Control Effector Unsaturation Modification to the Cascading Generalized Inverse Control Allocation Algorithm

Michael J. Acheson * and Irene Gregory†
NASA Langley Research Center, Hampton, VA, 23681

J.V.R Prasad ‡
Georgia Institute of Technology, Atlanta, GA, 30332

Control allocation has sufficiently progressed such that it is used in front-line fighter aircraft such as the F-18 Super Hornet and the F-35 Joint Strike Fighter. Published literature shows the F-35 utilizes Nonlinear Dynamic Inversion in conjunction with an Effector Blender that incorporates the Cascading Generalized Inverse control allocation algorithm. While the Cascading Generalized Inverse algorithm is one of the premier generalized inverse methods, it does suffer from three deficiencies. In particular, it suffers from an inability to achieve some desired outcomes, it intermittently provides non-optimal solutions and generally fails to preserve moment direction near maximal achievable moments. An effector unsaturation method based on a Scalar Difference Quadratic was first introduced and implemented on the iterative Prediction Method control allocation algorithm which was shown to consistently achieve optimal (weighted) control allocation solutions throughout the entire Attainable Moment Set while preserving desired moment direction. In this paper, the shortcomings of the Cascading Generalized Inverse algorithm are addressed by augmenting the baseline algorithm with Scalar Difference Quadratic unsaturation identification and location at each iteration. Numerical case studies demonstrate that the Modified Cascading Generalized Inverse algorithm resolves the aforementioned deficiencies.

Nomenclature

$\delta(\Omega)$	=	boundary of Allowable Control Set
$\delta(\Phi)$	=	boundary of Attainable Moment Set
Ω	=	Allowable Control Set (ACS), $\Omega \subset \Re^m$
Φ	=	Attainable Moment Set (AMS), $\Phi \subset \Re^n$

*NASA Researcher, Dynamics Systems and Control Branch, NASA Langley Research Center, Michael.J.Acheson@nasa.gov, AIAA member
†NASA Senior Technologist for Advanced Control Theory and Application, Dynamics Systems and Control Branch, NASA Langley Research Center, Irene.M.Gregory@nasa.gov, AIAA Associate Fellow
‡Professor, School of Aerospace Engineering, JVR.Prasad@aerospace.gatech.edu, AIAA Associate Fellow

B = control effectiveness matrix, $B \in \Re^{n \times m}$

$\hat{B}$ = orthogonal basis vector matrix containing $\mathcal{N}(B)$ basis vectors

$\vec{m}_{des}$ = desired moment, $\vec{m}_{des} \in \Re^n$

$\hat{m}_{des}$ = unit vector in the direction of desired moment, $\hat{m}_{des} \in \Re^n$

P_{gi} = Generalized inverse generated with positive (semi) definite weighting matrix W_{gi}, $P_{gi} \in \Re^{m \times n}$

P_{min} = Moore-Penrose generalized inverse of control effectiveness matrix B, $P_{min} \in \Re^{m \times n}$

$\mathcal{P}_{P_{min}}$ = Orthogonal projection operator onto Moore-Penrose surface

$\vec{r}_{lwr}$ = Lower bounds of control effector rate limitations, $\vec{r}_{lwr} \in \Re^m$

$\vec{r}_{upr}$ = Upper bounds of control effector rate limitations, $\vec{r}_{upr} \in \Re^m$

S_1 = Set of indices for all currently unsaturated control effectors, $S_1 = \{1, 2, \ldots, m\} \setminus S_2$

S_2 = Set of indices for all currently saturated control effectors (e.g. $S_2 = \{1, 5, 10\}$)

$\vec{s}_2$ = Vector of active control effectors position limits for all indices in S_2

$\hat{u}_{des}$ = Control unit vector in the direction of $P_{min}\vec{m}_{des}$, $\hat{u}_{des} \in \Re^m$

$\vec{u}_{lwr}$ = Lower bounds of control effector position limitations, $\vec{u}_{lwr} \in \Re^m$

$\vec{u}_{opt}$ = Optimal control allocation solution, $\vec{u}_{opt} \in \Re^m$

$\vec{u}_{upr}$ = Upper bounds of control effector position limitations, $\vec{u}_{upr} \in \Re^m$

W_{gi} = Positive (semi) definite weighting matrix for generation of arbitrary generalized inverse, $W_{gi} \in \Re^{m \times m}$

I. Introduction

Control allocation continues to make in-roads in the flight control systems of cutting-edge aerospace vehicles such as the F-35 Joint Strike Fighter and the F-18 Super Hornet aircraft series [1, 2]. The rationale for control allocation's rise in prominence in advanced flight control is multi-faceted. Some of control allocation's benefits include: enlarging the set of available control outcomes, increasing aircraft performance [2], achieving control objectives while simultaneously optimizing secondary objectives (e.g., minimal control cost), incorporating frequency apportioned control allocation [3, 4], incorporating structural load feedback [3, 5] and enhancing safety for control effector failures.

