JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 25, No. 3, May–June 2002

Fast Implementation of Direct Allocation with Extension to Coplanar Controls

John A. M. Petersen* and Marc Bodson†
University of Utah, Salt Lake City, Utah 84112

The direct allocation method is considered for the control allocation problem. The original method assumed that every three columns of the controls effectiveness matrix were linearly independent. Here, the condition is relaxed, so that systems with coplanar controls can be considered. For fast online execution, an approach using spherical coordinates is also presented, and results of the implementation demonstrate improved performance over a sequential search. Linearized state-space models of a C-17 aircraft and of a tailless aircraft are used in the evaluation.

I. Introduction

TO increase the reliability of aircraft, configurations with a large number of actuators and control surfaces are advantageous. Reconfigurable control laws may be used to exploit all of the available control power despite failures and damages.[1,2] Control allocation is the problem of distributing the control requirements among multiple actuators to satisfy the desired objectives while accounting for the limited range of the actuators. Although solutions exist for the control allocation problem, an issue of current interest is that of the feasibility of their implementation on existing computers for aircraft with a large number of actuators.[3]

The direct allocation approach[4–6] is based on the concept of the attainable moment set (AMS), which is the set of all of the moment vectors that are achievable within the control constraints. The method of direct allocation allows one to achieve 100% of the AMS, whereas some other approaches such as daisy chaining, pseudoinverse, and generalized inverse solutions have been shown to achieve a smaller volume.[7]

In the direct allocation method, the moment vectors are assumed to be related to the controls through the linear transformation $\boldsymbol{m} = \boldsymbol{CBu}$, where $\boldsymbol{m}$ is the resultant moment, $\boldsymbol{u}$ is the set of controls, and $\boldsymbol{CB}$ is referred to as the controls effectiveness matrix. The original method for three moments developed by Durham[4–6] was restricted to systems in which any three columns of $\boldsymbol{CB}$ are linearly independent. For this case, the boundary of the AMS consists of parallelograms defined by pairs of controls varying between their limits. As it turns out, the control needed to produce any moment on the boundary of the AMS is unique. In the direct allocation method, moments lying inside the boundary of the AMS are obtained by scaling the controls required to produce a moment of maximum magnitude in the same direction. In a similar manner, moments lying outside the boundary are scaled down to the achievable values. Therefore, controls are always uniquely defined.

If the restriction on $\boldsymbol{CB}$ is not satisfied, then the boundary of the AMS is defined by polygons rather than parallelograms, and each facet is bounded by $2p$ sides, where p is the number of controls defining the polygonal facet. With more than two variables describing the facet, the solution is not always unique, even on the boundary of the AMS. Because this situation occurs when the effects of three or more controls are linearly dependent in a three-dimensional space, the terminology coplanar controls is introduced to refer explicitly to this case. Systems with coplanar controls are loosely called coplanar systems. The geometry of the AMS boundary is further described in the paper, and a possible choice for the selection of the control is proposed given the nonuniqueness properties.

Next, we consider that most of the computational burden in using the direct allocation method lies in finding the facet on the AMS boundary in the direction of the desired moment. Generally, computations may be split into offline and online computations. Offline computations are defined to be those that may be performed at the design stage or, in the case of a reconfigurable control law, at a slower rate than the normal sampling rate. Online computations are those that are required for the determination of the control input at every sampling instant. A significant portion of the computations may be performed offline in the direct allocation method and consist of the determination of the set of attainable moments. Online computations include the search for the facet in the attainable moment set that is aligned with the desired moment and the determination of the control input using appropriate scaling.

To reduce the online computations, a representation of the AMS in two-dimensional space, using spherical coordinates, is shown to be beneficial. The new method converts the AMS representation into a two-dimensional system, where special techniques can be used to accelerate the search. Two options are suggested for the implementation. The first method computes facet boundaries that are used online to eliminate rapidly a large number of facets from the search. The second method creates a two-dimensional array relating the spherical coordinates of the desired moment to a corresponding facet identifier. The appropriate facet is found online by table lookup, requiring no iterations and virtually no computations. The spherical methods are also developed for coplanar systems. Rather than using polygonal facets for the rapid search, a representation using multiple coplanar subfacets is considered. Examples used to illustrate the concepts proposed include a C-17 aircraft model with 16 actuators and an advanced tailless fighter model with 11 actuators.

II. Problem Statement

Consider the linearized aircraft model

$$\dot{\boldsymbol{x}} = \boldsymbol{Ax} + \boldsymbol{Bu}, \qquad \boldsymbol{y} = \boldsymbol{Cx} \tag{1}$$

where $\boldsymbol{x} \in R^5$, $\boldsymbol{u} \in R^n$, and $\boldsymbol{y} \in R^3$. The states of the aircraft are given by $\boldsymbol{x}$ and include the angle of attack, the pitch rate, the angle of sideslip, the roll rate, and the yaw rate. The output $\boldsymbol{y}$ contains the pitch rate, the roll rate, and the yaw rate. The control input $\boldsymbol{u}$ is constrained to limits

$$\boldsymbol{u}_{i,\min} \le \boldsymbol{u}_i \le \boldsymbol{u}_{i,\max} \qquad \text{for} \qquad i = 1, \ldots, n$$

The matrix $\boldsymbol{B}$ specifies the forces and moments generated by the actuators. These forces and moments are limited by the allowable range of control inputs. Because we are interested in controlling the output $\boldsymbol{y}$, we consider the derivative of $\boldsymbol{y}$, which is given by

$$\dot{\boldsymbol{y}} = \boldsymbol{CAx} + \boldsymbol{CBu} \tag{2}$$

Model reference control laws[8] and dynamic inversion control laws[9] allow one to specify the trajectories of the output of the system by

Received 30 June 2000; revision received 5 November 2001; accepted for publication 14 November 2001.

*Graduate Student, Department of Electrical Engineering.
†Professor of Electrical Engineering, Department of Electrical Engineering; bodson@ee.utah.edu. Senior Member AIAA.

selecting the value of the term $\boldsymbol{CBu}$ due to the control input. The direct allocation problem can be formally stated as follows: Given a desired vector $\boldsymbol{m}_d$, find the vector $\boldsymbol{u}$ such that $\boldsymbol{CBu}$ is closest to $\boldsymbol{m}_d$ in magnitude, with $\boldsymbol{u}$ satisfying the constraints and $\boldsymbol{CBu}$ proportional to $\boldsymbol{m}_d$.

In the original formulation of Durham,[4–6] the vector $\boldsymbol{m}_d$ was a desired moment. Here, the vector represents three desired rotational accelerations. We will nevertheless continue to refer to the set of achievable $\boldsymbol{CBu}$ as the AMS.

III. Attainable Moments and Direct Allocation

Initially, we make the following assumption (noncoplanar controls): Every 3×3 submatrix of $\boldsymbol{CB}$ is full rank. Under this assumption, the following properties are obtained.

A. Systems with Noncoplanar Controls

1. Properties of the AMS

The AMS is a convex polyhedron whose boundary is the image of the facets of the control space. A facet of the control space is defined as the set obtained by taking all but two controls at their limits and varying the two free controls within the limits. A two-dimensional facet in control space is rectangular. The projection of such a facet to moment space is a linear transformation resulting in a two-dimensional parallelogram in three-dimensional space. When every set of three columns of the $\boldsymbol{CB}$ matrix are linearly independent, every facet on the boundary of the AMS originates from a unique facet on the boundary of the control space. There are $2^{n-2}n!/[2!(n-2)!]$ facets in the control space. However, most of these facets map to the interior of the AMS, and the boundary of the AMS comprises only $n(n-1)$ facets.[6] The four corners of each facet of the AMS are called vertices, and the four sides are called edges. There are $n(n-1)+2$ vertices in the AMS.

2. Computation of the AMS

The boundary of the AMS is made of facets corresponding to all of the possible pairs of input variables. For each pair, there is a multitude of facets in the original control space, but only two of them map to the boundary of the AMS. They may be found by looking for the combination of the other controls that maximizes the distance between the two facets. Hereafter, we refer to one of the facets as a max facet and the other as a min facet. The collection of all of these pairs of facets then constitutes the boundary of the AMS.

