JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 17, No. 2, March–April 1994

Constrained Control Allocation: Three-Moment Problem

Wayne C. Durham*
Virginia Polytechnic Institute and State University, Blacksburg, Virginia 24061

This paper presents a method for the solution of the constrained control allocation problem for the case of three moments. The control allocation problem is to find the "best" combination of several flight control effectors for the generation of specified body-axis moments. 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 controls are assumed to be individually linear in their effect throughout their ranges of motion and complete in the sense that they generate moments in arbitrary combinations. The best combination of controls is taken to be an apportioning of the controls that yields the greatest total moment in a specified ratio of moments without exceeding any control constraint. The method of solving the allocation problem is presented as an algorithm and is demonstrated for a problem of seven aerodynamic controls on an F-18 airplane.

Nomenclature

B = matrix of control effectiveness
m = number of controls
n = dimension of moment space, 3
$\boldsymbol{u}$ = vector of controls
Φ = subset of attainable moments
Ω = subset of constrained controls
$\partial(\Sigma)$ = boundary of the set Σ

Subscript

d = desired

Superscript

$*$ = lies on $\partial(\cdot)$
$\hat{}$ = unit vector

Introduction

CLASSICALLY, airplane flight controls are designed with the idea of a single moment-generating controller for each rotational degree of freedom. Modern tactical aircraft have more than three moment generators. In the high angle-of-attack research vehicle, one has potentially 13 or more independent moment controllers: horizontal tail, aileron, leading edge flap, trailing edge flap, and rudder, each left and right; and three thrust-vectoring moment generators. To this add spoilers on the leading edge extensions and vertical lift and side-force generators, and the number of controls nears 20. These controls are all constrained to certain limits, determined by the physical geometry of the control actuators, or in some cases by aerodynamic considerations. The allocation, or blending, of these several controls to achieve specific objectives is the control allocation problem.

Aside from the near universal application of generalized inverse solutions to the problem (see Refs. 1 and 2, for example) other methods of interest include techniques called "pseudocontrols"[3,4] and "daisy chaining."[5,6] It can be shown[7] that no single generalized inverse can, for arbitrary moment demands, yield solutions that attain the maximum available moment without violating some control constraint. The use of pseudocontrols is an application of generalized inverses, imbedded in a dynamic inversion control law.[7] Daisy chaining likewise cannot yield maximum available moments everywhere, and additionally tends to demand higher control actuator rates than other methods.

The geometry of the constrained control allocation problem was developed in Ref. 7. In that reference, a nonlinear solution was described called the direct method. The direct method by definition yields maximum available moments and is the subject of the algorithm to be presented. The direct method loosely consists of two parts: the determination of the attainable moment subset and the calculation of those controls that yield moments within and on the boundary of that subset without violating the control constraints.

The determination of the boundary of the attainable moment subset may be done by inspection for the two-moment problem, but is decidedly nontrivial for the three-moment problem. There are generic methods of finding the boundary,[8] such as "package wrapping," that begin with an arbitrary set of points and extensively use trigonometric relationships to find the convex hull of the set of points. These methods do not exploit the structure of the control allocation problem and generally do not yield descriptions of the boundary suitable for the calculation of controls. The first part of the method presented in this paper describes a means of determining the boundary, yielding a description of the boundary that contains the necessary information for the determination of controls in the allocation problem.

Once the boundary of the attainable moment subset has been found, the task is to allocate the controls in a manner that generates specified moments within the constraints placed on the controls. We consider some desired moment for which we wish to determine the allocation of controls. The determination of the desired control consists of 1) the identification of the point of intersection of a line in the direction of the desired moment with the bounding surface of attainable moments and 2) calculation of the controls that generate that intersection, with possible scaling of the controls if the desired moment is interior to the subset of attainable moments. These steps were demonstrated in Ref. 7 for the two-moment problem, but as with the determination of the boundary, it is nontrivial for the three-moment problem. The second part of the method presented in this paper describes a means of determining the controls for the three-moment problem.

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 constrained control subset Ω

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

Received June 30, 1992; revision received April 8, 1993; accepted for publication May 10, 1993.

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

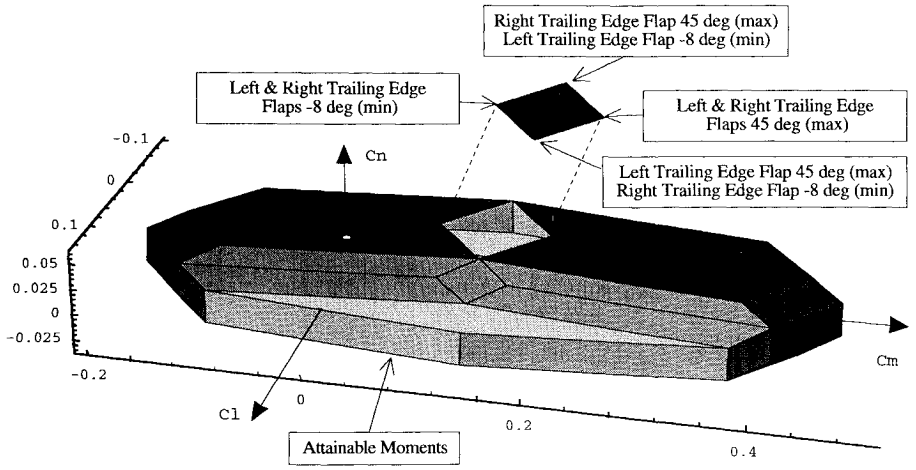

Fig. 1 Subset of attainable moments.

