Control Allocation for the Next Generation of Entry Vehicles

by

Richard H. Shertzer

B.S. Engineering Mechanics
United States Air Force Academy, 1999

SUBMITTED TO THE DEPARTMENT OF AERONAUTICS AND ASTRONAUTICS
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN AERONAUTICS AND ASTRONAUTICS
AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2001

© 2001 Richard H. Shertzer, All rights reserved.

The author hereby grants to MIT permission to reproduce and to distribute publicly paper and electronic copies of this thesis document in whole or in part.

Signature of Author____
Department of Aeronautics and Astronautics
May 11, 2001

Approved by____
Douglas J. Zimpfer
Charles Stark Draper Laboratory
Technical Supervisor

Certified by____
Rudrapatna V. Ramnath
Senior Lecturer, Department of Aeronautics and Astronautics
Thesis Advisor

Accepted by____
Wallace E. Vander Velde
Professor of Aeronautics and Astronautics
Chair, Committee on Graduate Students

MASSACHUSETTS INSTITUTE
OF TECHNOLOGY
SEP 1 1 2001
LIBRARIES
AERO

[This page intentionally left blank]

Control Allocation for the Next Generation of Entry Vehicles

by

Richard H. Shertzer

Submitted to the Department of Aeronautics and Astronautics
on May 11, 2001 in Partial Fulfillment of the
Requirements for the Degree of Master of Science in
Aeronautics and Astronautics

Abstract

Control allocation is the process of assigning control responsibility amongst redundant actuators. A control allocation algorithm for an entry vehicle is presented that uses a linear program to optimally specify bounded aerosurface deflections and jet firings in response to torque commands. Actuator preference is introduced via an objective function to produce a unique solution when the system of linear equations is underdetermined. A multivariable control law is adopted to drive the control allocation algorithm and to track the desired state of the entry vehicle model. Open loop and closed loop tests are conducted to demonstrate dynamic objective calculation, blended aerosurface/jet capability, and efficient reconfiguration in the event of actuator failure. An approach is also presented to define the relationship between systematic errors in measured vehicle state and the actuator commands produced by the control allocation algorithm. Potential control allocation applications beyond the entry problem and other recommendations for further research are stated in the concluding remarks.

Technical Supervisor: Douglas J. Zimpfer
Title: Senior Member Technical Staff, C.S. Draper Laboratory, Inc.

Thesis Supervisor: Rudrapatna V. Ramnath
Title: Senior Lecturer, Department of Aeronautics and Astronautics

[This page intentionally left blank]

Acknowledgement

My greatest debt of gratitude goes to my wife, Amy, who has been loving and thoughtful during two years of separation. Although she will miss the visits to Boston, neither of us will miss the good-byes. I also appreciate the support from both my family and hers during this research effort.

I would like to thank the Charles Stark Draper Laboratory for the opportunity to pursue my graduate education at MIT with both technical support and financial means. I am indebted to my Draper supervisors, Doug Zimpfer, Pat Brown, Piero Miotto, and Gregg Barton for volumes of advice and assistance. Steve Kolitz and Steve Clark also offered their resources and experience regarding optimization techniques. I would also like to thank Jennifer Hamelin and Roberto Pileggi for their technical guidance during my first year at the Draper Laboratory.

I am also grateful to the faculty and staff at MIT's Department of Aeronautics and Astronautics. Particular thanks go to my thesis advisor, Dr. Rudrapatna Ramnath, for his guidance and support.

Lastly, I would like to thank my friends for enlivening the MIT experience. Raja and Steve always made lunch an interesting time and the MIT weight room will never quite be same now that Raja has left his mark. Best of luck to Chris and Ted in the upcoming year. John J. and Rich are great housemates and I wish them luck as they enter pilot training. To all others with whom I've enjoyed intramural basketball, ski weekends, sightseeing jaunts, and baseball games – thanks for the memories.

This thesis was prepared at The Charles Stark Draper Laboratory, Inc. under Internal Research & Development Project #13033.

Publication of this thesis does not constitute approval by Draper or the sponsoring agency of the findings or conclusions contained herein. It is published for the exchange and stimulation of ideas.

Richard H. Shertzer
2Lt, USAF

[This page intentionally left blank]

Table of Contents

INTRODUCTION 15

1.1 THESIS OBJECTIVES 16

1.2 THESIS PREVIEW 16

ENTRY CONTROL OVERVIEW 19

2.1 SPACE SHUTTLE 19

2.2 CURRENT EFFORTS 21

SYSTEM ARCHITECTURE 27

3.1 OVERVIEW 27

3.2 VEHICLE DESCRIPTION AND DYNAMICS 28

3.3 TRAJECTORY GENERATION 34

3.4 GUIDANCE AND CONTROL 34

CONTROL ALLOCATION 39

4.1 OVERVIEW 39

4.2 THE LINEAR PROGRAM 40

4.3 UPPER AND LOWER BOUNDS 51

4.4 OBJECTIVE FUNCTION COEFFICIENTS 52

4.5 ACTUATOR CONTROL AUTHORITIES 55

4.6 PULSING LOGIC 58

4.7 OPEN LOOP TESTS: CONTROL ALLOCATION ISOLATION 59

CLOSED LOOP SIMULATIONS 67

5.1 LOW ALTITUDE ENTRY SIMULATIONS 68

5.2 HIGH ALTITUDE ENTRY SIMULATIONS 73

5.3 ACTUATOR FAILURE SIMULATIONS 81

5.4 COMPUTATIONAL EFFICIENCY 85

ACTIVITY VECTOR UNCERTAINTY 87

6.1 ANALYSIS METHOD 87

6.2 SAMPLE DATA 89

CONCLUSIONS AND RECOMMENDATIONS 95

APPENDIX A: THE SOLUTION ALGORITHM 99

APPENDIX B: ADDITIONAL SIMULATION RESULTS **103**

REFERENCES **113**

List of Figures

Figure 2-1. Shuttle-based Guidance and Control 20

Figure 2-2. Shuttle-based Longitudinal Control 21

Figure 2-3. MIMO/Control Allocation Architecture 22

Figure 3-1. Simulation Architecture 27

Figure 3-2. Diagram of the X-34 [12] 29

Figure 3-3. X-34 RCS Configuration 31

Figure 3-4. LQ-Servo with Integrators 35

Figure 4-1. Control Allocation Architecture 39

Figure 4-2. Geometry of Example 1 43

Figure 4-3. Geometry of Example 1 with Upper Bound on x_2 45

Figure 4-4. Geometry of Example 1 with Objective Function $z = |x_1| + 4 \cdot |x_2|$ 46

Figure 4-5. Pitching Moment vs. Elevon Deflection 50

Figure 4-6. Example of c_{rate} Objective Coefficient Contribution 54

Figure 4-7. Open Loop Test #1: Commanded and Measured Torque 60

Figure 4-8. Open Loop Test #1: Actuators 61

Figure 4-9. Open Loop Test #2: No Objective Function 62

Figure 4-10. Open Loop Test #2: Objective Function Applied 62

Figure 4-11. Open Loop Test #3 64

Figure 5-1. Low Altitude Entry Simulation: Nominal Case (1 of 6) 69

Figure 5-2. Low Altitude Entry Simulation: Nominal Case (2 of 6) 69

Figure 5-3. Low Altitude Entry Simulation: Nominal Case (3 of 6) 70

Figure 5-4. Low Altitude Entry Simulation: Nominal Case (4 of 6) 70

Figure 5-5. Low Altitude Entry Simulation: Nominal Case (5 of 6) 71

Figure 5-6. Low Altitude Entry Simulation: Nominal Case (6 of 6) 71

Figure 5-7. Low Altitude Entry Simulation: 60° Bank Turns (1 of 3) 72

Figure 5-8. Low Altitude Entry Simulation: 60° Bank Turns (2 of 3) 72

Figure 5-9. Low Altitude Entry Simulation: 60° Bank Turns (3 of 3) 73

Figure 5-10. High Altitude Entry Simulation: Pitch Maneuver / Low Elevon Cost...................... 75

Figure 5-11. High Altitude Entry Simulation: Pitch Maneuver / Medium Elevon Cost............... 75

Figure 5-12. High Altitude Entry Simulation: Pitch Maneuver / High Elevon Cost..................... 76

Figure 5-13. High Altitude Entry Simulation: Roll Maneuver (1 of 2).. 77

Figure 5-14. High Altitude Entry Simulation: Roll Maneuver (2 of 2).. 78

Figure 5-15. High Altitude Entry Simulation: RCS Only (1 of 3).. 79

Figure 5-16. High Altitude Entry Simulation: RCS Only (2 of 3).. 79

Figure 5-17. High Altitude Entry Simulation: RCS Only (3 of 3).. 80

Figure 5-18. Actuator Failure Simulation: Elevon Stuck at t = 20s (1 of 3)................................ 81

Figure 5-19. Actuator Failure Simulation: Elevon Stuck at t = 20s (2 of 3)................................ 82

Figure 5-20. Actuator Failure Simulation: Elevon Stuck at t = 20s (3 of 3)................................ 82

Figure 5-21. Actuator Failure Simulation: No Rudder until t = 20s (1 of 3)................................ 83

Figure 5-22. Actuator Failure Simulation: No Rudder until t = 20s (2 of 3)................................ 84

Figure 5-23. Actuator Failure Simulation: No Rudder until t = 20s (3 of 3)................................ 84

Figure 6-1. Open Loop Path from M_{cmd} to M_{meas} (1 of 3).. 87

Figure 6-2. Open Loop Path from M_{cmd} to M_{meas} (2 of 3).. 88

Figure 6-3. Open Loop Path from M_{cmd} to M_{meas} (3 of 3).. 89

Figure 6-4. Example Data: K_3 vs. Full Range of M_{cmds} / de_{cmds} 90

Figure 6-5. Example Data: K_3 vs. Reasonable Range of M_{cmds} / de_{cmds} 91

Figure 6-6. Example Data: K_3 and Magnitude of Gain Uncertainty vs. % Error in Mach 92

Figure 8-1. Solution Algorithm Overview.. 99

Figure 9-1. Low Altitude Entry Simulation: 60° Bank Turns (1 of 6) 103

Figure 9-2. Low Altitude Entry Simulation: 60° Bank Turns (2 of 6) 103

Figure 9-3. Low Altitude Entry Simulation: 60° Bank Turns (3 of 6) 104

Figure 9-4. Low Altitude Entry Simulation: 60° Bank Turns (4 of 6) 104

Figure 9-5. Low Altitude Entry Simulation: 60° Bank Turns (5 of 6) 105

Figure 9-6. Low Altitude Entry Simulation: 60° Bank Turns (6 of 6) 105

Figure 9-7. Actuator Failure Simulation: Elevon Stuck at t = 20s (1 of 6)................................. 106

Figure 9-8. Actuator Failure Simulation: Elevon Stuck at t = 20s (2 of 6)................................. 106

Figure 9-9. Actuator Failure Simulation: Elevon Stuck at t = 20s (3 of 6)................................. 107

Figure 9-10. Actuator Failure Simulation: Elevon Stuck at t = 20s (4 of 6)............................... 107

Figure 9-11. Actuator Failure Simulation: Elevon Stuck at t = 20s (5 of 6)............................... 108

Figure 9-12. Actuator Failure Simulation: Elevon Stuck at t = 20s (6 of 6)............................... 108

Figure 9-13. Actuator Failure Simulation: No Rudder until t = 20s (1 of 6).............................. 109

Figure 9-14. Actuator Failure Simulation: No Rudder until t = 20s (2 of 6).............................. 109

Figure 9-15. Actuator Failure Simulation: No Rudder until t = 20s (3 of 6).............................. 110

Figure 9-16. Actuator Failure Simulation: No Rudder until t = 20s (4 of 6).............................. 110

Figure 9-17. Actuator Failure Simulation: No Rudder until t = 20s (5 of 6).............................. 111

Figure 9-18. Actuator Failure Simulation: No Rudder until t = 20s (6 of 6).............................. 111

[This page intentionally left blank]

List of Tables

Table 3-1. Physical Characteristics of the X-34 [11] 29

Table 3-2. Entry Actuator Characteristics 30

Table 3-3. X-34 RCS Firing Patterns 32

Table 3-4. Description of State Variables 33

Table 3-5. LQ-Servo Variable Definitions 35

Table 4-1. Open Loop Simulation Parameters 60

Table 5-1. High Altitude Simulation Parameters 74

Table 5-2. Control Allocation Algorithm Iterations 85

Table 6-1. Example Data: Worst-Case Parameter and Gain Uncertainty 92

Table 6-2. Potential Sources of Error in Modeling Activity Vectors 93

1 INTRODUCTION

The next generation of aircraft and aerospace vehicles will require control laws that utilize the full capability of available control effectors. This is necessary to meet safety and operational cost requirements, but it is a task that becomes increasingly difficult as control actuators grow in both number and complexity. Traditional aircraft possess three basic aerodynamic controls, one for each of the rotational degrees of freedom. Conversely, modern tactical aircraft have unconventional control effectors, such as canards and thrust vectoring, which serve to complement or replace the standard assortment of aileron, elevator, and rudder. The one-to-one correspondence between rotational degrees of freedom and vehicle controls no longer exists with the addition of modern control effectors. Vehicles that possess this type of control redundancy present both unique design possibilities and potential complications. One obvious benefit is that reconfiguration, necessary in the event of individual actuator failures, is more easily achieved when a collection of redundant control effectors is present. Redundancy across many actuators saves both cost and weight because no longer are individual actuators burdened with excess levels of redundancy. Of course, designing control laws that take advantage of control redundancy can be a difficult task. In an effort to avoid complexity, many prospective control approaches compartmentalize tasks rather than directly command a complex arsenal of actuators. A controller, in the traditional sense, is still responsible for tracking, stability, and disturbance rejection, while a separate control allocation algorithm transforms generalized commands from the controller into actuator commands. This modular approach preserves simplicity by allowing command of controlled degrees of freedom rather than scheduling the concerted effort of many redundant actuators.

Control redundancy is not exclusively inherent to tactical aircraft with unconventional aerosurfaces and thrust vectoring. Any entry vehicle requires control redundancy during the transition from exo- to endo-atmospheric flight. At extreme altitudes, reaction control system (RCS) jets are used to stabilize the vehicle because aerosurfaces lack sufficient control authority. A gradual transition ensues as the aerosurfaces gain control authority and the RCS jets become ineffective. Future aerospace vehicles must be equipped to manage not only the entry transition and subsequent atmospheric flight, but also powered ascent, the control transition from atmospheric ascent to orbit, and on-orbit operations. These procedures require a combination of traditional aircraft and spacecraft control effectors. To command this gamut of actuators, existing aerospace vehicles piece together different control strategies according to the current flight phase or transition. As aerospace vehicles expand the operational envelope, carrying with them a host of control effectors, piecemeal control algorithms become cumbersome and complicated. In order to move toward flexible and robust control approaches it is necessary to logically separate the tasks of control and actuator assignment. A control allocation algorithm supplements a control law by distributing control responsibility amongst the different families of currently available actuation devices.

1.1 THESIS OBJECTIVES

The objective of this research task is to develop a simple and physically intuitive control allocation algorithm for an entry vehicle. The algorithm will translate generalized control commands into RCS and aerosurface commands, applying feasible limits of the physical hardware as necessary. Reliable and efficient reconfiguration as the flight environment evolves or in the instance of actuator failure must be inherent to the design. In addition, the design should exhibit computational efficiency insofar as to not preclude eventual onboard implementation.

The aforementioned tasks coincide with the Draper Laboratory's ongoing development of next-generation guidance and control algorithms. Advances in real-time trajectory generation and abort technology will hasten the departure of sequential, gain-scheduled control systems. As more robust control strategies are required, control allocation of redundant actuators will become a necessary design element.

1.2 THESIS PREVIEW

The chapters in this thesis illustrate the development and evaluation of the control allocation algorithm. Chapter 2 describes the genesis and progression of control allocation techniques, characterizing this research effort in the context of other approaches. As described above, control allocation is a single element in a complete flight control structure; Chapter 3 is an overview of the comprehensive model. Included in this chapter are descriptions of necessary simulation elements, such as a vehicle model and guidance and control algorithms. The details of the control allocation algorithm are revealed in Chapter 4. This chapter includes an academic discussion of the algorithm and practical issues surrounding application to entry flight. Open loop simulations isolating the control allocation algorithm are also presented in Chapter 4. Closed loop simulations follow in Chapter 5. The first sections in Chapter 5 consider entry flight, highlighting the transition from exo- to endo-atmospheric flight. This control effector transition implies efficient revision of actuator assignment as the flight environment evolves. The latter sections in Chapter 5 investigate less common circumstances requiring actuator reconfiguration; the cases presented here involve actuator failure. The final section in Chapter 5 briefly discusses the computational efficiency exhibited by the algorithm throughout all simulations. Control allocation is a model-based algorithm, and Chapter 6 explores the relationship between errors in state feedback and modeling errors. Lastly, Chapter 7 discusses conclusions to be drawn from this work and recommendations for future research.

2 ENTRY CONTROL OVERVIEW

This chapter presents a brief evolution of entry control techniques and control allocation methods. Control allocation algorithms were born when deficiencies surfaced in the guidance and control algorithms that govern advanced aerospace vehicles. Because this thesis applies control allocation specifically to the entry problem, the heritage of advanced aircraft guidance and control is ignored in favor of the Space Shuttle. Current demonstrators of entry and reusable launch technologies still apply the guidance and control algorithms developed for the Space Shuttle. Alternative designs have matured but none has replaced the original approach. Many of these alternative guidance and control algorithms employ some form of control allocation. The underlying problem in control allocation is to find a way to map generalized control requests into meaningful actuator commands. Currently, methods of solving the allocation problem fall into two groups: those that consider the geometry of the problem and those that do not.

2.1 SPACE SHUTTLE

The legacy of Space Shuttle entry guidance and control has survived to this day because it is a proven solution. Prescribed entry trajectories, heavily constrained by heating concerns, are executed by sequentially applied control strategies during descent. Entry trajectories consist of two basic parts: an initial high angle-of-attack portion intended to induce drag and lower the vehicle's energy state, and a low angle-of-attack segment for trajectory control [1]. The shift between these trajectory divisions incorporates an obvious angle-of-attack transition and a control effector transition. In general, RCS jets are treated as high altitude, low dynamic pressure controllers while aerosurfaces dominate the terminal flight phase where the air is thicker and the vehicle speeds are slower [2]. Particular attention is paid to the yaw channel during transition because the rudder is the last aerosurface to gain adequate control authority. This is true because the vehicle body effectively blocks airflow over the rudder at high angle-of-attack. Aileron and elevator could be used to control yaw while the rudder is ineffective, but then only two aerosurfaces are controlling three axes and control of each axis is no longer independent of the other axes. Because the need for blended control between the RCS jets and aerosurfaces is evident, the entry digital autopilot (DAP) uses gain-scheduled control commands to aileron, elevator, and rudder while simultaneously applying a simplified phase-plane logic for supplemental RCS control [2]. The entry DAP philosophy treats the jets as low frequency control devices, sufficient to provide rate damping and attitude limiting, with wide deadbands, particularly for pitch and roll. The assumption is made that the aerosurfaces, high frequency control devices, will exert precise control within these deadbands.

Figure 2-1 shows the basic architecture of the entry DAP. The principal characteristic is the separation of lateral and longitudinal dynamics, a shortcoming from a control perspective. If the vehicle rolls it loses lift and drops in altitude, however, the longitudinal controller does not respond to this maneuver until sensors detect altitude errors. The control loops must react to each other rather than work together to efficiently achieve common goals. Separating longitudinal and lateral dynamics ignores the coupling between axes that exists in a typical flight vehicle. Inertia and the influence of control surfaces cause such coupling.

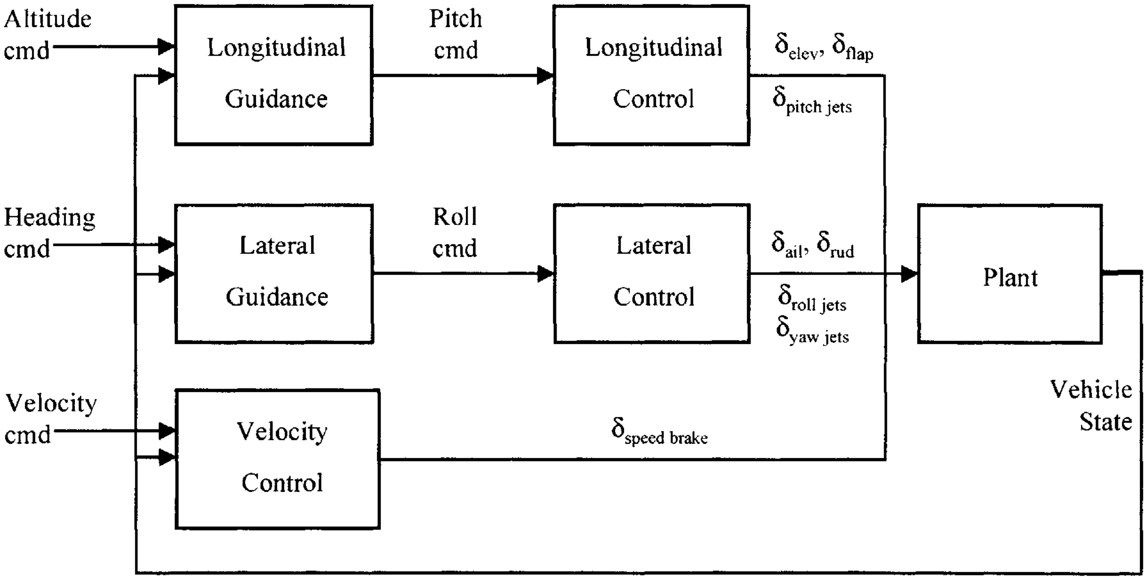

Altitude cmd
Longitudinal Guidance
Pitch cmd
Longitudinal Control
δ_{elev}, δ_{flap}
$\delta_{pitch\ jets}$
Heading cmd
Lateral Guidance
Roll cmd
Lateral Control
δ_{ail}, δ_{rud}
$\delta_{roll\ jets}$
$\delta_{yaw\ jets}$
Plant
Velocity cmd
Velocity Control
$\delta_{speed\ brake}$
Vehicle State

Figure 2-1. Shuttle-based Guidance and Control

Figure 2-2 is a closer look into the longitudinal control block from above. The figure illustrates a control allocation deficiency evident in traditional entry guidance and control algorithms. The presence of many single-input-single-output (SISO) loops greatly limits coupling of redundant actuators. The body flap, elevators, and pitch jets provide an example of such a limitation. These are redundant actuators during entry, but they are scheduled via separate control loops with predetermined control gains. Because no cross-channel communication exists, each loop operates without knowledge of what the other loops are doing. Managing groups of actuators with independent control logic works well for predefined trajectories. However, a Shuttle-based entry algorithm could give rise to reduced controllability in an instance of actuator saturation or failure. In these extreme cases the Shuttle-based algorithm must alter the specific allocation of physical controls, a process resulting in control inefficiency at the very least. The alternative is to schedule gains for all control redundancy/deficiency conditions. This is an undesirable design task, especially because these conditions will change appreciably during flight.

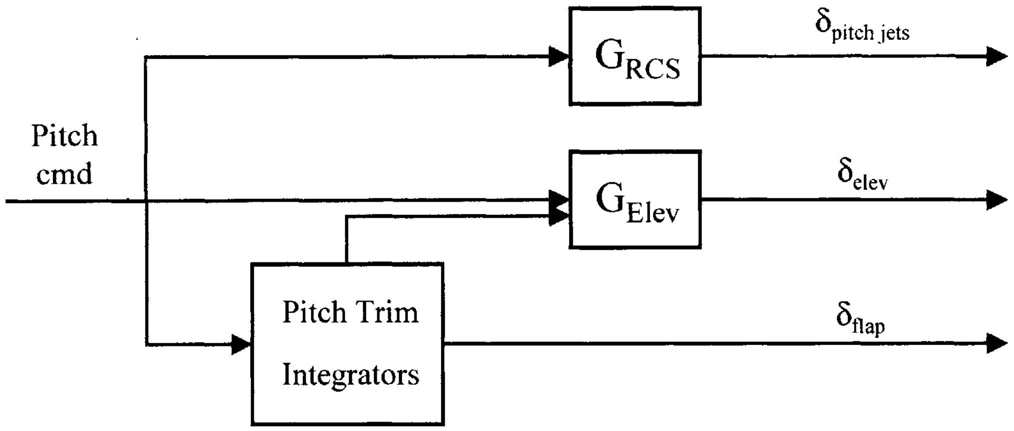


Figure 2-2. Shuttle-based Longitudinal Control

2.2 CURRENT EFFORTS

A multiple-input-multiple-output (MIMO) controller resolves many of the shortcomings evident in traditional SISO design. A multivariable approach considers the effects of all control inputs on all of the vehicle states. This allows the entire set of vehicle dynamics to be incorporated in a single design rather than separating dynamics and actuators into separate loops. Although MIMO designs offer the benefit of coupled dynamics, they tackle the subject of control allocation in essentially identical fashion as their SISO cousins: gain scheduling to account for instances of actuator redundancy and failure. A design is more straightforward and equally effective if it separates the control and actuator allocation problems (Figure 2-3). This alleviates the need for actuator gains and permits the MIMO design to produce meaningful vehicle commands in response to vehicle state errors. Regardless of the control law, control allocation algorithms are valuable design elements. This is true because, in the presence of control redundancy or deficiency, it is both simpler and more robust to design for actuator selection using a dedicated control allocation design. The marriage of control allocation and a multivariable control law is not essential; any control law can be designed to produce generic rotational and/or translational commands. However, pairing a MIMO design and control allocation algorithm results in a flexible and robust flight control structure, precisely the characteristics required by the next generation of aerospace vehicles.

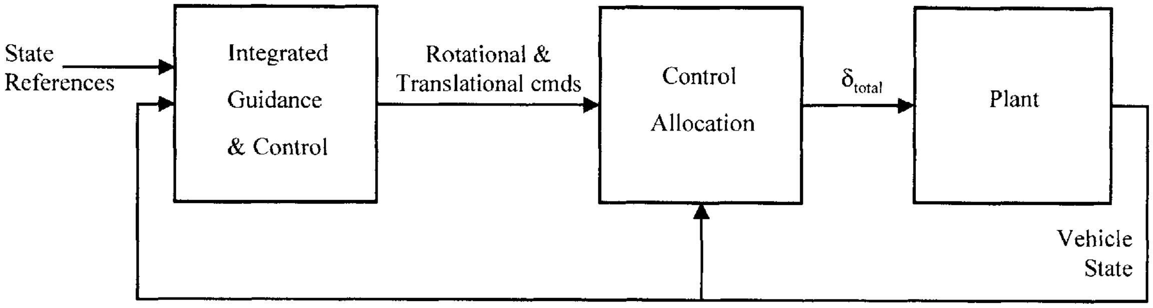


Figure 2-3. MIMO/Control Allocation Architecture

2.2.1 The Control Allocation Problem

The constrained control allocation problem is as follows: derive physically meaningful actuator commands from generalized control commands. Three possible outcomes exist: 1) one unique solution, 2) many acceptable solutions, and 3) no solution. In the first case no control redundancy exists. This problem can be solved just as effectively by a controller directly commanding actuator deflections. In the second case there is control redundancy and some method of actuator prioritization must be established in order to distribute commands between redundant actuators. In the final case there is control deficiency. The problem now becomes a prioritization between roll, pitch, and yaw because available actuators cannot simultaneously satisfy all control commands.

Initial control allocation efforts used pseudo-control or pseudo-inverse solutions to map generalized controller commands to aerosurface commands [3]. Conventional pseudo-control and pseudo-inverse techniques do not explicitly allow for actuator limits or designer preferences for specific effector usage. In order to realize these goals the control laws must be carefully tuned, much like the original Shuttle-based algorithm. Based on experience gained from these first attempts, designers chose to augment pseudo-inverse and pseudo-control methods with objective functions. This methodology enforces actuator limits and provides performance criteria through the objective function. At present, most control allocation techniques rely on some form of objective optimization. However, a technique also exists that solves for the optimal allocation solution without using an objective function. This method will be described first, followed by approaches using objective optimization.

2.2.2 Geometric Approach

The geometric problem is developed in references [4-6]. Simply put, this method first defines an attainable moment set based on all available physical control actuators and their limits. Torque commands, the generalized control command, represent a vector within this attainable moment set. One benefit of the geometric approach is the utilization of torque commands. Specifying physically meaningful control variables permits the designer to use intuition when tackling the control and control allocation problems. The set of actuators that geometrically aligns with torque commands is chosen as the optimal control effector combination. Whereas an objective function provides user-defined optimization criteria, the optimal solution in this approach is always defined as the collection of actuators providing the maximum torque. A stated disadvantage of this method is the complexity involved with computing the geometry of the problem with every actuator selection [4]. However, a distinct advantage of this approach is the guarantee that the issued commands will yield the maximum attainable moments.

2.2.3 Objective Optimization

Objective functions are used for three basic purposes: to extract performance optimization (in the control redundancy case), to establish axis prioritization (in the control deficiency case), and to enforce feasible limits on the actuator selection process. Several real-time and off-line allocation algorithms are summarized in references [7-10]. Algorithms of varying complexity and computational intensity exist, but most have been applied exclusively to tactical aircraft. Due to anticipated aggressive maneuvering, the primary concerns addressed by these algorithms are control deficiency, command limiting, and integrator wind-up due to actuator saturation. Additionally, generalized control commands in these approaches, while mathematically elegant, do not hold any physical significance. The insight that designers typically have with regard to the magnitudes of the physical controls and the axes in which they produce rotation is lost with the use of generalized controls.

2.2.4 Research Implementation

For most entry vehicles, control redundancy exists only during the transition from high to low angle-of-attack. At all points afterward the control allocation problem results in a unique, aerosurface-only solution because the RCS jets no longer offer redundancy. Given this fact, the geometric approach to control allocation is not appealing because the extra computational penalty will rarely produce a superior solution. However, algorithms that specifically focus on actuator saturation and command limiting are not necessarily applicable to an entry vehicle either. Even when presented with the prospect of more aggressive trajectories, actuator saturation and the hazards of integrator wind-up will not dominate the design of entry vehicles. Control allocation methods developed for tactical aircraft do not address the unique issues of blended RCS/aerosurface control. Two control allocation algorithms are specifically tailored to the entry problem. One approach is a robust pseudo-inverse problem intended for off-line computation [9]. The second is an optimization method developed at the Draper Laboratory [10]; it was a direct extension of on-orbit efforts to blend the effects of RCS jets and control moment gyroscopes.

Within the general control allocation framework, the method developed in this thesis is of the optimized objective variety. The algorithm relies on a bounded linear programming with an objective function that can easily be tailored to reflect specific performance goals. Simplicity and a desire to retain physical insight into the control allocation problem were key design drivers. Issues such as command limiting and integrator wind-up are not specifically addressed by the algorithm, and, although the geometric approach to control allocation is not pursued, the algorithm developed in this text retains the benefits of using physically meaningful rotational commands.

3 SYSTEM ARCHITECTURE

The purpose of this chapter is to illustrate the top-level design environment. This design encompasses control allocation in a real-time, onboard autonomous guidance and control system. The simulation architecture, shown in Figure 3-1, incorporates vehicle flight dynamics, onboard trajectory generation, and model-based guidance and control algorithms. This chapter will provide a summary of the aforementioned design elements. These elements are used to support and evaluate the control allocation routine. To facilitate assessment of the allocation algorithm, a simple MIMO control design is utilized that incorporates gain-scheduled control laws. All elements presented are crucial for both a comprehensive flight control system and closed-loop simulation. Still, the systems described in this chapter serve only to demonstrate the capabilities of the control allocation algorithm. Subsequent chapters will divulge the details of the linear program and other necessary control allocation design elements.

3.1 OVERVIEW

Figure 3-1 is a top-level block diagram of the simulation environment. This particular control design was selected primarily because its modular design allows for simple incorporation of the control allocation system.