To further explain the procedure, let $\boldsymbol{CB}$ be subdivided as

$$\boldsymbol{CB} = [\boldsymbol{cb}_1, \boldsymbol{cb}_2, \ldots, \boldsymbol{cb}_n]$$

where $\boldsymbol{cb}_i$ is a column vector and $\boldsymbol{m}_i = \boldsymbol{cb}_i \boldsymbol{u}_i$ is the moment vector corresponding to the single control $\boldsymbol{u}_i$. For a pair of controls $(\boldsymbol{u}_i, \boldsymbol{u}_j)$ $i \in \{1, \ldots, n-1\}$, $j \in \{i+1, \ldots, n\}$, let the normal to the plane of the facet $\boldsymbol{\eta}_{ij}$ be defined by taking the cross product of the two vectors defining the facet

$$\boldsymbol{\eta}_{ij} = \boldsymbol{cb}_i \times \boldsymbol{cb}_j \tag{3}$$

Then, the two farthest facets are determined through the two vectors

$$\boldsymbol{m}_{\max} = \sum_{k=1, k\neq i,j}^{n} \boldsymbol{\mu}_{k,\max} \tag{4}$$

where

$$\boldsymbol{\mu}_{k,\max} = \begin{cases} \boldsymbol{cb}_k \boldsymbol{u}_{k,\max} & \text{if} \quad (\boldsymbol{cb}_k)^T \boldsymbol{\eta}_{ij} > 0 \\ \boldsymbol{cb}_k \boldsymbol{u}_{k,\min} & \text{if} \quad (\boldsymbol{cb}_k)^T \boldsymbol{\eta}_{ij} < 0 \end{cases}$$

$$\boldsymbol{m}_{\min} = \sum_{k=1, k\neq i,j}^{n} \boldsymbol{\mu}_{k,\min} \tag{5}$$

where

$$\boldsymbol{\mu}_{k,\min} = \begin{cases} \boldsymbol{cb}_k \boldsymbol{u}_{k,\max} & \text{if} \quad (\boldsymbol{cb}_k)^T \boldsymbol{\eta}_{ij} < 0 \\ \boldsymbol{cb}_k \boldsymbol{u}_{k,\min} & \text{if} \quad (\boldsymbol{cb}_k)^T \boldsymbol{\eta}_{ij} > 0 \end{cases}$$

Note that the case where the vector product $(\boldsymbol{cb}_k)^T \boldsymbol{\eta}_{ij}$ is zero is impossible for noncoplanar systems by virtue of the assumption of linear independence of every three columns of the $\boldsymbol{CB}$ matrix. However, this case is possible with coplanar systems and is discussed in detail in Sec. III.B. In coding this procedure, it is convenient to store an array of flags indicating the control values (max, min, or free) associated with each facet. A facet may also be assigned a number to index the array.

The vertices of the two facets are determined by using the maximum and minimum values of the other two free controls. For instance, the vertices for the max facet are

$$\begin{aligned} \boldsymbol{v}_1 &= \boldsymbol{m}_{\max} + \boldsymbol{cb}_i \boldsymbol{u}_{i,\min} + \boldsymbol{cb}_j \boldsymbol{u}_{j,\min} \\ \boldsymbol{v}_2 &= \boldsymbol{m}_{\max} + \boldsymbol{cb}_i \boldsymbol{u}_{i,\min} + \boldsymbol{cb}_j \boldsymbol{u}_{j,\max} \\ \boldsymbol{v}_3 &= \boldsymbol{m}_{\max} + \boldsymbol{cb}_i \boldsymbol{u}_{i,\max} + \boldsymbol{cb}_j \boldsymbol{u}_{j,\min} \\ \boldsymbol{v}_4 &= \boldsymbol{m}_{\max} + \boldsymbol{cb}_i \boldsymbol{u}_{i,\max} + \boldsymbol{cb}_j \boldsymbol{u}_{j,\max} \end{aligned} \tag{6}$$

Figure 1 shows the results of this procedure for a C-17 aircraft model. A three-dimensional view of the boundary is shown. The

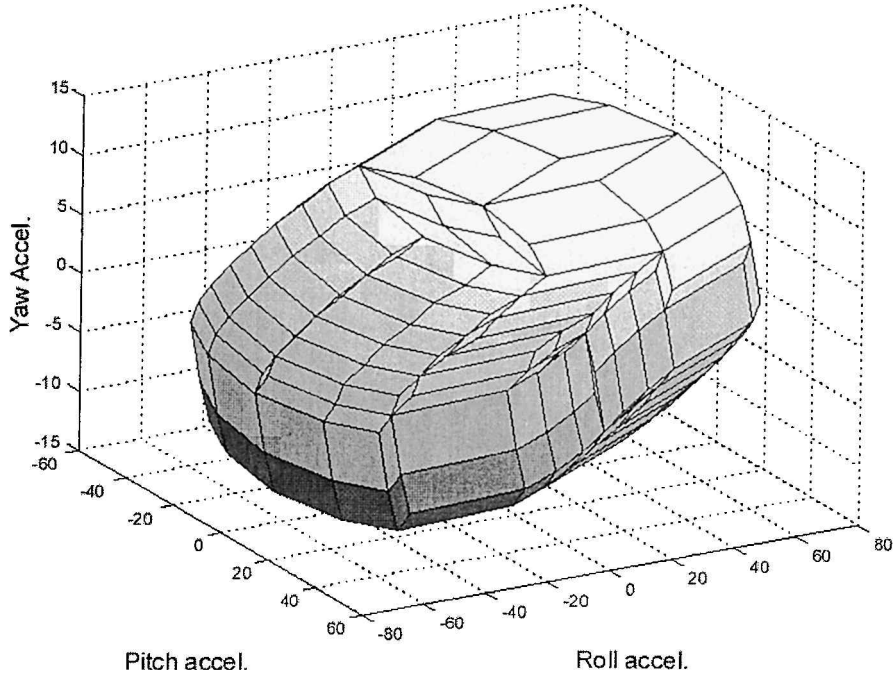

Fig. 1 Set of attainable moments for a C-17 model.

facets are shaded according to height in the yaw acceleration axis. The C-17 model includes 16 separately controlled surfaces: four elevators, two ailerons, two rudders, and eight spoilers. The set is delimited by 240 facets.

3. Computation of the Control Input

The control input is obtained by scaling the desired moment so that the scaled vector reaches the boundary of the AMS. On the boundary, there is a unique relationship between the moment and the value of the input needed to achieve it. If the desired moment is larger than the one attainable in the given direction, the moment vector is scaled to the achievable value. If the desired moment is smaller, the control input associated with the maximum attainable moment is scaled to obtain the desired moment.

The algorithm proceeds as follows. For a given facet, a basis spanning the moment space is formed by using the vector from the origin to one vertex of the facet and the vectors from this vertex to the two adjacent vertices. Let $\boldsymbol{m}_{\text{base}}$ be the vector to one of the vertices and $\Delta\boldsymbol{m}_i$ and $\Delta\boldsymbol{m}_j$ be the vectors from this vertex to the other two vertices.

Using $\boldsymbol{m}_d$ as the desired moment, we have

$$\boldsymbol{m}_{\text{base}} = \boldsymbol{v}_1, \qquad \Delta\boldsymbol{m}_i = \boldsymbol{v}_3 - \boldsymbol{v}_1 = \boldsymbol{cb}_i \Delta\boldsymbol{u}_i$$

$$\Delta\boldsymbol{m}_j = \boldsymbol{v}_2 - \boldsymbol{v}_1 = \boldsymbol{cb}_j \Delta\boldsymbol{u}_j$$

$$\rho_3 \boldsymbol{m}_d = \rho_1 \Delta\boldsymbol{m}_i + \rho_2 \Delta\boldsymbol{m}_j + \boldsymbol{m}_{\text{base}} \tag{7}$$

where $\Delta\boldsymbol{u}_i = (\boldsymbol{u}_{i,\max} - \boldsymbol{u}_{i,\min})$. The free parameters ρ_1, ρ_2, and ρ_3 are found by solving the 3×3 set of linear equations

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = [\Delta\boldsymbol{m}_i \quad \Delta\boldsymbol{m}_j \quad \boldsymbol{m}_{\text{base}}]^{-1} \boldsymbol{m}_d \tag{8a}$$

and by letting $\rho_1 = \lambda_1/\lambda_3$, $\rho_2 = \lambda_2/\lambda_3$, and $\rho_3 = 1/\lambda_3$. Figure 2 shows graphically the moment vector equation. The value $\rho_3 \boldsymbol{m}_d$ is the point at which the vector $\boldsymbol{m}_d$ intersects the facet. The values of (ρ_1, ρ_2, ρ_3) determine whether $\boldsymbol{m}_d$ intersects the facet. If

$$\left\{ \begin{array}{c} 0 < \rho_1 < 1 \\ 0 < \rho_2 < 1 \\ \rho_3 > 0 \end{array} \right\} \tag{8b}$$

then a vector in the direction of the desired moment intersects the facet defined by $\boldsymbol{m}_{\text{base}}$, $\Delta\boldsymbol{m}_i$, and $\Delta\boldsymbol{m}_j$.

