JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 18, No. 5, September–October 1995

Closed-Form Solutions to Constrained Control Allocation Problem

Kenneth A. Bordignon* and Wayne C. Durham†
Virginia Polytechnic Institute and State University, Blacksburg, Virginia 24061-0203

This paper describes the results of recent research into the problem of allocating several flight control effectors to generate moments acting on a flight vehicle. The results focus on the use of various generalized inverse solutions and a hybrid solution utilizing daisy chaining. In this analysis, the number of controls is greater than the number of moments being controlled, and the ranges of the controls are constrained to certain limits. The control effectors are assumed to be individually linear in their effects throughout their ranges of motion and independent of one another in their effects. A standard of comparison is developed based on the volume of moments or moment coefficients a given method can yield using admissible control deflections. Details of the calculation of the various volumes are presented. Results are presented for a sample problem involving 10 flight control effectors. The effectivenesses of the various allocation schemes are contrasted during an aggressive roll about the velocity vector at low dynamic pressure. The performance of three specially derived generalized inverses, a daisy-chaining solution, and direct control allocation are compared.

Nomenclature

B = matrix of control effectiveness
m = number of controls
$\boldsymbol{m}$ = vector of moments
n = dimension of generalized moment space
$\boldsymbol{u}$ = vector of controls
$\partial(\Sigma)$ = boundary of set Σ
Π = subset of moments attainable using some allocation scheme
Φ = subset of attainable moments
Ω = subset of constrained controls

Superscript

$*$ = lies on $\partial\cdot$

Introduction

THE control allocation problem is to find a set of admissible control effector deflections that will generate a specified set of forces/moments. Many physical systems are designed with the idea of a single controller for each controlled degree of freedom. For example, in the case of airplanes, elevators usually control pitching moments, rudders control yawing moments, and ailerons control rolling moments. With the same number of independent controls and forces/moments to be generated, solutions to the control allocation problem are unique.

Additional control effectors may be added to the physical system for purposes of redundancy or to extend the operational envelope of the system. Many modern aircraft have more than three independent moment generators. These additional moment generators may arise from the freeing of opposing control surfaces to operate independently of one another. These controls are all constrained to certain limits, determined by the physical geometry of the control actuators, or by other considerations. If the control constraints are not considered, then all control deflections are admissible and the mathematical solutions to the allocation of these controls are, in general, infinite in number.

Received Aug. 19, 1994; revision received Jan. 24, 1995; accepted for publication Feb. 1, 1995.

*Graduate Student, Department of Aerospace and Ocean Engineering. Student Member AIAA.

†Assistant Professor, Department of Aerospace and Ocean Engineering. Senior Member AIAA.

Many closed-form solutions to the control allocation problem involve inverting nonsquare matrices using generalized inverses.[1–4] Mathematically, the moments produced by constrained controls may be expressed as a mapping of the subset of constrained controls into the (lower dimension) moment space. This results in a closed subset of attainable moments (or AMS). It may be shown that no generalized inverse can yield admissible controls for every moment in the AMS.[5] There are various methods of weighting, or tailoring, generalized inverses that may improve their capability to provide admissible controls.

Other than the use of generalized inverses, there are only two other methods commonly used to solve the allocation problem. The first is widely used in robotics and involves augmenting the control effectiveness matrix[6] to make it invertible. This method effectively exploits solutions that lie in the null space of B, which is the subject of ongoing research by the authors. The second method is called daisy chaining. In daisy chaining, one breaks the problem up into two or more smaller problems by using the control effectors in groups. It may be shown that daisy chaining, like generalized inverses, cannot yield admissible controls for every moment in the AMS.[5] The use of this method will be treated by example.

In previous works, we have described a direct method of solving the allocation problem.[5,7] The direct method yields admissible controls for every moment in the AMS. There are infinite combinations of admissible controls that generate moments in the interior of the AMS, but on the boundary the combinations are unique. The direct method calculates admissible controls in the interior of the AMS as scaled-down versions of the unique solutions on the boundary and provides continuous solutions for continuously varying moment demands. The direct method is therefore used as a benchmark for comparison of the other methods.

In this paper, we offer as a standard of comparison for the various solutions to the control allocation problem the idea of the total volume of moment-generating capability of an allocation scheme. The procedure for calculating these volumes for several schemes is presented here for the first time. There are many other concerns, such as control energy and control rate demands, that must be addressed before any comparison can be complete. However, the volume of moment-generating capability is of major concern since a deficiency in this criterion translates directly to a loss in potential maneuverability or to the extra weight of controls that must be borne to achieve some desired level of maneuverability. We succinctly state the control allocation problem and then describe the methods of solving the problem subsequently used. We then describe the means used to calculate the moment-generating volumes of each of the methods. Sample results are shown for a representative airplane control

allocation problem. The sample results are then applied to a tactically important maneuver known as the velocity vector roll. The allocation methods are contrasted by determining their ability to perform this maneuver, as measured by the highest steady-state roll rate they can attain using admissible controls. This example clearly illustrates the loss in potential maneuverability that can result from the choice of allocation scheme.

Problem Statement

We consider an m-dimensional control space $\boldsymbol{u} \in R^m$. The controls are constrained to minimum and maximum values, defined by the subset Ω:

$$\Omega = \{\boldsymbol{u} \in R^m \mid u_{i,\min} \leq u_i \leq u_{i,\max}\} \subset R^m \tag{1}$$

The subset of controls that lie on the boundary of Ω, $\partial(\Omega)$, are denoted by $\boldsymbol{u}^*$:

$$\boldsymbol{u}^* = \{\boldsymbol{u} \mid \boldsymbol{u} \in \partial(\Omega)\} \tag{2}$$

These controls generate moments through a mapping B onto n-dimensional moment space through a linear matrix multiplication of $\boldsymbol{u}$, $B\boldsymbol{u} = \boldsymbol{m}$, where $B: R^m \rightarrow R^n$, where $m > n$. Here, B is the control effectiveness matrix with respect to the moments. In order to have unique solutions on the boundaries, we require that every $n \times n$ partition of B be nonsingular.

