J., and Glover, K., *Robust and Optimal Control*, Prentice Hall, Englewood Cliffs, NJ, 1995.

[30] Shamma, J. S. and Cloutier, J. R., "Gain-Scheduled Missile Autopilot Design Using Linear Parameter Varying Transformations," *Journal of Guidance, Control, and Dynamics*, Vol. 16, 1993, pp. 256 – 263.

[31] Biannic, J.-M. and Apkarian, P., "Missile Autopilot Design via a Modified LPV Synthesis Technique," *Aerospace Science and Technology*, Vol. 3, 1999, pp. 153 – 160.

[32] Marcos, A. and Balas, G. J., "Development of Linear-Parameter-Varying Models for Aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 27, March 2004, pp. 218 – 228.

[33] Packard, A., "Gain scheduling via Linear Fractional Transformations," *Systems & Control Letters*, Vol. 22, 1994, pp. 79 – 92.

[34] Helmersson, A., "μ-Synthesis and LFT Gain Scheduling with Mixed Uncertainties," *International Journal of Robust and Nonlinear Control*, Vol. 8, June 1998, pp. 631 – 642.

[35] Magni, J., "An LFT Approach to Robust Gain Scheduling," *Proceedings of the 44th IEEE Conference on Decision and Control*, 2005.

[36] Natesan, K., Da-Wei, G., Postlethwaite, I., and Jianchi, C., "Design of Flight Controllers based on Simplified LPV model of a UAV," *Proceedings of the 45th IEEE Conference on Decision and Control*, Dec. 2006.

[37] Stilwell, D. J., "State Space Interpolation for a Gain-Scheduled Autopilot," *Journal of Guidance, Control, and Dynamics*, Vol. 24, May 2001, pp. 460 – 465.

[38] Johansen, T. A. and Fossen, T. I., "Control Allocation - A Survey," *Automatica*, Vol. 49, May 2013, pp. 1087 – 1103.

[39] Harkegaard, O., "Dynamic Control Allocation Using Constrained Quadratic Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 27, Nov. 2004, pp. 1028 – 1034.

[40] Enns, D., "Control Allocation Approaches," 1998.

[41] Nieto-Wire, C., *Eigenstructure Assignement for a Tailless Aircraft*, Ph.D. thesis, The City University of New York, NY, 2012.

[42] Wise, K. A., Brinker, J. S., Calise, A. J., Enns, D. F., Elgersma, M. R., and Voulgaris, P., "Direct Adaptive Reconfigurable Flight Control for a Tailless Advanced Fighter Aircraft," *International Journal of Robust and Nonlinear Control*, Vol. 9, Dec. 1999, pp. 999 – 1012.

[43] Chen, S., *Adaptive Control of Nonlinear Systems with Actuator Failures and Uncertainities*, Ph.D. thesis, University of Virginia, VA, 2004.

[44] Naskar, A. K., Patra, S., and Sen, S., "A Reconfigurable Direct Control Allocation Method," *2013 European Control Conference (ECC)*, July 2013.

[45] Kroo, I., "Tailless Aircraft Design - Recent Experiences," *Proceedings of the Symposium on Aerodynamics and Aeroacoustics*, Tucson, AZ.

[46] Nickel, K. and Wohlfahrt, M., *Tailless Aircraft in Theory and Practice*, Butterworth - Heinemann, Burlington, MA, 2003.

[47] Peterson, T., *Handling Qualities of a Blended Wing Body Aircraft*, Master's thesis, University of Toronto, Canada, 2011.

[48] Kroo, I., *Aerodynamics, Aeroelasticity and Stability of Hang Gliders*, Ph.D. thesis, Stanford University, CA, 1983.

[49] Cusher, A. A. and Gopalarathnam, A., "Drag Reduction Methodology for Adaptive Tailless Aircraft," *Journal of Aircraft*, Vol. 49, Jan. 2012, pp. 161 – 172.

[50] Ashkenas, I. and Clyde, D., "Tailless Aircraft Performance Improvements with Relaxed Static Stability," Tech. Rep. NASA CR-181806, Systems Technology, Inc., Hawthorne, CA, 1989.

[51] Calise, A., Lee, S., and Sharma, M., "Direct Adaptive Reconfigurable Control of a Tailless Fighter Aircraft," 1998.

[52] Peterson, T., *On the use of Active Flow Control to Trim and Control a Tailless Aircraft Model*, Master's thesis, The University of Arizona, Tucson, AZ, 2017.

[53] Saephan, S., *Determination of Tailless Aircraft Tumbling and Stability Characteristics Through Computational Fluid Dynamics*, Ph.D. thesis, University of California, Davis, CA, 2006.

[54] Park, J., Choi, J.-Y., Jo, Y., and Choi, S., "Stability and Control of Tailless Aircraft Using Variable Fidelity Aerodynamic Analysis," *Journal of Aircraft*, Vol. 54, Nov. 2017, pp. 2148 – 2164.

[55] Stenfelt, G., *Aerodynamics and Lateral Control of Tailless Aircraft*, Ph.D. thesis, Royal Institute of Technology, Stockholm, Sweden, 2011.

[56] "RQ-3 Darkstar," 2019.

[57] Cavalcanti, S. and Papini, M., "Preliminary Model Matching of the EMBRAER 170 Jet," *Journal of Aircraft*, Vol. 41, July 2004, pp. 703 – 710.

[58] Watters, B., *Adaptation of Digital Datcom into a Conceptual Design Process*, Master's thesis, The University of Texas at Arlington, TX, 2011.

[59] "USAF Stability and Control Datcom," Tech. rep., McDonnell Douglas Corporation, 1978.

[60] Botez, R. M., Bardela, P. A., and Bournisien, T., "Cessna Citation X Simulation Turbofan Modelling: Identification and Identified Model Validation using Simulated Flight Tests," *The Aeronautical Journal*, Vol. 123, April 2019, pp. 433 – 463.

[61] Dreier, M. E., *Introduction to Helicopter and Tiltrotor Simulation*, AIAA Education Series, Reston, VA, 2007.

[62] "Flying Qualities of Piloted Aircraft," Tech. Rep. MIL HDBK 1797, U.S. Department of Defense, 1997.

[63] Balas, G., Chiang, R., Packard, A., and Safonov, M., *Robust Control Toolbox - User's Guide*, The Mathworks Inc., Natick, MA, 2019.

[64] Shamma, J. and Athans, M., "Gain Scheduling: Potential Hazards and Possible Remedies," *IEEE Control Systems*, Vol. 12, June 1992, pp. 101 – 107.

[65] Gahinet, P., Nemirovski, A., Laub, A., and Chilali, M., *LMI Control Toolbox - User's Guide*, The Mathworks Inc., Natick, MA, 1995.

[66] Boyd, S., Ghaoui, L. E., Feron, E., and Balakrishnan, V., *Linear Matrix Inequalities in System and Control Theory*, SIAM, Philadelphia, PA, 1994.

[67] Harkegaard, O., "QCAT Documentation," 2019.