The control vector at the boundary is

$$\boldsymbol{u}_{\text{boundary}} = \boldsymbol{u}_{\text{base}} + \rho_1 \Delta\boldsymbol{u}_i + \rho_2 \Delta\boldsymbol{u}_j \tag{9}$$

where $\boldsymbol{u}_{\text{base}}$, $\Delta\boldsymbol{u}_i$, and $\Delta\boldsymbol{u}_j$ are the sets of controls that determine $\boldsymbol{m}_{\text{base}}$, $\Delta\boldsymbol{m}_i$, and $\Delta\boldsymbol{m}_j$, respectively. Here, $\boldsymbol{u}_{\text{boundary}}$ is the control associated to the maximum moment in the direction of the desired moment and within the control constraints. If $\rho_3 < 1$, the desired moment exceeds the maximum available moment and $\boldsymbol{u}_{\text{boundary}}$ is taken to be the control. If $\rho_3 > 1$, the control is scaled to match the moment requirement, with $\boldsymbol{u} = \boldsymbol{u}_{\text{boundary}}/\rho_3$.

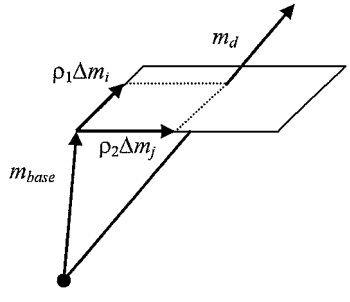

Fig. 2 Desired moment intersecting a facet, with basis vectors shown in relation to the facet.

4. Sequential Search for Direct Allocation

The computation of the control input involves the solution of a linear system of three equations in three unknowns and the linear combination of three input vectors. If the correct facet is used, the computations are minor, and the resulting control input satisfies the limits. If the incorrect facet is used, the values of (ρ_1, ρ_2, ρ_3) exceed their limits, and the control input will not satisfy the constraints. The computation may be used as a test of whether the facet is the correct one. If all of the facets are tested sequentially in this manner, the procedure may be used for control allocation. We will refer to this approach as the sequential search procedure.

The computations for this procedure may be separated into offline and online computations. The offline code creates a table containing the four vertices associated to each facet. The online code consists of retrieving the vertex data, computing the control, and checking its feasibility. Once the correct facet is encountered, computations stop. The search will be time consuming if the number of facets is large. The sequential search was nevertheless implemented to provide a baseline for the evaluation of the benefits of the methods proposed. More intelligent search techniques have been proposed,[5,10] but these were not implemented for this paper. Instead, the use of spherical coordinates is investigated to accelerate the search.

B. Systems with Coplanar Controls

1. Properties of the AMS

For systems with coplanar controls, a p-dimensional volume $(p \geq 2)$ in control space maps into a two-dimensional facet in moment space and has $2p$ sides. The facet becomes a polygon defined by p controls. Figure 3 shows the AMS for an advanced tailless fighter model.[11] According to Ref. 11, the output vector $\boldsymbol{y}$ is composed of modified rotational rates. Specifically, the components of $\boldsymbol{y}$ are the pitch rate, the stability axis roll rate, and a blend of sideslip and stability axis yaw rate.

The advanced tailless fighter model includes 11 separately controlled surfaces consisting of elevons, pitch flaps, thrust vectoring, outboard leading-edge flaps, spoiler slot deflectors, and all-moving tips. In this model, pitch thrust vectoring and the pitch flaps produce linearly dependent moments yielding coplanar controls with any third control variable. It was found that up to four control variables were coplanar $(2 \leq p \leq 4)$. Some polygonal facets are indeed clearly visible in Fig. 3. The boundary of the AMS is delimited by 78 such facets.

Polygonal facets can also be described by a set of subfacets that are the projections of the two-dimensional facets of the control space. Each two-dimensional facet is determined by two controls as in the noncoplanar case. For every pair of the p controls, there are 2^{p-2} identical subfacets offset from each other in the same plane. Because there are $p(p-1)/2$ pairs of controls, the total number of subfacets covering a polygonal facet is given by

$$n_{pf} = p(p-1)2^{p-3} \tag{10}$$

Figure 4 is a series of figures that details the relationship between a polygonal facet and its subfacets. The top diagram in Fig. 4 shows a polygonal facet. The middle diagram in Fig. 4 shows how the six subfacets cover the polygonal facet. The bottom diagram in Fig. 4 shows the three sets of subfacets in an exploded view. The representation of the AMS by subfacets has been found to be practical for the computation of the control input in the extension of the direct allocation method proposed here.

2. Computation of the AMS

The algorithm is similar to the one described in Sec. III.A.2. For every pair of controls $(\boldsymbol{u}_i, \boldsymbol{u}_j)$, $i \in \{1, \ldots, n-1\}$, $j \in \{i+1, \ldots, n\}$, each column of $\boldsymbol{CB}$, excluding columns i and j, is evaluated. Coplanar controls can be determined by the product $(\boldsymbol{cb}_k)^T \boldsymbol{\eta}_{ij}$, where $k \neq i, j$. If this product is zero, then the kth control is coplanar with the facet created by the control pair $(\boldsymbol{u}_i, \boldsymbol{u}_j)$. The noncoplanar controls (that correspond to the columns of $\boldsymbol{CB}$ that are linearly independent of columns i and j) are then selected to maximize the distance of the facet from the origin. As before, for each facet, there is an identical facet that lies on the opposite side of the AMS

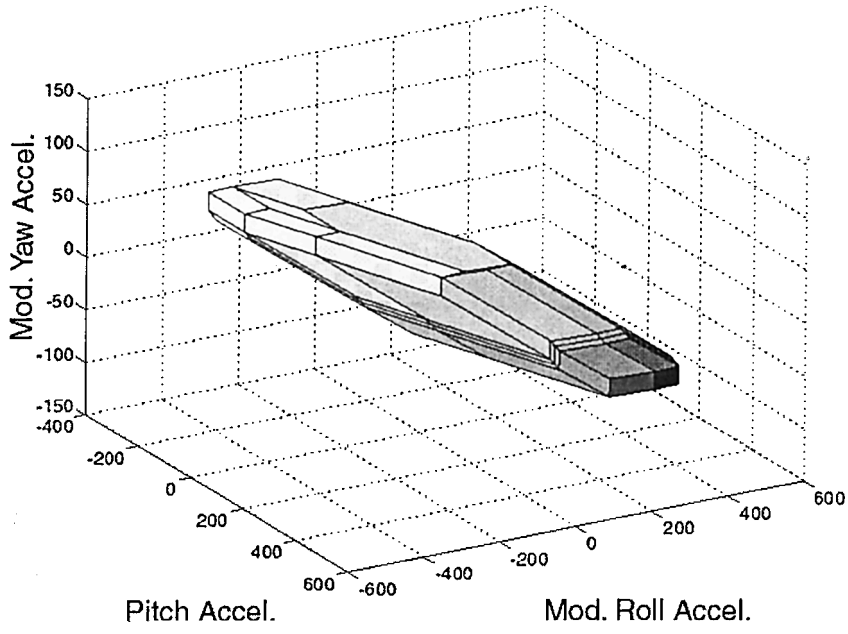

Fig. 3 Set of attainable moments for an advanced tailless fighter model.

