Constrained Control Allocation for

Systems with Redundant Control Effectors

by

Kenneth A. Bordignon

Dissertation submitted to the Faculty of the

Virginia Polytechnic Institute and State University

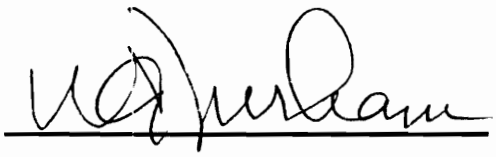
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

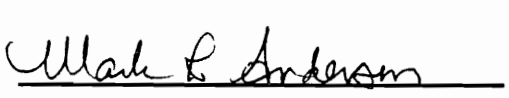
IN

AEROSPACE ENGINEERING

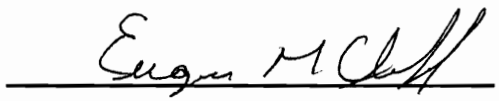
APPROVED:



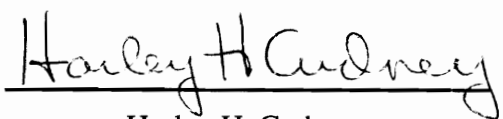
Wayne C. Durham, Chair



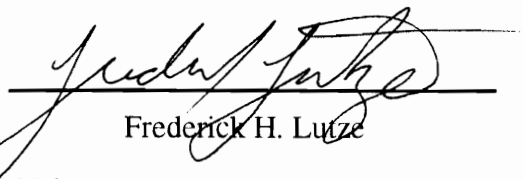
Mark R. Anderson



Eugene M. Cliff



Harley H. Cudney



Frederick H. Lutze

December 19, 1996

Blacksburg, Virginia

Keywords: Controls, Redundant, Allocation, Rate

CONSTRAINED CONTROL ALLOCATION FOR SYSTEMS WITH REDUNDANT CONTROL EFFECTORS

by

Kenneth A. Bordignon

Wayne C. Durham, Chairman

Aerospace Engineering

(ABSTRACT)

Control allocation is examined for linear time-invariant problems that have more controls than degrees of freedom. The controls are part of a physical system and are subject to limits on their maximum positions. A control allocation scheme commands control deflections in response to some desired output. The ability of a control allocation scheme to produce the desired output without violating the physical position constraints is used to compare allocation schemes.

Methods are developed for computing the range of output for which a given scheme will allocate admissible controls. This range of output is expressed as a volume in the n-dimensional output space. The allocation schemes which are detailed include traditional allocation methods such as Generalized Inverse solutions as well as more recently developed methods such as Daisy Chaining, Cascading Generalized Inverses, Null-Space Intersection methods, and Direct Allocation.

Non-linear time-varying problems are analyzed and a method of control allocation is developed that uses Direct Allocation applied to locally linear problems to allocate the controls. This method allocates controls that do not violate the position limits or the rate limits for all the desired outputs that the controls are capable of producing. The errors

produced by the non-linearities are examined and compared with the errors produced by globally linear methods.

The ability to use the redundancy of the controls to optimize some function of the controls is explored and detailed. Additionally, a method to reconfigure the controls in the event of a control failure is described and examined. Detailed examples are included throughout, primarily applying the control allocation methods to an F-18 fighter with seven independent moment generators controlling three independent moments and the F-18 High Angle of Attack Research Vehicle (HARV) with ten independent moment generators.

ACKNOWLEDGMENTS

I would like to thank all of those who have helped guide me through wilderness of academia and who have helped me navigate the turbulent waters of life. This list of people is quite expansive, covering a wide range of friends, family, and mentors. In the interest of brevity, I won't list them all here, but you know who you are.

I would like to single out several individuals who have made a particularly strong impact on my life. Dr. Eric Jumper from Notre Dame, who gave me good advice, good opportunities, and a belief in my abilities. My advisor, Dr. Wayne C. "Bull" Durham, for the perfect work environment: I got to do work I enjoyed, with someone I truly enjoyed working with. The members of my committee, Dr. Anderson, Dr. Lutze, Dr. Cliff, and Dr. Cudney, for all their support and advice. And last but not least, I'd like to thank my friends Chris Niven and Jeff Noethe for, well, just being there. Thanks guys!

CONTENTS

1. Introduction 1
2. Review of Literature 11
3. Nomenclature and Conventions 19
4. Preliminaries 22
    Problem Statement 22
    Geometry of the Problem 24
5. Generalized Inverse Solutions 29
    The Geometry of Generalized Inverse Solutions 29
    Specifying a Generalized Inverse 30
    Calculating Π for a given Generalized Inverse 36
    Example 5-1
        Area Calculations 43
    Example 5-2
        Volume Calculations 48
    Best Generalized Inverses 56
    Example 5-3
        Maximizing the Volume of Π 56
    Unattainable Moments 60
6. Daisy Chaining Solutions 64
    Method Origin 64
    Description of Method 64
    Calculating Π for a given Daisy Chain 66
    Example 6-1
        4 Controls & 2 Moments 69
    Example 6-2
        F-18 HARV 77
    Unattainable Moments 80
7. Cascading Generalized Inverse Solutions 82
    Description of Method 82
    Example 7-1
        How the Method Works 85
    Example 7-2
        $B\mathbf{u} \neq \mathbf{m}_d$ 88
    Example 7-3
        Concavity 92
    Unattainable Moments 100
8. Null-Space Intersection Solutions 101
    Method Origin 101
    Description of Method 102
    Example 8-1
        A Low Order Example 105
    Example 8-2
        Revisiting the Cascading Generalized Inverse $B\mathbf{u}=\mathbf{m}$ 112
    Initial Solutions 115
    Unattainable Moments 115
9. Direct Allocation 118

Geometric Issues 118
Description of Method 120
Determining the Boundary of Φ 121
Example 9-1
Determining $\partial(\Phi)$ 128
Example 9-2
Checking a Facet 128
Unattainable Moments 130
Computational Considerations 131
Combining with Tailored Generalized Inverses 131
Example 9-3
Tailoring a Generalized Inverse 133
Example 9-4
Tailoring to a Set of Desired Moments 136
10. Linear Combinations of Solutions 140
Description of Method 140
Example 10-1
Combining Direct Allocation and Daisy Chaining 143
11. Performance 150
Relating Π to Performance 150
Weight Reduction 152
12. Discrete Time Direct Allocation 153
The Effects of Time 153
Description of Method 154
Numerical Considerations 161
Example 12-1
F-18 with $B = B(\mathbf{u})$ 165
Implementation Issues 192
Example 12-2
Filtering $\Delta\mathbf{u}$ 197
Error Reduction 199
Path Dependency 207
13. Function Optimization 209
Capabilities of Redundant Controls 209
Description of Method 209
Computational Considerations 211
Example 13-1
F-18 Minimize $\|\mathbf{u}\|_2$ 215
14. Control Failures 221
Types of Failures 221
Example 14-1
An Unidentified Control Failure 222
Example 14-2
An Identified Control Failure 226
Conclusions 230
References 233
Bibliography 238

Appendix A 239
Proof 1
No Generalized Inverse Will Allocate $\mathbf{u} \in \Omega \ \forall \ \mathbf{m} \in \Phi$ 239
Proof 2
Convexity is Preserved Under Linear Transformations 239
Proof 3
Objects Remain Parallel Under Linear Transformations 243
Proof 4
Points on $\partial(\Phi)$ Map to Unique Points on $\partial(\Omega)$ 243
Vita 252

1. INTRODUCTION

Control allocation algorithms determine how the controls of a system should be positioned so that they produce some desired effect. Constrained controls have limits on their maximum positions and maximum deflection rates. Systems with redundant control effectors have more controls to position than desired effects. For example, two controls may be used to produce a single output. There are many types of systems which have redundant control effectors that are constrained. This dissertation examines control allocation techniques applied to modern aircraft. The techniques are general and may be applied to other systems. However, some of the terminology used is specific to aircraft. When applying these techniques to other systems, be aware that "the aircraft" refers to the system being controlled and that "the moments" are the desired effects that the controls produce.

Control allocation is the part of an airplane's control system which determines how to deflect the control surfaces in response to some input of desired effects. In the early days of aviation, airplane control systems consisted entirely of mechanical linkages, and control allocation was a hardware problem dealt with during the design and construction phases. Control systems have greatly increased in their complexity since the inception of manned flight. This complexity arises from many different sources, most of which come from an attempt to enhance the maneuverability of the piloted vehicle. In many modern aircraft, the pilot no longer directly positions the control surfaces with the cockpit controls. In fly-by-wire aircraft, the pilot inputs are fed to a computer which uses the commands along with other data to calculate the desired forces and moments which need to be generated by the controls. A control allocation algorithm (often integral to the

control law) determines how to deflect the controls to produce these forces and moments. Signals are then sent to the system which powers the control surfaces, typically a hydraulic system, and the control surfaces are deflected. This dissertation addresses the control allocation algorithm portion of this problem.

Before delving into the intricacies of modern control allocation, a general review of how aircraft are controlled is appropriate. Conventional aircraft, which include a broad range of past and present vehicles, offer control through the use of various aerodynamic surfaces which can be deflected by the pilot via a system of levers, cables, and pulleys. Figure 1-1 from Reference 1 shows an example of a conventional aircraft. Of the six degrees of freedom a rigid body possesses, a pilot generally has direct control of only four: one linear degree of freedom, and the three rotational degrees of freedom. The linear degree of freedom (force in the longitudinal direction) is controllable using the force generated by the engine and is usually controlled by a lever which varies the thrust produced by the engine.

In airplane terminology, the three rotational degrees of freedom are designated roll, pitch, and yaw. These rotational degrees of freedom are controlled by the aerodynamic control surfaces. For an aircraft, an aerodynamic control surface is any movable object that alters the air flow around the vehicle and thus alters the local aerodynamic force. The aerodynamic controls for a conventional aircraft are ailerons (roll), elevators (pitch), and rudder (yaw).

When the ailerons are deflected, they change the lift of a wing section which primarily generates a rolling moment. However, by changing the air flow around the

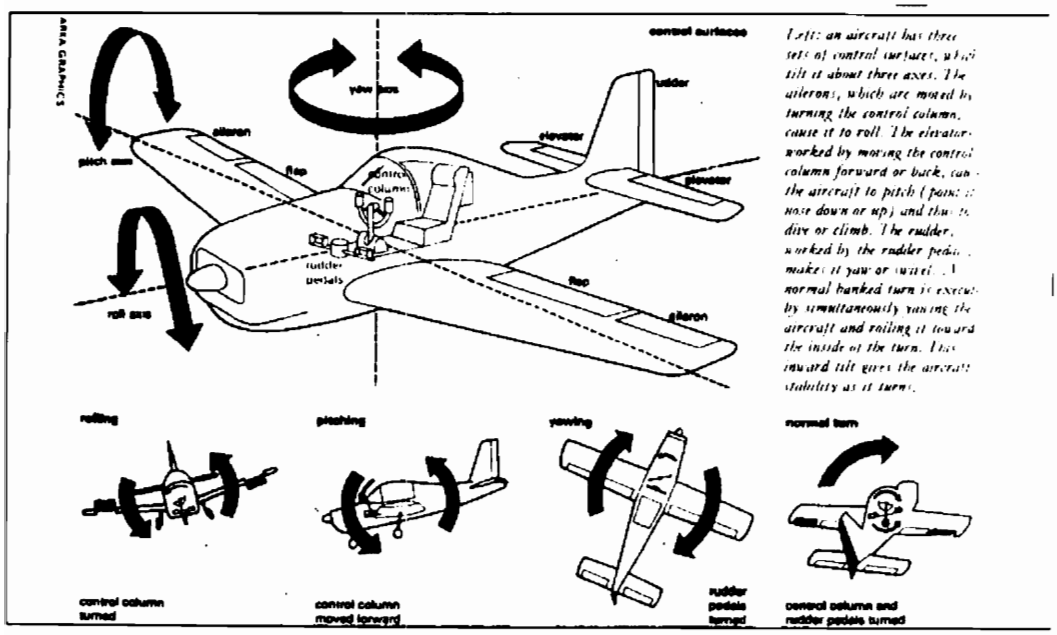


Figure 1-1: A Conventional Aircraft

wing, they also generate drag and alter the pitching and yawing moments produced by the wing section. Similarly, the elevators primarily produce a pitching moment, but also generate forces in the longitudinal and vertical directions. The rudder is primarily a yaw controller, although it also produces an unbalanced force in the lateral direction and a tendency to roll. Conventional aircraft do not have direct control of forces in the vertical and lateral directions. Most aircraft are equipped with flaps that are primarily used to generate lift. However, flaps are not generally used during maneuvering in conventional aircraft and are not considered a primary flight control.

An early example of control complexity can be found in aileron/rudder interconnects. As stated above, control surfaces do not typically generate purely single axis moments. In basic flight instruction, one is taught that during a turn "the outside wing produces more drag than the inside wing. This causes a yawing tendency toward the outside of the turn, called adverse yaw. You use rudder pressure to counteract this tendency."[2] Thus it is necessary to deflect the rudders when rolling to produce what is referred to as a coordinated turn. Some aircraft are equipped with an aileron/rudder interconnect so that the rudders are automatically deflected when the ailerons are deflected to help counter the adverse yaw tendency.

Further complexity arises when additional controls are developed and added to the conventional design to produce greater maneuverability. Spoilers are added for additional roll control and to effect lift and drag. Existing controls are allowed to move independently of one another. In this vein, controls called flaperons, which operate symmetrically as flaps and asymmetrically as ailerons, can be introduced. Examples of other control surfaces that may exist on modern aircraft include: elevons, tailerons, multiple rudders, trailing-edge flaps, leading-edge flaps, canards, maneuvering flaps,

speed brakes, etc. In the never ending search for control power, new controls are constantly being devised. Controls such as thrust vectoring and vortex flow control are being examined for next generation aircraft. The search for greater control power introduces a new problem in flight control systems. The number of controls is greater than the number of pilot inputs. This trend has significantly alters the complexion of the control allocation problem.

To understand control allocation, it is necessary to see how controls enter into an aircraft's equations of motion. These equations are based on Newton's Second Law of Motion which is frequently expressed as force is equal to mass multiplied by acceleration. Therefore, an aircraft's controls generate forces and moments which alter the aircraft's acceleration.

The equations of motion are typically written in the form of 12 non-linear ordinary differential equations. They assume that the aircraft is a rigid body and that the Earth is flat. Of the twelve equations, three are related to forces and three are related to moments. The remaining equations are related to vehicle position and orientation.

These nonlinear equations are frequently linearized about some reference condition and written in the common form:

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u} \tag{1-1}$$

$$\mathbf{y} = C\mathbf{x} + D\mathbf{u} \tag{1-2}$$

Where A, B, C, and D are matrices, $\mathbf{x}$ is a vector of the states, $\mathbf{u}$ is a vector of the controls, and $\mathbf{y}$ is a vector of the outputs. Many control law implementations for an aircraft will interpret the pilot inputs as desired changes to the aircraft rates, $\dot{\mathbf{x}}_d$. These

changes are partially due to the aircraft's inherent dynamics, $A\mathbf{x}$, and partially due to the controls, $B\mathbf{u}$. Once the control law determines the necessary changes in $\dot{\mathbf{x}}$ due to the controls, the control allocation problem is determined. Since most aircraft controls, with the exception of throttle, are primarily regarded as moment generators, the desired changes in $\dot{\mathbf{x}}$ due to the controls will be designated $\mathbf{m}_d$. The control allocation problem can now be stated as follows: Given a set of desired moments, $\mathbf{m}_d$, find the controls which satisfy:

$$\mathbf{m}_d = B\mathbf{u} \tag{1-3}$$

In conventional aircraft, there are typically three controls (ailerons, elevators, and rudders) and three desired moments(roll, pitch, and yaw). Often, the effects of the controls on the others states are negligible or non-existent and are ignored. Thus, the B matrix in Equation 1-3 is a square matrix and can usually be inverted to yield: $\mathbf{u} = B^{-1}\mathbf{m}_d$. However, for aircraft with redundant controls, B is no longer square, and other methods must be used to determine the control deflections.

The primary objective of a control allocator is to find a set of controls which satisfy the equation, $\mathbf{m}_d = B\mathbf{u}$. For systems with redundant control effectors, there will, in general, exist an infinite number of control combinations which can produce the desired forces and moments. Some of these combinations will be impossible for the hardware of the aircraft attain due to the physical limitations on the range of motion and rate capabilities of the control surfaces. These limitations are mathematically represented as bounds on $\mathbf{u}$ and $\dot{\mathbf{u}}$. The primary difficulty in control allocation is to command only control deflections which lie within the operational range of the aircraft hardware. Such control deflections will be referred to as admissible controls.

It is important to note that if there are no control combinations that satisfy the equation $\mathbf{m}_d = B\mathbf{u}$ and lie within the operational limits of the aircraft hardware, the control allocator is not at fault. A control allocator's function is to allocate admissible controls for any moment which it is possible to generate. Moments which can be generated using some combination of admissible controls will be referred to as attainable moments.

The control law should not command moments which are impossible to achieve. Saying this, however, does not prevent it from happening. In conventional aircraft, the pilot knew the limits of the aircraft controls simply from the fact that the control stick didn't move any farther. To prevent a control law from commanding too much of the controls, a complex relationship involving the pilot, the control law, and the control allocator needs to be developed. The solution to this dilemma is the subject of much on going research. Because this problem has not yet been solved, it is important that any method of control allocation to be implemented in a real aircraft will not run into numerical difficulties when no valid solutions exist. As a notable professor and ex-fighter pilot is fond of saying, "It is unacceptable to have a red warning light in the cockpit which reads: Algorithm did not converge, Eject!".

Other difficulties in control allocation arise from the fact that the linearized equation, $\mathbf{m}_d = B\mathbf{u}$, is only an approximation. As the aircraft flight condition changes over time, non-linear effects can introduce errors. A good control allocator should be able to deal with these changes in an effective and efficient manner which reduces these errors to an acceptable level.

In addition to solving for controls which satisfy $\mathbf{m}_d = B\mathbf{u}$, control allocators may be asked to take advantage of the redundant controls. They do this by attempting to minimize or maximize some function, such as drag or lift to drag ratio. Also, in the event that one or more controls becomes inoperable, due to some in-flight damage or malfunction, the control allocator may be asked to reconfigure the remaining controls to allow the pilot to maintain control.

This dissertation provides a comprehensive compilation of control allocation research performed at Virginia Polytechnic Institute and State University under NASA Grant NAG-1449, supervised by John V. Foster of NASA Langley Research Center. Other related research dealing with implementation issues is ongoing at Virginia Polytechnic Institute and State University under U.S. Navy contracts, and will be touched upon in this dissertation where appropriate. Archival publications that report results of research performed under this grant and Navy contracts may be found in the Bibliography.

Some of the results presented in this dissertation represent the state of research at the beginning of the author's involvement. These results provided a basis for much subsequent research. These include seminal results in the geometry of the control allocation problem and early attempts at the problem's solution. The majority of material in this dissertation constitutes the author's original research.

The author's contribution to the science and understanding of control allocation is primarily as follows:

- Techniques to compute the subset of moments for which a Generalized Inverse solution allocates admissible controls.

- Techniques to approximate the subset of moments for which a Daisy Chaining solution allocates admissible controls.
- Development of a method called Cascading Generalized Inverses and techniques which approximate the subset of moments for which this method allocates admissible controls.
- Development of a method called Null-Space Intersection solutions which allocates admissible controls for all the attainable moments.
- Techniques developed to apply control allocation to non-linear time varying problems.
- The current results of ongoing research conducted by the author and co-investigators to optimize functions of the controls and to reconfigure the controls in the event of control failures.

The material presented in this dissertation has been organized as follows: A general overview of control allocation is presented in the Problem Statement. The method of control allocation currently most widely in use (generalized inverse solutions) is then examined. The sections on Daisy Chaining solutions and the method of Cascading Generalized Inverses are presented immediately following the section on generalized inverses, as both of these methods make use of generalized inverses in finding solutions. The method of Null-Space Intersections uses ideas developed in the section on generalized inverses to find admissible control solutions for all the attainable moments, and is presented after the sections which make use of generalized inverses. Direct Allocation is then presented as a superior method for finding admissible controls for all

the attainable moments. The following sections describe methods for combining solutions and compare the previously discussed methods. Next, techniques which apply the method of Direct Allocation to non-linear time varying problems are discussed. Finally, the results of ongoing research are presented.

2. REVIEW OF LITERATURE

Early attempts to perform control allocation for aircraft with redundant control effectors involved various control mixing schemes and ad hoc solutions. As mentioned in the introduction, deflecting an aerodynamic control surface tends to produce multi-axis effects. However, these effects are often much stronger about one of the axes. For example, the rudder is much more effective at producing yaw than it is at producing roll or pitch. These early allocators would divide the controls into different groups which corresponded to different moment commands. For example, a 1982 NASA Conference publication describes a YF-16 with stabilators, flaperons and a rudder. For pitch control, collective stabilators are used. For roll control, differential stabilators and flaperons are used. For yaw control, only the rudder is used. [3] These methods were derived from experience and a logical, common sense approach to how surfaces typically acted. Unfortunately, these ad hoc methods limited the operational effectiveness of the controls.

Many aircraft were being given new control surfaces that were intended to provide additional capabilities. However, these new controls were not being fully utilized as a result of the control allocation methods implemented. The ability to reconfigure controls helped spur the movement for better control allocators. Reference 3 includes a "conceptual block diagram" which contains a surface allocation block, which is intended to reconfigure the aircraft in the event of a control failure. As Edmund G. Rynaski states, "These aircraft [AFTI-16, AFTI-111, X-29, and the Space Shuttle] have not a redundant, but a superfluous set of controls, and to the best of my knowledge...the restructurable control potential was not really considered." [4] He continues, "Today's aircraft are not configured for redundancy of controls. Also, aircraft control systems are not designed to

consider multiple use of existing controls." [4] The realization of this untapped potential provided the impetus for research into control allocation.

Methods which better utilize the controls came in the form of a generalized inverse solution. Generalized inverse solutions are defined as all matrices P which satisfy $BP = I$, when B has more columns than rows. The generalized inverse solution then yields controls from a single matrix multiplication. Unlike the early control mixing schemes, this method makes better use of the controls' multi-axis moment generating capabilities. Generalized inverses are often preferred for their ease of implementation and numerical efficiency. They became quite popular and dominate much of the literature. [5-13]

The generalized inverse known as the pseudo-inverse is frequently used. The pseudo-inverse is the matrix which will minimize the 2-norm of the control vector **u**, and it can be calculated using the formula [5] $P = B^T[BB^T]^{-1}$. The pseudo-inverse gives rise to a family of generalized inverses through the introduction of a weighting matrix N, $P = N(BN)^T[BN(BN)^T]^{-1}$. N is typically a diagonal matrix whose entries are used to emphasize/de-emphasize the various controls. This weighting is usually done in an attempt to prevent the control solutions from exceeding the physical limitations. Reference 6 uses a control allocator "based on generalized inverses which normalize the control effectiveness with respect to generalized inputs." Their method "normalizes the control effectiveness" using a diagonal weighting matrix "to take advantage of available control redundancy by allowing for control redistribution...." [6] In the example in Reference 6, once N is chosen, it does not vary.

Reference 7 suggests using different weighting matrices for different tasks. "By using complementary filters and different control selector solutions (through different

choices of *N*), one set of control effectors can be used for high frequency control and another set for low frequency, trim control." [7] In attempting to make the control allocator frequency-dependent, this method addresses the problem of rate saturation, which can occur at higher frequencies. Allowing for variations of this nature increase the numerical complexity of the control allocator since "the control selector transformation matrices will have to be calculated on-line or scheduled with flight condition and power setting." [7] References 8, 9, and 10 give further examples of the use of generalized inverses in control allocation.

In the mid-1980's, Frederick Lallman developed a method he refers to as pseudocontrols. [11,12] This method is similar to the above mentioned generalized inverses, however, it does have significant differences. The similarity lies in the fact that he multiplies a vector of desired quantities by a matrix to get the vector of controls. The difference comes from the method he uses to calculate what he calls the control mixing matrix.

Unlike generalized inverses, Lallman's method uses eigenvalue decomposition and modal analysis to compute the control mixing matrix. This procedure is done so that one may use the controls to drive the aircraft's modes (Dutch roll, spiral mode, roll mode, etc.). Thus, the method of pseudocontrols offered a new way to compute generalized inverses. In some applications of the method [13], the pseudocontrols are related to wind-axis moment generators rather than mode controllers. The wind-axis is defined such that the longitudinal axis points into the relative wind. By relating the pseudocontrols to wind-axis moments, pure wind-axis rolling moments or yawing moments may be commanded.

The major drawback of generalized inverse solutions lies in their inability to provide admissible control solutions for attainable moments. This property is due to the fact that the span of the space defined by a single generalized inverse will be less than the dimension of the control space. As a result, there are control combinations that are inaccessible using a generalized inverse. Some of these inaccessible controls are admissible and they map to moments that cannot be produced using only the admissible controls which a generalized inverse will allocate. The percentage of attainable moments for which a given generalized inverse will produce admissible controls can be quite small, which can degrade the system's performance.

There are methods that attempt to improve upon the performance of a single generalized inverse by using multiple generalized inverses. For example, one method deals with saturated controls by redistributing the controls with a second weighted pseudo-inverse. [15] This method uses a weighted generalized inverse to get a control solution. If any of the controls are commanded to be greater than the physical limits allow, the moment produced will not equal the desired moment. To remedy this situation, the controls that did not saturate are recomputed using a second generalized inverse in an attempt to provide the required moment. "The saturated controls are prevented from participating in the redistribution effort by nulling the associated weighting elements." [15] This method can significantly improve upon the performance of a single generalized inverse. However, it does not guarantee admissible control solutions for all attainable moments.

Initial concerns about the life-cycle costs of using thrust vectoring spurred the development of a control allocation method that utilized a certain set of controls only when necessary. [14] This method became known as daisy chaining. The method of daisy

chaining divides the controls into primary, $\mathbf{u}_1$, and secondary, $\mathbf{u}_2$, controls. Initially, only the controls in $\mathbf{u}_1$ are used. They are solved for using either matrix inversion or a generalized inverse. If any of the controls in $\mathbf{u}_1$ are commanded to move further or faster than is physically possible, the moment produce by these controls will not be equal to the desired moment. The controls in $\mathbf{u}_2$ are used to provide a moment equal to the difference between the desired moment and the moment produced by the controls in $\mathbf{u}_1$. Again, the controls in $\mathbf{u}_2$ are allocated using matrix inversion or a generalized inverse. Further information about daisy chaining can be found in References 6 and 14. Daisy chaining also suffers from an inability to provide admissible controls for all the attainable moments. However, it is usually an improvement over the performance of a single generalized inverse.

The topic of redundant controls is one that frequently arises in other fields, most notably that of robotics. A redundant robot has more degrees of freedom than rectilinear coordinates which specify the position of the end-effector. To position the end-effector, one must deal with the problem of mapping the desired manipulator position to a higher order control space which describes the joint angles. The equation for this is often written as $\dot{\mathbf{x}} = J(\theta)\dot{\theta}$. In this expression, $\dot{\mathbf{x}}$ is a vector representing the desired rectilinear velocities, similar to the desired moments, $\mathbf{m}_d$. $\dot{\theta}$ is a vector of the joint angle velocities and is similar to the control vector $\mathbf{u}$. There are two methods commonly used in robotics to solve this problem. The first is to use a generalized inverse such as those discussed previously. The second method involves adding rows to the J matrix to make it square and invertible. [16] These additional rows may be constructed in several different ways. Often, they are related to some "useful additional task, such as obstacle avoidance." [16]

Another aspect of control allocation frequently seen in robotics is the use of redundancy to minimize some auxiliary objective function. This minimization is usually done by using a pseudo-inverse to generate an initial control solution. Then, the gradient of some objective function is projected onto the null-space of the control effectiveness matrix and added to the control solution. [17] The null-space contains the directions in which the controls can vary without affecting the commanded variables. Thus, the additional $\dot{\theta}$ term has no effect on the resulting $\dot{\mathbf{x}}$ term. The objective functions used in robotics are often constructed with the intention of minimizing the joint deflection angles, θ.

Linear programming [18, 19] is a method widely used to solve constrained optimization problems. Some aspects of the control allocation problem may be cast as a standard linear programming problem. The constraints are the control position limits and $\mathbf{m}_d = B\mathbf{u}$. The cost to be minimized can be control magnitude, drag due to the controls, or some other function. Classical linear programming techniques, such as the Simplex method, first search for a basic feasible solution, and then improve upon this solution until the minimum is reached.

The major shortcoming of applying the linear programming method to flight controls involves unattainable commanded moments. In the event that the basic requirements of $\mathbf{m}_d = B\mathbf{u}$ cannot be satisfied using any combination of admissible controls, it is necessary in a real-time application that some rational choice of $\mathbf{u}$ be applied. The Simplex method may exit gracefully when the constraints cannot be satisfied, but it provides no information regarding the best control deflection to apply.

A newer method of problem solving that has become quite popular recently is neural network computing. Neural networks have been successfully used in a wide variety of fields including power conversion, noise filtering, and the effect of solar activity on orbit prediction. [20] Neural networks function by constructing a relationship between a set of inputs and a set of desired outputs. To construct this relationship, a set of training data is required. A neural network can then be used to emulate some existing method of control allocation.

The advantages of using a neural network stem from their ability to learn from experience. Research has been conducted to apply neural network technology to control allocation problems by Robert L. Grogan. [20] He suggests that "A neural network could be trained off-line on existing control effectiveness data. A second neural network could be linked in parallel to the first and trained on-line to compensate for modeling errors and parameter drift in real time...." [20] The benefits of this technology lie in the application of control allocation to real aircraft and adapting to the differences that exist between mathematical expressions and physical reality. However, a method of control allocation is still needed to provide the training data.

Additionally, there exist methods which combine several of the above mentioned techniques. One rather eclectic method was developed by R.D. Jones for missile control. [21] Initially, a generalized inverse variant is used to get the control deflections. To deal with control saturation, a linear programming problem is created. The solution to this linear programming problem is then obtained using a "Hopfield Artificial Neural Network." This method is a good illustration of the significant increase in the complexity of control allocation schemes that has occurred since the first simple control mixers.

Many existing methods fail to take full advantage of the capabilities of the controls which are available on modern aircraft. These methods fail to produce admissible control deflections for attainable moments. A deficiency in this area is a major concern because it translates directly to a loss in potential maneuverability or to the extra weight of controls which must be borne to achieve some desired level of maneuverability. By effectively allocating controls, new levels of performance and safety can be achieved.

3. NOMENCLATURE AND CONVENTIONS

A ..... The matrix [A]

A_i ..... Vector which is the i-th column of A

$\mathbf{a}$..... The vector {a}

a_i ..... The i-th element of the vector $\mathbf{a}$

$\mathbf{a}_i$ ..... A sub-vector of $\mathbf{a}$

$\mathbf{a_i}$ ..... The i-th vector $\mathbf{a}$

$_d$ ..... Subscript meaning desired

$\mathbf{u}$ ..... Vector of controls

m..... Number of controls

Ω..... Subset of constrained controls

$\partial(\Sigma)$ ..... Boundary of the set Σ

B ..... Matrix of control effectiveness

B_i ..... A vector which is the i th column of B

$\mathbf{m}$ ..... Vector of moments

$\mathbf{m}_d$..... Vector of desired moments

n ..... Dimension of the generalized moment space

Φ ..... Subset of attainable moments

AMS ..... Attainable Moment Subset, the same as Φ

$\exists$ ..... Such that

$\equiv$ ..... Identically equal to

n-D ..... Abbreviation for n-dimensional

$\aleph(B)$ ..... The null-space of B

P ..... A generalized inverse of B

P_S ..... The subspace in R^m defined by the columns of the generalized inverse P

θ ..... Subset of admissible controls allocated using a generalized inverse

$A^{\dagger}$ ..... The pseudo-inverse of matrix A

Π ..... Subset of moments attainable using some allocation scheme

N ..... An mxm weighting matrix, often diagonal

I_n ..... An nxn identity matrix

GI ..... Abbreviation for Generalized Inverse

$\mathbf{u}_{(sat)}$ .....Denotes that some or all of the controls in $\mathbf{u}$ are at their position limits

$S \times T$ ..... The Cartesian product, $S \times T = \{(s,t): s \in S \text{ and } t \in T\}$

DC ..... Abbreviation for Daisy Chaining

CGI ..... Abbreviation for Cascading Generalized Inverse

NSI ..... Abbreviation for Null-Space Intersection method

$\aleph$ ..... Matrix whose vectors form a basis for the null-space of B

$\mathbf{x}$..... Vector of dimension $(m\text{-}n)$x1

DA ..... Abbreviation for Direct Allocation

DTDA..... Abbreviation for Discrete Time Direct Allocation

$\dot{\mathbf{u}}$ ..... The derivative of $\mathbf{u}$ with respect to time

Ψ ..... The subset of $\partial(\Omega)$ which maps to $\partial(\Phi)$

4. PRELIMINARIES

Problem Statement

The simplest case of constrained control allocation will be examined initially. Consider an m-dimensional control space, $\mathbf{u} \in R^m$. The controls are constrained by their position limits only. These minimum and maximum values define the closed subset Ω.

$$\Omega = \{ \mathbf{u} \in R^m \mid u_{i\,Min} \leq u_i \leq u_{i\,Max} \} \subset R^m \tag{4-1}$$

Ω is a closed and bounded set. The subset of controls which lie on the boundary of Ω are denoted by $\partial(\Omega)$. If an element of the control vector, $\mathbf{u}$, is equal to one of its limiting values, $u_{i\,Min}$ or $u_{i\,Max}$, it will be referred to as a saturated control.

The controls generate moments through a mapping B onto n-dimensional moment space through the matrix multiplication of $\mathbf{u}$, $B\mathbf{u} = \mathbf{m}$, where $B : R^m \rightarrow R^n$. Since the class of problems to be examined involves redundant controls, $m > n$. This mapping arises from the linearization of the functional dependency of the moments on the controls.

$$\mathbf{m} = \mathbf{f}(\mathbf{u},\mathbf{x}) \tag{4-2}$$

The methods and techniques described in this dissertation can be applied to any system with redundant control effectors. However, the data and examples in this dissertation are aircraft oriented. In general, aircraft controls are thought of as moment generators. Sometimes the effects of the controls are expressed in other terms, such as changes in accelerations. To simplify terminology, the vector of desired quantities will be referred to as moments. It should be understood that $\mathbf{f}(\mathbf{u},\mathbf{x})$ need not represent

moments, only the effects of the controls expressed in terms of the states, $\mathbf{x}$, and/or the state derivatives $\mathbf{f}(\mathbf{u},\mathbf{x},\dot{\mathbf{x}})$.

B is the control effectiveness matrix. The elements of B are the partial derivatives of $\mathbf{f}$ with respect to the controls, $\mathbf{u}$. The reference condition at which these derivatives are evaluated is usually some predetermined flight condition, $\mathbf{x}_{ref}$, with the controls at zero deflection, $\mathbf{u}_{ref}$.

$$B_{ref} \equiv \frac{\partial \mathbf{f}}{\partial \mathbf{u}}_{ref} = \begin{bmatrix} \frac{\partial \mathbf{f}_1(\mathbf{u},\mathbf{x})}{\partial \mathbf{u}_1} & \frac{\partial \mathbf{f}_1(\mathbf{u},\mathbf{x})}{\partial \mathbf{u}_2} & \cdots & \frac{\partial \mathbf{f}_1(\mathbf{u},\mathbf{x})}{\partial \mathbf{u}_m} \\ \frac{\partial \mathbf{f}_2(\mathbf{u},\mathbf{x})}{\partial \mathbf{u}_1} & \frac{\partial \mathbf{f}_2(\mathbf{u},\mathbf{x})}{\partial \mathbf{u}_2} & \cdots & \frac{\partial \mathbf{f}_2(\mathbf{u},\mathbf{x})}{\partial \mathbf{u}_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \mathbf{f}_n(\mathbf{u},\mathbf{x})}{\partial \mathbf{u}_1} & \frac{\partial \mathbf{f}_n(\mathbf{u},\mathbf{x})}{\partial \mathbf{u}_2} & \cdots & \frac{\partial \mathbf{f}_n(\mathbf{u},\mathbf{x})}{\partial \mathbf{u}_m} \end{bmatrix}_{ref} \tag{4-3}$$

Denote by Φ the image of Ω in R^n, $\Phi \subset R^n$, mapped using B. The subset Φ represents all the moments that are attainable within the constraints of the controls and is often referred to as the AMS, for attainable moment subset. Note that these moments represent only the portion of the total moment which is due to the controls, defined as $\mathbf{m}_\mathbf{u}$. The moments generated by the effects of the states on the airframe are not included in the desired moment term. Moments which lie on the boundary of Φ are denoted by $\partial(\Phi)$.

The control allocation problem is defined as follows: given B, Ω, and some desired moment, $\mathbf{m}_d \in \Phi$, determine controls, $\mathbf{u} \in \Omega$, which generate that moment.

Geometry of the Problem

The geometry of the problem is easiest to visualize using lower dimensional problems. The same principles and concepts can be applied to higher dimensional problems, but it becomes more difficult to draw recognizable pictures. As an example, consider the case of three controls and two moments. The set of constrained controls, Ω, can be represented as a three-dimensional box (see Figure 4-1).

In order to speak clearly about the geometry of this problem, some terms should be defined. A subspace is a lower dimensional portion of a higher dimensional space which contains the origin of the higher dimensional space. For example, an infinite line which goes through the origin of a 3-D space is a subspace. A linear variety is a subspace which may have been translated from the origin. An n-dimensional linear variety is sometimes referred to as an n-flat. The term "object" will be used to refer to closed subsets in linear varieties. For example, an edge of the box in Figure 4-1 is a closed subset of an infinite line (1-D linear variety), and will be referred to as a 1-D object. The boundary of the 3-D box in Figure 4-1 is composed of objects of various dimensions: vertices of zero dimension, one dimensional edges, and two dimensional faces. Objects of dimension 4 or greater will be referred to as n-D hyper-boxes. To define an object, it is sufficient to know the location and connectivity of its vertices. [22]

To keep track of the vertices, a simple numbering system has been devised. At a vertex, each of the controls is at either a minimum or maximum. If a control is at its minimum, it is designated by a 0. If it is at a maximum it is designated with a 1. This way, a particular vertex can be represented by a base-2 number (i.e. $\{\mathbf{u}_1\,\mathbf{u}_2\,\mathbf{u}_3\} = \{101\} = 5$). The connectivity can then be determined by examining the base-2 numbers. If only

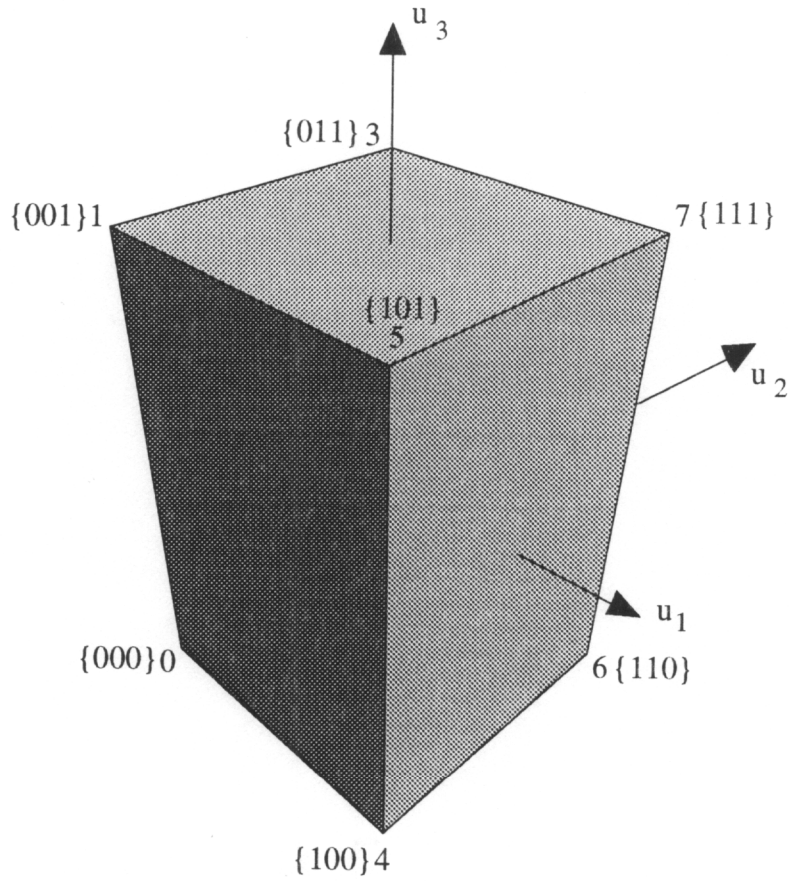


Figure 4-1: Constrained Control Subset, Ω

one digit is different for two numbers, then the vertices are connected by an edge. For example, vertex 5 {101} is connected to vertices 1 {001}, 7 {111} and 4 {100}.

Under the mapping B, the admissible control subset maps to a set of moments which can be attained using these controls. Figure 4-2a shows the 2-D projection of Figure 4-1, and Figure 4-2b shows the boundary of the attainable moment subset. As can be seen from Figure 4-2, some of the edges which are on the boundary of Ω map to the interior of Φ.

The exterior of Ω maps to the interior of Φ as a result of the loss of dimension that occurs under the mapping B. Most control allocation schemes will allocate controls in Ω for moments near the origin of Φ, but they allocate controls outside of Ω for moments near the boundary of Φ.

A control allocation scheme tries to establish a rule for mapping from Φ to Ω. Because the control space is of higher dimension than the moment space, a point in R^n will not map to a single point in R^m. There will exist a subspace of R^m which projects to the zero dimensional space in R^n. This subspace is called the null-space of B. It is denoted with by $\aleph(B)$ and is defined as follows:

$$\mathbf{u} \in \aleph(B) = \{ \mathbf{u} \in R^m \mid \exists\, B\mathbf{u} \equiv \mathbf{0} \} \subset R^m \tag{4-4}$$

According to Sylvester's law of nullity [23], a B matrix of rank n will have a null-space of dimension m-n. The case of 3 controls and 2 moments has a null-space which is a single direction in control space. Varying the controls by some scalar multiple of this vector will have no effect on the moment produced. For any point in moment space, there exists an infinite line in control space which will map to that point. For higher

dimensional problems, the linear varieties may be planes or hyperplanes in control space which will map to a single point in moment space.

If a point in moment space lies in Φ, only a portion of the linear variety in control space lies inside Ω. The challenge of constrained control allocation is to consistently find solutions which lie in Ω. If the moment is outside Φ, then the linear variety does not intersect Ω. The method of control allocation which will be described in Section 9 of this dissertation is called Direct Allocation. It uses the geometry of Φ to allocate the controls. This use of geometry will insure that controls in Ω are allocated for every moment in Φ.

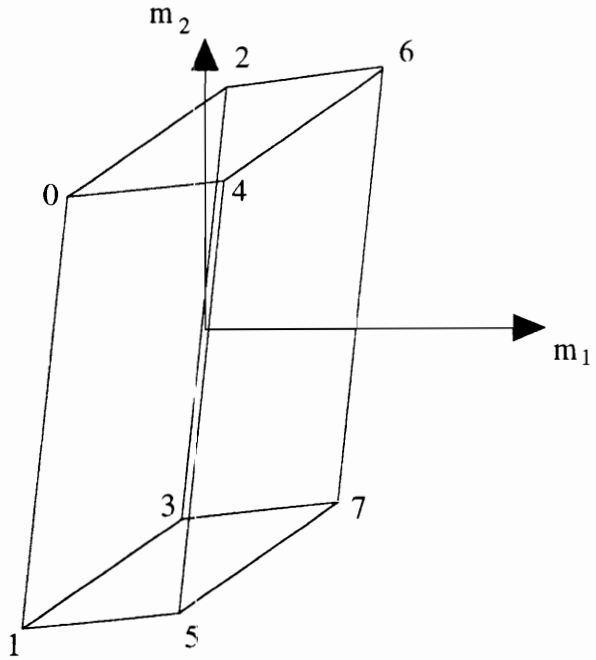


Figure 4-2a: Projection of Ω onto Moment Space

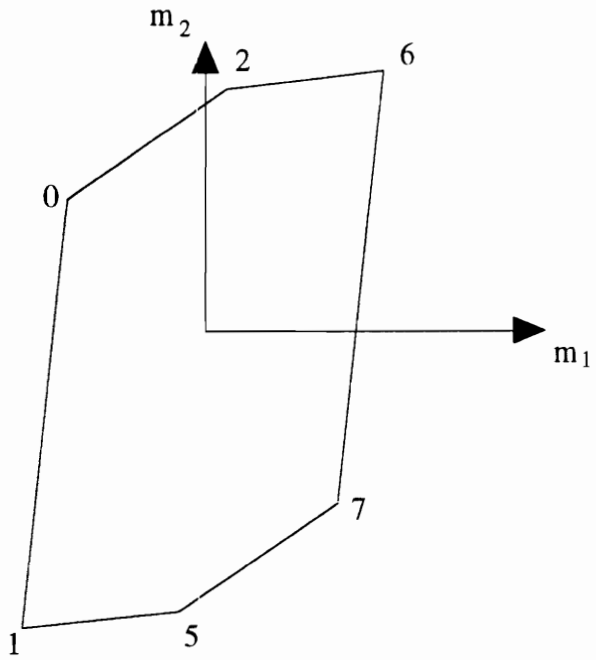


Figure 4-2b: $\partial(\Phi)$, The Boundary of The Attainable Moment Subset

5. GENERALIZED INVERSE SOLUTIONS

The Geometry of Generalized Inverse Solutions

The generalized inverse solution to the control allocation problem involves constructing a matrix which satisfies the equation $BP = I_n$, where I_n is an nxn identity matrix. The matrix P has n columns of length m. These columns are vectors in control space. These vectors form the basis of an n-dimensional subspace of R^m to which the moments will be mapped. This subspace will be referred to as P_s. Since this subspace is expressed as a matrix, it is easily seen that the zero vector in moment space maps to the zero vector in control space.

$$P\mathbf{0} = \mathbf{0} \tag{5-1}$$

Thus, P_s will contain the origin of control space. As long as the origin of control space is contained within Ω, P_s and Ω are guaranteed to intersect. This intersection represents all the valid control solutions which can be obtained using a particular generalized inverse, and will be denoted as θ.

$$\theta = \{P_s \cap \Omega\} \subset \Omega \subset R^m \tag{5-2}$$

By mapping this intersection using B, the moments for which the generalized inverse will allocate admissible controls can be seen. Since the columns of the generalized inverse do not span the control space, the generalized inverse cannot allocate admissible controls for all the attainable moments, except in certain degenerative cases. Proof of this can be found in Appendix A. The symbol Π will be used to designate the subset of moments attainable using some control allocation scheme.

$\Pi \subseteq \Phi$, i.e. $\Pi = B\theta$ (5-3)

In the example of the three-control two-moment problem, the generalized inverse defines a 2-D plane in 3-D control space. Different generalized inverses will define different planes which appear to rotate about the origin, as this is the only point that they must all contain. Figures 5-1a through 5-1d show different generalized inverse planes slicing through Ω. Note that the planes in these figures do not necessarily correspond to the same B matrix.

Specifying a Generalized Inverse

The pseudo-inverse is a standard method for solving an underdetermined set of linear equations. It solves the equations in a manner which minimizes the 2-norm of the vector $\mathbf{u}$, $\|\mathbf{u}\|_2$, and is sometimes referred to as the minimum norm solution. It can be derived from minimization principles as follows:

Minimize $\mathbf{u}^T\mathbf{u}$

Subject to $B\mathbf{u} = \mathbf{m}$

Define the scalar function $H(\mathbf{u},\lambda)$:

$$H(\mathbf{u},\lambda) = 0.5\mathbf{u}^T\mathbf{u} + \lambda^T(\mathbf{m} - B\mathbf{u}) \quad (5\text{-}4)$$

Where λ is an nx1 vector of Lagrange multipliers.

$H(\mathbf{u},\lambda)$ is an extremum when $\frac{\partial H(\mathbf{u},\lambda)}{\partial \mathbf{u}} = 0$ and $\frac{\partial H(\mathbf{u},\lambda)}{\partial \lambda} = 0$

$$\frac{\partial H(\mathbf{u},\lambda)}{\partial \mathbf{u}} = \mathbf{u}^T - \lambda^T B = 0 \quad (5\text{-}5)$$

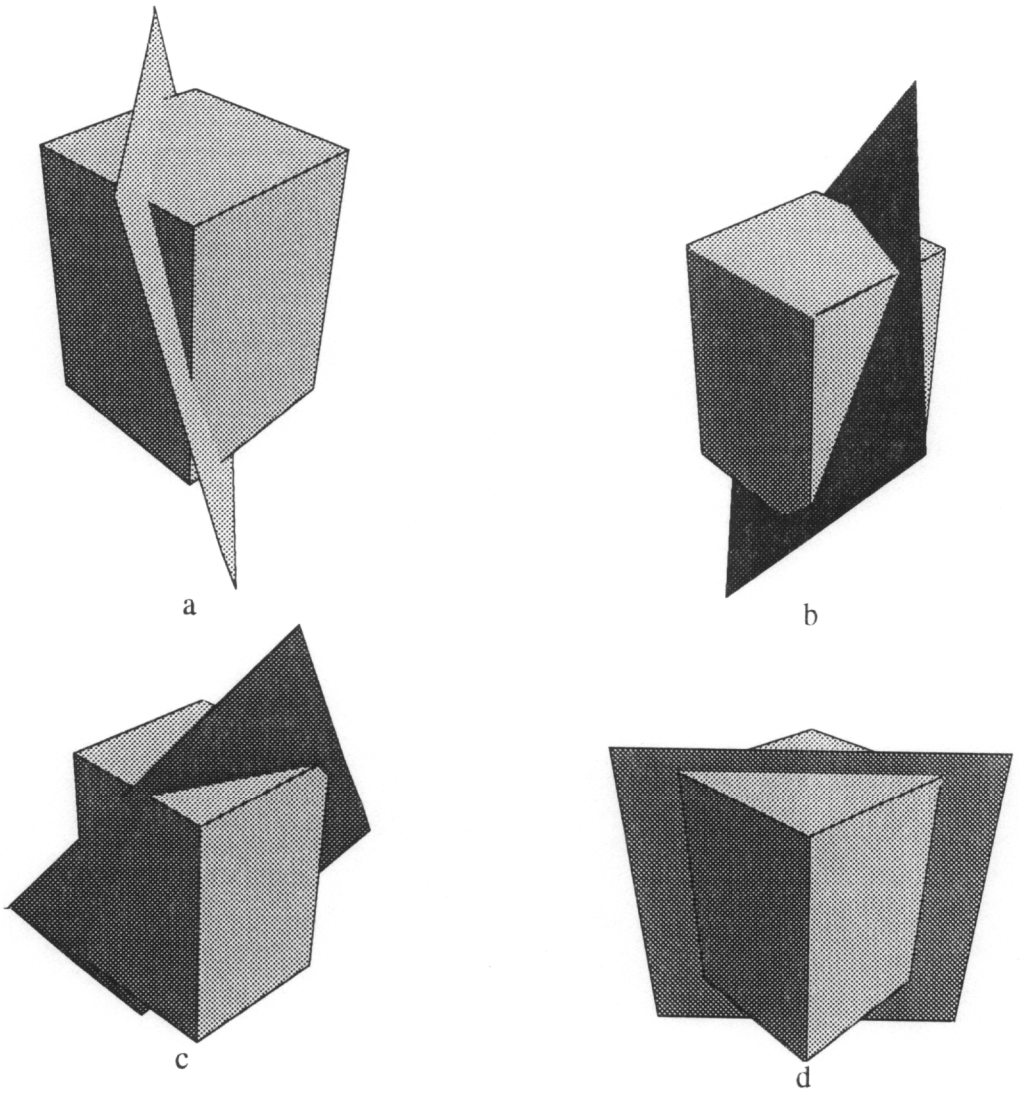


Figure 5-1: Generalized Inverse Planes Intersecting Ω

$$\frac{\partial H(\mathbf{u},\lambda)}{\partial \lambda} = \mathbf{m} - B\mathbf{u} = 0 \tag{5-6}$$

From Equation 5-5,

$$\mathbf{u}^T = \lambda^T B \Leftrightarrow \mathbf{u} = B^T\lambda \tag{5-7}$$

From Equation 5-6,

$$B\mathbf{u} = \mathbf{m} \tag{5-8}$$

Thus, the constraint equations are satisfied.

Plugging $\mathbf{u}$ from Equation 5-7 into Equation 5-8 yields:

$$B(B^T\lambda) = \mathbf{m} \tag{5-9}$$

Solving Equation 5-9 for λ

$$\lambda = (BB^T)^{-1}\mathbf{m} \tag{5-10}$$

Note that BB^T is an nxn matrix, and will have rank = n if B has rank = n. Thus, BB^T should be invertible. Plugging λ from 5-10 back into 5-7 yields

$$\mathbf{u} = B^T(BB^T)^{-1}\mathbf{m} \tag{5-11}$$

$$P = B^T(BB^T)^{-1} \tag{5-12}$$

$$\mathbf{u} = P\mathbf{m} \tag{5-13}$$

Note $BP = BB^T(BB^T)^{-1} = I_n$ and that $\frac{\partial^2 H(\mathbf{u},\lambda)}{\partial \mathbf{u}^2} = 1$ and $\frac{\partial^2 H(\mathbf{u},\lambda)}{\partial \lambda^2} = 0$, so this solution is a minimum.

Equation 5-12 is sometimes referred to as the right pseudo-inverse to denote that it is used for matrices with more columns than rows. The left pseudo-inverse is used for matrices with more rows than columns. The pseudo-inverse of B is sometimes denoted $B^{\dagger}$. The pseudo-inverse is just one of an infinite number of matrix solutions to $BP = I_n$. There are many different methods of computing P. A generalized inverse has mxn elements. However, when specifying a generalized inverse one is not free to specify all mxn elements. Some of the elements must be used to satisfy the equation, $BP = I_n$.

When specifying a generalized inverse, one defines an n-D subspace in an m-flat. The number of degrees of freedom of a n-flat in an m-flat is $(m - n)(n + 1)$. If the n-flat has r points fixed, the number of degrees of freedom of the n-flat is $(m - n)(n - r + 1)$. [25] Because the generalized inverse defines a subspace, one point, namely the origin, is already specified. Thus, the number of degrees of freedom in picking a generalized inverse is equal to $(m-n)n$.

One popular method for generating generalized inverses involves using a weighting matrix N, which is mxm. The weighting matrix is introduced in the pseudo-inverse equation to become:

$$P = N(BN)^T[BN(BN)^T]^{-1} \tag{5-14}$$

These P's are often referred to as weighted generalized inverses or weighted pseudo-inverses. Many allocation methods use a diagonal N matrix. [6,7] Using a diagonal N

offers the intuitive interpretation of using the weighting coefficients to emphasize/de-emphasize the controls to which they correspond. Frequently, the diagonal elements of N are equal to the reciprocal of the magnitude of the range of motion of the control, $N(i,i) = 1/|u_{i\,Max} - u_{i\,Min}|$.

However, in terms of degrees of freedom, this approach is inefficient. For a diagonal matrix, there are m degrees of freedom in picking N. m can be more or less than the number of degrees of freedom which exist for the generalized inverse. For 4 controls and 3 moments, there are 4 degrees of freedom in N, but only 3 degrees of freedom in P. This means that there are different N matrices which produce the same P. For 5 controls and 3 moments, there are 5 degrees of freedom in N, but 6 degrees of freedom in P. This means that there are generalized inverse solutions which cannot be generated using N. If all the elements of N are allowed to be non-zero, then there are mxm degrees of freedom in picking N, which will always be greater than the degrees of freedom in P.

A method for specifying generalized inverses which uses only independent variables is called Tailored Generalized Inverses. This method partitions B and P as follows:

$$B = [B_1\ B_2],\ B_1 \in R^{n\text{x}n},\ |B_1| \neq 0,\ B_2 \in R^{n\text{x}(m-n)} \tag{5-15}$$

[Note: if $|B_1| = 0$, rearrange the columns of B so that $|B_1| \neq 0$]

$$P = \begin{bmatrix} P_1 \\ P_2 \end{bmatrix},\ P_1 \in R^{n\text{x}n},\ P_2 \in R^{(m-n)\text{x}n} \tag{5-16}$$

The matrix P_2 contains the $(m - n)n$ independent variables. Using the equation $BP = I_n$, P_1 can be written in terms of B and P_2:

$$BP = I_n \Leftrightarrow B_1P_1 + B_2P_2 = I_n \Leftrightarrow B_1P_1 = I_n - B_2P_2 \Leftrightarrow P_1 = B_1^{-1} - B_1^{-1}B_2P_2 \tag{5-17}$$

While the matrix P_2 lacks the intuitive appeal of a diagonal weighting matrix, it has the important property of having the same number of degrees of freedom as P. This fact will be used later when talking about "best" generalized inverses. Also, P_2 can be used in conjunction with Direct Allocation to align the generalized inverse subspace to achieve the maximum attainable moment in specified directions. This method is detailed in Section 9 , which discusses Direct Allocation.

The method of pseudocontrols [11, 12] specifies a generalized inverse solution which controls the various modes of the system. The following description of this method is primarily taken from Reference 11. The method of pseudocontrols generates a control mixing matrix by first transforming the system $\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$ to a block diagonal form:

$$\dot{\mathbf{y}} = \Lambda\mathbf{y} + \Gamma\tilde{\mathbf{u}} \tag{5-18}$$

$$\mathbf{y} = M^{-1}\mathbf{x} \tag{5-19}$$

$$\tilde{\mathbf{u}} = W^{-1}\mathbf{u} \tag{5-20}$$

$$\Lambda = M^{-1}AM \tag{5-21}$$

$$\Gamma = M^{-1}BW \tag{5-22}$$

The matrix M is a diagonalizing similarity transformation matrix, whose columns are the Left Inverse Eigenvectors of A. The matrix W is a diagonal weighting matrix which has the maximum value of each of the control variables as its diagonal elements. Next, a vector of pseudocontrols, $\mathbf{v}$, is created. Where the length of $\mathbf{v}$ is equal to the number of modes to be controlled. The vector $\tilde{\mathbf{u}}$ is related to the pseudocontrols through the control mixing matrix, C.

$$\tilde{\mathbf{u}} = C\mathbf{v} \tag{5-23}$$

The columns of C, $\mathbf{c}_j$, are found such that they maximize

$$J = \mathbf{c}_j^T \sum_D \gamma_i \gamma_i^T \mathbf{c}_j - \mathbf{c}_j^T \sum_{ND} \gamma_i \gamma_i^T \mathbf{c}_j \tag{5-24}$$

subject to $\mathbf{c}_j^T \mathbf{c}_j = 1$

Once C is determined, the controls can be computed using $\mathbf{u} = WC\mathbf{v}_d$. This is analogous to the equation $\mathbf{u} = P\mathbf{m}_d$. The difference lies in the fact that the vector of commands, $\mathbf{v}_d$, is specifying modal commands instead of moment commands.

Calculating Π for a given Generalized Inverse

In order to calculate Π, it is necessary to be able to calculate the intersection of an n-D subspace with a bounded m-D subset. To describe how this is done, it will be necessary to use the terminology defined in Section 4. Additionally, the numbering system used to describe the vertices of Ω will be expanded so that it can be used to describe all of the objects which comprise $\partial(\Omega)$.

Recall that a vertex of Ω was described using a base-2 number. To describe the various objects on the boundary of Ω, a base-3 numbering system will be used. Again, the controls will be arranged in ascending order, and if a control is at its maximum it will be given a 1 and if it is at a minimum it will be given a 0. For example, $\{u_{1\,Max}\ u_{2\,Max}\ u_{3\,Min}\} = \{110\}$. However, only vertices can be defined with all the controls set at limiting values.

It was noted that two vertices are connected by an edge if only one digit is different in their describing numbers, i.e. {110} is connected to {111} by an edge. Along this edge, u_3 varies from its minimum to its maximum. To describe controls which are free to vary in a particular object, the number 2 will be used. Thus, the edge which connects {110} and {111} will be denoted as {112}. Higher dimensional objects can be represented in a similar fashion. For example, faces are 2-D objects which have two controls which vary. It can been seen that the face which lies on the top of the box in Figure 4-1 is {221}. The entire box, Ω, can be represented as all 2's, {222}.

This numbering system is especially useful when dealing with higher dimensional objects which are difficult to visualize. For example, it is easy to write that {1200210212} is a 4-D hyper-box which lies on the boundary of a 10-D hyper-box. This is seen from the fact that {1200210212} is a 10 digit number with 4 two's. However, drawing a recognizable picture of a 4-D hyper-box on the boundary of a 10-D hyper-box would prove more difficult.