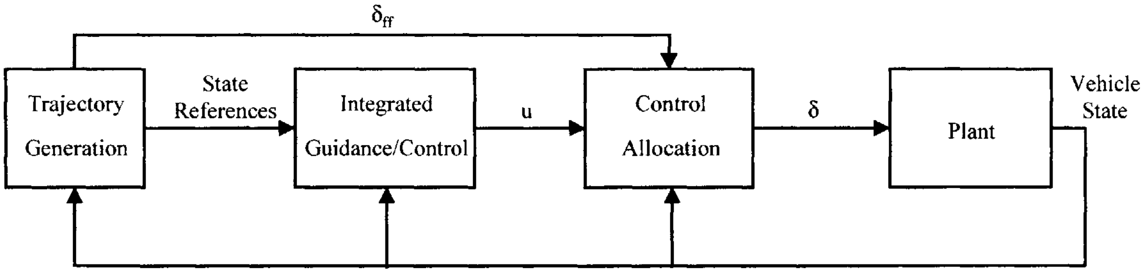


Figure 3-1. Simulation Architecture

The trajectory algorithm generates a course to glide the vehicle to a landing site. This block provides full-vehicle state references and feed-forward actuator trim conditions. The full-state linear quadratic guidance and control algorithm regulates state errors about the trajectory references. The controller issues generalized vehicle torque commands and speed brake commands with respect to the trimmed vehicle conditions dictated by the trajectory. The use of generic rotational commands is a key element of this modular architecture. This separates the tracking problem, still accomplished by the controller, from the control allocation problem. The control allocation algorithm solves for actuator commands, both aerosurface and RCS, that provide the desired vehicle moments. Actuator commands drive the vehicle model, or plant, which consists of vehicle dynamics and sensors.

3.2 VEHICLE DESCRIPTION AND DYNAMICS

The X-34 was selected as the sample entry vehicle for two primary reasons. First, the X-34's current guidance and control design is similar to that of the Space Shuttle. The X-34 is therefore a perfect example of Shuttle-based guidance and control algorithms; algorithms that the architecture presented in Figure 3-1 proposes to replace. Second, the Draper Laboratory has worked closely with Orbital Sciences Corporation, the primary designers of the X-34, during the development of the vehicle guidance algorithms. A byproduct of this relationship is a large amount of available technical information pertinent to the vehicle.

The X-34 is designed to be a test bed for reusable launch vehicle technologies, and although it is not intended for launch or orbital operations, it still enters the atmosphere under conditions similar to those encountered by a reusable launch vehicle. The vehicle is dropped from the belly of an L-1011, at which time the engine propels it to an approximate altitude and speed of 250,000 feet and Mach 8, respectively [11]. From this point the vehicle applies a typical entry trajectory as described in Section 2.1.

3.2.1 Stability

A diagram of the X-34 is presented in Figure 3-2 while Table 3-1 summarizes related physical characteristics. Of particular interest in Table 3-1 is the disparity between launch and landing weight. This difference is due almost entirely to the expulsion of propellant

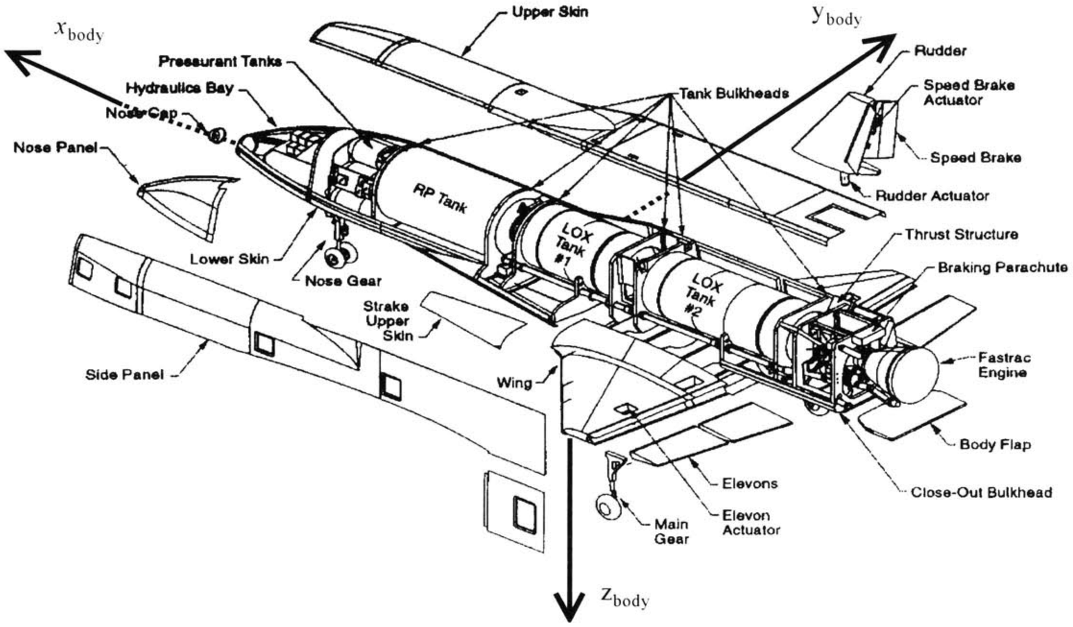

x_{body}
Upper Skin
y_{body}
Pressurant Tanks
Hydraulics Bay
Nose Cap
Nose Panel
Tank Bulkheads
Rudder
Speed Brake Actuator
Speed Brake
Rudder Actuator
RP Tank
LOX Tank #1
LOX Tank #2
Thrust Structure
Braking Parachute
Lower Skin
Nose Gear
Strake Upper Skin
Side Panel
Wing
Fastrac Engine
Body Flap
Elevons
Elevon Actuator
Main Gear
Close-Out Bulkhead
z_{body}

Figure 3-2. Diagram of the X-34 [12]

during powered ascent. The rapid drop in mass also dictates a dramatic change in the vehicle's mass distribution. By the time the X-34 reaches the apex of its flight and the engine is switched off, the center of gravity is pushed well aft of the center of pressure, a statically unstable aerodynamic configuration. Except in situations involving high performance tactical aircraft, this pitch instability is usually undesirable. Nonetheless, the X-34 is an unstable platform throughout entry, an inevitable result of a trade between launch and landing stability and control requirements. Fortunately, control techniques can reliably be used to augment vehicle dynamics, providing a stable closed-loop system.

Table 3-1. Physical Characteristics of the X-34 [11]

Length	58.3 ft
Wing Span, b	27.67 ft
Mean Aerodynamic Chord, $\overline{c}$	14.54 ft
Planform Area, S	357.5 ft^2
Approximate Launch Weight	46,500 lbm
Approximate Landing Weight	18,000 lbm

3.2.2 Actuators

The actuation devices used to stabilize the vehicle during entry are outlined in Table 3-2. The bandwidths are approximate and represent the minimum expected capabilities of the actuators. The aerodynamic control surfaces include a rudder, speed brake, body flap, and elevons. The elevons accomplish the traditional functions of both elevators and ailerons. Aileron control is achieved through differential commands to the elevon actuators, while elevator control is carried out with synchronized elevon commands. The RCS contains ten jets, each producing sixty pounds of force in a vacuum. The jets are clustered about the main engine at the rear of the vehicle (Figure 3-3). Thrust produced by jet firings is translated into rotational motion via the offset between jet positions and the vehicle's center of mass. The activity of any or all of these control devices exerts forces and moments on the X-34 and represents the means by which the autopilot controls vehicle attitude.

Table 3-2. Entry Actuator Characteristics

Control	Symbol	Range of Motion [deg]	Actuator Bandwidth [Hz]
Elevon	δ_e	-30 to 20	8
Aileron	δ_a	-20 to 20	8
Rudder	δ_r	-20 to 20	6
Body Flap	δ_{bf}	-15 to 10	1
Speed Brake	δ_{sb}	0 to 90	0.5
RCS	R,P,Y1,Y2,Y3 (see Table 3-3)	ON / OFF	Not Applicable

The sign convention for the control devices is summarized in the following paragraph. In all cases the standard body-centered reference frame is used (Figure 3-2). Regarding directional references, assume the observer is positioned as the pilot of the vehicle. A positive δ_a signifies downward motion of the right elevon and upward motion of the left elevon. This causes the vehicle to roll left, a negative rolling moment. A positive δ_e represents downward motion of both elevons and a positive δ_{bf} denotes downward motion of the body flap. Either movement initiates a downward pitching of the vehicle nose, a negative pitching moment. A positive δ_r implies that the trailing edge of the rudder moves toward the left wing, causing the vehicle to yaw left, a negative yawing moment. Additionally, due to coupling of the lateral dynamics the effect of aileron and rudder motion is not as simple as outlined above. Positive aileron motion usually produces positive yawing moments in addition to the desired negative rolling moments. Likewise, motion of the rudder tends to produce both yawing and rolling moments. The speed brake induces minimal drag when δ_{sb} is 0° and maximum drag when both partitions are fully open at 90°. The family of RCS jets is geometrically arranged such that specific combinations of jet firings can produce moments about all three body-centered axes. Figure 3-3 and Table 3-3 define the logical firing patterns and resulting rotations. The presence of a three-tiered yaw jet hierarchy stems from control concerns initially expressed in Section 2.1. Multiple jet thrust levels lead to increased controllability when the rudder offers little control authority.

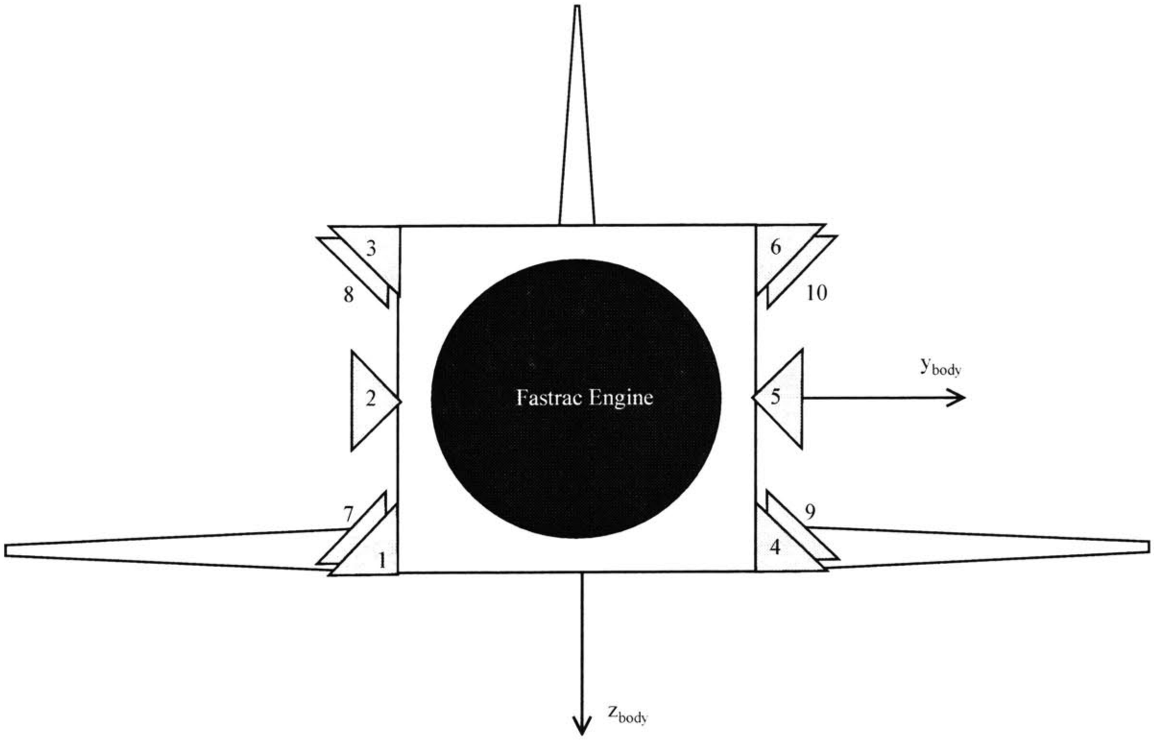


Figure 3-3. X-34 RCS Configuration

Table 3-3. X-34 RCS Firing Patterns

Rotation	Direction / Sign	Active Jet(s)
Roll (R)	Left / Negative	7 & 10
	Right / Positive	8 & 9
Pitch (P)	Down / Negative	3 & 6
	Up / Positive	1 & 4
Yaw (Y1)	Left / Negative	2
	Right / Positive	5
Yaw (Y2)	Left / Negative	1 & 3
	Right / Positive	4 & 6
Yaw (Y3)	Left / Negative	7 & 8
	Right / Positive	9 & 10

3.2.3 Equations of Motion

Entry flight involves both lateral and longitudinal dynamics. A complete representation of rigid body vehicle dynamics requires twelve quantities: three position states, three velocity states, three attitude states, and three attitude rate states. Table 3-4 identifies the state variables selected for this application. This set of state variables, generally referred to as flight path components, is applied because it allows simple integration of the guidance and control functions [3]. The combination of these twelve variables completely describes the state of the X-34 at any point in space. Using these state variables, the full nonlinear equations of motion were obtained using Newton's second law and the conservation of momentum. The definition and derivation of the nonlinear and linear equations of motion, coordinate systems, and reference frames has been completed in previous work and can be found in [13].

Table 3-4. Description of State Variables

State Description	Symbol
Downrange Position	x
Crossrange Position	y
Altitude	h
Inertial (Ground-Relative) Speed	V
Flight Path Angle	γ
Heading Angle	χ
Bank Angle about the Velocity Vector	μ
Angle of Attack	α
Sideslip Angle	β
Body Roll Rate	P
Body Pitch Rate	Q
Body Yaw Rate	R

Before departing from this brief synopsis of X-34 dynamics it is important to list one important assumption that was used in deriving the differential equations. The vehicle mass properties, including mass, products and moments of inertia, and center of mass location, are treated as constants during entry. This would obviously be a poor assumption during powered ascent but it seems reasonable for the entry portion of flight. Error may be introduced because the X-34 uses RCS jets for control during a portion of entry. However, even if the entire supply of RCS propellant is expelled, the vehicle will experience only a 0.44% change in landing mass. Of course, even small changes in mass can result in large deviations in the center of mass location. Consequently, the assumption that mass properties are constant reduces to the assumption that the center of mass of the RCS propellant lies close to the vehicle center of mass.

3.3 TRAJECTORY GENERATION

The trajectories defined by the onboard trajectory generation algorithm contain references for all state variables and trimmed aerosurface positions at every point along the trajectory. The guidance and control and control allocation algorithms use these references. For this research, the state variable references and trimmed aerosurface information for the entire trajectory are calculated prior to simulation and stored as tabulated data. An independent variable must be defined before accessing trajectory references during flight. The downrange position, x, is selected as this independent variable. For an unpowered vehicle, the estimated downrange position is an important parameter in identifying the current desired state because it represents the remaining flight time. For this reason, the reference profiles that describe the flight path are stored as functions of this quantity.

3.4 GUIDANCE AND CONTROL

Perfect navigational information is presumed to be available for simulations presented in this text. All states of the dynamic system must be estimated or directly measured because the control law provided is full state. The algorithm is fundamentally a linear quadratic regulator (LQR) design that combines the functions of guidance and control. The rudiments of the design are taken from [13]. A block diagram of the LQR can be seen in Figure 3-4. The design differs from a pure LQR in two senses. First, the regulator portion of the title has been dropped in favor of servo. When applied to a flight control problem, the algorithm aims to track commands rather than to regulate about a set point. In this case the commands are the state variables acquired from the trajectory generation block. Because these trajectory commands correspond to a trimmed vehicle state, the LQ-Servo makes small corrections about trim. Consequently, aerosurface commands issued by the control allocation algorithm are applied with respect to feed-forward trimmed aerosurface positions, also provided by the trajectory generator. The second deviation from pure LQR theory is the addition of integrators. Specific vehicle states are integrated in order to guarantee zero steady-state error on those particular states. Table 3-5 supplements Figure 3-4 by identifying variables in the block diagram.

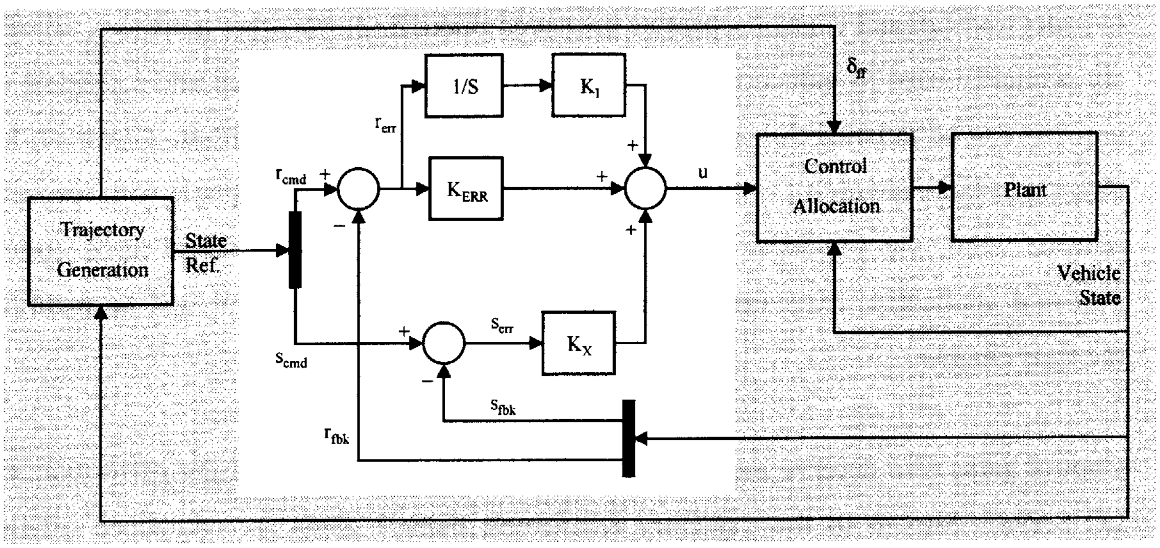

δ_{ff}
1/S
K_I
r_{err}
+
r_{cmd} +
K_{ERR}
+
u
Control Allocation
Plant
Trajectory Generation
State Ref.
−
+
s_{err}
K_X
+
s_{cmd}
−
s_{fbk}
r_{fbk}
Vehicle State

Figure 3-4. LQ-Servo with Integrators

Table 3-5. LQ-Servo Variable Definitions

Symbol	**Description**
r_{cmd}	Tracking Commands: y, h, V
r_{meas}	Tracking Feedback
r_{err}	Tracking Error: integrated for zero steady-state error
s_{cmd}	State References: γ, χ, μ, α, β, P, Q, R
s_{meas}	State Feedback
s_{err}	State Error
$K_{I,ERR,X}$	Control Gains
u	Control Commands: roll, pitch, and yaw torque + speed brake command

Note the choice of control variables; three are generalized rotational commands while the fourth is dedicated to an aerodynamic actuator. The control allocation algorithm assigns control responsibility amongst the entry actuators (Table 3-2) but the speed brake is excluded from this selection. The LQ-Servo produces a speed brake command and simply passes it through the control allocation block and into the plant model. This decision was made because the speed brake modulates to induce drag, which in turn affects velocity. The ability to exact translational control separates the speed brake from the other actuators, all of which are primarily used to induce moments, not forces. Both translational and rotational control requests can be satisfied through the use of a control allocation algorithm. However, because direct translational control is usually accomplished by rotating the vehicle with respect to the relative wind, the LQ-Servo is not configured to deliver generalized force commands. Potential application extensions involving direct translational control will be mentioned in the final chapter.

Control gains also deserve mention because they significantly affect the performance of the LQ-Servo. Similar to the trajectory references, control gains are calculated off-line and stored in table format. For simplicity in computation and implementation, time-invariant, steady-state LQR theory is used to calculate gains rather than applying optimal control principles. Because the vehicle model is a time-varying dynamic system, invoking a time-invariant routine involves multiple operating points at which LQR gains are calculated. The trajectory consists of a finite number of reference points; at each reference point a linear system is created to emulate the vehicle dynamics (Equation (3.1)). When sequentially applied as the vehicle travels along the trajectory, these linear models with time-invariant coefficients replicate the nonlinear, time-varying dynamic behavior of the entry vehicle.

LQR theory provides control gains by solving an optimization problem [14]. Consider the linearized system dynamics given in Equation (3.1), where A and B are constant coefficient matrices and $[\,A \;\; B\,]$ is assumed to be stabilizable.

$$\dot{\bar{x}} = A\bar{x} + B\bar{u} \tag{3.1}$$

A quadratic cost functional is defined based on the vectors of state variables, $\bar{x}$, and control variables, $\bar{u}$.

$$J = \int_0^\infty [\bar{x}^T Q \bar{x} + \bar{u}^T R \bar{u}]dt \tag{3.2}$$

Control gains are found by minimizing the cost, J, which involves solving the algebraic Riccati equation. Again, for a more detailed explanation of LQR theory and gain calculation please see [13,14].

The focus here will rest on the Q and R matrices; these are quantities that the designer uses to adjust the characteristics of the controller. Matrix Q is the state weighting matrix and R is the control weighting matrix. These matrices are used to specify the relative importance of minimizing both the state errors and control commands. Greater weighting on a state variable tends to produce tighter tracking of the reference, while greater weighting on a control variable discourages its use as a control command. Because LQR is a multivariable technique, the controller simultaneously considers the effects of all control inputs on all state variables. This means that adjusting a single weighting parameter can have unpredictable effects on the controller characteristics. Techniques are available to guide the designer in selecting weightings, but trial-and-error is acceptable for this application because LQR design is not the focus of this research. The controller is implemented only to demonstrate the control allocation algorithm. Little attention is paid to robustness, disturbance rejection, and sensitivity. Control performance is deemed acceptable provided that the vehicle dynamics are stabilized, trajectory references are tracked, and the control bandwidths do not exceed the physical capabilities of the actuators.

To summarize this chapter, the equations of motion illustrate the vehicle response to its environment, mass characteristics, and actuator states. These differential equations accurately describe the X-34's dynamic behavior. Without compensation from a controller, the X-34's equations of motion reveal instability in the longitudinal plane due to the relative location of the center of mass and center of pressure. Trajectory references and full-state feedback are provided to a controller, which provides the required stability augmentation and tracking of the trajectory commands. The vehicle model demands actuator commands to provide forces and moments that will effect a stable vehicle attitude throughout entry. The speed brake position is calculated by the LQ-Servo and the control allocation algorithm provides the remaining aerosurface and RCS commands in response to LQ-Servo moment commands.

4 CONTROL ALLOCATION

Chapter 4 explains how the control allocation algorithm translates torque requests into actuator commands. The control allocation block executes a linear program to determine the optimal mix of bounded aerosurface deflections and RCS jet firings that yield the commanded vehicle response. Each actuator within the allocation framework possesses an activity vector, an objective function coefficient, and upper and lower bounds that determine its control authority, desirability, and hardware limits, respectively. The fundamentals of the linear program and supporting control allocation elements are presented in this chapter. Please consult Appendix A for a detailed account of the linear programming technique.

4.1 OVERVIEW

In most entry vehicle control frameworks, particular actuators are dedicated to controlling specific rotations during specific flight phases. Custom logic is often introduced to decouple actuators that possess control authority in multiple axes. Additionally, aerosurfaces are usually used in predetermined ways to compensate for aircraft instabilities. This method might lead to reduced efficiency and control margin, especially in the case of actuator saturation or failure. The block diagram in Figure 4-1 displays an alternative to the traditional approach.

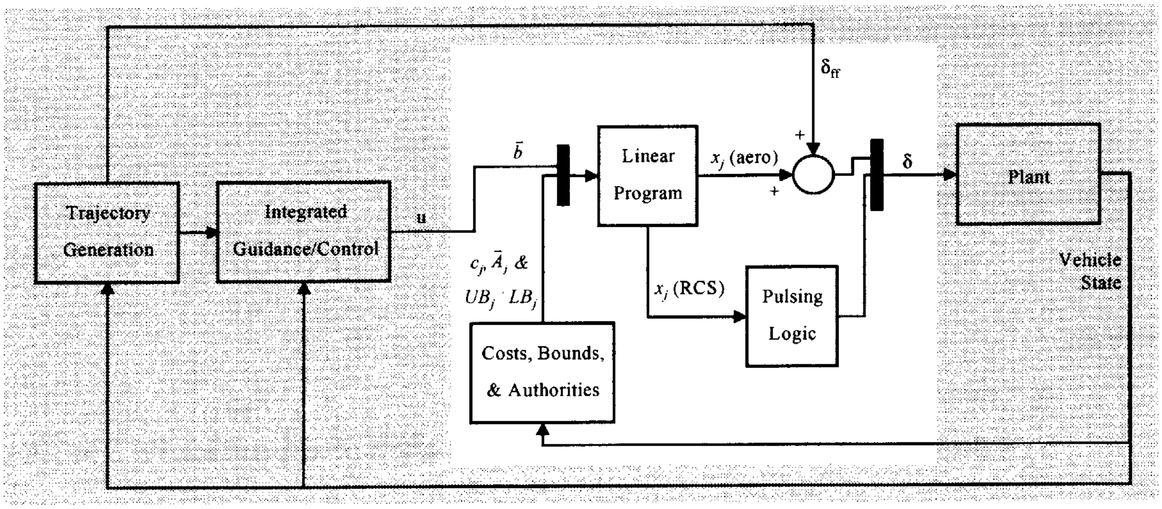

δ_{ff}
$\vec{b}$
Linear Program
x_j (aero)
+
+
δ
Plant
Trajectory Generation
Integrated Guidance/Control
u
$c_j, \bar{A}_j$ &
UB_j ' LB_j
x_j (RCS)
Pulsing Logic
Vehicle State
Costs, Bounds, & Authorities

Figure 4-1. Control Allocation Architecture

Control allocation logic does not explicitly assign aerosurfaces to pre-specified channels. The control request is a generalized vehicle command about independently controlled axes, in this case torque commands ($\vec{b}$), and all entry actuators are considered together in a common pool in an effort to satisfy these torque commands. Actuator control authorities are modeled by the activity vectors ($\vec{A}_j$), while actuator usage is encouraged or discouraged through the objective function coefficients (c_j) and upper and lower bounds (UB_j / LB_j). The value of each decision variable (x_j) in the linear programming solution corresponds to action of an aerosurface or RCS jet family. Any aerosurface activity derived from the control allocation algorithm is applied with respect to the feed-forward trimmed aerosurface positions. RCS solutions must also be discretized into jet on/off commands by the pulsing logic block.

4.2 THE LINEAR PROGRAM

The linear program embedded within the control allocation design is summarized in the following statements and equations.

Minimize the objective function:

$$z = \sum_{j=1}^{n} c_j \left| x_j \right| \tag{4.1}$$

Subject to equality constraints...

$$\sum_{j=1}^{n} \vec{A}_j x_j = \vec{b} \tag{4.2}$$

and inequality constraints:

$$LB_j \leq x_j \leq UB_j; \qquad j \in \{1,2,\ldots,\mathrm{n}\} \tag{4.3}$$

where:

n = number of decision variables

x_j = decision variable corresponding to control effort of aerosurface/RCS jet family

c_j = objective function coefficient associated with the j^{th} decision variable

$UB_j \,/\, LB_j$ = upper/lower bounds associated with the j^{th} decision variable

$\bar{A}_j$ = activity vector representing the control authority of the j^{th} decision variable

$\bar{b}$ = generalized control command

Despite the purported use of linear programming, the objective function is nonlinear due to the absolute values. Aerosurface motion and RCS commands are unrestricted-in-sign and absolute values accomplish the goal of minimizing total control effort, both positive and negative. This presents an apparent contradiction between the objective function formulation and the prescribed linear programming technique. The algorithm resolves this dilemma by defining a weighted sum of nonnegative decision variables and a separate array containing their corresponding sign information. Algebraic operations are executed with nonnegative decision variables; this ensures that linear programming techniques are valid [15,16]. Using absolute values and separate sign information is not the typical approach when unrestricted-in-sign variables are involved. The conventional method expresses an unrestricted-in-sign decision variable x_j as the difference of two nonnegative variables $x_{j(a)}$ and $x_{j(b)}$ in each constraint and in the objective function [15]. This action doubles the size of a problem when all variables are unrestricted-in-sign, as is the case in this application. The algorithm used in this application is a tidy way to account for unrestricted-in-sign variables without creating twice the number of decision variables.

The primary function of the linear program is to map control commands into actuator commands. It accomplishes this task by determining the values of the decision variables that satisfy the equality constraints. The objective function and inequality constraints specify user-defined performance criteria and hardware feasibility limits, respectively. The following bullet statements attempt to further clarify these three elements of the linear selection.

- *Objective Function*: Each objective coefficient is defined by the designer and is chosen to encourage or discourage a particular actuator response. Because it is a minimization problem, assigning a coefficient of great magnitude to a decision variable penalizes the use of the actuator corresponding to that decision variable. These coefficients, also referred to as costs or penalties, are dynamically updated between control cycles during entry.

- *Equality Constraints*: These constraints define the feasible solution space of the decision variables. Each $\vec{A}_j$ corresponds to the ability of a particular actuator to offer instantaneous rolling, pitching, and yawing moments (per unit value of decision variable). The number of equations comprising the equality constraints is equal to the number of independent control axes (i.e., 3 for rotational control only, 6 for rotation and translation). The calculation of activity vectors is customized for each actuator family and is discussed at a later point. The $\vec{b}$ vector comes from the vehicle controller and its dimension also reflects the number of control axes.

- *Inequality Constraints*: These constraints further define the solution space. They are simply limits on the allowable value of each decision variable and might represent such quantities as aerosurface displacement restrictions. Bounded decision variables are necessary to ensure that the solutions of the linear program do not violate vehicle hardware constraints. Bounds, like objective function penalties, might also be used to promote particular actuator combinations.

The linear program solves the constrained optimization problem with a bounded simplex-based algorithm. The simplex method is an algebraic procedure but it can easily be illustrated through geometry. The following two-dimensional example problem affords insight into the fundamentals of the algorithm.

Example 1: minimize, $z = x_1 + x_2$

subject to, $$\begin{bmatrix} 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 4$$

and, $x_1, x_2 \geq 0$

In this example, the equality constraint corresponds to a line and no upper bounds are applied. Only the first quadrant is shown in Figure 4-2 because the non-negativity condition restricts the feasible solution space to this area. Solutions that satisfy both equality and inequality constraints lie along the line segment. Of particular note are the corner points, defined as intersections between constraint boundaries. An essential principle exploited by the simplex method is the fact that a basic feasible solution (BFS) exists at a corner point [15,16].

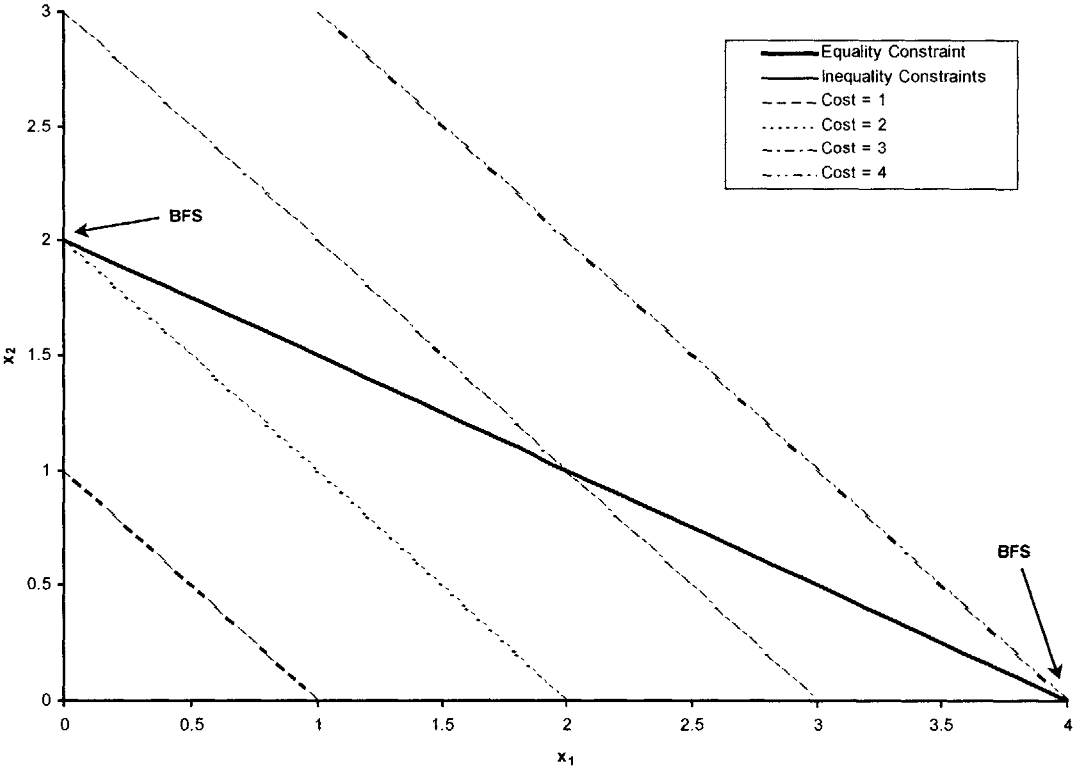


Figure 4-2. Geometry of Example 1

To see why this statement is true, examine the family of parallel lines representing objective functions of increasing cost (e.g., $1 = x_1 + x_2$, $2 = x_1 + x_2$, etc...). The objective function with zero cost intersects the origin. This is obviously the "cheapest" solution, but it does not satisfy all constraints because the objective function does not intersect the line segment representing the feasible solution space. The goal becomes straightforward: find the least expensive objective function that intersects the feasible solution space. From Figure 4-2 it is apparent that, whether minimizing or maximizing an objective, a corner point always yields an optimal solution. Because this is true, the simplex method begins iterations at a corner point and searches for the optimal solution by moving along constraint boundaries to adjacent corner points. The simplex method finds efficiency in limiting the search to only these corner point solutions. Iterations continue until a move to any other adjacent corner point cannot improve the current solution. When this occurs, the algorithm returns the current corner point and associated cost because this must be an optimal solution. In the minimization example problem the optimal solution is the point (0,2) with a cost of 2. Likewise, the other corner point solution, the point (4,0) with a cost of 4, is the optimum to the maximization problem.