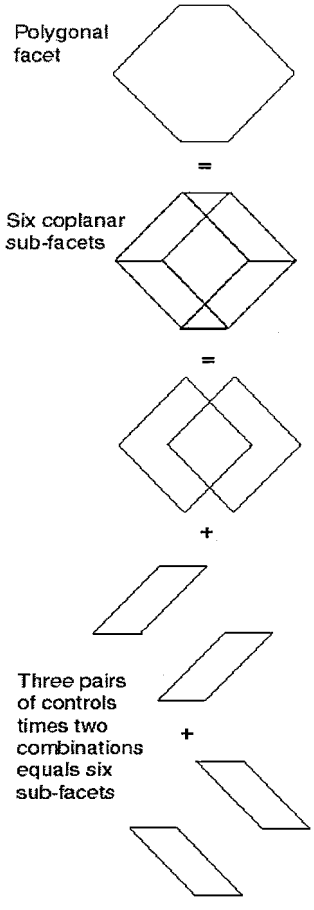

Fig. 4 View of the relationship between a parent polygonal facet and its subfacets.

boundary and can be found using the opposite values for each of the $n-p$ noncoplanar controls. When the set of indices

$$K = \{i,\ j,\ \text{and indices of all controls coplanar with } \boldsymbol{u}_i \text{ and } \boldsymbol{u}_j\}$$

is defined, the maximum displacement vector is

$$\boldsymbol{m}_{\max} = \sum_{\substack{k=1 \\ k \notin K}}^{n} \boldsymbol{\mu}_{k,\max} \tag{11}$$

where $\boldsymbol{\mu}_{k,\max}$ is defined in Eq. (4). The opposite facet is determined by the minimum displacement vector

$$\boldsymbol{m}_{\min} = \sum_{\substack{k=1 \\ k \notin K}}^{n} \boldsymbol{\mu}_{k,\min} \tag{12}$$

where $\boldsymbol{\mu}_{k,\min}$ is defined in Eq. (5).

Whereas the noncoplanar $n-p$ controls determine the distance of a polygonal facet from the origin, the remaining p controls determine the shape of the polygonal facet. Each polygonal facet is made up of $p(p-1)/2$ sets of $r = 2^{p-2}$ subfacets. Each subfacet lies in the same plane, but has a different offset that shifts it with respect to the other subfacets. The offset is determined by the sum of the r combinations of max and min control values of the coplanar controls. For every unordered pair of controls $\{(\boldsymbol{u}_a, \boldsymbol{u}_b) | a \in K, b \in K, a \neq b\}$, the offset of each subfacet is computed using the controls $\{\boldsymbol{u}_c | \boldsymbol{c} \in K, \boldsymbol{c} \notin \{a, b\}\}$ in different combinations of their upper and lower limits, resulting in

$$\text{offset}_q = \boldsymbol{c}\boldsymbol{b}_c \boldsymbol{u}_c \tag{13}$$

where $q \in \{1, \ldots, r\}$. Because $\boldsymbol{c}$ is a vector of $p-2$ indices, $\boldsymbol{c}\boldsymbol{b}_c$ in Eq. (13) is a matrix with $p-2$ columns.

Finally, subfacets are defined by vertices obtained by summing the displacement vector, the offset, and the four combinations of maximum and minimum values of the two free controls $\boldsymbol{u}_a$ and $\boldsymbol{u}_b$. For instance, the four vertices for the qth max subfacet are

$$\begin{aligned}
\boldsymbol{v}_{1,q} &= \boldsymbol{m}_{\max} + \boldsymbol{c}\boldsymbol{b}_a \boldsymbol{u}_{a,\min} + \boldsymbol{c}\boldsymbol{b}_b \boldsymbol{u}_{b,\min} + \text{offset}_q \\
\boldsymbol{v}_{2,q} &= \boldsymbol{m}_{\max} + \boldsymbol{c}\boldsymbol{b}_a \boldsymbol{u}_{a,\min} + \boldsymbol{c}\boldsymbol{b}_b \boldsymbol{u}_{b,\max} + \text{offset}_q \\
\boldsymbol{v}_{3,q} &= \boldsymbol{m}_{\max} + \boldsymbol{c}\boldsymbol{b}_a \boldsymbol{u}_{a,\max} + \boldsymbol{c}\boldsymbol{b}_b \boldsymbol{u}_{b,\min} + \text{offset}_q \\
\boldsymbol{v}_{4,q} &= \boldsymbol{m}_{\max} + \boldsymbol{c}\boldsymbol{b}_a \boldsymbol{u}_{a,\max} + \boldsymbol{c}\boldsymbol{b}_b \boldsymbol{u}_{b,\max} + \text{offset}_q
\end{aligned} \tag{14}$$

3. Computation of the Control Input

With all but two of the controls at their limits for a given subfacet, the control that will achieve a moment vector intersecting the subfacet can be defined as before. However, because overlapping subfacets may exist at a particular boundary point, there is not, in general, a unique relationship between the moment and the value of the input needed to achieve it. Whereas the solution corresponding to any subfacet could be taken as a solution to the direct allocation problem, we propose to take instead the average of the inputs resulting from all overlapping subfacets. Taking the average is a simple solution that gives the desired moment, usually reduces the number of saturated controls, and guarantees the continuity of the solution. Figure 5 helps one visualize the advantage of averaging the controls. Figure 5 shows two subfacets in control space that overlap in

moment space. Any point along the dashed line will result in the desired moment, but the midpoint will offer a control vector that yields an attractive compromise between control limits.

4. Sequential Search for Direct Allocation

The sequential search procedure described in Sec. III.A.4 may be employed for the search for the right subfacet. Although all subfacets containing the desired moment must be found, once a correct subfacet is encountered, one only needs to check the other subfacets lying in the same plane (i.e., those with the same parent polygonal facet) to complete the search. The sequential search procedure is useful as a baseline for evaluation.

IV. Rapid Search Using Spherical Coordinates

A. Systems with Noncoplanar Controls

1. Representation of the AMS in Spherical Coordinates

Because the determination of the applicable facet does not depend on the magnitude of the desired moment, the search may be performed in a two-dimensional space instead of the original three-dimensional space. Each vertex of the moment space, determined by (x, y, z) coordinates, can be expressed in spherical coordinates $(\theta, s\varphi, \rho)$, with

$$\theta = \tan^{-1}(y, x) \tag{15}$$

$$s\varphi = \sin(\varphi) = z/\rho \tag{16}$$

$$\rho = \sqrt{x^2 + y^2 + z^2} \tag{17}$$

where θ represents the azimuth angle (the horizontal angle in the x, y plane), φ represents the elevation angle (the vertical angle from the x, y plane), and ρ represents the distance from the origin. This third spherical coordinate is irrelevant for the search of the facet. For the azimuth, note that a two-argument inverse tangent function is used. The value $\sin(\varphi)$ (henceforth abbreviated $s\varphi$) is also used instead of φ to simplify online computations.

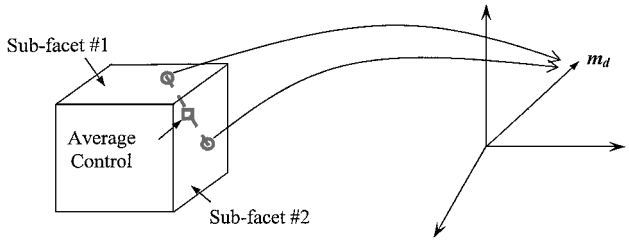

Fig. 5 Visualization of averaging the controls.

Figure 6 shows the result of transforming the boundary of the C-17 AMS to spherical coordinates, with θ shown on the x-axis in the range of $\pm\pi$. The sine of the elevation angle is shown on the y-axis. Figure 6 shows that the lines that form the edges of the facets become curves, because of the nonlinear change of coordinates. In fact, these curves are the well-known great circles used in navigation. They are the projection on the unit sphere of three-dimensional line segments, or the intersection of the unit sphere with a plane including the origin and the two vertices. The idea of using the spherical coordinates is that the desired moment is represented in the two-dimensional space as a point, and that the control allocation problem becomes the simpler problem of determining to which two-dimensional facet the point belongs.

Some terminology is introduced to help convey the algorithms. An edge joins two adjacent vertices of a facet. A crossing is defined as an edge between two vertices, with azimuth angles θ_1 and θ_2 crossing the $\pm\pi$ boundary. Existence of a crossing may be determined by testing

$$\delta\theta = \theta_{\max} - \theta_{\min} \tag{18}$$

with $\theta_{\max} = \max(\theta_1, \theta_2)$ and $\theta_{\min} = \min(\theta_1, \theta_2)$. If $\delta\theta > \pi$, one may conclude that the edge crosses the boundary. A split facet is one in which the facet is bisected by the line $\theta = \pm\pi$.