An important feature of n-D geometrical objects is that they are bounded by (n-1)-D objects. This fact can be seen in lower dimensional objects that are frequently dealt with: 1-D edges are bounded by 0-D vertices, 2-D faces are bounded by 1-D edges, 3-D boxes are bounded by 2-D faces. This is less easily seen, but nonetheless true for higher dimensional objects. For example, Figure 5-2 shows a 4-D hyper-box with one of the 3-D boxes which is on its boundary shaded. It is important to note that everything inside the 3-D box is on the boundary of the 4-D hyper-box.

The subset Ω is a convex polytope in R^m. A linear subspace, such as P_s, is convex. Thus, the intersection of P_s and Ω , θ, will be a convex polytope. [22] To define this

polytope, it is sufficient to find all of the extreme points, or the vertices of the polytope. [22] These extreme points will be points that are on the boundary of Ω and also in the subspace P_s. Through a systematic method of examination and elimination, it is possible to determine all such points.

To determine the extreme points of θ, it is necessary to search the objects which lie on $\partial(\Omega)$. The number of conditions that an n-flat and a q-flat in S_m should intersect in an r-flat is $(r+1)(m-n-q+r)$. This implies that $n+q \leq m+r$. If $n+q > m+r$ they intersect in a region of dimensions $n+q-m$, which is greater than r. [25] To find points ($r = 0$) of intersection for the n-D subspace P_s, in an m-D space, we must search objects of dimension q, such that $n+q \leq m \Leftrightarrow q \leq m-n$. Thus, the dimension of the objects which need to be searched is $(m-n)$-D or smaller. For example, a 2-D plane may intersect a 1-D edge of a 3-D box at a point ($3-2 \leq 1$), but it can not intersect a 2-D face of the 3-D box at a point. Note, however, that a 2-D plane may intersect a 2-D object of a 4-D (or higher) Ω at a single point. This fact may seem counterintuitive, but in dealing with higher order problems, intuition is often unreliable. This rule can also be applied to determine higher order intersections. For example, lines ($r = 1$) will be formed in intersections with objects of dimension $\leq (m-n+1)$ and faces ($r = 2$) will be formed in intersections with objects of dimension $\leq (m-n+2)$.

To search for the vertices of θ, one must generate every $(m-n)$-D object and find their intersections with P_s. Following the discussion above, on an $(m-n)$-D object n controls are at a minimum or maximum deflection and $(m-n)$ are free to vary. $(m-n)$-D flats are generated by setting every combination of m controls taken n at a time to every combination of their minimum/maximum deflections. There are $m!/(n!(m-n)!)$ combinations of control groupings, and 2^n combinations of minimum/maximum

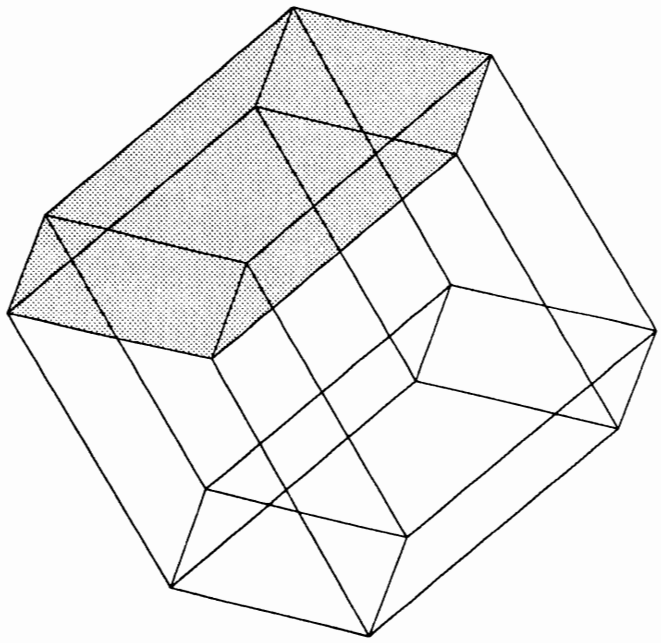


Figure 5-2: 4-D Hyper-Cube and 3-D Box on Boundary

deflections for each group. The values of the remaining (free) (m-n) controls are then determined. If the value of each of the (m-n) controls lies within its constraints, then the intersection is within Ω, otherwise it is not.

The determination of the intersections is done as follows:

1) Partition P into 2 sections so that P_1 corresponds to the n controls to be set to their minimum/maximum values. Partition $\mathbf{u}$ similarly.

$$\begin{bmatrix} P_1 \\ P_2 \end{bmatrix} \mathbf{m} = \begin{Bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{Bmatrix} \Rightarrow P_1\mathbf{m} = \mathbf{u}_1 \text{ and } P_2\mathbf{m} = \mathbf{u}_2$$

2) Set the controls in $\mathbf{u}_1$ to each of the possible combinations of minimum and/or maximum values.

3) Solve for $\mathbf{m}$: $\mathbf{m} = P_1^{-1}\mathbf{u}_1$

P_1 should be square and invertible. If P_1 is singular, then there are an infinite number of solutions. This can occur if the n-D subspace contains some lower dimension object. This P_1 can be ignored because the defining intersections will be found by searching other objects. However, note that such intersections may be found by searching several objects.

4) Solve for $\mathbf{u}_2$: $\mathbf{u}_2 = P_2\mathbf{m} = P_2P_1^{-1}\mathbf{u}_1$

5) Repeat (1)-(4) trying all possible combinations of P_1 and P_2.

Solving for the controls in this manner guarantees that they lie in P_s. If controls in $\mathbf{u}_2$ are at or within their limits then that point is also on $\partial(\Omega)$, and is a vertex of θ. Each vertex is denoted by its control vector, $\mathbf{u}$, and corresponding object number, e.g. {121}.

Once the vertices are determined, they can be mapped to moment space using B. The vertices of θ will map to become the vertices of Π. Once the vertices of Π are known, the connectivity of the vertices needs to be determined to completely define Π. The connectivity can be determined by either of two means. The first involves using a convex hull generating algorithm such as the quickhull algorithm described in Reference 26. Such algorithms can be used to accurately determine Π because θ is convex, and under the linear transformation B, Π will also be convex. A proof of this is included in Appendix A.

The second way to determine the connectivity of the vertices involves using the object numbers. The vertices were found on objects of (m-n)-D or smaller. They are connected if there is an (m-n+1)-D or smaller object which contains both points. By examining the object numbers, it can be determined if two vertices are contained by a (m-n+1)-D object. An (m-n)-D object has n fixed controls. If two vertices have the same n fixed controls, they have the same object number and lie on the same (m-n)-D object. If they have n-1 fixed controls in common, they lie on the same (m-n+1)-D object and are connected. For example, the edges {002} and {020} have one fixed control in common, $u_1 = u_{1\,Min}$, and lie on the face {022}. Because the criterion for intersection contains a less than or equal to condition, there is a need to check to see if there is an (m-n+1)-D or smaller object that contains two vertices. If such an object exists, then the two vertices are connected.

Once the vertices and their connectivity are known, the volume of Π can be found. This volume can be compared to the volume of Φ to give a relative measure of how well a particular generalized inverse allocates the controls. Although n is usually equal to 3,

this is not necessary for these techniques to be applied. Therefore, methods for the calculating *n*-D volumes will be described.

The subset of moments attainable by an allocation scheme can usually be represented as a convex *n*-Dimensional object bounded by (*n*-1)-D objects. To calculate the volume of such an object, the (*n*-1)-D bounding objects are treated as the bases of *n*-D pyramids with their apices at some point on or within Π, typically the origin of moment space. The sum of the volumes of all these pyramids is the volume of the *n*-D object. The following is a generic method for calculating the volumes of pyramids with arbitrary polygons for bases that may be applied to both Φ and to Π.

The n-volume of a parallelepiped is the (n-1) volume of one of its faces times the altitude on that face. [27] If σ is an n-D object in R^n defined by the points $A_0 \ldots A_n$ then its volume can be computed by the following equation: [27]

$$V(\sigma) = \frac{1}{n!}[\overrightarrow{A_0A_1} \cdots \overrightarrow{A_0A_n}] \tag{5-25}$$

In the nomenclature of Reference 27, the quantity in square brackets is the outer product of the vectors. The 1/n! comes from the fact that this formula calculates the volume enclosed by the vectors not the parallelepiped described by the vectors. Due to the definition of the outer product, this equation can also be written as follows:

$$V(\sigma) = \frac{1}{n!}\mathrm{Det}([A]) \tag{5-26}$$

[A] is a matrix whose rows are the vectors $\overrightarrow{A_0A_1} \cdots \overrightarrow{A_0A_n}$. σ is an *oriented* volume. This means that if σ changes its orientation, then σ changes its sign. [27] Since the quantity of interest is the magnitude and not the orientation of these volumes, the absolute value of the above equations is taken. To calculate the volume of Π, an *n*-dimensional volume,

divide its bounding objects into (n-1)-D objects containing n points (i.e. divide the polygons into triangles).The n-D objects can be divided into the (n-1)-D objects once the connectivity of the vertices is known. Use some point on or within Π as the n+1 point from which the vectors are formed. If the origin is contained within Π, it is easy to form the vectors by taking the origin as the n+1 point and simply use the coordinates of the points as the rows of [A]. If there are k such bounding objects, then the n-volume of Π can be calculated as:

$$V(\Pi) = \frac{1}{n!}\sum_{i=1}^{k} |\mathrm{Det}([A_i])| \tag{5-27}$$

Example 5-1: Area Calculations

As a simple example of the above ideas, consider a problem with 3 controls and 2 moments. The control effectiveness matrix is:

$$B = \begin{bmatrix} 7.35e\text{-}4 & 7.55e\text{-}4 & -1.35e\text{-}4 \\ 8.56e\text{-}5 & 5.13e\text{-}4 & -1.37e\text{-}3 \end{bmatrix} \tag{5-28}$$

And the limits on the controls positions are:

$-20 \le u_1 \le 20$

$-20 \le u_2 \le 20$

$-30 \le u_3 \le 30$

Figure 4-1 shows Ω for this problem. The pseudo-inverse of B is:

$$P_{pseudo} = \begin{bmatrix} 7.54e2 & -1.83e2 \\ 6.41e2 & 4.92e1 \\ 2.87e2 & -7.23e2 \end{bmatrix} \tag{5-29}$$

Figure 5-3 shows the subspace of the pseudo-inverse intersecting Ω for this problem. For $m = 3$ and $n = 2$, all the edges of Ω need to be searched for intersections. Table 5-1 shows the results of this search.

Table 5-2 shows the vertices of Π, numbered in the order they were found, and their connectivity. For $m = 3$ and $n = 2$, two vertices are connected if the object numbers have one fixed control in common. Figure 5-4 shows a wire-frame of Φ with Π from the pseudo-inverse inside. The convex hull of the AMS can be calculated by mapping the vertices of Ω and using a convex hull algorithm such as that of Reference 26. A more efficient means of computing the boundary will be detailed in the section on Direct Allocation.

Once the connectivity is known, the area of Π can be determined by dividing it into triangles and computing their areas using Equation 5-27. The area of Π is 4.3528e-03 which is approximately 83.6% of the area of Φ, 5.2051e-03.

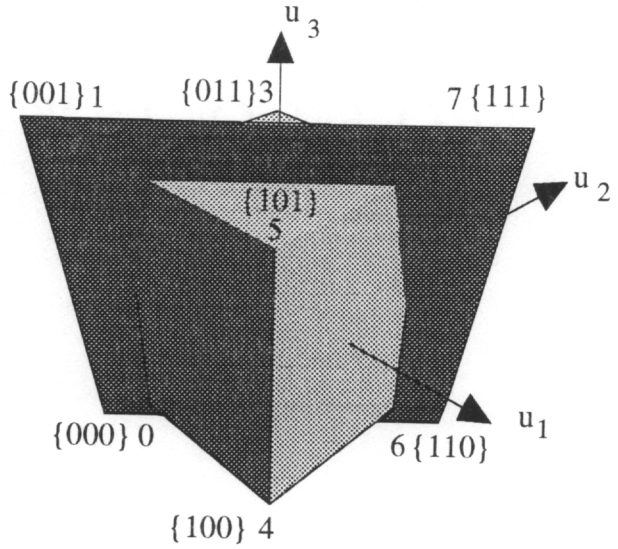


Figure 5-3: $P_s \cap \Omega$ for Example 5-1

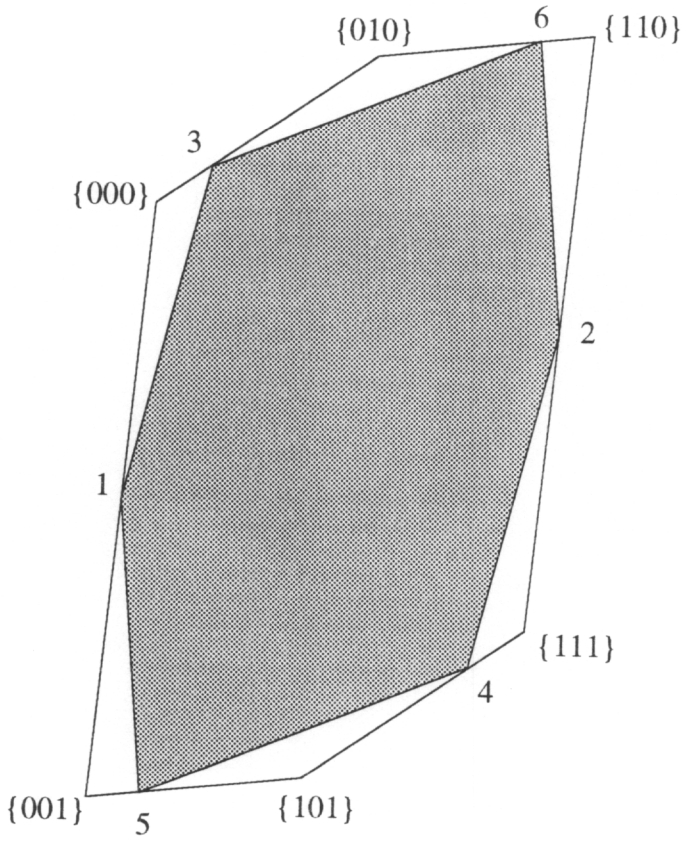


Figure 5-4: Φ and Π for Example 5-1

Table 5-1: Search Results for Pseudo-Inverse

Object #	$\mathbf{u}_1$	$\mathbf{u}_2 = P_2P_1\text{-}1\mathbf{u}_1$	Is it a vertex of Π?
{002}	$u_1 = -20$ $u_2 = -20$	$u_3 = 1.9396$	yes
{012}	$u_1 = -20$ $u_2 = 20$	$u_3 = -125.5006$	no
{102}	$u_1 = 20$ $u_2 = -20$	$u_3 = 125.5006$	no
{112}	$u_1 = 20$ $u_2 = 20$	$u_3 = -1.9396$	yes
{020}	$u_1 = -20$ $u_3 = -30$	$u_2 = -9.9750$	yes
{021}	$u_1 = -20$ $u_3 = 30$	$u_2 = -28.8074$	no
{120}	$u_1 = 20$ $u_3 = -30$	$u_2 = 28.8074$	no
{121}	$u_1 = 20$ $u_3 = 30$	$u_2 = 9.9750$	yes
{200}	$u_2 = -20$ $u_3 = -30$	$u_1 = -30.3397$	no
{201}	$u_2 = -20$ $u_3 = 30$	$u_1 = -10.9161$	yes
{210}	$u_2 = 20$ $u_3 = -30$	$u_1 = 10.9161$	yes
{211}	$u_2 = 20$ $u_3 = 30$	$u_1 = 30.3397$	no

Table 5-2: Vertices of Π

Vertex #	Object #	Vertices it connects to	Coordinates in Moment Space
1	{002}	3 & 5	{-0.0301, -0.0146}
2	{112}	4 & 6	{ 0.0301, 0.0146}
3	{020}	1 & 6	{-0.0182, 0.0343}
4	{121}	2 & 5	{ 0.0182, -0.0343}
5	{201}	1 & 4	{-0.0272, -0.0523}
6	{210}	2 & 3	{ 0.0272, 0.0523}

Example 5-2: Volume Calculations

The following example applies the above techniques to a higher order problem involving 10 independent controls. The data in the B matrix are based on a linear approximation to the F-18 High-Angle-of-Attack-Research-Vehicle (HARV) aerodynamic data. The model was linearized about a flight condition of 10,000 ft., Mach 0.3, and 12.5° angle-of-attack. The controls are: horizontal tails, ailerons, rudders, trailing edge flaps, and three thrust vectoring controls. The left and right ailerons, horizontal tails, and trailing edge flaps are allowed to move independently of one another and are treated as separate controls. The left and right rudders are constrained to move together and are treated as a single control. The thrust vector control effectiveness was approximated from data in Reference 6. This data will be used in several examples throughout this dissertation and shall in the future be referred to as the F-18 HARV data.

$$B = \begin{bmatrix} \text{-4.382e-2} & \text{4.382e-2} & \text{-5.841e-2} & \text{5.841e-2} & \text{1.674e-2} & \\ \text{-5.330e-1} & \text{-5.330e-1} & \text{-6.486e-2} & \text{-6.486e-2} & \text{0.000e+0} & \cdots \\ \text{1.100e-2} & \text{-1.100e-2} & \text{3.911e-3} & \text{-3.911e-3} & \text{-7.428e-2} & \end{bmatrix}$$

$$\cdots \begin{bmatrix} \text{-6.280e-2} & \text{6.280e-2} & \text{2.920e-2} & \text{1.000e-5} & \text{1.000e-2} \\ \text{6.234e-2} & \text{6.234e-2} & \text{1.000e-5} & \text{3.553e-1} & \text{1.000e-5} \\ \text{0.000e+0} & \text{0.000e+0} & \text{3.000e-4} & \text{1.000e-5} & \text{1.485e-1} \end{bmatrix} \tag{5-30}$$

For this combination of B matrix and control deflection limits, shown in Table 5-3, the subset of attainable moments is as shown in Figure 5-5.

Table 5-3: F-18 HARV Control Position Limits

	Control Surface	Minimum Value (radians)	Maximum Value (radians)
u_1	Right Horizontal Tail	-4.189e-1	1.833e-1
u_2	Left Horizontal Tail	-4.189e-1	1.833e-1
u_3	Right Aileron	-5.236e-1	5.236e-1
u_4	Left Aileron	-5.236e-1	5.236e-1
u_5	Combined Rudders	-5.236e-1	5.236e-1
u_6	Right Trailing Edge Flap	-1.396e-1	7.854e-1
u_7	Left Trailing Edge Flap	-1.396e-1	7.854e-1
u_8	Roll Thrust Vector Vane	-5.236e-1	5.236e-1
u_9	Pitch Thrust Vector Vane	-5.236e-1	5.236e-1
u_{10}	Yaw Thrust Vector Vane	-5.236e-1	5.236e-1

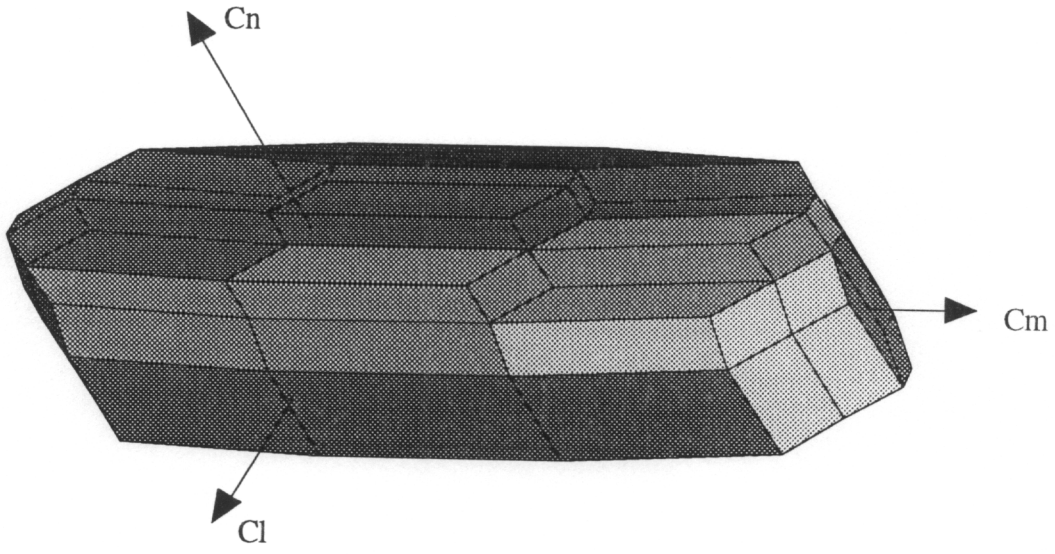


Figure 5-5: Φ for Example 5-2

The pseudo-inverse of B is:

$$P_{pseudo} = \begin{bmatrix} -2.201e+0 & -7.500e-1 & -3.350e-1 \\ 2.202e+0 & -7.500e-1 & 3.353e-1 \\ -2.960e+0 & -9.125e-2 & 2.210e-1 \\ 2.960e+0 & -9.128e-2 & -2.210e-1 \\ 9.492e-1 & 1.383e-5 & -2.693e+0 \\ -3.177e+0 & 8.774e-2 & 8.643e-2 \\ 3.177e+0 & 8.771e-2 & -8.646e-2 \\ 1.477e+0 & 6.081e-6 & -2.940e-2 \\ 3.958e-4 & 4.999e-1 & 2.5786e-4 \\ 3.015e-1 & -2.537e-5 & 5.325e+0 \end{bmatrix} \quad (5\text{-}31)$$

For $m = 10$ and $n = 3$, all the 7-D hyper-boxes of Ω need to be searched for intersections. Table 5-4 shows only the intersections which are vertices of Π. If the object numbers of two vertices have two or more fixed controls in common, they are connected by an edge. If the object numbers have 1 or more fixed control in common, they lie on the same face of Π.

These vertices connect to form the faces which bound the three dimensional Π. Table 5-5 list vertex connections and Table 5-6 lists the vertices which lie on the same face. Figure 5-6 shows the vertices of Π_{pseudo} connected by a wire-frame.

Figure 5-7 shows Π_{pseudo} inside a wire-frame of Φ. Again, Equation 5-27 can be used to find the volumes of Π_{pseudo} and Φ. For this example, the volume of Π_{pseudo}, 1.238e-2, is only 13.7% of the volume of Φ, 9.013e-2.

Table 5-4: Vertices of Π

Vertex #	Object #	Coordinates in Moment Space
1	{2022220220}	{-0.0562, 0.3509, -0.0951}
2	{2022220221}	{-0.0535, 0.4467, 0.1014}
3	{1222220220}	{-0.0446, -0.0707, -0.0958}
4	{1222220221}	{-0.0361, -0.1831, 0.1004}
5	{2122202220}	{ 0.0361, -0.1831, -0.1004}
6	{2122202221}	{ 0.0446, -0.0706, 0.0958}
7	{0222202220}	{ 0.0535, 0.4467, -0.1013}
8	{0222202221}	{ 0.0562, 0.3510, 0.0951}
9	{0022222220}	{ 0.0151, 0.5585, -0.0992}
10	{0022222221}	{-0.0151, 0.5585, 0.0992}
11	{1122222220}	{ 0.0151, -0.2444, -0.0992}
12	{1122222221}	{-0.0151, -0.2444, 0.0992}

Table 5-5: Vertex Connections

Vertex #	Connects to Vertices
1	2,3,9
2	1,4,10
3	1,4,11
4	2,3,12
5	6,7,11
6	5,8,12
7	5,8,9
8	6,7,10
9	1,7,10
10	2,8,9
11	3,5,12
12	4,6,11

Table 5-6: Faces of Π

Face #	Vertices on Face
1	1,2,4,3
2	1,2,10,9
3	1,3,11,5,7,9
4	2,4,12,6,8,10
5	3,4,12,11
6	5,6,8,7
7	5,6,12,11
8	7,8,10,9

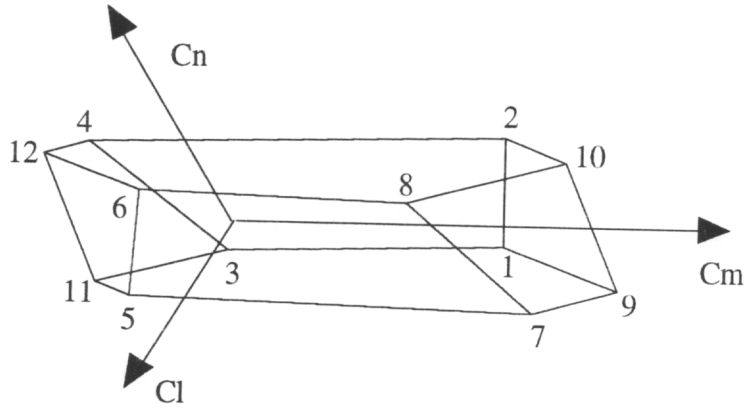


Figure 5-6: Π_{pseudo} for Example 5-2

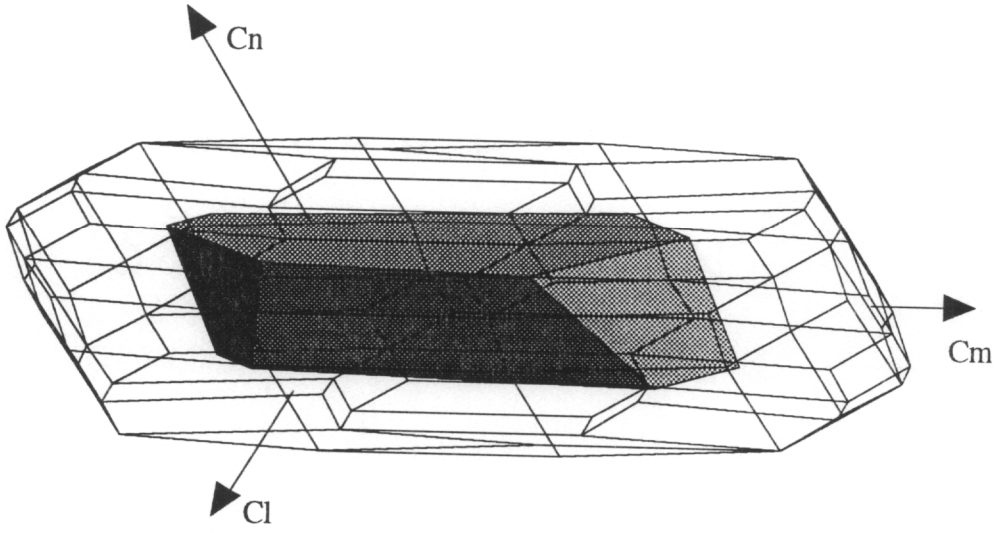


Figure 5-7: Π_{pseudo} inside Φ for Example 5-2

Best Generalized Inverses

With the ability to calculate Π for a given generalized inverse, and the ability to systematically vary generalized inverses, it is possible to search for the generalized inverse which maximizes the volume of Π. This generalized inverse will be referred to as the best generalized inverse, P_{Best}. To search for the best generalized inverse, the method of Tailoring will be used. This method is used because Tailoring exercises the proper number of degrees of freedom in specifying a generalized inverse. The elements of the P_2 matrix can be systematically varied using searching algorithms until a maximum is found. Typically, the pseudo-inverse is used as an initial guess.

While the existence of a unique global maximum has not been proven, some parameter sweeps can be used to suggest that one exists. Figure 5-8 shows some representative parameter sweeps for the F-18 HARV data. The initial values in P_2 are those given in the bottom 7 rows of P_{pseudo}, Equation 5-31. By varying a single element of P_2 and holding the others fixed, it can be seen that the volume has a single maximum for a given parameter sweep. Additionally, several tests were conducted in which different initial guess for P_2 were used and each time a maximization algorithm converged to the same final result. This also suggests that a global maximum exists. However, it is possible that there will not exist a unique P_2 which corresponds to the global minimum.

Example 5-3: Maximizing the Volume of Π

The data for the B matrix is the F-18 HARV data used in the previous example. The elements of P_2 were varied using a Simplex routine from Reference 28.

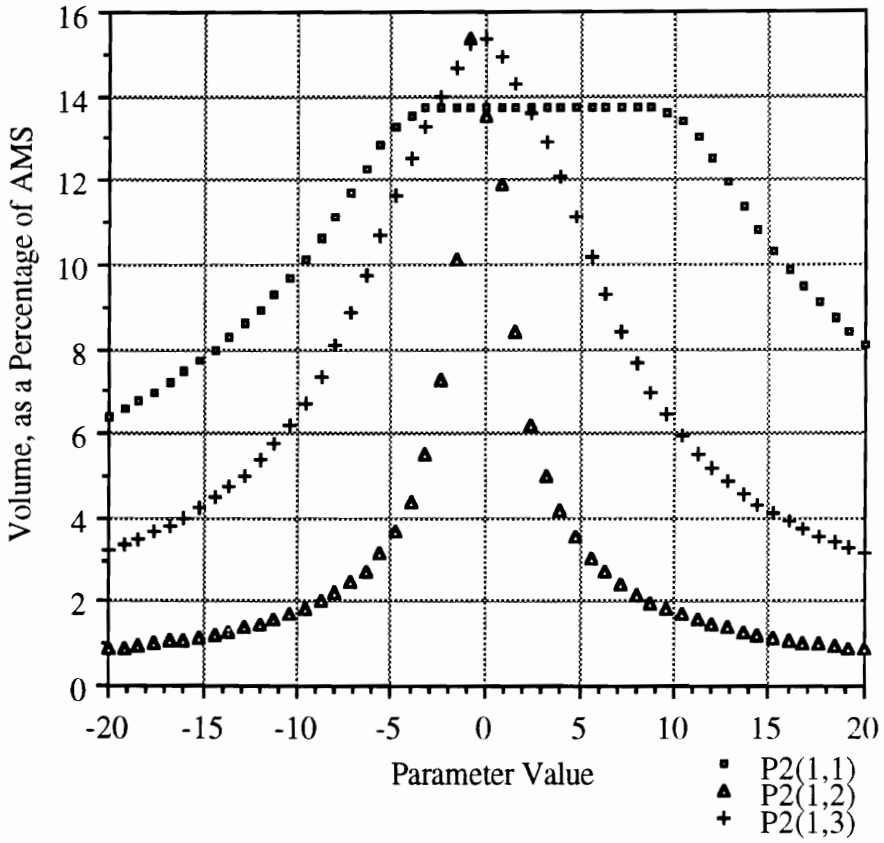


Figure 5-8: Parameter Sweeps

The best generalized inverse was calculated to be:

$$P_{Best} = \begin{bmatrix} -6.047e\text{-}3 & -6.600e\text{-}1 & -1.130e\text{-}2 \\ 1.321e\text{-}1 & -6.586e\text{-}1 & -5.608e\text{-}3 \\ -5.543e{+}0 & -4.889e\text{-}3 & -8.715e\text{-}2 \\ 5.543e{+}0 & 5.098e\text{-}3 & 8.283e\text{-}2 \\ 3.917e\text{-}1 & -1.641e\text{-}6 & -4.268e{+}0 \\ -1.398e{+}0 & 3.934e\text{-}2 & -1.882e\text{-}1 \\ 1.452e{+}0 & 2.897e\text{-}2 & 3.182e\text{-}2 \\ 5.520e{+}0 & -3.700e\text{-}5 & 4.878e\text{-}2 \\ 1.041e\text{-}3 & 8.243e\text{-}1 & 1.166e\text{-}3 \\ 4.753e\text{-}1 & 1.064e\text{-}4 & 4.602e{+}0 \end{bmatrix} \quad (5\text{-}32)$$

Figure 5-9 shows the resulting $\Pi_{Best\ GI}$ inside Φ. The volume of $\Pi_{Best\ GI}$ = 3.852e-2 which is 42.7% of Φ. For problems such as this, in which m is much greater than n, the volume of $\Pi_{Best\ GI}$ is significantly less than the volume of Φ. This smaller volume can translate directly to a loss of performance capability.

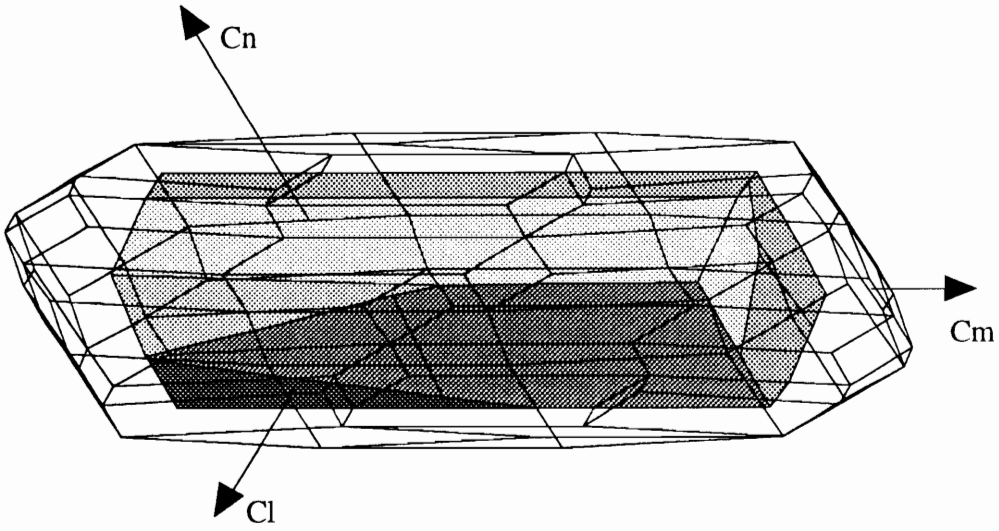


Figure 5-9: $\Pi_{Best\ GI}$ inside Φ

Unattainable Moments

As stated in the introduction, many control laws are configured so that they may command moments for which the control allocator cannot allocate admissible controls. If the desired moment lies outside of Π_{GI}, then the moment produced by the commanded controls can vary widely from the desired moment. The variation in the moment produced is inconsistent and thus difficult to predict. When the allocated controls are truncated by position saturation, the moment produced by the controls may not have the same direction. Furthermore, the moment produced may be of greater or lesser magnitude than the desired moment. Also, desired moments which lie in the same direction in moment space need not produce moments which lie in the same direction in moment space.

The effects of commanding moments which lie outside of Π_{GI} can be demonstrated using the F-18 HARV data and the pseudo-inverse, Equations 5-30 and 5-31. If the desired moment is:

$$\mathbf{m}_d = \begin{Bmatrix} 0.08 \\ 0.08 \\ 0.08 \end{Bmatrix} \in \Phi \tag{5-33}$$

then the pseudo-inverse will command the controls to be:

$$\mathbf{u} = \{-0.2630,\ 0.1430,-0.2265,\ 0.2119,-0.1396,$$

$$\mathbf{-0.2403},\ 0.2543,\ 0.1158,\ 0.0401,\ 0.4502\}^T \tag{5-34}$$

The right trailing edge flap, u_6, is commanded to exceed its limit. Excessive commands sent to actuators are truncated by the physical limits. To denote that some of

the controls are saturated, the subscript (sat) will be used, i.e. $\mathbf{u}_{(sat)}$. After the position limits are enforced, the controls are:

$$\mathbf{u}_{(sat)} = \{-0.2630, 0.1430, -0.2265, 0.2119, -0.1396,$$

$$\mathbf{-0.1396}, 0.2543, 0.1158, 0.0401, 0.4502\}^T \quad (5\text{-}35)$$

The moment produced by the truncated controls is:

$$\mathbf{m}_{out} = \begin{Bmatrix} 0.07368 \\ 0.08628 \\ 0.08000 \end{Bmatrix} \quad (5\text{-}36)$$

This moment has a different direction and a greater magnitude than the desired moment.

$$|\mathbf{m}_d| = 0.13856 < |\mathbf{m}_{out}| = 0.13883 \quad (5\text{-}37)$$

If the control vector is scaled so that controls which are commanded to move beyond their limits are reduced to be equal to their limits, then the direction of the desired moment can be preserved. This scaling would need to be implemented as part of the control allocation algorithm. For the desired moment in Equation 5-32, the controls from the pseudo-inverse given in Equation 5-33 can be scaled so that u_6 is at its limit:

$$\mathbf{u}_{scaled} = \{-0.1528, 0.0831, -0.1316, 0.1231, -0.0811,$$

$$\mathbf{-0.1396}, 0.1478, 0.0673, 0.0233, 0.2615\}^T \quad (5\text{-}38)$$

For these commanded controls, the direction of the desired moment is preserved:

$$\mathbf{m}_{out} = \begin{Bmatrix} 0.0465 \\ 0.0465 \\ 0.0465 \end{Bmatrix} = 0.581 * \mathbf{m}_d = \begin{Bmatrix} 0.08 \\ 0.08 \\ 0.08 \end{Bmatrix} \quad (5\text{-}39)$$

A second desired moment in the same direction as the first is considered.

$$\mathbf{m}_d = \begin{Bmatrix} 0.10 \\ 0.10 \\ 0.10 \end{Bmatrix} \in \Phi \tag{5-40}$$

For this desired moment, the pseudo-inverse allocated controls are:

$$\mathbf{u} = \{ -0.3287, 0.1787, -0.2831, 0.2649, -0.1744,$$

$$\mathbf{-0.3003}, 0.3179, 0.1448, 0.0501, \mathbf{0.5627}\}^T \tag{5-41}$$

The right trailing edge flap, u_6, and the yaw thrust vector vane, u_{10}, are commanded to exceed their limits. The truncated controls are:

$$\mathbf{u}_{(sat)} = \{ -0.3287, 0.1787, -0.2831, 0.2649, -0.1744,$$

$$\mathbf{-0.1396}, 0.3179, 0.1448, 0.0501, \mathbf{0.5236}\}^T \tag{5-42}$$

The moment produced by these controls is:

$$\mathbf{m}_{out} = \begin{Bmatrix} 0.08951 \\ 0.11002 \\ 0.09419 \end{Bmatrix} \tag{5-43}$$

This moment has a different direction and smaller magnitude than the desired moment.

$$|\mathbf{m}_d| = 0.17321 > |\mathbf{m}_{out}| = 0.17026 \tag{5-44}$$

Even though the two desired moment in this example have the same direction, the moments produced have different directions. While the subject of what is best to do when a commanded moment is unattainable is debatable, producing a moment which

varies inconsistently from the desired moment can have undesirable effects on aircraft performance.

6. DAISY CHAINING SOLUTIONS

Method Origin

Some controls can put severe stresses on aircraft components. For example, thrust vectoring vanes are exposed to high temperatures when they are employed. It was proposed that using thrust vectoring only when necessary would help reduce life-cycle costs. [14] The method of daisy chaining allows a specified set of controls, such as thrust vectoring vanes, to be used only when the rest of the controls fail to achieve the desired moment. "This results in a savings (in terms of heat loads) on the thrust vectoring vanes." [14]

The method of daisy chaining usually divides the controls into two groups [6, 14], $\mathbf{u}_1$ and $\mathbf{u}_2$. The controls in $\mathbf{u}_2$ are used only when the controls in $\mathbf{u}_1$ fail to generate the desired moment. This method can be applied using 3 or more control groupings, with each successive grouping being used only when the previous groupings fail to achieve the desired moment. However, the typical case uses two groupings, and only this case will be discussed in detail.

Description of Method

Initially, the controls are divided into two groups. The first group, $\mathbf{u}_1$, consists of the controls which may be used at all times. These controls are often the conventional aerodynamic surfaces such as ailerons, elevators, and rudders. The second group, $\mathbf{u}_2$, contains the controls which will be used only when the first group of controls fails to

generate the desired moment. These controls are often those which move slower or are subjected to higher stresses such as flaps and thrust vector vanes.

The B matrix is partitioned to correspond to these control groupings:

$$B\mathbf{u} = [B_1\ B_2]\begin{Bmatrix}\mathbf{u}_1\\ \mathbf{u}_2\end{Bmatrix} = B_1\mathbf{u}_1 + B_2\mathbf{u}_2 \tag{6-1}$$

For a desired moment, the controls in a given group, $\mathbf{u}_i$, are allocated using matrix multiplication. If the inverse of B_i exists, it is computed and used to allocate the controls in $\mathbf{u}_i$. If B_i is not square, a generalized inverse of B_i is generally used. To simplify the notation, P_i will be used to represent the matrix used to allocate the controls in $\mathbf{u}_i$, whether it is some generalized inverse of B_i or B_i^{-1}. Initially, the controls are allocated as follows:

$$\mathbf{u}_1 = P_1\mathbf{m}_d \text{ and } \mathbf{u}_2 = \mathbf{0} \tag{6-2}$$

If the controls in $\mathbf{u}_1$ are at or within their limits, the controls in $\mathbf{u}_2$ remain unused. If some or all of the controls in $\mathbf{u}_1$ are commanded to exceed their limits, then the controls which are commanded to exceed their limits are positioned at their limits. The truncated controls are designated $\mathbf{u}_{1(sat)}$. Because the commanded controls have been truncated, they will not produce the desired moment. The moment produced by the controls in group one, $B_1\mathbf{u}_{1(sat)}$, is then subtracted from the desired moment to get the part of the desired moment which needs to be produced by the controls in $\mathbf{u}_2$. These controls are then allocated using P_2.

$$\mathbf{u}_2 = P_2(\mathbf{m}_d - B_1\mathbf{u}_{1(sat)}) \tag{6-3}$$

If the controls in $\mathbf{u}_2$ are commanded to exceed their limits, they are positioned at their limits. This will only occur if the desired moment is unattainable using this method. In general, this method will not allocate controls in Ω to generate every desired moment in Φ.

Figure 6-1, taken from Reference 14, shows a block diagram of the daisy-chaining algorithm. The "Saturation" blocks typically represent both position and rate limits. The subject of rate limiting will be discussed in later sections. The problem being considered will deal only with the position limits.

Calculating Π for a given Daisy Chain

Calculating Π for this method is more difficult than for the generalized inverse solutions. The moments produced by the individual control groupings can be computed using previously described techniques. However, because of the manner in which the controls are allocated, the effects of the different control groupings can not be combined in a simple additive fashion.

Consider a group of controls, $\mathbf{u}_i$. Corresponding to the controls in $\mathbf{u}_i$, will be an Ω_i, the admissible controls, and Φ_i, the attainable moments to which the controls in Ω_i map.

$$\Omega_i \subset \Omega \text{ and } \Phi_i \subset \Phi \tag{6-4}$$

The matrix P_i will define a set of moments for which admissible controls will be allocated. If B_i has more columns than rows, Π_i will be a subset of Φ_i and can be computed as described in Section 5. If B_i is square and invertible then there is a unique relationship between the moments in Φ_i and the controls in Ω_i. This means that Π_i will be equal to Φ_i.

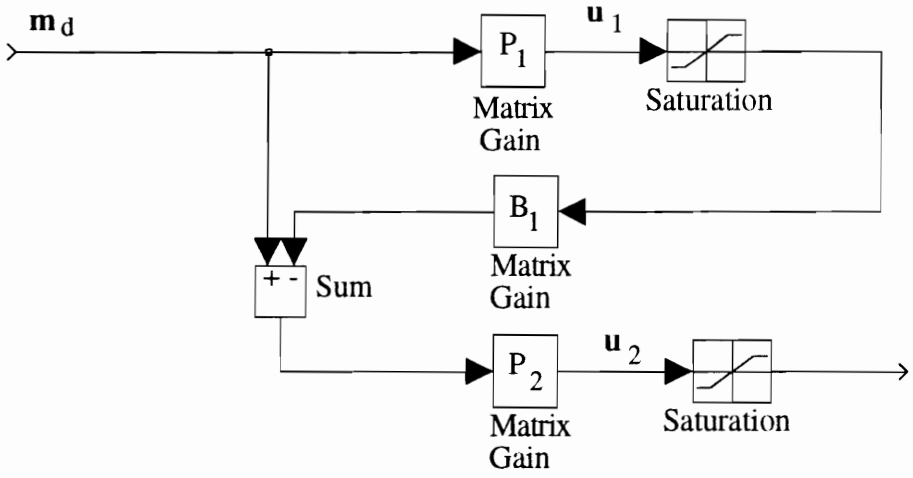


Figure 6-1: Block Diagram of Daisy Chaining Algorithm

The case where B_i has more rows than columns has not been previously discussed because it is not a redundant control problem. However, daisy chaining may divide the B matrix into sections which have fewer columns than rows. When this occurs, the controls, Ω_i, define a subset in moment space, Φ_i, which is of lower dimension than n. Each moment in Φ_i can be mapped to controls in Ω_i using a matrix multiplication. Thus, Π_i will be equal to Φ_i. The matrix used to map Φ_i to Ω_i can be computed as the least-squares solution to $\mathbf{m} = B\mathbf{u}$. This matrix is sometimes known as the left pseudo-inverse, and it satisfies $PB = I_m$. It can be derived from minimization principles as follows:

Minimize $\|\mathbf{m}-B\mathbf{u}\|_2$

Define the scalar function f($\mathbf{u}$):

$$f(\mathbf{u}) = (\mathbf{m} - B\mathbf{u})^T(\mathbf{m} - B\mathbf{u}) = (\mathbf{m}^T - \mathbf{u}^T B^T)(\mathbf{m} - B\mathbf{u}) \tag{6-5}$$

$$f(\mathbf{u}) = \mathbf{m}^T\mathbf{m} - \mathbf{m}^T B\mathbf{u} - \mathbf{u}^T B^T\mathbf{m} + \mathbf{u}^T B^T B\mathbf{u} \tag{6-6}$$

Since is $\mathbf{u}^T B^T\mathbf{m}$ a scalar, it is equal to its transpose:

$$\mathbf{u}^T B^T\mathbf{m} = \mathbf{m}^T B\mathbf{u} \tag{6-7}$$

Rewriting Equation 6-6:

$$f(\mathbf{u}) = \mathbf{m}^T\mathbf{m} - 2\mathbf{m}^T B\mathbf{u} + \mathbf{u}^T B^T B\mathbf{u} \tag{6-8}$$

f($\mathbf{u}$) is an extremum when $\frac{\partial f(\mathbf{u})}{\partial \mathbf{u}} = 0$

$$\frac{\partial f(\mathbf{u})}{\partial \mathbf{u}} = -2\mathbf{m}^T B + 2\mathbf{u}^T B^T B = 0 \Leftrightarrow 2\mathbf{u}^T B^T B = 2\mathbf{m}^T B \Leftrightarrow B^T B\mathbf{u} = B^T\mathbf{m} \tag{6-9}$$

$$\mathbf{u} = (B^T B)^{-1} B^T\mathbf{m} \tag{6-10}$$

$$P_{pseudo} = (B^T B)^{-1} B^T \tag{6-11}$$

This solution minimizes $\|\mathbf{m}-B(P\mathbf{m})\|_2$. This matrix can be calculated using several other methods. Reference 24 suggests using QR factorization or singular value decomposition to calculate P_{pseudo}, especially when B is rank deficient. For all of the moments in Φ_1, $P_{1\,pseudo}$ will allocate admissible controls, and $\|\mathbf{m}-B(P\mathbf{m})\|_2$ will be zero.

The admissible controls and attainable moments which correspond to the different control groups can be combined to form the total admissible controls and total attainable moments. This can be done because the problem is linear. For the typical case of 2 control groupings, this combination can be written as the Cartesian product [23] of the two sets:

$$\Omega_1 \times \Omega_2 = \Omega \text{ and } \Phi_1 \times \Phi_2 = \Phi \tag{6-12}$$

However, the Cartesian product of Π_1 and Π_2 is not equal to the total volume of moments for which the scheme will allocate admissible controls, Π_{DC}.

$$\Pi_1 \times \Pi_2 \neq \Pi_{DC} \tag{6-13}$$

This point is best illustrated using a low order example.

Example 6-1: 4 Controls & 2 Moments

$$B = \begin{bmatrix} -0.5864 & 0.6620 & 0.3992 & -0.4406 \\ 0.9171 & -0.4438 & 0.2091 & 0.2237 \end{bmatrix} \tag{6-14}$$

$$B_1 = \begin{bmatrix} -0.5864 & 0.6620 \\ 0.9171 & -0.4438 \end{bmatrix}, B_2 = \begin{bmatrix} 0.3992 & -0.4406 \\ 0.2091 & 0.2237 \end{bmatrix} \tag{6-15}$$

$$P_1 = \begin{bmatrix} 1.2793 & 1.9084 \\ 2.6438 & 1.6905 \end{bmatrix}, P_2 = \begin{bmatrix} 1.2329 & 2.4288 \\ -1.1525 & 2.2006 \end{bmatrix} \tag{6-16}$$

$$-1 \le u_i \le 1,\ \mathbf{u}_1 = \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix},\ \mathbf{u}_2 = \begin{Bmatrix} u_3 \\ u_4 \end{Bmatrix} \tag{6-17}$$

Because B_1 and B_2 are square and invertible, $\Pi_1 = \Phi_1$ and $\Pi_2 = \Phi_2$. Figure 6-2 shows a wire-frame of Φ for this problem. In Figure 6-2, Π_1 is shown as the shaded region centered at the origin of moment space. The shaded area centered at (-2, -1.5) corresponds to an area the size and shape of Π_2.

As the controls in $\mathbf{u}_1$ vary, the origin of Π_2 moves to the moment produced by the controls in $\mathbf{u}_1$. Figure 6-3 shows that by sweeping the origin of Π_2 around the boundary of Π_1, the entire area of Φ can be attained. While it is always true that Φ_1 and Φ_2 can be combined to get Φ, Π_1 and Π_2 can be combined to get Φ only if Π_1 is equal to Φ_1 and Π_2 is equal to Φ_2.

Daisy chaining does not allocate the controls in a fashion where these two areas can simply be combined to get Π_{DC}. Figure 6-4 shows Φ and desired moments which are on the boundary of Φ, represented by circles. The moments produced using daisy chaining to allocate controls for these moments are indicated by +'s. The fact that the +'s and o's are not coincident illustrates the fact that the moments produced are not equal to the desired moments. The lines shown in Figure 6-4 connect a desired moment and the resulting moment produced.

The area formed by the produced moments in Figure 6-4 is not equal to Φ and is not convex. The reason for this can been seen by examining the commanded control deflections for several moments. The control deflections corresponding to the commanded moments shown in Figure 6-5 are given in Table 6-1. Saturated controls are indicated in bold.

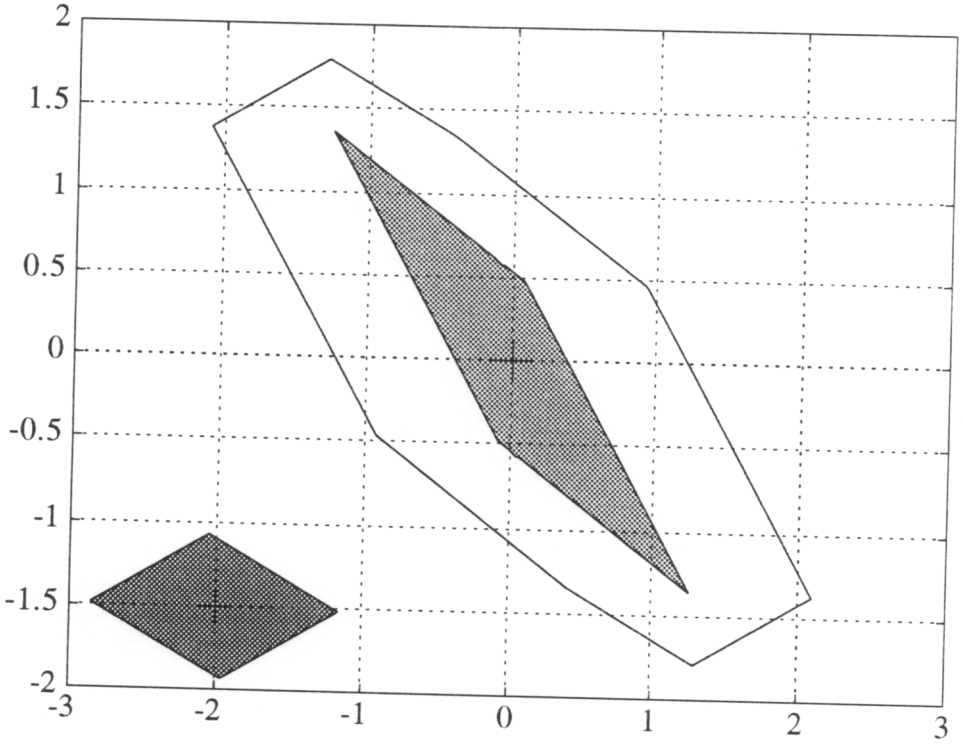


Figure 6-2: Φ, Π_1, and Π_2 for Example 6-1

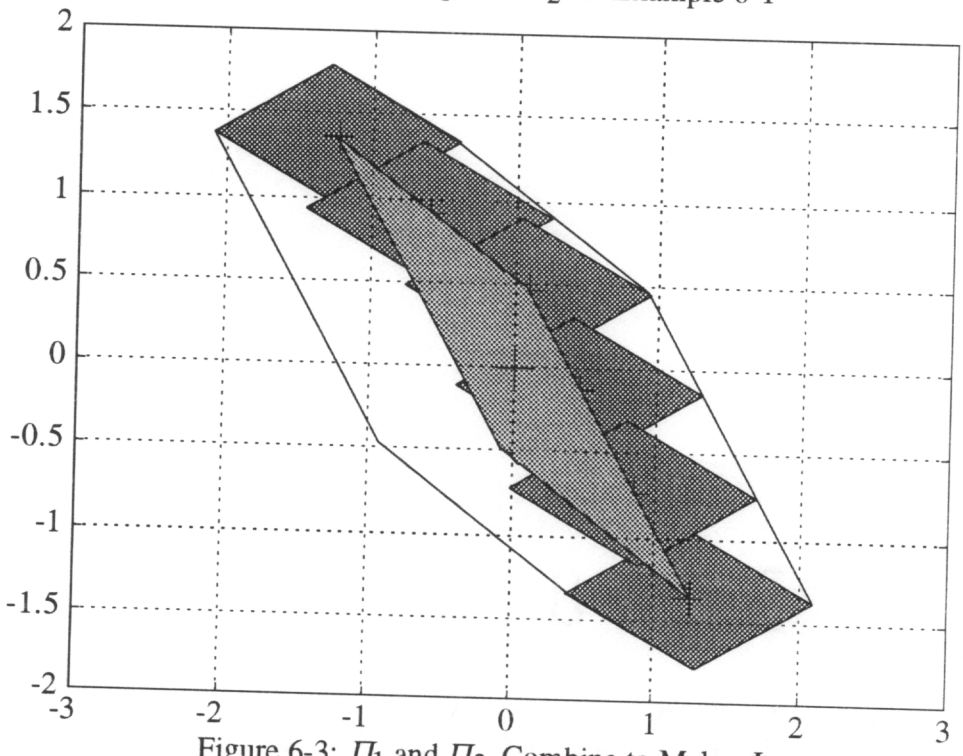


Figure 6-3: Π_1 and Π_2 Combine to Make Φ

At Point 1 in Figure 6-5, the first two controls are set to their minimum values. This produces a moment which is a vertex of Π_1, designated point 'a'. Point 'a' becomes the origin for Π_2, as shown in Figure 6-5. From point 'a', Point 1 almost lies inside Π_2, and the moment produced is very close to the desired moment. However, for Point 2, the controls in $\mathbf{u}_1$ also produce the moment at point 'a'. When Π_2 is centered at point 'a', Point 2 is well outside of Π_2. Thus, the moment produced can not be equal to the desired moment.

For points 3 and 4, the origin of Π_2 moves away from point 'a' along an edge of Π_1 towards point 'b'. The movement away from point 'a' is seen by the fact that only one control in $\mathbf{u}_1$ is saturated. However, the origin does not move nearly far enough for points 3 and 4 to lie inside the relocated Π_2, and the moments produced are not close to the desired moments.

Due to the nature of the daisy chaining allocation scheme, it is difficult to exactly determine the volume of $\Pi_{Daisy\ Chaining}$ for a given problem. Instead, $\Pi_{Daisy\ Chaining}$ can be approximated using a convex hull generating program and a set of moments for which daisy chaining allocates admissible controls.

To obtain this set of moments, it is necessary to pick a moment direction and increase the magnitude of the vector until one of the controls in $\mathbf{u}_2$ saturates. It is not sufficient to pick a large moment and the use the moment produced when that moment is commanded. There is no guarantee that the moment produced by a large commanded moment will be produced when it is commanded. For example, when the commanded

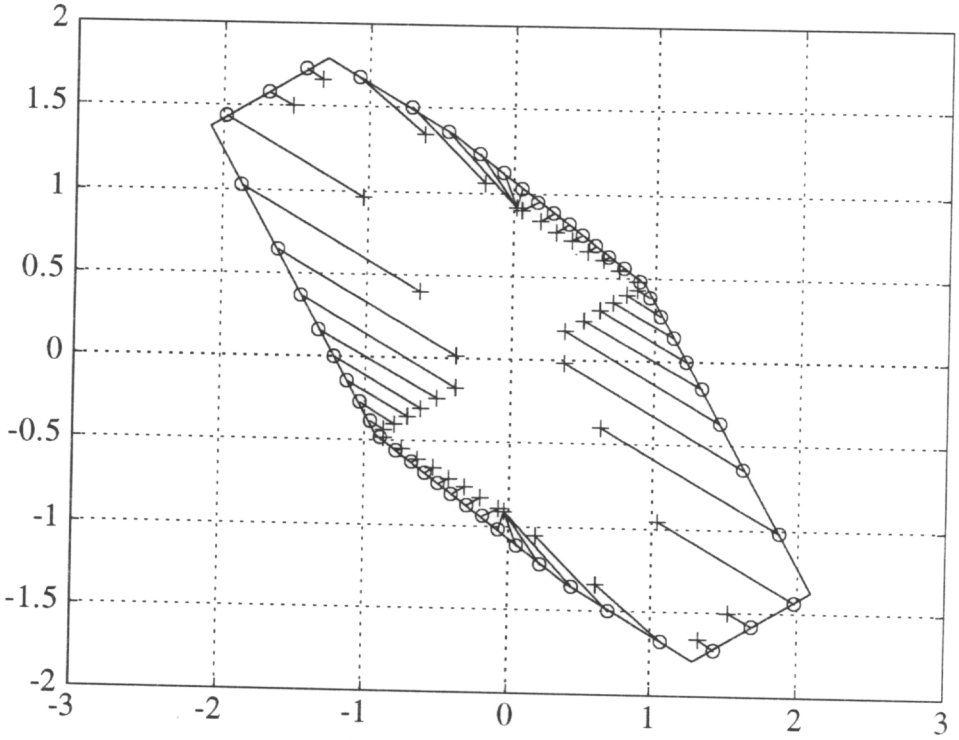


Figure 6-4: Desired and Produced Moments

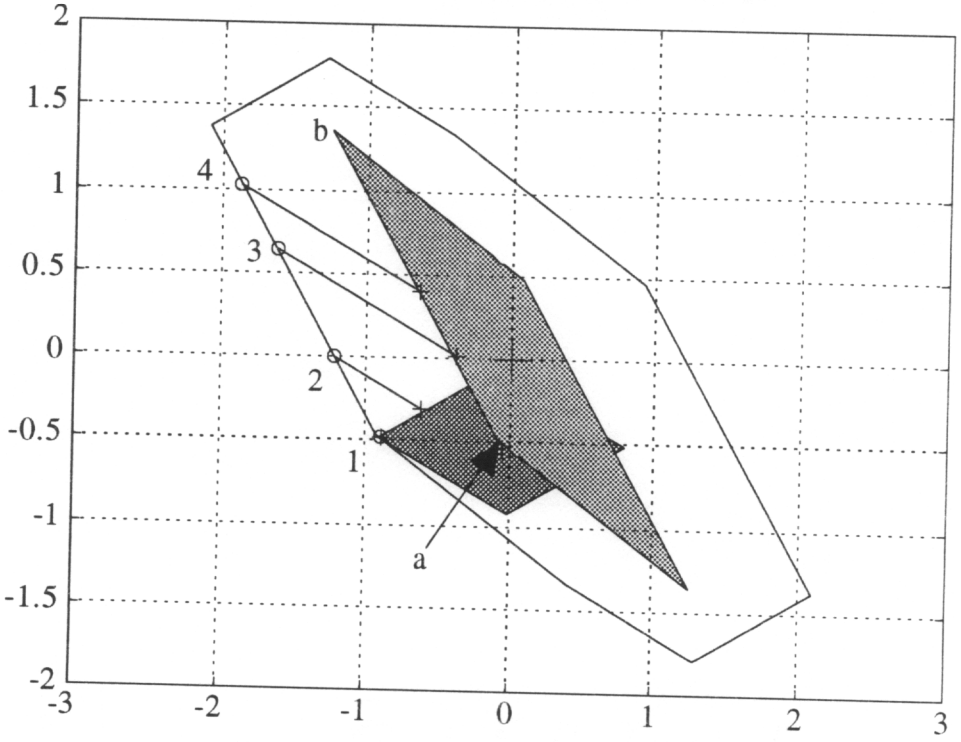


Figure 6-5: Examining the Concavity

Table 6-1: Controls Allocated using Daisy Chaining

	$\mathbf{u}_1$		$\mathbf{u}_2$		Desired Moment	Moment Produced
	u_1	u_2	u_3	u_4		
1	**-1.0000**	**-1.0000**	**-1.0000**	0.9041	(-0.88,-0.48)	(-0.87,-0.48)
2	**-1.0000**	**-1.0000**	-0.2474	**1.0000**	(-1.21, 0.00)	(-0.62,-0.30)
3	-0.8476	**-1.0000**	0.5745	**1.0000**	(-1.62, 0.64)	(-0.38, 0.01)
4	-0.4289	**-1.0000**	0.5745	**1.0000**	(-1.86, 1.02)	(-0.62, 0.39)

moment is (1.0662, -1.6801) the moment produced by the allocated controls is (0.6137, -1.3404). However, when (0.6137, -1.3404) is commanded, the moment produced is (0.2019, -1.0643).