The same principle applies when the problem is extended beyond two dimensions. During simulation there are n decision variables, n inequality constraints, and m equality constraints. Now the feasible region is no longer a line segment but a set of hyperplanes in n-dimensional space. Still, the solution yielding the minimum cost is found by testing feasible combinations of the n activity vectors. With each iteration activity vectors are swapped in and out of the solution. This corresponds to a switch between corner points within the feasible solution space. If the problem is properly posed, an activity vector exchange is only executed if it will improve, or at least maintain, the evaluation of the objective function.

4.2.1 Blending Techniques

One purpose of introducing control allocation logic is to blend the effects of aerosurfaces and RCS jets during the control effector transition as the vehicle enters the atmosphere. This can be accomplished in one of two ways. The first method is to use the bounds to redefine the feasible solution space. The linear objective function applied to both the example problem and this research effort is effective in minimizing control effort because nonzero decision variables are discouraged. However, if no upper bounds are actively enforced, the corner point solutions from this objective formulation contain only as many nonzero decision variables as equality constraints. The example problem has one equality constraint equation. Whether minimizing or maximizing the objective function, only one decision variable contributes a nonzero value to the optimal solution. All other decision variables must be zero, otherwise the result is a feasible solution but not an optimal solution. Now consider the same example problem with an upper bound of one applied to x_2. The geometry of this problem is shown in Figure 4-3. The solution space shrinks because answers containing x_2 greater than one are infeasible. Minimizing the objective function results in an optimal solution at (2,1) with a cost of 3. Note the nonzero contributions of both decision variables despite the fact that there is still only a single equality constraint equation. This concept also finds application in the simulation environment. During rotational control, when the number of equality constraints equals three, the optimal solution will utilize only three actuators if the upper bounds are inactive. Active upper bounds force the control allocation solution to include the effects of a greater number of actuators rather than force the action of the minimum number of control effectors.

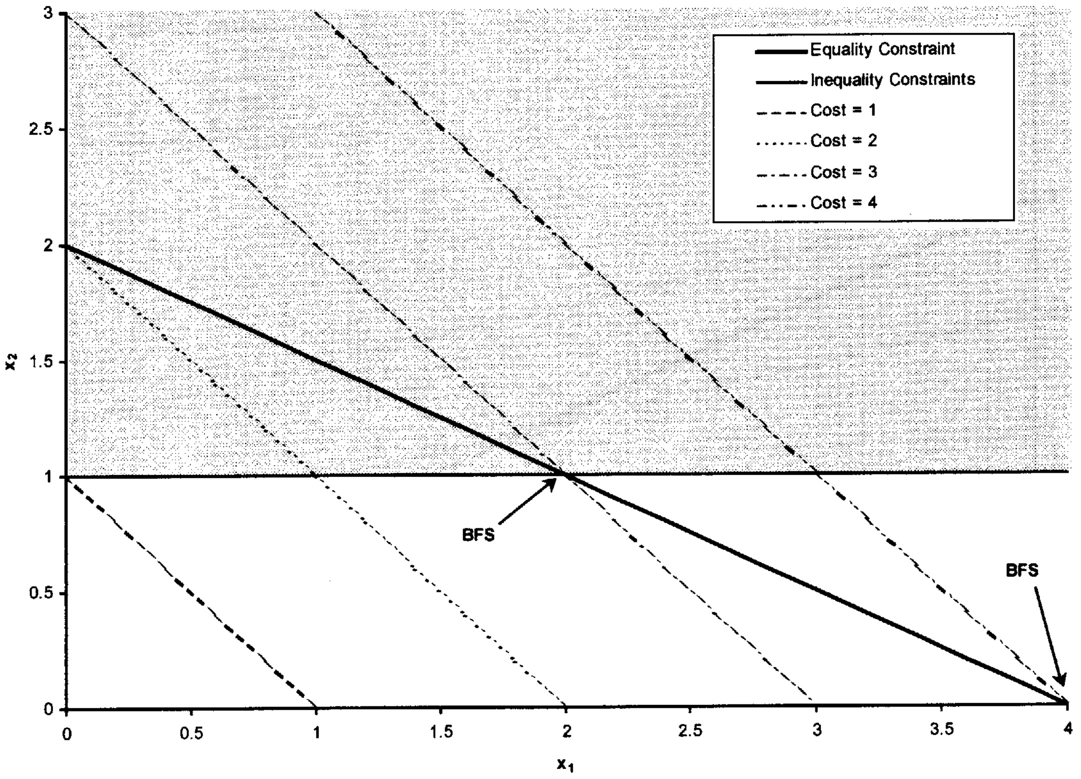


Figure 4-3. Geometry of Example 1 with Upper Bound on x_2

The other means by which the linear program can induce blending is through the objective coefficients. This method relies on the dynamic calculation of objective penalties between control cycles. Consider the original example problem's constraints and objective function to be a product of simulation at some initial time. The ensuing time step, with new control commands and linear selection, might yield identical constraints but a different objective. Figure 4-4 displays such a case. The decision variable x_2, in response to some environmental factor, is now four times more expensive in the objective function. The optimal solution to the minimization problem is no longer (0,2); it has shifted to the point (4,0). Increasing the objective penalty on x_2 removed it from the solution and it was replaced entirely by x_1. With regard to the entry application, this sort of substitution of one decision variable for another in successive control cycles could represent the dynamic effector transition from RCS yaw jets to rudder.

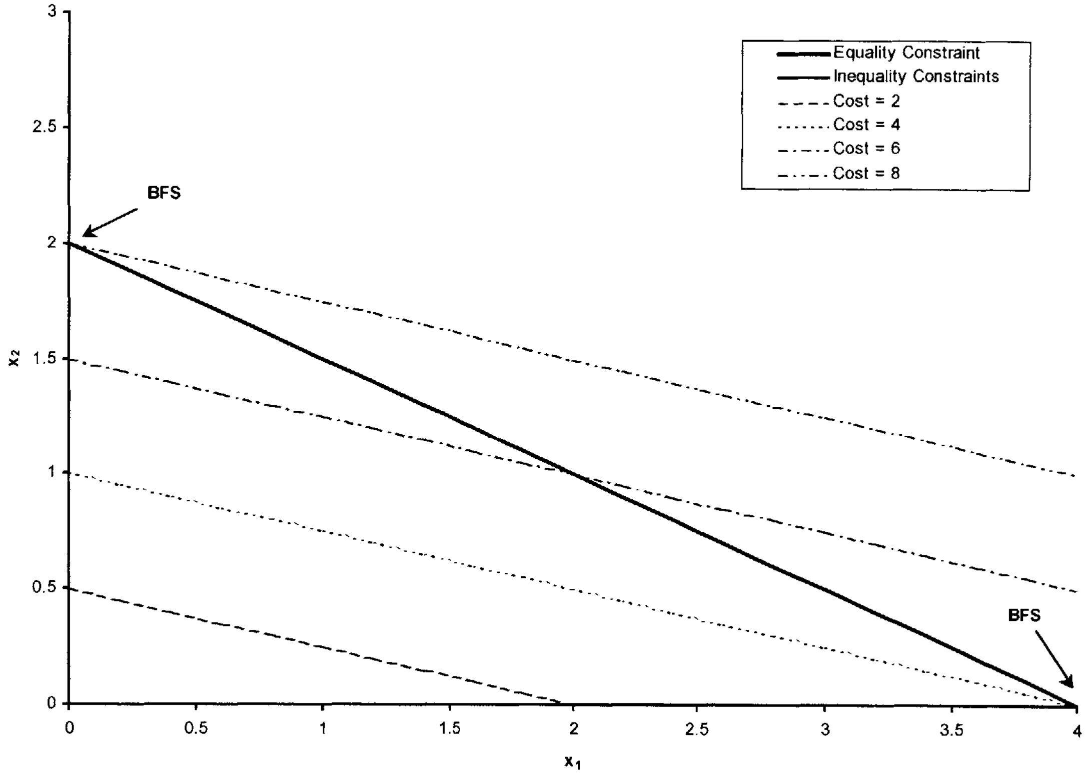


Figure 4-4. Geometry of Example 1 with Objective Function $z = |x_1| + 4 \cdot |x_2|$

The latter blending approach is adopted in this research effort. During simulation, LQ-Servo commands are issued every 20 milliseconds. The linear program must also allocate control responsibility amongst the entry actuators at this rate. The flight control frequency implies small torque commands and small updates to current actuator states with each time step. To use the former blending approach would require very tight, physically meaningless bounds on the decision variables. In an attempt to retain simplicity and tangible design insight, bounds are used only to reflect vehicle hardware limits. If the LQ-Servo tracks the trajectory effectively these bounds never become "active" because the actuators do not approach their limits.

This design decision means that, besides the objective coefficients, all elements of the control allocation algorithm are based solely on physical characteristics of the system. The inequality constraints signify actuator limits and the equality constraints are comprised of controller commands and estimates of actuator control authority. The control allocation problem is reduced from complicated, labor-intensive gain schedules to creative implementation of an objective function. The vehicle relies on the dynamic implementation of these objective penalties to create smooth effector transitions between redundant actuation devices.

4.2.2 Implementation Issues

A linear objective function can potentially produce a noisy and discontinuous solution history, even in the presence of smoothly varied constraints and objective coefficients. This occurs when the solution jumps between different currently optimal corner point solutions. Refer again to Figure 4-2 and Figure 4-4. In these two examples it is apparent that changes in either the constraints or objective penalties can result in optimal solutions at different corner points. No explicit measures are designed into the linear program to prevent this behavior. In the geometric sense, parallel or nearly parallel objective and constraints are the cause of such behavior.

Switching between drastically different solutions is not a problem in this application. Results in subsequent chapters expose choppy solutions only when examined on the millisecond time scale. Most of these irregularities can be attributed to either the discrete simulation environment or linearization errors introduced when deriving aerosurface activity vectors

(Section 4.5). The general trend reveals smooth actuator response. This can be attributed to the frequency at which the flight control system operates. Rare is the case when consecutive control commands are glaringly discontinuous because the controller is tracking trajectory cues at a rapid rate. When the input to the control allocation algorithm is well behaved the output tends to follow suite. Exceptions include cases when the linear program is asked to satisfy discontinuous torque commands or when RCS jets, naturally impulsive actuators, are used in coordination with aerosurfaces.

A quadratic objective function was briefly adopted for performance comparison to the linear objective. Quadratic optimization does not exhibit choppy response because, unlike its linear counterpart, optimal solutions are not restricted to corner points of the solution space. Quadratic objective functions are therefore less sensitive to slight changes in constraints and objective formulation. However, due to the rapid flight control frequency, the smoother responses inherent to quadratic programming are only apparent on a millisecond time scale. This benefit does not outweigh the additional computational burden accompanying quadratic programming so that avenue of research was not pursued. The simpler linear objective function is well suited to this application and seems to yield satisfactory results. This might not hold true for all problems, in which case quadratic optimization or nearly optimal quadratic programming might prove to be beneficial.

In Chapter 2, the general control allocation problem is shown to lead to one of three scenarios: 1) one unique solution exists, 2) many optimal solutions exist, or 3) no suitable solutions exist. The linear program enforces blending of control effectors to cope with the second scenario, but the other two situations also deserve mention. The third scenario arises if the controller issues an unsolvable set of torque commands to the control allocation algorithm. This is a rare phenomenon, only occurring if upper and lower bounds restrict the decision variables' action; unbounded decision variables can solve any command. To account for this possibility, unbounded "artificial" decision variables are used to augment the original problem. These variables are artificial in the sense that they correspond to no meaningful actuator activity. The linear programming problem now takes the form of Equations (4.4) through (4.6).

Minimize:

$$z = \sum_{j=1}^{n} c_j \left|x_j\right| + M \sum_{k=1}^{m} \left|y_k\right| \tag{4.4}$$

Subject to:

$$\sum_{j=1}^{n} \vec{A}_j x_j + \sum_{k=1}^{m} \vec{e}_k y_k = \vec{b} \tag{4.5}$$

$$LB_j \leq x_j \leq UB_j; \qquad j \in \{1,2,...,\mathrm{n}\} \tag{4.6}$$

where:

m = number of independent control axes

y_k = “artificial” decision variable

M = objective function coefficient associated with the “artificial” decision variables

$\vec{e}_k$ = column vector with 1 in the k^{th} row; other entries are 0

(all other variables are defined in Equations (4.1) through (4.3))

The solution is initialized with “artificial” variables because this guarantees a mathematically feasible solution to the linear programming problem. The first iterations naturally substitute “real” decision variables in place of “artificial” variables because the “artificial” variables are purposefully assigned an enormous objective penalty (M). Failure to remove an artificial variable from the optimal solution indicates that no physically realistic solution exists.

Introducing “artificial” decision variables safeguards against the worst-case scenario. Unfortunately, it also guarantees that the linear program will devote initial iterations to removing those same variables from the solution. For small problems such as this application (m =3, n =13), the extra iterations are not an excessive burden and they allow the algorithm to account for all three aforementioned scenarios. Nonetheless, these initial iterations are extra effort and they are especially obtrusive when one unique solution exists. For example, after roll and yaw jets no longer offer control authority, the only actuators capable of solving these commands are the aileron and rudder. The combination of these two aerosurfaces represents the unique solution to both roll and yaw torque commands. Ideally, the linear program requires no iterations to arrive at this conclusion. However, the algorithm in Appendix A uses “artificial” variables in all situations because it is a simple and effective approach. The programming technique always arrives at an optimal solution, regardless of whether that means a unique solution, the best choice from many possible solutions, or a solution that is only mathematically feasible.

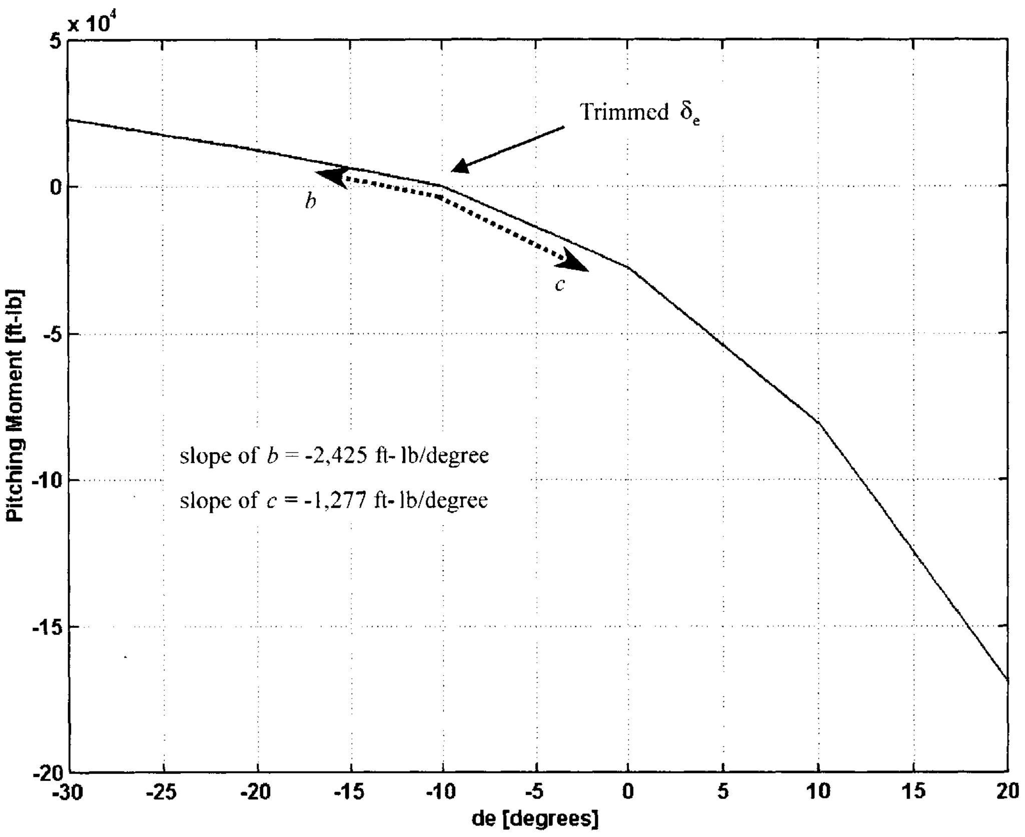


Figure 4-5. Pitching Moment vs. Elevon Deflection

A final precaution must be taken to accommodate potential aerosurface nonlinearities. Moving an aerosurface into the airstream can generate significantly greater torque than deflecting it into a region with stunted aerodynamic flow. An example case is shown in Figure 4-5. The section of curve where greater elevon deflection produces little change in torque evidences a shadowed region. This phenomenon usually occurs at high angles of attack when the wing blocks airflow from covering the top of the aerosurface. The linear activity vectors in Equation (4.2) must accurately reflect control authority, defined as the amount of torque generated when modulating an aerosurface from its trimmed position. Linearized aerosurface authority (i.e., the slopes of *b* and *c*) can be quite different for positive and negative deflections. Significant errors might be introduced into the linear programming solution if only one activity vector is used for both senses of aerosurface deflection. In order to avoid this situation, the linear program creates independent decision variables and consults different activity vectors for positive and negative aerosurface deflections.

4.3 UPPER AND LOWER BOUNDS

Bounds are applied to decision variables representing both aerosurfaces and RCS jets. Aerosurface bounds are used to enforce maximum displacement limits and RCS bounds define a deadband surrounding each family of jets. Calculating aerosurface bounds is straightforward and only requires knowledge of the hardware constraints ($\delta^{+/-}_{STOP(i)}$) and the trimmed actuator deflections ($\delta_{TRIM(i)}$).

$$\begin{aligned} UB_i &= \delta^{+}_{STOP(i)} - \delta_{TRIM(i)} \\ LB_i &= \delta^{-}_{STOP(i)} - \delta_{TRIM(i)} \end{aligned} \tag{4.7}$$

Of the n decision variables, the index i, in this equation and equations that follow, refers to those specifically tied to aerosurface actuators. Equation (4.7) prevents the linear selection from swiveling an aerosurface beyond its range of motion. Note that actuator slew constraints, related to the actuator bandwidths of Table 3-2, are not enforced through the bounds. The linear program is not used to limit modulation rates because it is a control responsibility to issue torque commands that do not violate actuator rate constraints. Control allocation is simply an algebraic mapping of torque commands to actuator commands and is not involved in monitoring or controlling actuator dynamics.

RCS decision variables are non-dimensional so their bounds have no relation to physical limitations. Hardware constraints, such as minimum pulse times, are accounted for in the pulsing logic function. The values of RCS decision variables define the fraction of torque from a jet family that is needed to satisfy the control command. For example, a decision variable with a value of one indicates that 100% of the available torque from that particular jet family is desired for this selection cycle. It seems logical to limit the RCS decision variables to one because the selection routine should not ask any more than 100% of any jet family, however, the linear program issues actuator commands every 20 milliseconds. The jets cannot pulse at 20 millisecond intervals, and even if the hardware permitted this behavior, the result would be very sloppy performance and an inefficient use of fuel. In an effort to avoid jet chatter, RCS bounds are applied in a range between one and four during simulation. The pulsing logic function does not direct a jet firing until the value of an RCS decision variable reaches such a bound. Jet decision variable bounds are used in this manner to define a deadband where attitude errors are tolerable. This is a region in which RCS decision variables increase in magnitude until a bound is reached and jets are fired to correct the growing attitude errors.

$$\begin{aligned} UB_l &= +B \\ LB_l &= -B; \qquad B = [1 \ldots 4] \end{aligned} \tag{4.8}$$

From this point forward, the index l refers to those decision variables among the original n representing RCS jet families.

4.4 OBJECTIVE FUNCTION COEFFICIENTS

Each decision variable also has an objective function coefficient that expresses its desirability. The only method employed to promote blending amongst RCS jets and aerosurfaces is the relative weight between redundant actuators. Provided that accurate control authority estimates are available, the control allocation algorithm selects appropriate actuator involvement even if all decision variables are weighted equally in the objective function. This results in an algorithm that simply chooses actuators offering the greatest control authority. The designer specifies objective coefficients because a lack of discrimination between redundant actuators results in poor control performance and inefficient actuator usage. The performance goals driving the design of the objective function coefficients in this research effort are as follows: to limit aerosurface chatter and aerosurface rate and position saturation. It is impossible to eliminate all aerosurface chatter during blended control because the aerosurfaces quickly respond to impulsive burns of the RCS jets. However, large deviations from the feed-forward trimmed aerosurface positions and rapid slew rates can be avoided with the proper objective function penalties.

The objective function presented here is one simple way to achieve the aforementioned performance goals. The general philosophy is to first try to realize torque commands with aerosurfaces. Aerosurfaces are the default choice of the control allocation algorithm because they provide smooth, continuous control and they require no fuel. Aerosurfaces are penalized more heavily than RCS jets only when torque commands either drive the aerosurfaces close to their bounds and far from their trimmed positions, or, torque commands drive the aerosurfaces at their maximum slew rates for many consecutive actuator selections. These conditions indicate that aerosurfaces are either flapping about or departing from their trim schedules in order to resolve controller commands. In these instances the aerosurfaces apparently lack sufficient control authority and jets should instead be used.

Aerosurface penalties consist of three parts: a bias, a deflection cost, and a rate cost. Equation (4.9) displays these aerosurface coefficients.

$$c_i = c_{bias} + c_{def} + c_{rate} \tag{4.9}$$

where:

$$c_{bias} = \text{constant}$$

$$c_{def} = K\left|\delta_i - \delta_{TRIM(i)}\right|$$

$$c_{rate} = \begin{cases} A \cdot tally^N & \text{if on bottom curve of hysteresis} \\ \left(tally/B \right)^{1/N} & \text{if on top curve of hysteresis} \end{cases}$$

The bias term dictates the general desirability of using an aerosurface decision variable. In this research only the body flap decision variables carry a bias. This is done to ensure that the elevons and the pitch jets are considered before resorting to longitudinal control via the body flap. The deflection penalty simply adds amplitude to the total cost according to deviations from trimmed positions. The multiplier in the deflection term allows the designer to easily adjust the magnitude of this portion of the cost. The rate penalty is the most important contribution to the aerosurface objective coefficients. This portion of the cost follows a hysteresis. As an aerosurface is modulated at its maximum slew rate, the quantity *tally* grows in magnitude and the penalty increases, following the bottom curve of the hysteresis path. The precise rate of increase in *tally* determines the number of selection cycles that pass before the aerosurface becomes too expensive for selection by the linear program. In turn, this establishes the acceptable level of aerosurface chatter. Once an aerosurface has been removed from selection and modulates back to its trim position, it is still important to retain a high cost because the aerosurface has displayed inadequate control authority. The quantity *tally* decreases in magnitude because the aerosurface is no longer modulating, but the penalty now follows the top portion of the hysteresis path. The rate of decrease in *tally* determines the number of selection cycles that pass before the aerosurface is again considered by the linear program for use in controlling the vehicle. Figure 4-6 shows examples of the hysteresis. The maximum values of c_{rate} and *tally* define the constants A and B in Equation (4.9) while N determines the shape of the hysteresis. These values are chosen by the designer and must be selected such that the aerosurface penalties are balanced against the costs assigned to RCS decision variables.

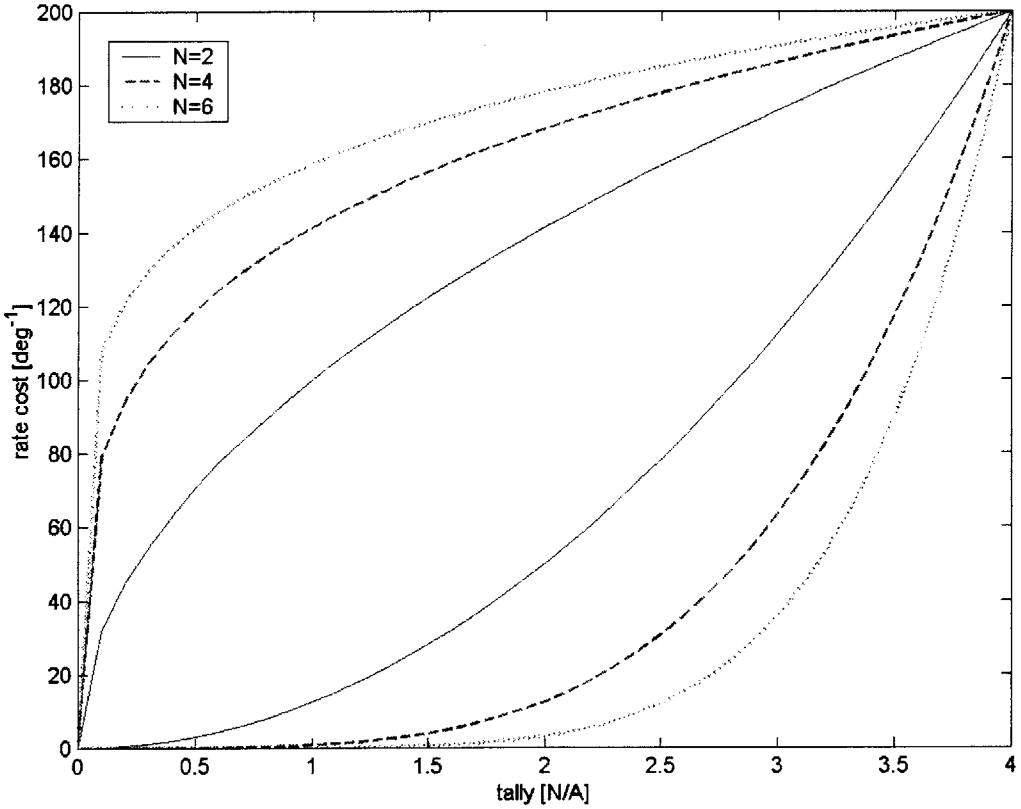


Figure 4-6. Example of c_{rate} Objective Coefficient Contribution

Jet penalties are simpler than aerosurface costs, consisting of only a single, constant term for each family of jets. For example, in the yaw channel, the least expensive jet family is Y1 and the most expensive is Y3. Because jet penalties are fixed, the balance between aerosurfaces and RCS decision variables depends entirely on the dynamic calculation of aerosurface costs.

4.5 ACTUATOR CONTROL AUTHORITIES

As introduced in Section 4.2.2, each aerosurface actuator has two decision variables and two corresponding activity vectors. One set represents positive actuator motion and control authority while the other corresponds to negative. The activity vectors are designed to accurately model the torque produced when aerosurfaces are swiveled from their feed-forward trimmed positions in either a positive or negative direction. This is accomplished through a simple linearization of the torque equations below. Geometrically, the slopes of line segments *c* and *b* in Figure 4-5 are examples of the pitch element in a positive and negative elevon activity vector, respectively.

$$\begin{aligned}
\tau_{roll} &= \bar{q}Sb\left[C_l + \left(\tfrac{z_{cm}}{b}\right)C_Y - \left(\tfrac{y_{cm}}{b}\right)\left\{-C_D\sin\alpha - C_L\cos\alpha\right\}\right] \\
\tau_{pitch} &= \bar{q}S\bar{c}\left[C_m + \left(\tfrac{x_{cm}}{\bar{c}}\right)\left\{-C_D\sin\alpha - C_L\cos\alpha\right\} - \left(\tfrac{z_{cm}}{\bar{c}}\right)\left\{-C_D\cos\alpha + C_L\sin\alpha\right\}\right] \quad (4.10) \\
\tau_{yaw} &= \bar{q}Sb\left[C_n + \left(\tfrac{y_{cm}}{b}\right)\left\{-C_D\cos\alpha + C_L\sin\alpha\right\} - \left(\tfrac{x_{cm}}{b}\right)C_Y\right]
\end{aligned}$$

where:

$\bar{q}$ = dynamic pressure [psf]

$S, b, \bar{c}$ (see Table 3-1, page29)

x_{cm}, y_{cm}, z_{cm} = center of mass offset from reference center of mass [ft]

C_l, C_m, C_n = non-dimensional aerodynamic moment coefficients (roll, pitch, yaw)

C_Y, C_D, C_L = non-dimensional aerodynamic force coefficients (side, drag, lift)

These torque equations apply exclusively to the aerosurfaces. The aerodynamic force and moment coefficients are stored in tables as functions of vehicle attitude, the environment, and actuator position. Using this tabular data and feedback from the plant it is possible to define aerosurface authorities as the vehicle state and environment evolve. Table look-ups are computationally expensive so assumptions and simplifications are made to expedite the linearization. One assumption is that the vehicle center of mass and the reference center of mass are collocated. The reference center of mass is the point about which all aerodynamic coefficient data is collected. If the true center of mass is not offset from the reference center of mass, no aerodynamic force coefficients need to be considered. This greatly simplifies the torque equations, as evidenced by Equations (4.11).

$$\begin{aligned} \tau_{roll} &= \bar{q}Sb\left(C_l\right) \\ \tau_{pitch} &= \bar{q}S\bar{c}\left(C_m\right) \\ \tau_{yaw} &= \bar{q}Sb\left(C_n\right) \end{aligned} \tag{4.11}$$

Further simplification stems from the nature of the control law. The controller issues torque commands with respect to a trimmed vehicle condition, which, by definition, entails zero moments acting on the vehicle. Accordingly, the control allocation algorithm applies changes in aerosurface deflections with respect to feed-forward trimmed aerosurface schedules. Therefore, the calculation of each activity vector includes the trimmed aerosurface position because it is always the origin of any motion. This is convenient because no table look-up is required to determine that the trimmed aerosurface position produces zero moment. All that remains is to establish a step size in both positive and negative directions from this trimmed aerosurface position. Because the control allocation algorithm only makes small corrections to the feed-forward trimmed deflections it is appropriate to use a small perturbation. Using a small step size also encourages greater accuracy in the linearized activity vectors, provided that the actual deflections do not stray far beyond the feed-forward positions. It is arbitrarily determined, as a performance measure, that the control allocation algorithm should produce aerosurface deflections within $^{+}/-$ 5° of the trimmed positions. The step size is chosen to be half these desired deflections in an effort to best describe the authority within the $^{+}/-$ 5° window. Equations (4.12) summarize the calculation of aerosurface activity vectors. These equations reveal that the calculation of each aerosurface activity vector calls for three table look-ups, one for each of the controlled degrees of freedom. Four aerosurfaces, each with two decision variables, require 24 table look-ups with every update of the aerosurface activity vectors.

$$\vec{A}_i^{+/-} \;=\; \begin{bmatrix} \left\{ \dfrac{\Delta \tau_{roll}}{\Delta \delta_i^{+/-}} \right\} \\ \left\{ \dfrac{\Delta \tau_{pitch}}{\Delta \delta_i^{+/-}} \right\} \\ \left\{ \dfrac{\Delta \tau_{yaw}}{\Delta \delta_i^{+/-}} \right\} \end{bmatrix} \;=\; \begin{bmatrix} \left\{ \dfrac{\bar{q}Sb(C_l|_{\delta^{+/-}_{PERT(i)}})}{\left(\delta^{+/-}_{PERT(i)} - \delta_{TRIM(i)}\right)} \right\} \\ \left\{ \dfrac{\bar{q}S\bar{c}(C_m|_{\delta^{+/-}_{PERT(i)}})}{\left(\delta^{+/-}_{PERT(i)} - \delta_{TRIM(i)}\right)} \right\} \\ \left\{ \dfrac{\bar{q}Sb(C_n|_{\delta^{+/-}_{PERT(i)}})}{\left(\delta^{+/-}_{PERT(i)} - \delta_{TRIM(i)}\right)} \right\} \end{bmatrix} \tag{4.12}$$

where:

$$\delta^{+/-}_{PERT(i)} \;=\; \delta_{TRIM(i)} \;{}^{+}/\!-\; 2.5^\circ$$

RCS activity vectors represent the amount of torque available from each family of jets. There are five decision variables tied to jets: one for roll jets, one for pitch jets, and three for different families of yaw jets. Unlike aerosurfaces, a single decision variable corresponds to one activity vector. Activity vectors are accurate in describing both positive and negative moments imparted to the vehicle. Each jet is designed to produce sixty pounds of force in a vacuum. RCS firings result in torque because the forces produced by the jets do not act through the vehicle center of mass. The location of each jet relative to the reference center of mass is stored in a three-element position vector. Calculating the rotational control authority of each jet is a simple matter of force and moment arm:

$$\vec{A}_l \;=\; \begin{bmatrix} \tau_{roll} \\ \tau_{pitch} \\ \tau_{yaw} \end{bmatrix} \;=\; \vec{r}_l \times F_l \tag{4.13}$$

where:

$\vec{r}_l$ = position of l^{th} jet relative to the reference center of mass [ft]

F_l = thrust produced by l^{th} jet [lbf]

An atmospheric jet firing produces different force and torque than a nominal vacuum firing. This is primarily due to plume interaction with aerodynamic flow and plume expansion in nonzero atmospheric pressure. Plume expansion attenuates jet thrust as altitude drops and ambient pressure rises. Estimating this effect for the calculation of activity vectors is accomplished using the relationship found below. This is the only correction to jet thrust incorporated in the activity vector calculations. Other plume impingement terms are dominated by the plume expansion correction and are difficult to model [10].

$$F_l = F_0 - A_e P_\infty \tag{4.14}$$

where:

F_0 = vacuum thrust [lbf]

A_e = jet nozzle cross-sectional area [ft^2]

P_∞ = atmospheric pressure [psf]

Lastly, jet activity vectors are assembled from individual jet control authorities according to the firing patterns in Table 3-3. For example, the control authorities of jets 8 and 9 are combined to form the activity vector for roll jets. Combining individual control authorities reduces the size of the problem by limiting the number of decision variables. It also allows each decision variable to reflect the logical firing patterns followed by the RCS.