Denote by Φ the image of Ω in R^n, $\Phi \subset R^n$. The subset Φ therefore represents all the moments that are attainable within the constraints of the controls. Moments that lie on the boundary of Φ, $\partial(\Phi)$, are denoted by an asterisk:

$$\boldsymbol{m}^* = \{\boldsymbol{m} \mid \boldsymbol{m} \in \partial(\Phi)\} \tag{3}$$

A unit vector in the direction of $\boldsymbol{m}$ is denoted by $\hat{\boldsymbol{m}}$.

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

Methods

The methods used to generate the results presented below, except those involving daisy chaining, are based entirely on algorithms developed in Refs. 5, 7, and 8. The algorithms were implemented in a stand-alone program called CAT (for Control Allocation Toolbox). The program permits user input of the control effectiveness matrix and control limits. The user may then calculate and display the AMS. The program accepts inputs of time histories of desired moments, allocates the controls, and plots the desired moments within the AMS. Additionally, the user may calculate various generalized inverse solutions to the problem, and the subsets of moments achieved using these solutions may be displayed as a figure within the AMS.

In the subsequent discussion, we assume the control configuration used in our example: 10 independent controls and 3 orthogonal moments.

Attainable Moment Subset

The controls in our example lie in a 10-dimensional space. With constraints imposed, the subset of constrained controls is a 10-dimensional hyperbox whose dimensions are the ranges of motion for each control. The AMS is the image of this hyperbox in three-dimensional moment space, as determined by the mapping B. The boundary of the AMS is made up of facets that are images of facets of the hyperbox. A face is a geometric feature of the constrained control and AMS. Faces are generated in m space by placing all but two of the m controls at either of their two limits and allowing two to vary, yielding $2^{m-2} \cdot m!/[2! \cdot (m-2)!]$ faces in the subset of constrained controls. Faces in the AMS are the images of the faces in the constrained control subset. A facet is any fact that lies on the boundary of the subsets Φ or Ω. All faces in Ω are facets, but a face in Φ may or may not be a facet. In Φ, there are two facets (front and back) for each of combinations of controls taken two at a time, yielding $2 \cdot m!/[2! \cdot (m-2)!]$ or $m(m-1)$ facets. In our example, the hyperbox has $2^8 \cdot 10!/(2! \cdot 8!) = 11{,}520$ facets, whereas the AMS has only $2 \cdot 10!/(2! \cdot 8!) = 90$ facets. The algorithm used by CAT, based on Ref. 8, identifies the facets of the AMS in 45 steps, each involving a 2×2 matrix inversion.

For each facet thus determined, a solution for the intersection of a line through the origin with the facet is computed and stored. This solution is subsequently used to allocate the controls.

Generalized Inverse Solutions

Generalized inverse solutions to the problem are defined as all matrices $P: R^n \rightarrow R^m$ such that $BP = I_n$. The generalized inverse solution is then $\boldsymbol{u} = P\boldsymbol{m}$. Quite often, the minimum-norm P (sometimes employing weighting based on the limits of the controllers) is used.[1,5] The minimum-norm solution is often called the Penrose (or Penrose–Moore) pseudoinverse, or simply the pseudoinverse. The pseudoinverse is calculated according to the well-known formula[9] $P = B^T[BB^T]^{-1}$.

When we speak of tailoring a generalized inverse solution, we generally will mean that we seek a generalized inverse such that, in certain specified directions in moment space, the generalized inverse will yield solutions that coincide with the AMS. For example, we may wish the generalized inverse that yields the maximum amount of rolling moment (within the control constraints) with zero pitching moment and zero yawing moment. Geometrically, we are seeking a 3-dimensional subspace of the 10-dimensional control space whose intersection with the subset of constrained controls maps to the required point(s) in moment space. The problem is harder to state than it is to solve; Ref. 5 outlines the method used. Because we are specifying a three-dimensional subspace, there are at most three directions free to be specified.

Another method for tailoring a generalized inverse is to introduce a weighting matrix N when calculating the pseudoinverse. This is sometimes referred to as a weighted pseudoinverse, and it is calculated from the equation $P = N(BN)^T[BN(BN)^T]^{-1}$.

The values in the matrix N are usually used to emphasize/deemphasize certain controls. Examples of weighted generalized inverses can be found in Refs. 2–4.

When we speak of the best generalized inverse, we mean the generalized inverse such that the volume of moments for which it yields admissible controls, Π, is greater than or equal to that of any other generalized inverse. In CAT, the best generalized inverse is found by systematically varying the elements of P, subject to $BP = I_n$, until the greatest volume is determined. In CAT, we use a simplex algorithm[10] to minimize the negative of the volume thus calculated. The calculations of the volumes are quite complicated, and there is no guarantee that the solutions found are global minima. Surprisingly, however, close examination of some of the solutions have revealed that the solutions obtained were unique and global.