Controls that lie on the boundary of Ω, $\partial(\Omega)$, are denoted by u^*, $u^* = \{u \mid u \in \partial(\Omega)\}$.

These controls generate moments m through a linear mapping $B{:}R^m \rightarrow R^n$ onto n-dimensional moment space $m \in R^n$ such that $Bu = m$, where $m > n$. B is the control effectiveness matrix with respect to the moments.

Denote by Φ the image of Ω in R^n, $\Phi \subset R^n$. In the mapping $R^m \rightarrow R^n$, some of the vertices, edges, and facets of $\partial(\Omega)$ project to the interior of Φ, whereas the others become the boundary of Φ. Parallel edges of $\partial(\Omega)$ remain parallel under the linear mapping, so that the facets of $\partial(\Phi)$ are parallelograms.

A typical attainable moment subset is shown in perspective view in Fig. 1, in which nondimensional moment coefficients have been used. Figure 1 was generated using seven controls and will be described later. Each vertex of the figure represents the moment generated by a combination of seven controls at one or the other of their constraints. Along each edge, one control is varying whereas six are on a constraint. Each facet is generated by saturating five controls and allowing two controls to assume values within their respective constraints. This figure is discussed in more detail later.

For the geometric discussion that follows, we require that the control limits defined in Eq. (1) be such that the origin of moment space be contained strictly within Φ. We will further require that *every* $n \times n$ matrix formed of the columns of B be of full rank. The significance of this requirement is that it makes all admissible solutions to the allocation problem unique. This requirement is not necessary for the general solution to the problem, and its impact will be mentioned where appropriate. Of natural concern is the potential to reallocate the controls following control failure, which will introduce a column of zeros in the B matrix. In this event, to implement the algorithm to be described, one simply removes the failed control (and the column associated with it) from the problem and recalculates the solution.

Moments that lie on the boundary of Φ, $\partial(\Phi)$, are denoted by m^*. A unit vector in the direction of m will be denoted by $\hat{m}$.

The control allocation problem is defined as follows: given B, Ω, and some desired moment m_d, find $\partial(\Phi)$ and then determine the controls $u \in \Omega$ that generate the desired moment for the largest possible magnitude $|m_d|$ in the direction $\hat{m}$.

Method

Determination of the Boundary of Attainable Moments

In outline, the method we will use to determine $\partial(\Phi)$ is as follows (section references are found in the Appendix):

1) Determine all of the nodes and connections of Φ. (A node is a geometric feature of the constrained control and attainable moment subsets. Nodes are generated in m-space by placing each of the m controls at either of its two constraints, yielding 2^m nodes in the subset of constrained controls. Nodes in the attainable moment subset are the images of the nodes in the constrained control subset.)

2) Begin with a node that is known to be a vertex. (A vertex is any node that lies on the boundary of the subsets Φ or Ω. Thus all nodes in Ω are vertices, but a node in Φ may or may not be a vertex.) Make this vertex the first of a list of vertices.

3) Test the connections from each vertex on the list of vertices to determine which are edges. (Connections are a geometric feature of the constrained control and attainable moment subsets. Connections are generated in m-space by placing all but one of the m controls at either of their two constraints and allowing one to vary, yielding $2^{m-1} \cdot m!/(1! \cdot (m-1)!)$ connections in the subset of constrained controls. Connections in the attainable moment subset are the images of the connections in the constrained control subset. All connections in Ω are edges, but a connection in Φ may or may not be an edge.)

4) Add the to-node of each edge thus determined to the vertex list. (When traversing a connection in a particular direction, the to-node is the node toward which one is proceeding. When considering a node and its various connections, the node under consideration is the from-node, and all those that define connections that include that node are to-nodes. The edge is any connection that lies on the boundary of the subsets Φ or Ω. All connections in Ω are edges, but a connection in Φ may or may not be an edge.)

5) Repeat steps 3 and 4 until no new vertices are added.

6) Compile a list of facets of $\partial(\Phi)$. (A face is a geometric feature of the constrained control and attainable moment subsets. Faces are generated in m-space by placing all but two of the m controls at either of their two constraints 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 attainable moment subset are the images of the faces in the constrained control subset. Facets are any face 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.)

The fact that the vertices of $\partial(\Phi)$ are simply connected by their edges insures that this procedure will find all of the vertices and edges. Of these steps, only the third is in any way challenging. The third step is a linear algebra problem whose solution is the heart of this algorithm. Each step is now described in detail.