4.6 PULSING LOGIC

The first function of the pulsing logic block is to map the values of decision variables, which represent an entire family of jets, to the individual jets that comprise each family. Building on the example at the close of the previous section, a positive decision variable corresponding to roll jets will apply to jets 8 and 9. Likewise, if that same decision variable is negative it is assigned to jets 7 and 10. The second function of the pulsing logic algorithm is to command efficient jet firings. Unlike the continuous behavior of aerosurfaces, a nonzero decision variable does not necessarily indicate action of RCS jets. It is not until a decision variable reaches its upper or lower bound that jets corresponding to that decision variable are commanded to fire. The hardware constraint adopted in simulation is a minimum jet pulse of 80 milliseconds. Once a jet is turned on, the pulsing logic function ensures that the jet fires for at least this minimum time. However, minimum jet pulses are often not sufficient to correct attitude errors. Rather than use a cycle of consecutive minimum firings, the pulsing logic dictates that a jet continues firing until its decision variable drops below a specified threshold. This threshold is defined by the designer and is usually a value close to zero. A jet is turned off before its decision variable is precisely zeroed in order to allow the vehicle to drift in the deadband toward zero attitude error.

In summary, the control allocation algorithm accepts torque commands from the controller and state variable and environmental feedback from the plant. Feedback is used to accurately estimate the rotational control authorities of the entry actuators and to calculate the bounds and objective penalties that are applied to decision variables. The simplex-based linear program solves for the values of all decision variables. Those values corresponding to aerosurfaces are added to feed-forward trimmed aerosurface deflections while the values of RCS decision variables are translated into jet firing commands by the pulsing logic function.

4.7 OPEN LOOP TESTS: CONTROL ALLOCATION ISOLATION

Closed loop simulation is the only way to validate a flight control structure that includes a separate control allocation algorithm. However, it is difficult to measure the performance of the control allocation routine when it is not isolated from the guidance and control algorithms. In order to separate the dynamics of the control law from the algebra of control allocation, open loop tests are conducted using arbitrary moment commands. Each open loop simulation is accomplished in an artificial environment at a constant altitude, velocity, and attitude. Characteristics of the control allocation algorithm, particularly the performance of the objective function, are easily highlighted through these simple simulations.

Three tests are conducted and each intends to underscore different characteristics of the control allocation algorithm. Each test is performed under different flight conditions. Table 4-1 lists the important simulation parameters. The commands issued to the control allocation algorithm are fabricated from step and sinusoidal functions and they do not attempt to represent logical attitude maneuvers. These tests are algebraic exercises in mapping torque commands to aerosurface deflections and jet firings. Actuator commands produced by the control allocation algorithm drive a vehicle force and moment model in order to compare the resultant moments to the commanded moments.

Table 4-1. Open Loop Simulation Parameters

Test #	Altitude (ft)	Mach [N/A]	Angle of Attack [deg]
1	45,000	0.7	15
2	175,000	3	25
3	240,000	5	20

The results of the first test are displayed in Figures 4-7 and 4-8. This is the simplest test because, as expected at this altitude, only aerosurfaces are required to satisfy the moment commands.

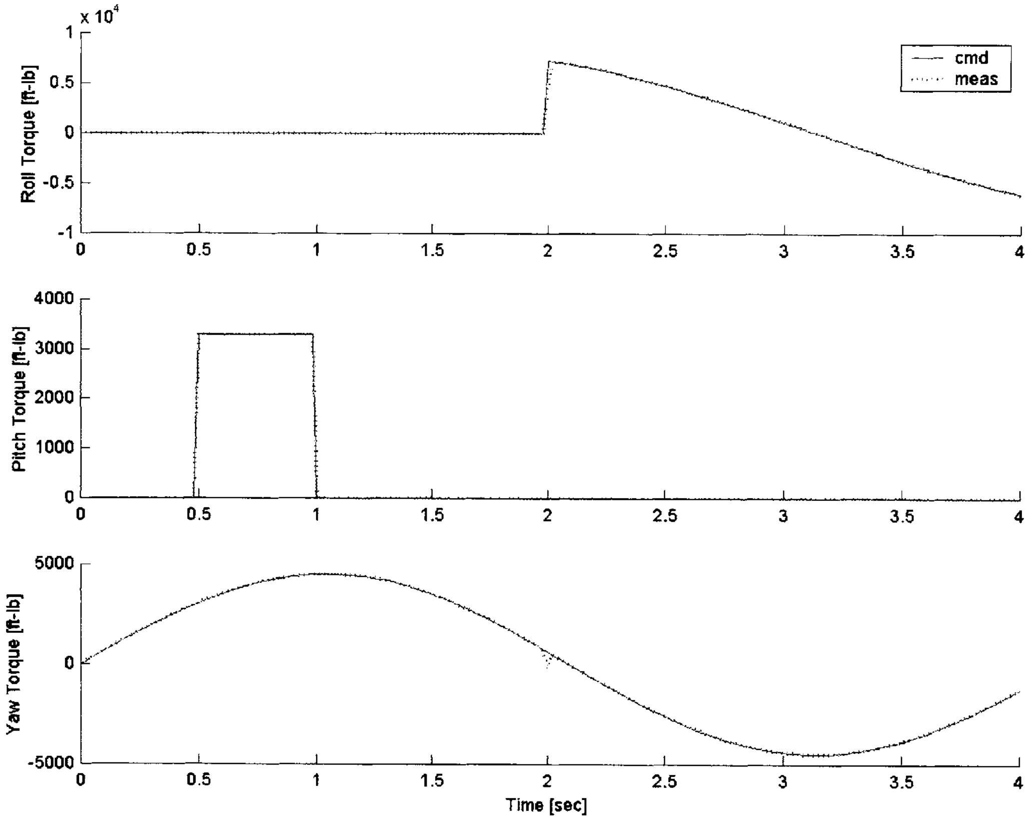


Figure 4-7. Open Loop Test #1: Commanded and Measured Torque

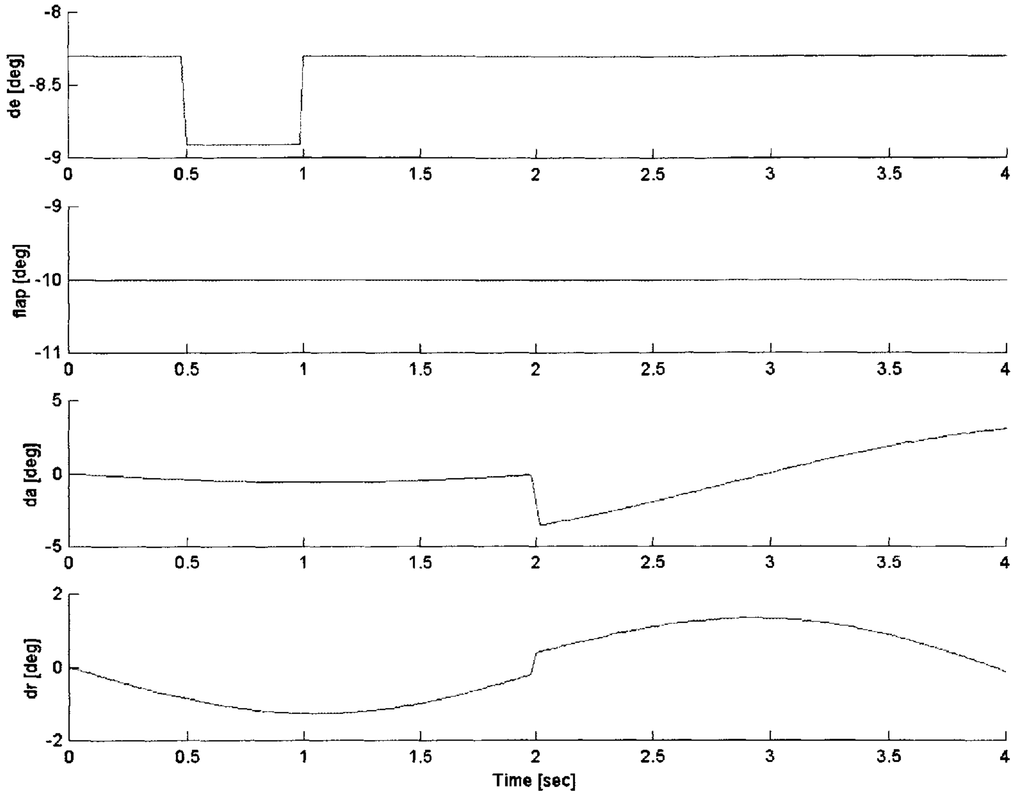


Figure 4-8. Open Loop Test #1: Actuators

This test is presented primarily to show the accuracy of linearized control authority estimates and to exhibit coordinated actuator response. There is virtually no discrepancy between commanded and measured torque; even sharp jumps in the commanded moments are resolved by the aerosurfaces. Of particular note is the coupling in the lateral channel, especially apparent when the step in aileron deflection momentarily disrupts the rudder/measured yaw torque. Because the activity vectors reflect control authority in all three axes, the rudder quickly recovers in subsequent iterations and operates in coordination with the aileron to solve both roll and yaw torque commands. Attention is drawn to the control authority estimates in this test because the objective function is not required to discourage jet firings at this altitude due to greatly diminished RCS control authorities. However, objective penalties are still necessary to encourage the elevons, rather than the body flap, to satisfy pitch commands.

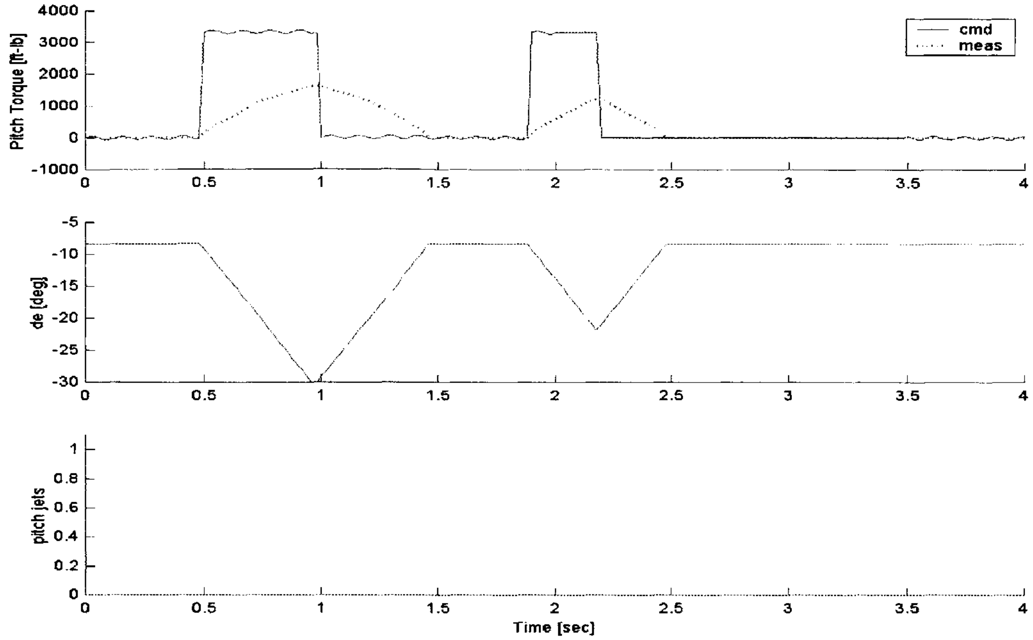


Figure 4-9. Open Loop Test #2: No Objective Function

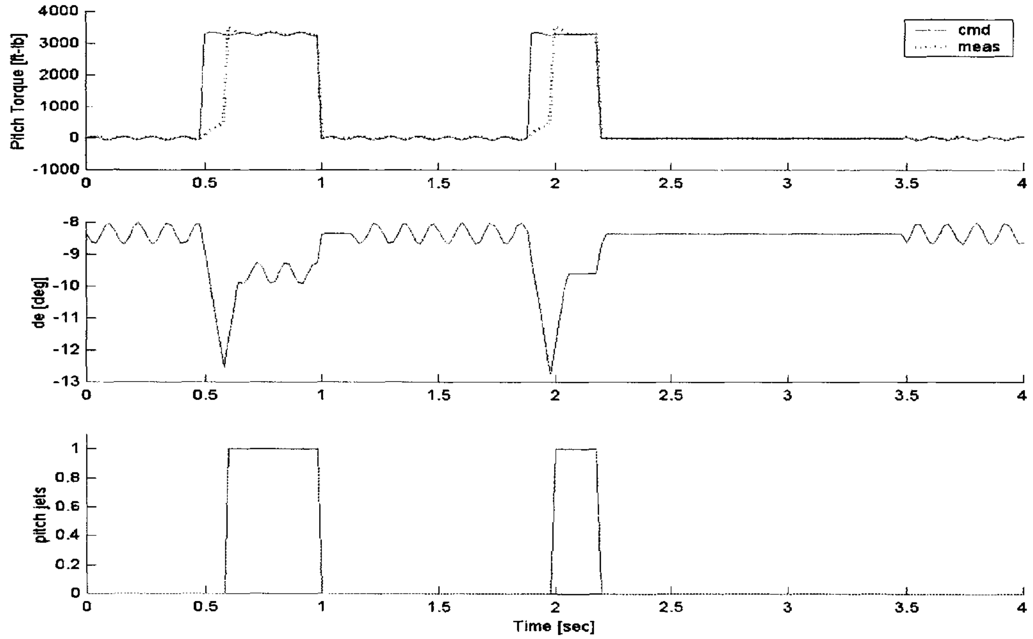


Figure 4-10. Open Loop Test #2: Objective Function Applied

The second test results can be found on the preceding page. This test focuses solely on the longitudinal channel and blending between the pitch jets and the elevons. Figure 4-9 is an example where all objective penalties are equal to zero, thus, there is no discrimination between redundant actuators. The result is disastrous; the elevons modulate at maximum slew rate to their hard stops but still fail to produce the desired pitching moment. Note that the initial jumps in commanded pitch torque are identical for both the first and second tests. This step command was resolved by less than one degree of elevon rotation under the flight conditions of the first test. However, at higher altitudes the aerosurfaces obviously offer insufficient control authority. Also note that the high frequency sinusoidal commands are not resolved by the aerosurfaces. The linear program, with no guidance from an objective function, selects the pitch jet decision variables to solve the sinusoidal portion of the command. Such a low amplitude command obviously initiates no jet firings and the command is unresolved.

A blended RCS and aerosurface response, courtesy of the objective penalties outlined in Section 4.4, is shown in Figure 4-10. Similar to the case with no objective function, the elevons initially try to chase the jump in commanded pitch torque. However, rate saturation penalties are now applied, so the elevons quickly become too expensive for selection and the pitch jets are instead used to provide the desired moment. Similar interaction between the elevons and pitch jets is also evident during the second commanded step in pitch torque. Aerosurfaces are always the choice of the linear program unless rising costs prohibit their selection. The other desirable performance measure exhibited in Figure 4-10 is the distribution of high and low frequency control responsibility to the appropriate actuators. Although ill suited for the high amplitude step command, the elevons are still the best control choices for the low amplitude, high frequency sinusoidal command. This is an example of aerosurfaces operating within the deadband of the RCS jets. Whether the jets are firing or not, the elevons respond to commands that would otherwise be ignored by the RCS. The most important point to absorb from these results is the reliance on the objective function coefficients to produce intelligible actuator commands. Unlike in the first test where aerosurfaces are the only logical actuator choices, control authority estimates are no substitute for a well-formulated objective function when control redundancy is an issue.

The final open loop test concentrates on lateral coupling. The torque commands displayed in Figure 4-11 imply that the vehicle is executing pure yaw maneuvers. Under atmospheric flight conditions these commands would rely on coordinated aileron and rudder motion to produce zero rolling moment and a nonzero yawing moment. However, the environment of the third test is such that the rudder lacks the control authority required to execute yaw maneuvers. Cross-channel coupling instead exists between RCS yaw jets and the aileron. Unlike the second test, where blending was enforced within the pitch axis, this test demands RCS and aerosurface commands across two independently controlled axes. Note that the aileron response is smooth but the measured roll torque is discontinuous. Choppy measurements occur when impulsive burns of the yaw jets produce off-axis moments. In closed loop simulation, off-axis torque and off-axis aerosurface deflections are likely to reflect a somewhat choppy time history.

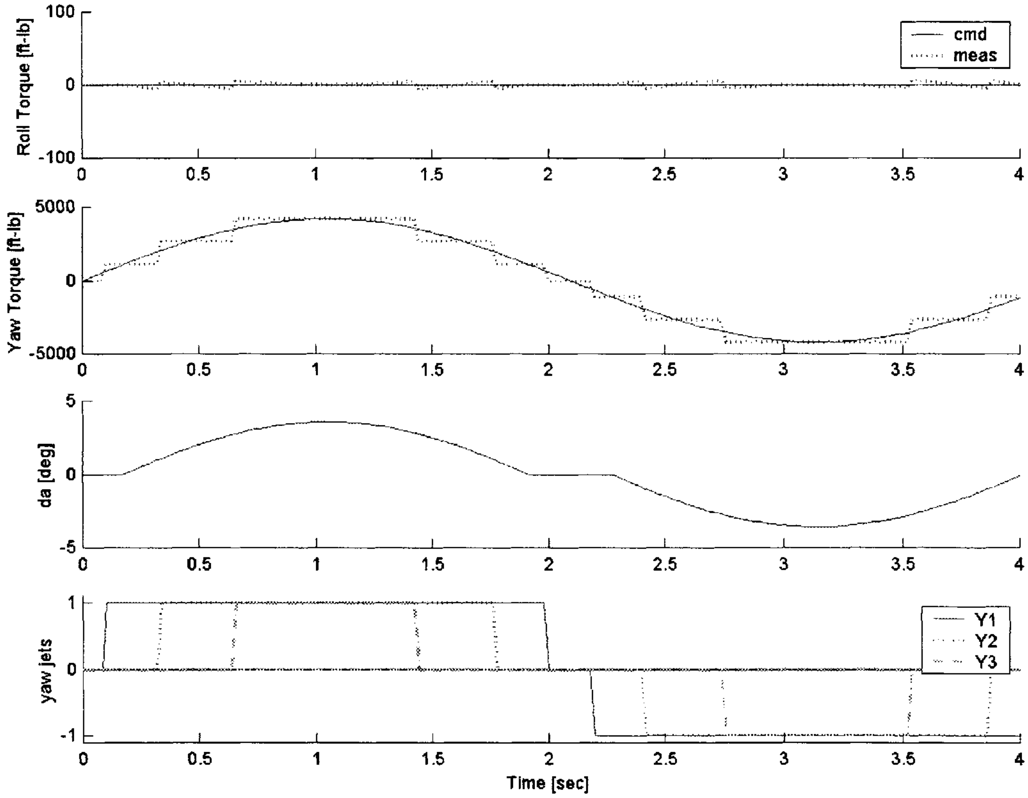


Figure 4-11. Open Loop Test #3

The open loop test results lend credence to the control allocation theory developed in this chapter. They are not a substitute for closed loop simulation, but open loop tests offer an opportunity to examine behavior that is specific to the control allocation algorithm. They also present an effective means of translating the qualitative descriptions of earlier sections into quantitative results. The three tests presented here intend to provide a quick glance at the control allocation response to a broad spectrum of flight conditions. Similar flight conditions and actuator configurations will resurface during closed loop simulation.

5 CLOSED LOOP SIMULATIONS

This chapter demonstrates the capability and characteristics of the control allocation algorithm through a series of closed loop simulations. The chapter is divided into three basic sections. The first two sections concentrate on different portions of entry flight. Mimicking the sequence of open loop tests, the examples in Section 5.1 are low altitude, low mach simulations, while tests in Section 5.2 move to higher altitudes and greater velocities. Section 5.3 investigates actuator failures, presenting two examples that demonstrate the adaptability of a control allocation algorithm. The simulations presented in this chapter accomplish the primary thesis objectives. Results are presented that demonstrate two types of actuator reconfiguration: reconfiguration as the flight environment evolves and reconfiguration in the instance of actuator failures.

Initially, aerosurfaces are trimmed and jets are idle in all simulations. Trajectory cues, guidance and control commands, and actuator commands are issued every 20 milliseconds. Jet costs are set at 200 and aerosurfaces, with the exception of the body flap, initially have zero cost. The body flap, due to its bias cost, is initialized to 10,000 – more expensive than all other longitudinal control effectors.

In all simulations, only one set of values is used to fill the state weighting and control weighting matrices (Section 3.4). When guidance and control performance is the main concern it is customary to apply different weighting matrices as flight progresses. This allows control gains to reflect evolving control priorities. For example, it is excusable to permit crossrange errors at high altitudes, but these errors must be eradicated before the vehicle enters the approach and landing flight phase. Neither guidance and control performance nor the destination of the vehicle is emphasized in this thesis. As such, adopting a single set of weighting matrices and control gains is adequate to demonstrate the capabilities of the control allocation algorithm.

The low altitude and actuator failure simulations (Sections 5.1 and 5.3) utilize the full-state feedback control law of the LQ-Servo. The first simulation results in Section 5.1 consist of six plots, presenting all state and control variable commands and measurements. For succinctness, all other examples in Sections 5.1 and 5.3 show only a selection of pertinent commands and references. Appendix B complements the results of Sections 5.1 and 5.3; it contains all plots that are excluded from these two sections. The high altitude simulations of Section 5.2 do not use the full-state control law and instead configure the LQ-Servo to resemble an α / μ controller. This is accomplished by zeroing control gains of the other state variables, resulting in a controller that is no longer required to balance relative errors between states. The LQ-Servo's only concerns are angle-of-attack and bank-angle errors. This guidance logic is consistent with Shuttle-based guidance algorithms for the initial, high altitude stages of entry [1]. Because the controller exclusively tracks angle-of-attack and bank-angle references in these simulations, aggressive commands are issued in order to showcase the performance of the control allocation algorithm. Plots in Section 5.2 contain all relevant longitudinal and/or lateral commands and references, rendering it unnecessary to publish additional results in Appendix B.

5.1 LOW ALTITUDE ENTRY SIMULATIONS

The first simulation is the initial 100 seconds of a benign entry trajectory. All actuators are available for selection within the control allocation algorithm, but only the standard assortment of elevon, aileron, and rudder is required for this flight. The aerosurfaces display sufficient control authority, even at the initial altitude of approximately 104,000 feet. The second simulation is also an aerosurfaces-only trajectory beginning at the same altitude and velocity as the first. This example is more remarkable than the first because the vehicle is commanded to execute two 60° banks during 300 seconds of flight. Both low altitude simulations are relatively uninteresting when compared to the high altitude examples of the next section, but they still demonstrate the functionality of control allocation. The controller issues torque commands and the control allocation algorithm satisfactorily resolves these commands with aerosurface deflections, as evidenced by stable flight. The first six figures, results from the first simulation, display all state variable, control variable, and actuator references and commands. The last three figures in this sequence summarize the results of the second simulation. These plots show that control allocation, despite considering separate roll and yaw commands, coordinates aileron and rudder during lateral maneuvers.

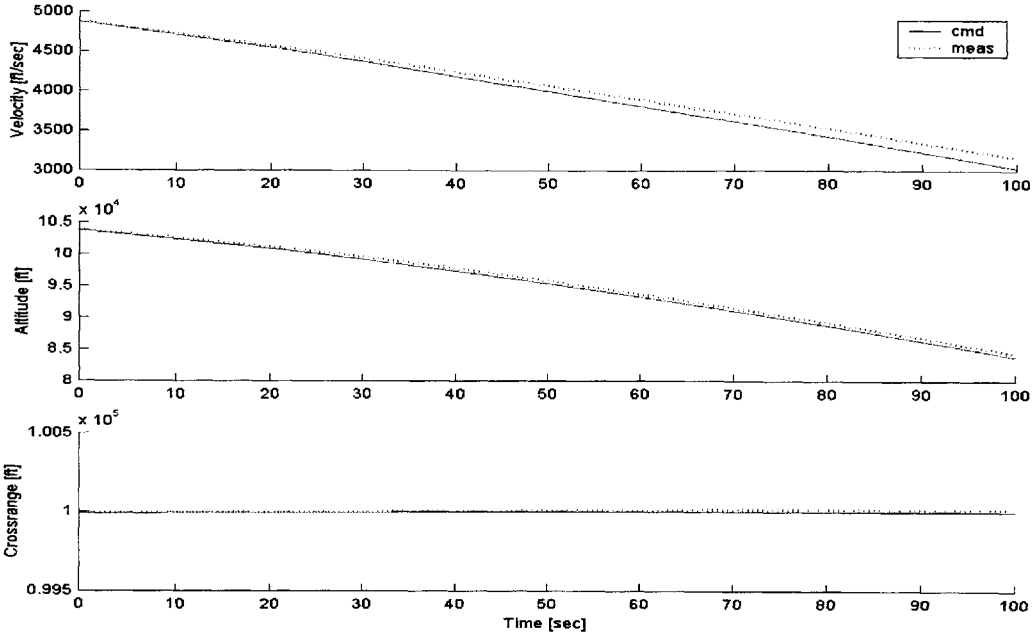


Figure 5-1. Low Altitude Entry Simulation: Nominal Case (1 of 6)

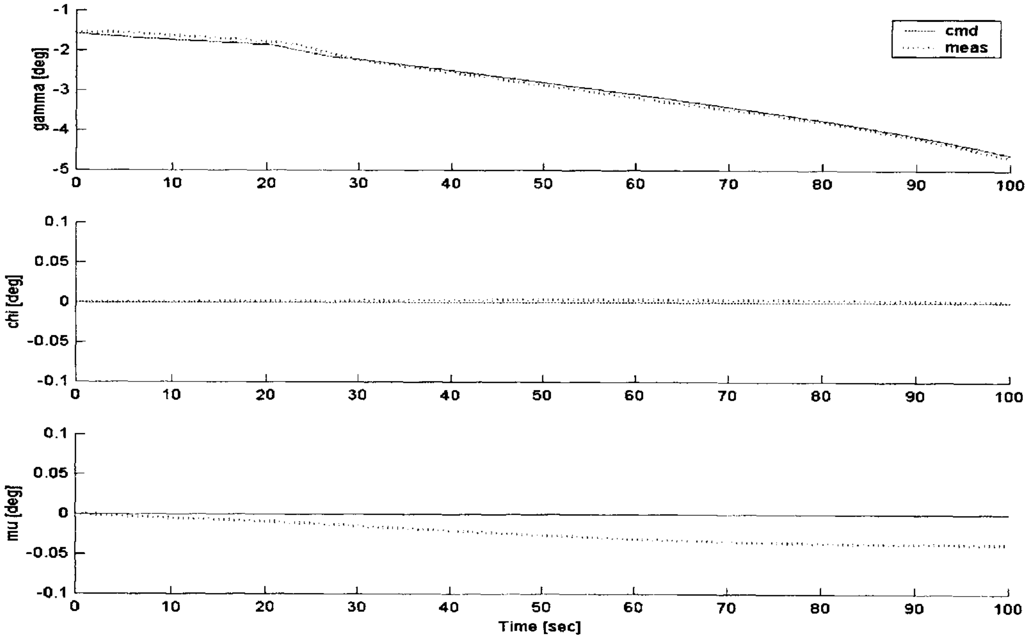


Figure 5-2. Low Altitude Entry Simulation: Nominal Case (2 of 6)

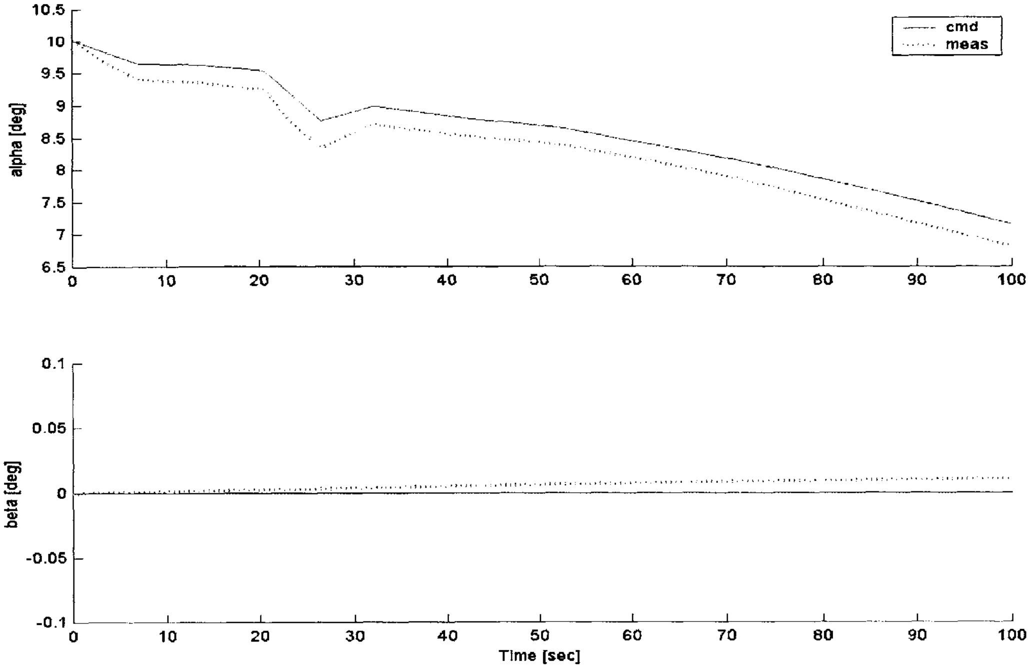


Figure 5-3. Low Altitude Entry Simulation: Nominal Case (3 of 6)