Daisy Chaining

The method of daisy chaining to solve the control allocation problem is described in Ref. 11 and discussed in Ref. 5. Reference 11 is used to illustrate the method. Two groups of controls are selected: there aerodynamic controls ($\boldsymbol{u}_1$) and three thrust-vectoring controls ($\boldsymbol{u}_2$). With three moments to be generated, the control effectiveness matrix is partitioned into two 3×3 matrices B_1 and B_2:

$$B\boldsymbol{u} = [B_1 \quad B_2]\begin{Bmatrix}\boldsymbol{u}_1\\ \boldsymbol{u}_2\end{Bmatrix} = B_1\boldsymbol{u}_1 + B_2\boldsymbol{u}_2 \tag{4}$$

For a given moment to be generated, the aerodynamic controls are first used to the point where one or more controls are saturated. Then thrust-vectoring controls are brought to bear. That is, when none of the aerodynamic controls are saturated,

$$\boldsymbol{u}_1 = B_1^{-1}\boldsymbol{m}_d \qquad \boldsymbol{u}_2 = \boldsymbol{0} \tag{5}$$

When any of the aerodynamic controls are saturated,

$$\boldsymbol{u}_1 = \boldsymbol{u}_{1(\text{sat})} \qquad \boldsymbol{u}_2 = B_2^{-1}\{\boldsymbol{m}_d - B_1\boldsymbol{u}_{1(\text{sat})}\} \tag{6}$$

Since in this application B_1 and B_2 are square and assumed invertible, their inverses are unique, and the solutions obtained are unique for the particular control groupings chosen. In the example

that follows, these matrices are not square, so generalized inverses are used. Examples of using daisy chaining when B_1 and B_2 are not square matrices can be found in Ref. 2.

Volume Calculations

Calculating n Volume

The subset of moments attainable by an allocation scheme can usually be represented as a convex n-dimensional object bounded by $(n-1)$-dimensional objects. To calculate the volume of such an object, the $(n-1)$-dimensional bounding objects are treated as the bases of n-dimensional pyramids with their apexes at the origin of moment space. The sum of the volumes of all these pyramids is the volume of the n-dimensional object. We therefore present a generic method for calculating the volumes of pyramids with arbitrary polygons for bases that may be applied to both the entire AMS and to that portion attained by other methods.

The n volume of a parallelepiped is the $n-1$ volume of one of its faces times the altitude on that face.[12] If σ is an n-dimensional object in R^n defined by the points $A_0, \ldots, A_n$, then its volume can be computed by the following equation[12]:

$$V(\sigma) = (1/n!)[A_0A_1 \cdots A_0A_n]$$

In the nomenclature of Ref. 12, the quantity is 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 properties of the outer product, this equation can also be written as

$$V(\sigma) = (1/n!)\det([A])$$

Here, $[A]$ is a matrix whose rows are the vectors $A_0A_1 \cdots A_0A_n$. The term $V(\sigma)$ is an oriented volume. By this we mean that if σ changes its orientation, then $V(\sigma)$ changes its sign.[12] Since we are interested in the magnitude and not the orientation of these volumes, we take the absolute value of the above equations. To calculate the volume of Π, an n-dimensional volume, divide its bounding objects into $(n-1)$-dimensional objects containing n points (i.e., divide the polygons into triangles). To do this, the connectivity of the vertices of the boundary must be 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. If there are k such bounding objects, then the n volume of Π can be calculated as

$$V(\Pi) = \frac{1}{n!}\sum_{i=1}^{k} |\det([A_i])|$$

Direct Allocation

Direct allocation, by design, yields admissible solutions in the entire AMS. The volume of the AMS is determined by finding the four vertices of each of the facets bounding the AMS. These facets are parallelograms that form the bases of the pyramids used in the volume calculations. Methods for determining these vertices and their connectivity have been previously described.[7,8]

Generalized Inverses

The calculations involved in determining the generalized inverse volumes are similar to the calculations involved here, except the bases of the pyramids are not generally parallelograms. Given the algorithm for determining the volumes of pyramids with arbitrary polygons for bases, we need only determine the points defining the polygons associated with generalized inverses. The geometry of generalized inverse solutions to this problem may be thought of as follows: First consider some three-dimensional subspace of the m-dimensional control space. This three-dimensional subspace is determined by some mapping $P: R^3 \rightarrow R^m$, the generalized inverse. Now calculate the intersection of this subspace with the subset of constrained controls. The mapping B onto three-dimensional moment space of this intersection then represents a subset of the AMS for which $\boldsymbol{u} = P\boldsymbol{m}$ yields $\boldsymbol{u} \in \Omega$, i.e., for which the controls thus calculated are admissible. Calculating the points of intersection is not too hard, but the ordering of these points into polygonal bases requires some work. To visualize these ideas, we first address a low-order illustration and use it as an analog in the higher order problems of interest.

Low-Order Illustration

We will examine a case of three controls and two moments. The subset of admissible controls, Ω, is an m-dimensional box with all right-angle intersections of edges. Figure 1a shows a three-dimensional representation. The columns of the generalized inverse

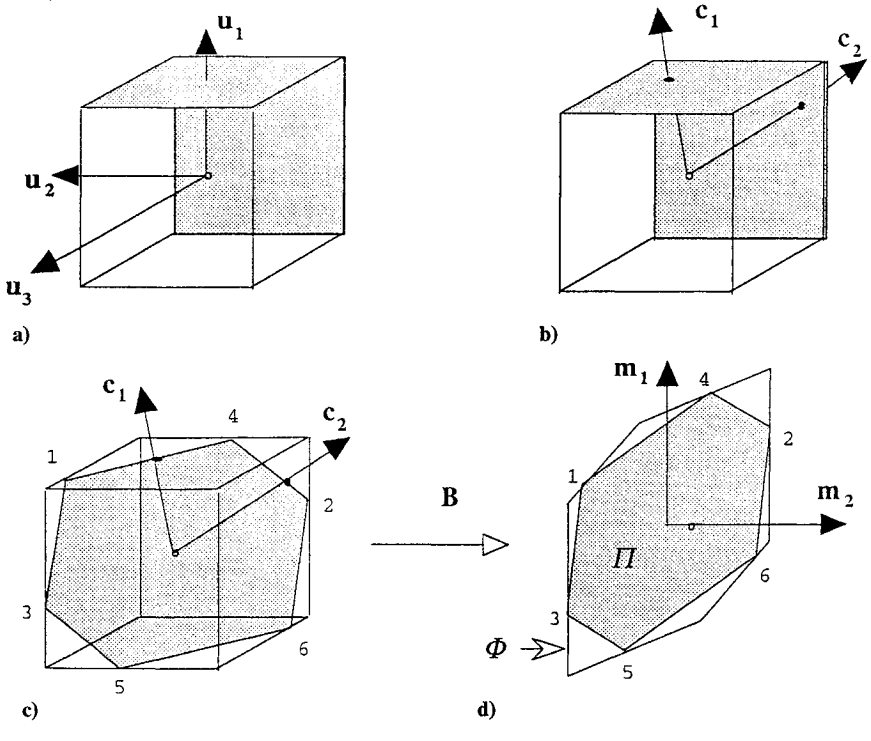

Fig. 1 Low-order illustration: a) Ω, admissible controls in R^m, b) columns of P, c) intersection of P_S with Ω, and d) mapping to moment space yielding Φ and Π.