1) *Determine all of the nodes and connections of* Φ. The nodes of Φ are the images of the vertices of $\partial(\Omega)$. They are calculated from $m_i = Bu_i^*$, $i = 0 \ldots (2^m - 1)$. For each node thus calculated, a record is established. The first entry in this record is a list of the to-nodes associated with each connec-

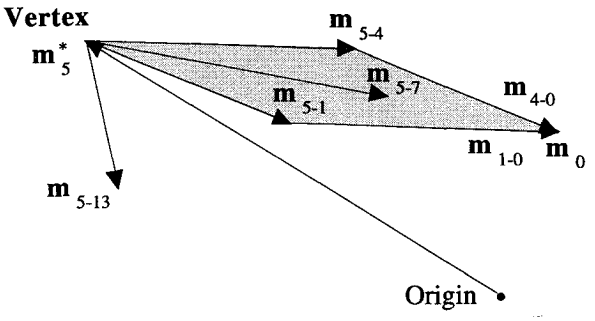

Fig. 2 Sample vertex and four connections.

tion. Section A.1 describes the system of enumeration used, and Sec. A.2 describes a method for to-node determination. For a given node, a list of its to-nodes completely defines all of its connections.

2) *Begin with a node that is known to be a vertex*. As each node is identified in step 1, its distance from the origin is calculated and compared with those of previously determined nodes to determine the node of maximum distance from the origin. The distance is the Euclidean $|\boldsymbol{m}_i|$. The furthest node from the origin is a prima facie vertex and heads the vertex list.

3) *Test the connections from each vertex on the vertex list to determine which are edges*. For illustration say we have four controls, and that $\boldsymbol{m}_5^*$ has been found to be a vertex, as shown in Fig. 2. The connections from $\boldsymbol{m}_5^*$ are to nodes $\boldsymbol{m}_4$, $\boldsymbol{m}_7$, $\boldsymbol{m}_1$, and $\boldsymbol{m}_{13}$ (see Sec. A.2 for the numbering scheme). Any combination of two of these four connections forms a face. No two connections may be collinear, a result of the requirement that every 3×3 matrix formed of the columns of B be of full rank. For example, the face formed by $\boldsymbol{m}_{5-1}$ and $\boldsymbol{m}_{5-4}$, has as its other two connections $\boldsymbol{m}_{1-0}$ and $\boldsymbol{m}_{4-0}$, node $\boldsymbol{m}_0$ completing the face. There will be five more (not depicted) faces in Fig. 2. We wish to determine if a particular face, such as that depicted in Fig. 2, is a facet.

We make the following definitions: if a line from the origin to any point on a given connection passes through any face first, that connection is superior to the face; if that line intersects the face after intersecting the connection, the connection is inferior to the face. Inferior connections are inside the three-dimensional attainable moment subset. For a given connection and a given face, a line from the origin through the connection may not intersect the face at all, since connections and faces are bounded in their respective dimensions. We do not consider the possibility of a connection lying exactly in the plane of a face of which it is not a side, a further consequence of the requirement that every 3×3 matrix formed of the columns of B be of full rank. We further agree not to compare a connection with a face of which it is a side. Note that if a connection is inferior to any face at a vertex, that connection cannot be an edge (nor can any face of which it is a side be a facet). If a connection is superior to every face at a vertex, then it is an edge, and vice versa. If a connection is an edge, then both its from-node and its to-node are vertices. A means for determining the inferiority or superiority of connections with respect to faces is described in Sec. A.3.

4) *Add the to-node of each edge thus determined to the vertex list*. Every connection at each vertex is tested against each face at that vertex, excluding faces of which it is a side and excluding faces formed of connections both of which have been previously determined to be inferior to other faces. When all connections have been thus tested, those that were inferior to no face are promoted to the status of edges, and their to-nodes are added to the list of vertices.

5) *Repeat steps 3 and 4 until no new vertices are added*. As a linear transformation of $\partial(\Omega)$, $\partial(\Phi)$ shares its properties of boundedness and connectedness. Since step 3 identifies every edge at every vertex, step 4 will ultimately identify every vertex, and $\partial(\Phi)$ is completely determined.

6) *Compile a list of facets of* $\partial(\Phi)$. The first part of this step is accomplished by traversing the completed vertex list. Associated with each vertex is a list of edges, at least three in number. Each combination of two of these edges is examined to determine the node that completes the face they form. If the completing node thus determined is itself a vertex, then the edges form a facet, otherwise not. See Sec. A.4 for details.

As each facet is determined, then if no description of that facet has already been determined (from another vertex; see Sec. A.5), it is added to a list of facets. Each facet is described by a principal vertex and two edges from that vertex. (The choice of which vertex to call the principal one is arbitrary.) Associated with each facet is a precomputed matrix inverse (3×3) for determining the intersection of a line from the origin with the plane of the facet (see Sec. A.6). The existence of this inverse relies on the requirement that the origin of moment space be strictly within Φ. For reasons to be described later, we also record each of the four neighboring facets associated with the facet thus determined (neighboring in the sense that the two facets share a common edge).

The list of facets, when finished, contains the complete description of $\partial(\Phi)$ needed for the subsequent calculation of controls. When plotted in three-dimensional space, the result is similar to that shown in perspective in Fig. 1.

Determination of the Controls

Having determined the boundary of the attainable moment subset, we now wish to allocate the controls in a manner that achieves the maximum available moment in any direction in moment space within the constraints placed on the controls. We consider the desired moment as a vector in moment space $\boldsymbol{m}_d$. If the magnitude of the desired moment is greater than or equal to that which is attainable ($\boldsymbol{m}_d$ lies outside or on the boundary), then we will take as our solution the controls that generate the maximum attainable moment in the same direction as the desired moment vector. If the magnitude of the desired moment is less than the maximum attainable in the desired direction, we will uniformly scale the controls that generate the maximum so as to yield the desired magnitude.