Figure 6-6 shows Φ and a set of moments, represented by circles, which were generated by picking a direction and increasing the magnitude until one control in $\mathbf{u}_2$ saturated. The dashed line represents a convex hull generated around these moments. This convex hull has an area approximately equal to the area of Π_{DC}. This estimate is overly generous due to the concavities in Π_{DC}. However, it is easier to compute, especially for higher order problems, than an actual volume of Π_{DC} which may be a very complicated, concave shape.

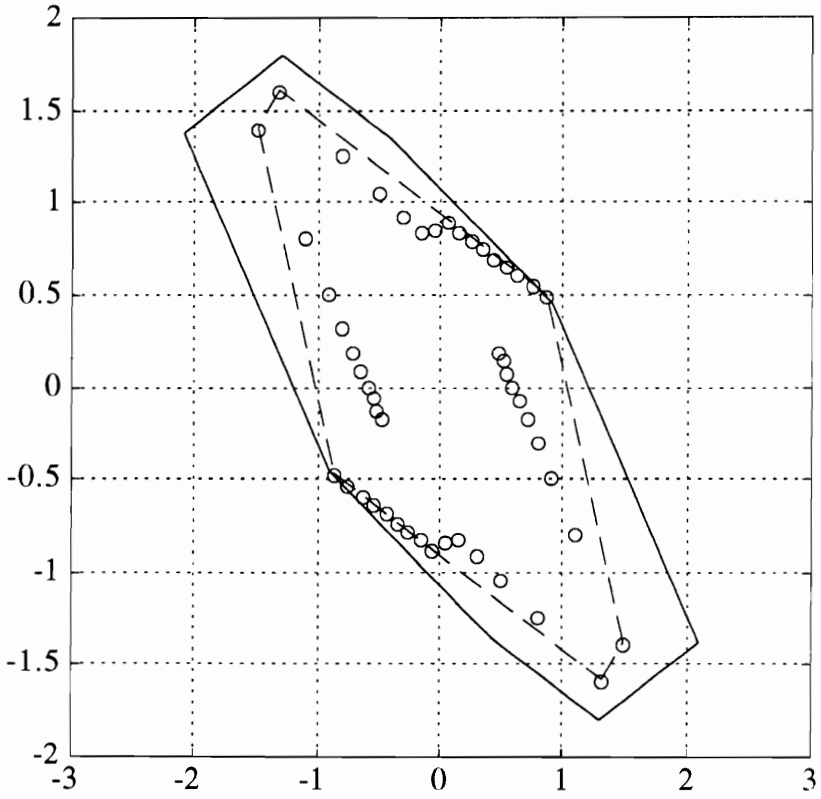


Figure 6-6: Maximum Attainable Moments and Approximate Π_{DC}

Example 6-2: F-18 HARV

Using the F-18 HARV data, the B matrix was partitioned into two 3x5 matrices. B_1 contains the more conventional aircraft controls: horizontal tails, ailerons, and rudders. B_2 contains the remaining controls: trailing edge flaps, and thrust vectoring vanes.

$$B_1 = \begin{bmatrix} -4.382e\text{-}2 & 4.382e\text{-}2 & -5.841e\text{-}2 & 5.841e\text{-}2 & 1.674e\text{-}2 \\ -5.330e\text{-}1 & -5.330e\text{-}1 & -6.486e\text{-}2 & -6.486e\text{-}2 & 0.000e+0 \\ 1.100e\text{-}2 & -1.100e\text{-}2 & 3.911e\text{-}3 & -3.911e\text{-}3 & -7.428e\text{-}2 \end{bmatrix} \tag{6-18}$$

$$B_2 = \begin{bmatrix} -6.280e\text{-}2 & 6.280e\text{-}2 & 2.920e\text{-}2 & 1.000e\text{-}5 & 1.000e\text{-}2 \\ 6.234e\text{-}2 & 6.234e\text{-}2 & 1.000e\text{-}5 & 3.553e\text{-}1 & 1.000e\text{-}5 \\ 0.000e+0 & 0.000e+0 & 3.000e\text{-}4 & 1.000e\text{-}5 & 1.485e\text{-}1 \end{bmatrix} \tag{6-19}$$

The matrices were inverted using the pseudo-inverse, producing P_1 and P_2:

$$P_1 = \begin{bmatrix} -4.167e+0 & -9.243e\text{-}1 & -2.429e+0 \\ 4.167e+0 & -9.243e\text{-}1 & 2.429e+0 \\ -5.337e+0 & -1.124e\text{-}1 & -3.254e\text{-}3 \\ 5.337e+0 & -1.124e\text{-}1 & 3.254e\text{-}3 \\ 6.722e\text{-}1 & 0.000e+0 & -1.274e+1 \end{bmatrix} \tag{6-20}$$

$$P_2 = \begin{bmatrix} -7.186e+0 & 4.653e\text{-}1 & 4.866e\text{-}1 \\ 7.185e+0 & 4.649e\text{-}1 & -4.868e\text{-}1 \\ 3.338e+0 & -1.567e\text{-}5 & -2.125e\text{-}1 \\ -2.637e\text{-}5 & 2.651e+0 & -1.508e\text{-}4 \\ -6.745e\text{-}3 & -1.784e\text{-}4 & 6.734e+0 \end{bmatrix} \tag{6-21}$$

To approximate the volume of Π_{DC}, 182 moments directions were chosen. These directions correspond to the directions of the 92 vertices which define Φ, and the centers of the 90 faces which bound Φ. The vertices of Φ are special cases because the maximum attainable moment in these directions have all the controls set at saturation. Also, the vertices are the extreme points of Φ which define its shape. The centers of the faces are special because they are as far as possible from the vertices. Thus, these directions provide a set that will rigorously test the abilities of the allocation scheme. For

each direction, the desired moment was increased in increments of 0.1% of the value of the maximum attainable moment in that direction until one or more of the controls in $\mathbf{u}_2$ saturated. The daisy chaining scheme implemented was unable to achieve the maximum attainable moment in any of the directions tested. The best it was able to do was achieve a moment 96.7% of the maximum value. In the worst directions tested, the moment attainable using this daisy chaining scheme was only 38.2% of the maximum value. On average, the moments in Π_{DC} were 75.5% of the maximum values.

Figure 6-7 shows an approximation of Π for this daisy chaining scheme inside a wire frame of the AMS. This Π was computed using the quickhull algorithm [26] for generating convex hulls. The quickhull algorithm requires a user input "delta" value to determined the accuracy of the convex hull. This number should be reduced until further reduction does not change the resulting convex hull. The approximate volume of Π_{DC} is 3.811e-2, which is about 42.3% of Φ.

It should be noted that using different control groupings and different generalized inverses can have a strong effect on the volume of Π_{DC}. However, for several reasons, the volume for Π_{DC} is not typically maximized. First of all, finding the volume of Π_{DC} for a given setup is very computationally expensive. Secondly, the control groupings are usually chosen to fit specific design criterion, and little or no variation is allowed. Finally, as Example 6-1 shows, even when all the volumes of the individual Π_i's are maximized, $\Pi_i = \Phi_i$, Π_{DC} may not be equal to Φ. If maximizing the volume of Π is of the highest priority, another allocation method should be used.

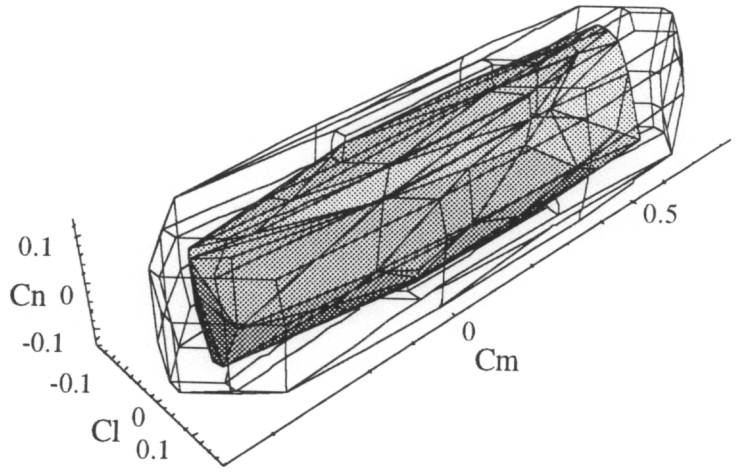


Figure 6-7: Approximate Π_{DC} for F-18 HARV

Unattainable Moments

Similar to the case of generalized inverse solutions, when a moment is commanded that is outside Π_{DC}, the moment produced can vary widely from the desired moment. When over-driven controls are truncated by saturation, the moment produced has neither the same magnitude nor direction as the desired moment. This fact can be seen from Figure 6-4 and Table 6-1. Section 5 showed that the controls can be scaled so that the direction of the desired moment is preserved. Scaling down over-saturated control commands is not usually done when implementing daisy-chaining [6, 14], but it is considered here to illustrate some of the effects.

Preserving the direction of the commanded moment does not enable a daisy-chaining algorithm to access all of the attainable moments. This is illustrated in Figures 6-8 and 6-9. The circle in Figure 6-8 shows Point 3 from Table 6-1 of Example 6-1. If Point 3 represents the desired moment, the scaled controls will produce the moment which is located where the line from the origin to Point 3 intersects the boundary of Π_1. This becomes the center of Π_2. With Π_2 centered here, Point 3 is still outside of Π_2, and thus unattainable. Figure 6-9 shows a location of the origin of Π_2 from which Point 3 is attainable. This point cannot be calculated by scaling. If the controls allocated using daisy-chaining are properly scaled, then the direction of the desired moment can be preserved. However, Daisy Chaining will not generally allocate admissible controls for all the attainable moments whether or not scaling is used.

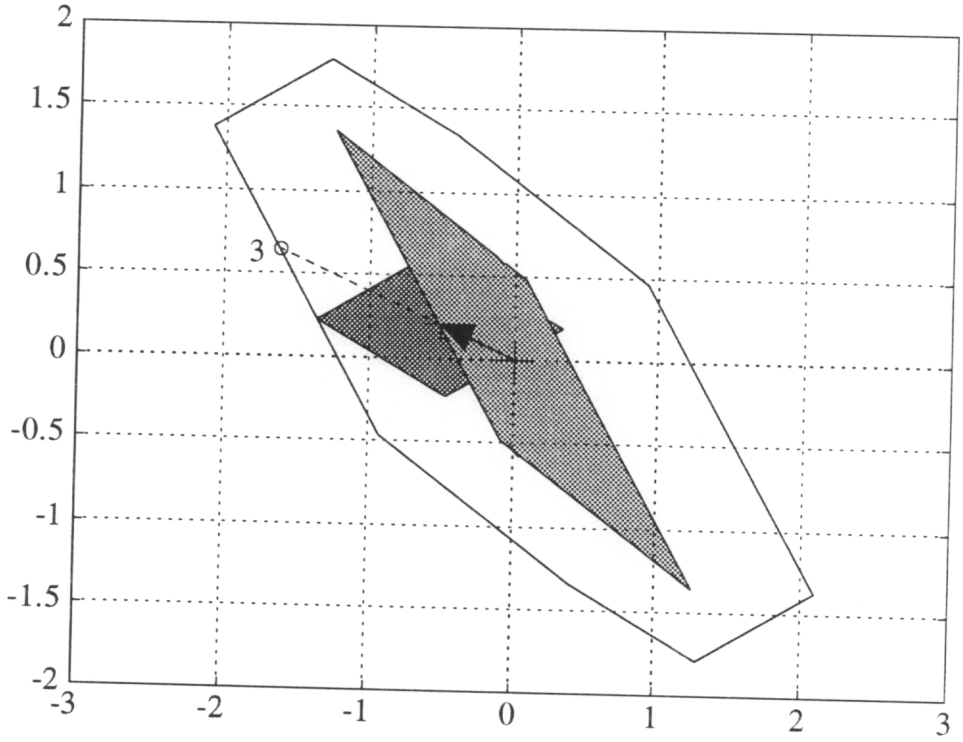


Figure 6-8: Using the Scaled down Controls

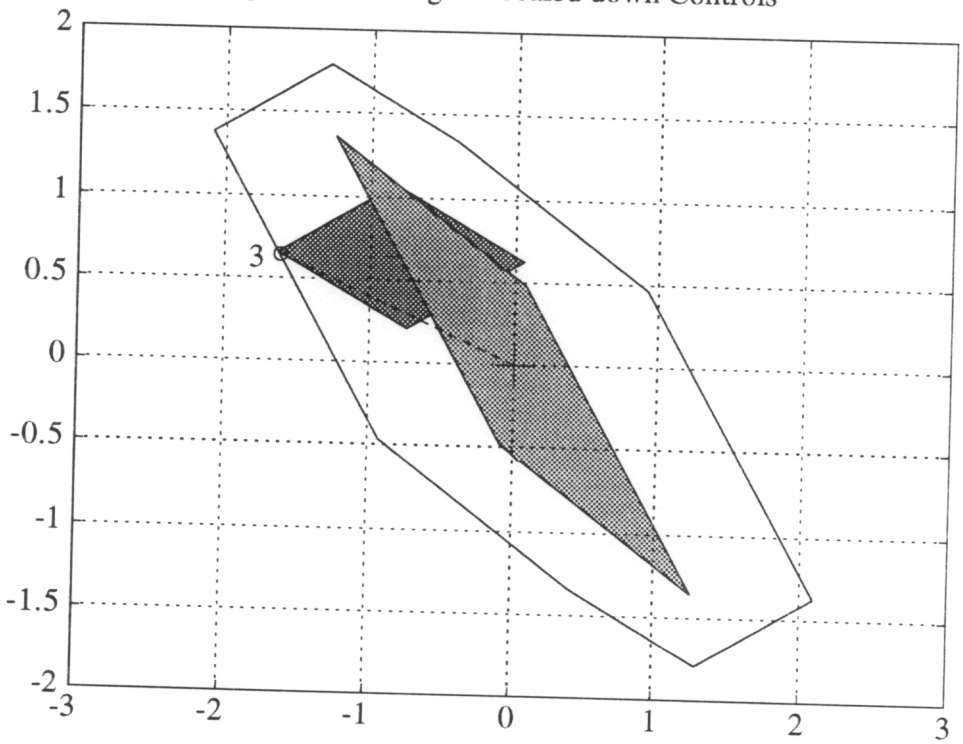


Figure 6-9: Required Origin for Π_2 to Attain Point 3

7. CASCADING GENERALIZED INVERSE SOLUTIONS

Description of Method

Since the volume of moments which are attainable using a single generalized inverse, Π_{GI}, can be a small percentage of the total volume of attainable moments, Φ, new methods which have greater volumes have been explored. The method of Cascading Generalized Inverses arises from the idea that if a generalized inverse commands a control to exceed a position limit, then that control should be set at the exceeded limit, and the rest of the controls redistributed to achieve the desired moment. The effect of this redistribution effort is to enlarge the region in moment space for which the scheme will allocate admissible controls. This procedure can be used with either pseudo-inverses, or generalized inverses weighted with a diagonal N matrix.

Initially, a generalized inverse is computed using either Equation 5-12 or 5-14. This matrix is used to allocate the controls given in response to some desired moment.

$$\mathbf{u} = P\mathbf{m}_d \tag{7-1}$$

If this solution yields no saturated controls, then the desired moment lies within Π for the generalized inverse. If any controls are commanded to exceed their limits, then those controls are set to their limits, and their effects at saturation are subtracted from the desired moment. The effect of a saturated control is equivalent to the control position multiplied by the column of the B matrix which corresponds to that control. The resulting moment is the part of the moment demand that must be satisfied by the remaining controls. It is denoted $\mathbf{m}_d$'. For example, if the i-th control saturates:

$$u_i = u_{i(sat)} \ , \ \mathbf{m}_d' = \mathbf{m}_d - B_i u_{i(sat)} \tag{7-2}$$

Next, the saturated controls are removed from the problem. If a pseudo-inverse is being used, this is done by removing the corresponding columns, B_i, from B. The smaller B matrix is denoted B'. The new pseudo-inverse is computed by plugging B' into Equation 5-12 to get P'. When using a diagonal N matrix, it is sufficient to put a zero in the (i,i) element of N, then calculate P' using Equation 5-14 and B. The controls which have not been saturated are then calculated using P'.

$$\mathbf{u}' = P' \mathbf{m}_d' \tag{7-3}$$

If P' is computed from Equation 5-12, $\mathbf{u}'$ will have fewer controls than $\mathbf{u}$. It is important to keep track of which controls are in $\mathbf{u}'$ and where they are located so that the final control vector can be assembled. This method is illustrated in Example 7-1 to follow. If P' is computed using Equation 5-14, then $\mathbf{u}'$ will have the same number of controls as $\mathbf{u}$, but the controls which have been saturated will be allocated zero deflection by P'. This method is illustrated in Example 7-2.

Some of the remaining controls determined in this step, $\mathbf{u}'$, may be commanded to exceed their position limits. If so, the process is repeated with P'', P''', etc.

$$u_j = u_{j(sat)} \ , \ \mathbf{m}_d'' = \mathbf{m}_d' - B_j u_{j(sat)} \tag{7-4}$$

$$\mathbf{u}'' = P'' \mathbf{m}_d'' \tag{7-5}$$

Ultimately either no new controls will be saturated, all the remaining controls will be saturated, or the reduced B will have n or fewer columns. When no new controls are saturated, an admissible solution is found. If all the controls are saturated, the controls

are set to their limits and the moment is unattainable using this method. If the reduced B is square, its inverse is taken, and the last n controls are determined.

When the removal of saturated controls results in a reduced B with fewer than n columns, several options are available. Equation 5-12 cannot be used when there are fewer than n columns in the reduced B matrix because the matrix inverse will not exist. Similarly, if the diagonal N matrix has fewer than n non-zero terms, Equation 5-14 cannot be used. One option is to remove only the number of controls which will leave n controls in the reduced B. When only these controls are removed, some of the saturated controls will be left in the problem and re-allocated. The saturated controls that are not removed can be chosen as those which correspond to the commanded controls that are closest to their limits. While this option is computationally viable, it has been observed that better results can be achieved by removing all the saturated controls and using a least-squares solution (Equation 6-11), as discussed in Section 6.

When a least-squares solution is used, the moment produced will be closer to the desired moment than when using the other method. If controls allocated by the least-squares solution saturate, they are removed, and the process is repeated, until none of the remaining controls saturate, or all of the remaining controls saturate. It is possible that the final solution will have unsaturated controls, but not produce the desired moment (to be shown in Example 7-2).

This method can greatly increase the size of Π for an initial generalized inverse, and it is not difficult to implement. However, as it may require several evaluations of a generalized inverse for each desired moment, it is more computationally expensive than using a single generalized inverse.

The method of Cascading Generalized Inverses is not guaranteed to produce admissible controls to generate all the attainable moments. Determining the volume of Π for this method can be done using the same techniques used to find $\Pi_{Daisy\ Chaining}$. Like Daisy Chaining, this method can have a concave Π, and so the convex hull method will yield an approximate volume which may be greater than the actual volume. Evidence of the concavities in Π is presented in Example 7-3.

Example 7-1: How the Method Works

This example uses the F-18 HARV data, the pseudo-inverse and the desired moment shown below. The pseudo-inverse produces the controls shown in Equation 7-9. Controls which have been commanded to exceed their limits are indicated by **bold** type.

$$B = \begin{bmatrix} -4.382e\text{-}2 & 4.382e\text{-}2 & -5.841e\text{-}2 & 5.841e\text{-}2 & 1.674e\text{-}2 \\ -5.330e\text{-}1 & -5.330e\text{-}1 & -6.486e\text{-}2 & -6.486e\text{-}2 & 0.000e{+}0 \\ 1.100e\text{-}2 & -1.100e\text{-}2 & 3.911e\text{-}3 & -3.911e\text{-}3 & -7.428e\text{-}2 \end{bmatrix} \ldots$$

$$\ldots \begin{bmatrix} -6.280e\text{-}2 & 6.280e\text{-}2 & 2.920e\text{-}2 & 1.000e\text{-}5 & 1.000e\text{-}2 \\ 6.234e\text{-}2 & 6.234e\text{-}2 & 1.000e\text{-}5 & 3.553e\text{-}1 & 1.000e\text{-}5 \\ 0.000e{+}0 & 0.000e{+}0 & 3.000e\text{-}4 & 1.000e\text{-}5 & 1.485e\text{-}1 \end{bmatrix} \quad (7\text{-}6)$$

$$P_{pseudo} = \begin{bmatrix} -2.201e{+}0 & -7.500e\text{-}1 & -3.350e\text{-}1 \\ 2.202e{+}0 & -7.500e\text{-}1 & 3.353e\text{-}1 \\ -2.960e{+}0 & -9.125e\text{-}2 & 2.210e\text{-}1 \\ 2.960e{+}0 & -9.128e\text{-}2 & -2.210e\text{-}1 \\ 9.492e\text{-}1 & 1.383e\text{-}5 & -2.693e{+}0 \\ -3.177e{+}0 & 8.774e\text{-}2 & 8.643e\text{-}2 \\ 3.177e{+}0 & 8.771e\text{-}2 & -8.646e\text{-}2 \\ 1.477e{+}0 & 6.081e\text{-}6 & -2.940e\text{-}2 \\ 3.958e\text{-}4 & 4.999e\text{-}1 & 2.5786e\text{-}4 \\ 3.015e\text{-}1 & -2.537e\text{-}5 & 5.325e{+}0 \end{bmatrix} \quad (7\text{-}7)$$

$$\mathbf{m}_d = \begin{Bmatrix} -0.1480 \\ 0.3167 \\ 0.1026 \end{Bmatrix} \in \Phi \quad (7\text{-}8)$$

$$\mathbf{u} = P_{pseudo}\mathbf{m}_d = \{0.0540, \mathbf{-0.5291}, 0.4320, -0.4898, -0.4168, 0.5070,$$

$$\mathbf{-0.4514}, -0.2216, 0.1583, 0.5016\}^T \quad (7\text{-}9)$$

The desired moment lies outside Π_{pseudo}. This is seen by the fact that the left horizontal tail, u_2, and the left trailing edge flap, u_7, are commanded to exceed their position limits. For this first example, the columns of B which correspond to the saturated controls will be removed to yield a reduced B matrix and Equation 5-12 will be used. The saturated controls are set to their limits and their columns are removed from B.

$$u_2 = u_{2\,Min} = -0.4189,\ u_7 = u_{7\,Min} = -0.1396 \quad (7\text{-}10)$$

$$B' = \begin{bmatrix} -4.382e\text{-}2 & -5.841e\text{-}2 & 5.841e\text{-}2 & 1.674e\text{-}2 & -6.280e\text{-}2 \\ -5.330e\text{-}1 & -6.486e\text{-}2 & -6.486e\text{-}2 & 0.000e+0 & 6.234e\text{-}2 \\ 1.100e\text{-}2 & 3.911e\text{-}3 & -3.911e\text{-}3 & -7.428e\text{-}2 & 0.000e+0 \end{bmatrix} \ldots$$

$$\ldots \begin{bmatrix} 2.920e\text{-}2 & 1.000e\text{-}5 & 1.000e\text{-}2 \\ 1.000e\text{-}5 & 3.553e\text{-}1 & 1.000e\text{-}5 \\ 3.000e\text{-}4 & 1.000e\text{-}5 & 1.485e\text{-}1 \end{bmatrix} \quad (5\text{-}11)$$

$$P' = \begin{bmatrix} -1.481e+0 & -1.191e+0 & -1.299e\text{-}1 \\ -4.255e+0 & 3.988e\text{-}2 & 1.749e\text{-}1 \\ 4.713e+0 & -3.688e\text{-}1 & -1.098e\text{-}1 \\ 1.286e+0 & -2.186e\text{-}2 & -2.687e+0 \\ -5.041e+0 & 3.798e\text{-}1 & -3.032e\text{-}2 \\ 2.241e+0 & -1.032e\text{-}1 & 1.040e\text{-}2 \\ -1.254e+0 & 9.008e\text{-}1 & -1.778e\text{-}1 \\ 7.654e\text{-}1 & -1.097e\text{-}1 & 5.372e+0 \end{bmatrix} \quad (7\text{-}12)$$

The new desired moment, $\mathbf{m}_d'$, is computed by subtracting out the contributions of these two controls.

$$\mathbf{m}_d' = \mathbf{m}_d - B_2 u_{2\,Min} - B_7 u_{7\,Min} = \begin{Bmatrix} -0.1209 \\ 0.1021 \\ 0.1072 \end{Bmatrix} \quad (7\text{-}13)$$

The remaining controls are then computed:

$$\mathbf{u}' = P'\mathbf{m}_d' = \{0.0435,\ \mathbf{0.5372},\ \mathbf{-0.6192},\ -0.4458,$$

$$0.6450,\ -0.2804,\ 0.2245,\ 0.4721\}^T \quad (7\text{-}14)$$

However, now the right aileron and the left aileron have been commanded to move beyond their limits. When using the reduced B matrix, it is important to keep track of which controls correspond to which elements of $\mathbf{u}'$. For example, the right aileron, the third element of $\mathbf{u}$, corresponds to the second element of $\mathbf{u}'$ because some controls have been removed. Again, these controls are set to their limits and their effects are subtracted from the desired moment.

$$u_3 = u_{3\,Max} = 0.5236,\ u_4 = u_{4\,Min} = -0.5236 \quad (7\text{-}15)$$

$$B'' = \begin{bmatrix} -4.382e\text{-}2 & 1.674e\text{-}2 & -6.280e\text{-}2 & 2.920e\text{-}2 & 1.000e\text{-}5 & 1.000e\text{-}2 \\ -5.330e\text{-}1 & 0.000e+0 & 6.234e\text{-}2 & 1.000e\text{-}5 & 3.553e\text{-}1 & 1.000e\text{-}5 \\ 1.100e\text{-}2 & -7.428e\text{-}2 & 0.000e+0 & 3.000e\text{-}4 & 1.000e\text{-}5 & 1.485e\text{-}1 \end{bmatrix} \quad (7\text{-}16)$$

$$P'' = \begin{bmatrix} -3.034e+0 & -1.143e+0 & -7.472e\text{-}2 \\ 2.909e+0 & -9.777e\text{-}2 & -2.738e+0 \\ -1.063e+1 & 6.479e\text{-}1 & 1.440e\text{-}1 \\ 4.726e+0 & -2.209e\text{-}1 & -6.735e\text{-}2 \\ -2.685e+0 & 9.857e\text{-}1 & -1.375e\text{-}1 \\ 1.221e+0 & -1.332e\text{-}1 & 5.358e+0 \end{bmatrix} \quad (7\text{-}17)$$

$$\mathbf{m}_d'' = \mathbf{m}_d' - B_3 u_{3\,Max} - B_4 u_{4\,Min} = \begin{Bmatrix} -0.0597 \\ 0.1021 \\ 0.1031 \end{Bmatrix} \quad (7\text{-}18)$$

The remaining controls are then computed:

$$\mathbf{u}'' = P''\mathbf{m}_d'' = \{0.0568,\ -0.4661,\ 0.7164,\ -0.3118,\ 0.2469,\ 0.4659\}^T \quad (7\text{-}19)$$

Because all of the controls in **u**'' are within their limits, an admissible solution is found. The final control vector, **u**, is assembled by combining the controls in **u**'' with the controls which have been saturated.

$$\mathbf{u} = \{0.0568, \mathbf{-0.4189}, \mathbf{0.5236}, \mathbf{-0.5236}, -0.4661, 0.7164, \mathbf{-0.1396}, -0.3118, 0.2469, 0.4659\}^T \quad (7\text{-}20)$$

Note that for these controls do in fact generate the desired moment, $B\mathbf{u} = \mathbf{m}_d$.

Example 7-2: $B\mathbf{u} \neq \mathbf{m}_d$

For the F-18 HARV data and a different desired moment,

$$\mathbf{m}_d = \begin{Bmatrix} 0.1083 \\ -0.0196 \\ -0.1234 \end{Bmatrix} \in \Phi \quad (7\text{-}21)$$

the pseudo-inverse allocates the following controls:

$$\mathbf{u} = \{-0.1825, \mathbf{0.2119}, -0.3462, 0.3498, 0.4352, \mathbf{-0.3566}, 0.3532, 0.1636, -0.0098, \mathbf{-0.6245}\}^T \quad (7\text{-}22)$$

The left horizontal tail, u_2, the right trailing edge flap, u_6, and the yaw thrust vector vane, u_{10}, are all commanded to move beyond their position limits. For this example, Equation 5-14 will be used. For the diagonal N matrix, an identity matrix, $N = I_{10}$, is used. This N is used so that the results will be the same as those for a pseudo-inverse, as used in Example 7-1. Removing the saturated controls from the problem yields:

$$u_2 = u_{2\,Max} = 0.1833,\ u_6 = u_{6\,Min} = -0.1396,\ u_{10} = u_{10\,Min} = -0.5236 \quad (7\text{-}23)$$

$$N(2,2) = 0,\ N(6,6) = 0,\ N(10,10) = 0 \quad (7\text{-}24)$$

$$P' = \begin{bmatrix} -8.998e\text{-}1 & -1.190e{+}0 & -9.002e\text{-}1 \\ 0.00e{+}0 & 0.00e{+}0 & 0.00e{+}0 \\ -4.553e{+}0 & 1.463e\text{-}1 & -4.330e\text{-}1 \\ 5.318e{+}0 & -5.102e\text{-}1 & 9.728e\text{-}1 \\ -3.766e\text{-}1 & 2.103e\text{-}1 & -1.340e{+}1 \\ 0.00e{+}0 & 0.00e{+}0 & 0.00e{+}0 \\ 5.042e{+}0 & -1.956e\text{-}1 & 1.278e{+}0 \\ 2.522e{+}0 & -1.734e\text{-}1 & 7.706e\text{-}1 \\ -2.094e{+}0 & 9.963e\text{-}1 & -1.476e{+}0 \\ 0.00e{+}0 & 0.00e{+}0 & 0.00e{+}0 \end{bmatrix} \tag{7-25}$$

$$\mathbf{m}_d' = \mathbf{m}_d - B_2 u_{2\,Max} - B_6 u_{6\,Min} - B_{10} u_{10\,Min} = \begin{Bmatrix} 0.0968 \\ 0.0868 \\ -0.0477 \end{Bmatrix} \tag{7-26}$$

$$\mathbf{u}' = P'\mathbf{m}_d' = \{-0.1475, 0.0000, -0.4072, 0.4240, \mathbf{0.6205},$$

$$0.0000, 0.4100, 0.1923, -0.0458, 0.0000\}^T \tag{7-27}$$

In $\mathbf{u}'$, the combined rudders, u_5, are saturated. Removing this control yields:

$$u_5 = u_{5\,Max} = 0.5236 \tag{7-28}$$

$$N(5,5) = 0 \tag{7-29}$$

$$P'' = \begin{bmatrix} -2.685e{+}0 & -1.932e\text{-}1 & -6.444e{+}1 \\ 0.00e{+}0 & 0.00e{+}0 & 0.00e{+}0 \\ -3.034e{+}0 & -7.016e\text{-}1 & 5.360e{+}1 \\ 4.740e{+}0 & -1.873e\text{-}1 & -1.959e{+}1 \\ 0.00e{+}0 & 0.00e{+}0 & 0.00e{+}0 \\ 0.00e{+}0 & 0.00e{+}0 & 0.00e{+}0 \\ 5.405e{+}0 & -3.985e\text{-}1 & 1.420e{+}1 \\ 3.040e{+}0 & -4.625e\text{-}1 & 1.918e{+}1 \\ -4.666e{+}0 & 2.432e{+}0 & -9.296e{+}1 \\ 0.00e{+}0 & 0.00e{+}0 & 0.00e{+}0 \end{bmatrix} \tag{7-30}$$

$$\mathbf{m}_d'' = \mathbf{m}_d' - B_5 u_{5\,Max} = \begin{Bmatrix} 0.0880 \\ 0.0868 \\ -0.0088 \end{Bmatrix} \tag{7-31}$$

$$\mathbf{u}'' = \{\mathbf{0.3122}, 0.0000, \mathbf{-0.7982}, \mathbf{0.5728}, 0.0000,$$

$$0.0000, 0.3165, 0.0591, \mathbf{0.6160}, 0.0000\}^T \quad (7\text{-}32)$$

In $\mathbf{u}''$, the right horizontal tail, u_1, the right aileron, u_3, the left aileron, u_4, and the pitch thrust vector vane, u_9, are all commanded to move beyond their position limits. Removing all these controls will result in fewer than n controls remaining ($n = 3$). As a result, the remaining controls are solved for using the P''' which is the least-squares solution, $P'''B''' = I_2$. Note that having this number of controls saturate does not necessarily mean that this moment is outside of Π_{CGI}. If the controls solved for using the least-squares solution are within their limits, it is possible that the moment is attainable using this method. Removing the saturated controls and solving for the remaining controls yields:

$$u_1 = u_{1\,Max} = 0.1833,\ u_3 = u_{3\,Min} = -0.5236$$

$$u_4 = u_{4\,Max} = 0.5236,\ u_9 = u_{9\,Max} = 0.5236 \quad (7\text{-}33)$$

$$B''' = \begin{bmatrix} 6.280\text{e-}2 & 2.920\text{e-}2 \\ 6.234\text{e-}2 & 1.000\text{e-}5 \\ 0.000\text{e+}0 & 3.000\text{e-}4 \end{bmatrix} \quad (7\text{-}34)$$

$$P''' = \begin{bmatrix} -3.787\text{e-}3 & 1.604\text{e+}1 & -1.661\text{e-}1 \\ 3.425\text{e+}1 & -3.450\text{e+}1 & 7.092\text{e-}1 \end{bmatrix} \quad (7\text{-}35)$$

$$\mathbf{m}_d''' = \mathbf{m}_d'' - B_1 u_{1\,Max} - B_3 u_{3\,Min} - B_4 u_{4\,Max} - B_9 u_{9\,Max} = \begin{Bmatrix} 0.0349 \\ -0.0015 \\ -0.0027 \end{Bmatrix} \quad (7\text{-}36)$$

$$\mathbf{u}''' = \{-0.0241, \mathbf{1.2447}\}^T \quad (7\text{-}37)$$

Once again, the saturated control, u_8, is removed and the final control is solved for.

$$u_8 = u_{8\,Max} = 0.5236 \quad (7\text{-}38)$$

$$B'''' = \begin{bmatrix} 6.280\text{e-}2 \\ 6.234\text{e-}2 \\ 0.000\text{e+}0 \end{bmatrix} \tag{7-39}$$

$$P''' = [\ 8.020\text{e+}0 \ \ 7.961\text{e+}0 \ \ 0.000\text{e+}0] \tag{7-40}$$

$$\mathbf{m}_d'''' = \mathbf{m}_d''' - B_8 u_{8\,Max} = \begin{Bmatrix} 0.0196 \\ -0.0015 \\ -0.0028 \end{Bmatrix} \tag{7-41}$$

$$u'''' = u_7 = \{0.1448\} \tag{7-42}$$

The final control vector for this desired moment is:

$$\mathbf{u} = \{\mathbf{0.1833}, \mathbf{0.1833}, \mathbf{-0.5236}, \mathbf{0.5236}, \mathbf{0.5236},$$

$$\mathbf{-0.1396}, 0.1448, \mathbf{0.5236}, \mathbf{0.5236}, \mathbf{-0.5236}\}^T \tag{7-43}$$

Although the final control did not saturate, the moment produced by these controls is not equal to the desired moment.

$$B\mathbf{u} = \mathbf{m}_{out} = \begin{Bmatrix} 0.0979 \\ -0.0090 \\ -0.1206 \end{Bmatrix} \neq \begin{Bmatrix} 0.1083 \\ -0.0196 \\ -0.1234 \end{Bmatrix} = \mathbf{m}_d \tag{7-44}$$

For cases when fewer than n controls are unsaturated, another option was described. That method removes only the number of controls which will result in a square reduced B matrix. Following this procedure for this example would produce the same results up to Equation 7-32. Of the controls which saturate in Equation 7-32, the commanded value for u_4 is the closest to its limit. Therefore, u_4 will not be removed. The remaining controls are solved for using the inverse of the remaining B matrix.

$$B''' = \begin{bmatrix} 5.841\text{e-}2 & 6.280\text{e-}2 & 2.920\text{e-}2 \\ -6.486\text{e-}2 & 6.234\text{e-}2 & 1.000\text{e-}5 \\ -3.911\text{e-}3 & 0.000\text{e+}0 & 3.000\text{e-}4 \end{bmatrix} \tag{7-45}$$

$$P''' = (B''')^{-1} = \begin{bmatrix} 1.983e+0 & -1.997e+0 & -1.929e+2 \\ 2.059e+0 & 1.396e+1 & -2.008e+2 \\ 2.585e+1 & -2.604e+1 & 8.179e+2 \end{bmatrix} \tag{7-46}$$

$$\mathbf{m}_d''' = \mathbf{m}_d'' - B_1 u_{1\,Max} - B_3 u_{3\,Min} - B_9 u_{9\,Max} = \begin{Bmatrix} 0.0654 \\ -0.0355 \\ -0.0047 \end{Bmatrix} \tag{7-47}$$

$$\mathbf{u}''' = \{\mathbf{1.1101}, 0.5859, \mathbf{-1.2394}\}^T \tag{7-48}$$

To differentiate these final controls from those previously determined, these will be designated $\mathbf{u}_f$.

$$\mathbf{u}_f = \{\mathbf{0.1833}, \mathbf{0.1833}, \mathbf{-0.5236}, \mathbf{0.5236}, \mathbf{0.5236},$$

$$\mathbf{-0.1396}, 0.5859, \mathbf{-0.5236}, \mathbf{0.5236}, \mathbf{-0.5236}\}^T \tag{7-49}$$

The moment produced by these controls, $B\mathbf{u}_f = \mathbf{m}_{outf} = \{0.0950, 0.0185, -0.1209\}^T$, is further away from the desired moment than the controls shown in Equation 7-43.

$$|\mathbf{m}_d - \mathbf{m}_{out}| = 0.0151, |\mathbf{m}_d - \mathbf{m}_{outf}| = 0.0404 \tag{7-50}$$

Because the method which uses the least-squares pseudo-inverse produces moments which are closer to the unattainable desired moments, all further examples use the least-squares solution.

Example 7-3: Concavity

To find the approximate volume of $\Pi_{Cascading\ GI}$, for the F-18 HARV data, 182 moment directions were chosen. These directions are the same ones used to approximate $\Pi_{Daisy\ Chaining}$. For each direction, the desired moment was increased in increments of 0.1% of the value of the maximum attainable moment in that direction until the maximum

moment for which the method of Cascading Generalized Inverses allocates admissible controls that produce the desired moment was found. The quickhull algorithm [26] was used to find the convex hull.

If a set of maximum moments is convex, all of the maximum moments will be on the boundary of the convex hull. Figure 7-1 shows an example of a convex set of maximum moments and their convex hull . However, if the set of maximum moments is convex, some of the maximum moments may be on the boundary of the convex hull, but not be extreme points of the convex hull. The point labeled 'a' in Figure 7-2 is on the boundary of the convex hull but may or may not be used as a vertex of the convex hull, depending upon the precision of the numbers used in the computation. If the set of maximum moments is concave, some of the points will not used to define the convex hull because they are not extreme points. See Figure 7-3.

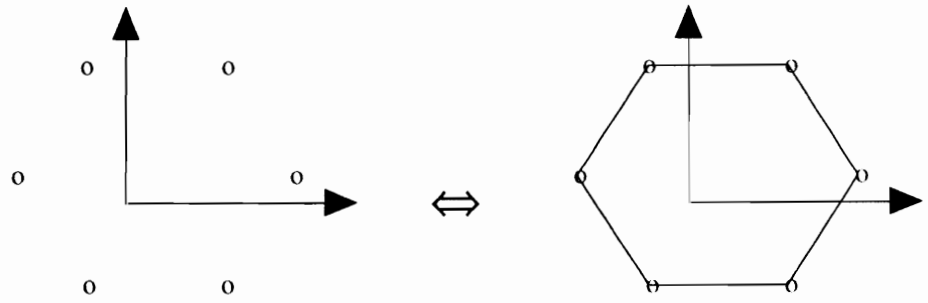


Figure 7-1: A Convex Set and Its Convex Hull

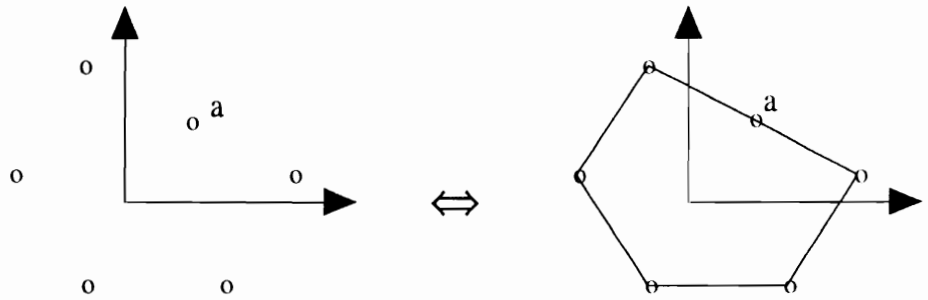


Figure 7-2: Another Convex Set and Its Convex Hull

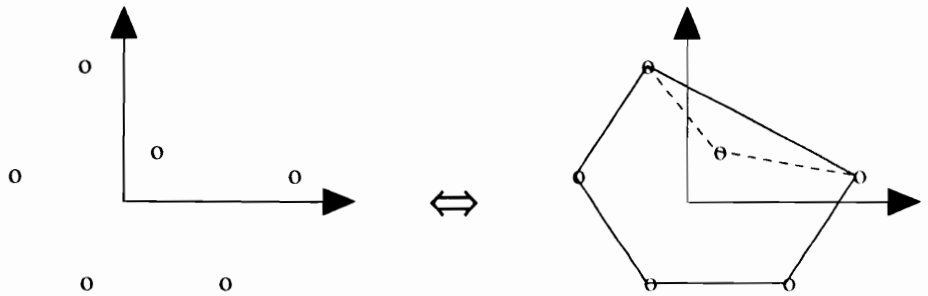


Figure 7-3: A Concave Set and Its Convex Hull

Figure 7-4 shows 4 of the 182 maximum moments. These points are connected by the convex hull algorithm to form the face shown in Figure 7-5. The fact that Point 2 is interior to this face can be shown by computing the intersection of the half-line connecting the origin and Point 2 with the face. See Figure 7-6. This intersection is computed using vector notation, see Figure 7-7.

$$\mathbf{v}_1 = \{-0.1381, 0.6728, 0.1201\}^T \tag{7-51}$$

$$\mathbf{v}_2 = \{-0.1061, 0.3343, 0.1209\}^T \tag{7-52}$$

$$\mathbf{v}_3 = \{-0.0811, 0.3519, 0.1275\}^T \tag{7-53}$$

$$\mathbf{v}_4 = \{-0.0964, 0.1658, 0.1274\}^T \tag{7-54}$$

$$\mathbf{v}_{1\text{-}3} = \mathbf{v}_3 - \mathbf{v}_1 = \{-0.0570, -0.3210, 0.0069\}^T \tag{7-55}$$

$$\mathbf{v}_{1\text{-}4} = \mathbf{v}_4 - \mathbf{v}_1 = \{0.0417, -0.5070, 0.0068\}^T \tag{7-56}$$

The face can be described as all points $\mathbf{v}_i$ which satisfy the following conditions:

$$\mathbf{v}_i = \mathbf{v}_1 + b\mathbf{v}_{1\text{-}3} + c\mathbf{v}_{1\text{-}4} \tag{7-57}$$

$$0 \le b \le 1, 0 \le c \le 1, b + c \le 1 \tag{7-58}$$

Because $\mathbf{v}_2$ points at this face, a scalar multiple of $\mathbf{v}_2$ will lie on the face:

$$a\mathbf{v}_2 = \mathbf{v}_i = \mathbf{v}_1 + b\mathbf{v}_{1\text{-}3} + c\mathbf{v}_{1\text{-}4} \tag{7-59}$$

If $a = 1$, then Point 2 lies on the face. If $a < 1$, then Point 2 is exterior to the face and $\mathbf{v}_2$ needs to be scaled down for Point 2 to lie on the face. If $a > 1$, the Point 2 is interior to the face and $\mathbf{v}_2$ needs to be scaled up for Point 2 to lie on the face.

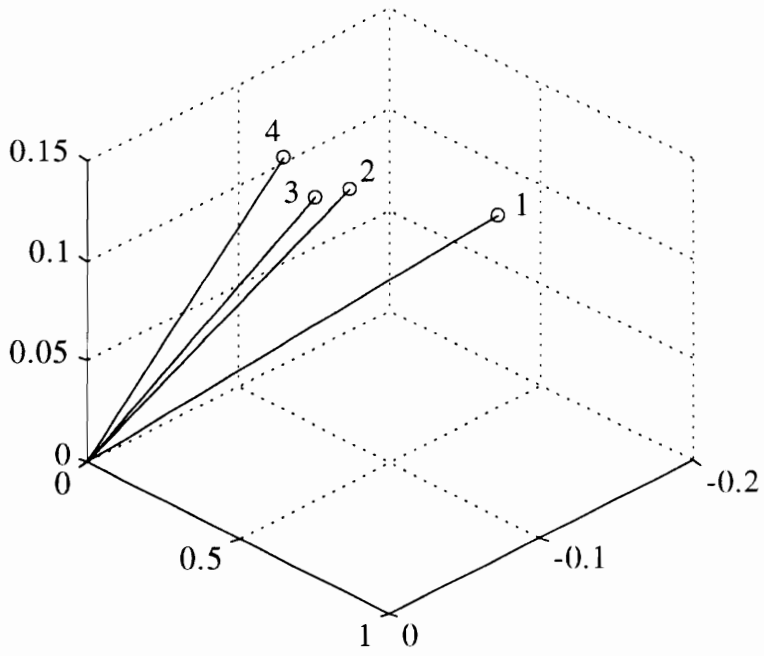


Figure 7-4: 4 Points from the Maximum Moment Set

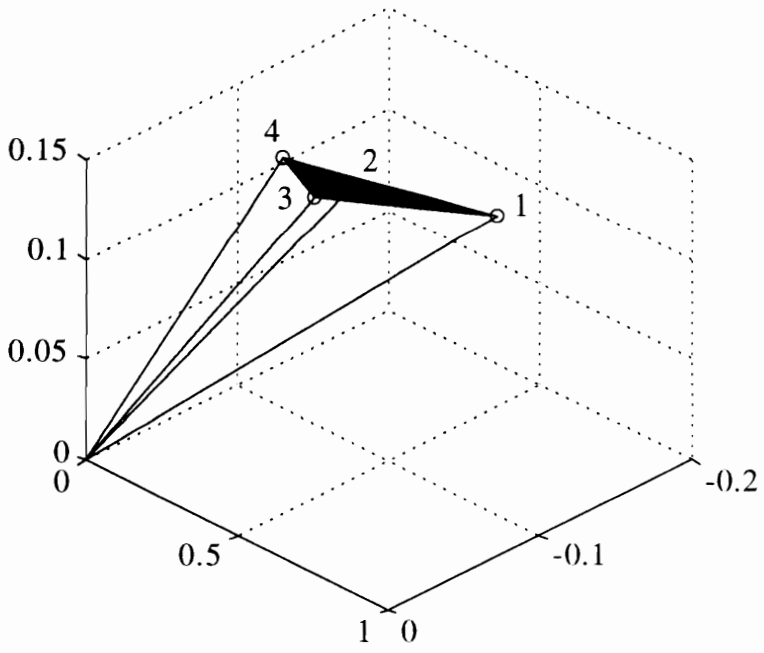


Figure 7-5: Points as Connected by Convex Hull

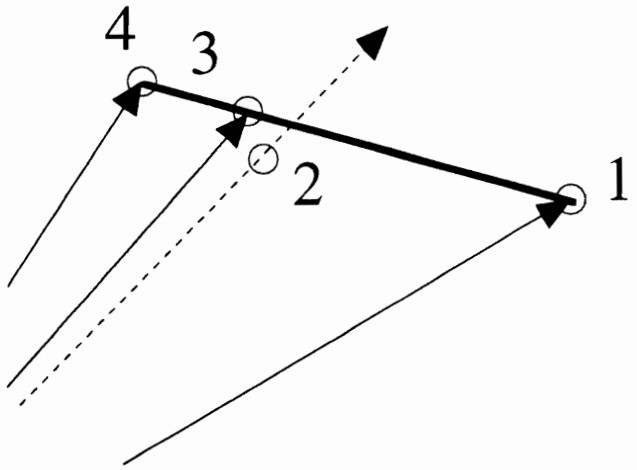


Figure 7-6: Enlarged Side View of Intersection

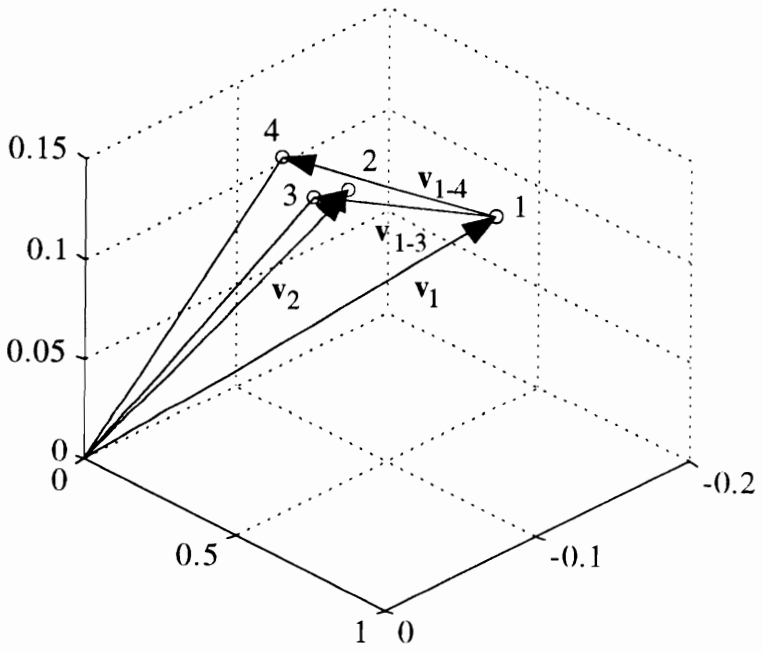


Figure 7-7: Vector Representation of the Points

The scalars a, b, and c can be solved for as follows:

$$a\mathbf{v}_2 - b\mathbf{v}_{1\text{-}3} - c\mathbf{v}_{1\text{-}4} = \mathbf{v}_1 \tag{7-60}$$

$$[\mathbf{v}_2 - \mathbf{v}_{1\text{-}3} - \mathbf{v}_{1\text{-}4}]\begin{Bmatrix} a \\ b \\ c \end{Bmatrix} = \mathbf{v}_1 \Leftrightarrow [\mathbf{v}_2 - \mathbf{v}_{1\text{-}3} - \mathbf{v}_{1\text{-}4}]^{-1}\mathbf{v}_1 = \begin{Bmatrix} a \\ b \\ c \end{Bmatrix} \tag{7-61}$$

$$\begin{Bmatrix} a \\ b \\ c \end{Bmatrix} = \begin{bmatrix} -0.6551 & 0.0474 & 7.5680 \\ -31.0093 & -2.8091 & -19.4521 \\ 20.0626 & 3.7195 & 7.3245 \end{bmatrix}\begin{Bmatrix} -0.1381 \\ 0.6728 \\ 0.1201 \end{Bmatrix} = \begin{Bmatrix} 1.0349 \\ 0.0457 \\ 0.6158 \end{Bmatrix} \tag{7-62}$$

The constraints on b and c are satisfied, so $\mathbf{v}_2$ will intersect this face. Because $a > 1$, Point 2 is interior to the face. Thus, Π for the method of Cascading Generalized Inverses is not necessarily convex.

This method allocates controls for a very large portion of Φ, failing only when the desired moment is very close to the boundary of Φ. The desired moment used in Example 7-2 is 97% of the value on the boundary of Φ. This method failed to achieve the maximum attainable moment for 24 of the 182 moment directions tested. For all 24 of these directions, a moment within 5% of the value on the boundary was attainable.

The approximate volume of Π_{CGI} is 9.0049e-02 which is about 99.9% of Φ. Figure 7-8 shows Π_{CGI}. In previous sections, Π was displayed inside a wire-frame of Φ. Because Π_{CGI} is very close to the total volume, it is difficult to distinguish from a wire-frame. For comparison purposes, Figure 7-9 shows Φ drawn with the same scale as Π_{CGI}.

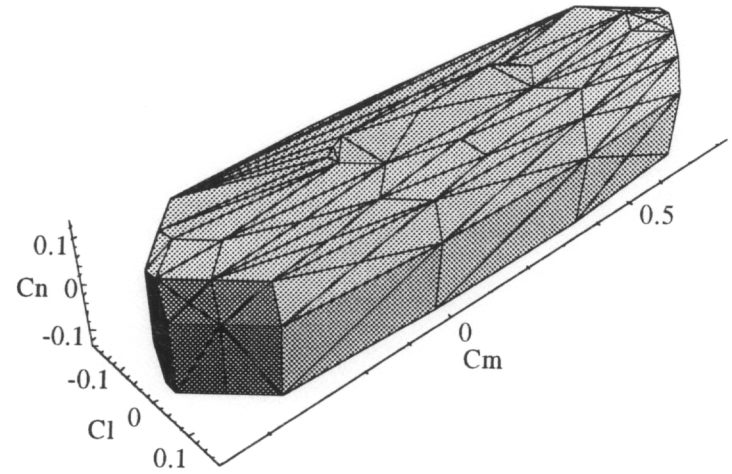


Figure 7-8: $\Pi_{Cascading\ GI}$

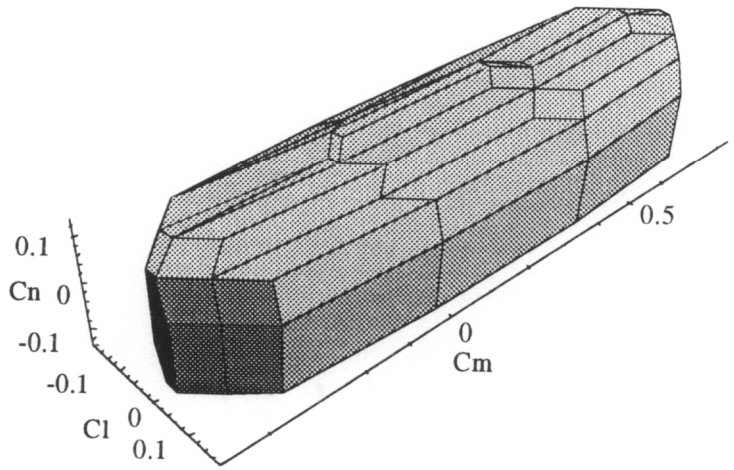


Figure 7-9: Φ for the F-18 HARV

Unattainable Moments

If the desired moment is within Φ and lies outside of $\Pi_{Cascading\ GI}$, the moment produced can differ in magnitude and direction from the desired moment (see Equation 7-44). The method of Cascading Generalized Inverses does not always allocate controls to generate every moment in Φ. Sometimes a generalized inverse will saturate a control in a direction which prevents the control vector from being in the set of all admissible controls which generate a given desired moment. This will be explained further and illustrated in the next section.

For moments which lie outside of Φ, the magnitude of the moment produced will be smaller than the magnitude of the desired moment. The direction of the desired moment will not, in general, be preserved.

8. NULL-SPACE INTERSECTION SOLUTIONS

Method Origin

The Null-Space Intersection method was derived from the same principles used to compute Π for a generalized inverse. Specifically, the principles involved in finding the intersection of a subspace with a higher dimensional polytope are used. When calculating Π, the subspace represents directions in which the controls should be deflected to generate specified moments. The Null-Space Intersection method involves the use of a subspace which represents directions in which the controls can be deflected without changing the moment generated.

If an allocation scheme asks a control to exceed a position limit, then the control vector is at a point outside of Ω in control space. If there exists a set of admissible controls which generate the desired moment, then they can be found by traveling through the null-space from the point outside of Ω to a point in Ω or on the boundary of Ω.

The subset of all admissible controls which generate a specified moment can be represented as the intersection of Ω with an $(m\text{-}n)$-flat. This $(m\text{-}n)$-flat is the null-space of B translated to correspond with the desired moment. The Null-Space Intersection method finds this intersection and selects the point which is closest to the original inadmissible control vector.

Description of Method

For an nxm matrix B, with $m > n$ and rank(B) = n, there exists a set of (m-n) orthonormal vectors which form a basis for the null-space of B in R^m. The null-space of a matrix can be extracted from the singular-value decomposition of the matrix. [29, 30] The set of orthonormal basis vectors will be denoted $\aleph$, where $\aleph$ is a matrix whose m-n columns are m-vectors.

$$\aleph = [\,\aleph_1 | \aleph_2 | \ldots | \aleph_{m-n}] \tag{8-1}$$

$$B\aleph\mathbf{x} = \mathbf{0} \;\forall\; \mathbf{x} \in R^{m-n} \tag{8-2}$$

The columns of the pseudo-inverse of B span the range space of B and is perpendicular to the null-space of B. P and $\aleph$ form a basis for R^m, and all $\mathbf{u} \in R^m$ can be written as linear combinations of the columns of P and $\aleph$. All solutions to $B\mathbf{u} = \mathbf{m}$ can be written as follows:

$$P_{pseudo} = B^{\mathrm{T}}[BB^{\mathrm{T}}]^{-1} \tag{8-3}$$

$$\mathbf{u_p} = P_{pseudo}\mathbf{m} \tag{8-4}$$

$$\mathbf{u_n} = \aleph\mathbf{x} \tag{8-5}$$

$$\mathbf{u_p} + \mathbf{u_n} = \mathbf{u} \text{ such that } B\mathbf{u} = \mathbf{m} \tag{8-6}$$

If none of the controls in $\mathbf{u_p}$ exceed their limits, then $\mathbf{u_p}$ is used and $\mathbf{x} = \mathbf{0}$. If one or more of the controls in $\mathbf{u_p}$ is outside its limits, the objective is to find an $\mathbf{x}$ such that $\mathbf{u_n}$ will bring these controls back to their limits without driving other controls outside their

limits. One way to do this is to find the intersection of the null-space with the subset of constrained controls.

It is possible to find the intersection of the $(m\text{-}n)$-D null-space with m-D subset of admissible controls, Ω. The method for finding this intersection is similar to the method used to find the intersection of an n-D subspace with Ω. The difference is that the null-space is first translated so that it is centered at $\mathbf{u_p}$, the pseudo-inverse solution. This translation is done so that the resulting controls will generate the desired moment. Also, to find the intersection for Π, $(m\text{-}n)$-D or smaller objects are searched to find the extreme points. To describe the intersection of an $(m\text{-}n)$-D subspace and an m-D object, bounding objects of size n-D and smaller need to be checked. However, the point of the intersection closest to the original control vector may not be an extreme point. Searching objects of size n-D or smaller is sufficient to describe the intersection, but insufficient to find the point closest to the original vector. All of the bounding objects, $(m\text{-}1)$-D and smaller, need to be searched. Example 8-1 is a low order example which illustrates this point.