matrix P are each in R^3; two such column vectors (two moments) are shown in Fig. 1b. We have assumed B is full rank, so P is full rank as well. Therefore the columns of P form a basis for a two-dimensional subspace in R^3. This subspace will be denoted P_S. All points mapped from R^2 to R^3 using P will lie in P_S. Both P_S and Ω contain the origin. Thus, they are guaranteed to intersect. Figure 1c shows the intersection of P_S and Ω. This intersection represents every combination of admissible controls achievable in the column space of P. Figure 1d shows the generation of the AMS by the mapping B. The intersection of P_S and Ω is mapped as well, and its image is the subset of moments attainable by the generalized inverse within the constraints on the controls. We denote this subset by the symbol Π. Keep in mind that Figs. 1a–1c represent a three-dimensional space whereas Fig. 1d represents a two-dimensional space.

In general, the mechanics of the solution go as follows. We first calculate the intersection of each of the edges of Ω with the plane P_S. We actually calculate the intersections of infinite lines along each edge with the plane P_S and then examine the solution to see if it lies on the segment that defines the edge. In Fig. 1c, there is in general an intersection for each of the 12 lines along the edges of Ω, but only 6 intersections lie strictly on the edges. Second, we determine an ordering of the vertices that generate the convex figure shown. The third step is to map each of the intersections thus determined into moment space, determining the vertices of the polygon Π. Finally, given the ordered set of vertices, we calculate the area (volumes of polytopes, for three-moment problems) of Π for purposes of comparison.

Higher Order Problems

We now address the problem of higher order control and moment spaces, where the dimension of the moment space is less than that of the control space. Certain other definitions are needed to clarify the geometry of the problem. A k-dimensional object will refer to a compact subset of a k-dimensional linear variety. A linear variety is a subspace translated from the origin. For example, a line through two arbitrary points in 3-space defines a one-dimensional linear variety. A compact set of points along the line (such as one of the edges in Fig. 1a) is a one-dimensional object. The facets of the box in Fig. 1a are two-dimensional objects in translated two-dimensional subspaces of R^3 (linear varieties).

We note that k-dimensional objects are bounded by $(k-1)$-dimensional objects (i.e., a line is bounded by points, a polygon is bounded by lines, etc.). Our admissible controls are no longer in a three-dimensional object bounded by two-dimensional objects. We now have an m-dimensional object bounded by $(m-1)$-dimensional objects.

In control space, k-dimensional objects have physical interpretations related to which controls are saturated and which are free to vary. A zero-dimensional object is a vertex, at which every control is at a minimum or maximum deflection. A one-dimensional object is an edge, on which exactly one control is free to vary. On a k-dimensional object $m-k$ controls are at a minimum or maximum deflection and k are free to vary. Thus, the configuration of controls associated with some point of intersection uniquely determines the lowest order object on which the point lies. This is important for determining the connectivity of the intersections.

Determining Points of Intersection

The way that Ω is defined, it is a convex polytope in R^m. A linear subspace, such as P_S, is convex. The intersection of P_S and Ω will be a convex polytope.[13] To define this polytope, it is sufficient to find all of the extreme points, or the vertices of the polytope.[13] 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. In Fig. 1c, the points in question are the six vertices of the shaded polygon.

Given that n and m are integers with $m > n$, to find the extreme points of the intersection of an n-dimensional space with an m-dimensional object (Ω), one need only search for points of intersection on the $(m-n)$-dimensional or smaller objects which are on the boundary of the m-dimensional object. For instance, a two-dimensional plane may intersect a one-dimensional edge of a three-dimensional cube at a point ($m-n=1$), but it cannot intersect a two-dimensional face of the three-dimensional cube at a point. Note, however, that a two-dimensional plane may intersect a two-dimensional object of a four-dimensional (or higher) Ω at a single point. In dealing with higher order problems, intuition is unreliable.

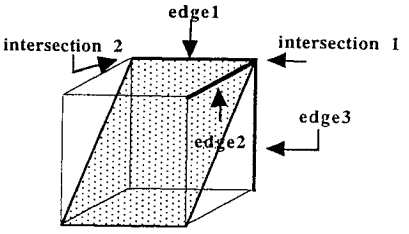

Fig. 2 Example of a singular P_1.

We therefore generate every $(m-n)$-dimensional object and find its intersection with P_S. Following our discussion above, on an $(m-n)$-dimensional object n controls are at a minimum or maximum deflection and $m-n$ are free to vary. We generate the $(m-n)$-dimensional objects 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 selections and 2^n combinations of minimum/maximum deflection for each. We then determine the values of the remaining (free) $m-n$ controls. If the value of each of the $m-n$ controls thus determined lies within its constraints, then the intersection is within Ω; otherwise it is not. In Fig. 1c, 6 of the 12 possible intersections were within Ω and 6 (not shown) were without.