The solution in either case requires the determination of the controls that generate the maximum moment in the desired direction. As a geometric figure in three-dimensional moment space, the set of attainable moments is a multifaceted solid figure (see Fig. 1). We have created a description of this figure that consists of, for each of its facets, the coordinates of one of the facet's vertices, the two edges of the facet that are associated with that vertex, a 3×3 matrix for the determination of the intersection of a line from the origin with the plane of the facet, and a list of the four neighboring facets. Before we can determine the controls, we need to find the facet toward which the desired moment points.

Finding the Right Facet

Consider a facet selected randomly for consideration, and some desired moment $\boldsymbol{m}_d$. From Sec. A.6 we may calculate the numbers a, b, and c in the equation

$$a\hat{\boldsymbol{m}}_d = \boldsymbol{m}_i^* + b\boldsymbol{m}_{i-j}^* + c\boldsymbol{m}_{i-k}^* \tag{2}$$

We will use our record of the four neighboring facets of each facet as a "referral service," along with the values of a, b, and c in the solution of Eq. (2) to determine the right facet. For the chosen facet to be the correct one, we must have

$$a > 0, \qquad 0 \le b \le 1, \qquad 0 \le c \le 1 \tag{3}$$

The requirement that $a > 0$ may be used to improve our initial guess for the right facet. For an arbitrary direction $\hat{\boldsymbol{m}}_d$, this is accomplished by precalculating a list of first-guess

facets to be used in beginning the search. There are eight first-guess facets, one for each octant of moment space. Within any octant, the first-guess facet is the one that contains the intersection of a line from the origin that passes through the center of the octant, i.e., that passes through (1, 1, 1), (1, −1, 1), (−1, −1, 1), etc. It is then a simple matter to determine the octant in which $\hat{m}_d$ lies and begin the search with the first-guess facet for that octant. This will assure that $a > 0$.

We now look in the vicinity of this guessed facet to find one that satisfies the conditions on b and c. To do this, we use our previously recorded list of the four neighbors of each facet. Consider Fig. 3, which shows a hypothetical section of the boundary of attainable moments. Let facet A be the facet we have assumed and facet D be the correct one. The intersection of $a\hat{m}_d$ with facet D is shown as a dot.

We have already assured that $a > 0$. We now examine the values of b and c for facet A. From this we learn that both b and c are > 1 and that $c > b$. Unless $0 \le b \le 1, 0 \le c \le 1$, then the vector sum $bm^*_{i-j} + cm^*_{i-k}$ cannot lie on facet A, and facet A does not contain the point of intersection. Because $c > b$, the point of intersection lies in the vicinity of neighboring facet B, or at least facet B is closer to the intersection than facet A.

The same test applied to facet B yields new values of b and c, which are analyzed to see if the intersection lies on facet B, and if not, in which direction to proceed. This further refers us to facet C, and finally to facet D, for which we find $0 \le b \le 1, 0 \le c \le 1$. Facet D therefore contains the intersection, and the values of a, b, and c may now be used to calculate the controls, as described later.

In an implementation for which the desired moment varies continuously, we keep track of the current facet. As some new direction $\hat{m}_d$ is received, we begin the facet search in the current facet. This strategy should require at most one referral to a neighboring facet. In practice this will depend on the speed with which the desired moment varies and the frequency of recalculating the allocation of controls, so we monitor the sign of a; reverting to the first-guess facet should a prove to be negative for the current facet.

In the implementation described later, the algorithm included a bail-out procedure. This procedure was called in the event the same facet was visited twice during a search and constituted a brute force search of all of the facets to find the one for which the conditions in Eq. (3) were satisfied.

Determining the Controls

Assume we know which facet the desired moment points to, that is, the facet intersected by the half-line in the direction of the desired moment. Call this facet's describing vertex and edges m^*_i, m^*_{i-j}, and m^*_{i-k}. The controls associated with $a\hat{m}_d$ are the desired controls u^*_d. Because we have a linear transformation

$$a\hat{m}_d = Bu^*_d = m^*_i + bm^*_{i-j} + cm^*_{i-k}$$

$$= Bu^*_i + bBu^*_{i-j} + cBu^*_{i-k}$$

Whereas the mapping from R^m to R^n is many-to-one, the mapping from $\partial(\Omega)$ to $\partial(\Phi)$ is one-to-one, so we have the unique solution

$$u^*_d = u^*_i + bu^*_{i-j} + cu^*_{i-k} \tag{4}$$

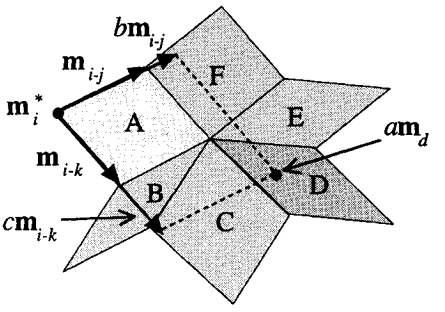

Fig. 3 Searching for the right facet.

In other words, the controls that generate the maximum moment in the direction of the desired moment are found from the point on the boundary of the constrained control subset that corresponds to the point of intersection in moment space, i.e., that which maps to the point of intersection.