A wide array of control allocation algorithms are well documented in the literature. These algorithms can be broadly classified as generalized inverse [3, 6–9, 21], linear and quadratic programming approaches [10–13] and object searching [14, 15]. One of the preeminent algorithms in use is the Cascading Generalized Inverse (CGI) algorithm. This method uses a monotonic increasing set of saturated control effectors to produce a sequence of (cascading) generalized inverses designed to achieve the desired control outcome (also known as the desired moment in literature). The F-35 aircraft series uses nonlinear dynamic inversion, and an onboard nonlinear aircraft model coupled with an Effector Blender to great success in unifying aircraft control over multiple aircraft series and a wide range of flight conditions. The F-35

Effector Blender consists of a CGI control allocation algorithm [1, 3] which achieves desired control objectives while honoring effector position and rate limitations, providing gain limiting, and preventing structural feedback and overload.

A literature review [16–19] suggests four core requirements for an idealized linear control allocation algorithm. This idealized algorithm must provide optimal (weighted) cost for any desired outcome over the entire set of all attainable outcomes, be numerically stable and predictable, enable real-time execution for safety critical systems and readily accomodate changes in number of desired outcomes and available control effectors. The CGI algorithm satisfies the last three idealized algorithm requirements, but is known to fail to attain all desired outcomes and to always obtain the optimal (weighted) control allocation [3, 20].

Closely related to the CGI algorithm are the Prediction Method (PM) [8], the Affine Generalized Inverse (AGI) algorithm [9] and the Sequential Null Space Generalized Inverse (SNSGI) algorithm [21]. The PM utilizes a single weighted generalized inverse augmented with a sequence of null space components analytically generated from a partitioned null space basis matrix [8]. This algorithm inherently generates the null space objects without searching based on a monotonic increasing set of saturated effectors. The PM is guaranteed optimal allocation solutions throughout the AMS provided the null space component uses the boundary object nearest the generalized inverse component [8] and the algorithm yields optimal allocation solutions in the large majority of test cases. The PM otherwise satisfies the idealized core allocation algorithm and has been shown to be more computationally efficient than the CGI algorithm [21]. The AGI algorithm uses a sequence of generalized inverses (equivalent to those in CGI) and a sequence of null space components generated from the monotonic increasing sequence of saturated effectors. The AGI was shown to yield equivalent solutions to the PM for identical sequences of saturated effector sets [9]. The Appendix of this paper contains a rigorous proof showing CGI solutions are equivalent to those of the PM, and therefore by transitivity CGI solutions are equivalent to those of AGI.

Effector unsaturation, or the process of allowing a previously saturated effector to unsaturate, was introduced into the PM and was identified as the source of the PM's failure to obtain weighted optimal allocation solutions for some desired outcomes [21]. The Scalar Difference Quadratic (SDQ) was shown to be a functional evaluation to identify and locate instances of effector unsaturation. The SNSGI algorithm incorporated the SDQ into the PM with a limited object search consisting of one null space object search for each saturated effector per iteration. The incorporation of effector unsaturation processing enabled to SNSGI to satisfy all four idealized requirements [21].

The focus of this research was to incorporate effector unsaturation into the CGI algorithm to enable it to achieve optimal allocation solutions for all desired outcomes. In subsequent sections of this paper, the Modified Cascading Generalized Inverse (MCGI) algorithm is introduced and evaluated for allocation performance and computational efficiency.

II. Background

For systems with redundant controls (i.e. more effectors than outcomes), the control allocation problem seeks the optimal control $\vec{u}$ to the under-determined linear system $B\vec{u} = \vec{m}_{des}$ for a given control effectiveness matrix B and desired outcome $\vec{m}_{des}$ (e.g., accelerations, forces or moments). The desired outcome $\vec{m}_{des}$ is used interchangeably with desired moment in literature. Typically, the effectors are subject to upper and lower position limitations ($\vec{u}_{lwr}$ and $\vec{u}_{upr}$) and rate limitations ($\vec{r}_{lwr}$ and $\vec{r}_{upr}$). The effector position limits define the Allowable Control Set (ACS or Ω) which defines the corresponding Attainable Moment Set (AMS or Φ) respectively as:

$$\Omega = \left\{\vec{u} \in \Re^m \mid \vec{u}_{lwr_i} \leq \vec{u}_i \leq \vec{u}_{upr_i},\ \forall i \in \{1, \ldots, m\}\right\} \tag{1}$$

$$\Phi = \{\vec{m} \in \Re^n \mid B\vec{u} = \vec{m}, \vec{u} \in \Omega\}\,. \tag{2}$$