The determination of the intersections is done as follows:

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

$$\begin{bmatrix} P_1 \\ P_2 \end{bmatrix} \boldsymbol{m} = \begin{Bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \end{Bmatrix} \Rightarrow P_1\boldsymbol{m} = \boldsymbol{u}_1 \qquad P_2\boldsymbol{m} = \boldsymbol{u}_2$$

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

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

Here, P_1 should be square and invertible. If P_1 is singular, then there are an infinite number of solutions on a particular k-dimensional object. This can occur if the n-dimensional subspace contains some lower dimension object. Figure 2 shows an example of this occurrence. The P_1 corresponding to a search along edge 1 will be singular. This P_1 can be ignored because the defining intersections (1 and 2) will be found by searching other edges. However, note that such intersections may be found by searching several edges (i.e., intersection 1 may be found by searching edge 2 or edge 3 in Fig. 2).

4) Solve for $\boldsymbol{u}_2$: $\boldsymbol{u}_2 = P_2\boldsymbol{m}$.

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 $\boldsymbol{u}_2$ are at or within their limits, then that point is also on $\partial(\Omega)$ and is a defining intersection.

Mapping Points to Moment Space

Each of the intersection points determined above is multiplied by B, yielding the corresponding points in moment space.

Determining Connectivity of Points

Under the linear mapping B, the convex polytope in R^m will remain a convex polytope in R^n. Furthermore, the two polytopes will be geometrically similar: points that are connected by lines in R^m are connected by lines in R^n. Assume in Figs. 1c–1d that the vertices were identified in the order listed. It is easy to see that the proper sequence should be 1–4–2–6–5–3–1, but for higher order problems this sequence is more difficult to determine. The method of determining the sequence is based on the vertices' relationships in control space. For example, the first point found (1) is on an edge adjacent to the edge containing point 4, which is adjacent to the

edge containing point 2, which is adjacent to the edge containing point 6, and so on back to point 1.

In higher dimensional problems we will perform a similar analysis by examining the intersections of P_S with the bounding $(m-1)$-dimensional objects of the m-dimensional subset Ω. We examine the geometric objects on which each of the points lie and then connect the points that lie on objects adjacent to each other.

A k_1-dimensional object is adjacent to a k_2-dimensional object if they share a common k_3-dimensional object, $k_3 \leq \min(k_1, k_2)$. The usual case of concern to us here is $k_1 = k_2 = m-3$. If $k_3 = m-3$, then two objects are identical. We are therefore interested in $k_3 = m-j$, $j = 4, \ldots, m$. On a k_3-dimensional object $m - k_3 = j$ controls are at a minimum or maximum deflection and k_3 are free to vary. For the two $(m-3)$-dimensional objects to have in common this k_3-dimensional object, it must be possible for each to achieve the same j controls that define the k_3-dimensional object. That is, there must be j controls that are fixed on each $(m-3)$-dimensional object to the same limiting value or that are free controls that may assume the same constrained value. The test for adjacency is therefore to compare the controls at the intersection point two at a time. If four or more of the same controls are constrained to the same limiting value or are unconstrained at each point, the points are adjacent.

In general, the order of reduction is preserved. This means that when going from R^m to R^n a loss of $m-n$ dimensions occurs. The intersections of P_S with the bounding $(m-1)$-dimensional objects of the m-dimensional subset Ω become the bounding $(n-1)$-dimensional objects of the n-dimensional subset Π. In terms of Fig. 1, the intersections of the two-dimensional P_S with the bounding two-dimensional facets of the three-dimensional subset Ω become the bounding one-dimensional edges of the two-dimensional subset Π. In going from any number of controls to three moments, the facets of Π result from intersections of P_S with the bounding $(m-1)$-dimensional objects of the m-dimensional subset Ω.

If our moment space is of dimension greater than 3, then we must keep track of the intersections that generate two-dimensional figures in R^n and determine the ordering that results in closed, convex two-dimensional objects (polygons). These two-dimensional objects must then be ordered to produce closed, convex three-dimensional objects, and so on, until the $(n-1)$-dimensional bounding objects of Π are determined. The ordering of these objects is based on their adjacency in Ω. We will discuss only the ordering of the intersections that generate two-dimensional figures in R^n; higher order orderings proceed in a similar fashion.

In the three-moment problem, for each of the bounding $(m-1)$-dimensional objects on which intersections occur, we order the points to describe a closed, convex polygon. The edges of this polygon are one-dimensional $(n-2=1)$ and are images of the $(m-2)$-dimensional objects that make up the $(m-1)$-dimensional object of interest. The vertices of this polygon are zero-dimensional $(n-3=0)$ and are images of the $(m-3)$-dimensional objects that make up the $(m-1)$-dimensional object of interest. To define an edge, two vertices must be adjacent in Π, and therefore the two $(m-3)$-dimensional objects of which they are images must be adjacent in Ω.

Once the connectivity of the points is known, the bounding objects can be divided up into $(n-1)$-dimensional objects containing n points. With the resulting closed and convex polygons, we then apply the volume calculations described above to each polygon to determine the total volume of moments attained by the generalized inverse using admissible controls.

Daisy Chaining

There is no obvious easy way to calculate the volume of moments attainable by daisy-chaining solutions. The authors have mapped the attainable moments of some representative daisy-chaining solutions using a brute-force point-and-shoot analysis. For various directions in moment space, the magnitude of the moment vector was scaled until one control in the kth group was just saturated. The results showed that the subset of attainable moments thus determined was concave in several regions with curved bounding surfaces. In this paper, we will estimate the volume of the moments attained by the daisy-chaining solution. This is accomplished by determining the volume of the convex hull of a large set of moments at the extremes of the capabilities of daisy chaining. This estimate will be somewhat generous, since it fills in the concavities of the actual bounding surface.

The program used to determine the convex hull is called quickhull, a general dimension convex hull program that reads a set of points from an input file and returns the smallest convex set that contains the points.[14] The output from quickhull is in the form of the vertices of triangular (for our three-dimensional problem) polygons, for which the volumes may be calculated and summed as described above.