The great circle for each pair of vertices looks like a distorted sinusoid in the spherical coordinate space. The curve reaches maximum and minimum values of the elevation angle φ that have equal magnitude and opposite sign. These points occur 180 deg apart in azimuth angle. For the mapping of two vertices in three-dimensional to two vertices in spherical coordinate space $\{(x_1, y_1, z_1), (x_2, y_2, z_2)\} \rightarrow \{(\theta_1, s\varphi_1), (\theta_2, s\varphi_2)\}$, it turns out that one of the extrema of the great circle occurs at $(\theta_{\text{pk}}, s\varphi_{\text{pk}})$ given by

$$\theta_{\text{pk}} = \tan^{-1}(x_1 z_2 - x_2 z_1, y_2 z_1 - y_1 z_2) \tag{19}$$

$$s\varphi_{\text{pk}} = s\varphi_1 \Big/ \sqrt{s\varphi_1^2 + \left(1 - s\varphi_1^2\right)[\cos(\theta_{\text{pk}} - \theta_1)]^2} \tag{20}$$

To obtain the correct values, θ_{pk} needs to be adjusted by $\pm\pi$ using the following rule:

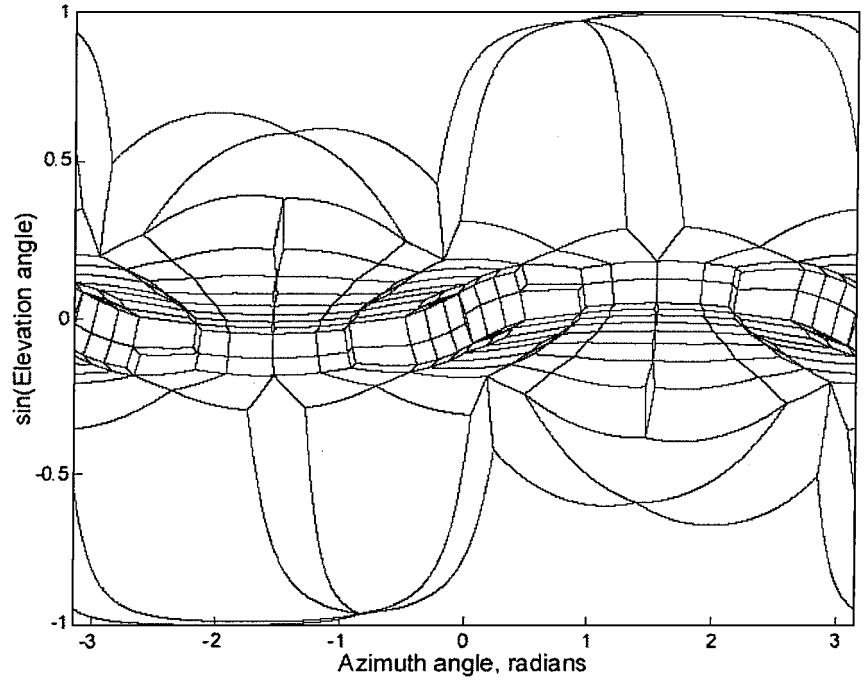

Fig. 6 Spherical mapping of C-17 AMS facets.

$$\text{if} \quad \cos(\theta_{pk} - \theta_1) < 0 \quad \text{then} \quad \theta_{pk} = \theta_{pk} - \text{sign}(\theta_{pk})\pi \tag{21}$$

The other extremum of the great circle is obtained by symmetry.

With the knowledge of the peaks of the great circle, the equations defining the great circle, that is, the edge of the facet under consideration, is given by

$$\cos(\varphi_{pk}) = 1 - s\varphi_{pk}^2, \qquad \alpha = s\varphi_{pk}\cos(\theta_{pk} - \theta_k)$$

$$s\varphi_k = \alpha \Big/ \sqrt{\cos(\varphi_{pk}) + \alpha^2} \tag{22}$$

where $(\theta_k, s\varphi_k)$ is a point on the great circle.

2. Rapid Search Using Facet Boxes

In the first option, ranges are computed for the coordinates of the facets, and they define boxes in which the facets are located. The boxes are used to assess quickly whether the desired moment is likely to lie within a given facet. The overall approach is similar to the exhaustive search, but simple inequality tests are used to drop facets from the list. For those facets that are left, the usual test of Eq. (8b) is performed. If the test is successful, the control input is quickly obtained. Otherwise, the search continues. The idea is that the test is required for very few facets.

a. Offline computations. Offline computations consist of the determination of the AMS and of the boxes that delimit the facets in spherical coordinates. For illustration, a facet box is outlined with a dashed line in Fig. 7. It shows that the box is determined not only by the coordinates of the vertices, but also by maxima reached within the edges of a facet. The peaks of the great circles are, therefore, determined using Eqs. (19–21), and their values are used in the computations of the box if the peaks lie between the vertices.

A difficulty with the implementation of the method is that facets may span the boundaries of the two-dimensional space. In particular, two facets include the north pole and the south pole. The north and south poles are the points with $\varphi = 90$ and -90 deg, respectively. Facets may also span the azimuth boundaries. Such facets could be split into two facets to perform box tests. Instead, however, facet types are defined, and those facets that span the azimuth boundary are redefined on the $(0, 2\pi)$ range so that they become contiguous. The procedure is then implemented in the following steps:

1) Compute AMS vertex coordinates and facets. The AMS is computed using the direct method as described earlier.

2) Compute spherical coordinates of all vertices.

3) Determine the facet type. Five types of facets are considered and are designated as types 0–4. The azimuth range is extended so that each facet is completely contained in at least one of the two ranges of θ: $-\pi < \theta < \pi$ and $0 < \theta < 2\pi$. A facet is assigned a label of type 0 (those facets in the first set) or type 2 (those facets in the second set, that is, split facets). Furthermore, three special cases are assigned: 1) facets that enclose a pole (type 1), 2) facets that border a pole and lie within $-\pi < \theta < \pi$ (type 3), and 3) facets that border a pole and lie within the range $0 < \theta < 2\pi$ (type 4). Facet type can be determined by testing the $\delta\theta$ of each facet edge. The value of $\delta\theta$ is categorized in four possible ranges describing four cases: 1) $\delta\theta < \pi$, 2) $\delta\theta > \pi$, 3) $\delta\theta = \pi$, and 4) $\delta\theta = 0$. A simple algorithm can be applied to determine the facet type based on the type and number of crossings, with

$$\text{facet type} = c + 3d \tag{23}$$

where $c \in \{0, 1, 2\}$ is the number of crossings and $d \in \{0, 1\}$ is the number of occurrences of case 3.

4) Determine box boundaries of spherical facets.

a) Compute the maximum and minimum spherical coordinates of the edges between vertices of each facet. If the facet is type 2 or type 4 (i.e., in the $0 < \theta < 2\pi$ range), negative values of θ_1, θ_2, and θ_{pk} of each edge are incremented by 2π before calculating $s\varphi_{pk}$.

b) Store the extremal values of θ and $\sin(\varphi)$ for each facet to define the facet box. For each facet, determine θ_{min}, θ_{max}, $s\varphi_{min}$, and $s\varphi_{max}$. Store these values in a facet box table. If the facet includes a north (south) pole, $s\varphi_{max}$ ($s\varphi_{min}$) is forced to its maximum (minimum) of 1 (-1). Distinguishing a north pole from a south pole can be done by calculating the great circle formed by any two of the facet vertices that form an edge and testing the location of a third facet vertex $(\theta_3, s\varphi_3)$ relative to this great circle. If $s\varphi_{gc}$ is computed from Eq. (22) with $\theta_k = \theta_3$, and if $s\varphi_3 > s\varphi_{gc}$, the facet is a north pole. Otherwise, it is a south pole.

5) Compute and store the 3×3 inverse of Eq. (8a) for each facet.

b. Online computations. Online computations are described in the following steps:

1) Convert the desired moment into spherical coordinates.

2) For each facet, check the feasibility of the desired moment. Compare the coordinate of the desired moment to the facet box. If the facet box is type 2 or type 4, add 2π to the azimuth of the point. If the desired moment lies within the box boundaries, compute (ρ_1, ρ_2, ρ_3) as shown in Sec. III.A.3. The rest of the controls are given by the control flags associated with the facet number.

3) Compute the control input. Once the correct facet is found and the test in Eq. (8b) is performed, one only needs to scale the control as necessary to satisfy the constraints.

3. Rapid Search Using Table Lookup

The second option consists in creating a lookup table $f(\theta, s\varphi)$ that gives the number of the facet associated with a given pair of spherical coordinates. If the azimuth and elevation angles are quantized with 1000 points each, this option requires an array with 1,000,000 values, a size that is large but within the reach of existing computers. The creation of the table is essentially the transcription of Fig. 6 into an array, and the marking of the elements of the array with the associated facet number.

The online computations could not be simpler with this approach: the facet toward which the desired moment points is found instantly by table lookup, and the appropriate control is determined with minor computations. Note that control allocation is guaranteed to be performed within a known and short period of time.

a. Offline computations: construction of the facet table. The steps to this method are as follows:

1) Compute AMS vertex coordinates and facets. The AMS is computed using the direct method as before. From this computation, one obtains coordinate information as well as knowledge of which vertices connect along an edge of the facet and the set of controls that form each facet.