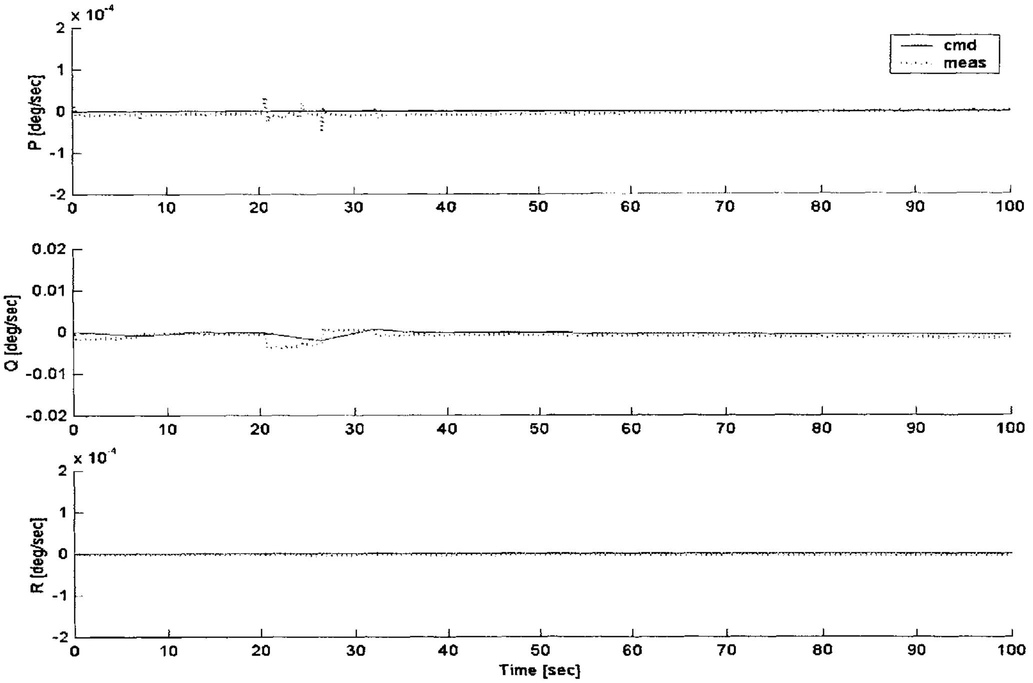


Figure 5-4. Low Altitude Entry Simulation: Nominal Case (4 of 6)

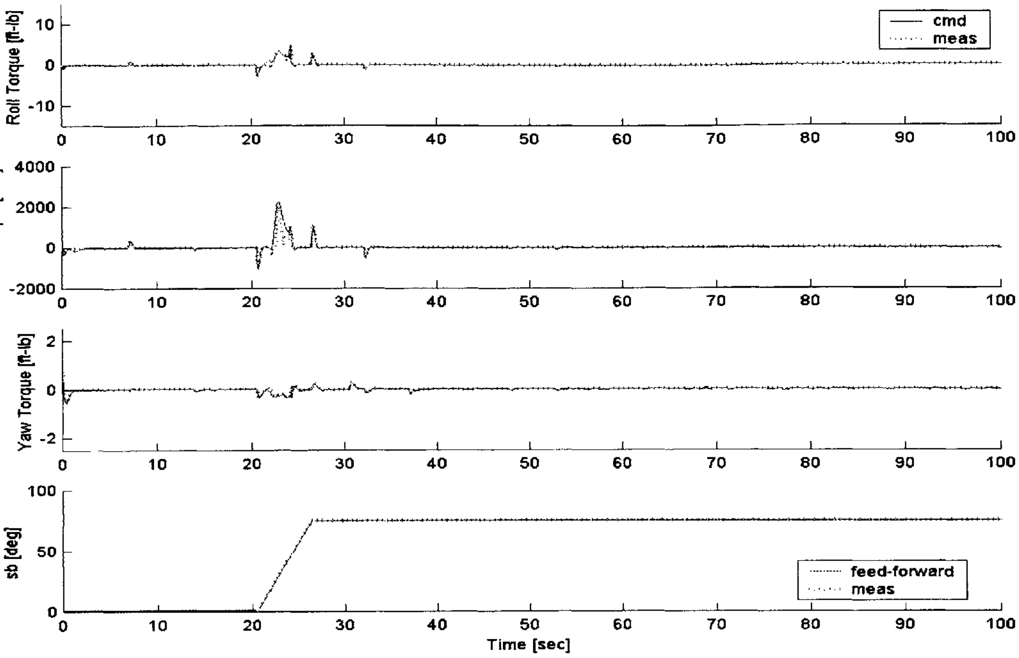


Figure 5-5. Low Altitude Entry Simulation: Nominal Case (5 of 6)

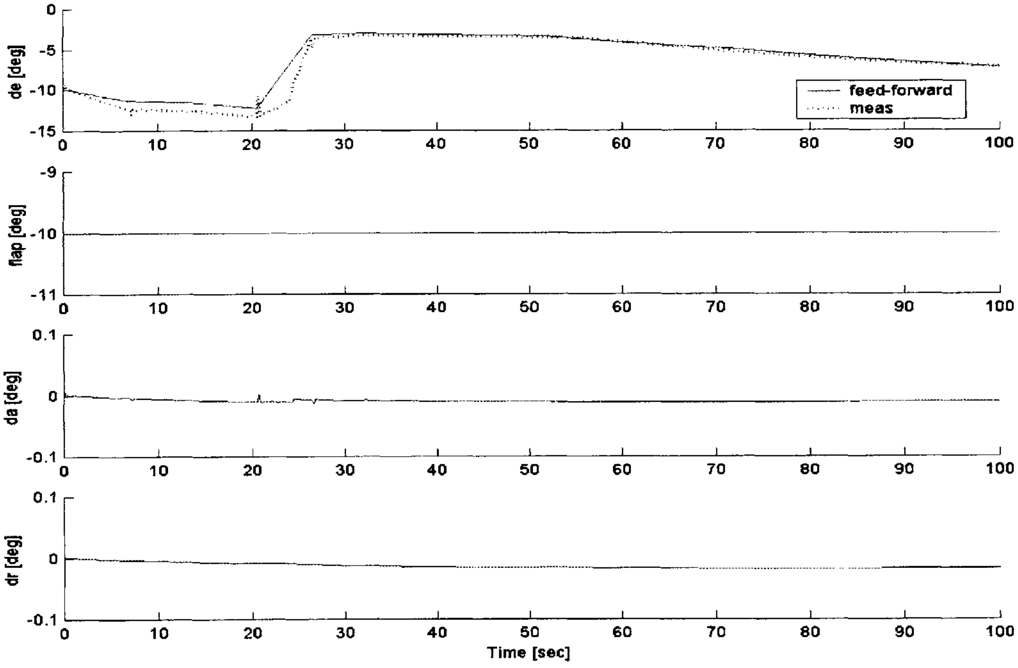


Figure 5-6. Low Altitude Entry Simulation: Nominal Case (6 of 6)

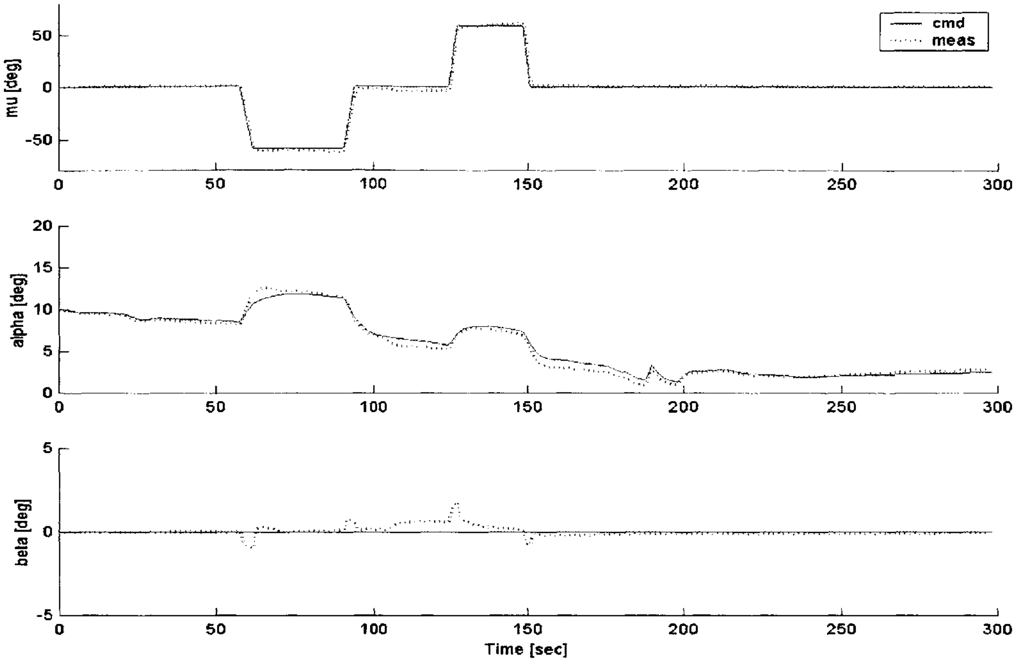


Figure 5-7. Low Altitude Entry Simulation: 60° Bank Turns (1 of 3)

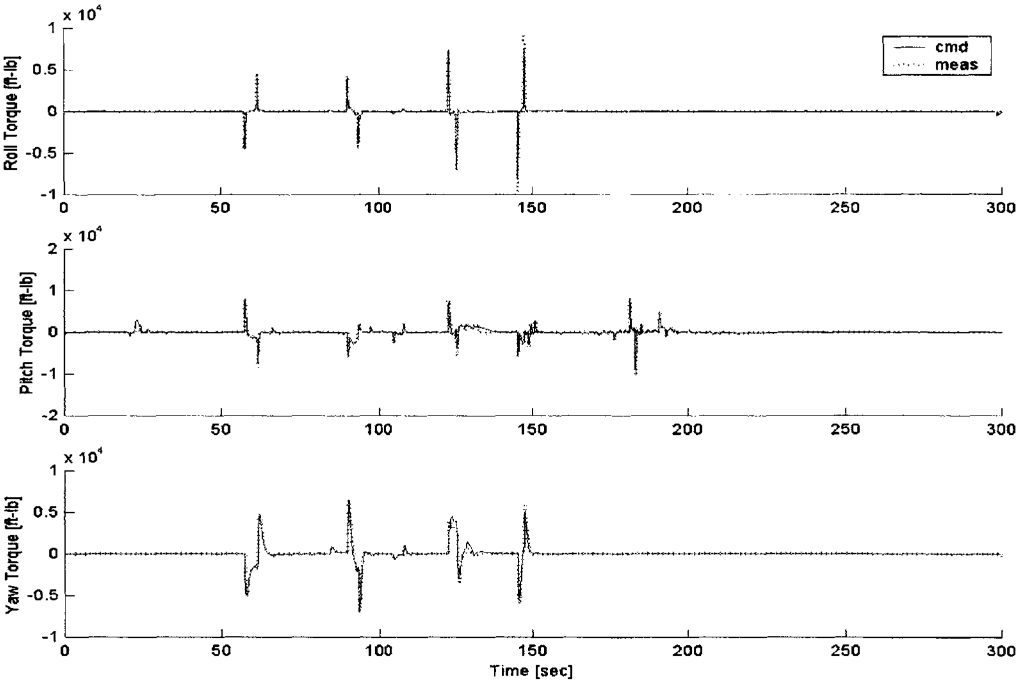


Figure 5-8. Low Altitude Entry Simulation: 60° Bank Turns (2 of 3)

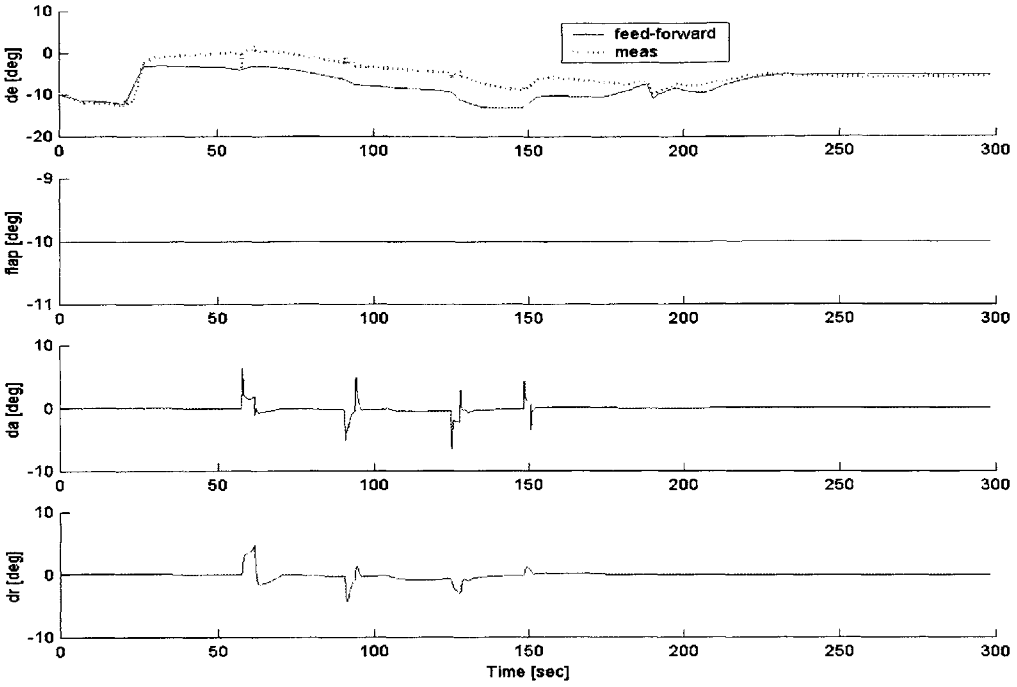


Figure 5-9. Low Altitude Entry Simulation: 60° Bank Turns (3 of 3)

The results from the first two simulations are important because they demonstrate successful mapping of torque commands to aerosurface deflections. Still, they are of limited utility because they do not carry the vehicle through a control effector transition. The next section addresses this flight transition.

5.2 HIGH ALTITUDE ENTRY SIMULATIONS

This section highlights the ability of the control allocation algorithm to dynamically adjust actuator assignment as the vehicle environment evolves. Four examples focus specifically on entry scenarios requiring control redundancy. These simulations, presented in Sections 5.2.1 and 5.2.2, perform attitude maneuvers that demand a mixed response of RCS jets and aerosurfaces. In all instances, the aerosurfaces gain sufficient control authority as the vehicle drops in altitude, eventually resulting in pure aerosurface control. The fifth simulation, found in Section 5.2.3, is a scenario to exhibit control via the RCS jets. In this case the aerosurfaces are purposefully assigned zero control authority and the jets are left to track the state of the vehicle model. Initial and final flight characteristics for the five simulations are listed in Table 5-1.

Table 5-1. High Altitude Simulation Parameters

Sim #	Initial Altitude (ft)	Initial Mach [N/A]	Final Altitude [ft]	Final Mach [N/A]
1-4	200,000	6	157,000	4.3
5	250,000	8	206,000	6.8

The first four examples highlight the control effector transition from exo- to endo-atmospheric flight. These simulations test the implementation of the objective function because actuator families must share control responsibility. The purpose is no longer to coordinate control redundancy in the final simulation. This example simply demonstrates the efficiency with which the control allocation algorithm issues RCS firing commands.

5.2.1 RCS/Aerosurface Simulations: Longitudinal

The following three examples focus on blending between longitudinal actuators; angle-of-attack commands are arbitrarily issued to force action of the pitch jets and elevons. The sequence of commands is identical in all three examples. However, in each simulation the objective function varies, resulting in three different responses.

The first item to note in Figure 5-10 is the control effector transition. RCS jets are necessary to track the initial angle-of-attack commands, but steps of the same magnitude at the end of the flight are solved with only elevon deflections. The other interesting characteristic is the manner in which blended control is carried out by pitch jets and elevons. The initial jumps in angle-of-attack stipulate large torque commands to commence and conclude the maneuver. The elevons attempt to resolve these torque commands, however, successfully accomplishing this maneuver using only elevons would violate slew rate constraints. Recognizing this fact, the control allocation algorithm drives up the cost of the elevons until the pitch jets fire, finally providing the requested moment. The response displayed in Figure 5-10 is adequate, but there is a significant amount of aerosurface chatter during the phase of blended actuator response. Figures 5-11 and 5-12 show the same sequence of commands, but each plot exhibits decreasing levels of elevon chatter.

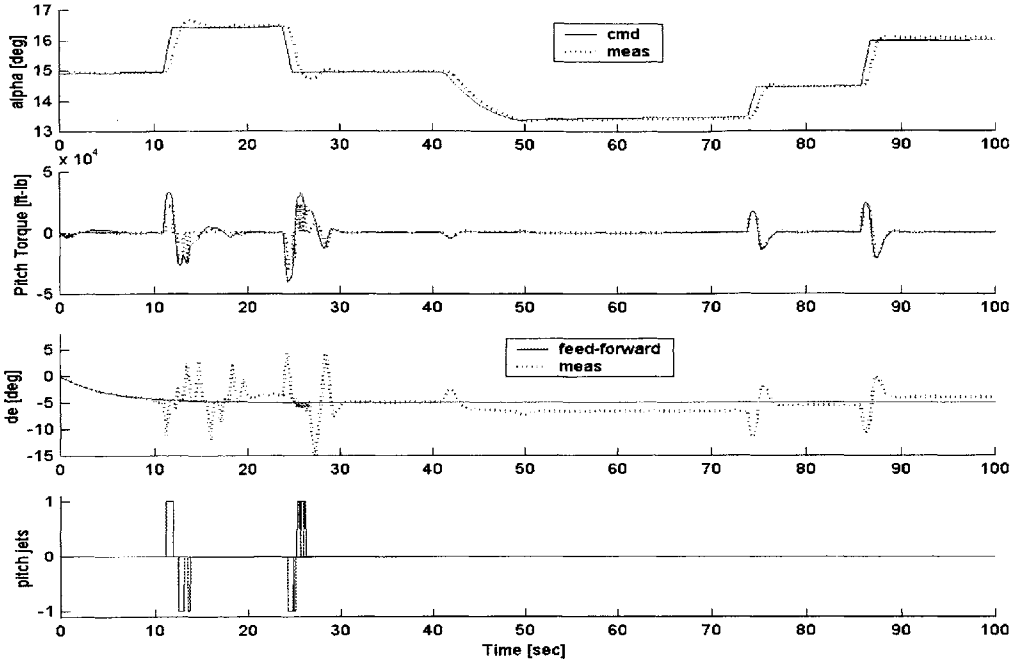


Figure 5-10. High Altitude Entry Simulation: Pitch Maneuver / Low Elevon Cost

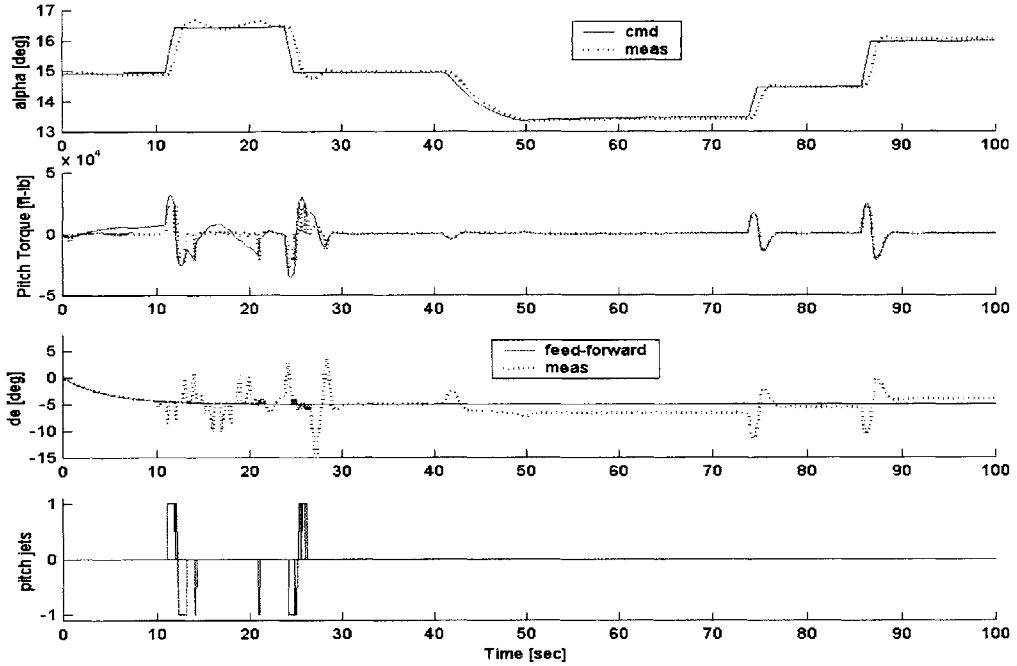


Figure 5-11. High Altitude Entry Simulation: Pitch Maneuver / Medium Elevon Cost

Restricting elevon chatter is accomplished by adjusting the rates of increase and decrease of *tally* (Section 4.4). With each simulation, the elevon is directed to tolerate fewer cycles at the maximum slew rate and its cost becomes prohibitive more quickly. Consequently, more frequent and longer duration jet firings are required to compensate for the lack of elevon activity. This is always the trade throughout the control effector transition. If very little aerosurface chatter is tolerated, the vehicle will inevitably require greater levels of response from the RCS. In all three simulations, the commanded torque issued after 40 seconds is always realized by the elevons. The objective function has little bearing on the latter attitude maneuvers because the elevons satisfactorily solve the torque commands without violating slew rate constraints.

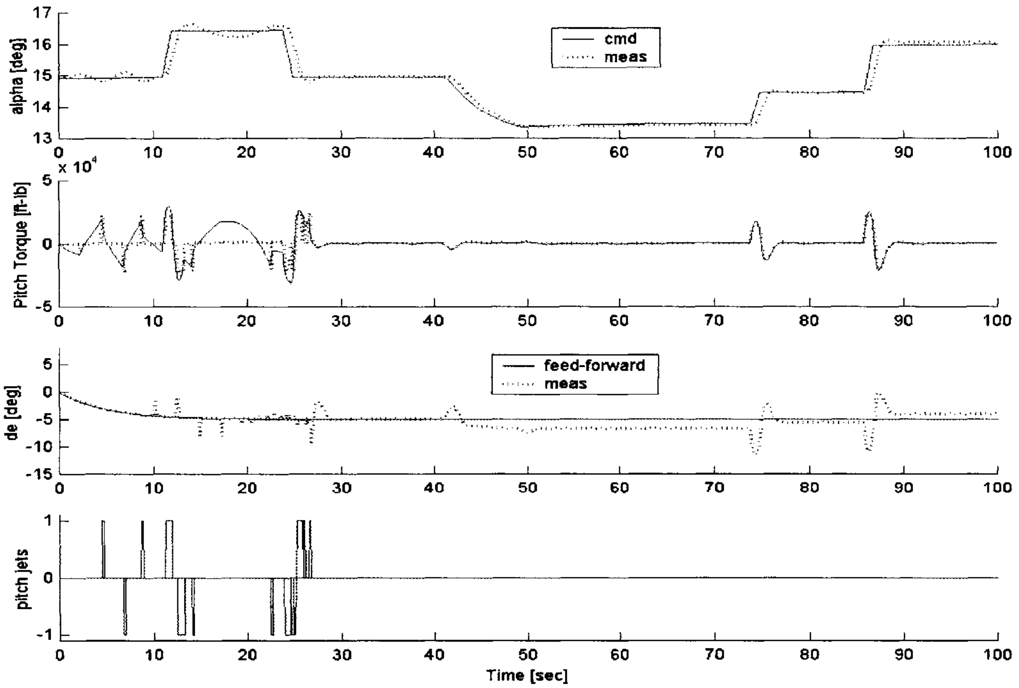


Figure 5-12. High Altitude Entry Simulation: Pitch Maneuver / High Elevon Cost

Results in the three previous figures show reasonable tracking of alpha; however, each example incorporates a different combination of RCS jet firings and aerosurface deflections. The differences in the blended response are attributed to the adjustment of one simple objective coefficient, and, the relationship between the variation in this objective coefficient and the variation in the response is clear and physically intuitive. There is no need to recalculate control gains or to develop a separate control law to specifically deal with the transition from exo- to endo-atmospheric entry.

5.2.2 RCS/Aerosurface Simulations: Lateral

A single bank maneuver requiring coordinated roll and yaw response is executed in this example. Like the simulations in Section 5.2.1, this simulation intends to showcase control redundancy. The difference is that the lateral axes are the subjects of investigation in this example. Note that the yaw channel actuators share control responsibility until approximately 75 seconds. In the roll channel, the ailerons replace the roll jets at approximately 35 seconds. The fact that the two control effector transitions occur during different windows of time permits this example to demonstrate both inter-channel and intra-channel blended response.

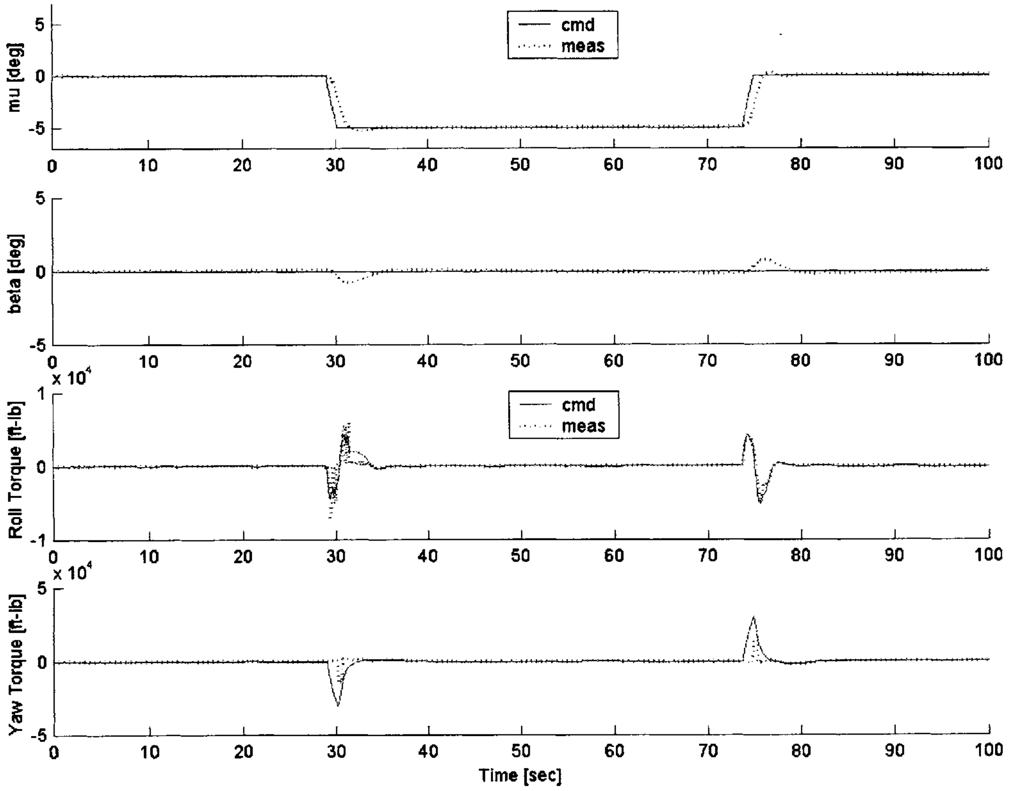


Figure 5-13. High Altitude Entry Simulation: Roll Maneuver (1 of 2)

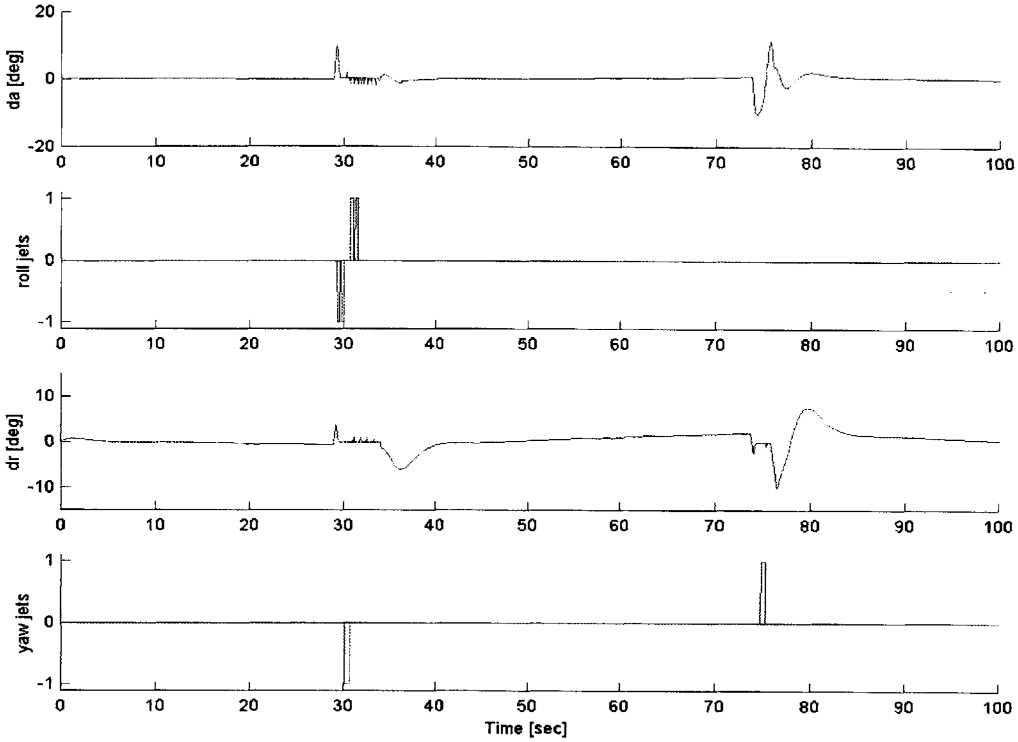


Figure 5-14. High Altitude Entry Simulation: Roll Maneuver (2 of 2)

5.2.3 RCS Only Simulation

At even higher altitudes there is little redundancy between RCS jets and aerosurfaces because the dynamic pressure approaches zero. In the final example of Section 5.2, zeroing the aerosurface activity vector entries ignores what little control authority these actuators offer. The purpose of this test is obviously not to demonstrate blended actuator response; it instead highlights the efficiency and effectiveness of the pulsing logic function. Despite the absence of proper phase-plane logic, the transfer from torque commands to RCS decision variables to jet on/off times produces intelligible jet firings and limits RCS chatter. Of course the jets are incapable of trimming the vehicle when atmospheric disturbances are present, but they are still effective in limiting the vehicle rates and regulating attitude errors. The results in Figures 5-15 through 5-17 serve as a preview to applications of control allocation to on-orbit operations. In fact, linear programming has already been employed in performing real-time optimal jet selections onboard the Shuttle orbiter so this is certainly a feasible extension of control allocation [17].