Example

The following example applies the above techniques to a 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 linearlized about a flight condition of 10,000 ft, Mach 0.3, and 12.5 deg 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 Ref. 2:

$$B^T = \begin{bmatrix} -4.382 \times 10^{-2} & -0.5330 & 1.100 \times 10^{-2} \\ 4.382 \times 10^{-2} & -0.5330 & -1.100 \times 10^{-2} \\ -5.841 \times 10^{-2} & -6.486 \times 10^{-2} & 3.911 \times 10^{-3} \\ 5.841 \times 10^{-2} & -6.486 \times 10^{-2} & -3.911 \times 10^{-3} \\ 1.674 \times 10^{-2} & 0.000 & -7.428 \times 10^{-2} \\ -6.280 \times 10^{-2} & 6.234 \times 10^{-2} & 0.000 \\ 6.280 \times 10^{-2} & 6.234 \times 10^{-2} & 0.000 \\ 2.920 \times 10^{-2} & 1.000 \times 10^{-5} & 3.000 \times 10^{-4} \\ 1.000 \times 10^{-5} & 0.3553 & 1.000 \times 10^{-5} \\ 1.000 \times 10^{-2} & 1.000 \times 10^{-5} & 0.1485 \end{bmatrix}$$

For this combination of B matrix and control deflection limits, shown in Table 1, the subset of attainable moments is as shown in Fig. 3.

In this example, the desired moments describe a maneuver known as a velocity vector roll. This is a first-order roll acceleration about the instantaneous velocity vector with constant angle of attack and

Table 1 Control surface deflection limits

Control	Control surface	Minimum value, rad	Maximum value, rad
u_1	Right horizontal tail	−0.4189	0.1833
u_2	Left horizontal tail	−0.4189	0.1833
u_3	Right aileron	−0.5236	0.5236
u_4	Left aileron	−0.5236	0.5236
u_5	Combined rudders	−0.5236	0.5236
u_6	Right trailing-edge flap	−0.1396	0.7854
u_7	Left trailing-edge flap	−0.1396	0.7854
u_8	Roll thrust vector vane	−0.5236	0.5236
u_9	Pitch thrust vector vane	−0.5236	0.5236
u_{10}	Yaw thrust vector vane	−0.5236	0.5236

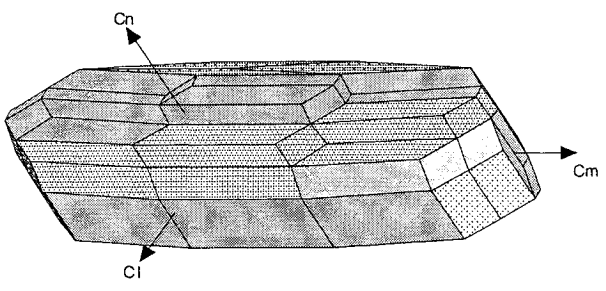

Fig. 3 Example AMS for F-18 HARV.

no sideslip. These moments were generated using a program called Rollerator[15] for the same flight condition as the B matrix. This program requires two inputs to generate a first-order response, the time constant τ and the steady state roll rate P_{ss}. According to MIL-STD-1797A,[16] used in conjunction with MIL-F-8785C,[17] the maximum value of the time constant for level 1 flying qualities is 1 s, and a roll through 30 deg should be performed in 1.1 s. For our maneuvers, we set τ equal to the maximum level 1 value of 1 s. We then specified that the maneuver would be from an initial condition of $\mu = -60$ deg through a final condition of $\mu = 60$ deg. For each allocation scheme, the steady-state roll rate was increased until the scheme could no longer generate the required moments.

Results

Direct Allocation (CAT)

The direct allocation method utilizes all of the available moment-generating capabilities and was able to perform the maneuver with a steady-state roll rate of 215 deg s^{-1}. The time through 30 deg was 0.57 s, which is well within the level 1 specifications. In fact, all of the allocation schemes met this specification. The time to complete the −60 to 60 deg roll was 1.26 s. Figure 4 shows a three-view of the required moments plotted within the AMS.

Pseudoinverse

The pseudoinverse was calculated to be

$$P_{\text{pseudo}} = \begin{bmatrix} -2.201 & -0.7500 & -0.3350 \\ 2.202 & -0.7500 & 0.3353 \\ -2.960 & -9.125 \times 10^{-2} & 0.2210 \\ 2.960 & -9.128 \times 10^{-2} & -0.2210 \\ 9.492 & 1.383 \times 10^{-5} & -2.693 \\ -3.177 & 8.774 \times 10^{-2} & 8.643 \times 10^{-2} \\ 3.177 & 8.771 \times 10^{-2} & -8.646 \times 10^{-2} \\ 1.477 & 6.081 \times 10^{-6} & -2.940 \times 10^{-2} \\ 3.958 \times 10^{-4} & 0.4999 & 2.5786 \times 10^{-4} \\ 0.3015 & -2.537 \times 10^{-5} & 5.325 \end{bmatrix}$$

Figure 5 shows Π for the pseudoinverse within a wire frame of the AMS. This Π fills 13.7% of the AMS. The time through 30 deg was 0.86 s. Although this is well within level 1 flying qualities, it is far below the maximum performance that the aircraft is capable of. The maximum steady-state roll rate achievable with this scheme was 105 deg s^{-1}, and the time to complete the maneuver was 2.0 s.