2) Compute spherical coordinates of the vertices. For a vertex at (x, y, z), the spherical coordinates $(\theta, s\varphi, \rho)$ are obtained using Eqs. (15–17).

3) Compute and quantize the facet edges. Quantizing the edges is done by converting the end points to a range of index values in θ, computing corresponding values of $s\varphi$, and then converting these values to an appropriate index. Continuous edges in the range $-\pi < \theta < \pi$ are straightforward. Edges that cross the $\theta = \pm\pi$

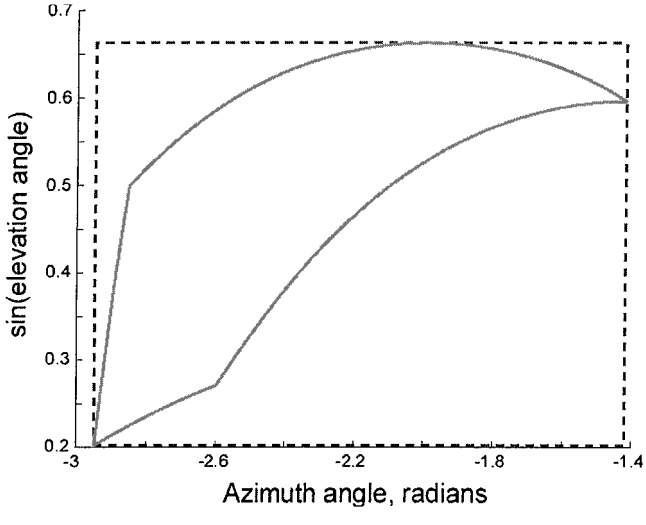

Fig. 7 Spherical mapping of a single facet outlined by a facet box.

boundary have $s\varphi$ indices that are not contiguous and must be managed properly. The indices of each edge of the facet are stored in a common array.

4) Modify pole facets. Facets containing a pole must be treated as special cases. A facet with one crossing, that is, one edge with $\delta\theta = \pi$ (corresponding to an edge that passes through a pole), contains a pole. Because the line $s\varphi$ is actually a single point in three dimensions, the edge $s\varphi = \pm 1$ (1 for north pole, -1 for south pole) for each θ index must be added to the common array of step 3.

5) Create a facet table. The facet table is a two-dimensional array of the facet numbers or identifiers. The indices represent quantized values of θ and $s\varphi$. When the common array is complete, the facet table is updated in the storage locations to which the facet indices just calculated correspond, including indices inside the boundary of the facet.

6) Compute and store the 3×3 inverse of Eq. (8a) for each facet.

b. Online computations. The online computations are described in the following steps:

1) Convert the desired moment into spherical coordinates and then to facet table indices.

2) Obtain the facet number from the lookup table.

3) Compute the control input. Compute (ρ_1, ρ_2, ρ_3) as shown in Sec. III.A.3 to arrive at the values for the free controls. The rest of the controls are given by the control flags associated with the facet number. Scale the control as necessary to satisfy the constraints.

4. C-17 Example

Each algorithm was tested with 1000 randomly selected desired moments for the C-17 example. The AMS has 240 facets and 242 vertices in moment space. The sequential search method was used to establish a baseline to evaluate the other search methods. The spherical facet table technique was simulated using quantizations of 100 and 1000 for each axis.

Figures 8 and 9 display comparisons of the number of floating point operations (obtained by using the flops command in MATLAB®) of the three algorithms. These histograms are intended only to provide a rough comparison of the algorithms described in this paper and should be used with caution because the results are dependent on the specific implementation as well as on the language and hardware used. Note, in particular, that MATLAB counts as floating pointing operations some operations that would normally be counted as integer operations. The code for each algorithm was written in MATLAB version 5.3.

The histogram in Fig. 9 shows a range and an average value for the online code of the sequential search and facet box methods. The range denotes the minimum to maximum possible values for the method, whereas the average was computed for the 1000 randomly sampled moments. Our simulations have shown that a sample size

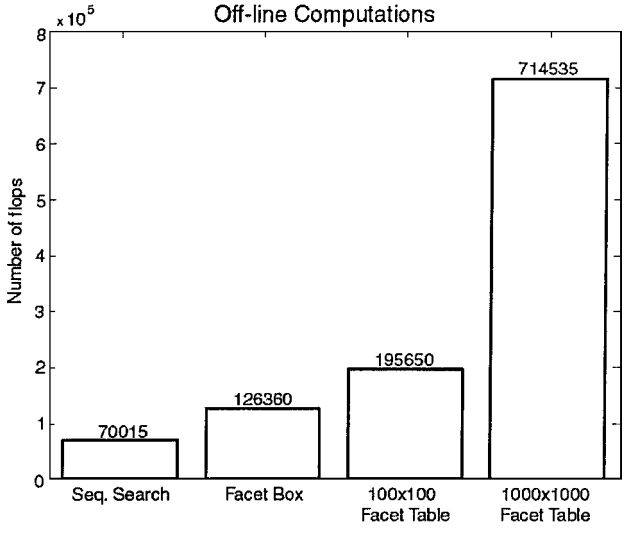

Fig. 8 Histogram showing comparison of offline computations for the different methods.

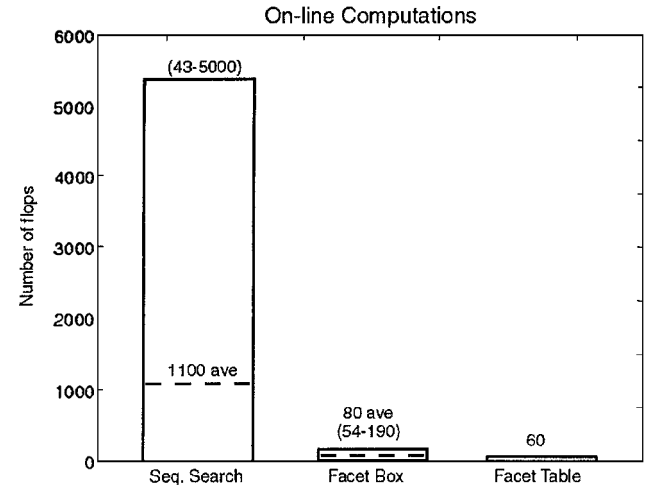

Fig. 9 Histogram showing comparison of online computations for the different methods.

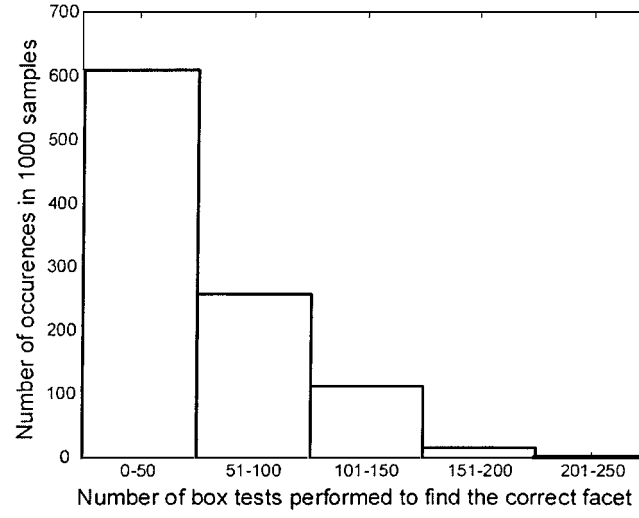

Fig. 10 Histogram of facet box tests performed.

of 1000 or more is typically large enough for the average numbers to be representative. One finds that the facet table approach considerably reduces the number of required computations to be performed online, and eliminates the variability in the number of those computations, making it ideal for real-time applications that demand reliable and predictable results. Memory requirements are significant, however. For memory rich systems requiring extremely fast operation, such as found in computer simulations, this approach is an attractive option. In an adaptive control application, the offline computations may also constitute an important burden to be considered. The facet box approach is a simple and useful intermediate option. The facet boxes may also be arranged using a hierarchical tree structure as is commonly done in range search techniques.[12] This may be advantageous for systems with a very large number of controls, but for systems with $n \leq 20$, the reduction in online processing is minimal and not considered advantageous enough to warrant its implementation.

Further analysis indicates some interesting characteristics of the facet box algorithm. Figure 10 gives a histogram of the number of box tests performed before the correct facet is found. The theoretical maximum is 240 in this example, but one finds that rarely more than 150 tests are required. The average number of tests was computed to be 49.7.