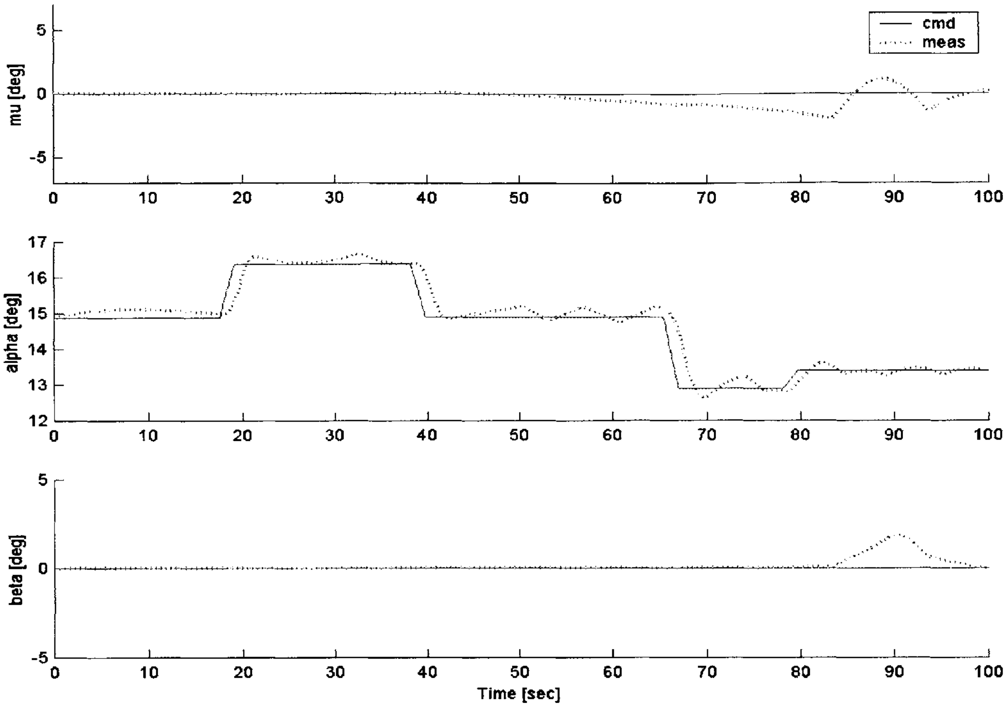


Figure 5-15. High Altitude Entry Simulation: RCS Only (1 of 3)

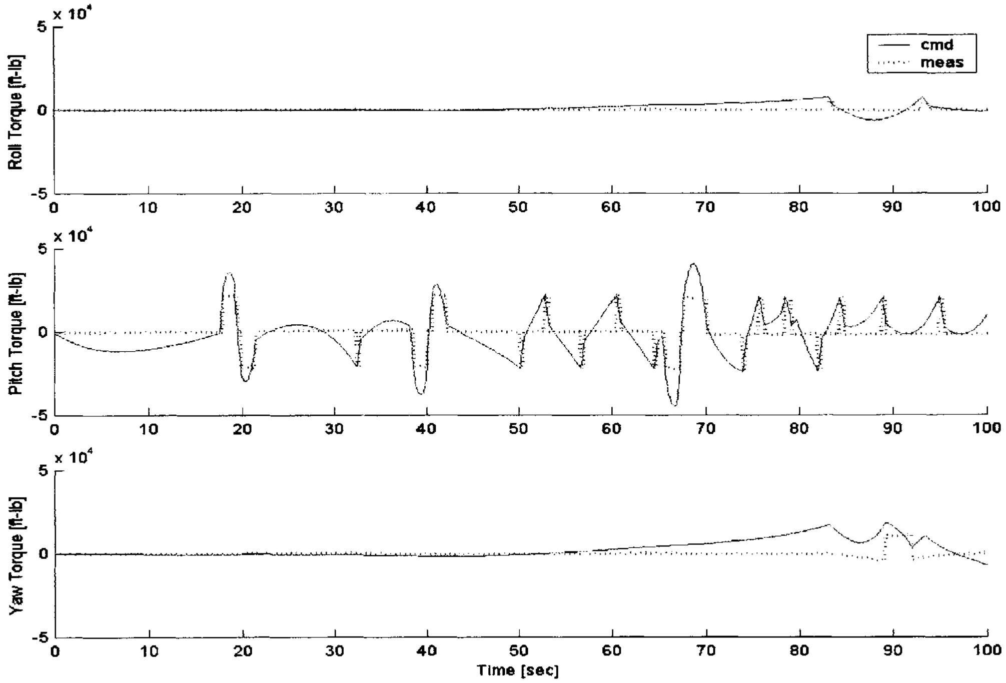


Figure 5-16. High Altitude Entry Simulation: RCS Only (2 of 3)

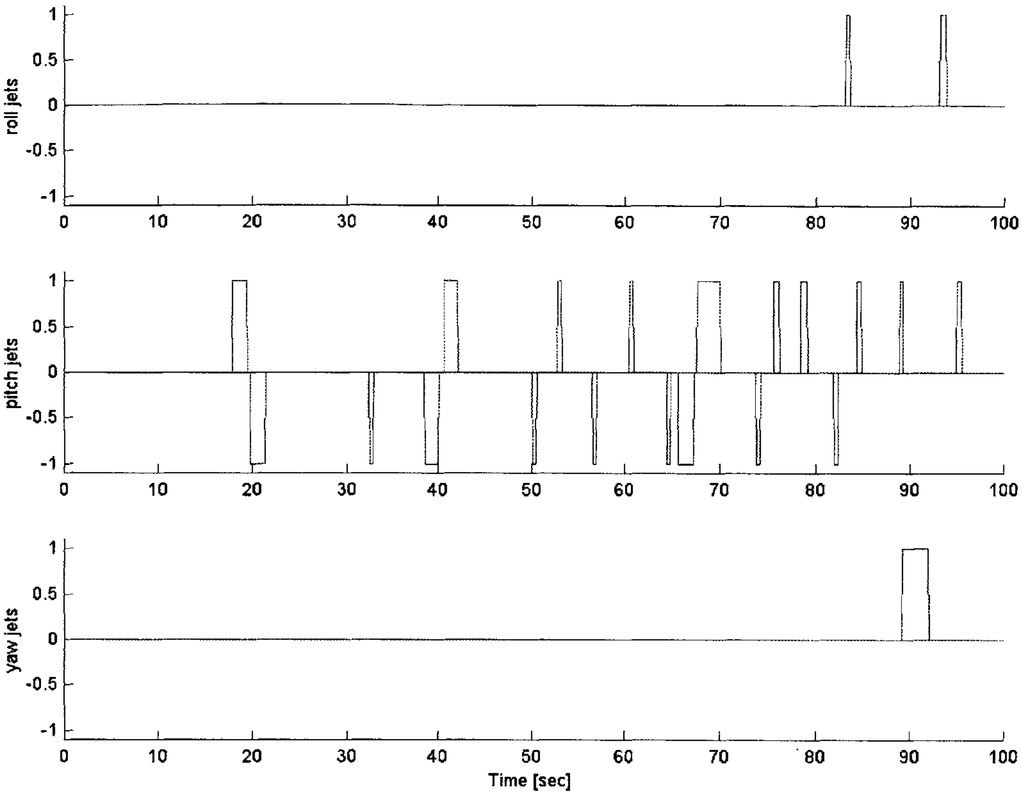


Figure 5-17. High Altitude Entry Simulation: RCS Only (3 of 3)

5.3 ACTUATOR FAILURE SIMULATIONS

The final two simulations in Chapter 5 deal with reconfiguration in the event of actuator failure. In Chapter 2 it was explained that the body flap and elevon are commanded with separate control loops in Shuttle-based control logic. In the traditional approach, the body flap is treated as a pitch trim device and is used to keep the elevons near or on a predetermined schedule [2]. This type of architecture relies on the elevons for pitch control and cannot stabilize the vehicle if the elevons fail. In the proposed architecture, the body flap is always available for control because the control allocation algorithm considers all actuators in a common pool. The body flap is usually discouraged from use, but as shown in Figures 5-18 through 5-20, the body flap can be used to control pitch instabilities if the elevons fail. In this example, the elevon is stuck at its current position when time reaches 20 seconds. After two seconds of lag, during which time it is assumed that vehicle software recognizes the failure, the elevons are removed from selection by zeroing their control authority. The body flap takes on longitudinal control responsibility for the remainder of the simulation.

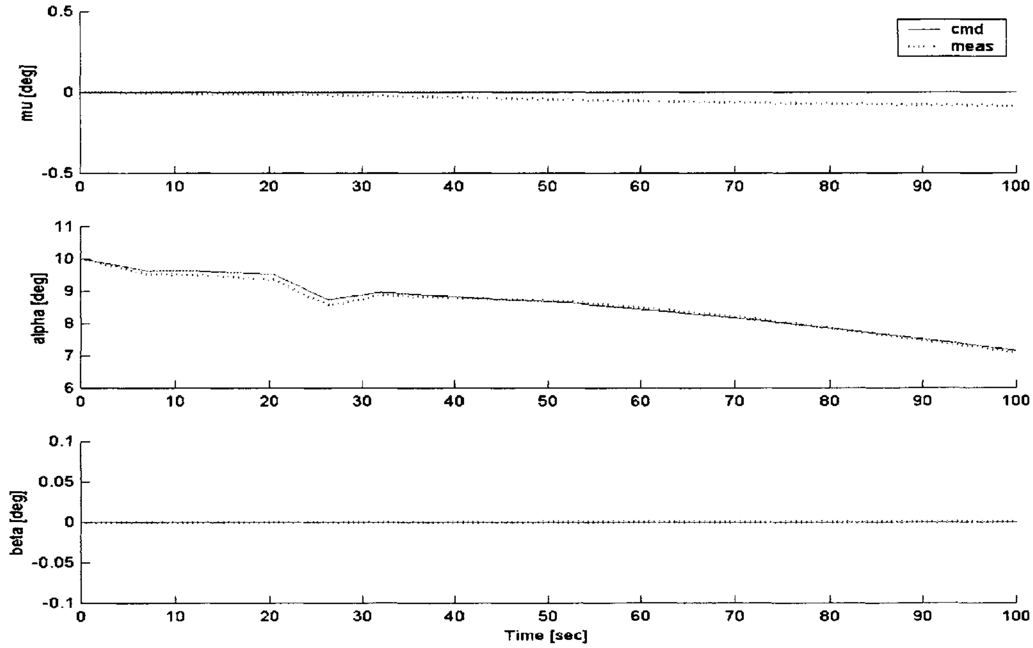


Figure 5-18. Actuator Failure Simulation: Elevon Stuck at t = 20s (1 of 3)

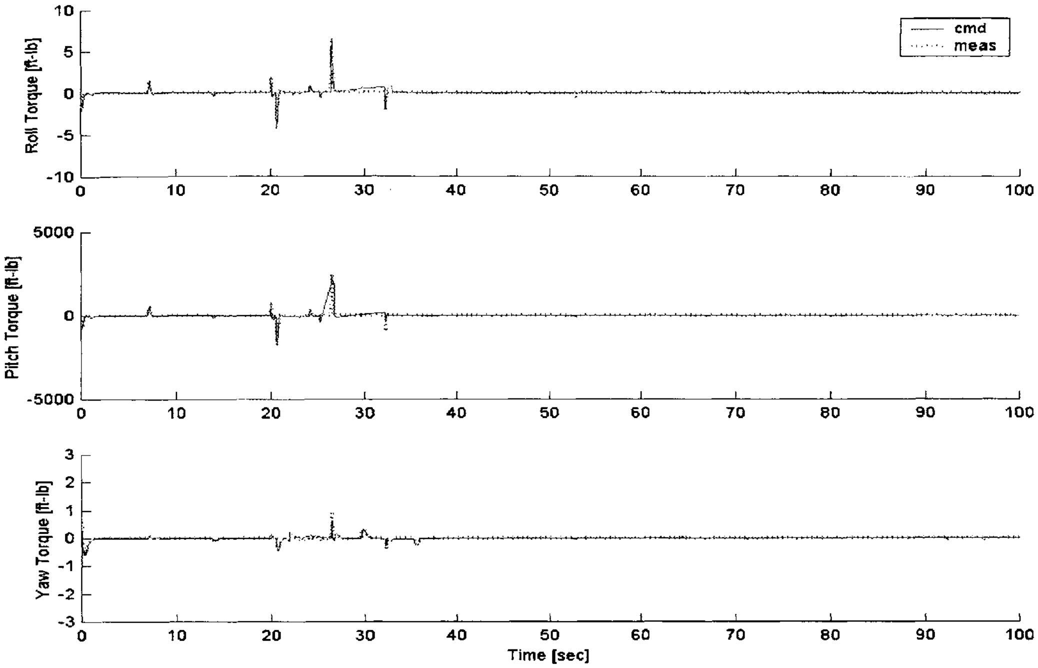


Figure 5-19. Actuator Failure Simulation: Elevon Stuck at t = 20s (2 of 3)

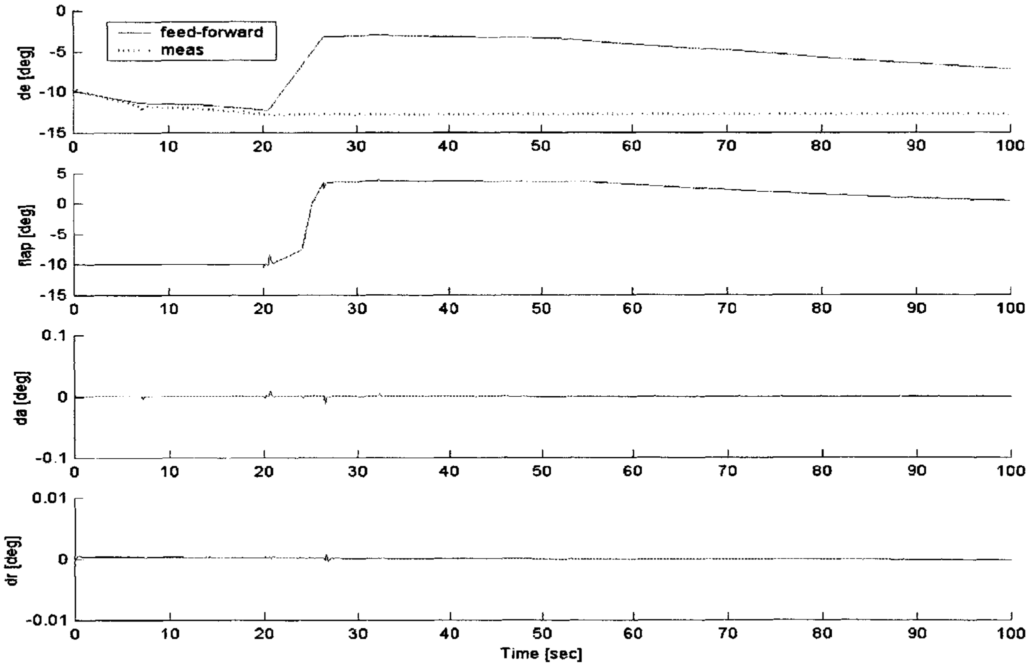


Figure 5-20. Actuator Failure Simulation: Elevon Stuck at t = 20s (3 of 3)

The previous example applies the same trajectory as the first simulation in Section 5.1. The elevon trim schedule changes when time equals 20 seconds because the speed brake opens. In this simulation, the elevons are unable to follow this feed-forward schedule and the body flap must be used to trim the vehicle. The body flap offers less control authority than the elevons, so specific circumstances certainly exist in which the body flap will be unable to compensate for elevon deficiencies. Nevertheless, if an elevon failure is recognized expediently, this simulation proves that the body flap can play a greater role in longitudinal control than it currently serves.

The final simulation, also adopting the trajectory of the first simulation in Section 5.1, assumes that the rudder is unavailable for selection until time reaches 20 seconds. Although no lateral maneuvers are executed, the yaw jets are still required to regulate the sideslip of the vehicle until the rudder assumes that responsibility. Note that the yaw jets, even at the lower altitudes of this test case, still produce enough torque to offer sufficient control authority. Shuttle-based guidance and control algorithms do not consider the RCS at these altitudes, but ignoring such redundancy in circumstances involving aerosurface failure can be potentially disastrous.

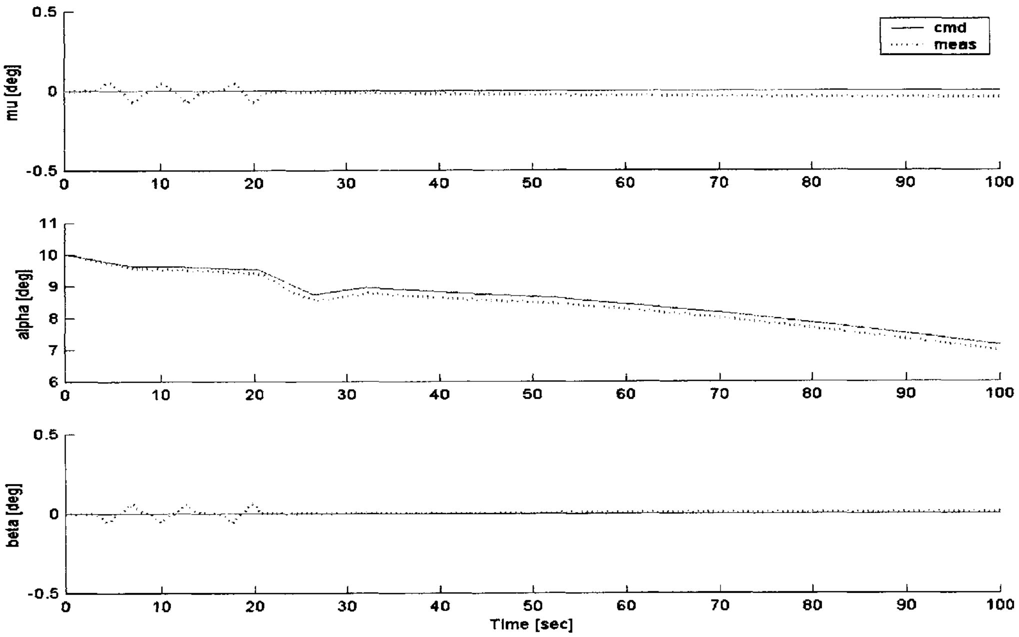


Figure 5-21. Actuator Failure Simulation: No Rudder until t = 20s (1 of 3)

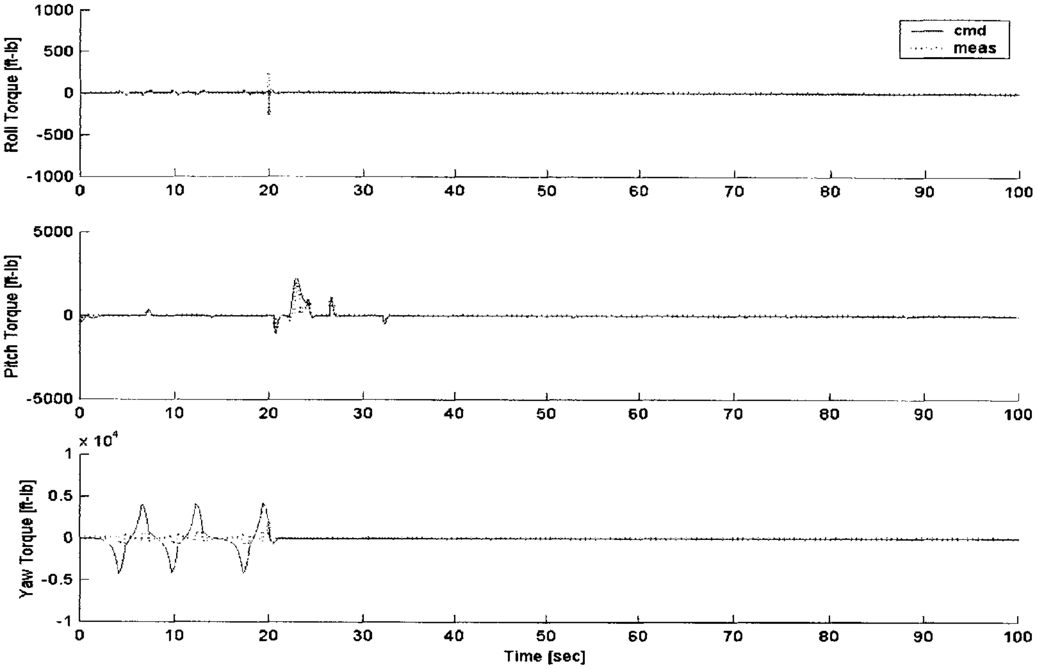


Figure 5-22. Actuator Failure Simulation: No Rudder until t = 20s (2 of 3)

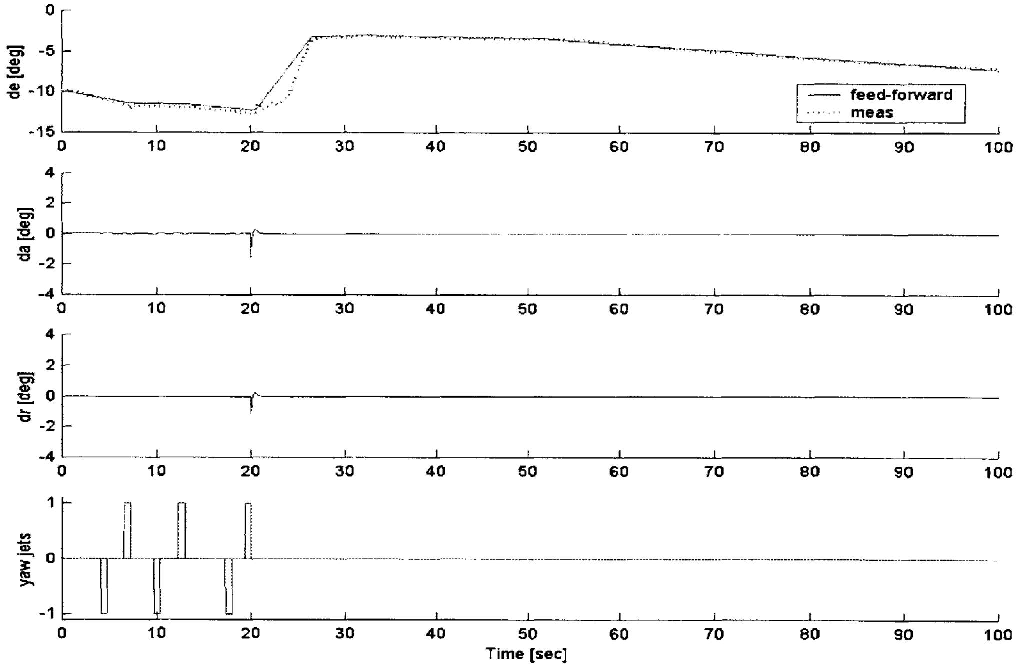


Figure 5-23. Actuator Failure Simulation: No Rudder until t = 20s (3 of 3)

5.4 COMPUTATIONAL EFFICIENCY

In closing Chapter 5, this section briefly addresses the topic of implementing the control allocation algorithm in a real-time system. No definitive conclusions can be drawn from test results because the simulations are conducted in a MATLAB® environment and the code by no means represents flight software. Nonetheless, several platform-independent characteristics of the control allocation algorithm indicate that real-time execution in a 50 Hz system is a reasonable design goal. First, the problem is small. The linear program consists of three equations, one for each independently controlled axis, and thirteen unknowns, eight decision variables for $^+/-$ aerosurface deflections and five decision variables for jet families. Second, the number of iterations the algorithm requires to reach an optimal solution is also small. The data presented in Table 5-2 is summarized from all closed loop simulations presented in this chapter. The numbers do not reflect computation required for each iteration; Appendix A discusses an algorithmic variation that might reduce this burden.

Table 5-2. Control Allocation Algorithm Iterations

Minimum	**Maximum**	**Mean**
3	6	3.6597

It is expected that the minimum number of iterations is three; the algorithm must execute at least this many iterations in order to remove "artificial" decision variables from the solution. Replacing "artificial" decision variables with an initialization heuristic can reduce the number of iterations, further streamlining the algorithm. Even without the benefit of such modifications, the current results show great promise for real-time implementation.

6 ACTIVITY VECTOR UNCERTAINTY

This chapter presents a technique for examining the relationship between systematic errors in measured vehicle state and the actuator commands produced by the control allocation algorithm. The previous open loop and closed loop examples incorporate perfect state feedback; aside from linearization assumptions, no modeling miscues are introduced between the predicted actuator effect and the actual vehicle response. In reality, errors in sensory data corrupt the actuator control authority estimates. This leads to errors when actuator activity is derived from moment commands via the activity vectors. Because estimation errors are inevitable, the control law must be robust to such modeling uncertainties. In order to design a robust control law, it is first necessary to quantify the uncertainty in the actuator control authority estimates. Basic control design techniques are applied to this problem in subsequent sections. The results are criteria that the flight control designer can use in design iterations.

6.1 ANALYSIS METHOD

The true measure of the control allocation algorithm is whether the actuator commands produce the commanded moments. Figure 6-1 follows this open loop path from moment commands to measured moments. For the purposes of analysis, functions along this path must be reduced to simpler elements. Control allocation and the vehicle force and moment model perform only algebraic and trigonometric operations, so they both essentially act as gain blocks.

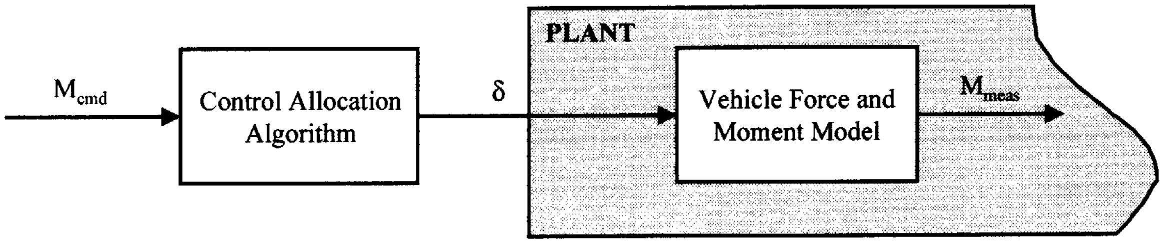


Figure 6-1. Open Loop Path from M_{cmd} to M_{meas} (1 of 3)

Figure 6-2 treats control allocation and the vehicle force and moment model as gains along the same open loop path as Figure 6-1. Equation (6.1) is derived from Figure 6-2:

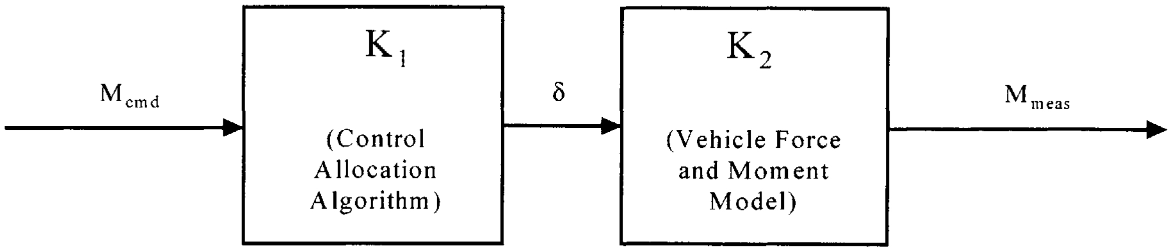

M_{cmd}
K₁
(Control Allocation Algorithm)
δ
K₂
(Vehicle Force and Moment Model)
M_{meas}

Figure 6-2. Open Loop Path from M_{cmd} to M_{meas} (2 of 3)

$$\begin{aligned} M_{meas} &= K_2\delta = K_2K_1M_{cmd} \\ \frac{M_{meas}}{M_{cmd}} &= K_1K_2 = 1 \qquad \text{(ideal)} \end{aligned} \tag{6.1}$$

Under ideal circumstances, the ratio of measured to commanded moments is unity. Because K_2 represents an element of the vehicle truth model, K_1 must equal the inverse of K_2 in order for the measured torque to precisely match the commanded torque. In reality, K_1 is comprised of activity vectors and only closely approximates the vehicle truth model. Any deviation in the ideal value of K_1, caused by modeling errors, is directly reflected in the ratio of measured moment to commanded moment.

$$\frac{M_{meas}}{M_{cmd}} = K_3 = K_1K_2 = \left(\frac{1\,{}^{+}\!/\!{}_{-}\,\Delta}{K_2}\right)K_2 = 1\,{}^{+}\!/\!{}_{-}\,\Delta \qquad \text{(actual)} \tag{6.2}$$

Figure 6-3 represents the path from commanded moments to measured moments in simplest terms.

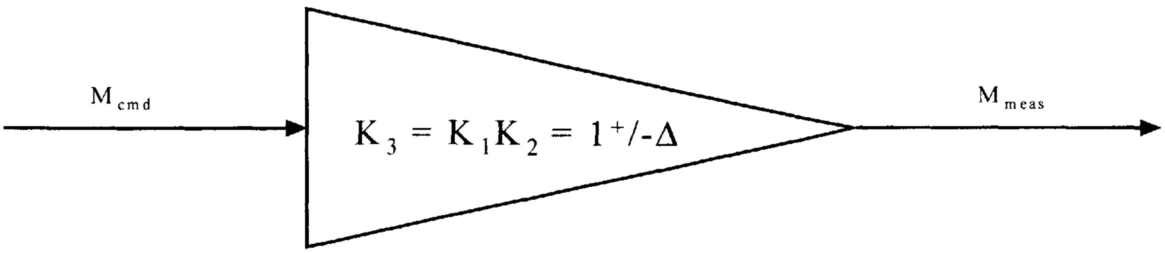


Figure 6-3. Open Loop Path from M_{cmd} to M_{meas} (3 of 3)

The magnitude of the gain uncertainty, Δ, can be expressed in decibels (dB), thereby allowing the control designer to account for control allocation modeling errors. This is accomplished in Equation (6.3). The gain uncertainty is especially important when considering the required gain margin of the closed loop system. When Δ equals zero, K_3 is unity and the gain uncertainty is 0 dB. However, combinations of errors in measured vehicle state and aerodynamic data result in tangible gain uncertainty. The control law must be robust to the worst of these estimates; this is why the gain uncertainty is calculated based on the maximum magnitude of Δ.

$$20 \cdot \log(1 + \max|\Delta|) \tag{6.3}$$

6.2 SAMPLE DATA

This section presents experimental data from a test case where the outlined analysis technique is applied. This is a simple example that intends only to attach some numerical data to the theory of the previous section. Conducting a complete uncertainty analysis requires exhaustive Monte Carlo simulation; this detailed work is beyond the scope of this thesis and remains a topic for future efforts. After presenting the example data, the text returns to this point in listing potential sources of error, all of which must be considered in a comprehensive gain uncertainty analysis.

The test case isolates the control authority estimates relating elevons to pitching moment. No other rotational axes, aerosurfaces, or RCS jets are examined in this example. Additionally, only errors in Mach are considered. All other measured state feedback and aerodynamic data is assumed to be perfect. The example examines a point along a feasible entry trajectory. The altitude, nominal Mach number, and angle-of-attack values used in this example are taken from the "Low Altitude Entry Simulation: Nominal Case" when time equals 50 seconds (Section 5.1).

The first step in this example is to separate modeling uncertainty due to Mach dispersions from any linearization errors. This is accomplished by determining the range of commanded moments where the linearized aerodynamic data produces accurate results. Conveniently, this step also validates the linearization technique used to obtain the aerosurface activity vectors. Using 0% Mach error, moment commands are issued to the control allocation algorithm that require the full range of elevon deflection. Ideally, these elevon commands should produce a one-to-one correspondence between measured and commanded pitching moments. However, Figure 6-4 shows that K_3 is unity for only approximately 20% of the issued moment commands. At first glance this appears to be a very poor linearization, but a closer inspection in Figure 6-5 reveals the contrary.