If the maximum attainable moment in the desired direction is less than the magnitude of the desired moment ($a < |m_d|$), then we use u^*_d as our solution. If the maximum attainable moment in the desired direction is greater than or equal to the magnitude of the desired moment ($a \ge |m_d|$), then we scale u^*_d according to

$$u_d = u^*_d |m_d| / a \tag{5}$$

Implementation

Two programs were written, corresponding to the two parts of the method presented earlier. The first program generated descriptions of the attainable moment subset. These were used as input for the second program, which returned the controls for arbitrary user-provided moments.

The allocation scheme was applied to a nonlinear dynamic inversion control law for a simulation of the F-18 airplane. The application consisted of an off-line determination of the attainable moment subset for the F-18 flight controls at a reference flight condition, and on-line control allocation using the allocation program as a subroutine. Since the control effectiveness may vary with flight condition (as the B matrix varies), the attainable moment subset used was strictly only valid at the reference conditions. However, the dynamic inversion control law that was utilized demonstrated little sensitivity to the actual variations in these parameters during the selected maneuver, and scheduling was not required.

Simulation Results and Discussion

We take for our example

$$m = \begin{Bmatrix} C_l \\ C_m \\ C_n \end{Bmatrix}, \qquad u = \begin{Bmatrix} u_1 \\ \vdots \\ u_7 \end{Bmatrix} \tag{6}$$

The nondimensional moment coefficients are used, and the controls are measured in degrees deflection. The controls u_1 and u_2 are the left and right horizontal tails, u_3 and u_4 are the left and right trailing edge flaps, u_5 and u_6 are the left and right ailerons, and u_7 is the symmetric contributions of the rudders.

The B matrix, whose entries have units of 1/deg, and the control limits used in the simulation were as follows:

$$B = \begin{bmatrix} 2.38 \times 10^{-4} & -2.38 \times 10^{-4} & 1.23 \times 10^{-3} & -1.23 \times 10^{-3} & 4.18 \times 10^{-4} & -4.18 \times 10^{-4} & 3.58 \times 10^{-5} \\ -6.98 \times 10^{-3} & -6.98 \times 10^{-3} & 9.94 \times 10^{-4} & 9.94 \times 10^{-4} & -5.52 \times 10^{-4} & -5.52 \times 10^{-4} & 0 \\ -3.09 \times 10^{-4} & 3.09 \times 10^{-4} & 0 & 0 & -1.74 \times 10^{-4} & 1.74 \times 10^{-4} & -5.62 \times 10^{-4} \end{bmatrix} \tag{7}$$

$$u^T_{\min} = \{-24.0 \ -24.0 \ -8.0 \ -8.0 \ -30.0 \ -30.0 \ -30.0 \text{ deg}\}$$

$$u^T_{\max} = \{10.5 \ \ 10.5 \ \ 45.0 \ \ 45.0 \ \ 30.0 \ 30.0 \ \ 30.0 \text{ deg}\} \tag{8}$$

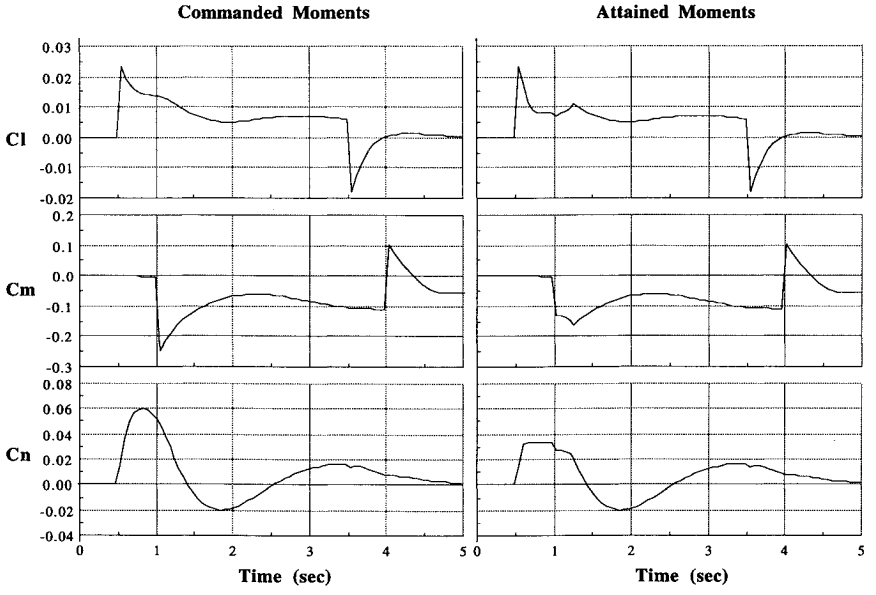

Fig. 4 Commanded and attained moments.

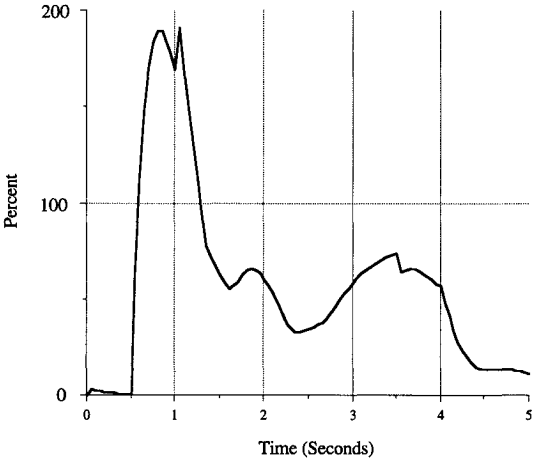

Fig. 5 Percent saturation of commanded moments.