The intersections may be found by partitioning $\aleph$ into two matrices, $\aleph_1$ and $\aleph_2$. $\aleph_1$ is associated with the controls which are set to their limits. $\aleph_2$ is associated with the controls which are solved for. Similarly partition $\mathbf{u}$.

$$\aleph = \begin{bmatrix} \aleph_1 \\ \aleph_2 \end{bmatrix} \tag{8-7}$$

$$\mathbf{u} = \begin{Bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{Bmatrix} \tag{8-8}$$

Then Equation 8-5 can be rewritten as:

$$\begin{bmatrix} \aleph_1 \\ \aleph_2 \end{bmatrix} \mathbf{x} = \begin{Bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{Bmatrix} - \begin{Bmatrix} \mathbf{u}_{p1} \\ \mathbf{u}_{p2} \end{Bmatrix} \tag{8-9}$$

The controls in $\mathbf{u}_1$ are set at saturation and the controls in $\mathbf{u}_2$ are determined by:

$$\mathbf{x} = \aleph_1^{\dagger}(\mathbf{u}_1 - \mathbf{u}_{p1}) \tag{8-10}$$

[Note: $\aleph_1^{\dagger}$ = the pseudo-inverse of $\aleph_1$. If $\aleph_1$ is tall, the least-squares pseudo-inverse is used. If $\aleph_1$ is wide, the minimum-norm pseudo-inverse is used. If $\aleph_1$ is square and invertible $\aleph_1^{-1}$ is used.]

$$\mathbf{u}_2 = \aleph_2\mathbf{x} + \mathbf{u}_{p2} \tag{8-11}$$

If the controls in $\mathbf{u}_2$ are on or within their limits, then this point is on the surface of Ω and lies in the translated null-space of B. Thus, it is a solution of admissible controls which generate the desired moment. To test every (m-1)-D or smaller object, every combination of controls placed in $\mathbf{u}_1$ is set to every combination of minimums and maximums. $\aleph_1$ initially contains only 1 of the m controls, which has 2 possible positions, minimum or maximum, such that there are $2m$ possible combinations. In this way, all of the (m-1)-D bounding objects are searched. If no solutions are found, $\aleph_1$ is enlarged to contain 2 of the m controls. There are $m!/2!(m-2)!$ ways to take m controls two at a time. For each of these combinations, there are now 2^2 possible combinations for the limits. In general, for m controls taken i at a time, there are $\frac{2^i m!}{i!(m-i)!}$ combinations. For m controls, i=1,2,...,m-1. For high order problems, the number of combinations, and thus computations, can become very large. For example, the F-18 HARV with 10

controls requires checking $\sum_{i=1}^{9} \frac{2^i 10!}{i!(10-i)!} = 58{,}024$ objects for possible intersections for each desired moment.

Once the intersections are found, the intersection which is closest to $\mathbf{u_p}$ is selected, using the Euclidean norm. This solution is selected to insure the continuity of the solutions and to prevent the rapid reconfiguration of the controls. Also, the pseudo-inverse solutions have the special property of minimizing the 2-norm of the control vector. The admissible solutions which are close to this solution will have smaller norms than those further away.

Example 8-1: A Low Order Example

For this example, a simple case with three controls and one desired moment will be used to illustrate some of the properties of the Null-Space Intersection method. Ω is a three dimensional box, the pseudo-inverse maps the moments to a line in control space, and the null-space defines a 2-D plane in control space.

$$B = [7.35e\text{-}4 \quad 7.55e\text{-}4 \quad -1.35e\text{-}4] \tag{8-12}$$

$$-20 \le u_1 \le 20$$

$$-20 \le u_2 \le 20$$

$$-30 \le u_3 \le 30 \tag{8-13}$$

$$P_{pseudo} = \begin{bmatrix} 651.3215 \\ 669.0445 \\ -119.6305 \end{bmatrix} \tag{8-14}$$

$$\mathcal{N} = \begin{bmatrix} -0.7220 & 0 \\ 0.6811 & 0.1760 \\ -0.1218 & 0.9844 \end{bmatrix} \tag{8-15}$$

$$\mathbf{m}_d = \{0.03\} \in \Phi \tag{8-16}$$

For this desired moment, the pseudo-inverse allocates the following controls:

$$\mathbf{u} = \{19.5396, 20.0713, -3.5889\}^T \tag{8-17}$$

The second control has exceeded its positive limit, $u_2 > 20$. Figures 8-1 through 8-3 show Ω and the desired moment represented by the solid circle.

The shaded triangle shows the intersection of the null-space with Ω. To define this intersection, the extreme points can be found by checking all the n-D objects bounding Ω. For this problem, $n = 1$, so all the edges must be checked. For example, on the edge {112} the intersection is computed as follows:

$$\mathcal{N}_1 = \begin{bmatrix} -0.7220 & 0 \\ 0.6811 & 0.1760 \end{bmatrix} \tag{8-18}$$

$$\mathcal{N}_1^{-1} = \begin{bmatrix} -1.3850 & 0 \\ 5.3594 & 5.6813 \end{bmatrix} \tag{8-19}$$

$$\mathbf{u}_1 = \begin{Bmatrix} 20 \\ 20 \end{Bmatrix} \tag{8-20}$$

$$\mathbf{u}_2 = \mathbf{u}_{p2} + \mathcal{N}_2 \mathcal{N}_1^{-1}(\mathbf{u}_1 - \mathbf{u}_{p1}) \tag{8-21}$$

$$\mathbf{u}_2 = -3.5889 + [-0.1218 \;\; 0.9844] \begin{bmatrix} -1.3850 & 0 \\ 5.3594 & 5.6813 \end{bmatrix} \left(\begin{Bmatrix} 20 \\ 20 \end{Bmatrix} - \begin{Bmatrix} 19.5396 \\ 20.0713 \end{Bmatrix} \right) \tag{8-22}$$

$$\mathbf{u}_2 = -1.4815 = u_3 \tag{8-23}$$

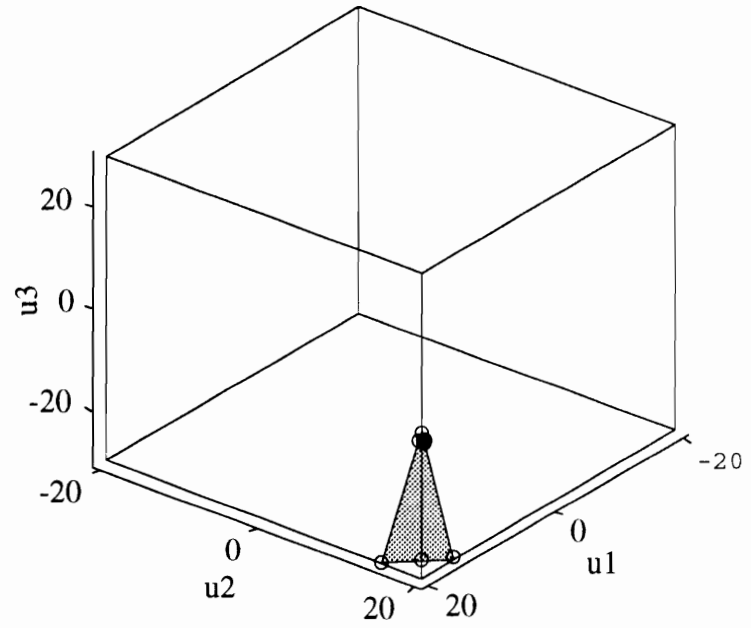


Figure 8-1: Null-Space Intersecting Ω

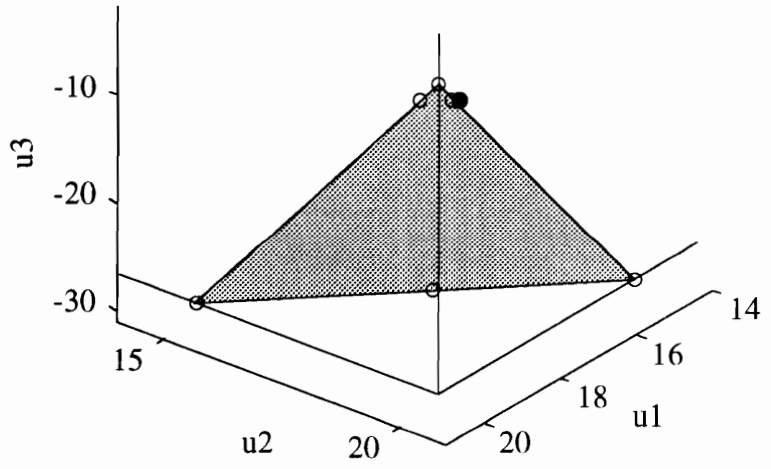


Figure 8-2: Null-Space Intersecting Ω : Magnified

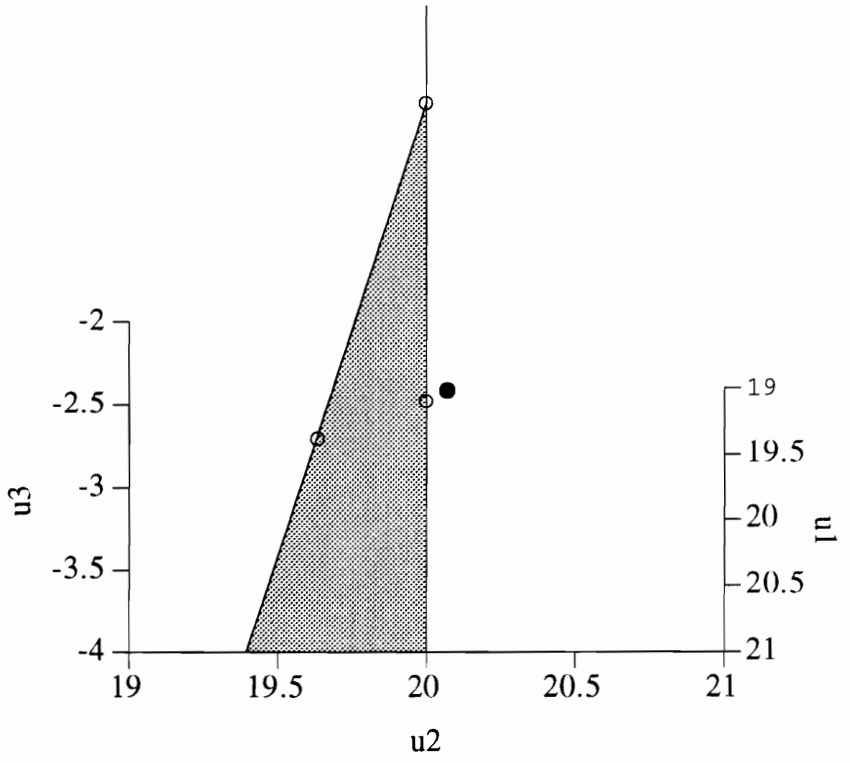


Figure 8-3: Null-Space Intersecting Ω: Side View

Because u_3 is within its limits, this point is a valid intersection. Repeating this procedure for all the edges yields the points shown in Table 8-1, which are the corners of the triangle in Figure 8-1.

However, the point on the boundary of Ω which is closest to the pseudo-inverse solution is not one of these corners, but a point along an edge of the triangle. This can be seen by computing the intersections which lie along the faces of the box. For these objects $\aleph_1$ is not square and so the minimum-norm pseudo-inverse is used.

Consider the face {122}:

$$\aleph_1 = [-0.7220 \;\; 0] \tag{8-24}$$

$$\aleph_1^{\dagger} = \begin{bmatrix} -1.3850 \\ 0.0000 \end{bmatrix} \tag{8-25}$$

$$\mathbf{u}_1 = \{20\} \tag{8-26}$$

$$\mathbf{u}_2 = \mathbf{u}_{p2} + \aleph_2 \aleph_1^{-1}(\mathbf{u}_1 - \mathbf{u}_{p1}) \tag{8-27}$$

$$\mathbf{u}_2 = \begin{Bmatrix} 20.0713 \\ -3.5889 \end{Bmatrix} + \begin{bmatrix} 0.6811 & 0.1760 \\ -0.1218 & 0.9844 \end{bmatrix} \begin{bmatrix} -1.3850 \\ 0.0000 \end{bmatrix} (20 - 19.5396) \tag{8-28}$$

$$\mathbf{u}_2 = \begin{Bmatrix} 19.6371 \\ -3.5113 \end{Bmatrix} = \begin{Bmatrix} u_2 \\ u_3 \end{Bmatrix} \tag{8-29}$$

Because u_2 and u_3 are within their limits, this is a valid intersection. Checking all the faces yields the points shown in Table 8-2. From this table, the point which corresponds to object number {212} is closest to the pseudo-inverse solution and has the smallest magnitude. The Null-Space Intersection method will allocate these controls:

$$\mathbf{u} = \{19.6105, 20, -3.6019\}^T \tag{8-30}$$

Table 8-1: Vertices of Intersection

Object #	u_1	u_2	u_3	$\|\mathbf{u}\|_2$
{112}	**20.0000**	**20.0000**	-1.4815	28.323
{120}	**20.0000**	14.9007	**-30.0000**	39.013
{210}	14.7619	**20.0000**	**-30.0000**	38.960

Table 8-2: Intersections on Faces

Object #	u_1	u_2	u_3	$\|\mathbf{u}\|_2$
{122}	**20.0000**	19.6371	-3.5113	28.248
{212}	19.6105	**20.0000**	-3.6019	28.241
{220}	17.1792	17.6467	**-30.0000**	38.814

Example 8-2: Revisiting the Cascading Generalized Inverse $B\mathbf{u}\neq\mathbf{m}$

This example uses the F-18 HARV data and the desired moment from Example 7-2.

$$B = \begin{bmatrix} \text{-4.382e-2} & \text{4.382e-2} & \text{-5.841e-2} & \text{5.841e-2} & \text{1.674e-2} \\ \text{-5.330e-1} & \text{-5.330e-1} & \text{-6.486e-2} & \text{-6.486e-2} & \text{0.000e+0} \\ \text{1.100e-2} & \text{-1.100e-2} & \text{3.911e-3} & \text{-3.911e-3} & \text{-7.428e-2} \end{bmatrix} \ldots$$

$$\ldots \begin{bmatrix} \text{-6.280e-2} & \text{6.280e-2} & \text{2.920e-2} & \text{1.000e-5} & \text{1.000e-2} \\ \text{6.234e-2} & \text{6.234e-2} & \text{1.000e-5} & \text{3.553e-1} & \text{1.000e-5} \\ \text{0.000e+0} & \text{0.000e+0} & \text{3.000e-4} & \text{1.000e-5} & \text{1.485e-1} \end{bmatrix} \tag{8-31}$$

$$P_{pseudo} = \begin{bmatrix} \text{-2.201e+0} & \text{-7.500e-1} & \text{-3.350e-1} \\ \text{2.202e+0} & \text{-7.500e-1} & \text{3.353e-1} \\ \text{-2.960e+0} & \text{-9.125e-2} & \text{2.210e-1} \\ \text{2.960e+0} & \text{-9.128e-2} & \text{-2.210e-1} \\ \text{9.492e-1} & \text{1.383e-5} & \text{-2.693e+0} \\ \text{-3.177e+0} & \text{8.774e-2} & \text{8.643e-2} \\ \text{3.177e+0} & \text{8.771e-2} & \text{-8.646e-2} \\ \text{1.477e+0} & \text{6.081e-6} & \text{-2.940e-2} \\ \text{3.958e-4} & \text{4.999e-1} & \text{2.5786e-4} \\ \text{3.015e-1} & \text{-2.537e-5} & \text{5.325e+0} \end{bmatrix} \tag{8-32}$$

$$\mathcal{N} = \begin{bmatrix} \text{-7.071e-1} & 0 & 0 & 0 & 0 & 0 & 0 \\ \text{4.236e-1} & \text{6.159e-2} & \text{1.458e-1} & \text{3.562e-2} & \text{-2.592e-2} & \text{5.171e-1} & \text{-1.612e-1} \\ \text{2.488e-1} & \text{-1.043e-1} & \text{-4.051e-1} & \text{4.798e-1} & \text{2.066e-1} & \text{2.128e-1} & \text{5.148e-1} \\ \text{-1.112e-1} & \text{-4.340e-1} & \text{2.137e-1} & \text{-2.220e-1} & \text{-9.979e-2} & \text{-2.384e-2} & \text{7.169e-1} \\ \text{-1.693e-2} & \text{8.035e-1} & \text{3.648e-2} & \text{-3.611e-2} & \text{-1.612e-2} & \text{1.060e-3} & \text{3.675e-1} \\ \text{1.294e-1} & \text{4.786e-3} & \text{8.619e-1} & \text{1.483e-1} & \text{6.672e-2} & \text{2.917e-2} & \text{8.955e-2} \\ \text{-2.616e-1} & \text{-6.741e-3} & \text{1.408e-1} & \text{8.269e-1} & \text{-7.313e-2} & \text{-9.186e-2} & \text{-9.517e-2} \\ \text{-9.107e-2} & \text{-1.893e-3} & \text{6.485e-2} & \text{-7.475e-2} & \text{9.674e-1} & \text{-2.821e-2} & \text{-4.453e-2} \\ \text{-3.769e-1} & \text{-5.547e-3} & \text{7.926e-3} & \text{-7.061e-2} & \text{-1.828e-2} & \text{8.213e-1} & \text{-1.609e-2} \\ \text{-1.015e-1} & \text{3.887e-1} & \text{2.361e-2} & \text{-3.903e-2} & \text{-1.616e-2} & \text{-4.401e-2} & \text{2.012e-1} \end{bmatrix}$$

(8-33)

$$\mathbf{m}_d = \begin{Bmatrix} 0.1083 \\ \text{-0.0196} \\ \text{-0.1234} \end{Bmatrix} \in \Phi \tag{8-34}$$

It was demonstrated in Section 7 that the method of Cascading Generalized inverses fails to allocate controls for this moment because one or more of the controls was fixed at saturation when the only admissible solutions which generate this moment have that control varying or saturated in the opposite sense. In the first step of the Cascading Generalized Inverse method, the left horizontal tail, u_2, is set to its maximum value. However, all the valid intersections for this moment have u_2 either fixed at its minimum or negative. Table 8-3 shows the valid intersections found when checking the 5-D bounding objects. Note that u_2 is negative at each intersection. Checking objects of other sizes produces 68 other points, all of which have u_2 either negative or fixed at its minimum.

The controls allocated by the Null-Space Intersection method for this moment are:

$$\mathbf{u} = \{0.1617, -0.0874, \mathbf{-0.5236}, \mathbf{0.5236}, \mathbf{0.5236}, \mathbf{-0.1396}, 0.6044, 0.2681, -0.0252, \mathbf{-0.5236}\}^T \quad (8\text{-}35)$$

For these controls, the moment produced is equal to the desired moment.

$$\boldsymbol{B}\mathbf{u} = \begin{Bmatrix} 0.1083 \\ -0.0196 \\ -0.1234 \end{Bmatrix} = \mathbf{m}_d \quad (8\text{-}36)$$

Table 8-3: Some Valid Intersections

u_1	**0.1833**	0.1617	0.1706	0.1820	0.0258
u_2	*-0.1014*	*-0.0874*	*-0.0719*	*-0.1007*	*-0.2591*
u_3	-0.4612	**-0.5236**	**-0.5236**	-0.4193	-0.4824
u_4	0.4767	**0.5236**	**0.5236**	**0.5236**	0.4540
u_5	**0.5236**	**0.5236**	**0.5236**	**0.5236**	**0.5236**
u_6	**-0.1396**	**-0.1396**	-0.0674	**-0.1396**	**-0.1396**
u_7	**0.7854**	0.6044	**0.7854**	**0.7854**	**0.7854**
u_8	0.1508	0.2681	0.0242	0.1377	0.1542
u_9	-0.0428	-0.0252	-0.0330	-0.0274	**-0.5236**
u_{10}	**-0.5236**	**-0.5236**	**-0.5236**	**-0.5236**	**-0.5236**

Initial Solutions

This method is general and can also be used with initial solutions other than $\mathbf{u_p}$. The value of this method lies in the fact that it can augment other methods to enable them to produce all of the attainable desired moments. The initial control vector can be generated using any means, as long as it satisfies $B\mathbf{u} = \mathbf{m}$.

The initial solutions can be used to impose special conditions on the solutions, such as minimizing the 2-norm of the control vector. In this way, the conditions are met whenever possible. When the conditions cannot be met because the controls are inadmissible, the conditions are violated so that admissible controls are used to produce the desired moment. For example, this method can be used to satisfy the conditions imposed on a daisy-chained solution which requires that a certain subset of the controls, $\mathbf{u}_2$, be used only when the rest of the controls, $\mathbf{u}_1$, fail to generate the desired moment. If the initial solution uses only the controls in $\mathbf{u}_1$, then $\mathbf{u_n}$ can be used to bring in other controls as necessary.

Unattainable Moments

The only moments for which this method will not allocate admissible solutions are those moments which are outside of Φ. However, when a moment outside of Φ is commanded, this method has a serious problem. It will find no intersections and thus have no solution. For an algorithm to produce no solution is unacceptable if the algorithm is to be implemented on a real-time control system.

An additional detriment to the practical implementation of this algorithm is its computational inefficiency. The number of computations necessary to find a solution

increases dramatically for higher order problems because the number of objects to be searched increases factorially as a function of the controls. Figure 8-4 shows the number of objects to be searched as a function of the number of controls for a three moment problem. As a result, the method of Null-Space Intersections is difficult to implement in real-time for higher order problems.

This method also has problems with numerical considerations. When searching for intersections, the method checks to see if controls violate their limits. If a control is close to its limit the computer may not have sufficient precision to accurately determine if the control is at or below its limit. This lack of precision can result in some valid intersections not being found. This inability to find intersections may in turn cause the algorithm to find no solution for some moments which are attainable .

The method of Null-Space Intersections can be used to generate admissible controls for all the attainable moments while satisfying various conditions. However, its implementation difficulties limit it to being primarily a research tool, with little potential for real world application.

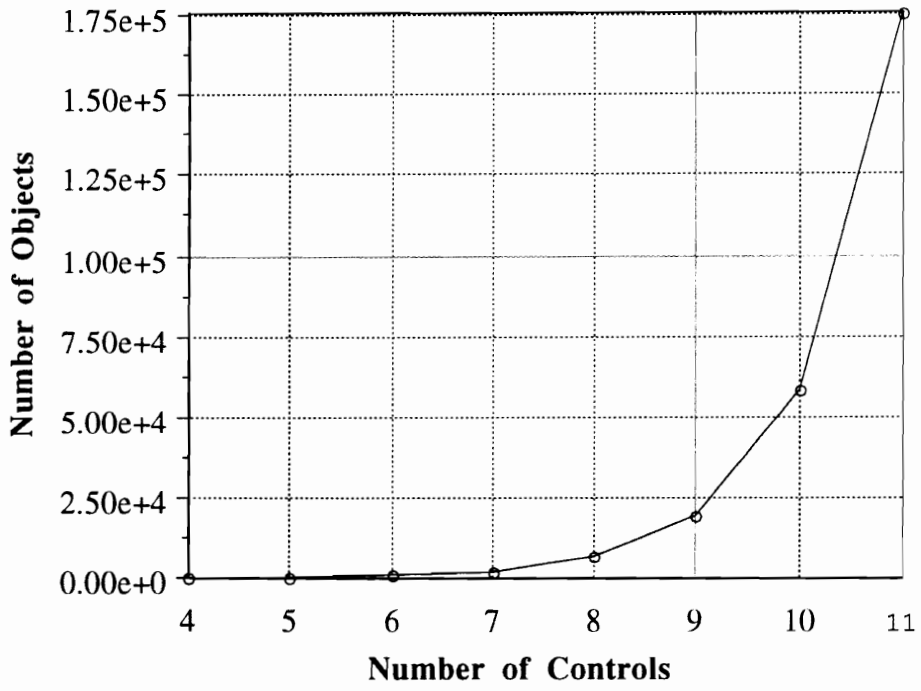


Figure 8-4: Number of Objects to be Searched

9. DIRECT ALLOCATION

Geometric Issues

The method of Direct Allocation uses the geometry of the AMS to determine controls which generate the moments inside Φ. There are two important geometrical features which Φ must possess for Direct Allocation to be used. They are:

1) A half-line from the origin of moment space will intersect the boundary of Φ at a single point.

2) Points on the boundary of Φ correspond to unique points on the boundary of Ω.

In order for an AMS to posses the first feature, it sufficient that the following two conditions are met: the origin of moment space is contained within Φ, and Φ is convex. These conditions insure that the half-line from the origin does not enter and then exit Φ, intersecting $\partial(\Phi)$ more than once.

Most controls are defined such that they have positive and negative position limits. If all the controls to be examined in a specific problem have positive and negative position limits, the origin of control space is contained within Ω. In order for the origin of moment space to be contained within Φ, it is sufficient that the origin of control space is contained within Ω. Under the linear mapping B, if the origin of R^m is contained within Ω, the origin of R^n will be contained within Φ.

It is possible for some control to have a range of motion that has been defined so that the position limits do not change sign, e.g. -5° to -20°. A control such as this would move the origin of control space outside of Ω. However, it is possible to change the control

limits so that the origin is contained in Ω. This can be done by adding a $\Delta\mathbf{u}$ to the control vector limits and changing the corresponding desired moment.

$$u_{i\,Min} \le u_i \le u_{i\,Max} \tag{9-1}$$

$$u_{i\,Min} + \Delta u_i \le u_i + \Delta u_i \le u_{i\,Max} + \Delta u_i \tag{9-2}$$

Define a new set of controls $\mathbf{u}'$, and a new desired moment, $\mathbf{m}_d'$:

$$\mathbf{u}' = \mathbf{u} + \Delta\mathbf{u} \Leftrightarrow \mathbf{u} = \mathbf{u}' - \Delta\mathbf{u} \tag{9-3}$$

$$\mathbf{m}_d = B\mathbf{u} = B(\mathbf{u}' - \Delta\mathbf{u}) = B\mathbf{u}' - \Delta\mathbf{m} \tag{9-4}$$

$$\mathbf{m}_d' = \mathbf{m}_d + \Delta\mathbf{m} = B\mathbf{u}' \tag{9-5}$$

By following the above procedure, it is possible to guarantee that the origin of moment space is contained within Φ.

To insure the convexity of Φ, it is necessary and sufficient that Ω be convex. If the position limits can be represented as planes or hyper-planes in R^m, Ω will be convex. The sides of the box in Figure 4-1 are made up of sections of such planes. The hyper-planes define an m-dimensional polyhedron which is homogenous, orthogonal, and convex. Under a linear transformation, such as a matrix multiplication, a convex set will remain convex. A proof of this is included in Appendix A.

In order for an AMS to possess the feature that every point on $\partial(\Phi)$ corresponds to a unique point on $\partial(\Omega)$, every nxn submatrix of B must be of full rank. If the submatrices are full rank, then every point on the boundary of Φ will be the image of a unique solution in $\partial(\Omega)$. A proof of this is included in Appendix A. Therefore, B is required to

have nxn submatrices which are non-singular. Time varying B matrices may or may not meet this condition. Techniques have been developed to handle cases when the conditions are not met and are described later in Section 12.

Description of Method

Direct Allocation uses geometric principles. If the previously stated geometric conditions are met, then a half-line from the origin of R^n through a moment will intersect the boundary of Φ at a single point. The point on the boundary of Φ will correspond to a unique point in control space. The controls at this point generate the moment on the boundary. Because the problem is linear, a solution on the boundary can be scaled down to satisfy a moment command which is in the interior of Φ.

$$a\mathbf{m}_d = aB\mathbf{u} = B(a\mathbf{u}) \tag{9-6}$$

The intersection of the desired moment direction with the boundary of Φ in R^n corresponds to the intersection of a subspace with the boundary of Ω in R^m. The specified direction in moment space corresponds to a specified subspace in control space. This subspace will be of dimension $(m\text{-}n+1)$ because there are $(m\text{-}n)$ directions for the null-space and one direction for the desired moment. The $(m\text{-}n+1)$-D subspace will intersect an m-D Ω at a point on an object of dimension n-1 or smaller. Thus, all the $(n\text{-}1)$-D objects on the boundary of Ω must be searched to find all the intersections. The intersection which generates the largest moment lies on the boundary of Φ.

However, as shown in Section 4, not all of the objects on the boundary of Ω are on the boundary of Φ. Since the $(n\text{-}1)$-D object sought is on the boundary of Φ, only those $(n\text{-}1)$-D objects which are on both the boundary of both Ω and Φ need to be searched.

Determining the Boundary of Φ

The boundary of Φ can be determined using several techniques. The first would involve mapping all the vertices of Ω to R^n and then fitting a convex hull around them. Finding the convex hull would determine the vertices, edges, and faces, etc. which formed the boundary of Φ. The second way of determining $\partial(\Phi)$ involves using knowledge of the geometry of Ω to search for the bounding objects in a more efficient fashion. Before describing this more efficient search, several ideas need to be discussed.

The techniques of Direct Allocation can be applied to any problem where $n < m$. To make the notation less cumbersome, consider the case of $n = 3$. Thus, n-1-D objects are 2-D faces which correspond to 2 controls free and m-2 controls fixed. The object numbers of the faces searched will have two 2's. All the possible combinations of m controls taken two at a time with the remaining controls set to all possible combinations of minimum and maximum values represent all the faces of Ω.

Faces which have the same two controls varying will be parallel in R^m. This is due to the fact that they are parallel in control space. Figure 9-1 shows a 4-D Ω projected into a 3-D moment space with one set of parallel faces shaded. All the edges of Ω in Figure 9-1 which correspond to a given control varying are parallel. Objects which are parallel in R^m remain parallel in R^n. Proof of this fact is given in Appendix A.

Using this information, several faces may be simultaneously checked to see which are on the boundary of Φ. If one of the axes in moment space is aligned to be perpendicular to the faces under consideration, all of those faces can be checked simultaneously. Only two of the faces can be on the boundary of Φ. The faces cannot be coplanar because that would violate the second geometric condition.

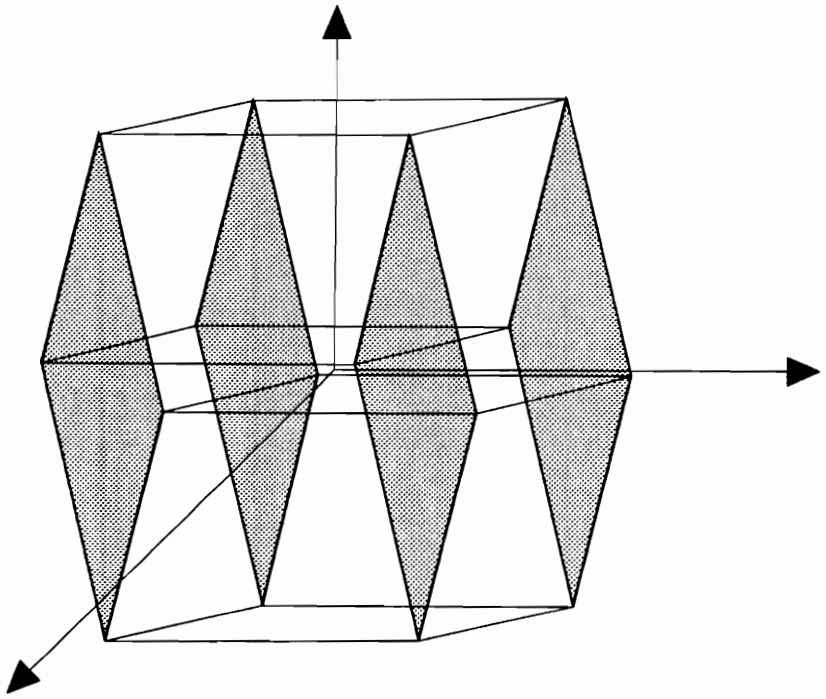


Figure 9-1: Parallel Faces in Ω Mapped to R^n

The term facet will be used to designate a face of Ω which is also a face on the boundary of Φ.

To align an axis of R^n perpendicular to a set of faces, consider an nxn transformation matrix T. The rotated matrix will be TB. Since only one axes is to be specified, $\hat{\mathbf{m}}_1$ for example, consider only one row of T. Let this row be called $\mathbf{t}$:

$$\mathbf{t} = [t_1\ t_2\ \ldots\ t_n] \tag{9-7}$$

Each column of B corresponds to a specific control:

$$B = [B_1\ B_2\ \ldots\ B_m] \tag{9-8}$$

To find the rotation which will align $\hat{\mathbf{m}}_1$ perpendicular to a given set of parallel faces, the two controls which define these faces, u_i and u_j, are examined. For $\hat{\mathbf{m}}_1$ to be perpendicular to these faces, $\mathbf{t}B_i$ and $\mathbf{t}B_j$ must be zero. Using this information, the row vector $\mathbf{t}$ is computed. For example, if $n = 3$:

$$\mathbf{t}B_i = [t_1\ t_2\ t_3]\begin{Bmatrix} B_{i1} \\ B_{i2} \\ B_{i3} \end{Bmatrix} = 0 \tag{9-9}$$

$$\mathbf{t}B_j = [t_1\ t_2\ t_3]\begin{Bmatrix} B_{j1} \\ B_{j2} \\ B_{j3} \end{Bmatrix} = 0 \tag{9-10}$$

$$\begin{bmatrix} B_{i1} & B_{i2} \\ B_{j1} & B_{j2} \end{bmatrix}\begin{Bmatrix} t_1 \\ t_2 \end{Bmatrix} + t_3\begin{Bmatrix} B_{i3} \\ B_{j3} \end{Bmatrix} = \mathbf{0} \tag{9-11}$$

$$\begin{Bmatrix} t_1 \\ t_2 \end{Bmatrix} = -\begin{bmatrix} B_{i1} & B_{j1} \\ B_{i2} & B_{j2} \end{bmatrix}^{-1} t_3\begin{Bmatrix} B_{i3} \\ B_{j3} \end{Bmatrix} \tag{9-12}$$

From Equation 9-12, t_3 can be chosen arbitrarily to satisfy Equations 9-9 and 9-10. Typically, it is set equal to 1. If the matrix in Equation 9-12 is not invertible, **t** can be found by inverting one of the other 2x2 matrices:

$$\begin{Bmatrix} t_1 \\ t_3 \end{Bmatrix} = -\begin{bmatrix} B_{i1} & B_{j1} \\ B_{i3} & B_{j3} \end{bmatrix}^{-1} t_2 \begin{Bmatrix} B_{i2} \\ B_{j2} \end{Bmatrix} \tag{9-13}$$

$$\begin{Bmatrix} t_2 \\ t_3 \end{Bmatrix} = -\begin{bmatrix} B_{i2} & B_{j2} \\ B_{i3} & B_{j3} \end{bmatrix}^{-1} t_1 \begin{Bmatrix} B_{i1} \\ B_{j1} \end{Bmatrix} \tag{9-14}$$

If all the 3x3 submatrices of B have rank 3, then at least one of the 2x2 submatrices will be invertible.

Once **t** is known, the vector $\mathbf{t}B$ can be examined to determine which pair of the parallel faces are on the boundary of Φ. The faces on the boundary of Φ will have controls which map to the greatest positive and negative distances in the specified direction. Thus, one facet will have controls which maximize $\mathbf{t}B\mathbf{u}$ and the other facet will have controls which minimize $\mathbf{t}B\mathbf{u}$.

The vector $\mathbf{t}B$ will have two zero entries, corresponding to the i-th and j-th elements. The other entries will be either positive or negative. The maximum of $\mathbf{t}B\mathbf{u}$ will have controls which correspond to positive elements of $\mathbf{t}B$ set to maximum values and controls which correspond to negative elements of $\mathbf{t}B$ set to minimum values. The minimum of $\mathbf{t}B\mathbf{u}$ will have controls which correspond to the positive elements set to minimum values and the negative elements set to positive values. For example, if $\mathbf{t}B = [0\ 0\ -\ +]$, the object numbers of the two facets are {2201} and {2210}. The vertices of these facets are computed by setting the varying controls to all combinations of minimum and maximum values. For the facet {2201}, the vertices are {0001}, {0101}, {1001}, and {1101}. Example 9-1 presents a more detailed example of a facet calculation.

This method finds all the facets of Φ by checking all combinations of m objects taken (n-1) at a time. This procedure greatly increases the efficiency of the Direct Allocation algorithm by computing the boundary of Φ. For a problem with 10 controls and 3 moments, there are 11,520 faces on Ω, but only 90 facets. Only 45 directions need to be searched to determine the boundary of Φ, because each search produces 2 facets. Once the boundary is determined, the facets are searched to determine the correct one.

A facet can be mathematically described using three vectors: $\mathbf{m}^*_i$, $\mathbf{m}^*_{i\text{-}j}$, and $\mathbf{m}^*_{i\text{-}k}$. $\mathbf{m}^*_i$ points from the origin to one of the vertices of the facet. $\mathbf{m}^*_{i\text{-}j}$ and $\mathbf{m}^*_{i\text{-}k}$ point from the i-th vertex to the two vertices of the facet to which the i-th vertex is connected. See Figure 9-2. A vector, $\mathbf{m}^*$, lies on the facet if the following conditions are satisfied:

$$\mathbf{m}^* = \mathbf{m}^*_i + b\mathbf{m}^*_{i\text{-}j} + c\mathbf{m}^*_{i\text{-}k} \tag{9-15}$$

$$0 \le b \le 1, 0 \le c \le 1 \tag{9-16}$$

A half-line in the direction of the desired moment intersects a given facet if the moment vector can be scaled to lie on the facet. Checking a facet involves computing the desired moment direction, and finding the intersection of this direction with the plane defined by equation 9-15.

$$\hat{\mathbf{m}}_d = \mathbf{m}_d/|\mathbf{m}_d| \tag{9-17}$$

$$a\hat{\mathbf{m}}_d = \mathbf{m}^*_i + b\mathbf{m}^*_{i\text{-}j} + c\mathbf{m}^*_{i\text{-}k} \tag{9-18}$$

$$a\hat{\mathbf{m}}_d - b\mathbf{m}^*_{i\text{-}j} - c\mathbf{m}^*_{i\text{-}k} = \mathbf{m}^*_i \tag{9-19}$$

$$[\hat{\mathbf{m}}_d - \mathbf{m}^*_{i\text{-}j} - \mathbf{m}^*_{i\text{-}k}]\begin{Bmatrix} a \\ b \\ c \end{Bmatrix} = \mathbf{m}^*_i \tag{9-20}$$

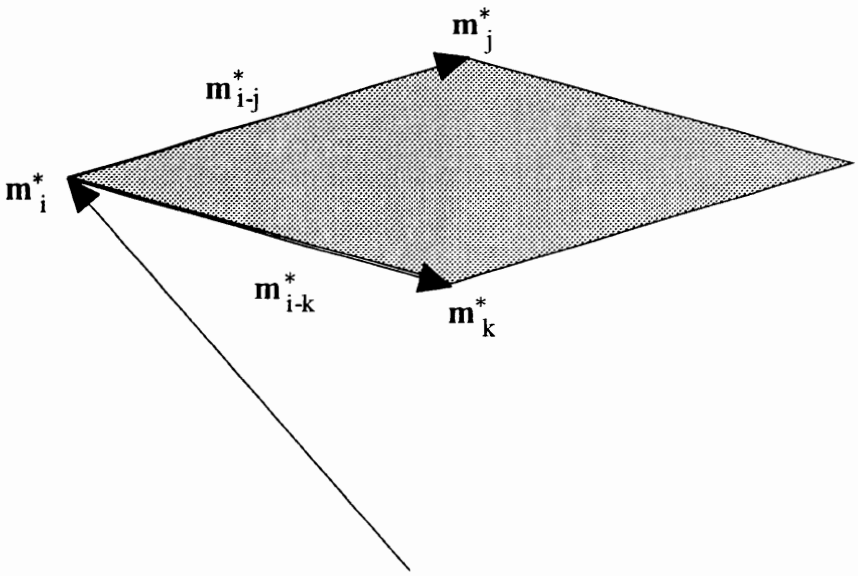


Figure 9-2: Example Facet

$$\begin{Bmatrix} a \\ b \\ c \end{Bmatrix} = [\hat{\mathbf{m}}_d - \mathbf{m}^*_{i-j} - \mathbf{m}^*_{i-k}]^{-1}\mathbf{m}^*_i \tag{9-21}$$

If $a > 0$ and the constraints of Equation 9-16 are satisfied, then a half-line in the direction of the desired moment intersects the facet being checked.

Once the correct facet is found, the controls can be determined. Each of the vertices of the facet have unique vectors of controls associated with them. The controls which generate the moment on the facet can be found by substituting the control vectors which generate the vertices, e.g. $B\mathbf{u}^*_i = \mathbf{m}^*_i$, into Equation 9-18.

$$a\hat{\mathbf{m}}_d = B\mathbf{u}^*_d = \mathbf{m}^*_i + b\mathbf{m}^*_{i-j} + c\mathbf{m}^*_{i-k} = B\mathbf{u}^*_i + bB\mathbf{u}^*_{i-j} + cB\mathbf{u}^*_{i-k} \tag{9-22}$$

$$\mathbf{u}^*_d = \mathbf{u}^*_i + b\mathbf{u}^*_{i-j} + c\mathbf{u}^*_{i-k} \tag{9-23}$$

The controls $\mathbf{u}^*_d$ correspond to the maximum attainable moment in the specified direction. If the moment is attainable ($a \geq |\mathbf{m}_d|$), then the controls which generate the desired moment are:

$$\mathbf{u}_d = \mathbf{u}^*_d|\mathbf{m}_d|/a \tag{9-24}$$

If the desired moment lies outside of the attainable moment subset ($a < |\mathbf{m}_d|$), then the controls which generate the maximum attainable moment in the desired direction are used, $\mathbf{u}^*_d$. Using these controls preserves the direction of the desired moment when the moment is unattainable.

Example 9-1: Determining $\partial(\Phi)$

Using the F-18 HARV data, the faces corresponding to the first 2 controls varying will be checked to see which are on the boundary. The columns of B corresponding to these controls are:

$$B_1 = \begin{Bmatrix} -4.382e\text{-}2 \\ -5.330e\text{-}1 \\ 1.100e\text{-}2 \end{Bmatrix} \tag{9-25}$$

$$B_2 = \begin{Bmatrix} 4.382e\text{-}2 \\ -5.330e\text{-}1 \\ -1.100e\text{-}2 \end{Bmatrix} \tag{9-26}$$

Solving for the transformation vector, **t**, yields:

$$\begin{Bmatrix} t_1 \\ t_2 \end{Bmatrix} = -\begin{bmatrix} -4.382e\text{-}2 & -5.330e\text{-}1 \\ 4.382e\text{-}2 & -5.330e\text{-}1 \end{bmatrix}^{-1} t_3 \begin{Bmatrix} 1.100e\text{-}2 \\ -1.100e\text{-}2 \end{Bmatrix} \tag{9-27}$$

For convenience, set $t_3 = 1$, yielding

$$\mathbf{t} = [-0.2510 \quad 0 \quad 1] \tag{9-28}$$

$$\mathbf{t}B = [0,\ 0,\ 0.0186,-0.0186,-0.0785,\ 0.0158,-0.0158,-0.0070,\ 7.5e\text{-}06,\ 0.1460] \tag{9-29}$$

By examining Equation 9-29, it can be seen that the object numbers of the facets of Φ corresponding to u_1 and u_2 varying are {2210010011} and {2201101100}. The vertices of the facet {2210010011} are: {0010010011}, {0110010011}, {1010010011}, {1110010011}.

Example 9-2: Checking a Facet

For the desired moment given in Equation 9-30, the facet {2210010011} identified in Example 9-1 will be checked.

$$\mathbf{m}_d = \begin{Bmatrix} -0.1105 \\ 0.2815 \\ 0.0965 \end{Bmatrix} \in \Phi \tag{9-30}$$

Any vertex of a facet may be used as $\mathbf{m}^*_i$. If {0010010011} is selected as $\mathbf{m}^*_i$, the vertices to which it connects, {0110010011} and {1010010011}, become $\mathbf{m}^*_j$ and $\mathbf{m}^*_k$.

$$\mathbf{u}_i = \begin{Bmatrix} -0.4189 \\ -0.4189 \\ 0.5236 \\ -0.5236 \\ -0.5236 \\ 0.7854 \\ -0.1396 \\ -0.5236 \\ 0.5236 \\ 0.5236 \end{Bmatrix}, \mathbf{u}_j = \begin{Bmatrix} -0.4189 \\ 0.1833 \\ 0.5236 \\ -0.5236 \\ -0.5236 \\ 0.7854 \\ -0.1396 \\ -0.5236 \\ 0.5236 \\ 0.5236 \end{Bmatrix}, \mathbf{u}_k = \begin{Bmatrix} 0.1833 \\ -0.4189 \\ 0.5236 \\ -0.5236 \\ -0.5236 \\ 0.7854 \\ -0.1396 \\ -0.5236 \\ 0.5236 \\ 0.5236 \end{Bmatrix} \tag{9-31}$$

$$\mathbf{m}^*_i = B\mathbf{u}_i = \begin{Bmatrix} -0.1381 \\ 0.6728 \\ 0.1206 \end{Bmatrix} \tag{9-32}$$

$$\mathbf{m}^*_j = B\mathbf{u}_j = \begin{Bmatrix} -0.1117 \\ 0.3519 \\ 0.1272 \end{Bmatrix} \tag{9-33}$$

$$\mathbf{m}^*_k = B\mathbf{u}_k = \begin{Bmatrix} -0.1645 \\ 0.3519 \\ 0.1140 \end{Bmatrix} \tag{9-34}$$

$$\mathbf{m}^*_{i\text{-}j} = \mathbf{m}^*_j - \mathbf{m}^*_i = \begin{Bmatrix} 0.0264 \\ 0.3210 \\ 0.0066 \end{Bmatrix} \tag{9-35}$$

$$\mathbf{m}^*_{i\text{-}k} = \mathbf{m}^*_k - \mathbf{m}^*_i = \begin{Bmatrix} -0.0264 \\ 0.3210 \\ -0.0066 \end{Bmatrix} \tag{9-36}$$

$$|\mathbf{m}_d| = 0.3174 \tag{9-37}$$

$$\hat{\mathbf{m}}_d = \begin{Bmatrix} -0.3480 \\ 0.8869 \\ 0.3039 \end{Bmatrix} \tag{9-38}$$

$$[\hat{\mathbf{m}}_d - \mathbf{m}^*_{i-j} - \mathbf{m}^*_{i-k}] \begin{Bmatrix} a \\ b \\ c \end{Bmatrix} = \mathbf{m}^*_i \tag{9-39}$$

$$\begin{Bmatrix} a \\ b \\ c \end{Bmatrix} = \begin{bmatrix} -0.3480 & -0.0264 & 0.0264 \\ 0.8869 & 0.3210 & 0.3210 \\ 0.3039 & -0.0066 & 0.0066 \end{bmatrix}^{-1} \begin{Bmatrix} -0.1381 \\ 0.6728 \\ 0.1206 \end{Bmatrix} = \begin{Bmatrix} 0.3968 \\ 0.5000 \\ 0.5000 \end{Bmatrix} \tag{9-40}$$

Because $a > 0$, $0 \le b \le 1$, and $0 \le c \le 1$, the facet checked is the correct facet. Because $a \ge |\mathbf{m}_d|$, the desired moment is attainable. The controls which produce the moment on the boundary are:

$$\mathbf{u}^*_d = \mathbf{u}^*_i + b\mathbf{u}^*_{i-j} + c\mathbf{u}^*_{i-k} = \{-0.1178,-0.1178, 0.5236,-0.5236,-0.5236,$$

$$0.7854,-0.1396,-0.5236, 0.5236, 0.5236\}^T \tag{9-41}$$

The controls which produce the desired moment are:

$$\mathbf{u}_d = \mathbf{u}^*_d|\mathbf{m}_d|/a = \{-0.0942,-0.0942, 0.4189,-0.4189,-0.4189,$$

$$0.6283,-0.1117,-0.4189, 0.4189, 0.4189\}^T \tag{9-42}$$

Unattainable Moments

If a desired moment is unattainable using this method, then it is unattainable using any method because Direct Allocation can allocate admissible controls to produce all of the attainable moments. When unattainable moments are commanded, the moment produced by this method will have the same direction as the desired moment.

Computational Considerations

The most computationally expensive part of Direct Allocation is the computing and searching the facets. By searching only the facets of Φ, instead of searching all the faces of Ω, the computational time is greatly reduced. Current implementations of Direct Allocation have a significant computational savings over the Null-Space Intersection method.

Initially, Direct Allocation algorithms were constructed to find all the facets of Φ and then search through them until the correct facet was found. It is possible to check the facets as they are found and stop searching when the correct facet is computed. Stopping the search once the correct facet is found can decrease the average number of computations needed to find the controls. However, in a worst case scenario, all of the facets will need to be computed and checked. If implementing Direct Allocation in a real-time digital computer that performs control law calculations in a discrete time frame, it is necessary for the computer to be able search all of the facets within the limits of the time frame being used. Currently, research is being conducted to discover faster techniques for finding the correct facet, such as the use of parallel processing.

Combining with Tailored Generalized Inverses

Direct Allocation can be used to specify a generalized inverse. In Section 5, the method of Tailoring a generalized inverse was mentioned. By using a method like Direct Allocation to determine a control solution for the maximum attainable moment in a specified direction, a generalized inverse can be made to allocate admissible controls for all attainable moments which lie in that direction. For a general case of an n-D moment space, n directions may be picked.

Recall that P can be partitioned into two matrices:

$$P = \begin{bmatrix} P_1 \\ P_2 \end{bmatrix}, P_1 \in R^{nxn}, P_2 \in R^{(m-n)xn} \tag{9-43}$$

Specifying P_2 is sufficient to determine P.

$$P_1 = B_1^{-1} - B_1^{-1}B_2P_2 \tag{9-44}$$

P_2 can be specified if control solutions are known for n linearly independent moment directions. Consider 3 linearly independent moment directions ($\mathbf{m}^*_1, \mathbf{m}^*_2, \mathbf{m}^*_3$) and their corresponding controls ($\mathbf{u}^*_1, \mathbf{u}^*_2, \mathbf{u}^*_3$). If a generalized inverse is tailored to fit these points, then it will be true that:

$$\begin{bmatrix} P_1 \\ P_2 \end{bmatrix} \mathbf{m}^*_i = \mathbf{u}^*_i = \begin{Bmatrix} \mathbf{u}^*_{i,1} \\ \mathbf{u}^*_{i,2} \end{Bmatrix} \text{ for } i = 1,2,3 \tag{9-45}$$

Equation 9-43 is used to specifyP_2:

$$P_2\mathbf{m}^*_i = \mathbf{u}^*_{i,2} \text{ for } i = 1,2,3 \tag{9-46}$$

$$\mathbf{m}^{*T}_i P_2^T = \mathbf{u}^{*T}_{i,2} \text{ for } i = 1,2,3 \tag{9-47}$$

$$\begin{bmatrix} \mathbf{m}^{*T}_1 \\ \mathbf{m}^{*T}_2 \\ \mathbf{m}^{*T}_3 \end{bmatrix} P_2^T = \begin{bmatrix} \mathbf{u}^{*T}_{1,2} \\ \mathbf{u}^{*T}_{2,2} \\ \mathbf{u}^{*T}_{3,2} \end{bmatrix} \tag{9-48}$$

$$P_2^T = \begin{bmatrix} \mathbf{m}^{*T}_1 \\ \mathbf{m}^{*T}_2 \\ \mathbf{m}^{*T}_3 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{u}^{*T}_{1,2} \\ \mathbf{u}^{*T}_{2,2} \\ \mathbf{u}^{*T}_{3,2} \end{bmatrix} \tag{9-49}$$

The matrix in Equation 9-47 will be invertible as long as the moment directions are linearly independent.

This method of tailoring does not require the use of maximum attainable moments, $\mathbf{m}^*$. Any linearly independent set of moments and corresponding controls may be used. However, using the maximum moments insures that the maximum attainable moment in the desired directions will be attainable using the tailored generalized inverse.

The major drawback to tailoring a generalized inverse is that the volume of attainable moments can be very small. Additionally, tailoring a generalized inverse to a certain moment direction does not guarantee the maximum attainable moment in the opposite direction. This is demonstrated in Example 9-3.

Example 9-3: Tailoring a Generalized Inverse

Using the F-18 HARV data, a generalized inverse is tailored to attain the maximum moment generating capabilities in the three positive moment directions:

$$\mathbf{m}^*_1 = \begin{Bmatrix} 0.172133 \\ 0 \\ 0 \end{Bmatrix}, \mathbf{m}^*_2 = \begin{Bmatrix} 0 \\ 0.798429 \\ 0 \end{Bmatrix}, \mathbf{m}^*_3 = \begin{Bmatrix} 0 \\ 0 \\ 0.127521 \end{Bmatrix} \tag{9-50}$$

$$\mathbf{u}^*_{1,1} = \{-0.4189,\ 0.1833,-0.5236\}^T \tag{9-51}$$

$$\mathbf{u}^*_{1,2} = \{\ 0.5236,\ 0.5236,-0.1396,\ 0.7854,\ 0.5236,-0.466765,\ 0.243852\}^T \tag{9-52}$$

$$\mathbf{u}^*_{2,1} = \{-0.4189,-0.4189,-0.5236\}^T \tag{9-53}$$

$$\mathbf{u}^*_{2,2} = \{-0.5236,-0.5236,\ 0.7854,\ 0.7854,\ 0.389970,\ 0.5236,-0.262729\}^T \tag{9-54}$$

$$\mathbf{u}^*_{3,1} = \{-0.4189,\ 0.1833,\ 0.5236\}^T \tag{9-55}$$

$$\mathbf{u}^*_{3,2} = \{-0.5236,-0.5236,-0.1396,\ 0.226995,\ 0.5236,-0.368797,\ 0.5236\}^T \tag{9-56}$$

$$P_2 = \begin{bmatrix} 0.172133 & 0 & 0 \\ 0 & 0.798429 & 0 \\ 0 & 0 & 0.127521 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{u}^*_{1,2}{}^T \\ \mathbf{u}^*_{2,2}{}^T \\ \mathbf{u}^*_{3,2}{}^T \end{bmatrix} \tag{9-57}$$

Substituting P_2 into Equation 9-42, produces a final P:

$$P_{Tailored} = \begin{bmatrix} -2.433e{+}0 & -5.246e{-}1 & -3.284e{+}0 \\ 1.064e{+}0 & -5.246e{-}1 & 1.437e{+}0 \\ -3.041e{+}0 & -6.557e{-}1 & 4.105e{+}0 \\ 3.041e{+}0 & -6.557e{-}1 & -4.105e{+}0 \\ 3.041e{+}0 & -6.557e{-}1 & -4.105e{+}0 \\ -8.109e{-}1 & 9.836e{-}1 & -1.094e{+}0 \\ 4.562e{+}0 & 9.836e{-}1 & 1.780e{+}0 \\ 3.041e{+}0 & 4.884e{-}1 & 4.105e{+}0 \\ -2.711e{+}0 & 6.557e{-}1 & -2.892e{+}0 \\ 1.416e{+}0 & -3.290e{-}1 & 4.105e{+}0 \end{bmatrix} \tag{9-58}$$

The Π for this generalized inverse fills only 13.18% of Φ. Figure 9-3 shows $\Pi_{Tailored\ GI}$ inside a wire-frame of Φ. From Figure 9-3, it can be seen that the maximum moment in the {0,-1,0} direction is unattainable using $\Pi_{Tailored\ GI}$, even though the maximum moment in the {0,1,0} is attainable.

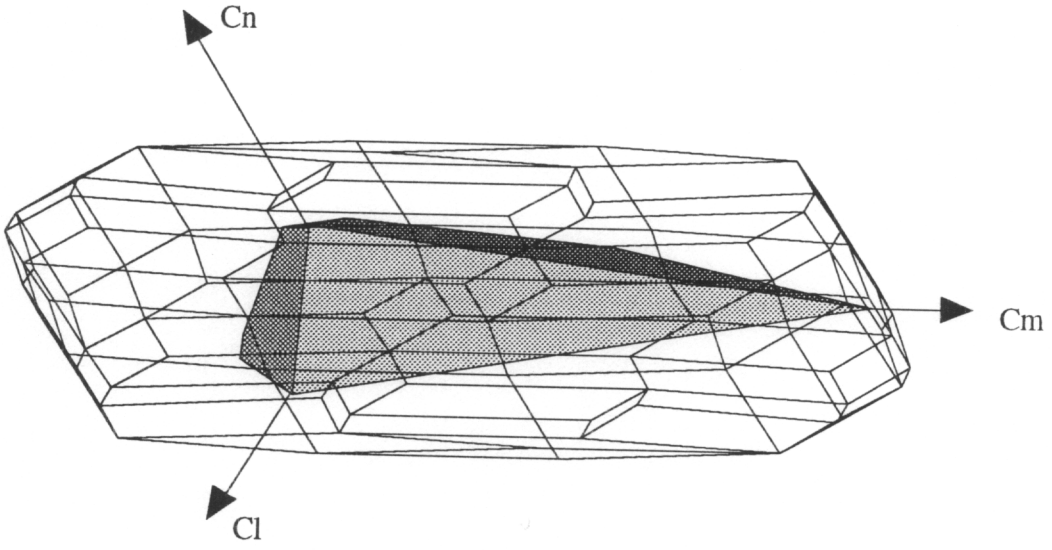


Figure 9-3: $\Pi_{Tailored\ GI}$ from Example 9-3

Example 9-4: Tailoring to a Set of Desired Moments

A practical application of tailoring generalized inverses can be found in its use as a research tool. If attainable desired moments are known in advance for a particular flight maneuver, and they are grouped close together, it may be possible to tailor a generalized inverse which will allocate admissible controls for the entire maneuver. In this way a single generalized inverse may be used to allocate admissible controls when examining a specific problem.

For example, consider a velocity vector roll. This maneuver is a first-order response to a commanded roll acceleration about the instantaneous velocity vector with constant angle of attack and no side slip. The program Rollerator [31] was used to compute the desired moments for the F-18 HARV performing a velocity vector roll with a steady state roll rate of 215°/s and a time constant of 1 second. Figure 9-4 shows these moments plotted as circles inside a wire frame of the Φ. These moments are all grouped rather closely together, and it is possible to tailor a single generalized inverse which will allocate admissible controls for the entire maneuver. If the desired moment directions are:

$$\mathbf{m}_1 = \{0.054705, -0.433686, 0.064673\}^T \tag{9-59}$$

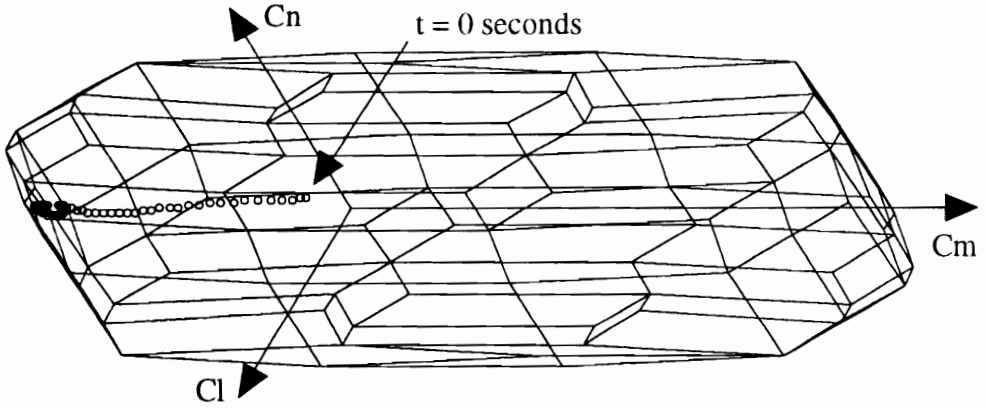
$$\mathbf{m}_2 = \{0.067862, -0.433686, 0.064673\}^T \tag{9-60}$$

$$\mathbf{m}_3 = \{0.067863, -0.00727, 0.09070\}^T \tag{9-61}$$

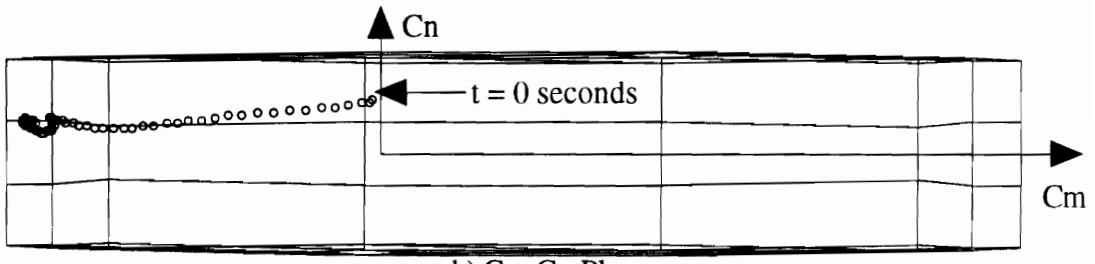
the tailored generalized inverse is:

$$P_{Tailored\ GI} = \begin{bmatrix} 3.606e\text{-}1 & -9.318e\text{-}1 & -3.731e{+}0 \\ 3.606e\text{-}1 & -1.969e\text{-}1 & 1.196e{+}0 \\ 1.030e{+}0 & -5.626e\text{-}1 & 3.417e{+}0 \\ 1.030e{+}0 & -6.400e\text{-}1 & 2.899e{+}0 \\ 2.063e{+}0 & -1.014e{+}0 & -5.858e{+}0 \\ -2.746e\text{-}1 & 1.500e\text{-}1 & -9.112e\text{-}1 \\ 1.445e{+}1 & 3.306e\text{-}1 & -4.438e{+}0 \\ 1.030e{+}0 & -5.626e\text{-}1 & 3.417e{+}0 \\ -1.030e{+}0 & 8.171e\text{-}1 & -1.711e{+}0 \\ 1.030e{+}0 & -5.626e\text{-}1 & 3.417e{+}0 \end{bmatrix} \tag{9-62}$$

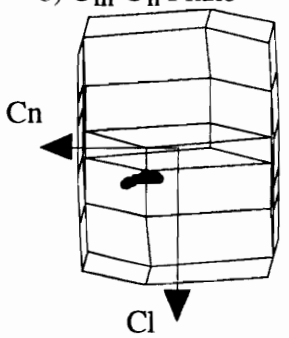
This Π fills only 7.69% of Φ. However, because all the desired moments lie inside of Π, this generalized inverse will allocate admissible controls for the entire maneuver. Figure 9-5 shows this Π inside a wire-frame of Φ.