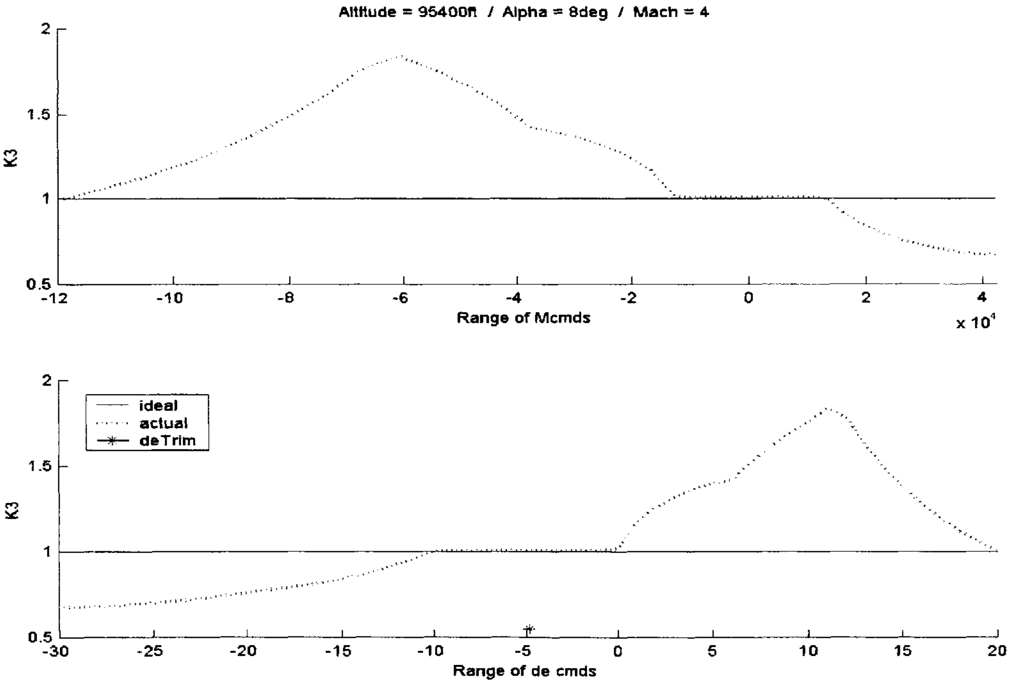


Figure 6-4. Example Data: K_3 vs. Full Range of M_{cmds} / de_{cmds}

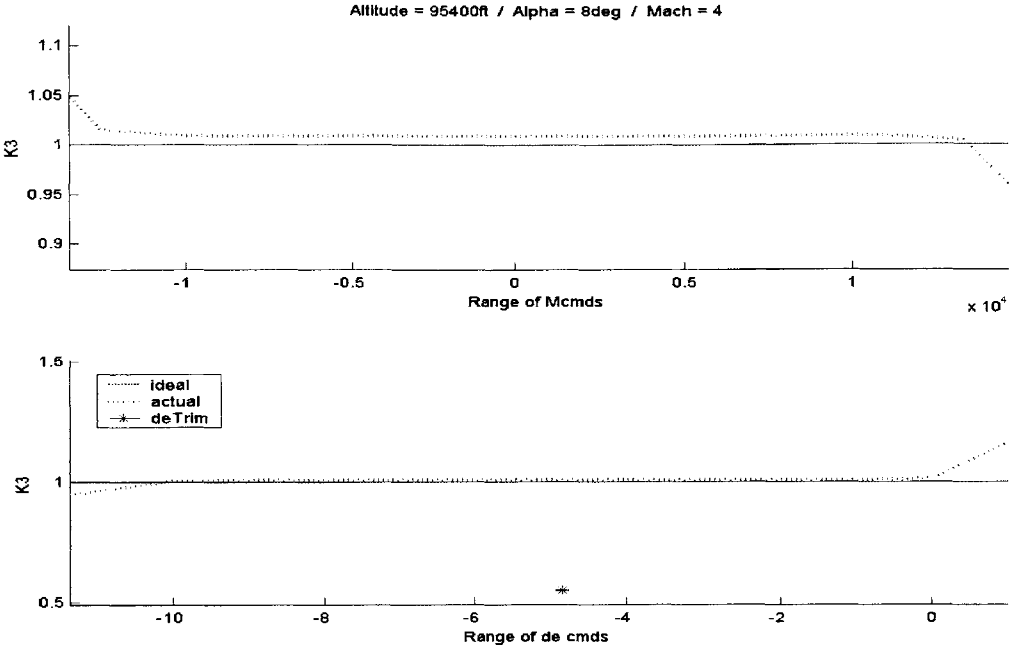


Figure 6-5. Example Data: K_3 vs. Reasonable Range of M_{cmds} / de_{cmds}

The range of moment commands where the linearization is accurate contains 99% of all pitching moment commands issued during the closed loop simulations in Chapter 5. Additionally, no more than $^+/-$ 5° of elevon modulation is required to resolve this range of moment commands. This is not to conclude that moment commands will never lie outside this range, but all simulations report that it is reasonable to assume that a vast majority of moment commands will fall within this region. Figure 6-5 also shows that slight linearization errors are always present; even within the range where the activity vectors accurately reflect actuator control authority.

The second step in this example is to quantify the relationship between Mach uncertainty and the value of K_3. Mach is perturbed by $^+/-$ 4% of its nominal value for the purposes of calculating the activity vectors. Only moment commands within the linear region are issued, and all data pertaining to K_3 is normalized by the 0% Mach error case of Figures 6-4 & 6-5. These measures ensure that the effects of Mach uncertainty are separated from linearization errors.

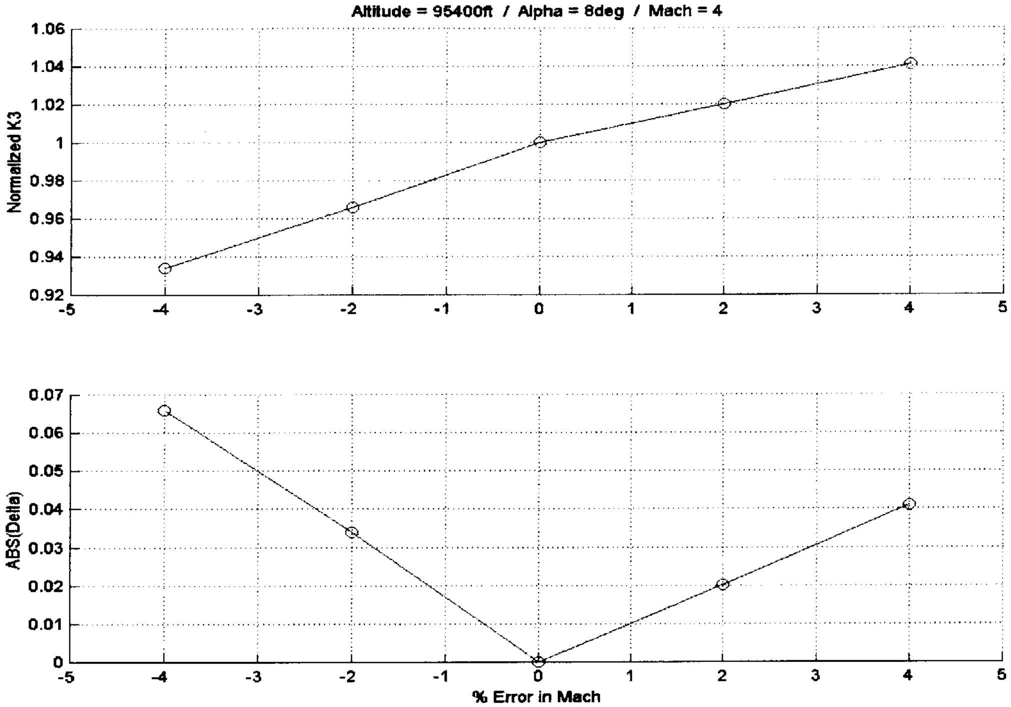


Figure 6-6. Example Data: K_3 and Magnitude of Gain Uncertainty vs. % Error in Mach

Figure 6-6 reveals that the maximum magnitude in gain uncertainty is 6.6%, occurring when Mach error is –4% of its nominal value. Using this worst-case scenario, the following table can be constructed to assist in control design.

Table 6-1. Example Data: Worst-Case Parameter and Gain Uncertainty

Parameter	**Uncertainty**	**Gain**	**Uncertainty**	
Mach Number	$^{+}$/- (4)%	K3	$^{+}$/- (6.6)%	$^{+}$/- (0.55) dB

This example applies exclusively to the gain uncertainty with variation in Mach number. Linearization errors have been removed from the equation through normalization and all other measured state variables and aerodynamic data is assumed to be perfect. If these conditions represent the worst-case scenario, then the control designer must ensure that the closed loop system has sufficient gain margin to account for control allocation variations of $^{+}$/- 0.55 dB. A standard design practice is to permit 6 dB of gain margin, so this case certainly does not test the limits of the closed loop system [14,18].

Conducting a thorough uncertainty analysis involves much more than the example problem reveals. Reasonable combinations of all uncertainties listed in Table 6-2 must be considered in Monte Carlo simulation in order to find the worst-case gain uncertainty.

Table 6-2. Potential Sources of Error in Modeling Activity Vectors

altitude / density
Mach number / velocity
angle-of-attack
dynamic pressure
center of mass position
measured aerosurface deflection
aerodynamic coefficients from data tables
thrust produced by atmospheric jet firing
jet plume expansion in aerodynamic flow
jet plume expansion under finite ambient pressure

Although beyond the scope of this research, conducting a comprehensive analysis offers many benefits. Not only is it important in determining the gain uncertainty, but, as the example proves, it also aids in identifying the limits of the linearization. This information can be used to define a desirable range of moment commands for different phases of flight. Additionally, allowable gain uncertainty defines the frequency at which activity vectors must be calculated. Activity vectors are updated with every control cycle in this research effort, but this an unnecessary precaution. A single set of activity vectors might be adequate for an entire phase of flight, only the allowable gain uncertainty sets the criteria for activity vector accuracy. Lastly, the sensitivity of activity vector calculation to perturbations in aerodynamic parameters is revealed in this analysis. This assists in defining tolerable measurement and estimation errors for each of the parameters listed in Table 6-2.

7 CONCLUSIONS AND RECOMMENDATIONS

This research effort demonstrated that control allocation algorithms show the potential to answer many of the flight control demands of future aerospace vehicles. A control allocation algorithm was successfully applied to the entry problem, producing aerosurface deflections and jet firings in response to rotational commands. Architecturally, incorporating control allocation streamlined the flight control design by permitting command of controlled degrees of freedom rather than commanding a complicated mix of actuators. The proposed architecture separated the tasks of control and control allocation along logical boundaries. This division also precluded the need for actuator gain tables; instead, an objective function was responsible for translating control commands into aerosurface deflections and jet firings.

Several open loop and closed loop simulations were performed to ascertain the features and utility of the control allocation algorithm. Open loop tests were conducted at constant altitude and velocity, while closed loop simulations examined various portions of entry flight. Highlighted among the simulations were test cases requiring a blended response from jets and aerosurfaces. Results from these examples resolved one of the thesis objectives: reliable and efficient actuator reconfiguration as the flight environment evolved. These simulations showed that the control allocation algorithm could efficiently combine the effects of jet firings and aerosurface modulation. Even when aerosurfaces were unavailable, the algorithm commanded intelligible firing commands, despite the lack of dedicated phase-plane logic. The prowess of the control allocation algorithm in managing actuator reconfiguration was also illustrated in several failure examples. Various aerosurfaces were failed, but vehicle control was maintained with the use of other aerosurfaces or the introduction of jet firings. These examples resolved another of the thesis objectives: reliable and efficient reconfiguration in cases of actuator failure.

In all simulations, actuator commands were no longer the product of gain schedules; the objective function was instead responsible for determining the relative contributions of redundant actuators. Of particular significance was the rate penalty contribution to the objective function. This factor was solely responsible for the balance between jet firings and aerosurface modulation during blended actuator simulations. The dynamic calculation of the rate penalty prevented both excessive aerosurface chatter and an unnecessary number of jet firings. Cost coefficients offered the designer a physical connection to the control allocation problem that actuator gains did not. Therefore, tuning the objective function for performance desires was a much simpler task.

Although no definitive conclusions can be drawn pertaining to real-time implementation, the problem characteristics and experimental results certainly do not preclude such expectations. The problem is small, the number of iterations performed by the solution algorithm are few, and similar algorithms have been successfully flight-tested onboard the Space Shuttle. All of these factors indicate that real-time execution is a reasonable design goal.

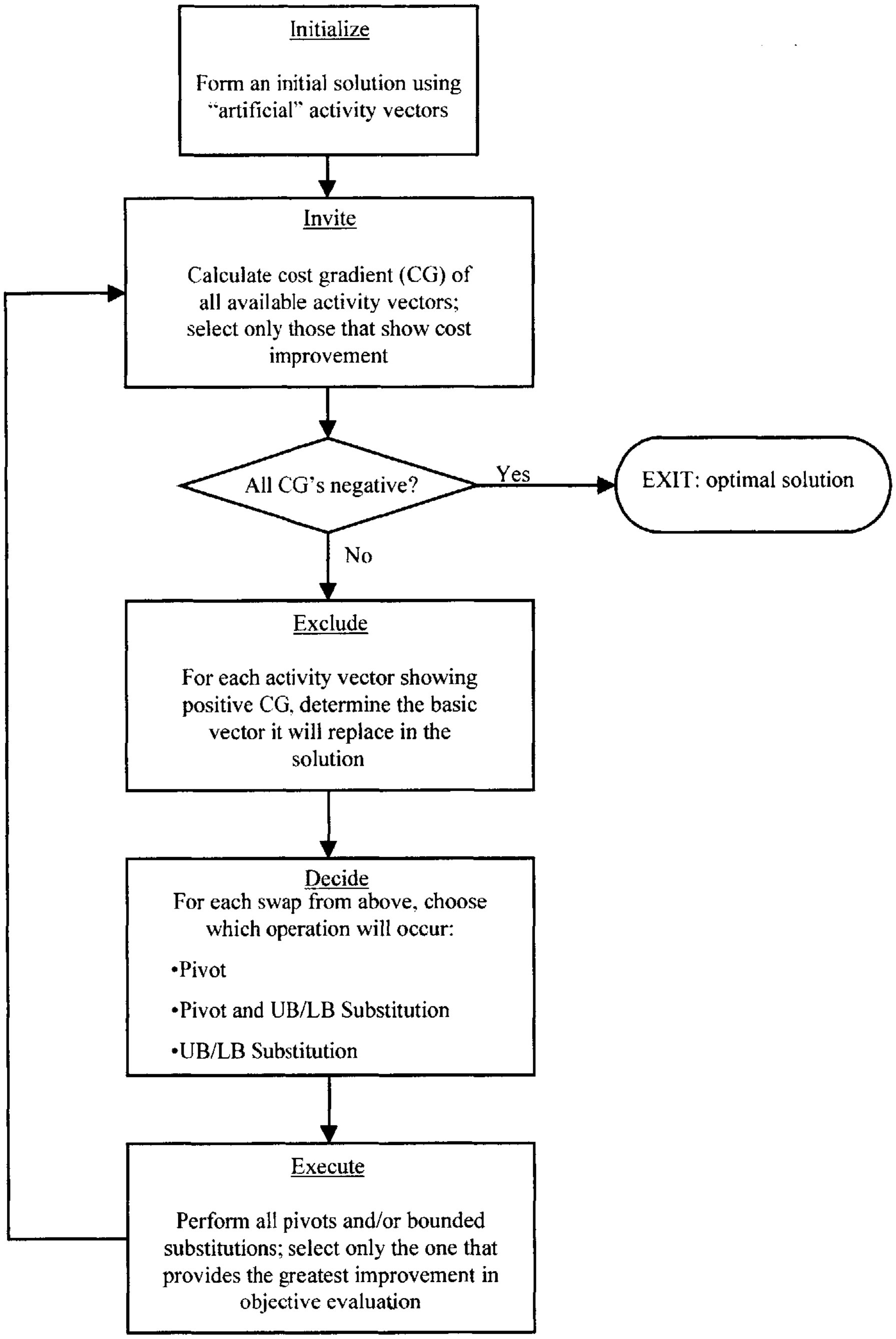
Methods were presented to aid in the study of modeling uncertainty, but additional research should be conducted along this vein in order to clearly establish the relationship between errors in measured vehicle state and errors in actuator control authority estimates. This investigation should also lead to a more judicious approach in calculating aerosurface activity vectors. Actuator control authority estimates do not need to be updated with every control cycle, but only a thorough analysis of estimation limitations will establish criteria for activity vector calculation. Additional measures should also be taken to expedite the calculation of the control authority estimates. For example, aerodynamic force and moment coefficient tables might be replaced with piecewise, linear functions of angle-of-attack and Mach in order to simplify execution of the algorithm. Slight algorithmic modifications or the adoption of a near-optimal search algorithm might also minimize the lag time introduced by a control allocation algorithm.

Further study in the areas of additional actuator types and greater numbers of independently controlled axes is also recommended. A different vehicle model would need to be adopted, but adding gyroscopes, thrust vectoring, or additional aerosurfaces to the selection scheme would certainly test the capabilities of the control allocation algorithm. This would also permit study of flight phases beyond the entry scenario. In particular, launch and on-orbit operations would be of great interest. Adapting such a variety of actuation devices to the existing architecture would be a challenging task.

The algorithm would also be able to accomplish translational control objectives if the number of independently controlled axes is increased from three. Specifically, the speed brake could be included in the selection scheme if translational and rotational control laws are developed. A translational controller could also enforce such goals as direct lift control during banking maneuvers. A slightly different opportunity for research might be to encourage translational objectives via the objective function rather than add independent translational commands.

8 APPENDIX A: THE SOLUTION ALGORITHM

The bounded simplex-based algorithm consists of five distinct sections: Initialize, Invite, Exclude, Decide, and Execute. Figure 8-1 is a flowchart outlining the basic functions of each of these blocks.



Initialize
Form an initial solution using "artificial" activity vectors
Invite
Calculate cost gradient (CG) of all available activity vectors; select only those that show cost improvement
All CG's negative?
Yes
EXIT: optimal solution
No
Exclude
For each activity vector showing positive CG, determine the basic vector it will replace in the solution
Decide
For each swap from above, choose which operation will occur:
•Pivot
•Pivot and UB/LB Substitution
•UB/LB Substitution
Execute
Perform all pivots and/or bounded substitutions; select only the one that provides the greatest improvement in objective evaluation

Figure 8-1. Solution Algorithm Overview

- *Initialize*: As outlined in Section 4.2.2, the algorithm employs an initial basis that is spanned by "artificial" activity vectors. All subsequent iterations work to remove these activity vectors from the basis. If the algorithm converges to a solution that still contains "artificial" variables, the constraints must preclude a physically meaningful solution.

- *Invite*: This loop considers all non-basic activity vectors and determines those that facilitate objective minimization. To accomplish this, a cost gradient (CG) is calculated for each non-basic activity vector/decision variable pair. In this algorithm, the CG is defined as the decrease in objective function, z, with unit increase in a non-basic decision variable, x_j. If all cost gradients are negative or zero then non-basic activity vectors cannot improve the objective function evaluation. The algorithm exits at this point because the current solution must be optimal. Conversely, decision variables with positive cost gradients are beneficial to the solution and further algebraic operations are conducted with these variables and their activity vectors. One exception exists when dealing with aerosurfaces because each aerosurface actuator defines two decision variables and two potentially linearly independent activity vectors; one $x_j / \vec{A}_j$ set is for positive deflection and one for negative. For example, negative aileron deflection might satisfy the yaw torque command while positive aileron is used for the roll command. This might be mathematically correct, but solutions containing simultaneous positive and negative deflection are physically nonsensical. Consequently, an aerosurface decision variable must demonstrate more than just a positive CG before it is considered for inclusion in the solution. If an aerosurface decision variable's complementary pair forms any part of the current solution, either as a basic decision variable or at a bound, then the algorithm is forced to ignore a positive CG a searches for other beneficial non-basic decision variables. This measure ensures that at least one of the decision variables corresponding to an aerosurface equals zero at all times.

- *Exclude*: For each of the non-basic decision variables with positive CG, the basis must be examined for the element most beneficial to exclude. The basic decision variable that either first reaches a bound or first goes to zero as the non-basic, invited decision variable is increased becomes the excluded element. The values of the basic decision variables must change in order to maintain the equality constraints when a non-basic, invited decision variable is increased. Each basic variable that decreases as the invited variable increases is examined; the basic decision variable that is driven to zero most quickly, thereby removing itself from the basis, is identified because it has the minimum pivot ratio [16]. Conversely, the basic decision variable that increases and reaches its bound most quickly is also identified because it has the minimum upper bound ratio [16].

- *Decide*: When increasing the value of an invited decision variable, one of three scenarios emerges:
    1. A basic decision variable is driven to zero, thereby becoming non-basic
    2. A basic decision variable is driven to its bound, thereby becoming non-basic
    3. The invited decision variable is driven to its bound, thereby remaining non-basic

  Only one of these conditions applies to each pair of invited/excluded decision variables; the determination is made based on the minimum pivot and upper bound ratios mentioned in the preceding paragraph. If the upper bound associated with the invited decision variable is smaller than both the minimum pivot ratio and the upper bound ratio, the third condition occurs. In this case, the invited decision variable is brought into the solution as a bounded element but the basic decision variables remain the same. If the minimum upper bound ratio is smaller than both the minimum pivot ratio and the upper bound of the invited decision variable, then condition 2 occurs. In this case, the basic decision variable is removed from the basis but remains in the solution as a bounded element. The invited decision variable and its activity vector replace the excluded element in the basis. Lastly, if the minimum pivot ratio is the smallest of the three quantities, the first condition occurs. The invited decision variable/activity vector becomes part of the basis and the excluded decision variable equals zero and is no longer part of the solution.

- *Execute*: One of the three operations is now defined for each of the invited/excluded decision variable pairs. This portion of the algorithm simply carries out the prescribed operations. A potentially new solution is calculated for each non-basic decision variable with positive CG. Of these potentially new solutions, all of which lower the objective evaluation from previous iterations, only the one resulting in the greatest cost improvement is selected. This operation defines the new basic decision variables and activity vectors. After execution, the algorithm returns to the *Invite* block for further iterations and objective improvement.

Decision variables are required to be non-negative in all sections of the algorithm. This ensures that programming methods that rely on pivot ratios and upper bound ratios are valid [15]. All operations consider the lower bound of a decision variable to be zero, while the upper bound can equal the absolute value of either UB_j or LB_j. A separate array contains the proper sign information of each decision variable. If the array contains a $^+1$, the upper bound is set to $|UB_j|$, and vice versa. As mentioned in Section 4.2, this is not the standard way to deal with unbounded-in-sign decision variables, but it is more efficient because the algorithm does not have to create extra decision variables in order to enforce non-negativity constraints.

Another unconventional characteristic of the solution algorithm is the examination of the basis for every activity vector possessing a positive cost gradient. It is customary to select only the decision variable with the greatest positive CG from the *Invite* block. All subsequent blocks then consider a single invited decision variable. In this framework, the *Invite* block produces a group of potential invitees, each with a positive CG. The other blocks perform hypothetical operations for each potential invitee, but only the procedure from these operations that results in the greatest cost benefit is adopted. The benefit from this approach is that the algorithm is guaranteed to converge at a solution in the minimum number of iterations because decisions are based on actual objective evaluation. The traditional approach determines the invited activity vector based on the potential for the greatest cost improvement rather than actual cost comparisons. The drawback to this algorithm is that it requires a greater number of operations and more memory. For this small application, only thirteen decision variables, no clear advantage is held by either method. However, the adopted solution algorithm loses its appeal for larger problems. The extra computational burden required for each iteration is likely to negate the advantage of reaching a solution in the minimum number of iterations.

9 APPENDIX B: ADDITIONAL SIMULATION RESULTS

The first six plots correspond to the simulation with two 60° bank turns.

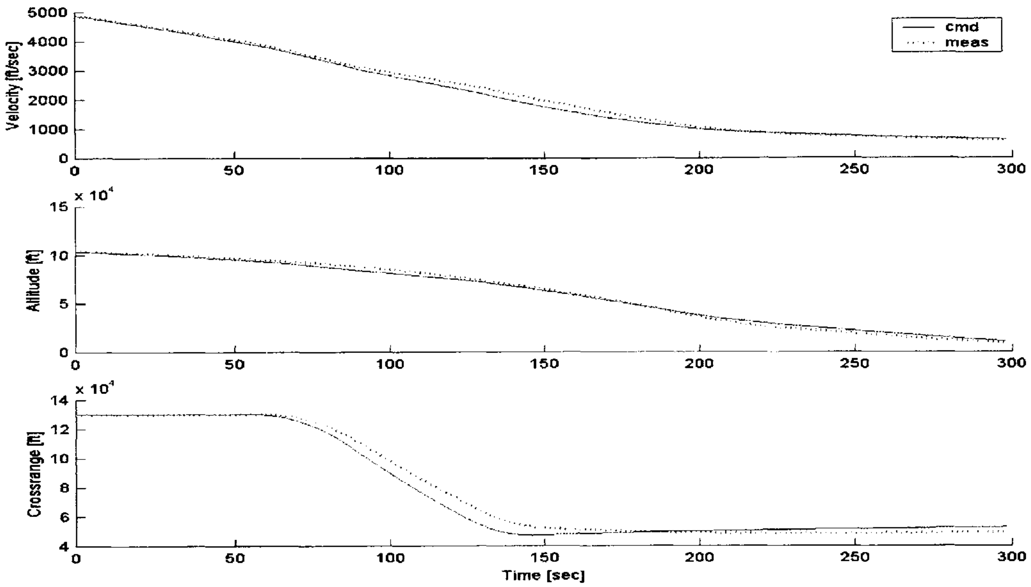


Figure 9-1. Low Altitude Entry Simulation: 60° Bank Turns (1 of 6)

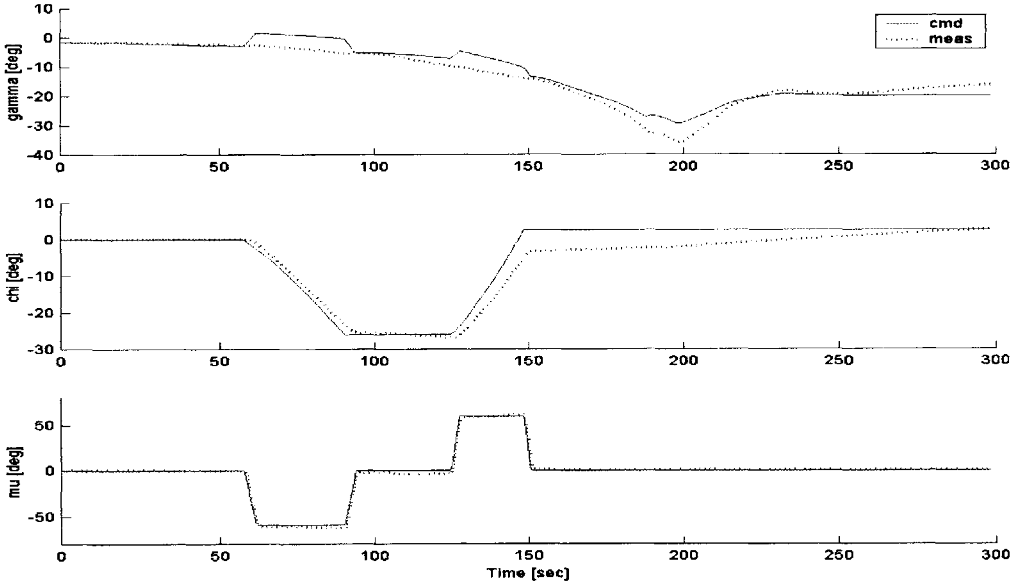


Figure 9-2. Low Altitude Entry Simulation: 60° Bank Turns (2 of 6)

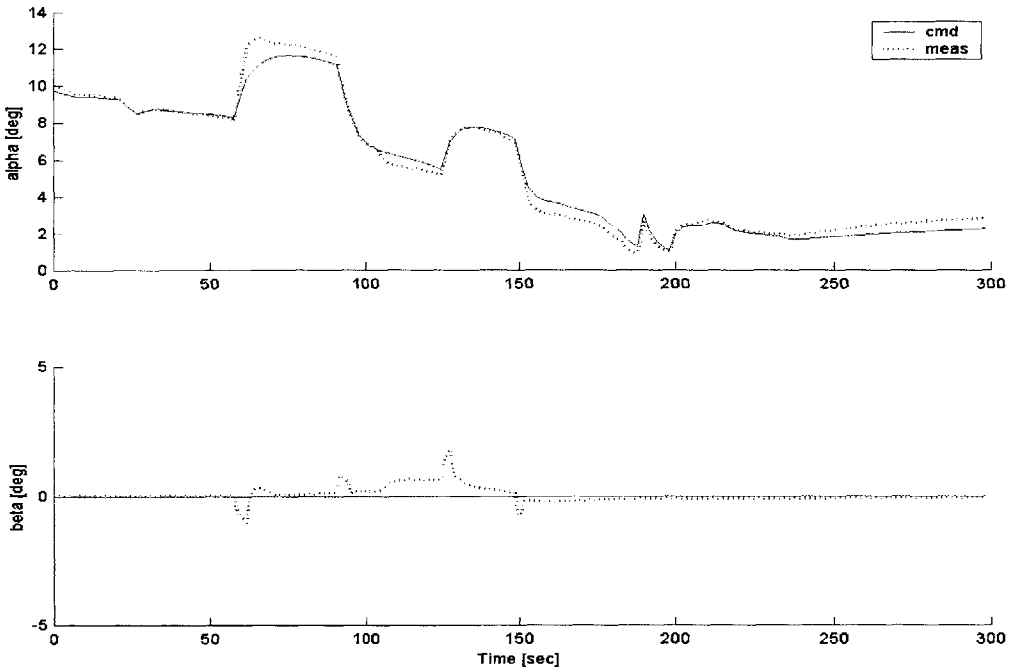


Figure 9-3. Low Altitude Entry Simulation: 60° Bank Turns (3 of 6)

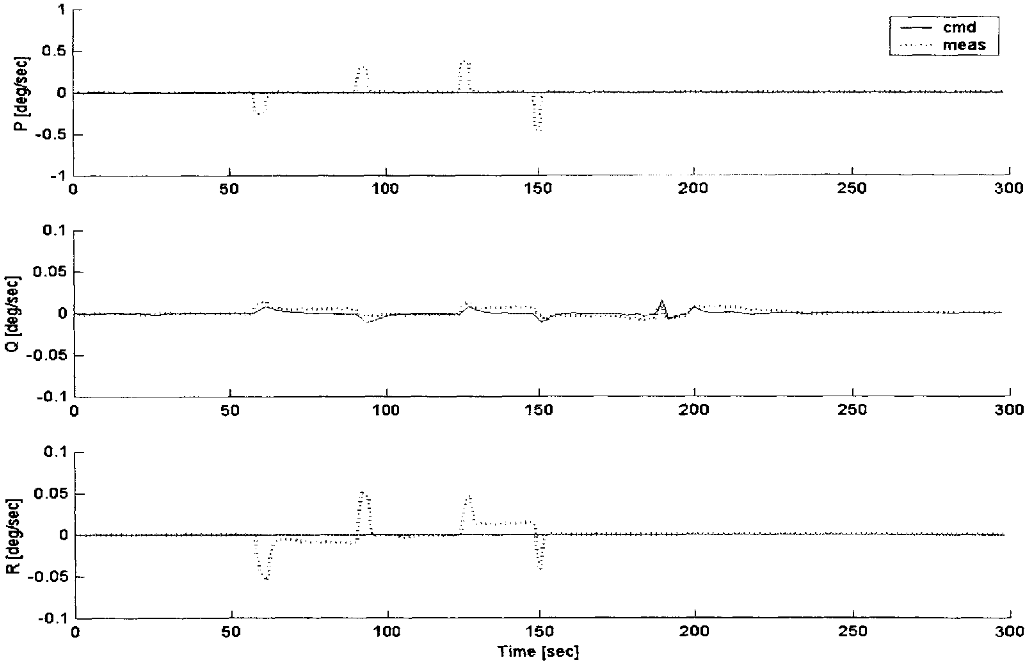


Figure 9-4. Low Altitude Entry Simulation: 60° Bank Turns (4 of 6)

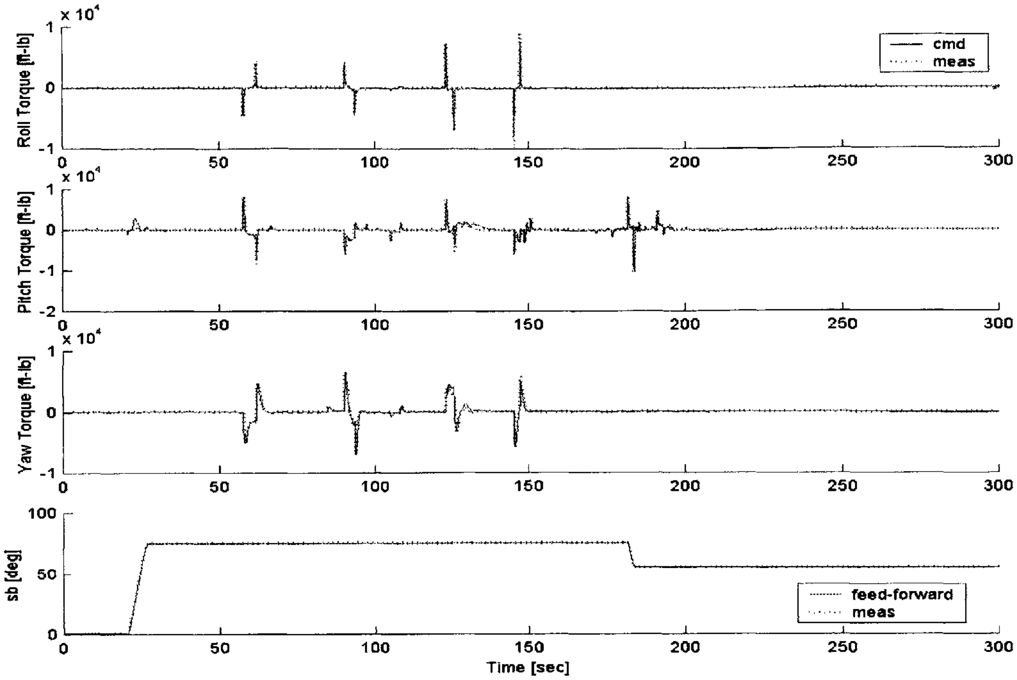


Figure 9-5. Low Altitude Entry Simulation: 60° Bank Turns (5 of 6)

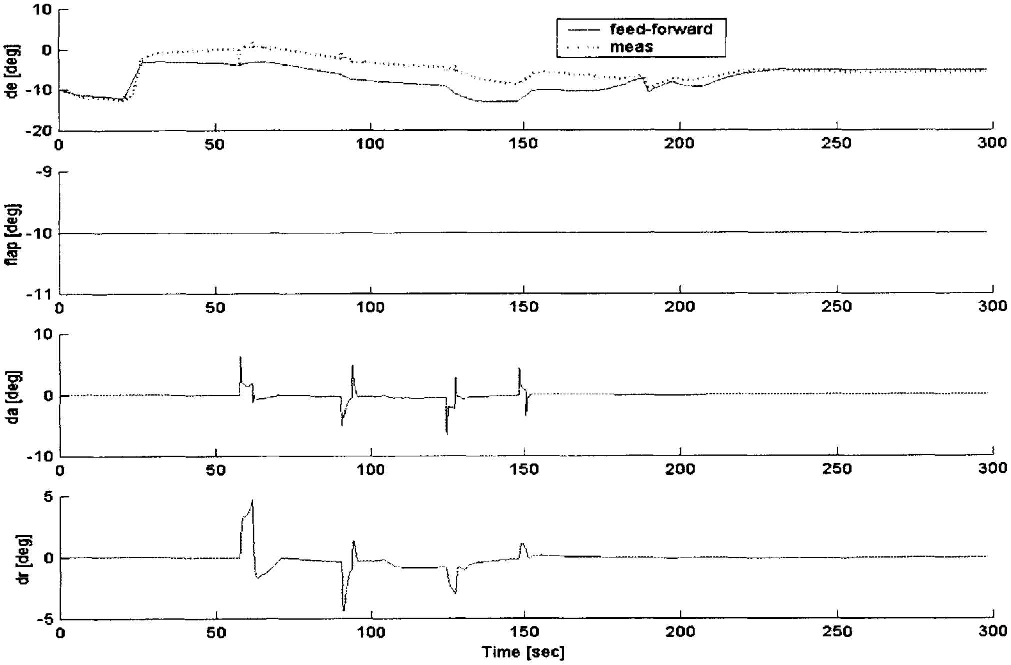


Figure 9-6. Low Altitude Entry Simulation: 60° Bank Turns (6 of 6)

The next six plots correspond to the example where the elevon is stuck at time = 20 seconds.