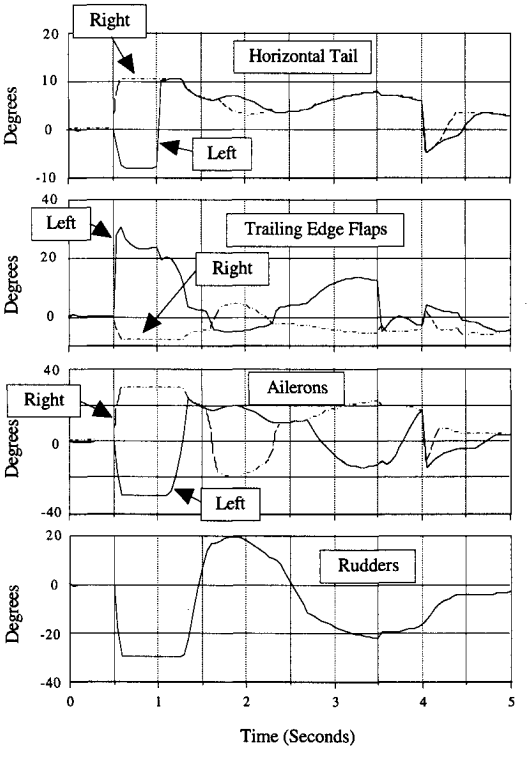

Fig. 6 Control time histories.

Figure 1 is the attainable moment subset generated by these controls. One of the facets has been lifted from the top to aid in visualization. The particular facet shown represents the moment coefficients generated by all permissible combinations of left and right trailing edge flaps, whereas the left horizontal tail is held fixed at its minimum (−24 deg), the right horizontal tail at its maximum (10.5 deg), the left aileron at its minimum (−30 deg), the right aileron at its maximum (30 deg), and the rudder at its minimum (−30 deg).

Results of a representative maneuver are shown in Figs. 4–8. The maneuver began in a 5-g level left turn and consisted of an aggressive roll to the right from $t = 0.5$ to 3.5 s. This roll was accompanied by a decrease in load factor to 3 g between $t = 1.0$ and 4.0 s, followed by a return to 5-g flight. The maneuver finishes in a level turn to the right. Details of this maneuver and the control law used are intentionally omitted to focus attention on the performance of the control allocation algorithm.

Figure 4 shows the moment requirements placed on the controls for the completion of this maneuver and the moment coefficients actually attained. At any instant of time, the combination of commanded moments shown in Fig. 4 represents the desired moment m_d. Figure 5 is the percentage of saturation of controls that results from attempting to generate m_d. This value a (as a percentage) from Eq. (2). It is seen that between $t = 0.60$ and 1.25 s, the demand exceeded the capabilities of the controls.

At any time before $t = 0.60$ s and after $t = 1.25$ s, none of the controls should be saturated, and we expect the moment generated to be the same as the desired moment, which is verified by Fig. 4. Figure 6 shows the control activity, and we verify that during the period of time when capabilities exceeded demand two controls were saturated, and that at other times none were saturated.

Figure 7 reproduces the attainable moment subset, cut away to show the interior. Within the interior are shown several of the commanded moments (m_d) in the first 1.2 s. The cluster of commanded moments labeled "$t < 1.0$ sec" shows the excess

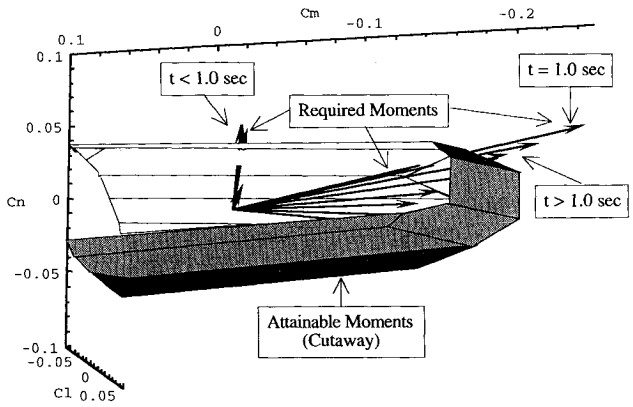

Fig. 7 Cutaway of attainable moments with required moments.