With the box test, the number of box tests passed is considerably less than the number of facets tested. Figure 11 shows that in nearly 50% of the cases, only one facet passed the box test. In 95% of the cases, a maximum of three facets passed box tests, and in none of the cases did more than five pass the test. The average number, for which

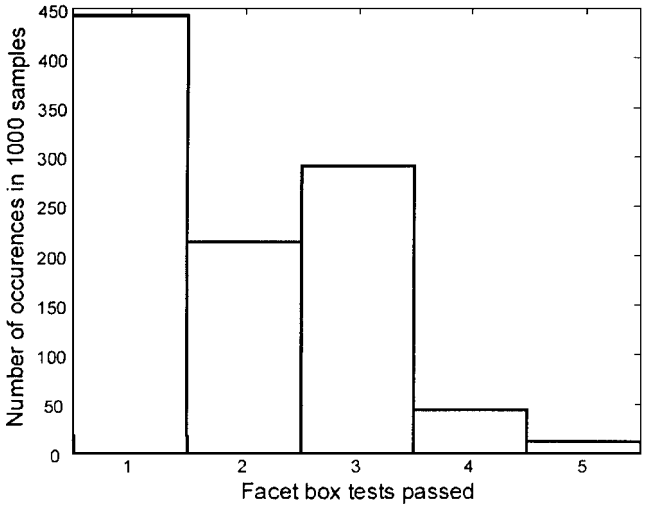

Fig. 11 Histogram of the facet box tests passed before finding the right facet.

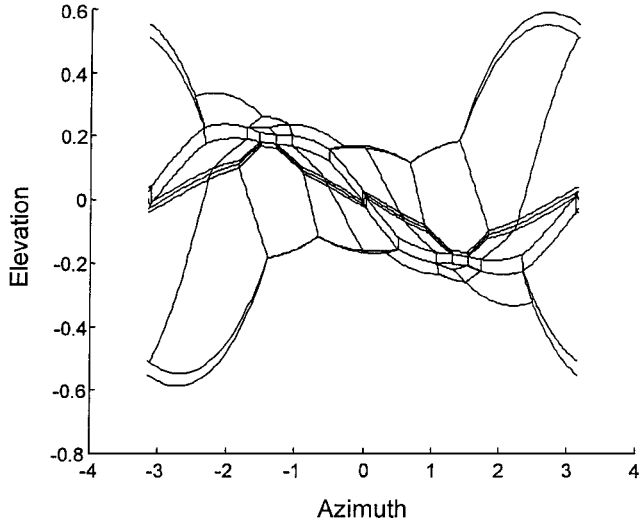

Fig. 12 Spherical mapping of overlapping facets outlined by facet boxes.

Eq. (8) must be computed, was found to be 2.0, to be compared with 49.7 required without the box test. Although the sequential search is viable, box tests in spherical coordinates make this approach much faster. The number of computations required, however, is not fixed.

B. Systems with Coplanar Controls

1. Representation of the AMS in Spherical Coordinates

Figure 12 shows the result of transforming the boundary of the advanced tailless fighter AMS to spherical coordinates. The transformed polygonal facets are visible. Although not shown, parallelogram subfacets are used to cover the area of every polygonal facet. Because the polygonal facets can be completely specified by subfacets, the same options as discussed in Secs. IV.A.2 and IV.A.3 can be used. However, modifications to the specific options are made here to extend its use to solve the problem of overlapping subfacets.

2. Rapid Search Using Facet Boxes

a. Offline computations. As in the original method, ranges are computed for the coordinates of the subfacets. Subfacets are treated no differently from noncoplanar facets. However, a methodology for tracking overlapping subfacets improves the online search. An array identifies a parent polygonal facet for each subfacet. Once a subfacet is found, only the other subfacets associated with the same parent are tested. For our example of the tailless fighter, which has 78 polygonal facets and 210 subfacets, the array has 210 rows and 78 identifiers. Figure 13 shows an illustration of overlapping subfacets outlined with facet boxes.

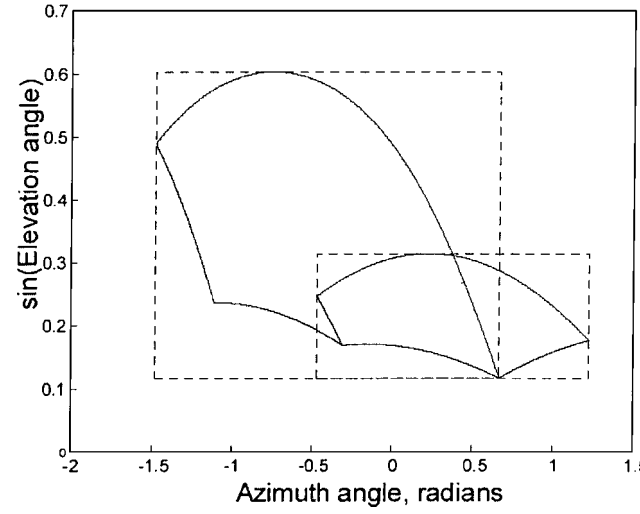

Fig. 13 Spherical mapping of overlapping subfacets outlined by facet boxes.

b. Online computations. The online computations are shown in the following steps:

1) Convert the desired moment into spherical coordinates.

2) For each subfacet, check the feasibility of the desired moment. Compare the coordinate of the desired moment to the facet box. If the desired moment lies within the box boundaries, compute the control for this subfacet. If the conditions on (ρ_1, ρ_2, ρ_3) are satisfied, check all other subfacets associated with the same parent facet. If not, go to the next subfacet.

3) Compute the control input. Once the correct subfacets are found and the tests of Eq. (8b) are performed, the applicable controls are averaged and scaled if the desired moment exceeds the achievable value.

3. Rapid Search Using Table Lookup

The creation of a spherical coordinate table for a system with coplanar controls can be done by including a third dimension. The added dimension is a vector of n_{pf} values corresponding to the subfacets that overlap at that location. Low quantization resolution may increase the apparent number of overlapping subfacets.

a. Offline computations: construction of the facet table. The facet table is a two-dimensional array of subfacet numbers or identifiers. Subfacet indices are recorded along a third dimension, if overlapping occurs. Again, the details of this method are identical to those explained in Sec. IV.A.3 with the modification of the vectorized facet identifiers in the table.

b. Online computations. The online computations consist of the following steps:

1) Convert the desired moment into spherical coordinates and then to indices of the facet table.

2) Obtain the array of facet numbers from the lookup table.

3) Compute the control input. Compute Eq. (8) for each subfacet in the array to arrive at the values for the free controls. Average the resulting controls of each subfacet. Scale the averaged control as necessary.

Again, the online computations are quite simple with the facet table approach: The subfacets toward which the desired moment points are found instantly by table lookup, and the appropriate control is determined with minor computations. Note that online computations could be even further reduced by converting the facet table into a control table. This would be done by computing the control for each facet table location in the offline code. The online code then becomes exclusively a lookup table to obtain the control. This method is ideal for applications where low resolution is acceptable or where simple and reliable online code is of premium importance.

4. Tailless Fighter Example

Each algorithm was tested with 1000 randomly selected desired moments and with the tailless fighter model. The AMS has 210

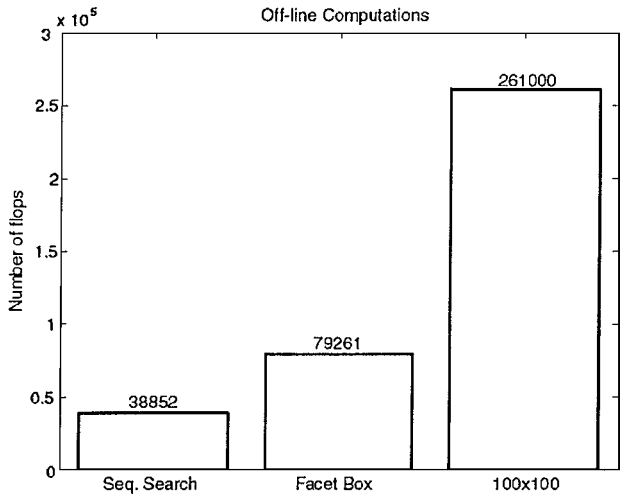

Fig. 14 Histogram showing comparison of offline computations for direct allocation methods for an 11-actuator coplanar system.