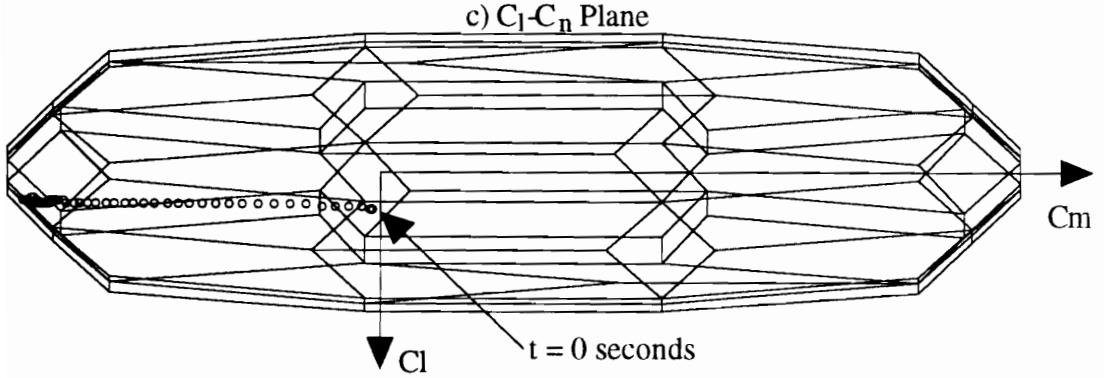
a) 3-D View



b) C_m-C_n Plane

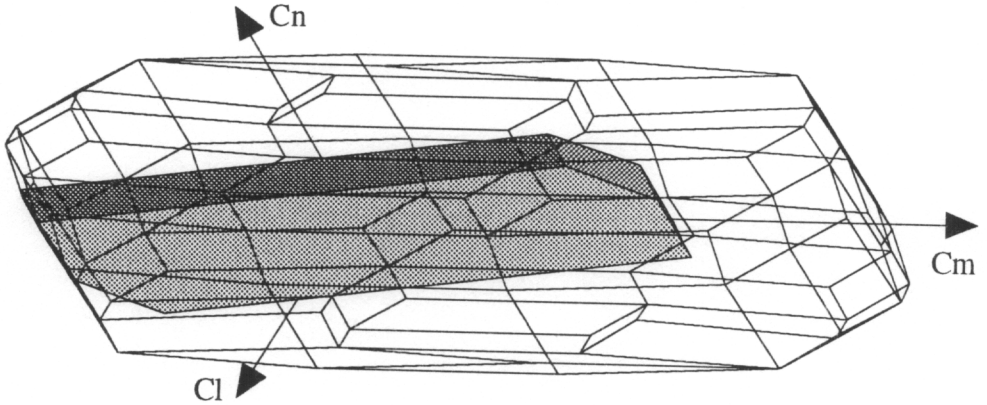


c) C_l-C_n Plane

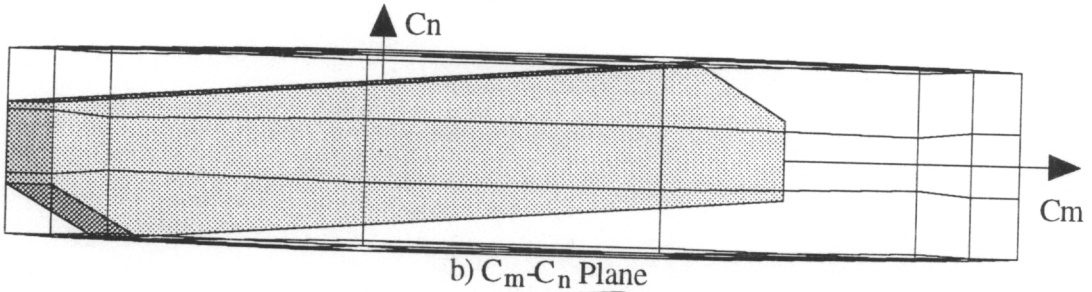


d) C_l-C_m Plane

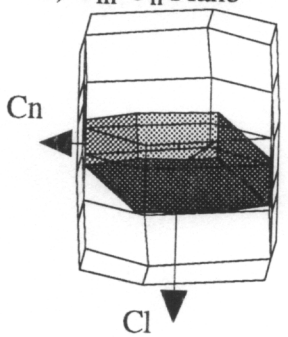
Figure 9-4: Desired Moments for a Velocity Vector Roll



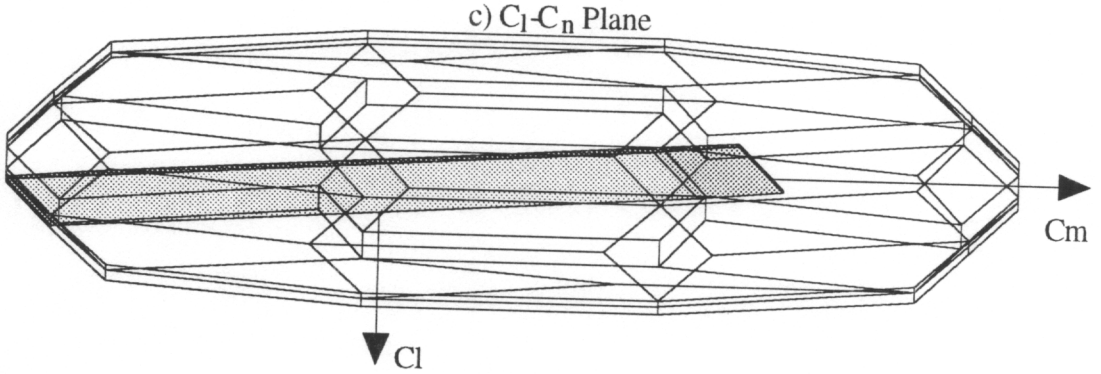
a) 3-D View



b) C_m-C_n Plane



c) C_l-C_n Plane



d) C_l-C_m Plane

Figure 9-5: $\Pi_{Tailored\ GI}$ Inside of Φ

10. LINEAR COMBINATIONS OF SOLUTIONS

Description of Method

Due to the fact that $\mathbf{m} = B\mathbf{u}$ is a linear equation, two or more different solutions to this equation can be linearly combined to yield new solutions. For example, the control vectors $\mathbf{u_1}$ and $\mathbf{u_2}$ can be added together.

$$B\mathbf{u_1} = \mathbf{m} \tag{10-1}$$

$$B\mathbf{u_2} = \mathbf{m} \tag{10-2}$$

$$B(c_1\mathbf{u_1} + c_2\mathbf{u_2}) = c_1B\mathbf{u_1} + c_2B\mathbf{u_2} = (c_1 + c_2)\mathbf{m} = \mathbf{m} \quad \text{for } c_1 + c_2 = 1 \tag{10-3}$$

Or, more generally,

$$B\Sigma c_i\mathbf{u_i} = \mathbf{m} \quad \text{for } \Sigma c_i = 1 \tag{10-4}$$

In this way, various aspects of different solutions can be combined. The combined controls are guaranteed to be admissible if the following conditions are met.

Each of the control solutions to be combined is admissible:

$$\mathbf{u_i} \in \Omega \ \forall \ i \tag{10-5}$$

The sum of the weighting terms is equal to one,

$$\Sigma c_i = 1 \tag{10-6}$$

Each of the weighting terms is between zero and one,

$$0 \le c_i \le 1 \tag{10-7}$$

If these conditions are met, then the combined controls will be admissible,

$$\mathbf{u} = \Sigma c_i \mathbf{u_i} \in \Omega \tag{10-8}$$

When the controls are combined in this manner, each control, u_i, will range between the smallest value in any of the $\mathbf{u_i}$ (when the corresponding $c_i = 1$ and the rest of the $c_i = 0$) and the largest value in any of the $\mathbf{u_i}$. For example, if two solutions are combined and the smallest value of u_i is $u_{i\,2} = -5$ and the largest value of u_i is $u_{i\,1} = 5$, the relationship between u_i, c_1 and c_2 can be seen in Figure 10-1.

If one or more of the solutions contains admissible controls, scalar weighting terms can be found to produce an admissible final solution. If the weighting on the admissible solution is one and the other weightings are zero, then the solution is admissible.

Consider combining the two solutions in Equation 10-3 when only $\mathbf{u_2}$ is admissible:

$$B\mathbf{u_1} = \mathbf{m} \tag{10-9}$$

$$B\mathbf{u_2} = \mathbf{m} \tag{10-10}$$

$$\mathbf{u_1} \notin \Omega, \mathbf{u_2} \in \Omega \tag{10-11}$$

To find the scalars c_1 and c_2 which scale $c_1\mathbf{u_1} + c_2\mathbf{u_2}$ so that it is on the boundary of Ω, examine each control. If u_i is within its limits for each $\mathbf{u_i}$, then as long as $0 \le c_1 \le 1$ and $c_1 + c_2 = 1$, u_i will be within its limits.

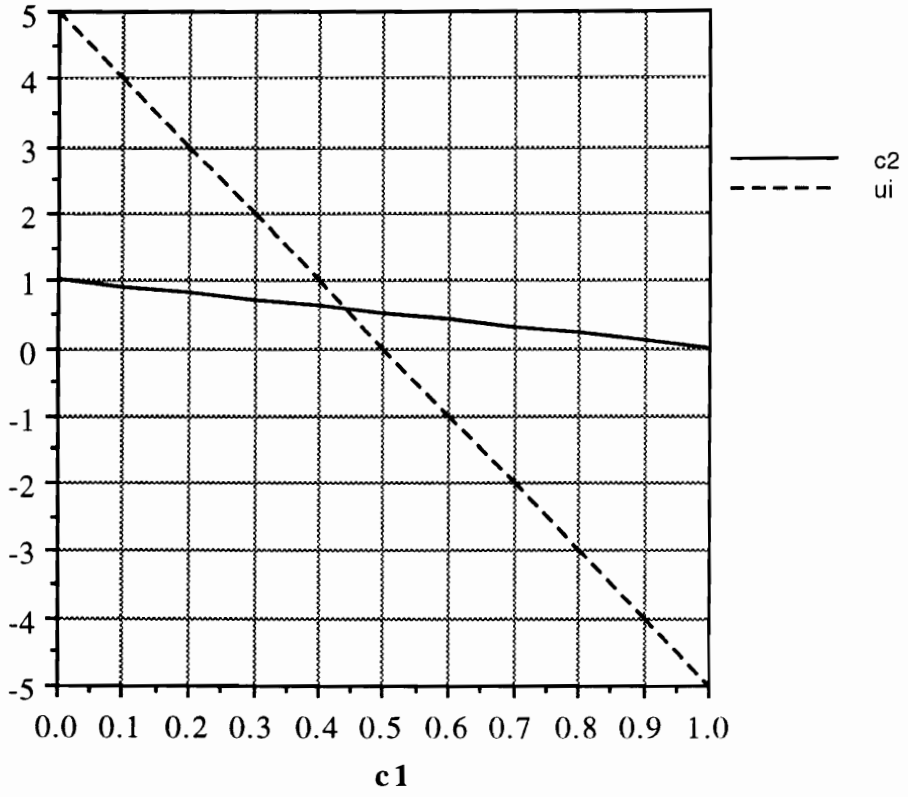


Figure 10-1: u_i Varies Between the Minimum and Maximum Values

If u_i exceeds a limit, it is possible to compute scale factors, a_i and b_i, which will make the sum of the control solutions equal to the limit:

$$a_i u_{i\,1} + b_i u_{i\,2} = u^*_i \tag{10-12}$$

Where u^*_i is the limit which u_{il} violates and,

$$b_i = 1 - a_i \tag{10-13}$$

Substituting into Equation 10-12,

$$a_i u_{i\,1} + (1-a_i) u_{i\,2} = u^*_i \tag{10-14}$$

Then solving for a_i,

$$a_i(u_{i\,1} - u_{i\,2}) = u^*_i - u_{i\,2} \tag{10-15}$$

$$a_i = \frac{u^*_i - u_{i\,2}}{u_{i\,1} - u_{i\,2}} \tag{10-16}$$

Because $u_{i\,1}$ is inadmissible and $u_{i\,2}$ is admissible:

$$|u_{i\,1}| > |u^*_i| \geq |u_{i\,2}|, \text{ and } 0 \leq a_i < 1 \tag{10-17}$$

After computing a_i for each of saturated controls , selecting the a_i which is smallest, $c_1 = \text{Min}\{a_i\}$, and $c_2 = 1 - c_1$ will insure that all of the final controls are at or within their limits. Note that if $\mathbf{u_2}$ is on the boundary of Ω, $c_1 = 0$ and $c_2 = 1$.

Example 10-1: Combining Direct Allocation and Daisy Chaining

This example combines the methods of Direct Allocation and Daisy Chaining to provide a method which meets the Daisy Chaining constraint that a subset of the controls

are used only when the primary controls fail to generate the desired moment, and also allocates admissible controls for all the attainable moments. Using the F-18 HARV data, consider the ailerons, horizontal tails and rudders as the primary controls, as done in Example 6-2.

$$B_1 = \begin{bmatrix} \text{-4.382e-2} & \text{4.382e-2} & \text{-5.841e-2} & \text{5.841e-2} & \text{1.674e-2} \\ \text{-5.330e-1} & \text{-5.330e-1} & \text{-6.486e-2} & \text{-6.486e-2} & \text{0.000e+0} \\ \text{1.100e-2} & \text{-1.100e-2} & \text{3.911e-3} & \text{-3.911e-3} & \text{-7.428e-2} \end{bmatrix} \tag{10-18}$$

The pseudo-inverse of B_1 will be used to allocate the control vector $\mathbf{u_1}$.

$$P_1 = B_1^T[B_1B_1^T]^{-1} = \begin{bmatrix} \text{-4.167e+0} & \text{-9.243e-1} & \text{-2.429e+0} \\ \text{4.167e+0} & \text{-9.243e-1} & \text{2.429e+0} \\ \text{-5.337e+0} & \text{-1.124e-1} & \text{-3.254e-3} \\ \text{5.337e+0} & \text{-1.124e-1} & \text{3.254e-3} \\ \text{6.722e-1} & \text{0.000e+0} & \text{-1.274e+1} \end{bmatrix} \tag{10-19}$$

This control solution will not use the secondary controls. To preserve this quality, P_1 is augmented with a matrix of zeros to form P.

$$P = \begin{bmatrix} P_1 \\ [0] \end{bmatrix} = \begin{bmatrix} \text{-4.167e+0} & \text{-9.243e-1} & \text{-2.429e+0} \\ \text{4.167e+0} & \text{-9.243e-1} & \text{2.429e+0} \\ \text{-5.337e+0} & \text{-1.124e-1} & \text{-3.254e-3} \\ \text{5.337e+0} & \text{-1.124e-1} & \text{3.254e-3} \\ \text{6.722e-1} & \text{0.000e+0} & \text{-1.274e+1} \\ \text{0.000e+0} & \text{0.000e+0} & \text{0.000e+0} \\ \text{0.000e+0} & \text{0.000e+0} & \text{0.000e+0} \\ \text{0.000e+0} & \text{0.000e+0} & \text{0.000e+0} \\ \text{0.000e+0} & \text{0.000e+0} & \text{0.000e+0} \\ \text{0.000e+0} & \text{0.000e+0} & \text{0.000e+0} \end{bmatrix} \tag{10-20}$$

$$\mathbf{u_1} = P\mathbf{m} \tag{10-21}$$

The second control vector, $\mathbf{u_2}$, is computed using Direct Allocation.

$$\mathbf{u_2} \in \Omega \,\forall\, \mathbf{m} \in \Phi \tag{10-22}$$

When the desired moment is attainable using $\mathbf{u_1}$, $c_1 = 1$ and $c_2 = 0$. Using these weightings insures that only the primary controls are used to generate these moments. For example, consider the desired moment:

$$\mathbf{m_d} = \begin{Bmatrix} -0.0262 \\ 0.1278 \\ 0.0229 \end{Bmatrix} \in \Phi \tag{10-23}$$

The controls in $\mathbf{u_1}$ are allocated as follows:

$$P\mathbf{m_d} = \mathbf{u_1} = \{-0.0645, -0.1718, 0.1256, -0.1543, -0.3096, 0, 0, 0, 0, 0\}^T \tag{10-24}$$

Since all of the controls in $\mathbf{u_1}$ are admissible, only $\mathbf{u_1}$ will be used.

$$\mathbf{u_1} \in \Omega \tag{10-25}$$

$$c_1 = 1, c_2 = 0 \Rightarrow \mathbf{u} = \mathbf{u_1} \tag{10-26}$$

The controls in $\mathbf{u_1}$ will not be admissible for all the attainable moments, as shown in Section 6. If the desired moment is unattainable using $\mathbf{u_1}$, c_1 and c_2 are found using Equations 10-12 through 10-16. When the desired moment is

$$\mathbf{m_d} = \begin{Bmatrix} -0.0815 \\ 0.3970 \\ 0.0711 \end{Bmatrix} \in \Phi \tag{10-27}$$

the controls in $\mathbf{u_1}$ are,

$$\mathbf{u_1} = \{-0.2003, \mathbf{-0.5336}, 0.3899, -0.4792, \mathbf{-0.9614}, 0, 0, 0, 0, 0\}^T \tag{10-28}$$

The left horizontal tail, u_2, and the rudders, u_5, are saturated. The controls allocated using Direct Allocation are:

$$\mathbf{u_2} = \{-0.2471,-0.2472,\ 0.3089,-0.3089,-0.3089,$$
$$0.4634,-0.0824,-0.3089,\ 0.3089,\ 0.3089\}^T \quad (10\text{-}29)$$

The controls of interest are u_2 and u_5, because they are commanded to exceed their limits in $\mathbf{u_1}$. Table 10-1 shows the scale factors for this moment computed using Equation 10-16.

The smallest a_i is 0.3290. This value should be used as c_1, because if 0.5996 is used, u_5 will remain saturated.

$$u_5 = 0.5996(-0.9614) + 0.4004(-0.3089) = -0.7001 < -0.5326 \quad (10\text{-}30)$$

Using $c_1 = 0.3290$, the final controls are:

$$\mathbf{u} = 0.3290\mathbf{u_1} + 0.6710\mathbf{u_2} = \{-0.2318,-0.3414,\ 0.3356,-0.3649,-0.5236,$$
$$0.3109,-0.0553,-0.2073,\ 0.2073,\ 0.2073\}^T \quad (10\text{-}31)$$

Note that all the controls in Equation 10-31 are admissible.

Figure 10-2 shows a set of desired moments which increase in magnitude until the maximum moment in that direction is achieved. Figure 10-3 shows the controls allocated using this method for these desired moments. Note that the controls in the second group, Figure 10-3b, are not used until the rudders in $\mathbf{u_1}$ saturate.

Table 10-1: Scaling factors

i	u_{i1}	u_{i2}	u^*_i	a_i
2	-0.5336	-0.2472	-0.4189	0.5996
5	-0.9614	-0.3089	-0.5236	0.3290

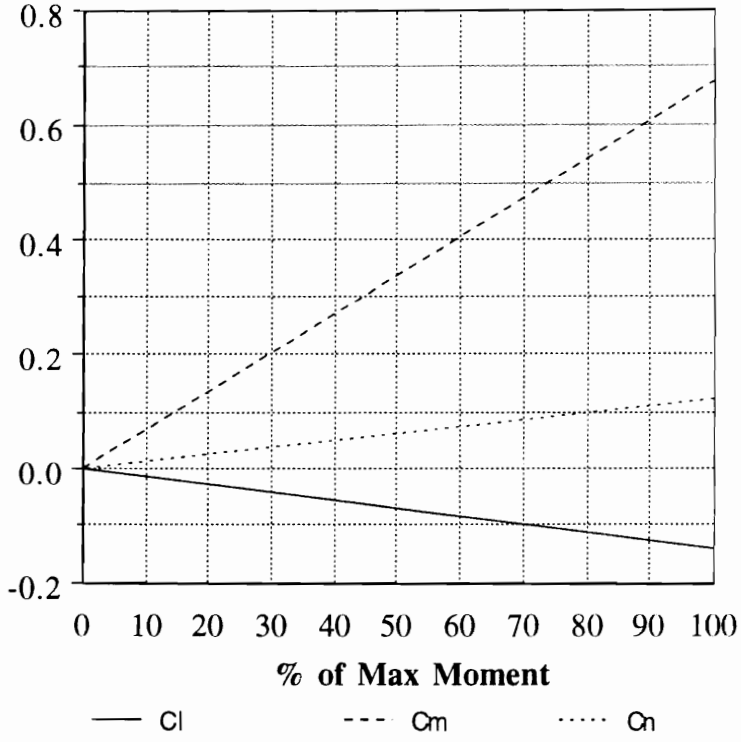
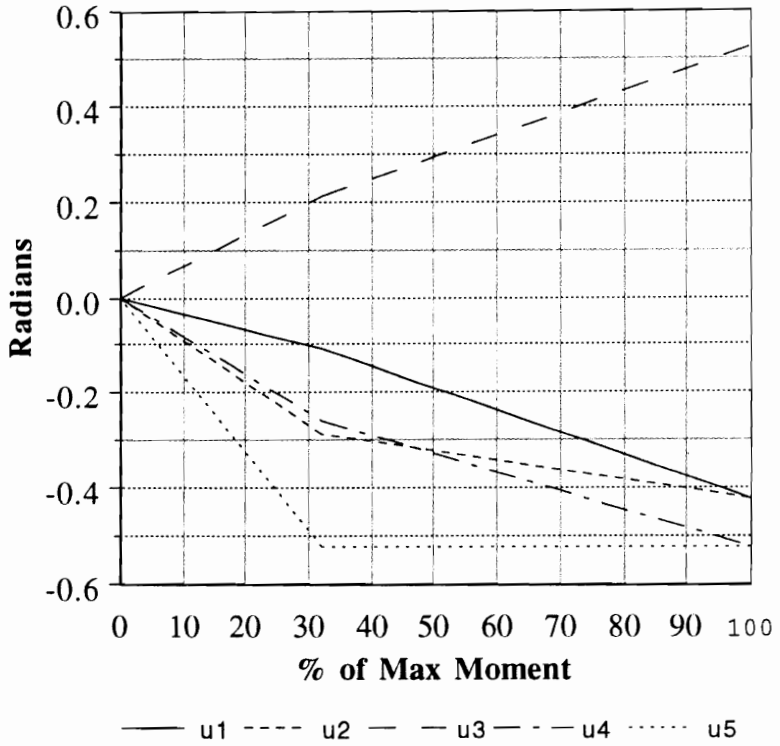
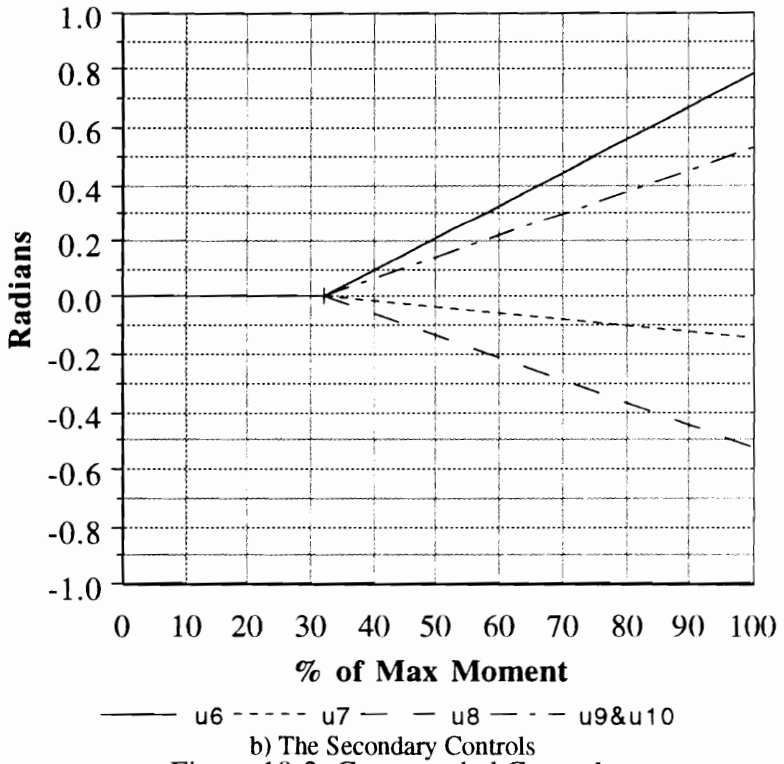


Figure 10-2: Desired Moments



a) The Primary Controls



b) The Secondary Controls

Figure 10-3: Commanded Controls

11. PERFORMANCE

Relating Π to Performance

To illustrate the importance of control allocation, the moment generating capability of a control allocation scheme will be related to the aircraft's ability to perform a maneuver. The F-18 HARV data is used as an example aircraft. The example maneuver is a roll through 30° taken from military specifications, MIL-STD-1797A [34] used in conjunction with MIL-F-8785C [35]. Using the military specifications [34,35], the F-18 HARV data corresponds to a Class IV, Category A vehicle flying at low speed. For this aircraft and flight condition, a roll through 30° should be performed in 1.1s with a maximum roll mode time constant of 1.0s to meet the Level 1 specification.

The moments for the desired maneuver are generated using the program Rollerator [31]. To specify the velocity vector roll, the program Rollerator requires the user to input the time constant and the maximum steady state roll rate, P_{ss}. Using a time constant of 1.0s, the maximum steady state roll rate was increased until a particular control allocation scheme saturated controls. In this way, the fastest velocity vector roll for which a given control allocation scheme will not saturate any controls was found. Table 11-1 shows the results for Direct Allocation, the pseudo-inverse, the Best Generalized Inverse from Example 5-3, and the Daisy Chaining scheme detailed in Example 6-2. By effectively allocating the controls, an aircraft can achieve greater performance from its existing controls. Direct Allocation, which allocates admissible controls for the entire AMS, enables the aircraft to perform the maneuver in the fastest time, 0.56s. Using the other methods, the aircraft cannot perform the maneuver as quickly, taking as long as 0.86s using the pseudo-inverse.

Table 11-1: Results Summary

Allocation Method	Volume of Π (Percentage of Φ)	Maximum P_{ss}	Time Through 30°
Direct Allocation	100%	$215s^{-1}$	0.56s
Pseudo-Inverse	13.7%	$105s^{-1}$	0.86s
Best GI	42.7%	$156s^{-1}$	0.68s
Daisy Chaining	≈42.3%	$134s^{-1}$	0.76s

The maximum steady state roll rate which can be achieved using Direct Allocation is significantly higher than the maximum steady state roll rate the other methods can achieve, and is more than double the maximum steady state roll rate that the pseudo-inverse can achieve. Even though the Best Generalized Inverse solution and the Daisy Chaining solution allocate admissible controls for a similar percentage of the AMS (42.7% compared to 42.3%), the Best Generalized Inverse is able to perform the maneuver significantly faster (0.68s compared to 0.76s) because a greater portion of $\Pi_{Best\ GI}$ is coincident with the region of moment space in which the desired moments lie.

Weight Reduction

All of the allocation methods compared in Table 11-1 satisfy the Level 1 requirement. In fact, the controls on this aircraft are capable of providing performance which far exceeds the requirement. Using Direct Allocation, it is possible to accomplish the maneuver in nearly half the required time.

The fact that the aircraft has greater capabilities than are required raises the issue of control sizing. When designing an aircraft, the desire to improve performance tends to increase the size and number of the controls. The desire to decrease weight and reduce complexity calls for a reduction in the size and number of controls. A control allocation scheme which maximizes Π will allow aircraft designers to meet performance requirements using fewer or smaller controls. Using fewer or smaller controls can represent a savings in weight and a reduction in drag.

12. DISCRETE TIME DIRECT ALLOCATION

The Effects of Time

In addition to the constraints imposed by the physical limits on the positions of the controls, there are other factors which significantly effect the performance of a control allocation scheme. Among these factors are: rate limiting, time varying B matrices, optimization issues, and reconfiguration in the event of control failure. Time is a key element in all of these factors. The problem statement in Section 4 makes no mention of time and examines the equation $\mathbf{m} = B\mathbf{u}$ only at an instant. In this section, the effects of time are examined, particularly the speed at which the controls are commanded to move, $\dot{\mathbf{u}}$, and the effect of changing control effectiveness as a function of time, $B(t)$.

The systems which drive the control actuators, typically hydraulic systems, do not have an infinite amount of power available. As a result, there is a limit to the speed at which the controls can move, called the rate limit. If the moment produced by the controls is to be equal to the desired moment, the controls must be allocated in a manner that will not command the controls to move faster than their capabilities permit.

In real world applications, the B matrix is a function of time because changes in control positions and flight conditions can change control effectiveness. These changes can cause the moment produced to differ from the moment desired. The magnitude of these errors can be quite significant, especially for parts of the flight envelope with a high angle-of-attack and low Mach number.

Description of Method

The method of Discrete Time Direct Allocation arose from implementing control allocation methods on a digital computer. In a digital computer, the desired moments are commanded in a sequence and separated in time by some discrete amount, Δt. In previous sections, the control allocation problem examined was treated as globally linear. Figure 12-1 shows an example of global linearization. The value of the slope of the dashed line in Figure 12-1 corresponds to the (j,i) element of B, 15. Discrete Time Direct Allocation treats the problem as being locally linear, with commanded moments separated in time by the amount Δt. In Figure 12-2, the dashed line shows the derivative of m_j with respect to u_i evaluated at $u_i = 15$. This local slope is approximately 10.

In addition to the position constraints on the controls ($\mathbf{u} \in \Omega$) the rate limits of the controls will be considered:

$$\dot{u}_{i\,Min} \leq \dot{u}_i \leq \dot{u}_{i\,Max} \tag{12-1}$$

In Section 4, the moments were expressed as a function of the controls and the states:

$$\mathbf{m} = \mathbf{f}(\mathbf{u},\mathbf{x}) \tag{12-2}$$

This equation can be approximated by a first order Taylor series:

$$\Delta\mathbf{m} \cong \left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_{ref} \Delta\mathbf{u} + \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{ref} \Delta\mathbf{x} \tag{12-3}$$

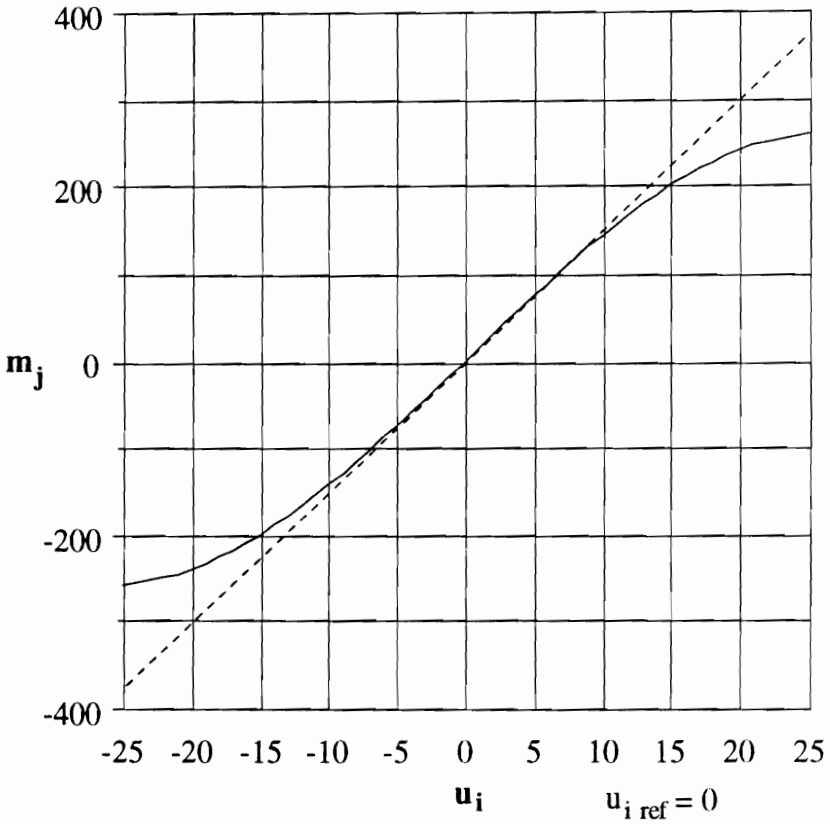


Figure 12-1: Globally Linear Approximation

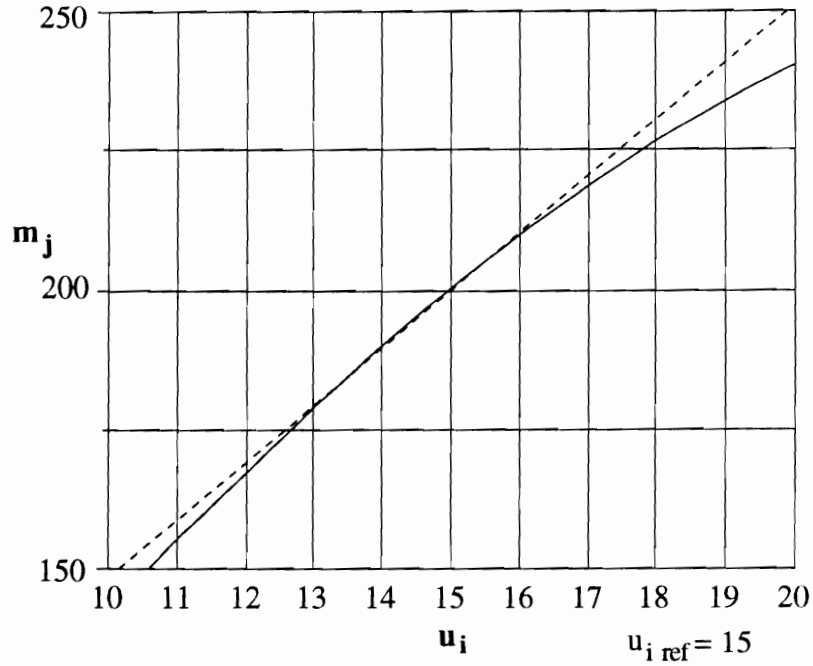


Figure 12-2: Locally Linear Approximation

Which may also be written as,

$$\Delta \mathbf{m} \cong \Delta \mathbf{m}_{\mathbf{u}} + \Delta \mathbf{m}_{\mathbf{x}} \tag{12-4}$$

where $\Delta \mathbf{m}_{\mathbf{u}}$ is the change in the total moment generated by the controls. The control effectiveness matrix is defined as the partial derivatives of $\mathbf{f}$ with respect to the controls.

$$B_{\text{ref}} \equiv \left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_{\text{ref}} = \left[\begin{array}{cccc} \frac{\partial \mathbf{f}_1(\mathbf{u},\mathbf{x})}{\partial \mathbf{u}_1} & \frac{\partial \mathbf{f}_1(\mathbf{u},\mathbf{x})}{\partial \mathbf{u}_2} & \cdots & \frac{\partial \mathbf{f}_1(\mathbf{u},\mathbf{x})}{\partial \mathbf{u}_m} \\ \frac{\partial \mathbf{f}_2(\mathbf{u},\mathbf{x})}{\partial \mathbf{u}_1} & \frac{\partial \mathbf{f}_2(\mathbf{u},\mathbf{x})}{\partial \mathbf{u}_2} & \cdots & \frac{\partial \mathbf{f}_2(\mathbf{u},\mathbf{x})}{\partial \mathbf{u}_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \mathbf{f}_n(\mathbf{u},\mathbf{x})}{\partial \mathbf{u}_1} & \frac{\partial \mathbf{f}_n(\mathbf{u},\mathbf{x})}{\partial \mathbf{u}_2} & \cdots & \frac{\partial \mathbf{f}_n(\mathbf{u},\mathbf{x})}{\partial \mathbf{u}_m} \end{array}\right]_{\text{ref}} \tag{12-5}$$

Thus, the moments, as a function of the controls, are given by:

$$\Delta \mathbf{m}_{\mathbf{u}} \cong \left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_{\text{ref}} \begin{bmatrix} \Delta \mathbf{u}_1 \\ \Delta \mathbf{u}_2 \\ \vdots \\ \Delta \mathbf{u}_m \end{bmatrix} = B_{\text{ref}} \Delta \mathbf{u} \tag{12-6}$$

The elements of B must be known, either as explicit functions, or, more commonly, they are generated from a data base. If B can be found for the current reference condition, Equation 12-6 can be solved to find $\Delta \mathbf{u}$ using any of the previously described techniques which solve the equation $\mathbf{m} = B\mathbf{u}$ for $\mathbf{u}$. Once $\Delta \mathbf{u}$ is known, the new controls are computed:

$$\mathbf{u} = \mathbf{u}_{\text{ref}} + \Delta \mathbf{u} \tag{12-7}$$

The new moment is approximately,

$$\mathbf{m} \cong \mathbf{m}_{ref} + \Delta\mathbf{m}_{\mathbf{u}} + \Delta\mathbf{m}_{\mathbf{x}} \tag{12-8}$$

Discrete Time Direct Allocation uses Direct Allocation to solve Equation 12-6. The advantage of using Direct Allocation is that the rate limits can be incorporated into the problem by examining the limits on $\Delta\mathbf{u}$. Most computers used in simulations and on board aircraft are digital machines that collect data at regular intervals. The sample rate of the machine in use will determine the size of the time step, Δt. This time step can be used to determine the control limits for the local problem by using the rate limits of the controls:

$$\Delta u_{i\,Min} = \dot{u}_{i\,Min}\Delta t \tag{12-9}$$

$$\Delta u_{i\,Max} = \dot{u}_{i\,Max}\Delta t \tag{12-10}$$

These limits on $\Delta\mathbf{u}$ represent the maximum distance the controls can move in one time step without violating any of the rate limits. However, the position limits of the controls should also be accounted for by checking each element of $\Delta\mathbf{u}_{Min}$ and $\Delta\mathbf{u}_{Max}$, so that $u_{i\,ref} + \Delta u_{i\,Min} \geq u_{i\,Min}$ and $u_{i\,ref} + \Delta u_{i\,Max} \leq u_{i\,Max}$. Whichever is more restrictive, the rate limit or the position limit, is used as the limit on $\Delta\mathbf{u}$:

$$\Delta u_{i\,Min} = Max\{\dot{u}_{i\,Min}\Delta t\,,\, u_{i\,Min} - u_{i\,ref}\} \tag{12-11}$$

$$\Delta u_{i\,Max} = Min\{\dot{u}_{i\,Max}\Delta t,\, u_{i\,Max} - u_{i\,ref}\} \tag{12-12}$$

By determining the control limits in this manner, the controls allocated from this scheme will not exceed their rate or position limits. Using the local *B* matrix and the local control limits, direct allocation can compute a subset of the AMS which is centered about the current moment and contains all the moments which can be attained in one time

step. This smaller version of the AMS will be referred to as the ΔAMS, or $\Delta\Phi$. See Figure 12-3. Moments which lie in $\Delta\Phi$ are those which the aircraft can generate in one time step without violating any of the controls' rate or position limits.

The ≡ sign in Equations 12-6 and 12-7 is due to the errors introduced by approximating the non-linearities in **f** with a linear B_{ref}. The accuracy of this approximation depends upon the severity of the non-linearities in **f** and the relative size of the Δ terms. The accuracy improves as the size of the Δ terms decreases. In application, the size of the Δ terms will be determined by the characteristics of the on-board flight computer.

In general, the desired change in moment, $\Delta\mathbf{m}_d$, is computed using the current commanded moment and the previously commanded moment:

$$\Delta\mathbf{m}_{d\,i} = \mathbf{m}_{d\,i} - \mathbf{m}_{d\,i-1} \tag{12-13}$$

If the information is available, it is recommended that the desired change in moment be computed using the current desired moment and the actual moment produced by the controls, $\mathbf{m}_{out}$:

$$\Delta\mathbf{m}_{d\,i} = \mathbf{m}_{d\,i} - \mathbf{m}_{out\,i-1} \tag{12-14}$$

By doing this, the control allocation algorithm will work to prevent errors from accumulating by attempting to correct for the errors already produced.

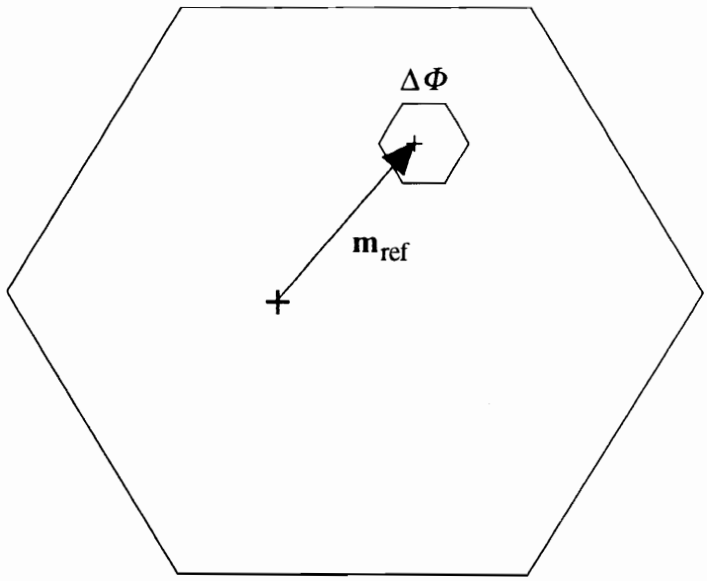


Figure 12-3: Φ and $\Delta\Phi$

Numerical Considerations

Because B changes in time, it is possible that there will be nxn submatrices of B which are singular. The presence of these singular submatrices means that some points on the boundary of Φ will correspond to multiple points in Ω. There are several special cases that need to be considered: faces which map to lines, faces which overlap, and the case in which the origin of R^n is on the boundary of Φ.

If a face in R^m maps to a line or a point in R^n on $\partial(\Phi)$, then the Direct Allocation algorithm will be unable to find a non-singular matrix when solving for the rotation vector, $\mathbf{t}$ (the matrices in Equations 9-12, 9-13, and 9-14 will all be singular). This situation occurs because $\mathbf{t}$ specifies the unique direction in R^n which is perpendicular to the (n-1)-D faces. If the faces map to a dimension smaller than (n-1), there will not be a unique vector in R^n which is perpendicular to the faces (i.e., for a 2-D plane in R^3, there is a unique direction which is perpendicular to that plane, but for a 1-D line in R^3, there are an infinite number of directions in R^3 which are perpendicular to that line). When this situation occurs, that facet can be skipped. Any line which intersects this line will also intersect a facet which has this line as one of its edges. In this manner, control solutions can be found. Note that when this situation occurs, multiple solutions are typically found.

If several faces in R^m map to the same plane on the boundary of $\Delta\Phi$, they may overlap. This condition can be seen in the occurrence of more than the 2 specified zeros in the vector $\mathbf{t}B$. The controls which correspond to the zero terms of $\mathbf{t}B$ can be set to either their minimum or maximum values to obtain a bounding facet. As a result, it is possible that more than one correct facet may be found. The presence of multiple correct facets can have the undesirable effect of causing the controls to reorient instantaneously.

There are several ways to deal with this situation. One way is to simply choose whichever solution was found first. Because the rate limits are imposed on the problem, any solution found will meet the constraints imposed by the rate limits and the position limits. Thus, any reorienting which would violate the rate constraints is prevented. A second method would be to compute all the solutions and choose the one which is closest to the previous solution in an effort to minimize control reconfigurations.

Because the previous moment is used as the reference condition when allocating the controls, it is possible that the origin may lie on bounding facet. The origin is on the boundary if there are (m-n-1) or more controls which have one of the limits on $\Delta\mathbf{u}$ set to zero. For example, for 5 controls and three moments, the bounding facets have 2 controls which are free to vary in either direction and 3 controls which are at limits and can only move in one direction.

In general, having the origin on the boundary is not a problem. However, there are two special cases when this situation can cause problems: when the desired moment lies on the same facet as the origin, and when the desired moment is unattainable. When the origin lies on a facet, the Direct Allocation algorithm will attempt to invert a singular matrix when checking the facet on which the origin lies (the matrix in Equation 9-19 will be singular). When the origin lies on a facet and the desired moment is unattainable, it is possible that the Direct Allocation algorithm will find no intersections. See Figure 12-4. These situations may be dealt with in using several techniques. A simple procedure which handles both cases is to move the origin by some small amount, $\delta\mathbf{m}$, so that it is contained within Φ, but no longer on $\partial(\Phi)$. This movement is done by taking controls which are at saturation and moving them some small amount before computing the

intersections. The size of the "small" amount will depend upon the precision of the computer being used and the relative magnitude of the control deflections.

The problem with this technique is that if $\delta\mathbf{m}$ is too large, it can introduce large errors by significantly changing the desired $\Delta\mathbf{m}$. See Figure 12-5. Thus, it is recommended that this procedure be used only when no solutions are initially found. Other methods examine particular cases and are the subject of future research.

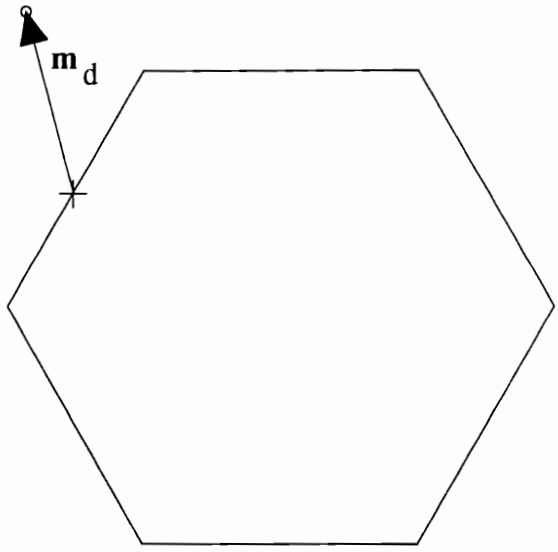


Figure 12-4: Origin on $\partial(\Phi)$ with Unattainable Desired Moment

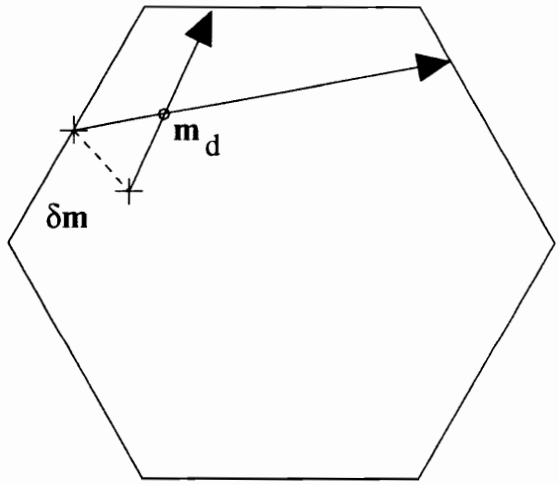


Figure 12-5: Moving the Origin can Cause Errors

Example 12-1: F-18 with $B = B(\mathbf{u})$

This example will use data taken from an F-18 simulation obtained from NASA Langley. It has seven independent moment generators: right and left horizontal tails, right and left ailerons, combined rudders, combined trailing edge flaps, and combined leading edge flaps. The rudders and flaps were not separated into right and left sides because of limitations in the available data . The deflection limits and rate limits are given in Table 12-1. Sign conventions for the controls are positive for trailing edge down or left, as appropriate to the surface. The control deflection limits for the ailerons and trailing edge flaps are different than for the actual airplane due to limitations of the aerodynamic data base used.

A sample rate of 80Hz, $\Delta t = 0.0125$s, was chosen because it is one that is commonly used in modern aircraft.[32] The control effectiveness was approximated using polynomial curve fits to data extrapolated from the data base.

Figures 12-6 through 12-8 show examples of the data and the curves used to approximate the data. For these figures, the controls not shown are held constant at zero. A line showing the slope at the origin is included to illustrate how the data differs from the globally linear approximation which is typically used. The equations for the moment coefficients as functions of the controls are given below.

$$C_l = -7.0256e\text{-}4\delta_{u_1} - 1.8200e\text{-}6\delta_{u_1}^2 + 7.0256e\text{-}4\delta_{u_2} + 1.8200e\text{-}6\delta_{u_2}^2 - 9.1835e\text{-}4\delta_{u_3} + 1.0235e\text{-}6\delta_{u_3}^2 - 2.1949e\text{-}7\delta_{u_3}^3 + 9.1835e\text{-}4\delta_{u_4} - 1.0235e\text{-}6\delta_{u_4}^2 + 2.1949e\text{-}7\delta_{u_4}^3 + 2.5542e\text{-}4\delta_{u_5} - 5.5e\text{-}6\delta_{u_7}\delta_{u_3} + 5.5e\text{-}6\delta_{u_7}\delta_{u_4} \quad (12\text{-}15)$$

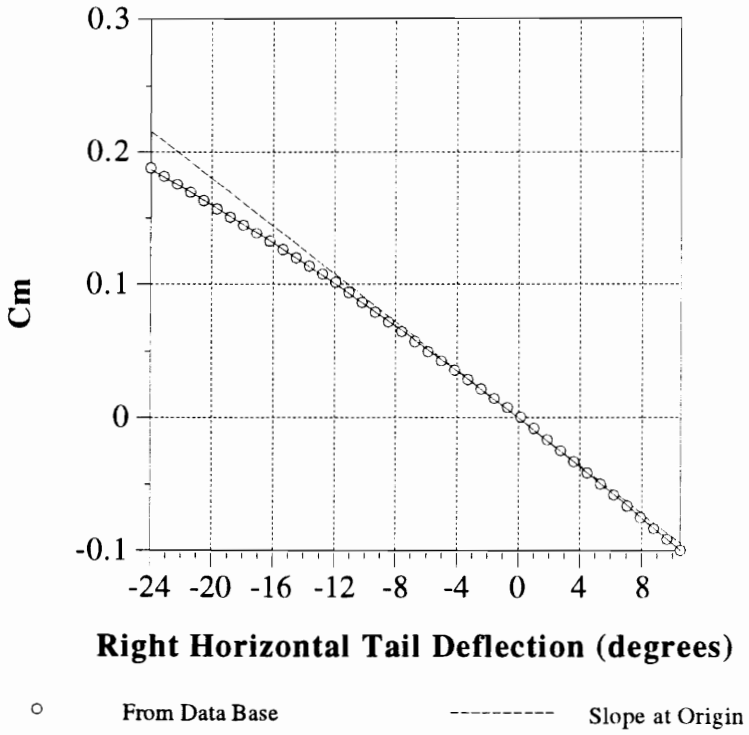


Figure 12-6: Pitching Moment Coefficient vs u_1

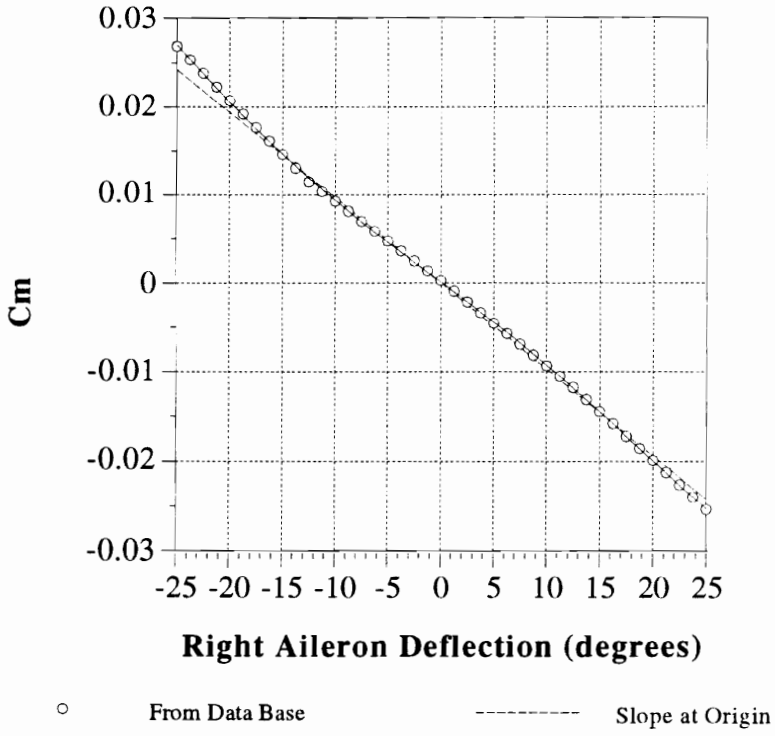


Figure 12-7: Pitching Moment Coefficient vs u_3

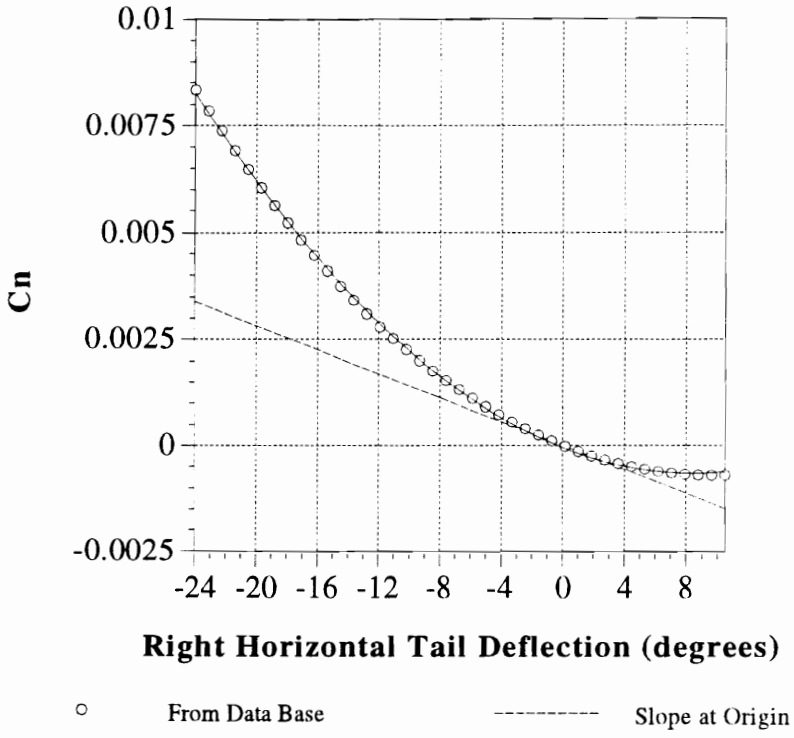


Figure 12-8: Rolling Moment Coefficient vs u_1

Table 12-1: Control Limits

Control	Min. (degrees)	Max. (degrees)	Rate Limit (°/s)
u_1 R. Horizontal tail	-24	10.5	±40
u_2 L. Horizontal tail	-24	10.5	±40
u_3 R. Aileron	-25	25	±100
u_4 L. Aileron	-25	25	±100
u_5 Rudders	-30	30	±56
u_6 Trailing Edge Flaps	-8	40	±18
u_7 Leading Edge Flaps	-3	34	±15

$$C_m = -8.9909e\text{-}3\delta_{u_1} - 5.0538e\text{-}5\delta_{u_1}^2 - 8.9909e\text{-}3\delta_{u_2} - 5.0538e\text{-}5\delta_{u_2}^2 - 9.6905e\text{-}4\delta_{u_3} - 1.5214e\text{-}5\delta_{u_3}^2 - 2.8961e\text{-}7\delta_{u_3}^3 + 1.8669e\text{-}8\delta_{u_3}^4 + 1.7488e\text{-}10\delta_{u_3}^5 - 9.6905e\text{-}4\delta_{u_4} - 1.5214e\text{-}5\delta_{u_4}^2 - 2.8961e\text{-}7\delta_{u_4}^3 + 1.8669e\text{-}8\delta_{u_4}^4 + 1.7488e\text{-}10\delta_{u_4}^5 + 1.9833e\text{-}3\delta_{u_5} + 2.6648e\text{-}5\delta_{u_5}^2 + 4.9094e\text{-}12\delta_{u_5}^3 - 1.4184e\text{-}8\delta_{u_5}^4 + 1.832e\text{-}3\delta_{u_6} - 1.4291e\text{-}3\delta_{u_7} \quad (12\text{-}16)$$

$$C_n = -1.4097e\text{-}4\delta_{u_1} + 7.3712e\text{-}6\delta_{u_1}^2 - 5.1567e\text{-}8\delta_{u_1}^3 + 1.4097e\text{-}4\delta_{u_2} - 7.3712e\text{-}6\delta_{u_2}^2 + 5.1567e\text{-}8\delta_{u_2}^3 - 1.2112e\text{-}3\delta_{u_5} - 2.9e\text{-}6\delta_{u_7}\delta_{u_3} + 2.9e\text{-}6\delta_{u_7}\delta_{u_4} + C_{n_{\delta ail}} \quad (12\text{-}17)$$

For positive aileron deflections,

$$C_{n_{\delta ail}} = 2.0796e\text{-}4\delta_{u_3} - 2.0796e\text{-}4\delta_{u_4} \quad (12\text{-}18)$$

For negative aileron deflections,

$$C_{n_{\delta ail}} = -5.7772e\text{-}5\delta_{u_3} - 4.6614e\text{-}6\delta_{u_3}^2 - 3.7762e\text{-}7\delta_{u_3}^3 - 7.5123e\text{-}9\delta_{u_3}^4 + 5.7772e\text{-}5\delta_{u_4} + 4.6614e\text{-}6\delta_{u_4}^2 + 3.7762e\text{-}7\delta_{u_4}^3 + 7.5123e\text{-}9\delta_{u_4}^4 \quad (12\text{-}19)$$

The aileron yawing coefficients were separated into positive and negative parts for two reasons: this method better approximated the data base, and discontinuities such as this are not uncommon in real data.

The above equations include mutual-interference effects between the ailerons and leading edge flaps. These effects are included as an example of another type of non-linearity commonly found in real aircraft. The modeling of the effect is not exact, as data was unavailable for the F-18. The data was approximated using equations and data from Reference 33, which analyzes an F-16. Figure 12-9 illustrates the effect of the leading edge flaps on the aileron yaw coefficients with the other controls fixed at zero.

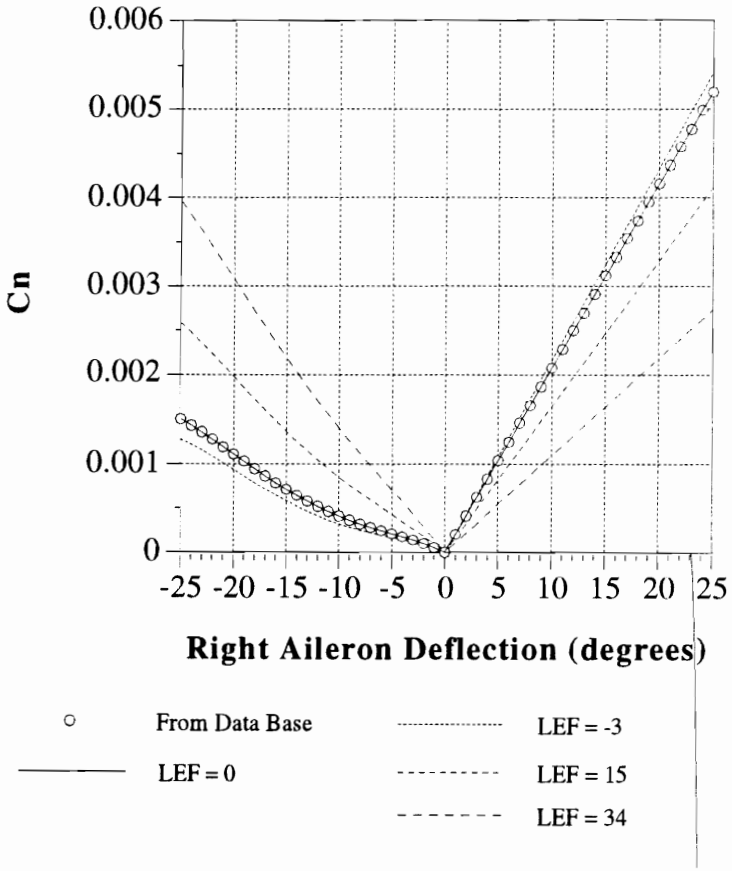


Figure 12-9: Yawing Moment vs u_3

The flight condition for this example was assumed to be constant with Mach = 0.3, α = 15° and β = 0° at an altitude of 10,000 ft. The example maneuver was a velocity vector roll generated using the program Rollerator [31]. The parameters used to generate the desired moments were a steady state wind-axis roll rate of 70°/s and a roll mode time constant of 1s, at the flight condition specified above. Figure 12-10 shows the desired moments for this maneuver expressed as non-dimensional moment coefficients.