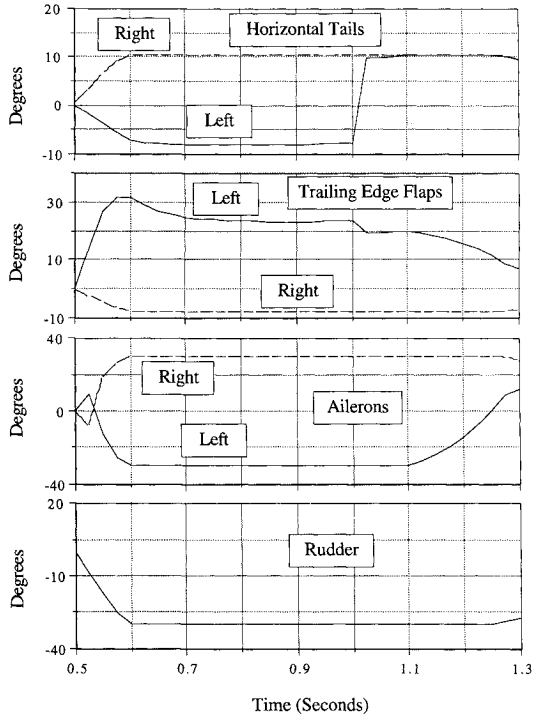

Fig. 8 Control time histories during saturation.

demand for rolling and yawing moments. At $t = 1.0$ s, a large nose-down pitching moment requirement is introduced, and $\boldsymbol{m}_d$ moves abruptly to the position shown, decreasing subsequently until $\boldsymbol{m}_d$ lies completely within the figure. The controls allocated during the period of saturation correspond to the intersections of $\boldsymbol{m}_d$ with the boundary, as shown.

Figure 8 focuses on the control activity during the period of control saturation. The moments being generated are all on the boundary of the attainable moments. Since the desired moment does not generally point directly at a vertex or an edge, we expect to see all controls saturated except two during the period $t = 0.60$ to 1.25 s. From Fig. 8 we see that one of these unsaturated controls is the left trailing edge flap. The other unsaturated control varies in this example. From $t = 0.60$ to 1.10 s, it is the left horizontal tail. At about $t = 1.0$ s, the desired moment moves from one facet to another, as may be seen in Fig. 7. At this time the facet referral system was called into play and immediately identified the new facet. Once on this new facet, the left horizontal tail is saturated, whereas the left aileron is off its constraint.

Conclusions

The determination of the subset of attainable moments involves an admittedly complicated algorithm. Nonetheless, it is completely deterministic, and should always yield correct results within the limitations of the assumptions made. Whether or not it could be used for the real-time reallocation of controls following the identification of control failures remains an open question. Part of the question involves the computational time, which may be moot given the steady increases in onboard computing capabilities. Of potentially more serious consequence are the assumptions made, in particular that every 3×3 submatrix of B be nonsingular. Singularities within the B matrix are not pathological cases, as may be seen from a cursory examination of the effects of pure thrust vectoring. Research in this area is ongoing.

Given the description of the subset of attainable moments, real-time allocation of the controls is decidedly practical. The actual allocation described herein requires a simple 3×3 matrix multiplication once the correct facet is determined. This is less computationally intensive than the use of generalized inverses, which requires an $m \times 3$ matrix multiplication. The facet referral system requires a 3×3 matrix multiplication for each facet tested, but a continuously varying moment demand seldom requires more than two such tests.

Future Research

There are many areas associated with the control allocation problem that are ripe for research. Primary among these areas is an extension of the control effects to include forces as well as moments. With the inclusion of forces, one is then prepared to address such problems as the determination of maximum control-generated lift during maneuvering or the determination of minimum control-generated drag during cruise or maneuvering. Of perhaps equal importance is the introduction of nonlinearities into the effectiveness of the various controllers, to include a loss of effectiveness at higher control deflections and changes in effectiveness that result from interference from other control deflections (for example, the change in horizontal tail effectiveness with flap deflection). Finally, the impact of violations of the assumptions made must be fully examined and a means of dealing with these impacts determined.

Appendix

In this Appendix, binary numerals will be distinguished by the subscript 2, such as 0101_2 as the binary representation of decimal 5. Since no number bases other than 2 and 10 are used, an unsubscripted numeral is understood to be a decimal representation, so that the number 5 in $\boldsymbol{m}_5^*$ is decimal 5.

A.1 Vertex Numbering

Vertices are numbered according to a binary representation that reflects which controls are minimum and which are maximum, such that a 0 in the ith least significant bit of the binary number indicates the ith control is a minimum, and a 1 indicates it is at a maximum. For example, a five-control problem will have a vertex in Ω numbered 9. Decimal 9 is binary 01001_2, so the first and fourth controls are at their maximum limit and the others at their minimum. Controls are enumerated $0 \ldots m-1$, so in this example the controls u_1, u_2, and u_4 are on their minimum constraints, whereas u_0 and u_3 are on their maximum constraints.

A.2. To-Node Determination

The list of to-nodes is easily determined by logical bit-wise XOR (exclusive OR) operations on binary representations of the node number and a number (1, 2, 4, 8 . . .) called the control mask. The control mask of the ith control is decimal 2^i, $i = 0 \ldots m-1$, and the effect of XORing a node number with a control mask is to flip the bit in the binary representation of the node number associated with the control, thus defining a to-node. Consider a typical vertex in a four-control

problem, for illustration say m_5^* (0101_2) has been found to be a vertex. For each control there is a connection at this vertex, and the to-nodes are obtained by XORing the number (0101_2) with each of the four control masks (0001_2, 0010_2, 0100_2, and 1000_2). Thus XOR(0101_2, 0001_2) = 0100_2 = 4 yields m_4^*, a to-node. The other to-nodes are quickly identified as m_7^* (0111_2), m_1^* (0001_2), and m_{13}^* (1101_2).