Best Generalized Inverse

The best generalized inverse was calculated to be

$$P_{\text{best}} = \begin{bmatrix} -6.047 \times 10^{-3} & -0.6600 & -1.130 \times 10^{-2} \\ 1.321 \times 10^{-2} & -0.6586 & -5.608 \times 10^{-3} \\ -5.543 & -4.889 \times 10^{-3} & -8.715 \times 10^{-2} \\ 5.543 & 5.098 \times 10^{-3} & 8.283 \times 10^{-2} \\ 0.3917 & -1.641 \times 10^{-6} & -4.268 \\ -1.398 & 3.934 \times 10^{-2} & -0.1882 \\ 1.452 & 2.897 \times 10^{-2} & 3.182 \times 10^{-2} \\ 5.520 & -3.700 \times 10^{-5} & 4.878 \times 10^{-2} \\ 1.041 \times 10^{-3} & 0.8243 & 1.166 \times 10^{-3} \\ 0.4753 & 1.064 \times 10^{-4} & 4.602 \end{bmatrix}$$

Figure 6 shows Π for the best generalized inverse inside a wire frame of the AMS. This Π fills 42.7% of the AMS. As expected, the increase in moment-generating capability led to better performance. The time through 30 deg was 0.68 s. The maximum steady-state roll rate achievable was 156 deg s^{-1}, and the time to complete the maneuver was 1.52 s.

Tailored Generalized Inverse

By examining Fig. 4, it can be seen that the desired moment coefficients for this entire maneuver are grouped together in a small portion of the AMS. Thus, it was possible to tailor a generalized inverse to this particular maneuver. A generalized inverse was tailored to fit the AMS exactly at the following points:

$$\boldsymbol{m}_1 = \{0.054705, -0.433686, 0.064673\}^T$$

$$\boldsymbol{m}_2 = \{0.067862, -0.433686, 0.064673\}^T$$

$$\boldsymbol{m}_3 = \{0.067862, -0.00727, 0.0907\}^T$$

The inverse was calculated using special subroutines in CAT to be

$$P_{\text{tailored}} = \begin{bmatrix} 0.3606 & -0.9318 & -3.731 \\ 0.3606 & -0.1969 & 1.196 \\ 1.030 & -0.5626 & 3.417 \\ 1.030 & -0.6400 & 2.899 \\ 2.063 & -1.014 & -5.858 \\ -0.2746 & 0.1500 & -0.9112 \\ 0.1445 & 0.3306 & -4.438 \\ 1.030 & -0.5626 & 3.417 \\ -1.030 & 0.8171 & -1.711 \\ 1.030 & -0.5626 & 3.417 \end{bmatrix}$$

Figure 7 shows Π for the tailored generalized inverse inside a wire frame of the AMS. This Π fills only 7.69% of the AMS. However, since the entire maneuver lies within its attainable region, this generalized inverse was able to match the results from the direct allocation method.

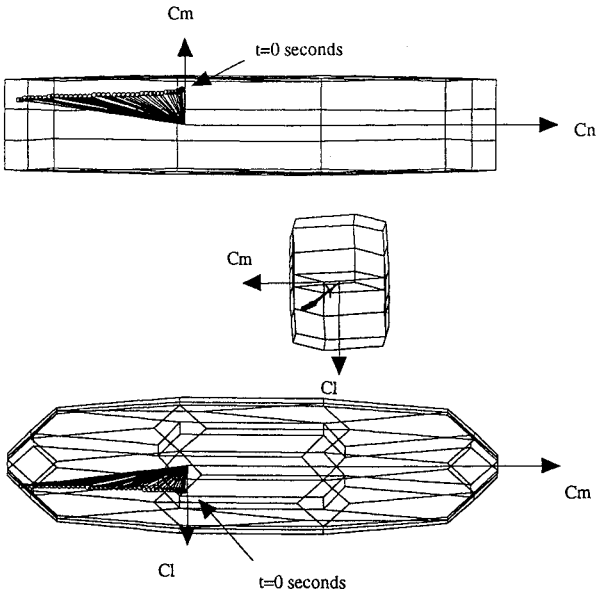

Fig. 4 Required moments for velocity vector roll.

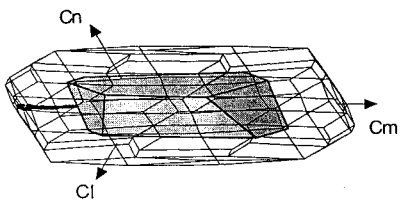

Fig. 5 Π for pseudoinverse.

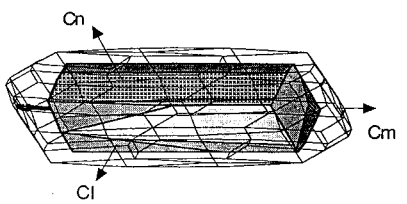

Fig. 6 Π for best generalized inverse.

Table 2 Results summary

Allocation method	Percent of Φ accessible	Maximum P_{ss}, s^{-1}	Time through 30 deg, s	Time for −60 to 60 deg, s
Direct	100	215	0.57	1.26
Pseudoinverse	13.7	105	0.86	2.00
Best generalized inverse	42.7	156	0.68	1.52
Tailored generalized inverse	7.69	215	0.57	1.26
Daisy chaining	22.0	134	0.76	1.72

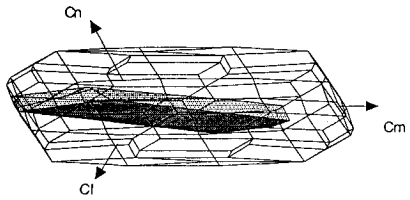

Fig. 7 Π for tailored generalized inverse.

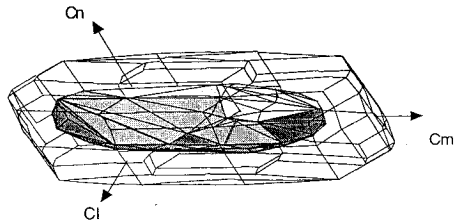

Fig. 8 Π for daisy-chaining scheme.