For comparison purposes, several methods are used to allocate the controls for this problem. The first two assume that the problem is globally linear, and use the B matrix computed at a reference condition of $\mathbf{u} = \mathbf{0}$:

$$B_0 = \begin{bmatrix} -7.025e\text{-}4 & 7.025e\text{-}4 & -9.183e\text{-}4 & 9.183e\text{-}4 & 2.554e\text{-}4 & \cdots \\ -8.990e\text{-}3 & -8.990e\text{-}3 & -9.690e\text{-}4 & -9.690e\text{-}4 & 1.983e\text{-}3 & \cdots \\ -1.409e\text{-}4 & 1.409e\text{-}4 & 7.509e\text{-}5 & -7.509e\text{-}5 & -1.211e\text{-}3 & \end{bmatrix}$$

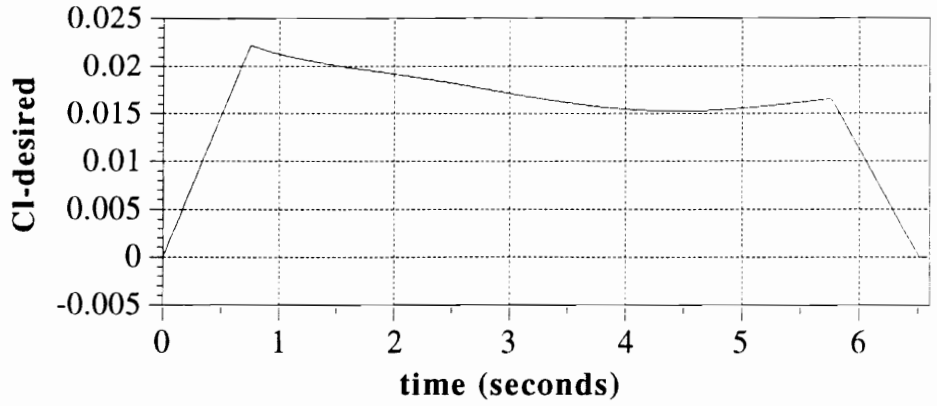
$$\begin{matrix} 0 & 0 \\ \cdots \quad 1.832e\text{-}3 & -1.429e\text{-}3 \\ 0 & 0 \end{matrix} \Bigg] \qquad (12\text{-}20)$$

Using B_0, the controls are allocated using Direct Allocation and the pseudo-inverse of B_0:

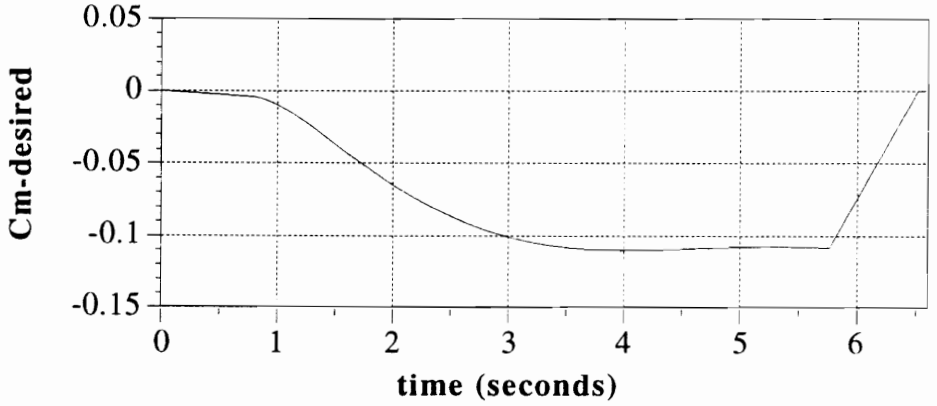
$$P_0 = B_0^{\dagger} = \begin{bmatrix} -2.666e+2 & -5.431e+1 & -2.226e+2 \\ 2.711e+2 & -5.203e+1 & 5.503e+1 \\ -3.355e+2 & -4.805e+0 & -1.322e+1 \\ 3.360e+2 & -6.657e+0 & -4.832e+0 \\ 2.095e+1 & 3.803e-1 & -7.938e+2 \\ -4.506e-1 & 1.083e+1 & 1.707e+1 \\ 3.515e-1 & -8.452e+0 & -1.331e+1 \end{bmatrix} \qquad (12\text{-}21)$$

The next two methods update B using the current control positions. When computing the controls in the i-th frame, use B_i:

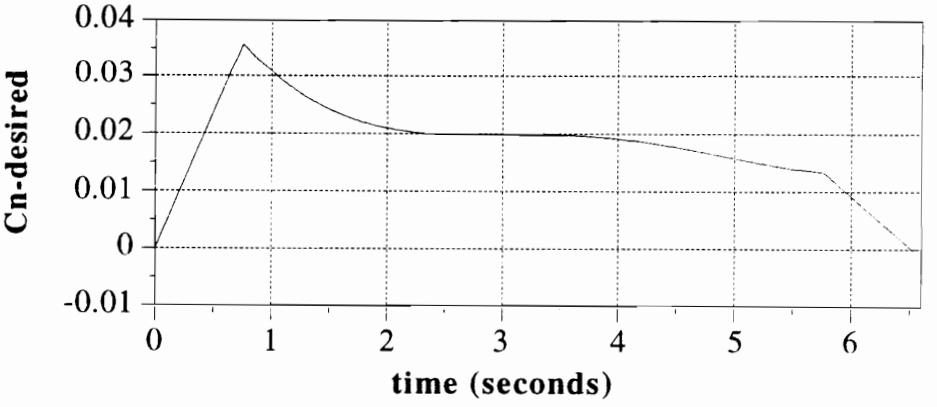
$$B_i = B(\mathbf{u}_{i-1}) \qquad (12\text{-}22)$$



a) Rolling Moment Coefficient



b) Pitching Moment Coefficient



c) Yawing Moment Coefficient

Figure 12-10: Desired Moments

$$P_i = B_i^{\dagger} \tag{12-23}$$

Again, Direct Allocation and the pseudo-inverse are used to allocate the controls. It should be noted that this approach is more commonly taken when B is a function of flight condition, rather than control deflections.

The final two methods treat the problem as being locally linear, and solve the problem in discrete time for $\Delta\mathbf{u}$:

$$\Delta\mathbf{m} = B(\mathbf{u}_{i-1})\Delta\mathbf{u} \tag{12-24}$$

Equation 12-24 is solved using Direct Allocation and using the pseudo-inverse of $B(\mathbf{u}_{i-1})$.

Figures 12-11 through 12-16 show the controls allocated using the various methods. All control deflections are expressed in degrees. If a control was commanded to exceed one of its limits, the commanded control is shown as a dashed line. Note that the controls in Figures 12-11 and 12-12 vary smoothly, as do the commanded moments. However, the controls from the other examples change direction more rapidly as a result of the changes in B.

Figures 12-11, 12-13, and 12-15 show that the methods which use the pseudo-inverse command the left horizontal tail, u_2, to move beyond its maximum position, 10.5°. The horizontal tails primarily provide pitching moment, and as a result of the saturation, these methods fail to produce the desired moments and are especially deficient in pitch. Direct Allocation will not saturate controls unless the commanded moments are on the boundary of Φ. To provide the required moments, the Direct Allocation methods, Figures 12-12 and 12-14, command smaller deflections from u_2 and compensate by

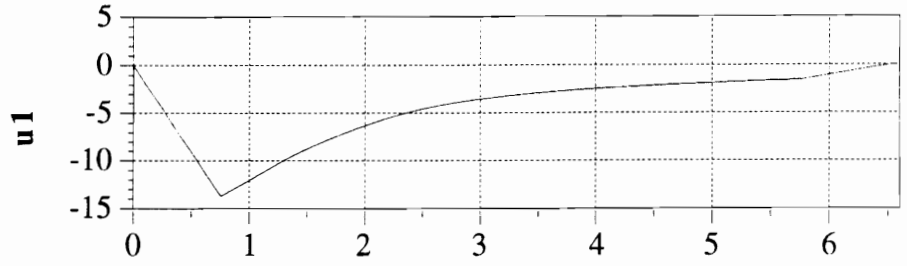
commanding much larger deflections from the leading edge flaps, u_7. The leading edge flaps are also primarily pitching moment generators, and they have a much greater maximum deflection, 34°, than the horizontal tails. However, the leading edge flaps move very slowly, 15°/s, and the Direct Allocation methods cause them to rate saturate. The Discrete Time Direct Allocation method, Figure 12-16, commands the left horizontal tail to move to its maximum position, but does not command it to move beyond this point. Instead, it moves other controls to generate the desired moments. Also note that Discrete Time Direct Allocation commands slower deflections from the leading edge flaps to prevent them from rate saturating.

Figures 12-17 through 12-22 show the magnitude of the error, $|\mathbf{m}_d - \mathbf{m}_{out}|$, for each method. For the globally linear methods, Figures 12-17 and 12-18, the error grows and shrinks as the magnitude of the commanded moment grows and shrinks. This behavior is typical, in that the error is larger further away from the point about which the equations were linearized. The error from the local pseudo-inverse, Figure 12-19, rising during the initial part of the maneuver and falls and levels off before returning to zero. The peak of the error occurs at t = 0.75s, when the maneuver transitions from a linearly increasing moment command to a non-linearly varying moment command. Figure 12-20 shows that using the local B matrix and Direct Allocation has a large error during the first 2s of the maneuver, during which time the leading and trailing edge flaps are severely rate limited. The discrete time local pseudo-inverse method produces a relatively small error, Figure 12-21, except during the time in which the left horizontal tail is position limited. The Discrete Time Direct Allocation method, Figure 12-22, produces the smallest errors, but possesses some interesting features which will discussed in more detail later. When $\Delta\mathbf{m}$

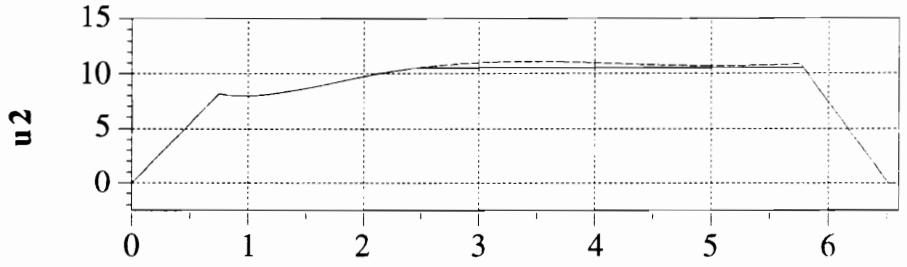
is computed for the discrete time methods, it is important to note that the current desired moment and the moment produced in the previous time step are used:

$$\Delta \mathbf{m}_i = \mathbf{m}_{d\,i} - \mathbf{m}_{out\,i-1} \tag{12-25}$$

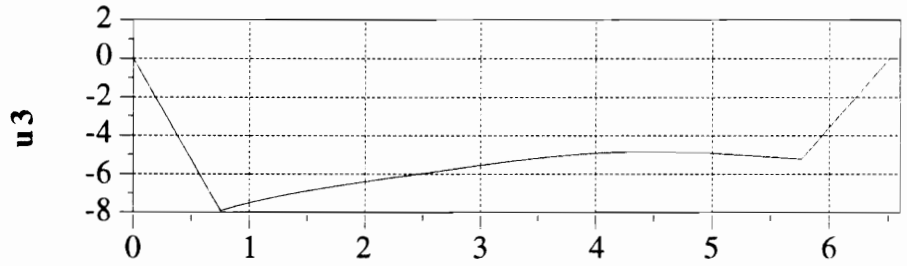
Table 12-2 compares the errors produced using the various allocation schemes and shows the controls which are saturated using each scheme. Discrete Time Direct Allocation produces the smallest errors, in part because it is the only method which does not command the controls to exceed any of their physical limits.



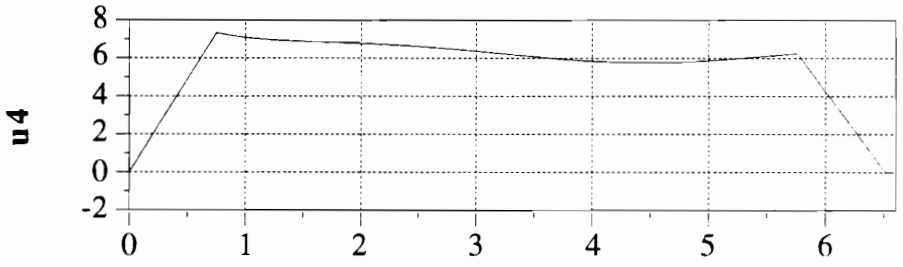
a) Right Horizontal Tail



b) Left Horizontal Tail

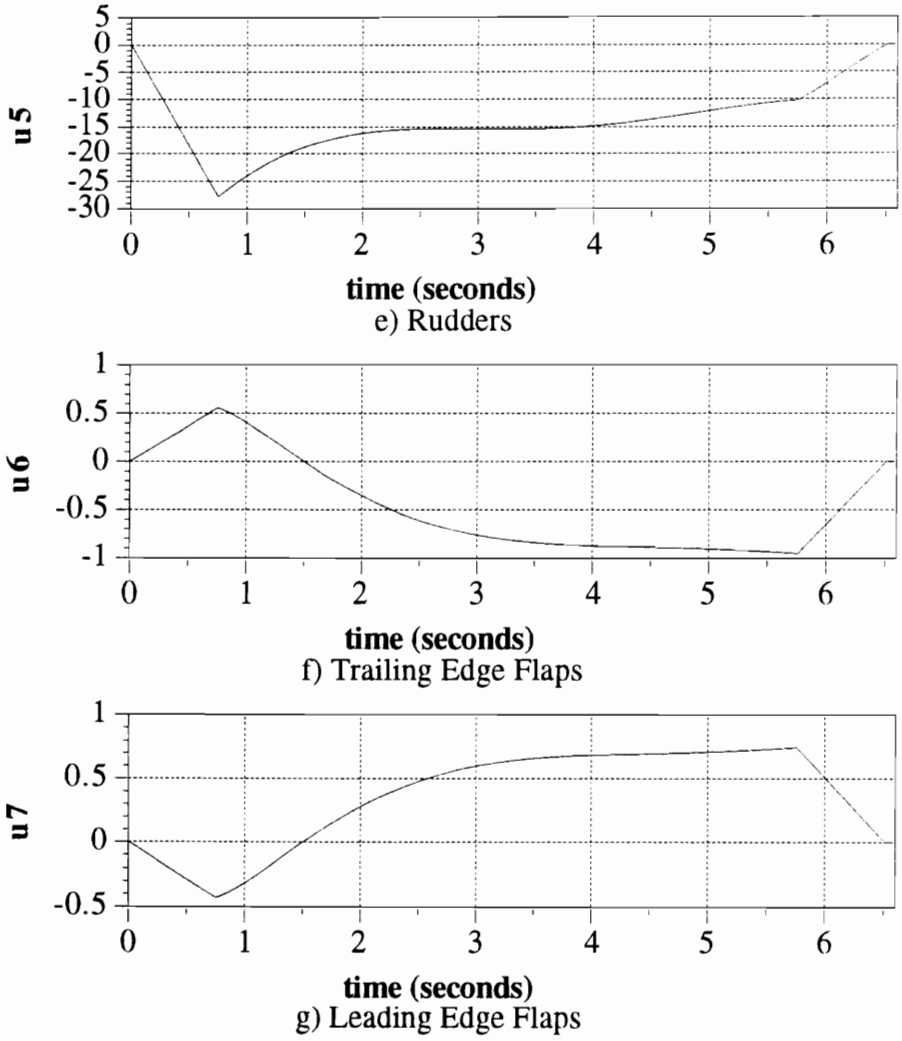


c) Right Aileron



d) Left Aileron

Figure 12-11: Controls Using Global Pseudo-Inverse



e) Rudders

f) Trailing Edge Flaps

g) Leading Edge Flaps

Figure 12-11: Controls Using Global Pseudo-Inverse

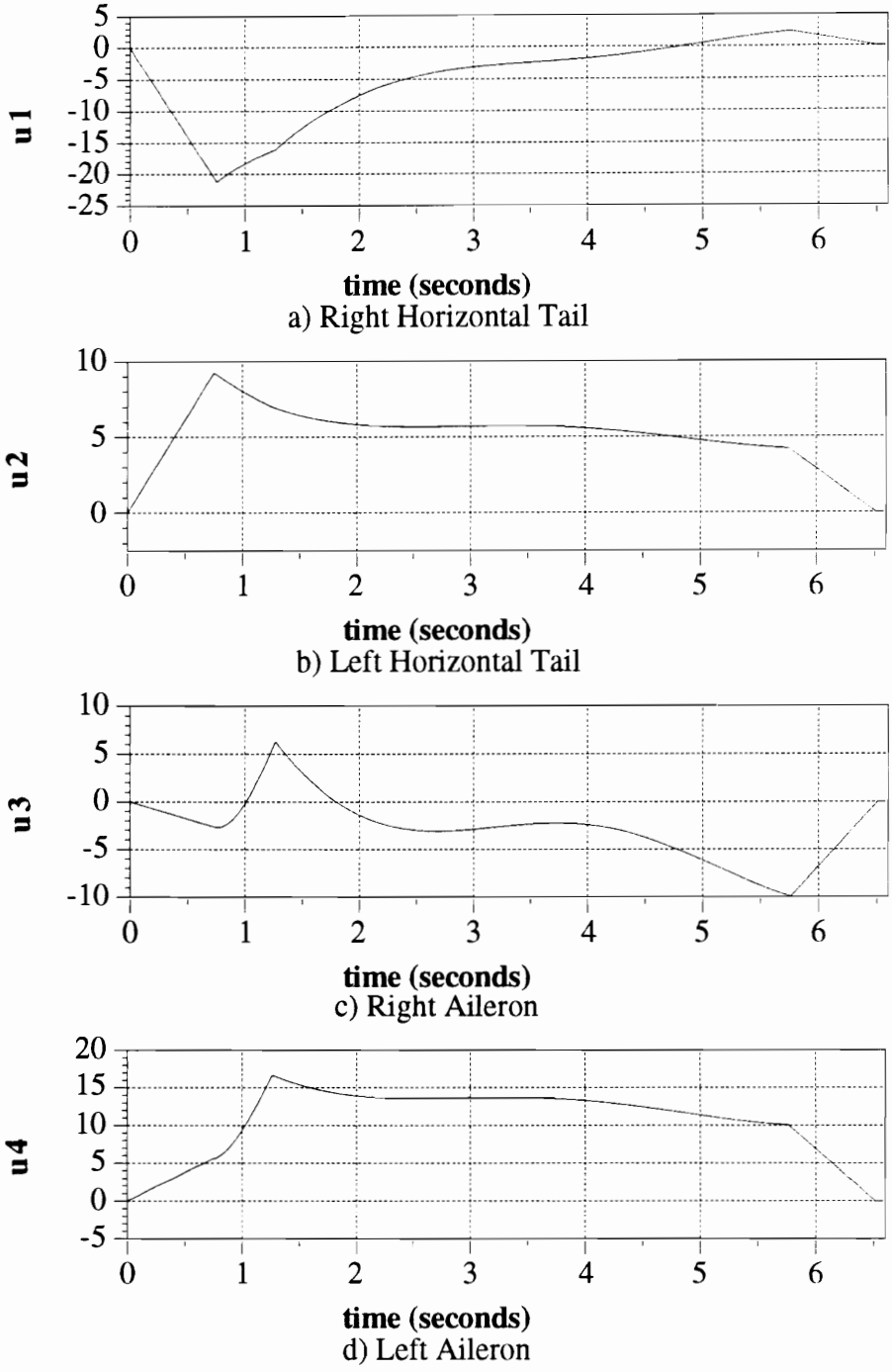
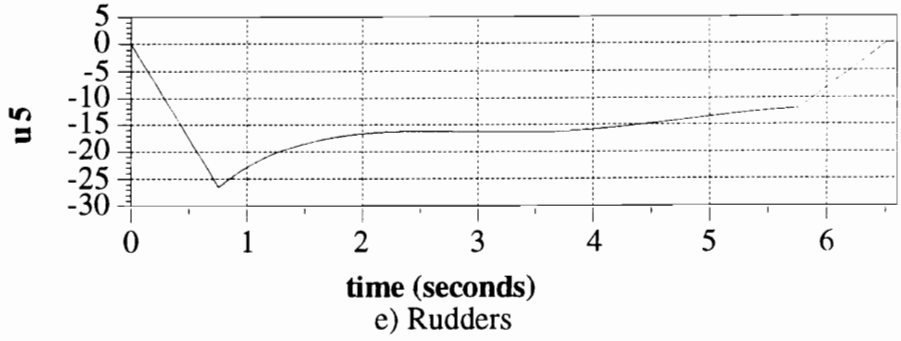
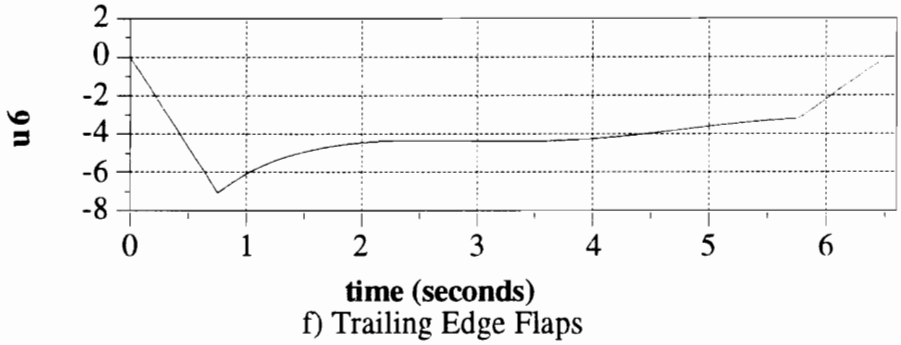


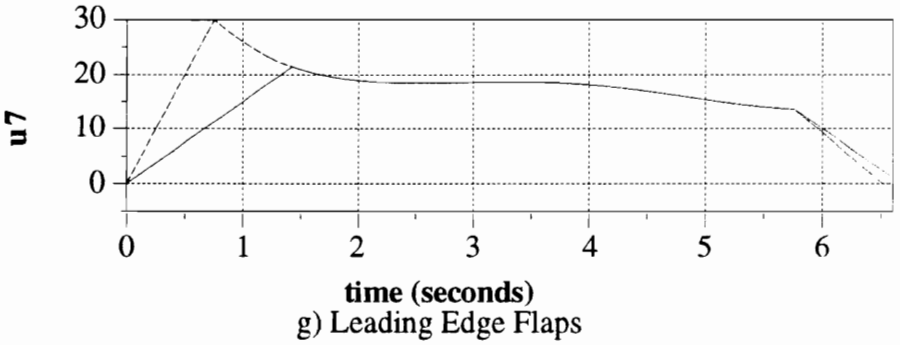
Figure 12-12: Controls Using Global B and Direct Allocation



e) Rudders

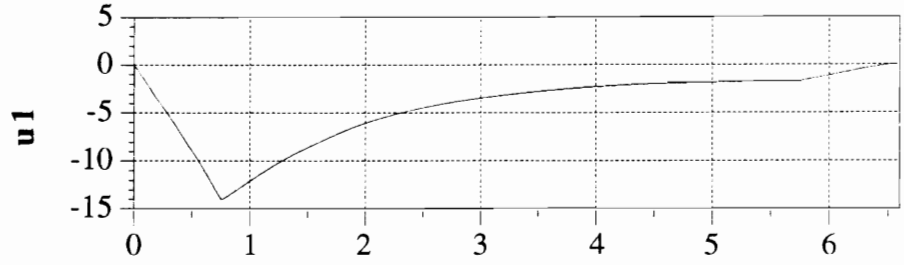


f) Trailing Edge Flaps

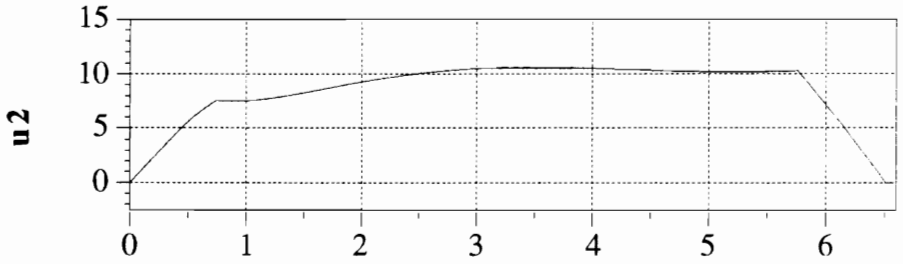


g) Leading Edge Flaps

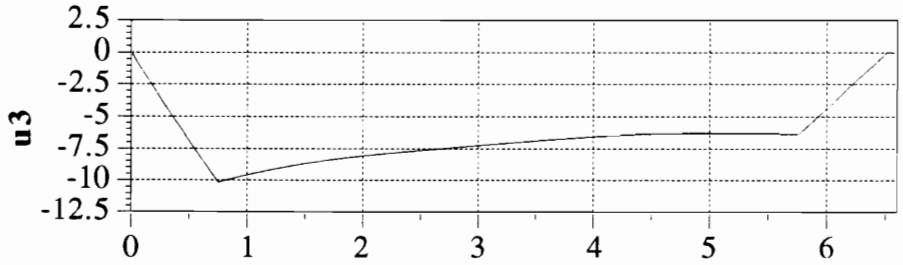
Figure 12-12: Controls Using Global B and Direct Allocation



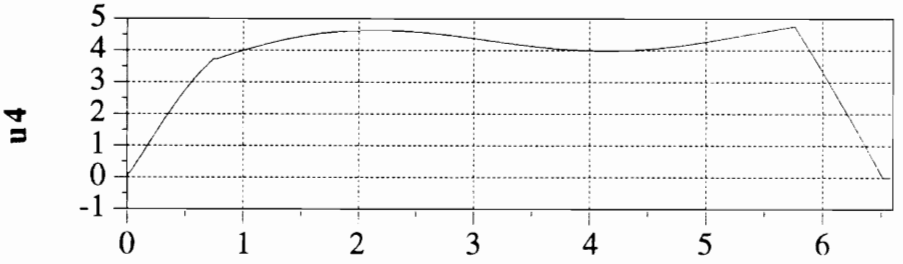
a) Right Horizontal Tail



b) Left Horizontal Tail

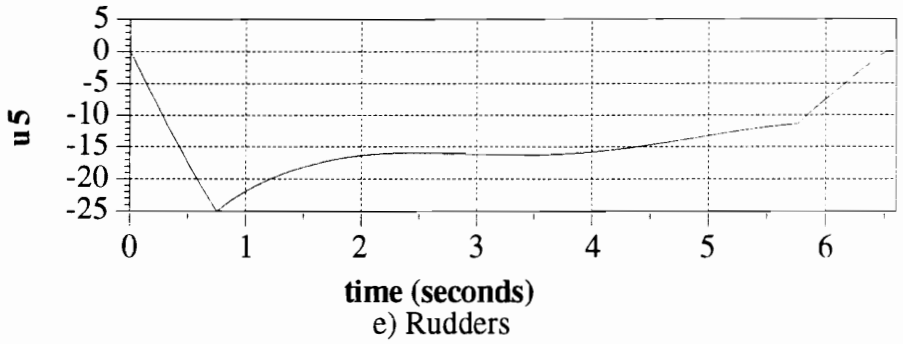


c) Right Aileron

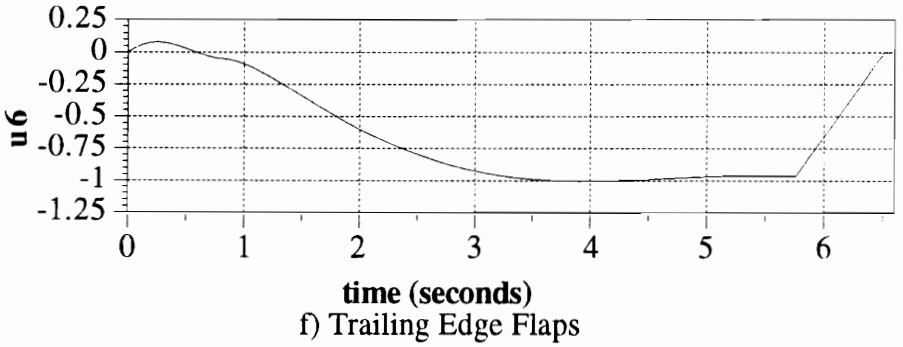


d) Left Aileron

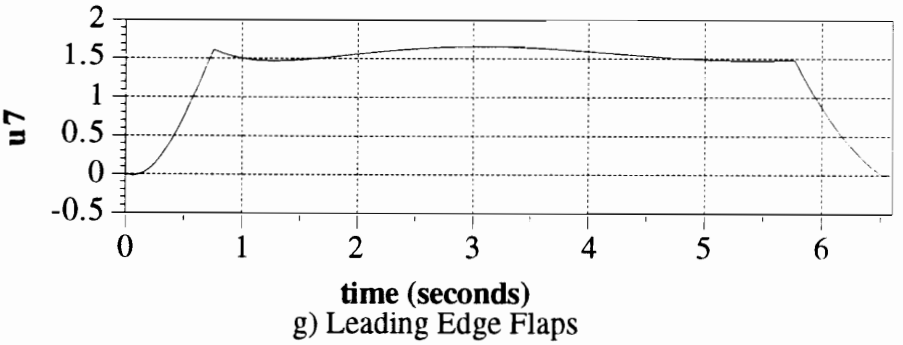
Figure 12-13: Controls Using Local Pseudo-Inverse



e) Rudders



f) Trailing Edge Flaps



g) Leading Edge Flaps

Figure 12-13: Controls Using Local Pseudo-Inverse

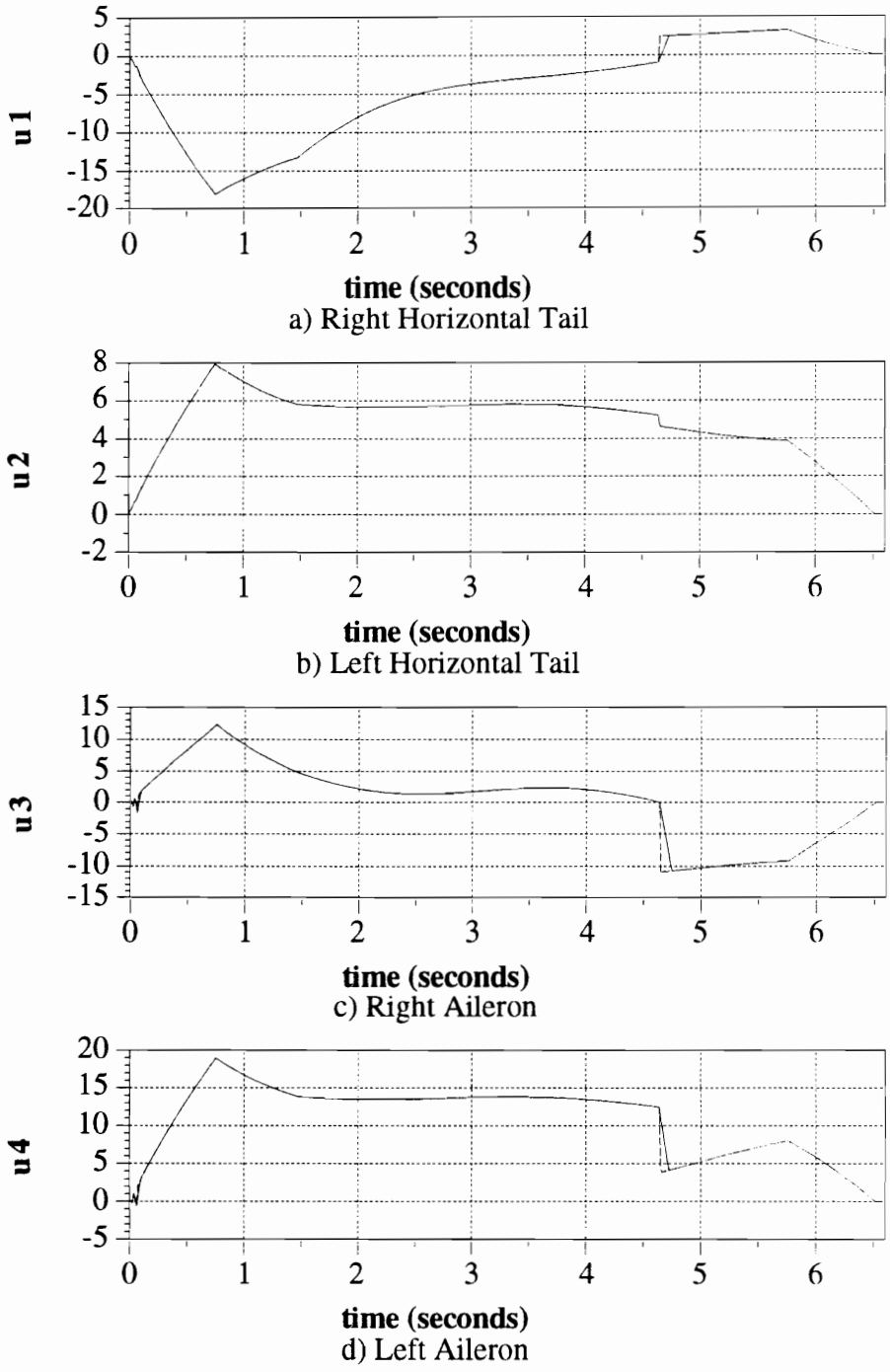
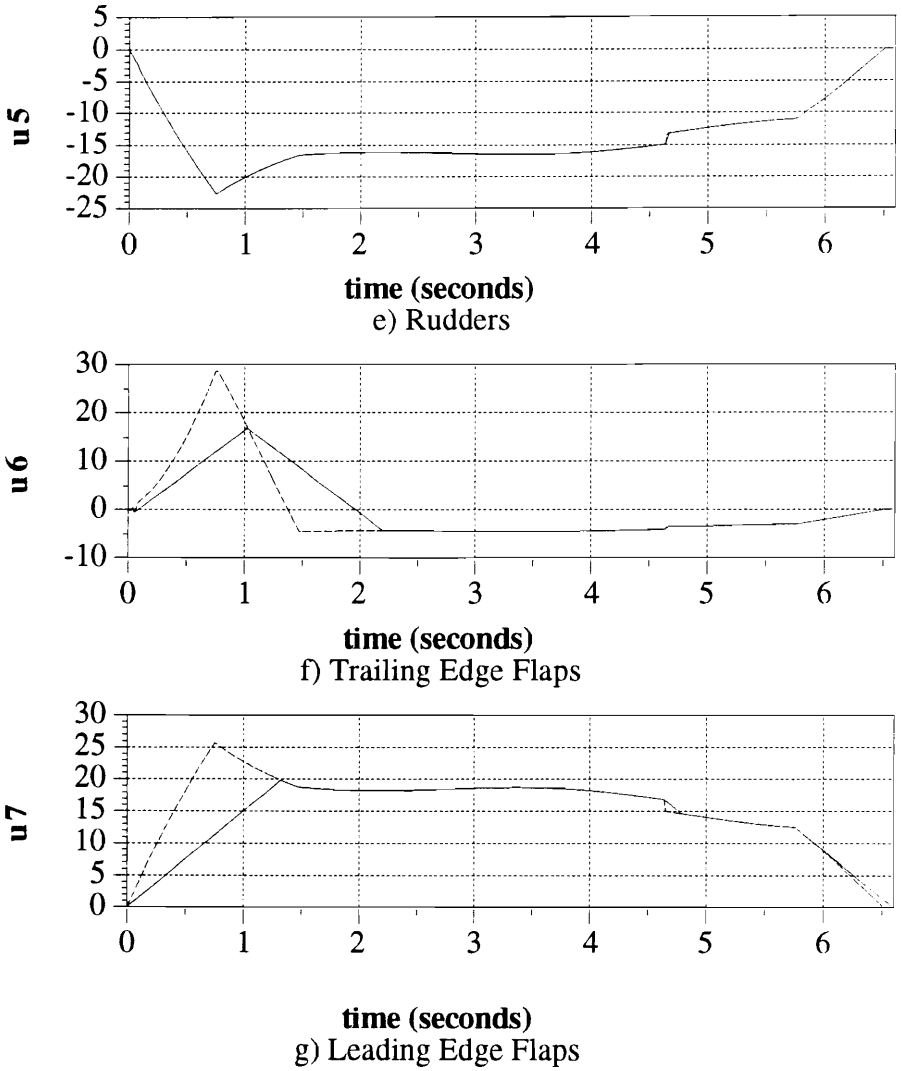


Figure 12-14: Controls Using Local *B* and Direct Allocation



e) Rudders

f) Trailing Edge Flaps

g) Leading Edge Flaps

Figure 12-14: Controls Using Local B and Direct Allocation

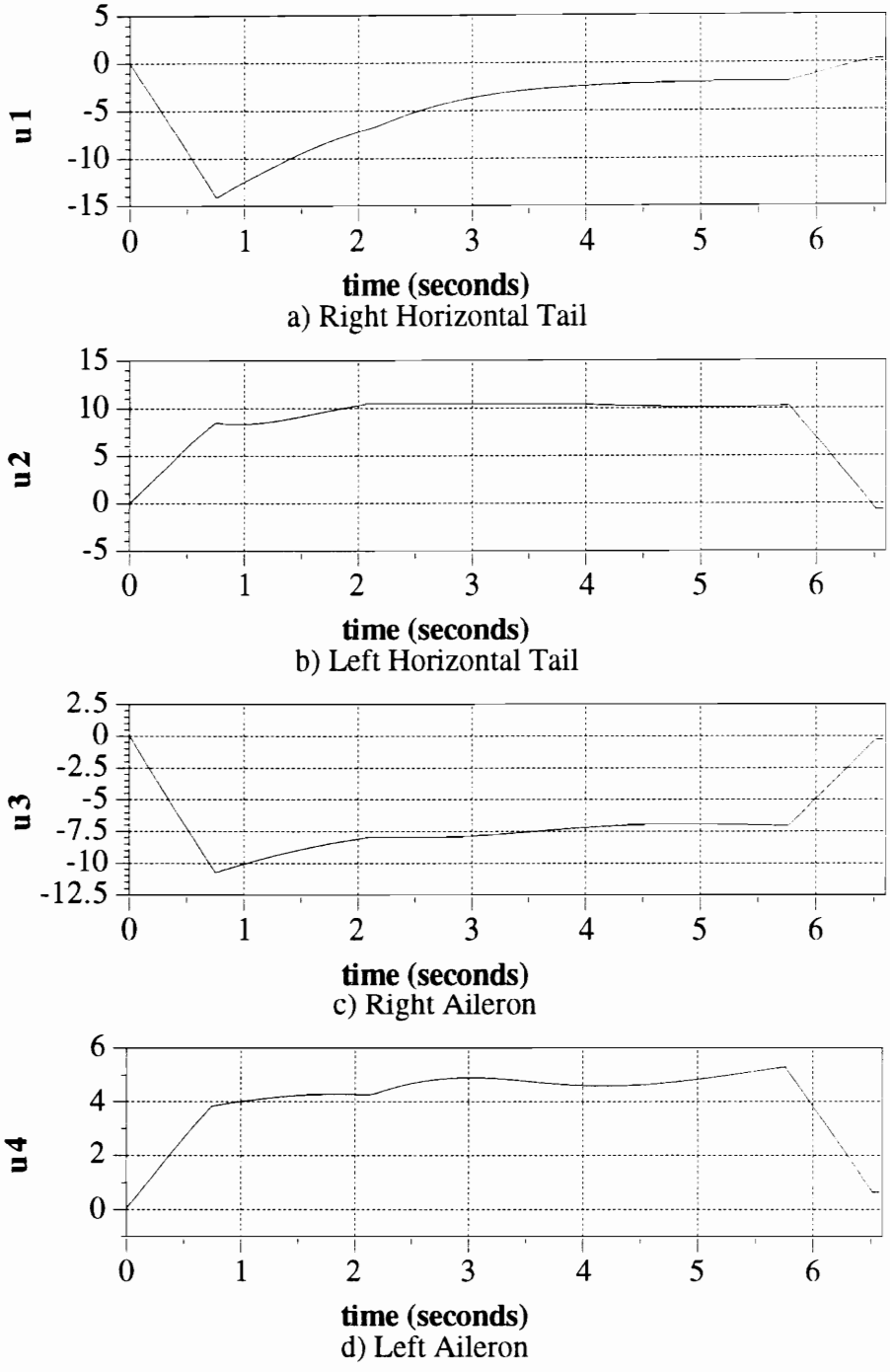
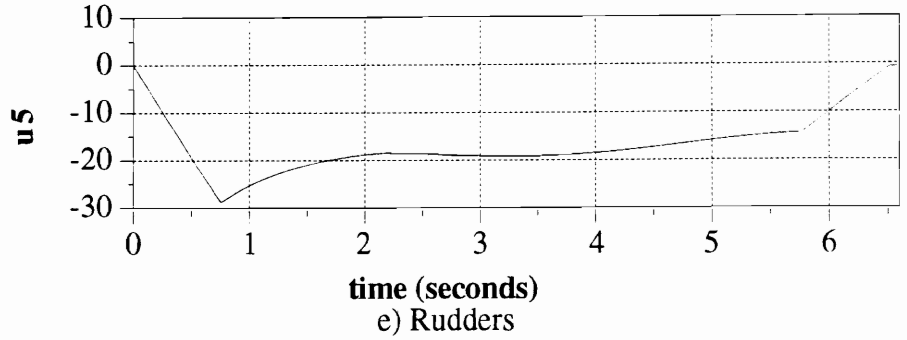
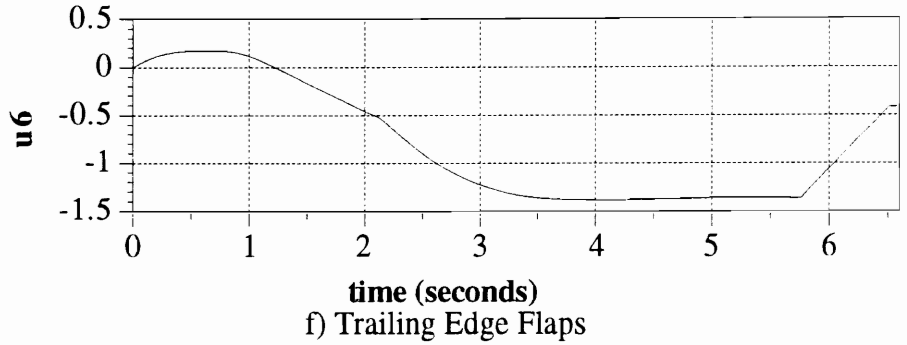


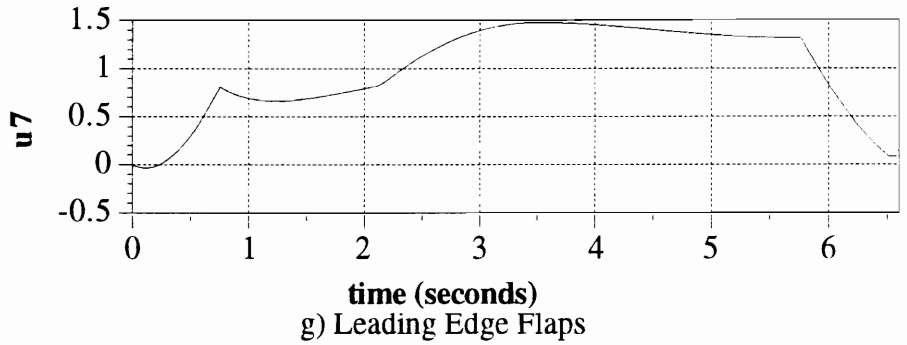
Figure 12-15: Controls Using Discrete Local Pseudo-Inverse



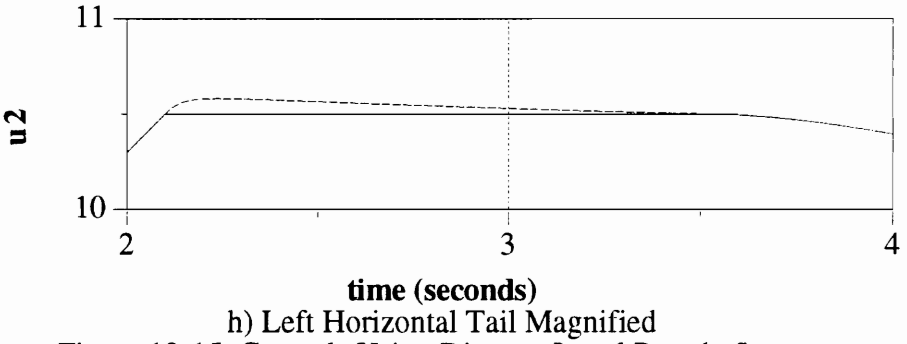
e) Rudders



f) Trailing Edge Flaps



g) Leading Edge Flaps



h) Left Horizontal Tail Magnified

Figure 12-15: Controls Using Discrete Local Pseudo-Inverse

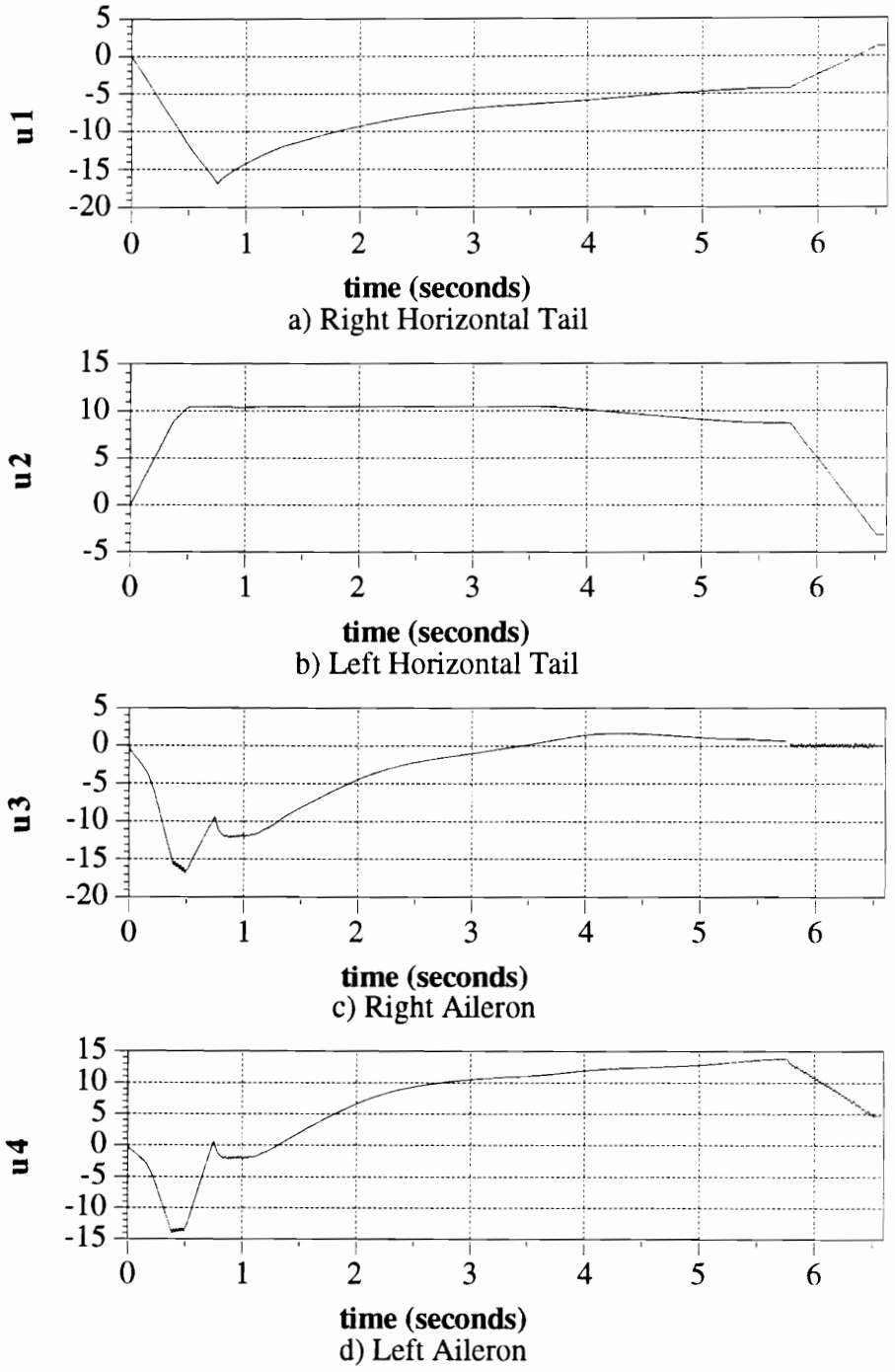
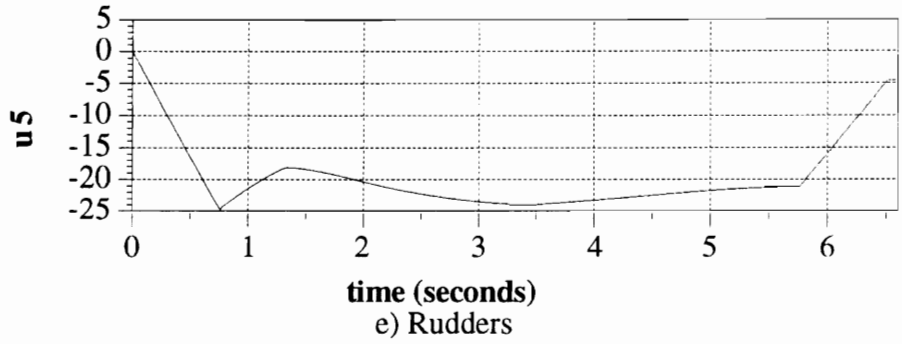
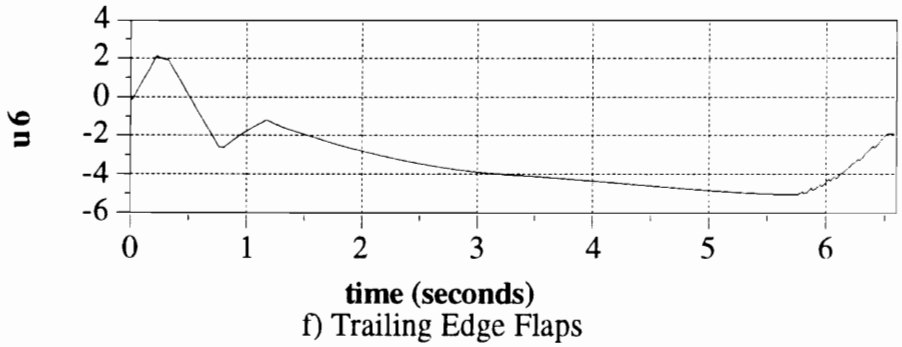


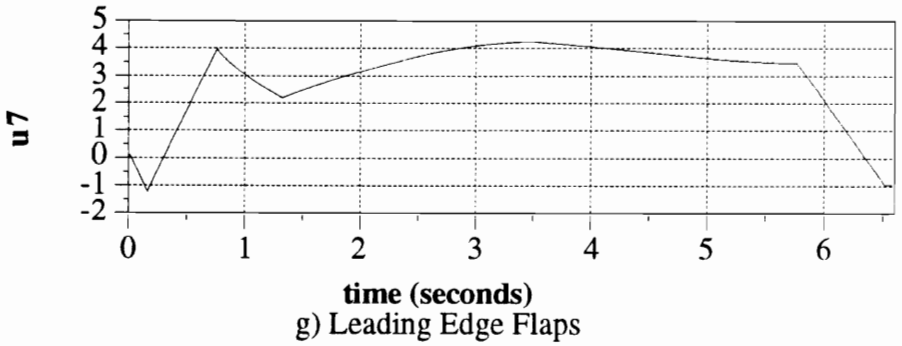
Figure 12-16: Controls Using Discrete Direct Allocation



e) Rudders



f) Trailing Edge Flaps



g) Leading Edge Flaps

Figure 12-16: Controls Using Discrete Direct Allocation

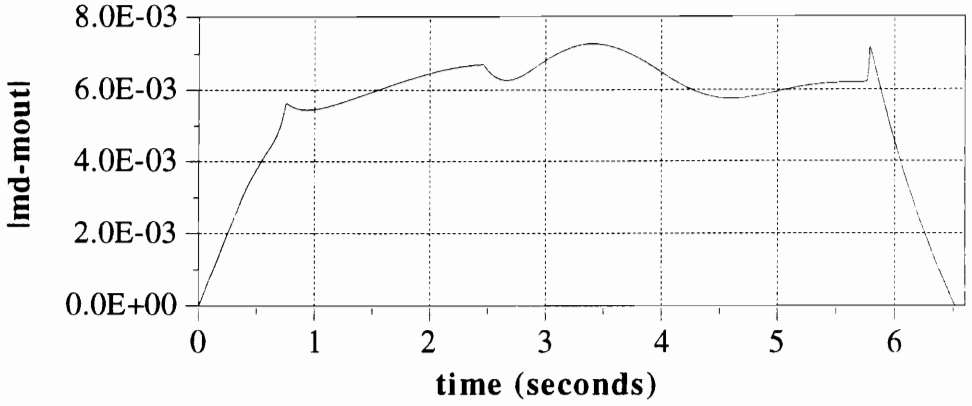


Figure 12-17: Error Using Global Pseudo-Inverse

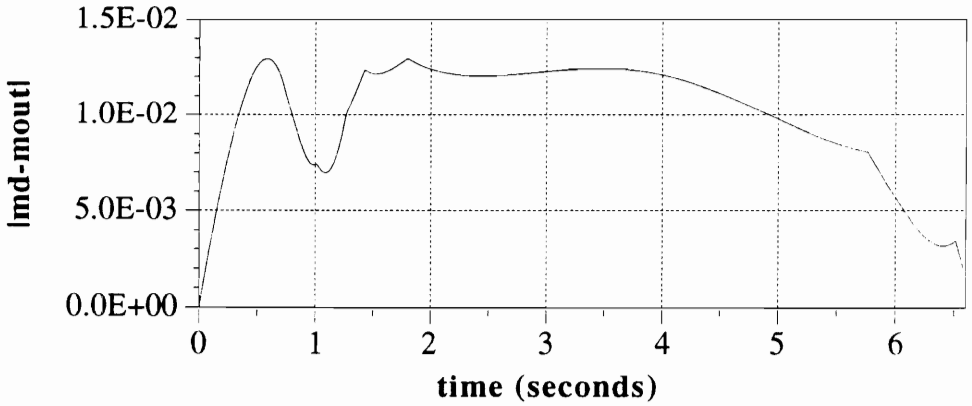


Figure 12-18: Error Using Global *B* and Direct Allocation

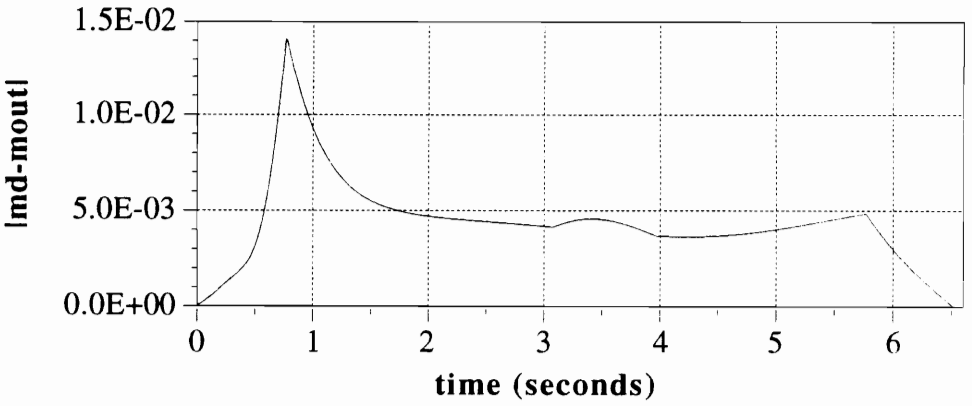


Figure 12-19: Error Using Local Pseudo-Inverse

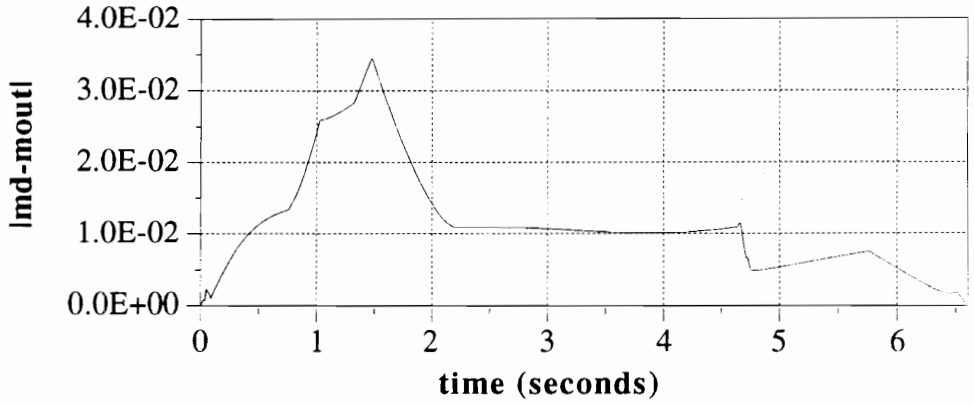


Figure 12-20: Error Using Local B and Direct Allocation

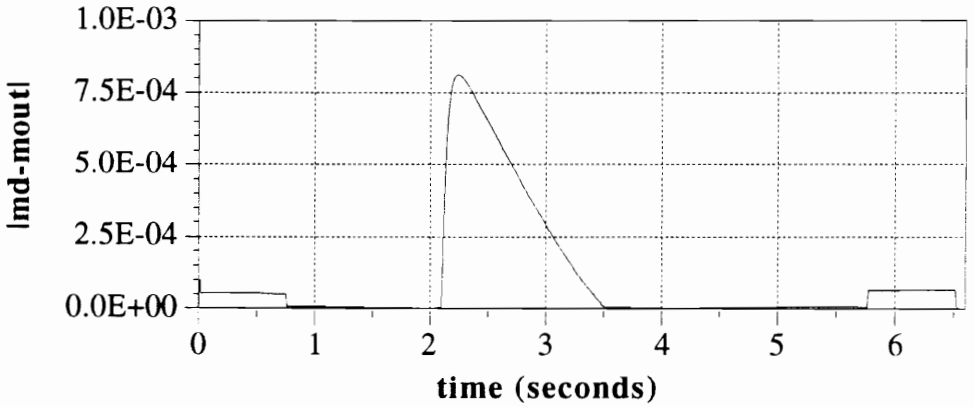


Figure 12-21: Error Using Discrete Local Pseudo-Inverse

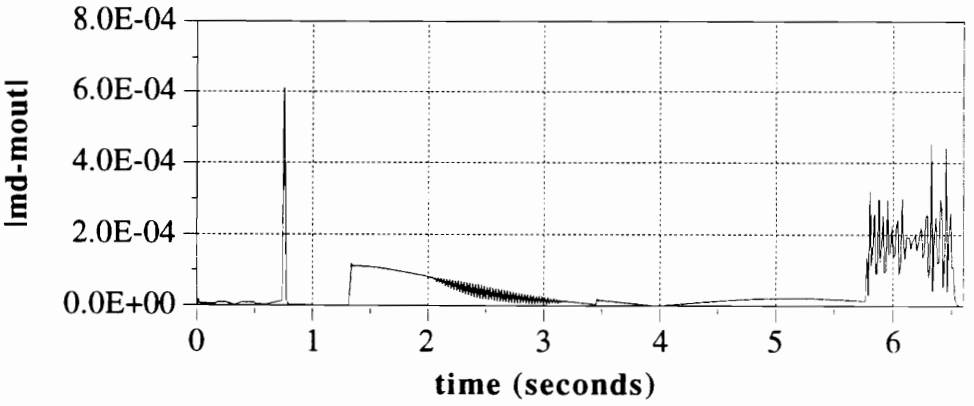


Figure 12-22: Error Using Discrete Direct Allocation

Table 12-2: Error Comparison

	Allocation Method	Maximum Error	Average Error	Position Saturated	Rate Saturated
Globally Linear	Pseudo-Inverse	7.2781e-3	5.4717e-3	u_2	none
	Direct Allocation	1.2979e-2	1.0033e-2	none	u_7
Locally Linear	Pseudo-Inverse	1.4100e-2	4.3855e-3	u_2	none
	Direct Allocation	3.4510e-2	1.1336e-2	none	u_1,u_3,u_4,u_5, u_6,u_7
Discrete Time	Pseudo-Inverse	8.1179e-4	1.0421e-4	u_2	none
	Direct Allocation	6.1047e-4	4.5691e-5	none	none

Implementation Issues

When examining the controls allocated using Discrete Time Direct Allocation, and the errors they produce, several characteristics stand out. First is the phenomenon in which a control rapidly oscillates, for example the right aileron in Figure 12-16c. This "chattering" of the controls can be caused by several factors. One cause of the chatter is the fact that the faces of Ω which map to facets in Φ can change as B changes.