A.3. Edge Testing

We claim without proof that any three vectors comprised of one vertex and any two connections from that vertex form a basis for R^3. We take these two connections to be those defining a face against which some connection is to be tested, and the vertex to be the common vertex of those two connections. Denote this vertex m_i^* and the two connections m_{i-j} and m_{i-k}. Call the connection being tested m_{i-x}. Any point on m_{i-x} may then be expressed as a linear combination of the three basis vectors m_i^*, m_{i-j}, and m_{i-k}. For such a point on the connection being tested we choose the to-vertex of that connection (since its coordinates are already known), and calculate the coefficients a, b, and c in the vector equation

$$m_x = am_i^* + bm_{i-j} + cm_{i-k}$$

$$\begin{Bmatrix} a \\ b \\ c \end{Bmatrix} = [m_i^* : m_{i-j} : m_{i-k}]^{-1} m_x$$

The inverse exists because its columns are basis vectors for three-space. With these results we reason as follows: if either b or c or both are negative, then there is no line from the origin that passes through any point on m_{i-x} and the face determined by m_{i-j} and m_{i-k}. If both b and c are positive, then the opposite is true. If both b and c are positive, then we examine the value of a: if $a > 1$, m_{i-x} is superior to this face, and if $a < 1$, it is inferior. We should never have $a \equiv 1$ for then m_{i-x} is coplanar with the face being tested, a possibility we do not allow.

A.4. Facet Identification

For a vertex m_i^*, obtained by the combination of controls u_i^*, assume that connections m_{i-j} and m_{i-k} have survived the tests in Sec. A.3 and are therefore edges m_{i-j}^* and m_{i-k}^*. Then m_j and m_k are promoted to vertices m_j^* and m_k^* reached from m_i^* by varying two different controls from one extreme to another. Call these controls u_a (for m_j^*) and u_b (for m_k^*). We wish to determine whether or not m_{i-j}^* and m_{i-k}^* form a facet. The fourth node of this face, say m_x, is identified from

$$x = \text{XOR}(j, 2^b) \text{ or } x = \text{XOR}(k, 2^a)$$

We then examine the vertex list to see if m_x is present; if it is, the face is a facet, else it is not.

A.5. Duplicate Facets

Facets are uniquely described by their four vertices. The easiest way to keep track of facets was found to be by association with a binary number whose bits were logical flags for each of the 2^m possible vertex numbers. For each facet exactly four of these bits are TRUE. These representations ("sets" in Pascal and other languages) may be directly compared for equality, avoiding the need to sort a list of vertex numbers. The maximum size of such a representation is compiler-specific and may limit its utility.

A.6. Facet Intersections

A facet is defined by three vectors m_i^*, m_{i-j}^*, and m_{i-k}^*. For some desired moment with direction $\hat{m}_d$, we wish to calculate the intersection of a line in the direction of $\hat{m}_d$ with the plane of the facet. At that intersection we have

$$a\hat{m}_d = m_i^* + bm_{i-j}^* + cm_{i-k}^*$$

or

$$\hat{m}_d = \frac{1}{a}[m_i^* + bm_{i-j}^* + cm_{i-k}^*]$$

$$= [m_i^* : m_{i-j}^* : m_{i-k}^*] \begin{Bmatrix} 1/a \\ b/a \\ c/a \end{Bmatrix}$$

The values of b and c must lie in (0, 1) for the intersection to be on the facet itself. If the intersection is on the facet, then the value of a is the magnitude of the maximum attainable moment in the direction $\hat{m}_d$. Since $\hat{m}_d$ is the variable in our problem, we compute and store for each facet the inverse

$$[m_i^* : m_{i-j}^* : m_{i-k}^*]^{-1}$$

from which we may then evaluate a, b, and c for a given $\hat{m}_d$.

Acknowledgment

A major portion of this work was conducted under NASA Research Cooperative Agreement NCC1-158, supervised by John V. Foster of NASA Langley Research Center.

References

[1]Snell, S. A., Enns, D. F., and Garrard, W. L., "Nonlinear Inversion Flight Control for a Supermaneuverable Aircraft," *Proceedings of the AIAA Guidance, Navigation, and Control Conference* (Portland, OR), AIAA, Washington, DC, 1990, pp. 808–817 (AIAA Paper 90-3406).

[2]Shaw, P. D., et al., "Design Methods for Integrated Control Systems," Air Force Wright Aeronautical Laboratories, AFWAL-TR-88-2061, June 1988.

[3]Lallman, F. J., "Relative Control Effectiveness Technique With Application to Airplane Control Coordination," NASA TP 2416, April 1985.

[4]Lallman, F. J., "Preliminary Design Study of a Lateral-Directional Control System Using Thrust Vectoring," NASA TM 86425, Nov. 1985.

[5]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, Edwards, CA, April 21–23, 1992.

[6]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, CA, April 21–23, 1992.

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

[8]Sedgewick, R., *Algorithms*, 2nd ed., Addison-Wesley, Reading, MA, 1988.

[9]Ficken, F. A., *Linear Transformations and Matrices*, Prentice-Hall, Englewood Cliffs, NJ, 1967.

[10]Brogan, W. L., *Modern Control Theory*, 2nd ed., Prentice-Hall, Englewood Cliffs, NJ, 1982, Chap. 5.