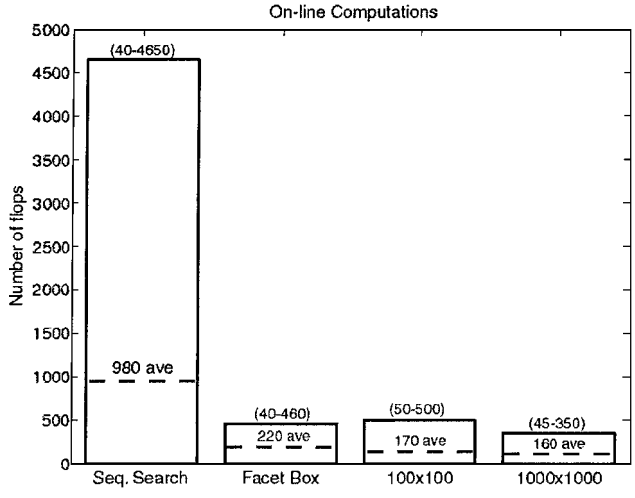

Fig. 15 Histogram showing comparison of online computations for direct allocation methods for an 11-actuator coplanar system.

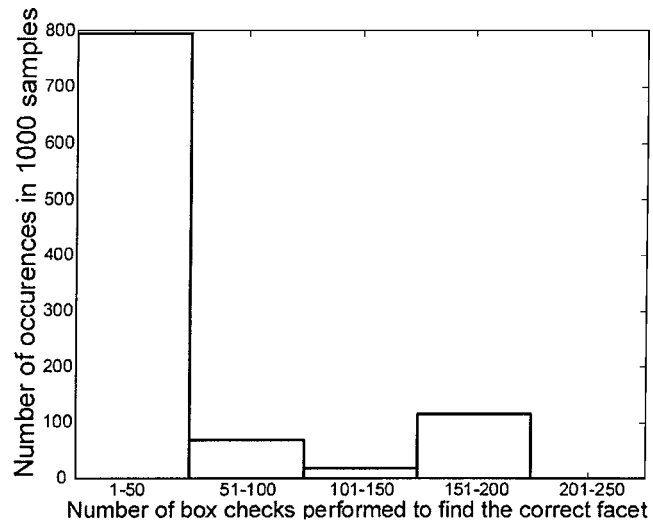

Fig. 16 Histogram of subfacets checked.

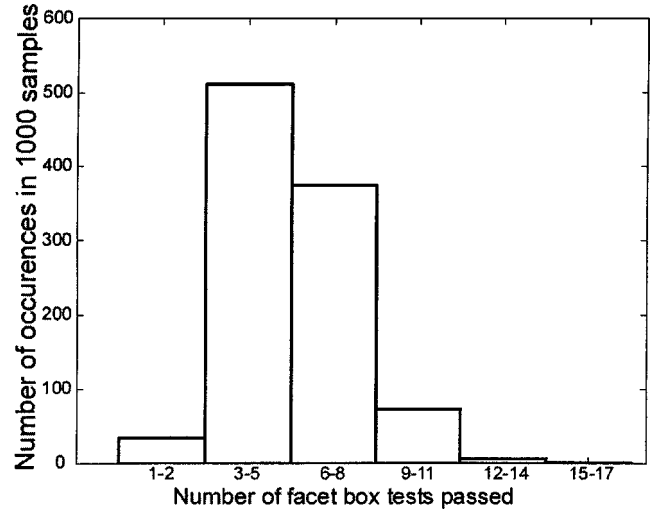

Fig. 17 Histogram of the number of facet box tests passed before finding the correct facet.

subfacets in moment space. The sequential search method was used to establish a baseline to evaluate the other search methods. The spherical facet table technique was implemented using quantizations of 100 and 1000 for each axis. Figure 14 displays a comparison of floating point operations for the offline computations. The number of computations for the 1000×1000 facet table was of the order of 10^7 flops and was not plotted with the others in the histograms.

Figure 15 shows the online computations. The histogram entries for the sequential search and facet box methods have a range and an average value. The range denotes the minimum to maximum possible values for the method, whereas the average was computed for the 1000 randomly sampled moments. As in the case for systems with noncoplanar controls, both spherical approaches considerably reduce the number of computations to be performed online and significantly reduce the variability in the number of those computations. The facet box approach is nearly equivalent to the facet table method in terms of online calculations and significantly cheaper in terms of offline computations.

The variation in online computations for the facet table method of different quantizations is due to variations in the number of apparent overlapping subfacets. Although two subfacets might not overlap, if the quantization is low enough, they effectively may. This results in more subfacets to be checked and averaged. Therefore, a higher quantization will typically yield slightly lower online computations.

Although the maximum number of box tests is 210, the average number is only 52. Figure 16 is a histogram of the number of box tests performed before the correct facet is found. Figure 17 shows a histogram of the number of box tests performed before a final control value is determined. In 92% of the cases, a maximum of 8 facets passed the test, and in none of the cases did more than 15 pass. The average number of tests of Eq. (8b) was computed to be 5.6, to be compared with 52 required without the box test. The number of computations required, however, is not fixed due to variability in the number of overlapping facets.

V. Conclusions

Direct allocation provides a solution to the control allocation problem that not only retains the direction of the desired moment, but takes advantage of the maximum attainable moment set. Direct allocation was previously only applicable to systems whose controls effectiveness matrix was such that every three columns were linearly independent. A system that is not limited in this way has been called a system with coplanar controls, or simply a coplanar system. The example of a tailless fighter aircraft model was shown to fall into this category.

The geometry of the attainable moment set for a coplanar system was explained, and an extension to the direct allocation method was given in this paper. The average of the multiple solutions was computed in the procedure, and the concept of overlapping subfacets was found useful for that purpose. Although the method of direct allocation is stressed, the shape of the AMS and nonuniqueness of the controls apply to any control allocation method.

The representation of the AMS in spherical coordinates makes it possible to perform rapidly the online computations required by the direct allocation method. No prior information is required about the approximate location of the correct facet. Two options were discussed, which have their respective advantages. The first option (facet box method) did not require large memory storage, but had a larger and variable number of online computations. The number of computations for a given control cycle will not exceed $n(n-1)$ box check comparisons (trivial) and a few three-dimensional vector–matrix multiplications. The number of these matrix operations is uncertain, but was found to not exceed five in our tests involving an aircraft model with 16 actuators.

The second option (facet table lookup method) required virtually no online computations and provided a guaranteed solution in a fixed time. The drawback was a potentially large memory requirement and longer offline execution time. Overall, both options provide a considerable improvement over a sequential search of the facets.

Slight modifications to the spherical coordinate methods were shown to provide direct allocation solutions to coplanar systems. The properties of those methods were similar to those for noncoplanar systems. Overlapping subfacets were identified by adding a third dimension to the table that stores the identifier of each subfacet.

The advantage of the method over ganging and other simple control allocation techniques is that it guarantees the use of the maximum control authority available, while at the same time requiring very few online computations. Compared to other rapid search techniques for direct allocation, its advantage is a high degree of predictability and reliability. It is also ideal for lengthy simulations because of the extremely fast execution time. However, the method requires a significant amount of memory and is not well suited to reconfigurable control, which would require continuous update of the lookup table in real time.

Acknowledgments

Effort sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, U.S. Air Force, under Grants F4962097-1-0405 and F49620-98-1-0013.

References

[1] Brinker, J., and Wise, K., "Reconfigurable Flight Control for a Tailless Advanced Fighter Aircraft," AIAA Paper 98-4107, Aug. 1998.

[2] Eberhardt, R., and Ward, D., "Indirect Adaptive Flight Control of a Tailless Fighter Aircraft," AIAA Paper 99-4042, Aug. 1999.

[3] Enns, D., "Control Allocation Approaches," AIAA Paper 98-4109, Aug. 1998.

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

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

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

[7] 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, 1995, pp. 1000–1007.

[8] Buffington, J., Chandler, P., and Pachter, M., "Integration of Online System Identification and Optimization-Based Control Allocation," AIAA Paper 98-4487, Aug. 1998.

[9] Bodson, M., and Pohlchuck, W., "Command Limiting in Reconfigurable Flight Control," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 4, 1998, pp. 727–736.

[10] Durham, W. C., "Computationally Efficient Control Allocation," AIAA Paper 99-4214, Aug. 1999.

[11] Buffington, J., "Tailless Aircraft Control Allocation," AIAA Paper 97-3605, Aug. 1997.

[12] Preparata, F. P., and Shamos, M. I., *Computational Geometry, An Introduction*, Springer-Verlag, New York, 1985, pp. 70–88.