For example, examine the chattering which occurs in controls u_3 and u_4 from t = 0.4s and t = 0.525s. During this time, the facet at which the desired moment is pointing varies between the facet defined by u_2 and u_3 and the facet defined by u_2 and u_4. Figure 12-23 shows $\Delta\Phi$ at t = 0.425s with the intersection of the desired moment and $\partial(\Delta\Phi)$ indicated by a white circle. The facet that the circle lies on is defined by controls u_2 and u_3. When the controls are allocated using this facet, the B matrix changes, so that different faces are on the exterior. Figure 12-24 shows $\Delta\Phi$ at the next time step, t = 0.4375s, with the intersection of the desired moment and $\partial(\Delta\Phi)$ indicated by a white circle. The facet that the circle lies on is now defined by controls u_2 and u_4. When the controls are allocated using this facet, the B matrix changes, so that the faces which were on the exterior at t = 0.425s are again on the exterior at t = 0.45s. During these times, three of the six faces of the box {0222001} are facets of Φ (See Figure 12-25). Six of the eight vertices shown in Figure 12-25 are on the boundary of the $\Delta\Phi$ during this entire time. However, only one of the two vertices indicated in bold is on the exterior of $\Delta\Phi$ at a particular instant, unless $[B_2\ B_3\ B_4]$ is singular. From t = 0.4s to t = 0.525s these vertices alternate being on the boundary.

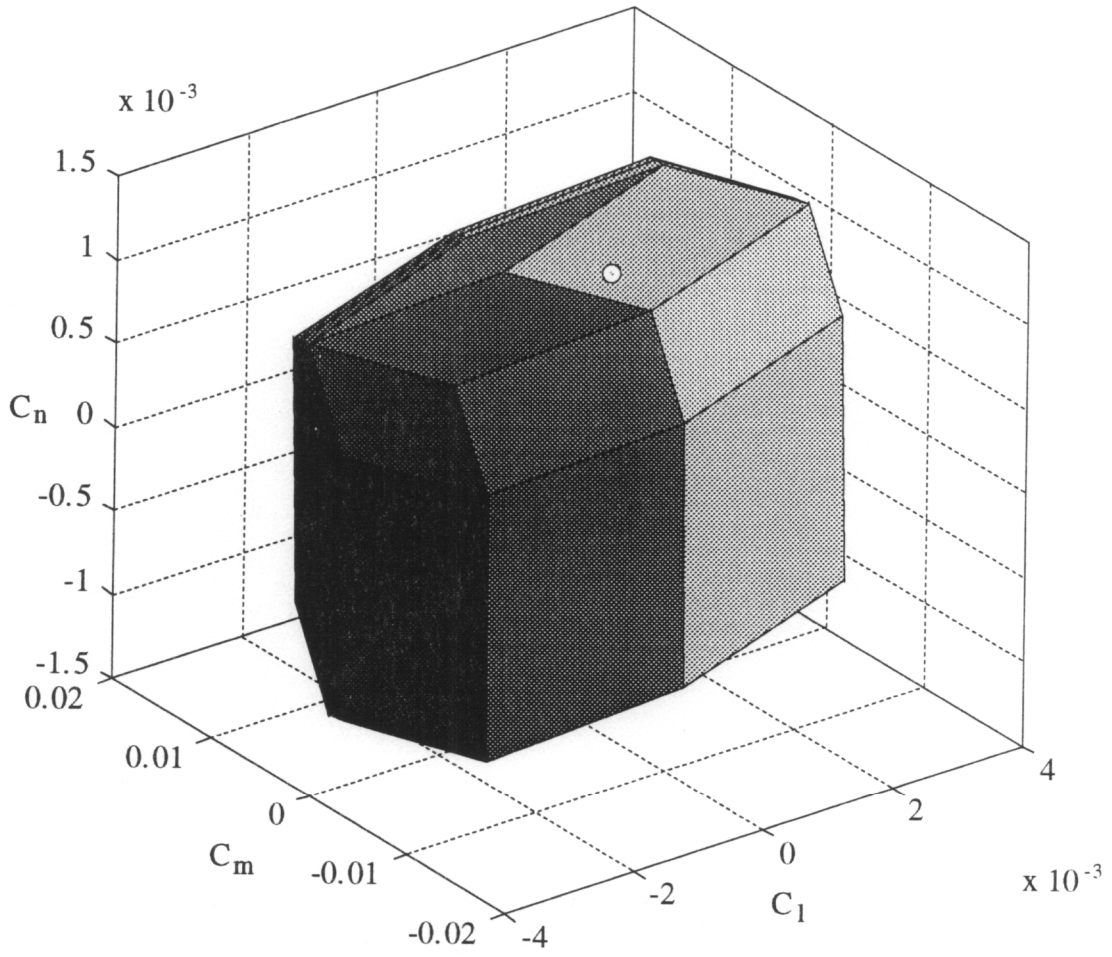


Figure 12-23: $\Delta\Phi$ at t = 0.425s, i =34

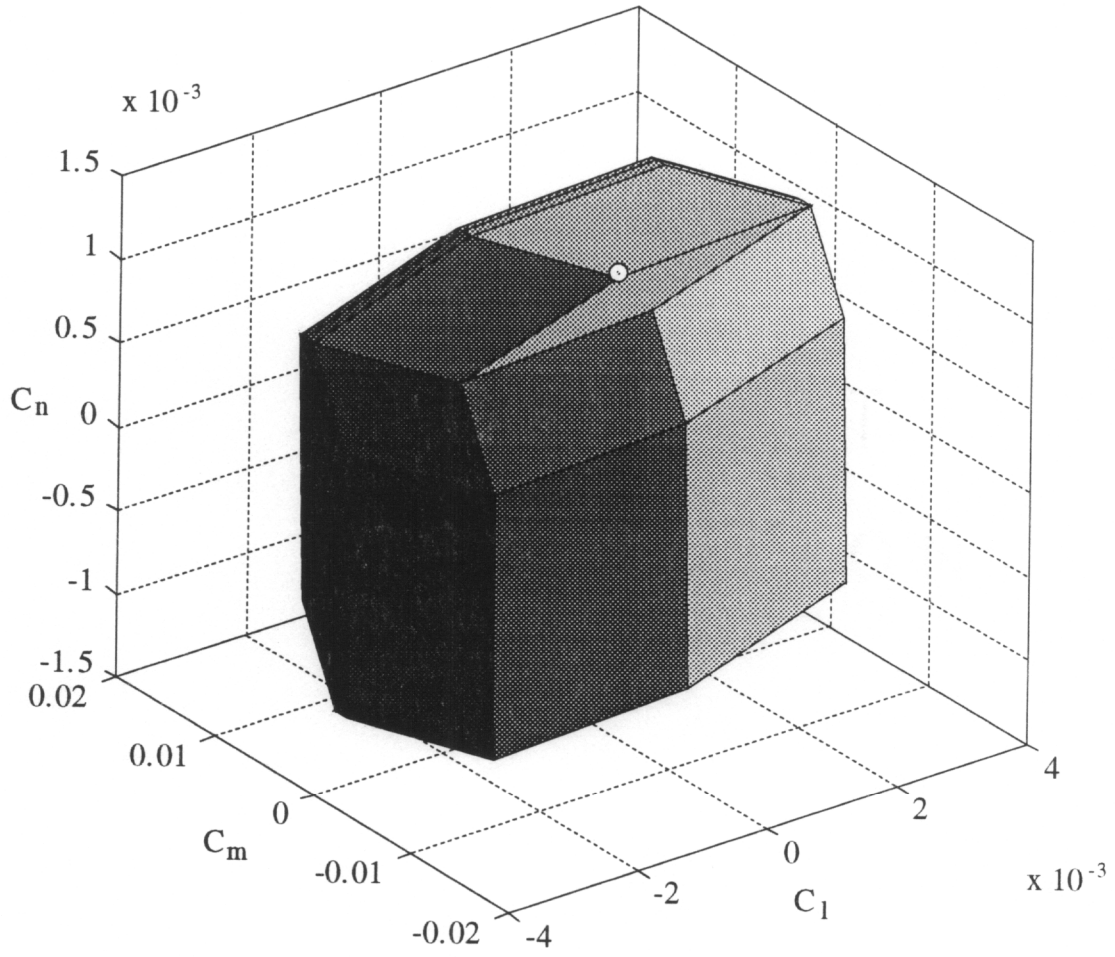


Figure 12-24: $\Delta\Phi$ at t = 0.4375s, i =35

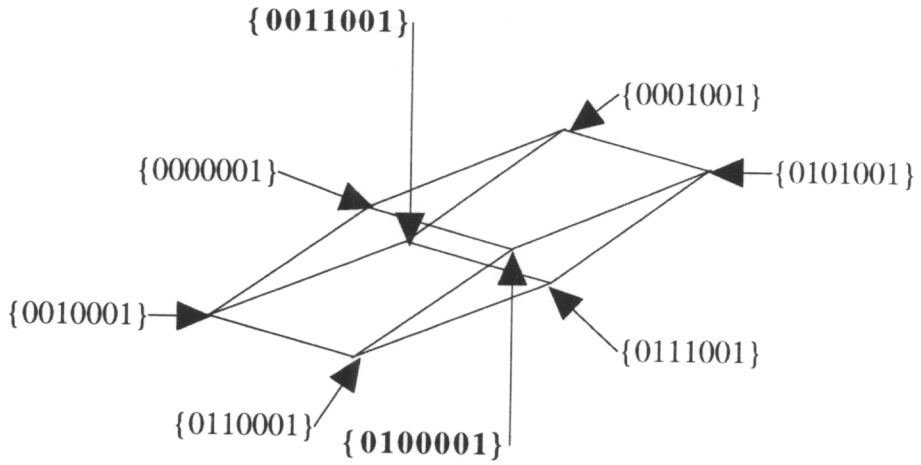
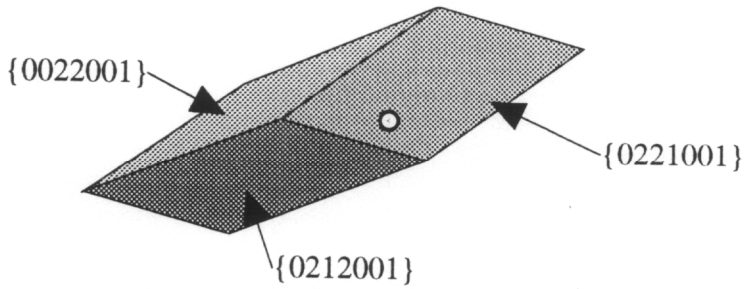
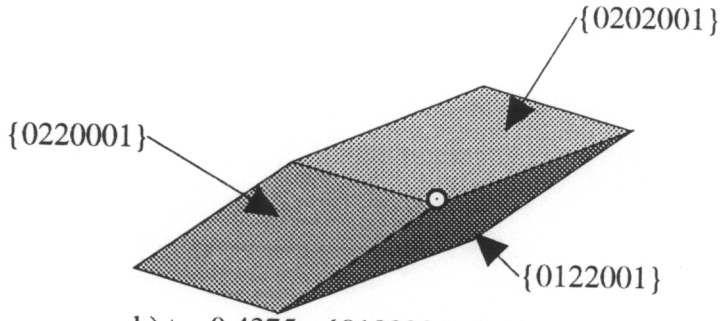


Figure 12-25: The box {0222001}



a) t = 0.425s, {0011001} Exterior



b) t = 0.4375s, {0100001} Exterior

Figure 12-26: The Faces of {0222001} which are on $\partial(\Delta\Phi)$

As a result, the faces which are on the exterior of Φ flip between [{0212001}, {0221001}, {0022001}] (at t = 0.0425s) and [{0202001}, {0220001}, {0122001}] (at t = 0.4375s). See Figure 12-26. The chattering seen from time t = 0.85 to t = 1.0625 is caused by similar circumstances.

A distinctive feature of Figure 12-22 is the sudden increases in the error at t = 0.775, t = 1.3375, and from t = 5.7625 onward. These spikes are cause by the same condition. At these times, one of the ailerons is changing sign. Recall that the yawing moment coefficient is discontinuous with respect to the ailerons, see Figure 12-9. When the aileron deflection is small, the local slope predicts a sign change in aileron deflection will produce a sign change in the yawing moment due to aileron deflection. However, the sign of the yawing moment produced does not change and errors result. The attempt to correct this error causes the chattering at the end of the maneuver. The right aileron deflection is tending to go to zero, but chatters back and forth because of the discontinuous derivatives.

The chattering in the error signal from t = 2 to t = 3.225 is the result of the face which is used to allocate the controls changing because the desired moment points almost at an edge and the facet used to allocate the controls changes faces rapidly. See Figure 12-27. However, this error does not cause a noticeable amount of chatter in any particular control, nor relatively large changes in the error.

While the chattering does not violate the rate limits on the controls, it can cause other problems. For example, such rapid changes can cause greater wear on the actuators and increase life-cycle costs. Also, it is unlikely that the physical actuators are capable of such rapid changes in the velocity of the controls. The acceleration limits on the controls

are not accounted for in this method and would likely be violated by such rapid control movements.

Example 12-2: Filtering Δu

One current method which attempts to reduce the chattering is to implement a digital low-pass filter [36] on the commanded velocity signal, $\Delta \mathbf{u}$. For example, a 20Hz low-pass filter will be added to the Discrete Time Direct Allocation algorithm used in Example 12-1. In continuous time, the transfer function for a low-pass filter with a 20Hz ($\approx$125.7rad/s) breakpoint is:

$$G(s) = \frac{125.7}{s + 125.7} \tag{12-26}$$

This transfer function can be converted to a discrete time transfer function using several methods. [36] Using the MATLAB™ command C2DM with a sample time of 0.0125 and a first-order hold, the discrete transfer function is:

$$G(z) = \frac{0.4957z + 0.2964}{z - 0.2079} \tag{12-27}$$

Calling the commanded change in the control at particular frame $\Delta \mathbf{u}_{ci}$ and the filtered change in control $\Delta \mathbf{u}_{fi}$ (See Figure 12-28), Equation 12-27 can be written as the difference equation:

$$\Delta \mathbf{u}_{fi} = 0.4957\Delta \mathbf{u}_{ci} + 0.2964\Delta \mathbf{u}_{ci-1} + 0.2079\Delta \mathbf{u}_{fi-1} \tag{12-28}$$

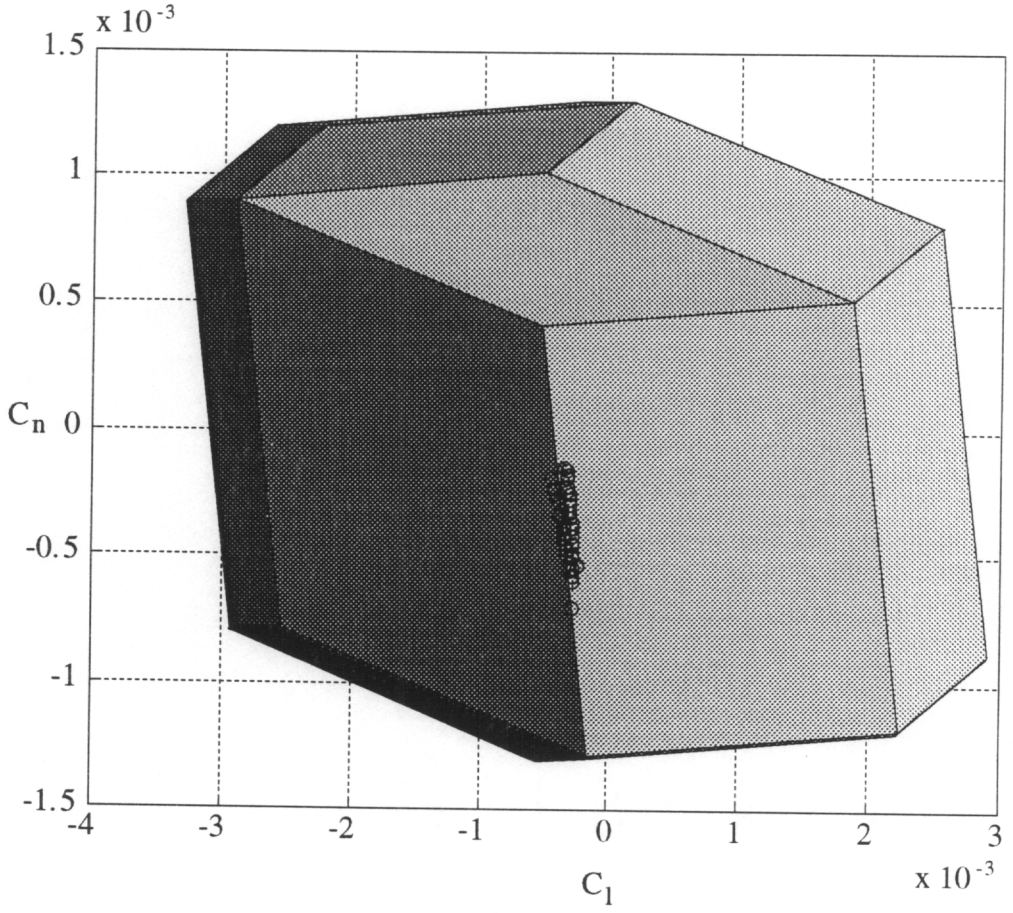


Figure 12-27: Chatter Along an Edge of $\Delta\Phi$

The effect of this filter is to prevent rapid changes in the velocity of the controls without preventing the controls from achieving their maximum velocities as specified by their rate limits. Figure 12-29 shows the effect of the filter on the right horizontal tail commanded to move at its maximum rate. Note that using this filter does introduce an undesirable time delay. See Figure 12-30.

Figure 12-31 shows the controls allocated using Discrete Time Direct Allocation with the digital filter from Equation 12-28. Figure 12-32 shows the magnitude of the error using this method. It can be seen from Figure 12-31 that this filter does not eliminate the chattering, but it does smooth out the control signal. It also has the effect of increasing by a small amount both the maximum error (from 6.1047e-4 to 9.5606e-4) and the average error (from 4.5691e-5 to 5.7461e-5).

Error Reduction

The magnitude of the error can be influenced through the choice of frame size, Δt. A reduction in frame size leads to a reduction in error. Figures 12-33 and 12-34 show the controls and the error for the Discrete Time Direct Allocation scheme using a time step twice the size of the one used in Example 12-1 ($\Delta t = 0.025$). For this example, a reduction in time step by a factor of 0.5 (from 0.025 to 0.0125) causes a reduction of average error by a factor of 0.5484 (from 8.3311e-05 to 4.5691e-5). For real-time use, there are practical limits on the size of Δt based on the capabilities of the electronics and hardware used.

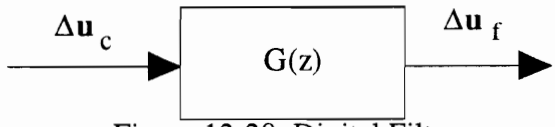


Figure 12-28: Digital Filter

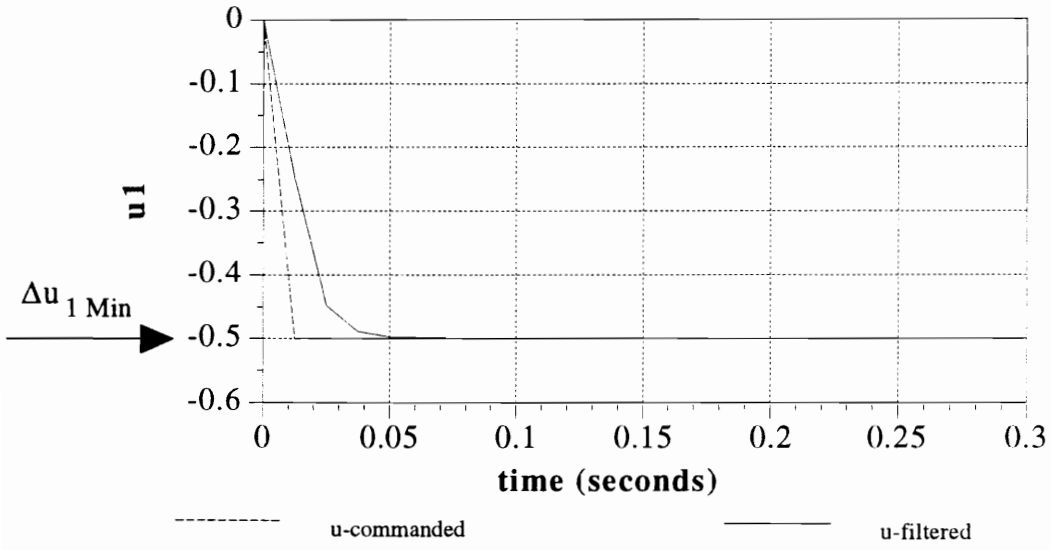


Figure 12-29: Commanded vs Actual Right Horizontal Tail

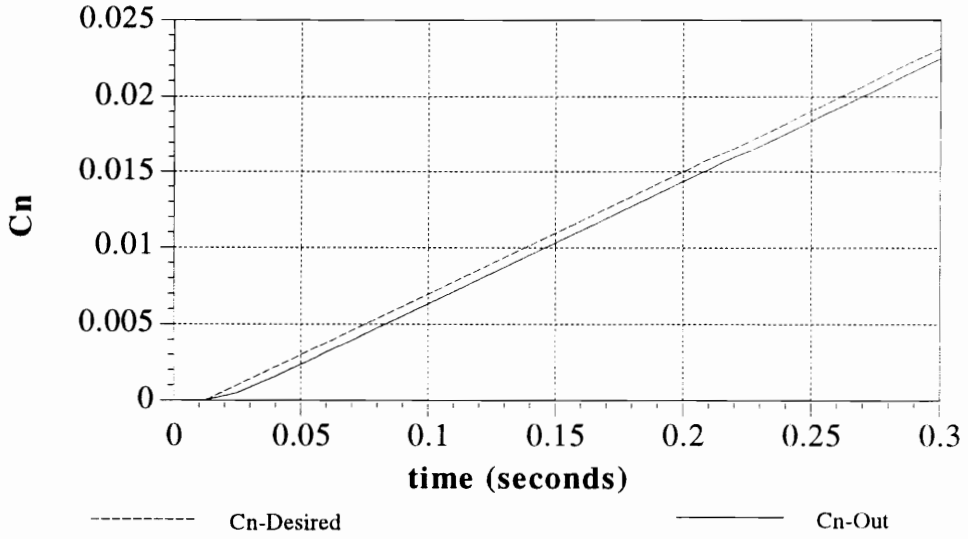


Figure 12-30: Time Delay Due to Low-Pass Filter

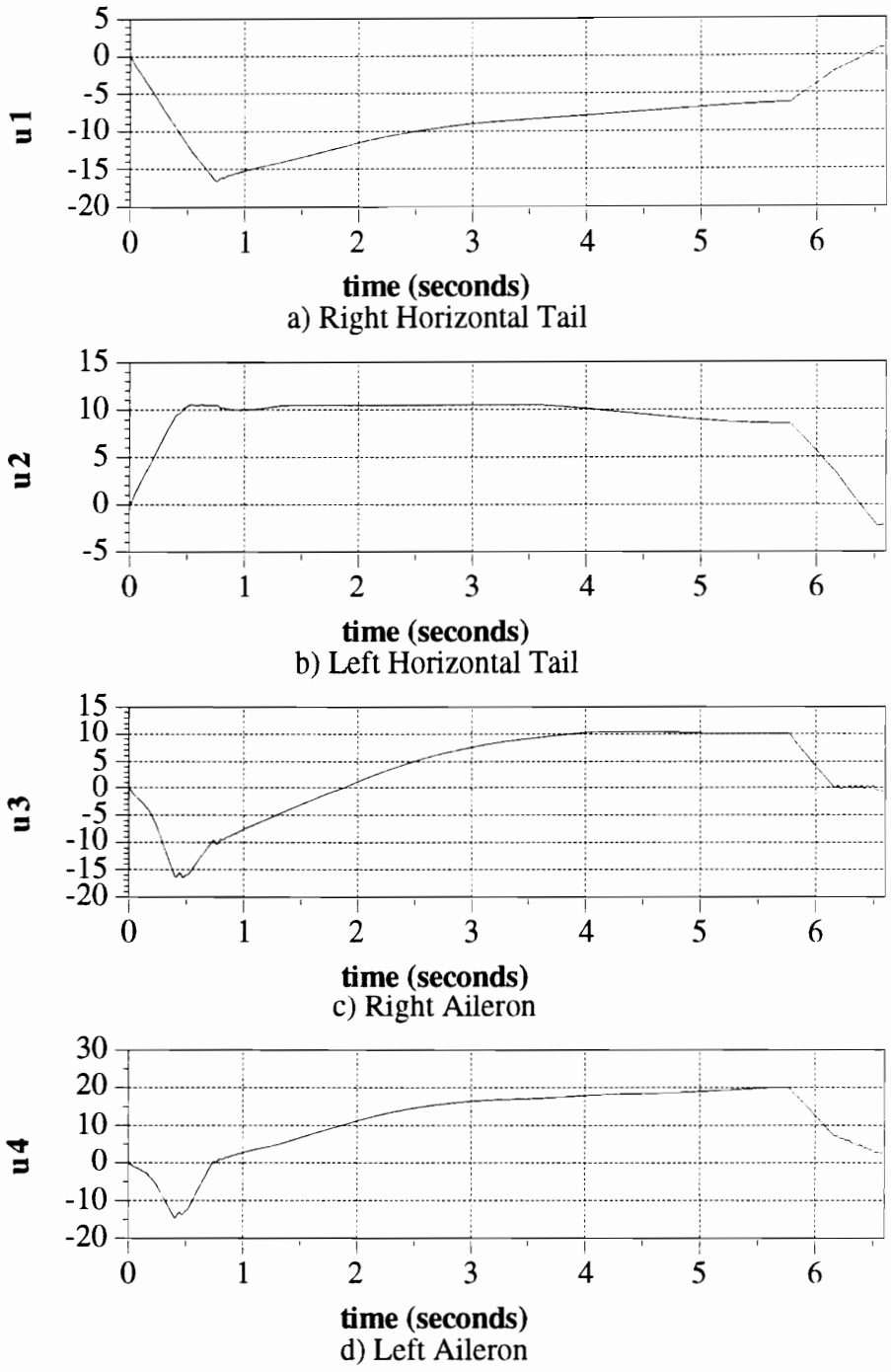
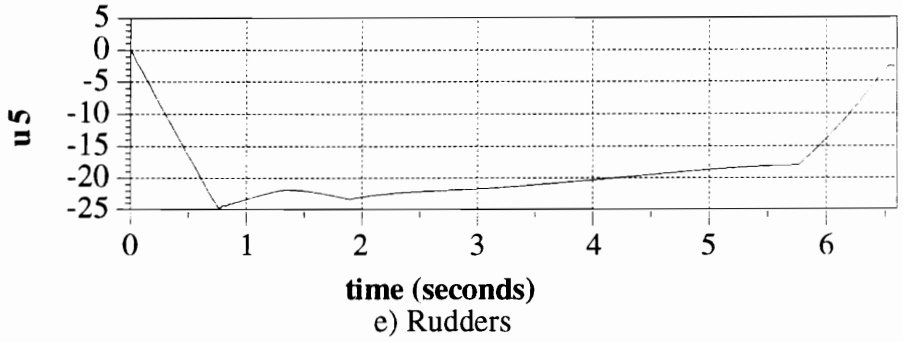
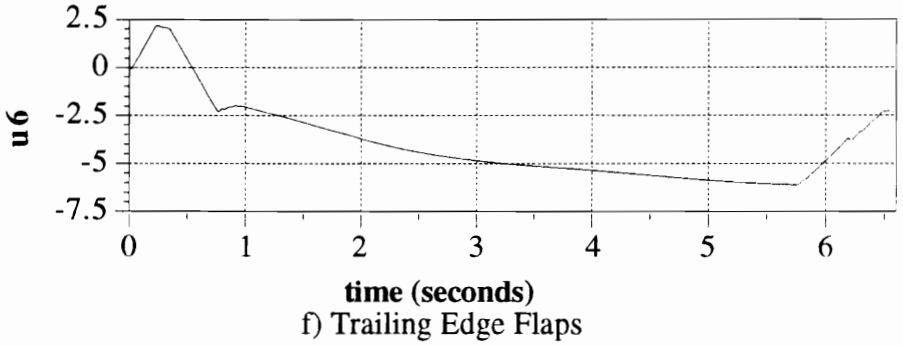


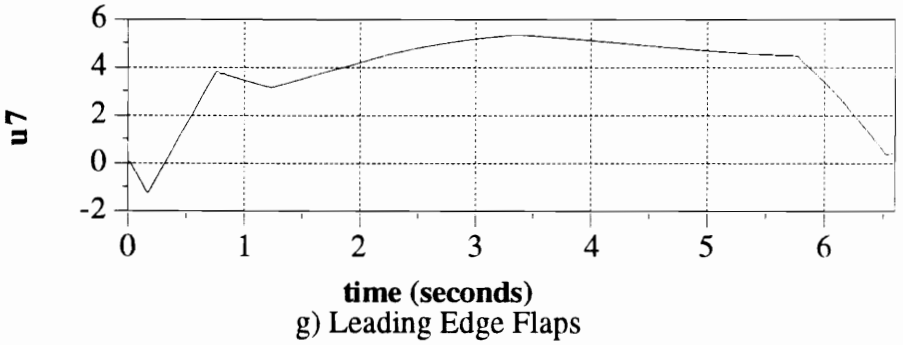
Figure 12-31: Controls Using DTDA with 20 Hz Filter



e) Rudders



f) Trailing Edge Flaps



g) Leading Edge Flaps

Figure 12-31: Controls Using DTDA with 20Hz Filter

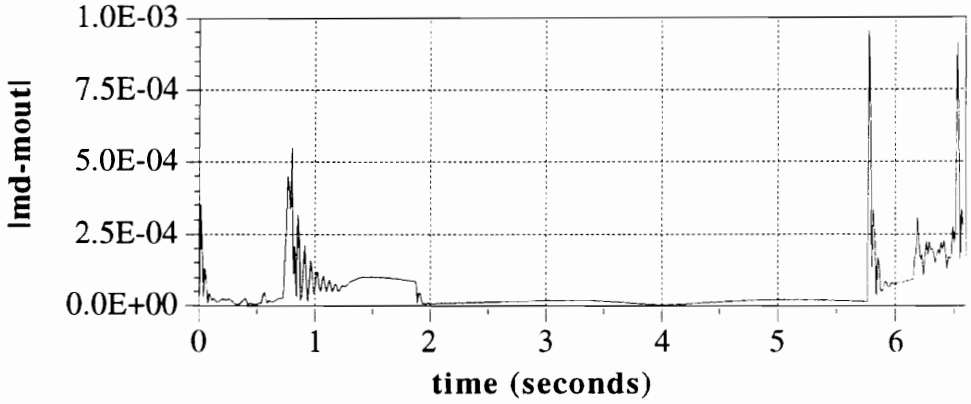


Figure 12-32: Error from DTDA with 20Hz Filter

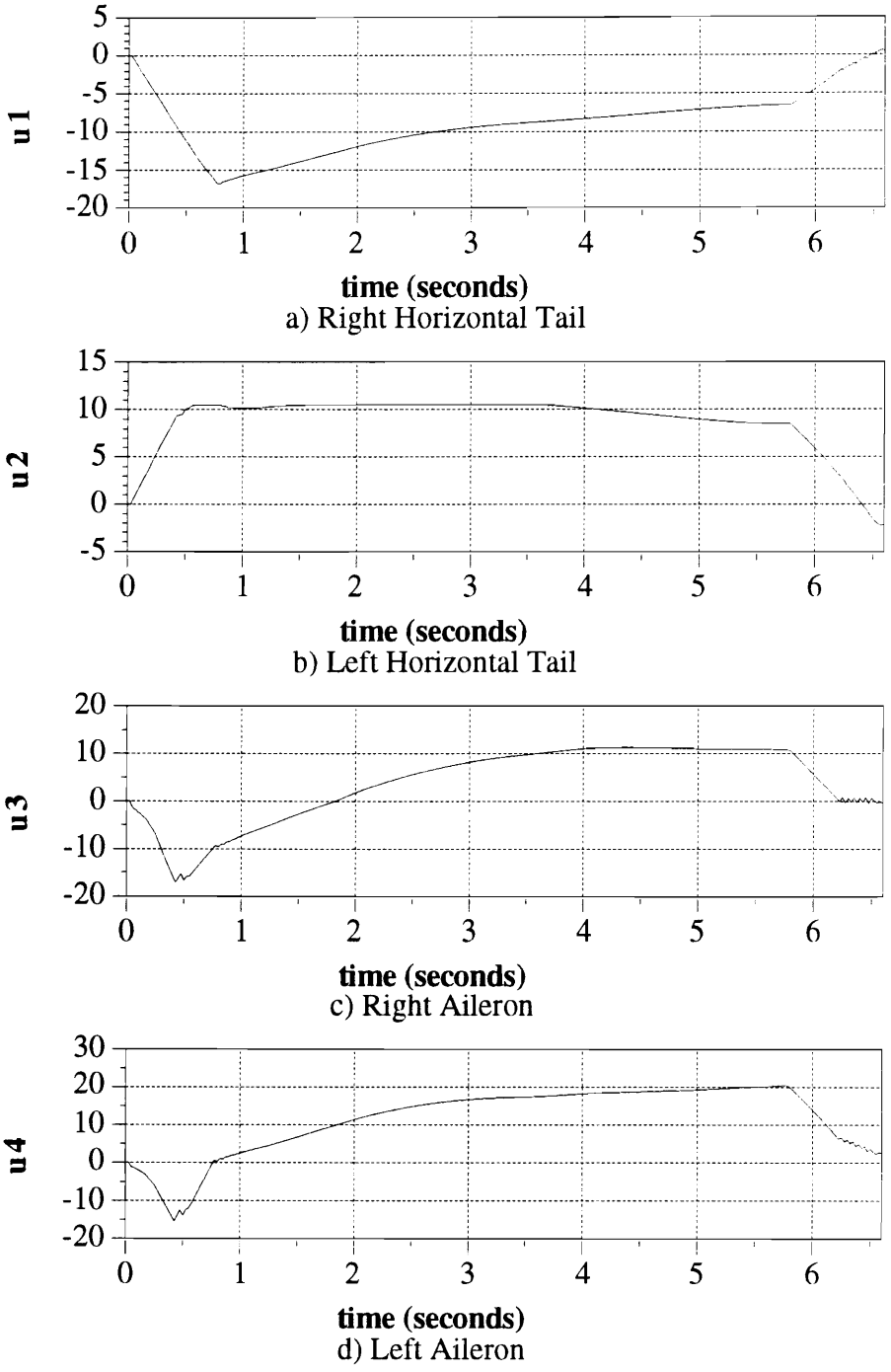
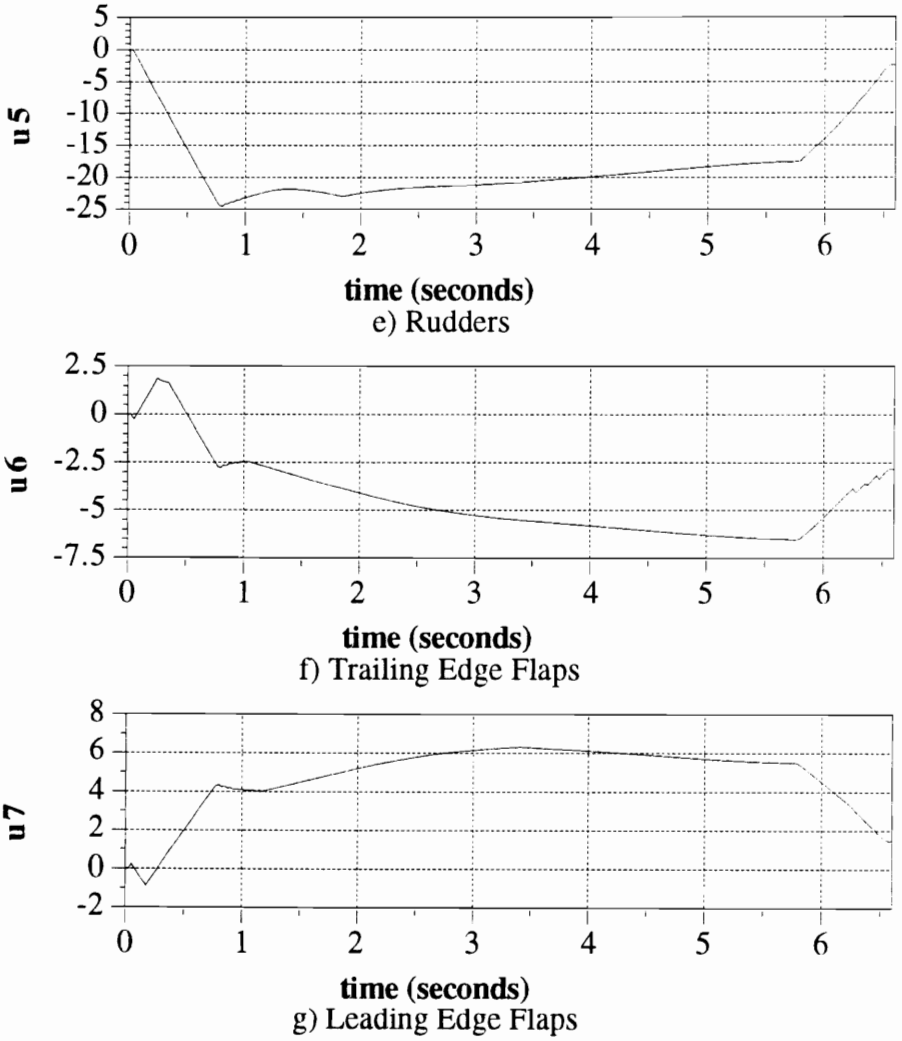


Figure 12-33: Controls Using DTDA with $\Delta t = 0.025$



e) Rudders

f) Trailing Edge Flaps

g) Leading Edge Flaps

Figure 12-33: Controls Using DTDA with $\Delta t = 0.025$

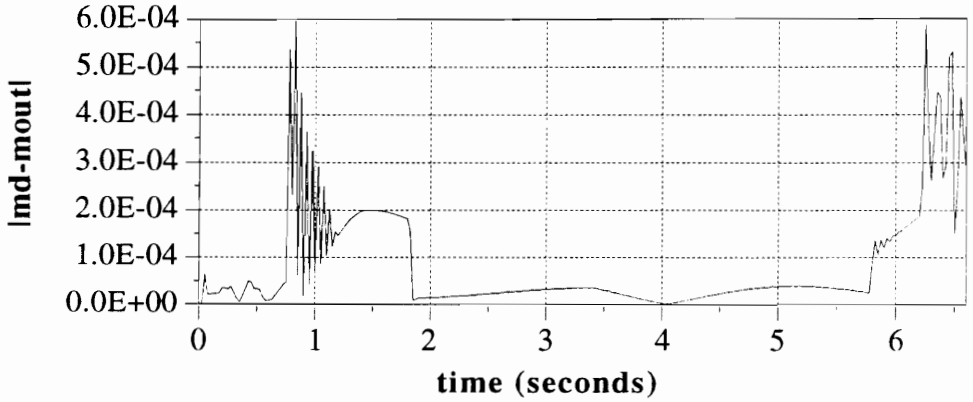


Figure 12-34: Error Using DTDA with $\Delta t = 0.025$

Example 12-1 uses realistic airplane data and a time step typical of modern aircraft. The moments produced by Discrete Time Direct Allocation are all within 0.032% of the commanded values and are on average within 0.0023% of the commanded values. The relatively small size of these errors suggests that current technology should be capable of producing acceptably small errors.

Path Dependency

Another issue which arises in Discrete Time Allocation is the fact that the allocation of the controls is now path dependent. This can be seen from an examination of Equation 12-6, $\Delta\mathbf{m} = B_{ref}\Delta\mathbf{u}$. Expanding this equation, it becomes:

$$\mathbf{m}_{d\,i} - \mathbf{m}_{out\,i-1} = B(\mathbf{u}_{i-1})\Delta\mathbf{u} \tag{12-29}$$

$$\mathbf{u}_i = \mathbf{u}_{i-1} + \Delta\mathbf{u} \tag{12-30}$$

From Equation 12-30, it is seen that the current controls, $\mathbf{u}_i$, depend upon the value of the previous controls, $\mathbf{u}_{i-1}$. From Equation 12-29, it is seen that the commanded change in controls depends upon the previous controls because B is a function of $\mathbf{u}_{i-1}$ and $\mathbf{m}_{out\,i-1}$ depends upon $\mathbf{u}_{i-1}$. Thus, the controls allocated for a particular desired moment, $\mathbf{m}_d$, are effected by the controls previously commanded.

Path dependency can have some undesirable effects. For example, the desired moments for Example 12-1 begin at zero and return to zero when the maneuver is completed. Most control allocation schemes will command zero control deflections for zero commanded moments, see Figures 12-11 through 12-14. However, moment commands of zero do not result in control deflection commands of zero for the discrete methods, Figures 12-15 and 12-16. As a result, a zero commanded moment produces

excess controls whose moment effects cancel each other out. This typically produces other undesirable aerodynamic effects, such as poor drag characteristics.

Additionally, the path dependency can cause controls to saturate unnecessarily. This saturation can produce errors when using a pseudo-inverse solution. When using Direct Allocation, the saturated controls reduce the size of $\Delta\Phi$. For example, if a control which can normally travel $\pm 0.5^\circ$ in one time frame (for a total range of 1°) is saturated at its positive limit, it can only travel -0.5°. This saturation reduces the set of possible moments produced, $\Delta\Phi$. This reduction of $\Delta\Phi$ can cause moments which ought to be attainable to become unattainable.

The best way to deal with the path dependency is a subject for future research. However, several methods have been explored which utilize function minimization to reduce the excess control commands. This subject is dealt with as a part of Section 13.

13. FUNCTION OPTIMIZATION

Capabilities of Redundant Controls

In general, a set of redundant controls have effects other than that of moment generators. It is frequently desirable to optimize some other effect of the controls, such as: minimizing drag due to the controls, reducing the norm of the control vector, minimizing hydraulic flow rate, reducing radar cross section, and maximizing lift-to-drag ratio. This optimization can be done using a wide variety of techniques, but will only be discussed here within the context of Discrete Time Direct Allocation.

Description of Method

Consider some scalar function of the controls, f(**u**), which is either known explicitly or is contained in a data base. If the gradient of this function, $\left.\frac{\partial f}{\partial \mathbf{u}}\right|_{ref}$, is also known, the controls can be allocated to drive this function toward an extremum. This control allocation is done in a two-step process. The controls which satisfy the moment equations, $\Delta\mathbf{u}_m$, are found using:

$$\Delta\mathbf{m}_d = B_{ref}\Delta\mathbf{u}_m \tag{13-1}$$

The moment equations are augmented with the function gradient to produce a larger B matrix:

$$\begin{Bmatrix} \Delta\mathbf{m}_d \\ f(\mathbf{u}) \end{Bmatrix} = \begin{bmatrix} B(\mathbf{u}_{ref}) \\ \left.\frac{\partial f}{\partial \mathbf{u}}\right|_{ref} \end{bmatrix} \Delta\mathbf{u} \tag{13-2}$$

Controls which optimize the function without affecting the moment produced, $\Delta\mathbf{u}_f$, can be found by specifying the left hand side of Equation13-2 to be:

$$\begin{Bmatrix} \Delta\mathbf{m}_d \\ f(\mathbf{u}) \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ \pm k \end{Bmatrix} = \begin{bmatrix} B(\mathbf{u}_{ref}) \\ \left.\frac{\partial f}{\partial \mathbf{u}}\right|_{ref} \end{bmatrix} \Delta\mathbf{u}_f \tag{13-3}$$

Where k is some arbitrary positive constant. If the function is to be minimized, -k is use, and if the function is to be maximized, +k is used . If Direct Allocation is used to solve Equation 13-3 for $\Delta\mathbf{u}_f$, the size of k is irrelevant, because Direct Allocation can find the intersection of the desired moment direction with the boundary of the $\Delta\Phi$ (now (n+1)-D). The controls $\Delta\mathbf{u}_m$ will produce some value of the function to be optimized. See Figure 13-1a. It is important to change the limits on the controls when using Direct Allocation so that the origin is centered at the commanded moment. See Figure 13-1b. The control limits are changed by subtracting the controls which produce $\mathbf{m}_d$ from the control constraints:

$$\Delta\mathbf{u} = \Delta\mathbf{u}_m + \Delta\mathbf{u}_f \tag{13-4}$$

$$\Delta\mathbf{u}_{Min} \leq \Delta\mathbf{u} \leq \Delta\mathbf{u}_{Max} \tag{13-5}$$

$$\Delta\mathbf{u}_{Min} - \Delta\mathbf{u}_m \leq \Delta\mathbf{u}_f \leq \Delta\mathbf{u}_{Max} \tag{13-6}$$

The two control solutions can then be added together to produce the final controls:

$$\Delta\mathbf{u}_m + \Delta\mathbf{u}_f = \Delta\mathbf{u} \tag{13-7}$$

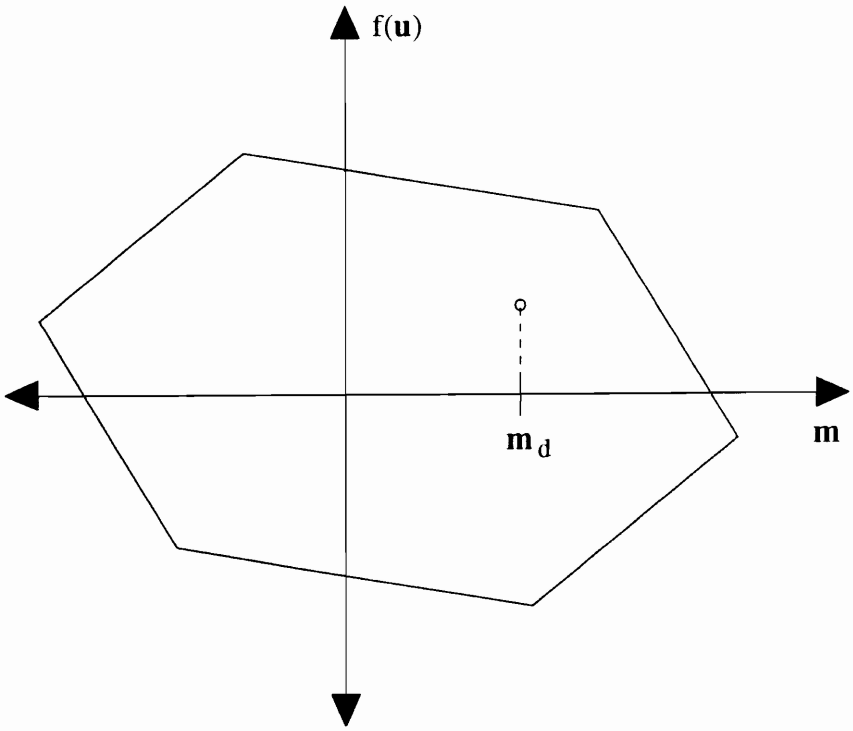
It is important to note that this method does not truly optimize the function, it only moves along the gradient towards an optimal point. Depending upon the shape of the function and the size of the steps taken, the optimal point may never actually be reached.

Also, this method prioritizes the satisfaction of the moment demands, and uses only remaining capabilities to optimize the function. As a result, a large change in moment demand may prohibit any movement towards the optimal point.

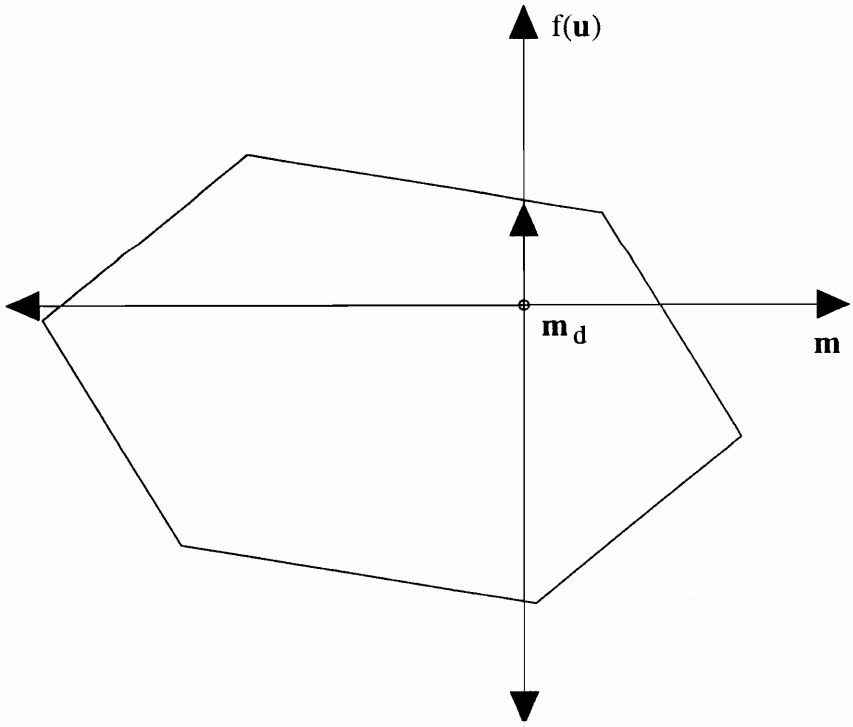
Computational Considerations

Solving for $\Delta\mathbf{u}_f$ using Direct Allocation allows the algorithm to move in the direction of the desired gradient as much as the limits of the controls will allow. However, the maximum allowable step size is not always desirable. A large step size can cause the controls to "step over" the optimal point. See Figure 13-2. Also, taking such large steps means that the controls are constantly being driven at their limits. Such large demands on the hydraulic system can increase fatigue and the life-cycle cost for the aircraft. Bearing this fact in mind, it is recommended that the size of $\Delta\mathbf{u}_f$ be scaled down to improve the behavior of the controls. Determining the best size for the scale factor is a complex issue and can be a subject for future research.

Solving Equation 13-3 using Direct Allocation is fairly computationally expensive because the number of computations required increases with n. The main benefit of using Direct Allocation is that the maximum step size that does not violate any of the control constraints is found. However, since $\Delta\mathbf{u}_f$ will normally be scaled down from this maximum value, a less computationally expensive method, such as using a pseudo-inverse can be used.



a) (n+1)-D $\Delta\Phi$ with a Desired Moment



b) (n+1)-D $\Delta\Phi$ with Origin Relocated

Figure 13-1: Using Direct Allocation to Minimize a Function, f(**u**)

When solving for $\Delta\mathbf{u}_f$ using a pseudo-inverse, the magnitude of k in Equation 13-3 becomes important because the magnitude of $\Delta\mathbf{u}_f$ will be proportional to k. Solving Equation 13-3 with a pseudo-inverse may cause the sum of $\Delta\mathbf{u}_m$ and $\Delta\mathbf{u}_f$ to violate the constraints on $\Delta\mathbf{u}$. To avoid this problem, $\Delta\mathbf{u}_f$ should be scaled so that the constraints on $\Delta\mathbf{u}$ are not violated.

This scaling can be done by solving for a scale factor which will produce a $\Delta\mathbf{u}$ with one or more of the controls at a limit and the rest of the controls within their constraints:

$$\Delta\mathbf{u}_m + \alpha\Delta\mathbf{u}_f = \Delta\mathbf{u}_{(sat)}, \text{ where } \alpha \geq 0 \tag{13-8}$$

If Direct Allocation is used to determine $\Delta\mathbf{u}_m$, there will always exist some $\alpha \geq 0$ for which Equation 13-8 satisfies the constraints on $\Delta\mathbf{u}$ because $\Delta\mathbf{u}_m$ satisfies the constraints on $\Delta\mathbf{u}$. The α which satisfies Equation 13-8 will again drive the controls at some limit. To avoid driving the controls at their rate limits, preliminary studies have indicated that α should be restricted to be less than or equal to one, and possibly further scaled down to provide a good balance between rate of convergence and control fatigue.

Additionally, the function which is chosen to be optimized can have undesirable characteristics which have detrimental effects on the controls allocated. Several undesirable characteristics are: discontinuous derivatives, multiple extreme points, or extreme points at infinity. Discontinuous derivatives can cause excessive chattering in the controls. Multiple extreme points can cause the controls to be driven toward a non-global minimum or maximum. Extreme points at infinity will drive the controls toward saturation which can reduce the size of $\Delta\Phi$, as discussed in Section 12.

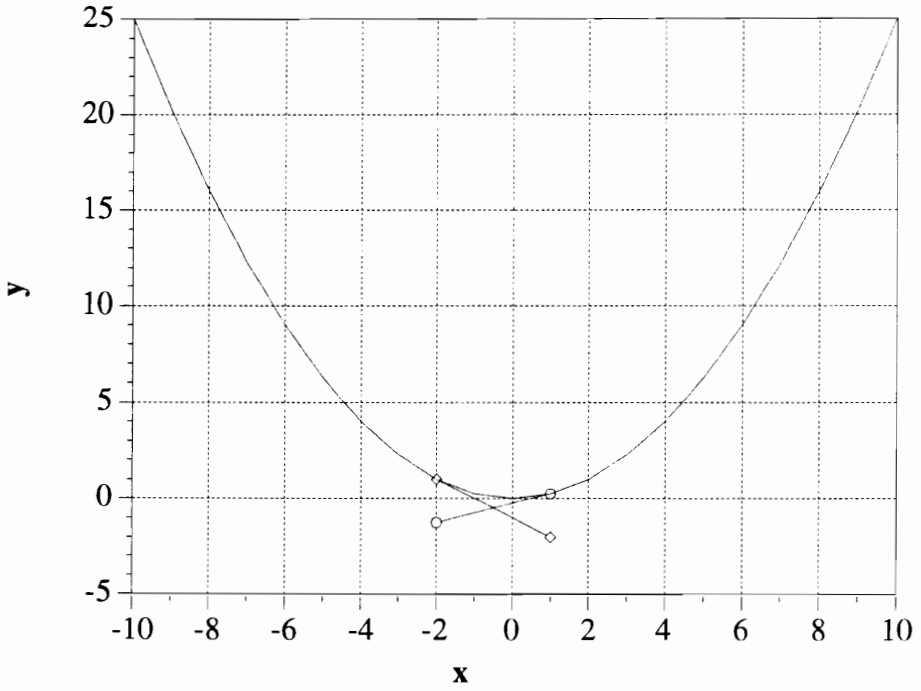


Figure 13-2: Large Steps can Overshoot Minimum

Example 13-1: F-18 Minimize $\|\mathbf{u}\|_2$

This example applies Discrete Time Direct Allocation to the F-18 data from Example 12-1 and the 20Hz digital filter from Example 12-2. The desired maneuver is the velocity vector roll used in Examples 12-1 and 12-2. The function to be minimized is the magnitude of the control vector, $\|\mathbf{u}\|_2$. This function was chosen because it is smoothly varying with a unique global minimum, and it is easily computed. Also, by attempting to minimize the control deflections, the control allocation algorithm will try to remove the excess controls which may result from the path dependency.

$$f(\mathbf{u}) = 0.5\left(\sum_{i=1}^{7} u_i^2\right) \tag{13-9}$$

$$\left.\frac{\partial f}{\partial \mathbf{u}}\right|_{ref} = [u_1\ u_2\ u_3\ u_4\ u_5\ u_6\ u_7]_{ref} \tag{13-10}$$

$\Delta\mathbf{u}_f$ is determined solving Equation 13-11 using the pseudo-inverse.

$$\begin{Bmatrix} 0 \\ 0 \\ 0 \\ -1 \end{Bmatrix} = \begin{bmatrix} B(\mathbf{u}_{i-1}) \\ \left.\frac{\partial f}{\partial \mathbf{u}}\right|_{i-1} \end{bmatrix} \Delta\mathbf{u}_f \tag{13-11}$$

If $\Delta\mathbf{u}_m + \Delta\mathbf{u}_f$ satisfies the constraints on $\Delta\mathbf{u}$, α is set equal to 1. Otherwise, α is computed to satisfy:

$$\Delta\mathbf{u}_m + \alpha\Delta\mathbf{u}_f = \Delta\mathbf{u}_{(sat)}, \text{ where } 1 \geq \alpha \geq 0 \tag{13-12}$$

The final controls are computed using an additional scale factor of 0.5, which was chosen to illustrate the effects of additional scaling:

$$\Delta\mathbf{u} = \Delta\mathbf{u}_m + 0.5\alpha\Delta\mathbf{u}_f \tag{13-13}$$

Figure 13-3 shows the controls allocated using this method. Figure 13-4 shows the error produced by these controls. Figure 13-5 shows the magnitude of the control vector during the maneuver, comparing the controls from this method (solid line) with the controls from Example 12-2 (dashed line). Table 13-1 summarizes the results.

By augmenting $\Delta\mathbf{u}_m$ with $\Delta\mathbf{u}_f$, the magnitude of the control vector is reduced during the maneuver, and the final control positions are much closer to zero. However, the chattering of the controls becomes worse, and the average error is increased. The increase in error can be partly attributed to the chattering of the controls. However, the errors are still only a small portion of the desired moment, and average 0.4317% of the desired moment.