Daisy Chaining

For daisy chaining, the B matrix was partitioned into two 3×5 matrices. Matrix B_1 contains the more conventional aircraft controls: horizonal tails, ailerons, and rudders. Matrix B_2 contains the remaining controls: trailing-edge flaps and thrust-vectoring vanes. The matrices are inverted using the pseudoinverse producing P_1 and P_2:

$$P_1 = \begin{bmatrix} -4.167 & -0.9243 & -2.429 \\ 4.167 & -0.9243 & 2.429 \\ -5.337 & -0.1124 & -3.254 \times 10^{-3} \\ 5.337 & -0.1124 & 3.254 \times 10^{-3} \\ 0.6722 & 0.000 & -0.1274 \end{bmatrix}$$

$$P_2 = \begin{bmatrix} -7.186 & 0.4653 & 0.4866 \\ 7.185 & 0.4649 & -0.4868 \\ 3.338 & -1.567 \times 10^{-5} & -0.2125 \\ -2.637 \times 10^{-5} & 2.651 & -1.508 \times 10^{-4} \\ -6.745 \times 10^{-3} & -1.784 & 6.734 \end{bmatrix}$$

Figure 8 shows an approximation of Π for this daisy-chaining scheme inside a wire frame of the AMS. This Π fills approximately 22.0% of the AMS. The time through 30 deg was 0.76 s. The maximum steady-state roll rate achievable was 134 deg s^{-1}, and the time to complete the maneuver was 1.72 s. A summary of the results is shown in Table 2.

Discussion

The pseudoinverse has been a standard method for solving control allocation problems due to the ease with which it can be calculated. However, it can be an extremely poor solution to problems that involve limits on the controls because the pseudoinverse is in no way related to these limits and does not account for them. The realization of this situation has begun a search for more elegant methods of solving the problem.

Finding a generalized inverse that maximizes the portion of the attainable moments that can be obtained results in a significant improvement in performance. However, calculating this best inverse can be mathematically difficult, involving iterative methods that introduce complicated function evaluations.

Generalized inverse tailoring is a simple method that can yield excellent results if the maneuver is known ahead of time and the desired moments are grouped close together. The weakness of tailoring stems from the fact that inverses must be calculated for each maneuver or collection of moments. Using different generalized inverses for different maneuvers can cause problems. Switching from one generalized inverse to another could cause an instantaneous reconfiguration of controls. This is impractical for real controls.

Daisy chaining appeared to perform well, but the method cannot access all the attainable moments. The particular maneuver performed and the manner in which the controls are grouped into B_1 and B_2 have a significant effect upon the results from daisy chaining.

The direct method of allocation is capable of allocating admissible controls for every attainable moment. If the desired moments exceed the attainable moments, then this method cannot allocate the controls necessary to achieve them (but no other method can either). We argue that a properly designed control law should never ask more of the control effectors than they are capable of delivering, since this invalidates the assumptions on which the control law is based. Anytime the control law makes demands in excess of the control effectors' capabilities, the airplane is de facto out of control.

The control allocation method that accesses the greatest volume of the attainable moments provides the greatest amount of controlled maneuvering capability. Allocation methods that do not access a substantial portion of the attainable moments results in untapped maneuvering potential. Seen another way, the capabilities of any allocation method that accesses a small portion of the attainable moments can be duplicated by a better allocation method accessing a smaller set of attainable moments. A smaller set of attainable moments can result from fewer or smaller control effectors, which is a savings in weight and/or complexity.

Conclusions

Many control allocation schemes used today cannot access all of the moments a given control configuration can generate. The deficiencies of these schemes can be measured and compared. The consequences of these deficiencies can be demonstrated to be a significant loss in performance.

Acknowledgment

This work was conducted under NASA Grant NAG-1-1449.

References

[1]Snell, S. A., Enns, D. F., and Garrard, W. L., "Nonlinear Inversion Flight Control for a Supermaneuverable Aircraft," AIAA Paper 90-3406, Aug. 1990.

[2]Adams, R. J., Buffington, J. M., Sparks, A. G., and Banda, S. S., "An Introduction to Multivariable Flight Control System Design," Wright Labs., WL-TR-92-3110, Oct. 1992, pp. 111–116.

[3]Shaw, P. D., Haiges, K. R., Rock, S. R., Vincent, J. H., Emami-Naeini, A., Anex, R., Fisk, W. S., and Berg, D. F., "Design Methods for Integrated Control Systems," Air Force Wright Aeronautical Labs., AFWAL-TR-88-2061, June 1988.

[4]Vincent, J. H., Emami-Naeini, A., and Khraishi, N. M., "Design Challenge in Automatic Flight Control: The SCT Solution," AIAA Paper 91–2633, Aug. 1991.

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

[6]Colbaugh, R., and Glass, K., "Cartesian Control of Redundant Robots," *Journal of Robotic Systems*, Vol. 6, No. 2, pp. 432–434.

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

[8]Durham, W., "Attainable Moments for the Constrained Control Allocation Problem," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 6, 1994, pp. 1371–1373.

[9]Brogan, W. L., *Modern Control Theory*, 2nd ed., Prentice–Hall, Englewood Cliffs, NJ, 1982.

[10]Stevens, B. L., and Lewis, F. L., *Aircraft Control and Simulation,* Wiley, New York, 1992.

[11]Bugajski, D., Enns, D., and Hendrick, R., "Nonlinear Control Law Design for High Angle-of-Attack," presented at High-Angle-of-Attack Projects and Technology Conference, NASA Dryden Flight Research Facility, Edwards, CA, April 1992.

[12]Hausner, M. A., *Vector Space Approach to Geometry,* Prentice–Hall, Englewood Cliffs, NJ, 1965.

[13]Gale, D., *The Theory of Linear Economic Models,* McGraw–Hill, New York, 1960.

[14]Barber, B. C., Dobkin, D. P., and Huhdanpaa, H., "The Quickhull Algorithm for Convex Hull," Research Rept. GCG53, Geometry Center, Univ. of Minnesota, July 1993.

[15]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.

[16]Anon., "Military Specification-Flying Qualities of Piloted Vehicles," MIL-STD-1797A, March 1987.

[17]Anon., "Military Specification-Flying Qualities of Piloted Airplanes," MIL-F-8785C, Nov. 1980.