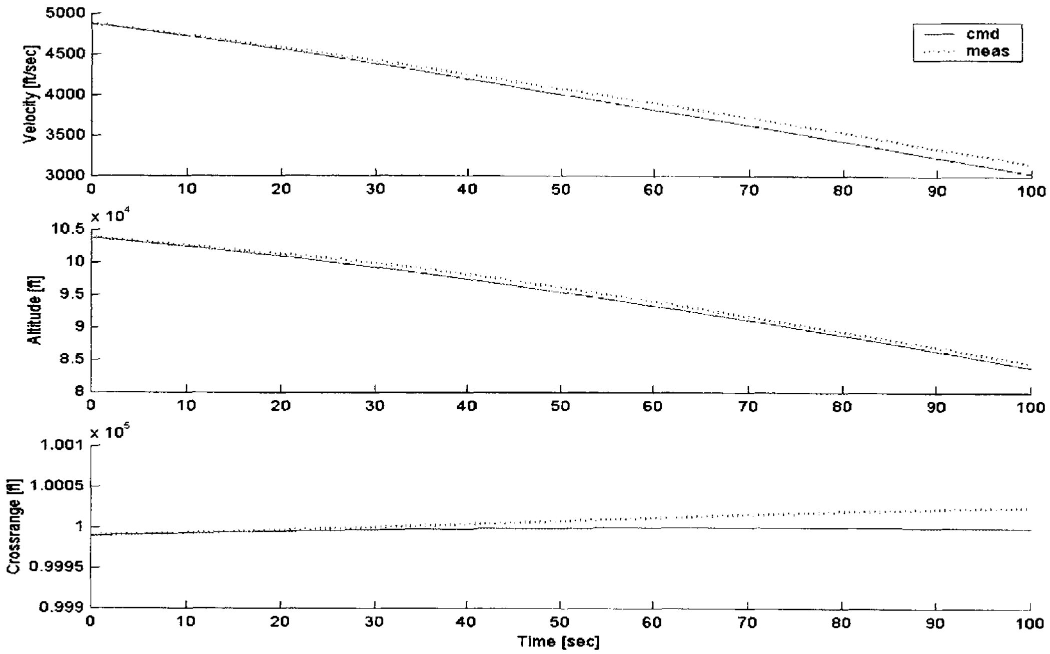


Figure 9-7. Actuator Failure Simulation: Elevon Stuck at t = 20s (1 of 6)

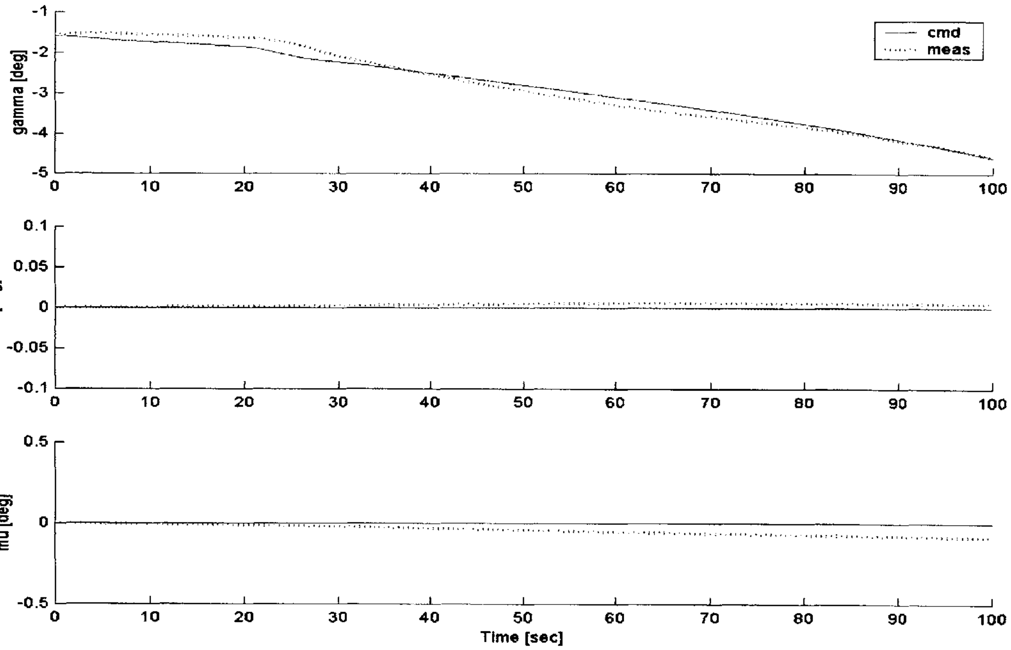


Figure 9-8. Actuator Failure Simulation: Elevon Stuck at t = 20s (2 of 6)

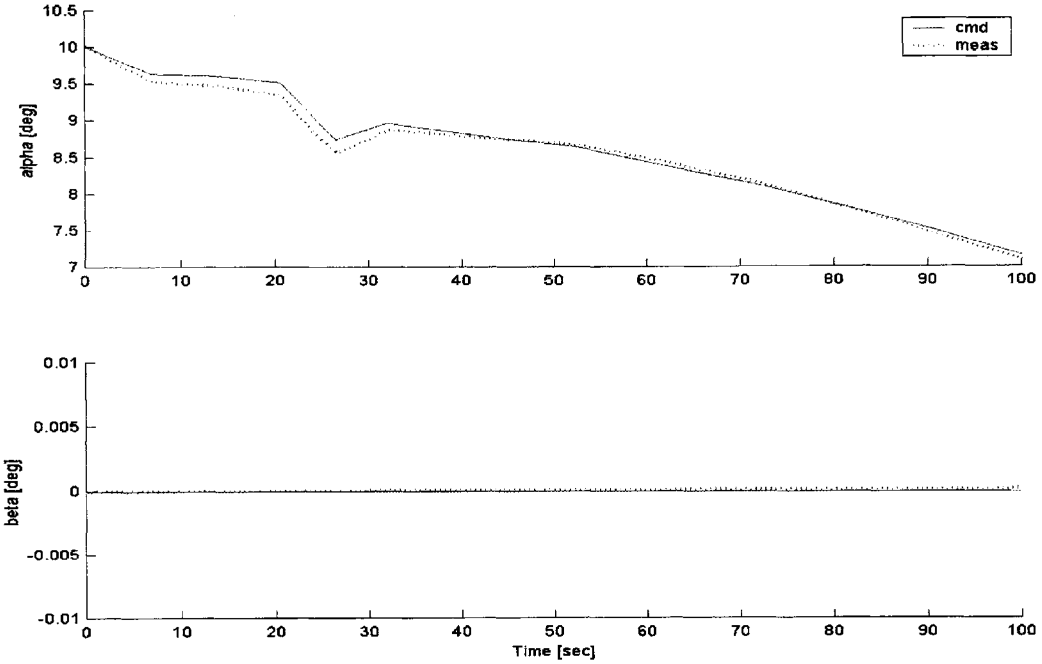


Figure 9-9. Actuator Failure Simulation: Elevon Stuck at t = 20s (3 of 6)

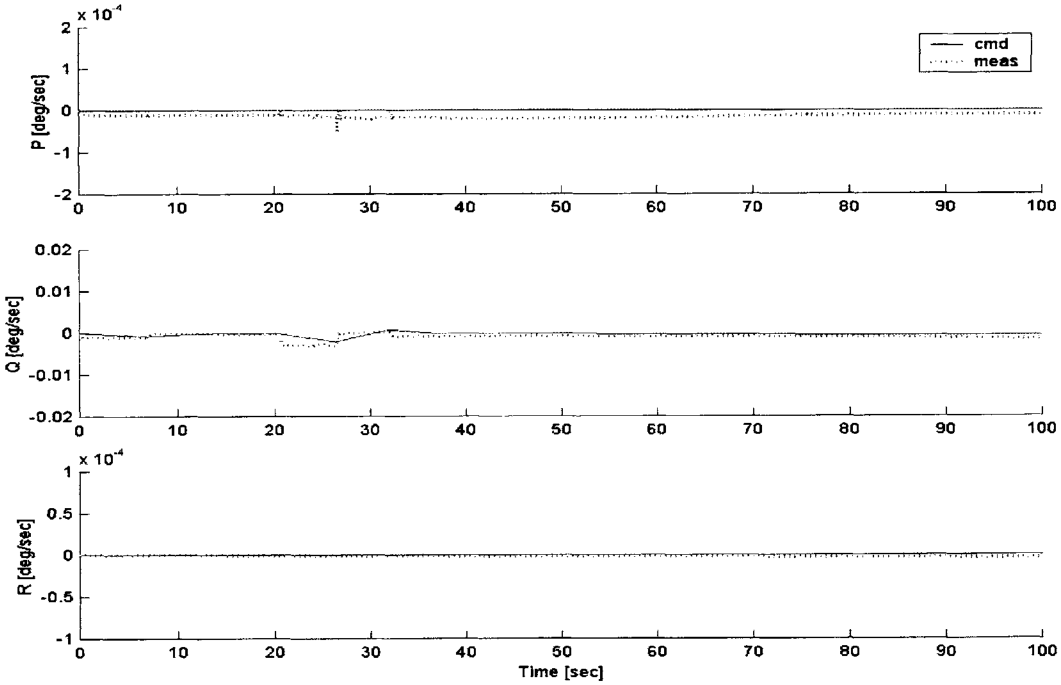


Figure 9-10. Actuator Failure Simulation: Elevon Stuck at t = 20s (4 of 6)

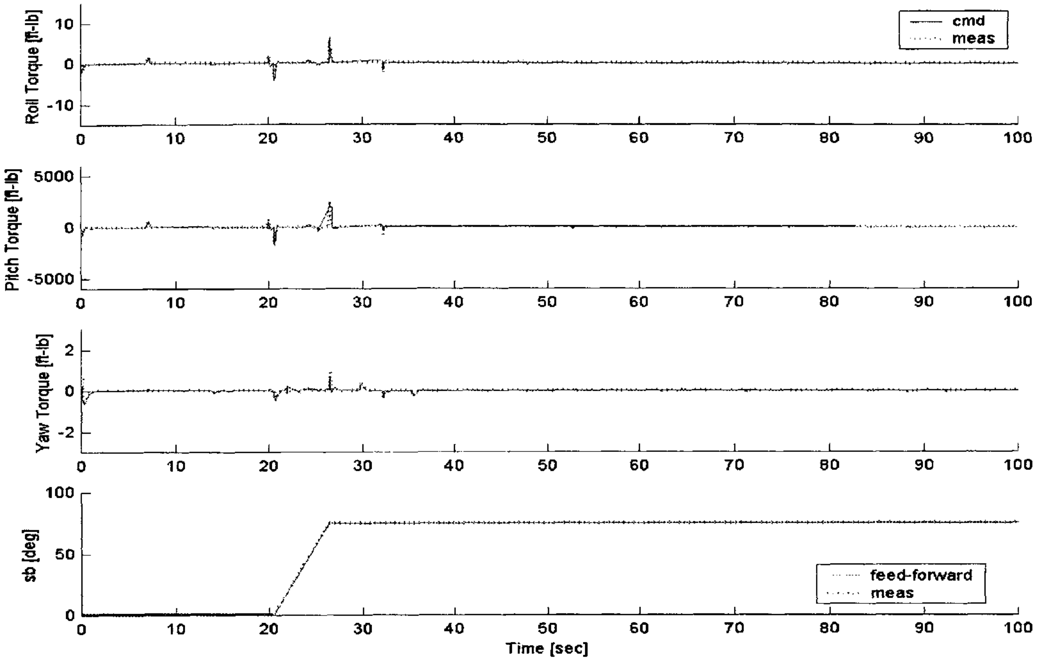


Figure 9-11. Actuator Failure Simulation: Elevon Stuck at t = 20s (5 of 6)

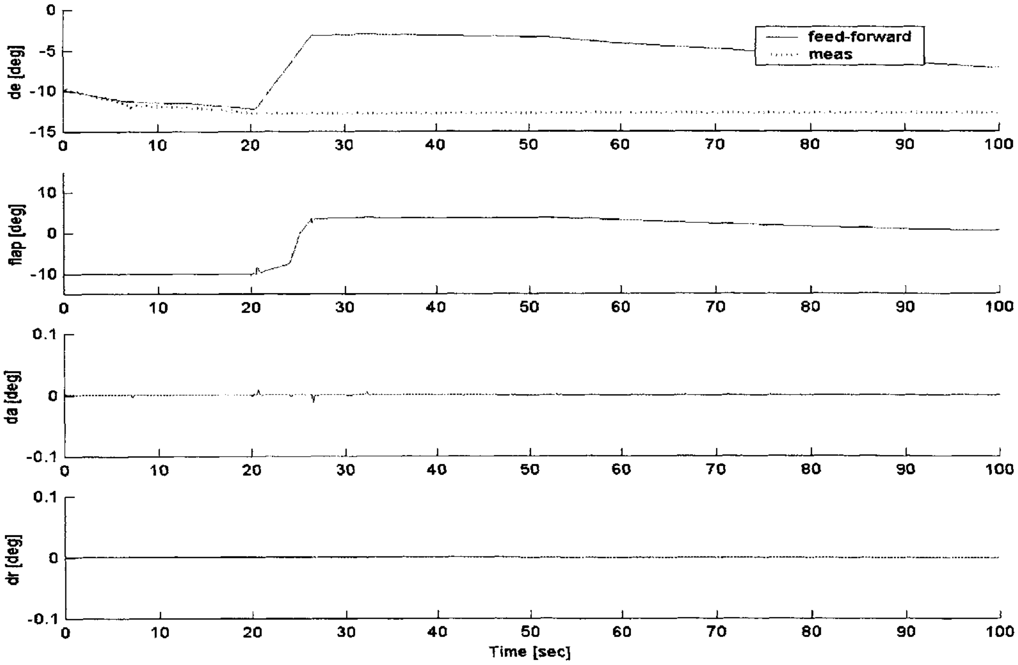


Figure 9-12. Actuator Failure Simulation: Elevon Stuck at t = 20s (6 of 6)

The final six plots relate to the case where the rudder is unavailable until time = 20 seconds

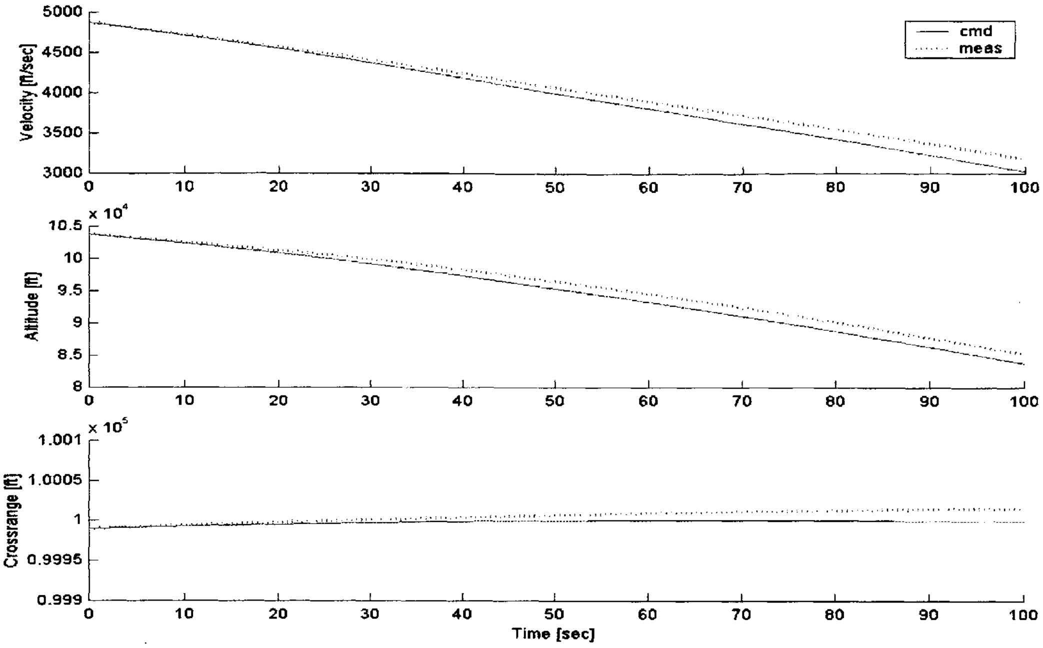


Figure 9-13. Actuator Failure Simulation: No Rudder until t = 20s (1 of 6)

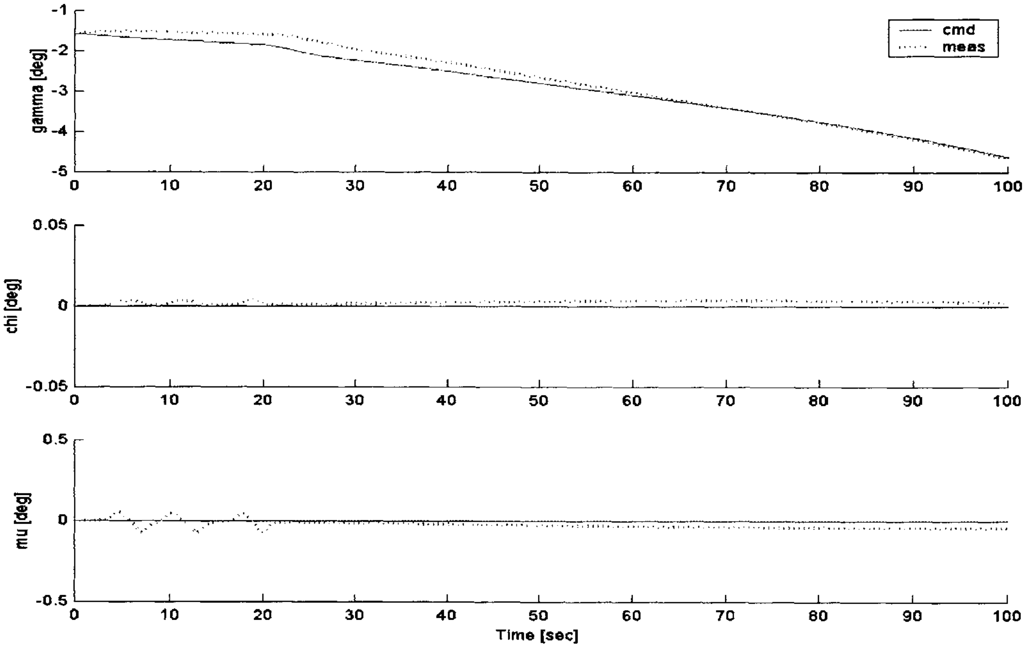


Figure 9-14. Actuator Failure Simulation: No Rudder until t = 20s (2 of 6)

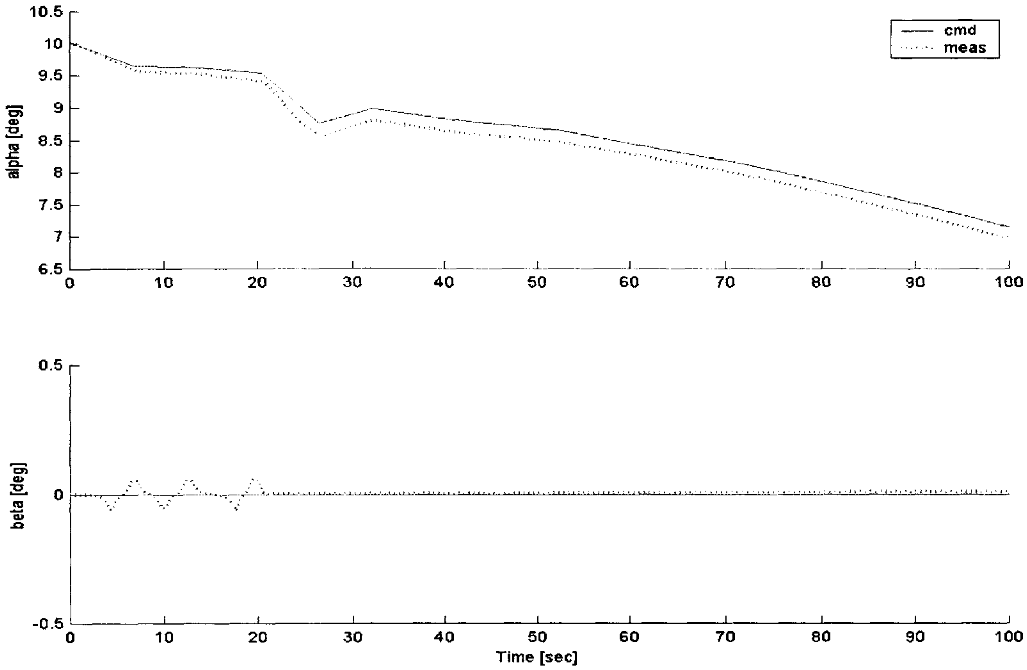


Figure 9-15. Actuator Failure Simulation: No Rudder until t = 20s (3 of 6)

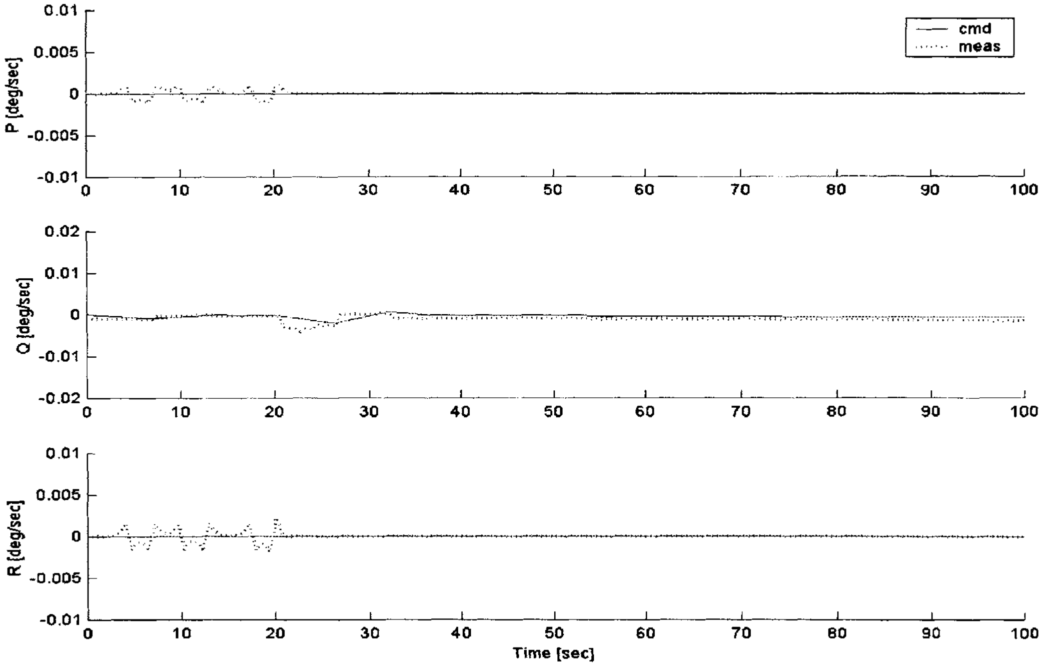


Figure 9-16. Actuator Failure Simulation: No Rudder until t = 20s (4 of 6)

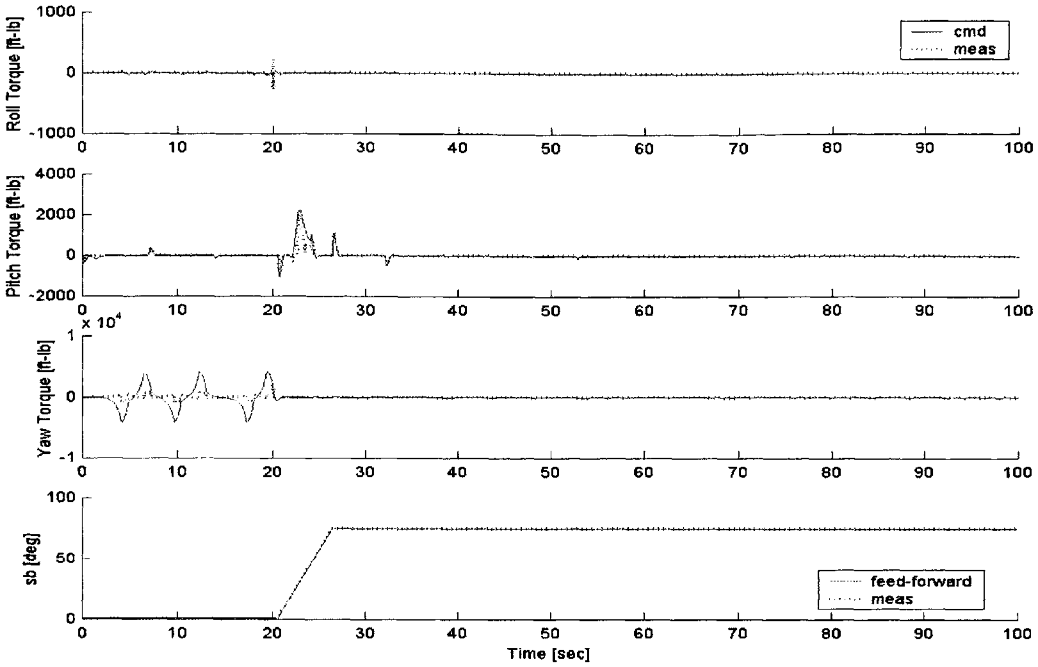


Figure 9-17. Actuator Failure Simulation: No Rudder until t = 20s (5 of 6)

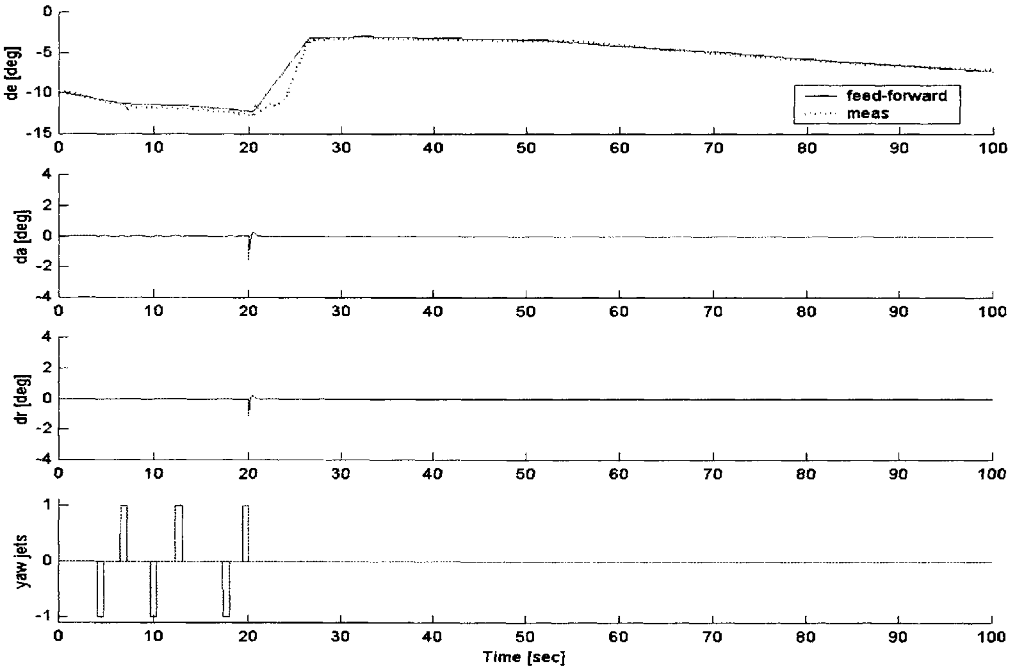


Figure 9-18. Actuator Failure Simulation: No Rudder until t = 20s (6 of 6)

References

[1] Deger, D., *Entry Guidance Workbook.* NASA Flight Training Branch, TD357A, October 1996.

[2] Weber, J., ed., *Entry Digital Autopilot.* NASA Space Flight Training Division, TD1012A, January 1995.

[3] *Application of Multivariable Control theory to Aircraft Control Laws.* Flight Dynamics Directorate, Wright Laboratory, WL-TR-96-3099, Wright-Patterson Air Force Base, OH, May 1996.

[4] Durham, W.C., "Constrained Control Allocation," *Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 4, 1993, pp. 717-725.

[5] Durham, W.C., "Constrained Control Allocation: Three-Moment Problem," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 2, 1994, pp. 330-336.

[6] Durham, W.C., "Computationally Efficient Control Allocation,' AIAA-99-4214, August 1999.

[7] Buffington, J.M., "Tailless Aircaft Control Allocation," AIAA-97-3605, August 1997.

[8] Virnig, J.C. and Bodden, D.S., "Multivariable Control Allocation and Control Law Conditioning when Control Effectors Limit," AIAA-94-3609, August 1994.

[9] Hodel, A.S., "Robust Inversion and Data Compression in Control Allocation," AIAA-2000-4154, August 2000.

[10] Paradiso, J.A., "Application of Linear Programming to Coordinated Management of Jets and Aerosurfaces for Aerospace Vehicle Control," C.S. Draper Laboratory Report, CSDL-R-2065, Nov 1988.

[11] Orbital Sciences Corporation, X-34 homepage, http://www.orbital.com/LaunchVehicles/X34/x-34.htm, May 2001.

[12] Dragone, T.L. and Hipp, P.A., "Materials Characterization and Joint Testing on the X-34 Reusable Launch Vehicle," SAMPE Journal, Vol. 34, No. 5, 1998, pp. 8.

[13] Chomel, C., "Design of a Robust Integrated Guidance and Control Algorithm for the Landing of an Autonomous Reusable Launch Vehicle," Masters Thesis, Department of Aeronautics and Astronautics, MIT, June 1998.

[14] Friedland, B., *Control System Design: An Introduction to State-Space Methods*, McGraw Hill, Inc., New York, 1986.

[15] Winston, W.L., *Operations Research: Applications and Algorithms*, International Thomson Publishing, Belmont, CA, 1994.

[16] Hillier, F.S. and Lieberman, G.J., *Introduction to Operations Research*, McGraw-Hill, Inc., New York, 1995.

[17] Kolcum, E.H., "Discovery Launch Demonstrates Shuttle Turnaround Capability," *Aviation Week & Space Technology*, June 24, 1985.

[18] Van de Vegte, J., *Feedback Control Systems*, 3rd ed., Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1994.

2966 · 37