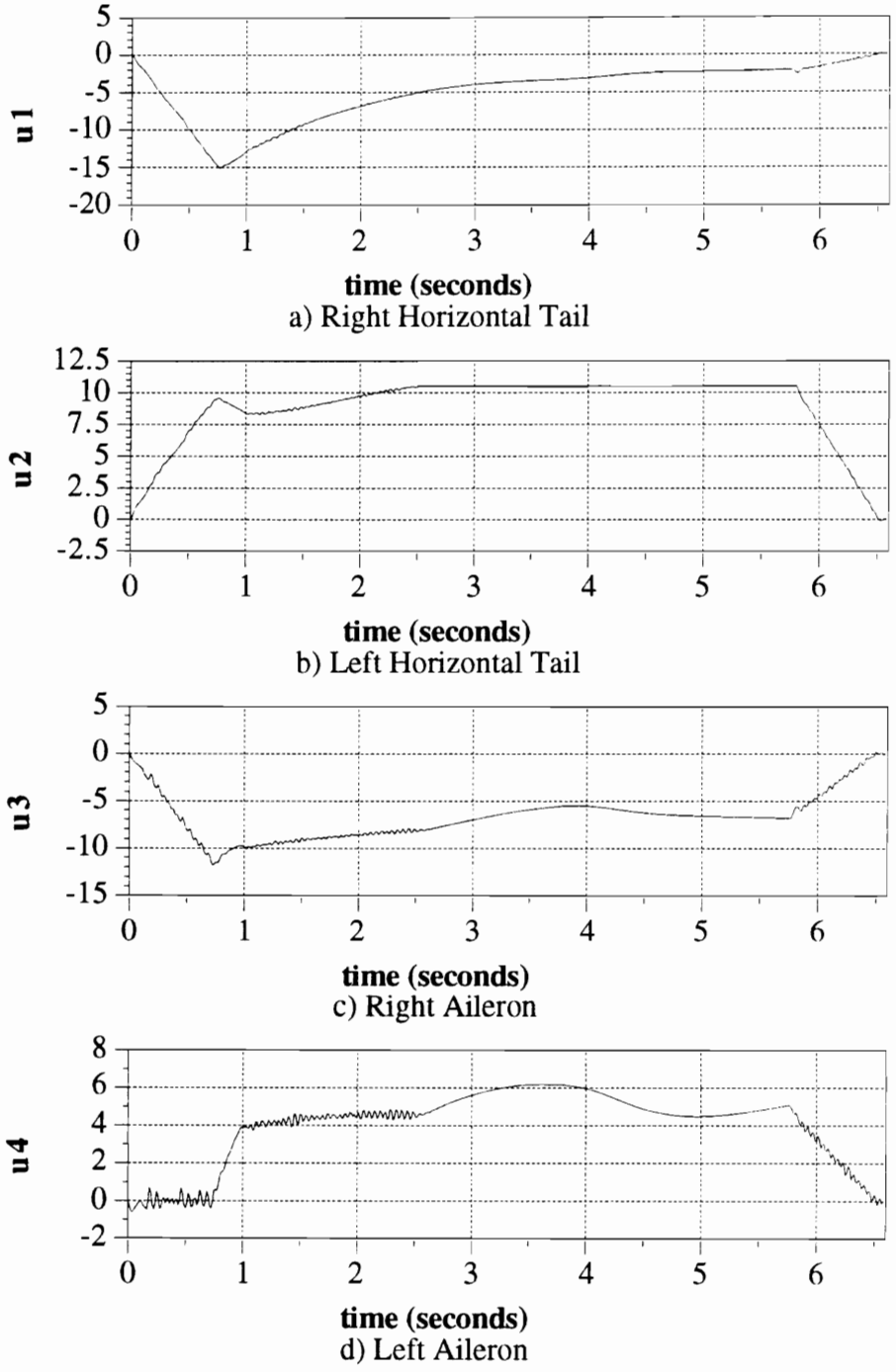
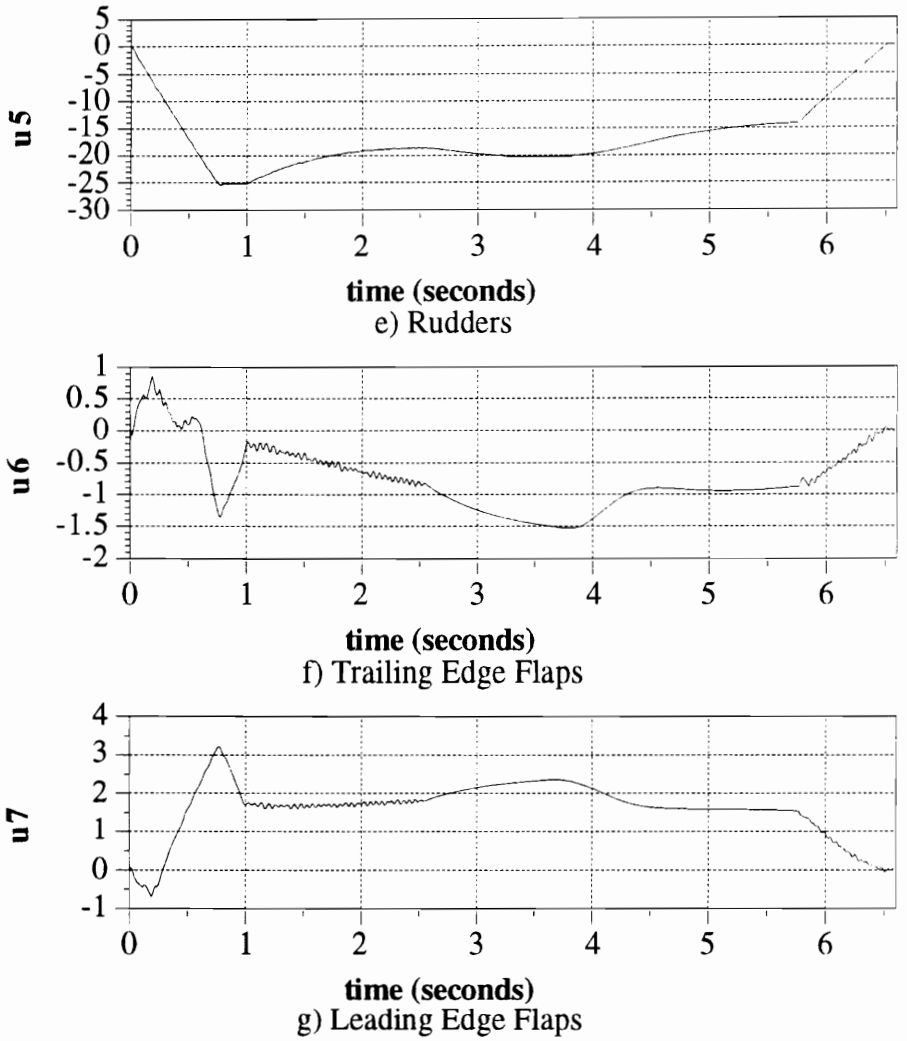


Figure 13-3: Controls Using DTDA and Minimizing $|\mathbf{u}|$



e) Rudders

f) Trailing Edge Flaps

g) Leading Edge Flaps

Figure 13-3: Controls Using DTDA and Minimizing $|\mathbf{u}|$

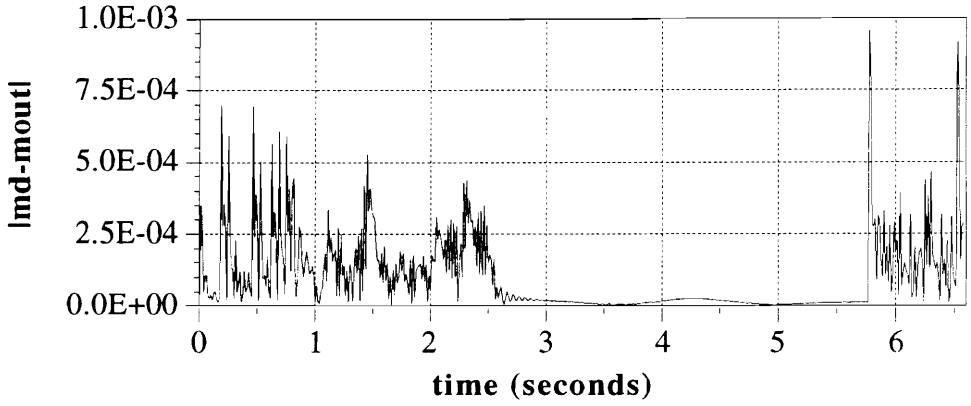


Figure 13-4: Error Using DTDA and Minimizing |**u**|

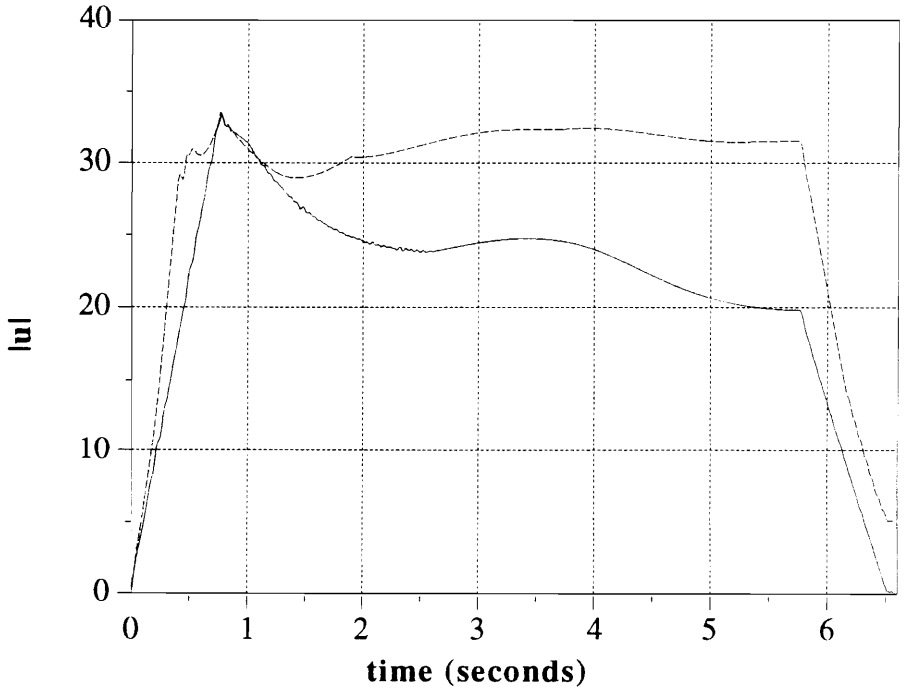


Figure 13-5: Comparison of |**u**| with and without Minimization

Table 13-1: Results Summary

	Average Error	Average $\|\mathbf{u}\|$	Final $\|\mathbf{u}\|$
With Minimization	1.0074e-04	21.4849	0.0875
Without Minimization	5.4883e-05	28.2475	5.0939

14. CONTROL FAILURES

Types of Failures

If a control actuator malfunctions in some way during flight, the moments produced by the controls will not be the same as the moments which are commanded. These malfunctions can produce very large errors depending upon the severity of the malfunction. Various types of malfunctions can occur, including, but not limited to:

- A surface may fail in some fixed position
- A failure may send a control to a fully deflected position
- The surface may be partially damaged, becoming less effective than the data predicts
- The surface may be fully detached, producing no moment

If there are sensors which determine the control deflections or measurements of the moments on the aircraft, the control allocation algorithm will tend to correct for these errors when $\Delta\mathbf{m}_d$ is computed. Sensor information will not completely eliminate the errors because the algorithm will attempt to reduce the errors in part by commanding the failed control. However, sensor information typically reduces the errors. The sensors themselves may be the source of error:

- A sensor may fail, telling the computer the control is at an incorrect position

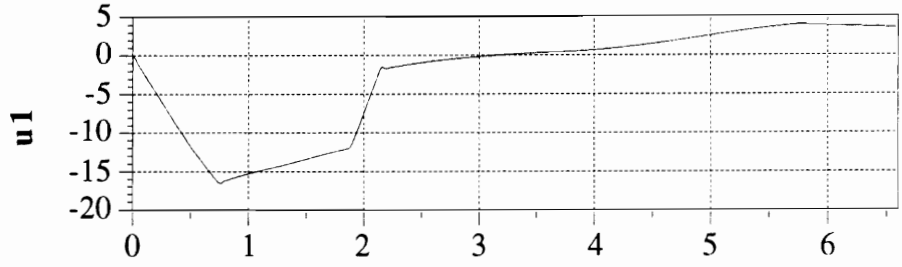
If the failed control is identified by some means, it can be accounted for by removing its corresponding column from the B matrix. When this removal is done, the control allocation algorithm can correct for the error by redistributing the functional controls.

If a large number of controls are lost, or if the controls lost are extremely effective, it may be possible that the aircraft is no longer controllable. However, by using Direct Allocation to redistribute the remaining controls, the maximum amount of moment generating capability available will be used. This fact is extremely important because it means that as long as the aircraft possesses sufficient control power to maneuver, the controls will generate all of the attainable desired moments. Put another way, the aircraft will not become uncontrollable because of deficiencies in the control allocation algorithm.

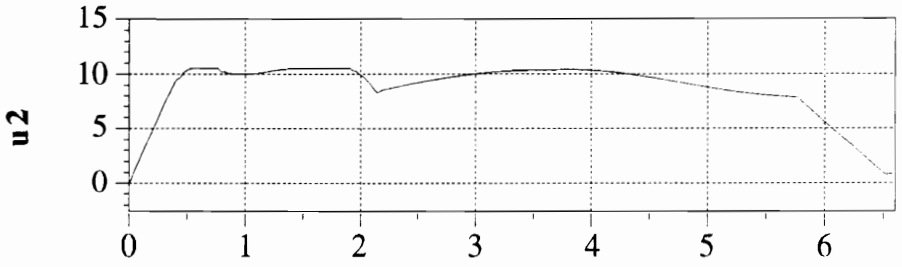
Example 14-1: An Unidentified Control Failure

This example uses the F-18 data from Example 12-1 and allocates the controls using Discrete Time Direct Allocation using the 20Hz low-pass filter from Example 12-2. The desired maneuver is the velocity vector roll used in Example 12-1.

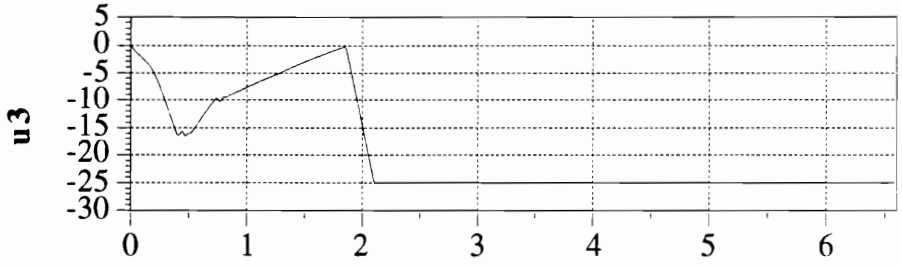
At t = 1.875s, a failure was simulated in which the right aileron goes to its full trailing edge up position (-25°) at its maximum rate. Figure 14-1 shows the controls allocated. Figure 14-2 shows the error produced by these controls compared to the error produced when the failure does not occur. Note that when the failed control stops moving, at t = 2.1125s, the moment due to the failed control becomes constant, and the error is quickly reduced to the same order of magnitude as the case with no failure. This reduction in error is a result of commanding moment changes based on the current moment, $\Delta \mathbf{m}_{d\,i} = \mathbf{m}_{d\,i} - \mathbf{m}_{out\,i-1}$. The current moment, $\mathbf{m}_{out\,i-1}$, is computed by the control law using information in the data base and possibly information provided from sensors on the aircraft.



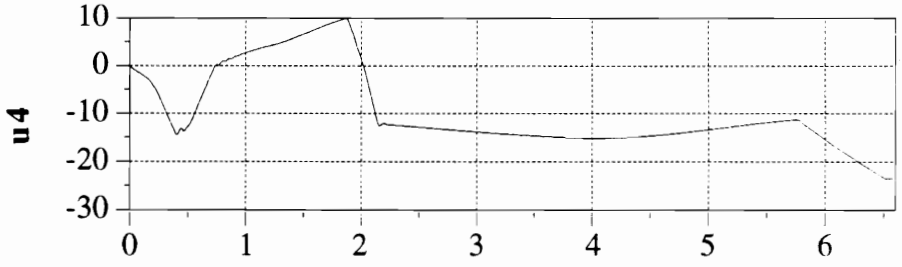
a) Right Horizontal Tail



b) Left Horizontal Tail

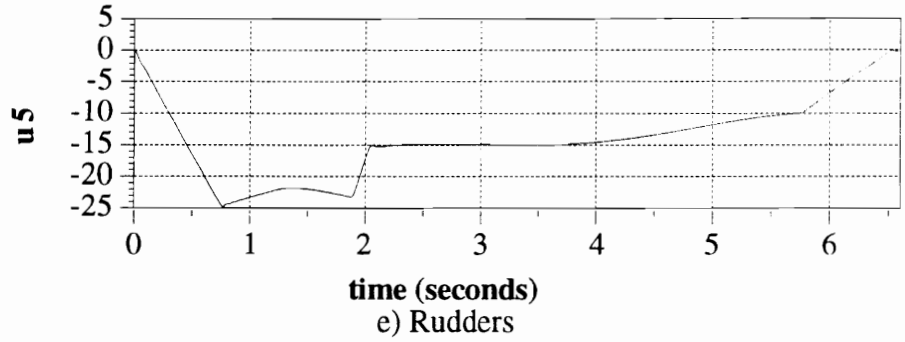


c) Right Aileron (with failure)

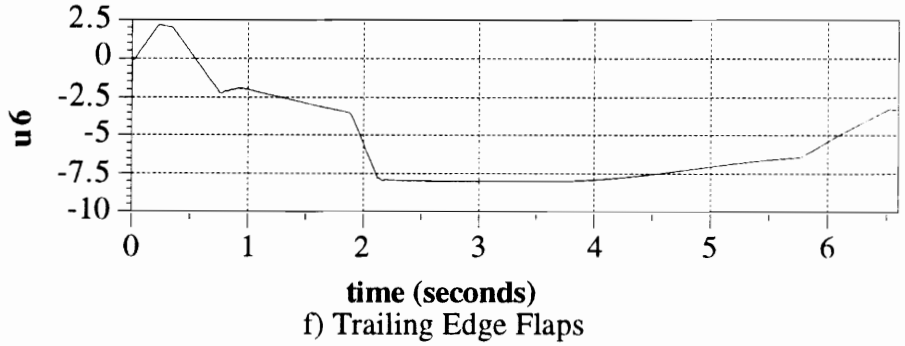


d) Left Aileron

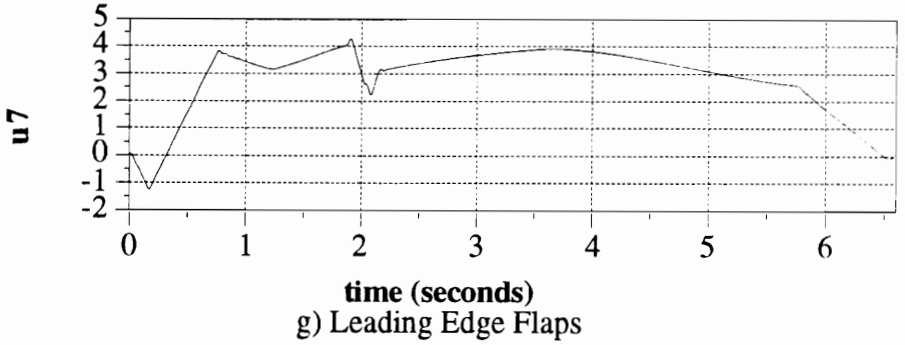
Figure 14-1: Controls from Example 14-1



e) Rudders



f) Trailing Edge Flaps



g) Leading Edge Flaps

Figure 14-1: Controls from Example 14-1

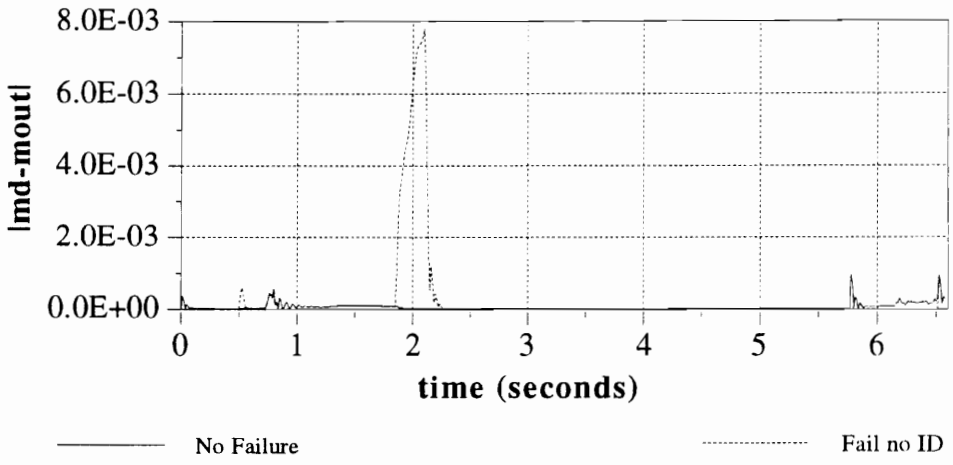


Figure 14-2: Error Induced by Failure

Example 14-2: An Identified Control Failure

This example is the same as Example 14-1, except that after 10 time frames, 0.125s, at t = 2s, the failed control is identified and its column is removed from the B matrix. Figure 14-3 shows the controls allocated using this method. Figure 14-4 shows the error produced by these controls and the error produced by the controls from Example 14-1. As soon as the error is identified, the algorithm begins attempting to reduce the error, even as the control failure tries to increase the error. As a result, the maximum error is reduced from 7.8090e-3 to 6.5545e-3 and the average error is reduced from 2.7151e-4 to 2.4640e-4.

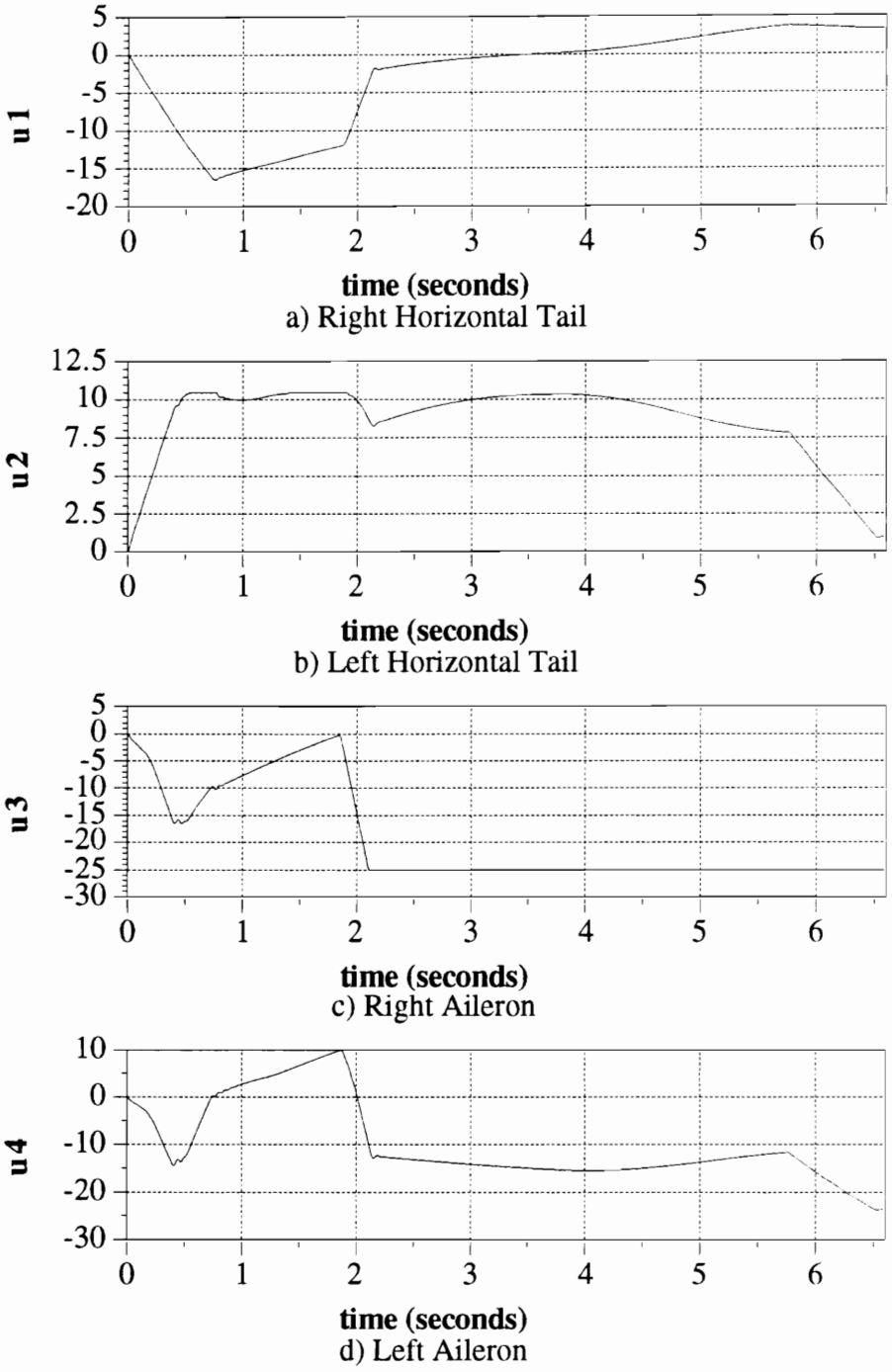
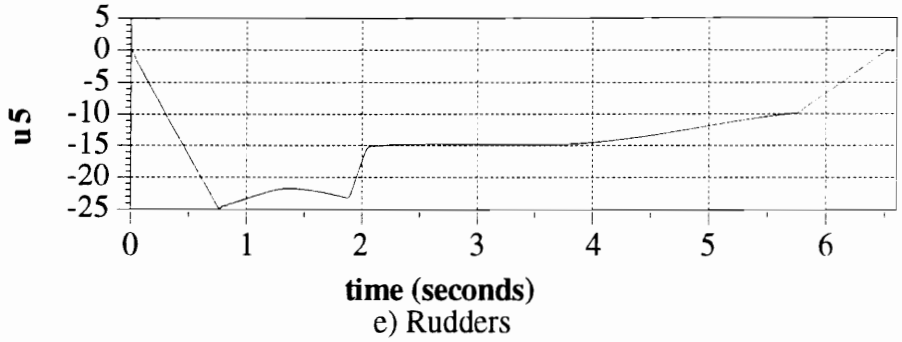
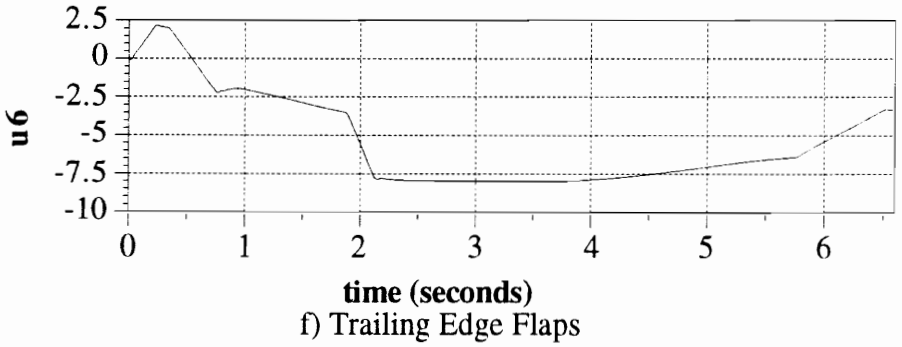


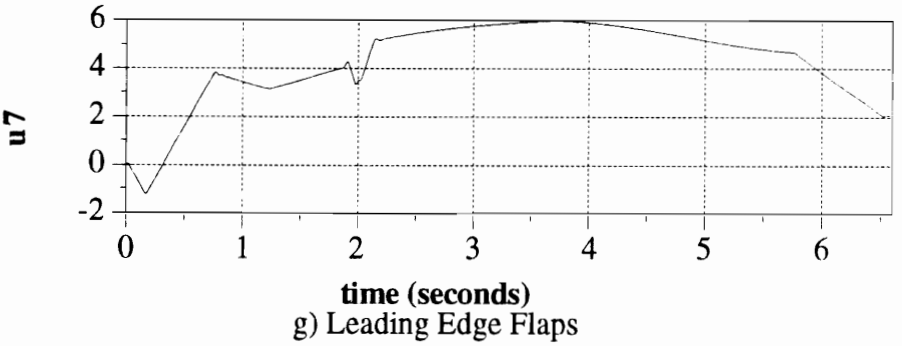
Figure 14-3: Controls from Example 14-2



e) Rudders



f) Trailing Edge Flaps



g) Leading Edge Flaps

Figure 14-3: Controls from Example 14-2

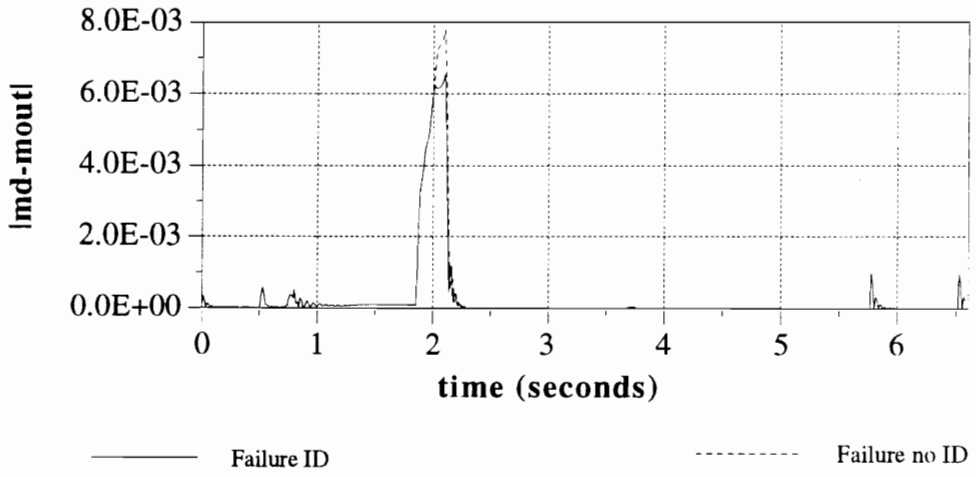


Figure 14-4: Error from Example 14-2

CONCLUSIONS

Many of the current methods for performing control allocation fail to generate admissible controls for a large portion of the set of attainable moments. The failure to generate admissible controls can result in a significant (and unnecessary) degradation in performance. Of the methods examined, only the Null-Space Intersection method and Direct Allocation are certain to find admissible controls for all the attainable moments.

While both of these methods can be computationally expensive, Direct Allocation offers a significant computational savings over the Null-Space Intersection method. The Null-Space Intersection method fails to provide a solution when the desired moment is unattainable. The failure to provide a solution when unattainable moments are commanded is an undesirable feature because some control laws may command moments which are unattainable. Direct Allocation finds the largest possible moment which has the same direction as an unattainable desired moment. Additionally, the Direct Allocation algorithm can compute the percentage of the desired moment which is attainable. This information may prove useful in the development of control laws which do not command unattainable moments.

If the effectiveness of the controls does not remain constant, it is not recommended that control positions be computed to generate the desired moments. Instead, compute the desired change in the control deflections which corresponds to the desired change in moment. Computing the controls in this manner can significantly reduce the errors introduced by changes in the control effectiveness. Reducing the magnitude of the desired change in moment by using a faster sample rate reduces the average error. An

example using real airplane data and a sample rate which is typical of modern computers shows that current technology can reduce the errors to an acceptable level.

By using Direct Allocation to command the change in control positions, the rate limits of the control effectors can be accounted for. As a result, Discrete Time Direct Allocation will command controls which do not violate any position or rate constraints for all the moments which can be generated.

It is possible to use the redundancy of the controls to optimize a function of the controls. The techniques described do not guarantee an optimal solution. Instead they provide a means of moving towards an optimal point. The ability to reach the optimal point depends upon the characteristics of the function to be minimized, the step sized used, and the moments being commanded.

If sensors provide information regarding current aircraft states and/or control positions, errors introduced by control failures are reduced by commanding changes moments. If the failed control can be identified, removing it from the control allocation process can further reduce the error.

The method of Discrete Time Direct Allocation provides a means of effectively allocating controls for all of the moments which lie within the physical constraints of the system. There are several areas where improvements to this method can be made, and it is suggested that these be considered as topics for future research:

- Methods which improve the computational speed of the Direct Allocation algorithm, possibly utilizing parallel processing

- Methods which eliminate the control chattering which occurs during Discrete Time Direct Allocation
- Methods to which provide the best way to deal with the path dependency which results from commanding changes in control deflections.
- Methods which provide the best step size to use when attempting to optimize a function of the controls

REFERENCES

[1] Raymond, E.T., and Chenoweth, C.C., *Aircraft Flight Control Actuation System Design*, SAE, Inc., Warrendale, PA, 1993, p.8

[2] *Private Pilot Maneuvers*, Jeppesen Sanderson, Inc. 1989 p2-16

[3] Cunningham, T.B., "Robust Reconfiguration for High Reliability and Survivability for Advanced Aircraft", *Restructurable Controls*, NASA Conference Publication 2277, ed. Montoya, R.J., Hampton, VA 1983, pp. 43-80

[4] Rynaski, E.G., "Experimental Experience at CALSPAN", *Restructurable Controls*, NASA Conference Publication 2277, ed. Montoya, R.J., Hampton, VA 1983, pp. 99-114

[5] Brogan, W.L., *Modern Control Theory*, 2nd ed., Prentice-Hall, Englewood Cliffs, NJ, 1982.

[6] Adams, R.J., Buffington, J.M., Sparks, A.G., and Banda, S.S., "An Introduction to Multivariable Flight Control System Design," Wright Laboratories, WL-TR-92-3110, October, 1992, pp. 43-44, pp. 112-116, pp. 156-162.

[7] Shaw, P.D. et al, "Design Methods for Integrated Control Systems," Air Force Wright Aeronautical Laboratories Report AFWAL-TR-88-2061, June, 1988, pp. 91-93.

[8] Vincent, James H., Emami-Naeini, Abbas, Khraishi, Nasser M., "Design Challenge in Automatic Flight Control: The SCT Solution," AIAA Paper 91-2633, AIAA Guidance and Control Conference, New Orleans, Louisiana, August, 1991.

[9] Snell, S.A., Enns, D.F., Garrard, W.L., "Nonlinear Inversion Flight Control for a Supermaneuverable Aircraft," AIAA Paper No. 90-3406, AIAA Guidance and Control Conference, Portland, Oregon, August, 1990, Proceedings pp. 808-817.

[10] Mattern, D., and Garg, S., "Integrated Flight/Propulsion Control System Design Based on a Decentralized, Hierarchical Approach", AIAA paper No. 89-3519, AIAA Guidance, Navigation and Control Conference, Boston, Ma., August ,1989.

[11] Lallman, F.J., "Relative Control Effectiveness Technique with Application to Airplane Control Coordination", NASA Technical Paper 2416, April 1985.

[12] Lallman, F.J., "Preliminary Design Study of a Lateral-Directional Control System Using Thrust Vectoring", NASA Technical Memorandum 86425, November 1985.

[13] Davidson, J.B., et al., "Development of a Control Law Design Process Utilizing Advanced Synthesis Methods With Application to the NASA F-18 HARV," High-Angle-of-Attack Projects and Technology Conf., NASA Dryden Flight Research Facility, April 21-23, 1992.

[14] Bugajski, D., Enns, D., and Hendrick, R., "Nonlinear Control Law Design for High Angle-of-Attack," High-Angle-of-Attack Projects and Technology Conference, NASA Dryden Flight Research Facility, Edwards, California, 21-23 April 1992, p.186

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

[16] Colbaugh, R., and Glass, K., "Cartesian Control of Redundant Robots", *Journal of Robotic Systems*, Vol. 6, No. 4, pp. 432-434

[17] Klein, C.A., and Huang, C.H., "Review of Pseudoinverse Control for Use with Kinematically Redundant Manipulators", *IEEE Trans. Sys., Man., and Cyber.*, SMC-13 245-250, 1983

[18] Gass, S.I., *Linear Programming, Methods and Applications,* 2nd Edition. McGraw-Hill, New York, 1964.

[19] Simonnard, M., *Linear Programming*. Prentice-Hall, Englewood Cliffs, New Jersey, 1966.

[20] Grogan, Robert L., "*On the Application of Neural Network Computing to the Constrained Control Allocation Problem*," Masters' Thesis, Virginia Polytechnic Institute & State University, 1994.

[21] Jones, R.D., and Bossi, J.A., "Development and Application of a Nonlinear Fin Mixer," The Boeing Company, Defense and Space Group, ACC/TA1, Seattle, WA., 1992, pp.1327-1331

[22] Gale, D., *The Theory of Linear Economic Models*, McGraw-Hill, New York, 1960, pp.66-68

[23] Moore, H.G., and Yaqub, A., *A First Course in Linear Algebra*, 2nd ed., HarperCollins Publishers, New York, NY, 1992, p.313, p.471

[24] Gill, P.E., Murray, W., and Wright, M.H., *Practical Optimization*, Academic Press, London, 1993, p.41

[25] Sommerville, D., *An Introduction to the Geometry of n Dimensions*, Dover Publishing, New York, 1958, pp.9-10

[26] Barber, B.C., Dobkin, D.P., and Huhdanpaa, H., "The Quickhull Algorithm for Convex Hull," Research Report GCG53, Geometry Center, U. of Minnesota, July 1993.

[27] Hausner, M.A., *A Vector Space Approach to Geometry*, Prentice-Hall, Englewood Cliffs, NJ, 1965, p.234

[28] Stevens, B.L., and Lewis, F.L., *Aircraft Control and Simulation*, Wiley, New York, 1992

[29] Bronson, R., *Theory and Problems of Matrix Operations*, McGraw-Hill Book Company, New York, 1989, p.193

[30] DeCarlo, R.A., *Linear Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1989, pp.297-301

[31] Durham, W., Lutze, F., and Mason, W., "Kinematics and Aerodynamics of the Velocity Vector Roll," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 6, 1994, pp. 1228-1233.

[32] F/A-18 Flight Control System Design Report, McDonnell Aircraft Company Report Number MDCA7813, Volume II "Flight Control System Analysis - Inner Loops", 15 June 1984, revised (revision A) 30 August 1985, Section 1

[33] Nguyen, Luat T., et al., "Simulator Study of Stall/Post-Stall Characteristics of a Fighter Airplane With Relaxed Longitudinal Static Stability", NASA Technical Paper 1538, December, 1979, pp. 37-40.

[34] Anon., "Military Specification-Flying Qualities of Piloted Vehicles," MIL-STD-1797A, March 1987.

[35] Anon., "Military Specification-Flying Qualities of Piloted Airplanes," MIL-F-8785C, Nov. 1980

[36] Franklin, G.F., Powell, J.D., and Workman, M.L., *Digital Control of Dynamic Systems*, Addison-Wesley Publishing Company, New York, 1994, pp.133-155

BIBLIOGRAPHY

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

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

Durham, W.C., "Attainable Moments of the Constrained Control Allocation Problem," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 6, November-December 1994, pp. 1371-1373

Bordignon, K.A., and Durham, W.C., "Closed-Form Solutions to Constrained Control Allocation Problem," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 5, September-October 1995, pp. 1000-1007

Durham, W.C., "Control Stick Logic in High Angle-of-Attack Maneuvering," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 5, September-October 1995, pp. 1092-1097

Durham, W.C., and Bordignon, K.A., "Multiple Control Effector Rate Limiting," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 1, January-February 1996, pp. 30-37

Bolling, J.G., Durham, W.C., and Bordignon, K.A., "Minimum Drag Control Allocation" to appear in *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 1, January-February, 1997

APPENDIX A
PROOFS

Proof 1: No Generalized Inverse Will Allocate $\mathbf{u} \in \Omega \ \forall \mathbf{m} \in \Phi$

Denote by Ψ the subset of $\partial(\Omega)$ which maps to $\partial(\Phi)$, and by $\mathbf{u}'$ the vectors in Ψ. The span(Ψ) = R^m unless there are columns of B which are all zeros. It would be impractical for a column of B to be all zeros because that would mean that a "control" had no effect.

If a generalized inverse allocates admissible controls for all the attainable moments, P must satisfy $\mathbf{u}' = PB\mathbf{u}' \ \forall \mathbf{u}'$, or $[PB - I_m]\mathbf{u}' = \mathbf{0} \ \forall \mathbf{u}'$. The constrained control vectors that map to $\partial(\Phi)$ must lie in $\aleph[PB - I_m]$, the null-space of $[PB - I_m]$:

$$\mathbf{u}' = \{ \mathbf{u} \in \Omega \cap \aleph[PB - I_m] \mid B\mathbf{u} \in \Phi \} \tag{A-1}$$

If the controls $\mathbf{u}'$ span R^m, they cannot all lie in $\aleph[PB-I_m]$, because the span($\aleph[PB-I_m]$) = R^n. ($n < m$) Therefore, there is no generalized inverse that yields solutions that are admissible everywhere on the boundary of Φ, and no single generalized inverse allocates admissible controls for all the attainable moments.

Proof 2: Convexity is Preserved Under Linear Transformations

If W is a convex set and T is a linear transformation, then $F = TW$ is a convex set. The transformation of a vector by a matrix multiplication is a linear transformation, and the following is true:[24]

$$A(\alpha\mathbf{x} + \beta\mathbf{y}) = A(\alpha\mathbf{x}) + A(\beta\mathbf{y}) = \alpha(A\mathbf{x}) + \beta(A\mathbf{y}) \tag{A-2}$$

W is a subset of the vector space R^m:

$$W \subset R^m \tag{A-3}$$

T is a matrix which maps vectors in R^m to some space R^n,

$$T: R^m \rightarrow R^n \tag{A-4}$$

F is the image of W in R^n,

$$F = TW,\ F \subset R^n \tag{A-5}$$

A is a vector in W to some point a, $\mathbf{A} \in W$, and **B** is a vector in W to some point b, $\mathbf{B} \in W$, such that $\mathbf{B} \neq \mathbf{A}$. See Figure A-1. A convex set is a set of points which contains, for any two points a,b in the set, the entire segment $\overline{ab}$.[27] **C** is any vector to a point c which lies on the line $\overline{ab}$. **C** can be expressed in terms of **A** and **B**:

$$\mathbf{C} = \mathbf{A} + \alpha(\mathbf{B} - \mathbf{A}) = (1 - \alpha)\mathbf{A} + \alpha\mathbf{B},\ 0 \leq \alpha \leq 1 \tag{A-6}$$

Because W is convex, $\mathbf{C} \in W$. **A'** and **B'** are the images of **A** and **B**:

$$\mathbf{A'} = T\mathbf{A},\ \mathbf{A'} \in F \tag{A-7}$$

$$\mathbf{B'} = T\mathbf{B},\ \mathbf{B'} \in F \tag{A-8}$$

D is a vector to any point d, which lies on $\overline{a'b'}$. **D** can be expressed in terms of the vectors **A'** and **B'**:

$$\mathbf{D} = \mathbf{A'} + \alpha(\mathbf{B'} - \mathbf{A'}) = (1 - \alpha)\mathbf{A'} + \alpha\mathbf{B'},\ 0 \leq \alpha \leq 1 \tag{A-9}$$

Substituting for **A'** and **B'** from Equations A-7 and A-8:

$$\mathbf{D} = (1- \alpha)T\mathbf{A} + \alpha T\mathbf{B} \tag{A-10}$$

Because T is a linear transformation, this can be written

$$\mathbf{D} = T((1- \alpha)\mathbf{A} + \alpha\mathbf{B}) = T\mathbf{C} \tag{A-11}$$

Equation A-11 shows that a point on $\overline{a'b'}$ is the image of a point on $\overline{ab}$.

All points in W map to points in F. If all points on $\overline{ab}$ are in W, and all points on $\overline{a'b'}$ are images of points in $\overline{ab}$, then all points on $\overline{a'b'}$ are in F:

$$\mathbf{D} \in F \tag{A-12}$$

If any two points a and b are in W, then they map to points a' and b' which are in F. If the entire line $\overline{ab}$ is contained in W, then the entire line $\overline{a'b'}$ is contained in F. Thus, if W is convex, under the linear mapping T, F is also convex.

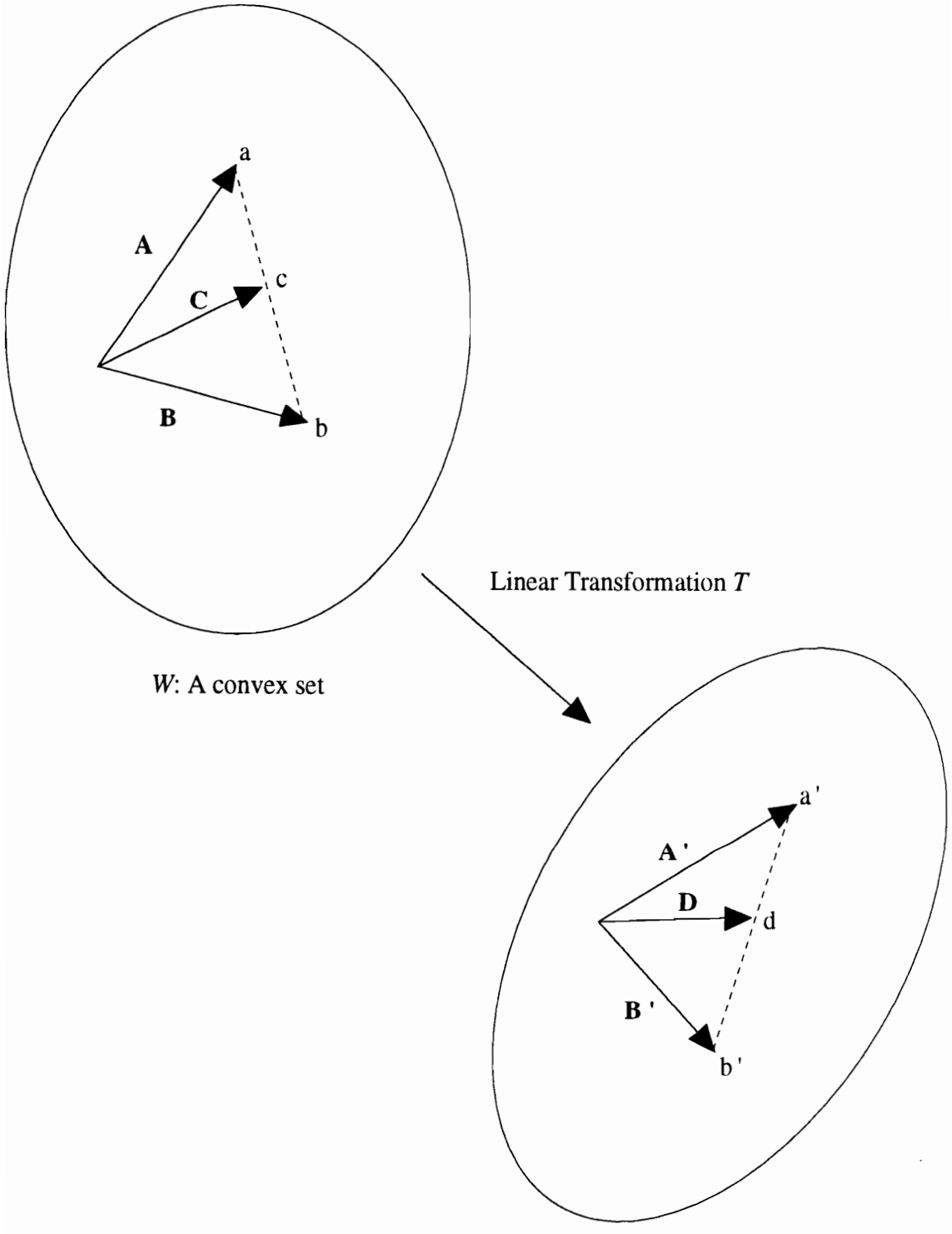


Figure A-1: Convexity Proof

Proof 3: Objects Remain Parallel Under Linear Transformations

If the vectors **A** and **B** are parallel, one can be expressed as a scalar multiple of the other:

$$a\mathbf{A} = \mathbf{B} \tag{A-13}$$

Under the linear transformation, T, they will remain parallel.

$$\mathbf{A}' = T\mathbf{A} \tag{A-14}$$

$$\mathbf{B}' = T\mathbf{B} = T(a\mathbf{A}) = aT\mathbf{A} = a\mathbf{A}' \tag{A-15}$$

Objects in R^m are defined by vectors in R^m. If parallel vectors in R^m map to parallel vectors in R^n, the objects defined by the parallel vectors in R^m will map to parallel objects in R^n.

Proof 4: Points on $\partial(\Phi)$ Map to Unique Points on $\partial(\Omega)$

To prove that if B contains no singular nxn submatrices, points on the boundary of Φ are images of unique points on the boundary of Ω, it will first be shown that points on $\partial(\Phi)$ must be images of points on $\partial(\Omega)$. Next, two solutions on $\partial(\Omega)$ which map to the same point on $\partial(\Phi)$ are examined. It will be shown that this can only occur if there is a nxn submatrix of B which is singular. An nxn submatrix of B is a matrix which contains any n of the columns of B.

Consider a moment on the boundary of Φ:

$$\mathbf{m}^*_1 \in \partial(\Phi) \tag{A-16}$$

Assume that there is a point in R^m, $\mathbf{u_1}$, which is admissible, but not on the boundary of Ω which maps to $\mathbf{m^*_1}$:

$$\mathbf{u_1} \in \Omega, \mathbf{u_1} \notin \partial(\Omega) \tag{A-17}$$

$$B\mathbf{u_1} = \mathbf{m^*_1} \in \partial(\Phi) \tag{A-18}$$

Because $\mathbf{u_1}$ is not on the boundary of Ω, there exists some $a > 1$ for which $a\mathbf{u_1}$ is on the boundary of Ω:

$$a\mathbf{u_1} \in \partial(\Omega) \text{ for some } a > 1 \tag{A-19}$$

However, because $\mathbf{m^*_1}$ is on the boundary of Φ and Φ is convex, there can be no moment greater than $\mathbf{m^*_1}$ which is in Φ (See Figure A-2):

$$a > 1 \Rightarrow B(a\mathbf{u_1}) = a\mathbf{m^*_1} \notin \Phi \tag{A-20}$$

By definition, all controls in Ω map to points in Φ.

$$a\mathbf{u_1} \in \Omega \Rightarrow B(a\mathbf{u_1}) \in \Phi \tag{A-21}$$

Equations A-20 and A-21 contradict each other. Thus, the assumption that a point not on the boundary of Ω can map to a point on the boundary of Φ is false, and all points on the boundary of Φ must be images of points on the boundary of Ω.

If there are two solutions in R^m which are not identical, they will have one or more of the components of $\mathbf{u}$ set to different values. In general, there will be between 1 and m controls which have different values. This general problem is divided into two categories. The first involves cases where the number of controls which are have different values is fewer than n. The second involves cases where n or more controls

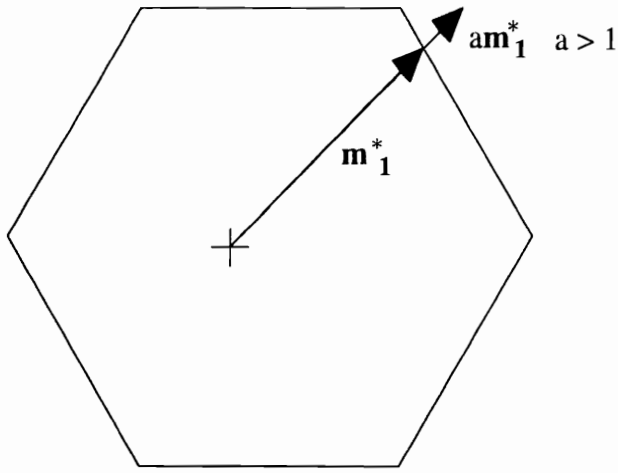


Figure A-2: Moment on Boundary of Φ

have different values. It will be shown for each category that if two points in $\partial(\Omega)$ map to the same point in $\partial(\Phi)$, then there must be a singular nxn submatrix of B.

Category 1: Two points in R^m on $\partial(\Omega)$ map to the same point in R^n on $\partial(\Phi)$ and the two points in R^m have n-1 or fewer controls which are set to different values.

Consider two points, $\mathbf{u}^*_1$ and $\mathbf{u}^*_2$, which map to $\mathbf{m}^*_1$. By definition,

$$\mathbf{u}^*_1 \in \partial(\Omega) \tag{A-22}$$

$$\mathbf{u}^*_2 \in \partial(\Omega),\ \mathbf{u}^*_2 \neq \mathbf{u}^*_1 \tag{A-23}$$

Therefore,

$$B\mathbf{u}^*_1 = \mathbf{m}^*_1 \tag{A-24}$$

$$B\mathbf{u}^*_2 = \mathbf{m}^*_1 \tag{A-25}$$

Subtracting Equation A-25 from A-24, yields:

$$B\mathbf{u}^*_1 - B\mathbf{u}^*_2 = B(\mathbf{u}^*_1 - \mathbf{u}^*_2) = \mathbf{0} \tag{A-26}$$

So that,

$$\mathbf{u}^*_1 \neq \mathbf{u}^*_2 \Rightarrow (\mathbf{u}^*_1 - \mathbf{u}^*_2) \neq \mathbf{0} \tag{A-27}$$

Because there are n-1 or fewer controls which have different values in the vectors $\mathbf{u}^*_1$ and $\mathbf{u}^*_2$, there will be n-1 or fewer non-zero elements in $(\mathbf{u}^*_1 - \mathbf{u}^*_2)$. Equation A-26 can be re-written, removing the columns of B which correspond to the elements of $(\mathbf{u}^*_1 - \mathbf{u}^*_2)$ which are zero.

Define B' and $\mathbf{u}_b$:

$B' \equiv$ a submatrix of B. The columns of B' are the columns of B which correspond to the non-zero elements of $(\mathbf{u}^*_1 - \mathbf{u}^*_2)$ (A-28)

$\mathbf{u}_b \equiv$ a vector containing the non-zero elements of $(\mathbf{u}^*_1 - \mathbf{u}^*_2)$ (A-29)

Then,

$B' \in R^{nxy}, \mathbf{u}_b \in R^y, y \leq (n\text{-}1)$ (A-30)

$B'\mathbf{u}_b = \mathbf{0}$ (A-31)

Equation A-31 shows that a set of $(n\text{-}1)$ or fewer columns of B are linearly dependent. If columns of an nxn matrix are linearly dependent, the matrix is singular. The nxn submatrices of B containing the linearly dependent columns in B' will be singular.

Category 2: Two points in R^m on $\partial(\Omega)$ map to the same point in R^n on $\partial(\Phi)$ and the two points in R^m have n or more controls which are set to different values.

1) If 2 points in $\partial(\Omega)$, $\mathbf{u}^*_1$ and $\mathbf{u}^*_2$, map to the same point in $\partial(\Phi)$, $\mathbf{m}^*_1$, and have n controls set to different values, there is some point in R^m which maps to $\mathbf{m}^*_1$ which has those n controls set to values within their constraints, $u_{i\,Min} < u_i < u_{i\,Max}$.

If $\mathbf{u}^*_1$ and $\mathbf{u}^*_2$ map to a point in R^n, then all points on a line between them also map to the same point in R^n. This can be seen from Equation A-6, which shows that any point on a line between two vectors can be expressed as a linear combination of those vectors. Consider a point, $\mathbf{u}^*_3$, which lies on a line between $\mathbf{u}^*_1$ and $\mathbf{u}^*_2$.

$$\mathbf{u}^*_3 = (1-\alpha)\mathbf{u}^*_1 + \alpha\mathbf{u}^*_2,\ 0 \le \alpha \le 1 \tag{A-32}$$

If $\mathbf{u}^*_1$ and $\mathbf{u}^*_2$ map to $\mathbf{m}^*_1$, so does $\mathbf{u}^*_3$,

$$B\mathbf{u}^*_3 = (1-\alpha)B\mathbf{u}^*_1 + \alpha B\mathbf{u}^*_2 = (1-\alpha)\mathbf{m}^*_1 + \alpha\mathbf{m}^*_1 = \mathbf{m}^*_1 \tag{A-33}$$

Consider Figure A-3 to be a box on the boundary of Ω. At $\mathbf{u}^*_1$ the three controls which define the box (u_i, u_j, u_k) are at their positive limits. At $\mathbf{u}^*_2$ these controls are at different admissible values. The line connecting them contains points which have all three controls at non-limited values.

2) Controls which are not at limiting values may be moved in either direction to generate moments.

3) The columns of B define directions in R^n.

A column of B, B_i, is a vector in R^n. Changing the value of the control, u_i, will move the moment produced in the direction defined by B_i. See Figure A-4.

4) From a point on $\partial(\Phi)$, the value of a control can be changed, changing the moment produced in a direction along the boundary or towards the interior.

If a control is saturated, it can move in only one direction. Changing a saturated control may move the moment produced to a point interior to the boundary without being able to move the moment produced to a point exterior to the boundary. See Figure A-5.

5) If a control is not saturated, it can move the moment produced only along the boundary.

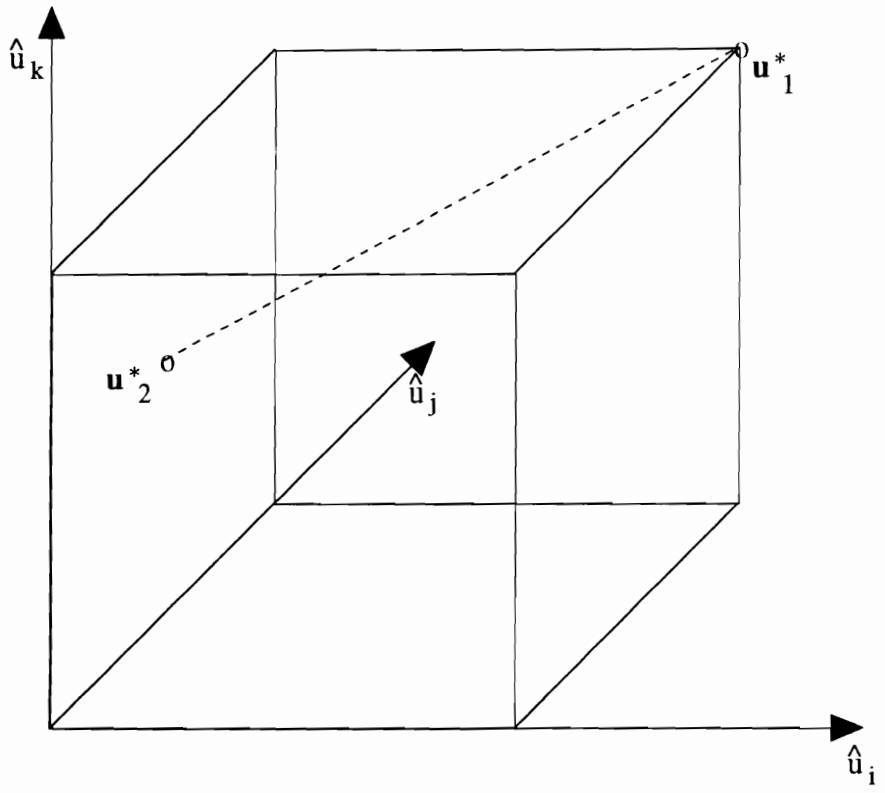


Figure A-3: A 3-D Object on $\partial(\Omega)$

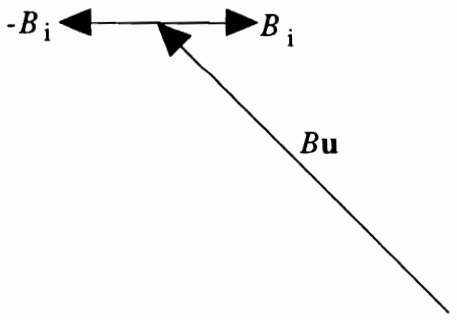


Figure A-4: Columns of B Are Directions in R^n

If changing an unsaturated control could move the moment produced interior to the boundary, it could also move the moment produced exterior to the boundary, which is not possible. See Figure A-6.

6) The n-D polytope Φ is bounded by (n-1)-D polytopes.

From a point on the boundary, there are at most (n-1) independent directions to move which are along the boundary. If there are n or greater non-saturated controls at some point on the boundary, then there are n or greater directions defined by these controls. However, these n or greater directions span a space of dimension n-1 or less. See Figure A-7.

If n columns of B span an n-1 dimensional space, the rank of the nxn matrix containing those columns will be n-1. Thus, an nxn submatrix of B will be singular.

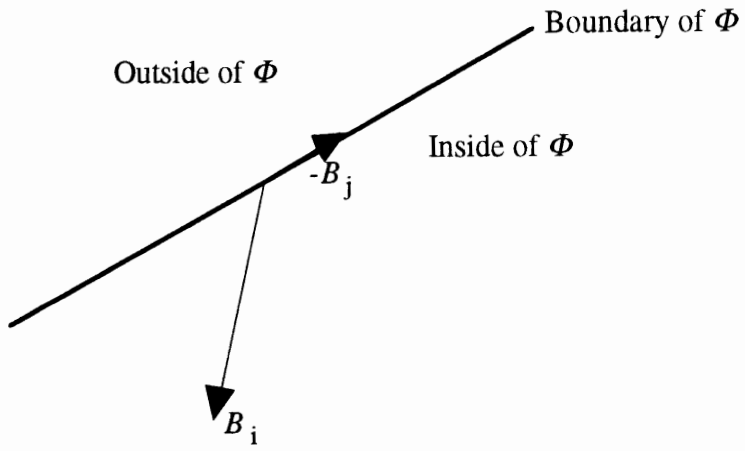


Figure A-5: Saturated Controls Can Move Either Along or Interior to $\partial(\Phi)$

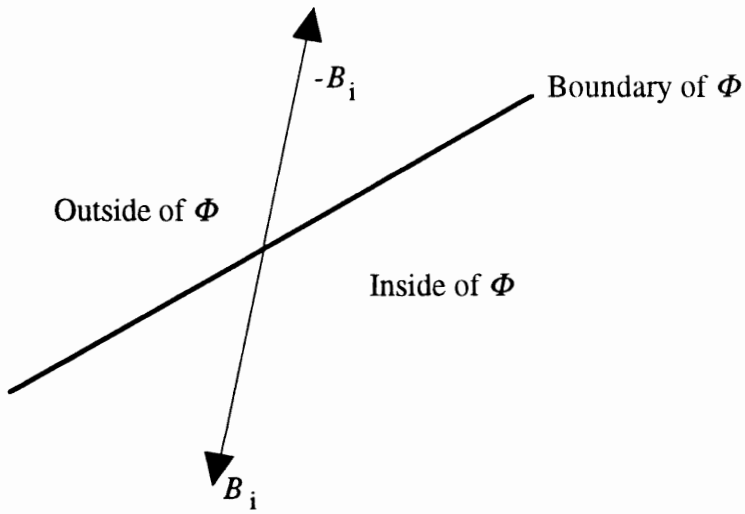


Figure A-6: Unsaturated Controls Can't Move Interior to $\partial(\Phi)$

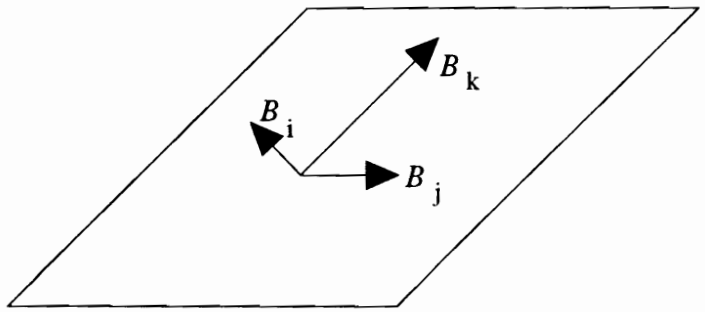


Figure A-7: 3 Columns of B Contained in a 2-D Bounding Object of Φ

Vita

Ken Bordignon was born in the spring of 1970, in a sprawling settlement on the shores of Lake Michigan. In time, he grew skilled in the ways of math and science. In 1988, he traveled across the Great Midwestern Plains to the land of the Golden Dome where he studied at the feet of learned men. In 1992, he withdrew to the mountains of southwestern Virginia to conspire with sages and visionaries to compose this tome.