The optimal allocation problem for a given $\vec{m}_{des}$ is:

$$\vec{u}_{opt} = \underset{\vec{u}}{\operatorname{argmin}} \left\{\vec{u}^T W_{gi} \vec{u} \mid \vec{m}_{des} = B\vec{u},\ \vec{u} \in \Omega\right\} \tag{3}$$

for a positive (semi) definite effector weighting matrix $W_{gi} > 0$ ($W_{gi} \geq 0$) and control effectiveness matrix B. In this research, specifying W_{gi} as the identity matrix is known as Moore-Penrose (MP) control allocation or minimal control problem. The generalized inverses used to solve Eq. (3) for the weighted (W_{gi}) and minimal control problem ($W_{gi} = I$) are obtain:

$$P_{gi} = W_{gi}^{-1} B^T \left(BW_{gi}^{-1} B^T\right)^{-1} \tag{4}$$

$$P_{min} = B^T \left(BB^T\right)^{-1} \tag{5}$$

It is known that a single generalized inverse only yields allocation solutions within the ACS (i.e., $\vec{u}_{opt} \in \Omega$) over a strict subset of the AMS [6]. For this reason, iterative generalized inverse methods use a sequence of generalized inverses in order to enlarge the subset of the AMS that yields admissible control allocation solutions. For a given moment $\vec{m}_{des}$, we define the desired moment direction vector as:

$$\hat{m}_{des} = \vec{m}_{des} / \|\vec{m}_{des}\|_2 \tag{6}$$

Examination of weighted optimal allocation solutions along a given $\hat{m}_{des}$ show that these solutions are piecewise linear and continuous. The generalized inverse algorithms and null space object searching methods in this research iteratively solve for each piecewise segment along $\hat{m}_{des}$ until the desired moment is achieved. Each iteration for a linear segment uses sets of unsaturated and saturated effector indices. Therefore, for each iteration we define the set notation of saturated effector indices S_2 and the saturated effector position limit vector $\vec{s}_2$ as:

$$S_2 = \{i \in 1, 2, \ldots, m | \, u_{opt_i} \in \{u_{lwr_i}, u_{upr_i}\}\} \tag{7}$$

$$\vec{s}_2 = \begin{bmatrix} \ldots, u_{opt_i}, \ldots \end{bmatrix}^T \; \forall i \in S_2 \tag{8}$$

Similarly the unsaturated effector indices set S_1 is:

$$S_1 = \{i \in 1, 2, \ldots, m | \, u_{opt_i} \notin \{u_{lwr_i}, u_{upr_i}\}\} \tag{9}$$

such that $S_1 \cup S_2 = \{1, 2, \ldots, m\}$ and $S_1 \cap S_2 = \emptyset$. Furthermore, these effector indices sets are used to reorder associated quantities into unsaturated and saturated components. The columns of the control effectiveness matrix B are separated according to the effector index sets S_1 and S_2 into:

$$B = \begin{bmatrix} B_1 & B_2 \end{bmatrix} \tag{10}$$

Similarly, $\hat{B}$ is a basis for the null space of B or $\mathcal{N}(B)$ whose rows are separated into the unsaturated and saturated components respectively:

$$\hat{B} = \begin{bmatrix} \hat{B}_1 \\ \hat{B}_2 \end{bmatrix} \tag{11}$$

In [8], it was shown that the general form of the minimal control solution for a given control effectiveness matrix B and moment $\vec{m}_{des}$ is:

$$\vec{u}_{opt} = \vec{u}_{\perp} + \vec{u}_{\parallel} \in \Omega \tag{12}$$

where $\vec{u}_{\parallel} = P_{min}\vec{m}_{des}$ is the unique vector that yields the desired moment $\vec{m}_{des} = B\vec{u}_{\parallel}$ [8]. Conversely, $\vec{u}_{\perp} \in \mathcal{N}(B)$ is the null-space component so $B\vec{u}_{\perp} = \vec{0}$. Optimal solutions to Eq. (12) fall in two categories. If $\vec{u}_{\parallel} \in \Omega$ then $\vec{u}_{\perp} = \vec{0}$ and $\vec{u}_{opt} = \vec{u}_{\parallel} = P_{min}\vec{m}_{des}$. Conversely if $\vec{u}_{\parallel} \notin \Omega$ then computation of minimal $\vec{u}_{\perp}$ is required. Since $\vec{u}_{\parallel}$ is orthogonal to $\vec{u}_{\perp}$, then the 2-norm of the allocation solution of Eq. (12) simplifies to:

$$\left\|\vec{u}_{opt}\right\|_2^2 = \|\vec{u}_{\perp}\|_2^2 + \left\|\vec{u}_{\parallel}\right\|_2^2 \tag{13}$$

Thus to minimize $\vec{u}_{opt}$ with $\vec{u}_{\parallel}$ fixed to yield $\vec{m}_{des}$, it is necessary and sufficient to minimize $\|\vec{u}_{\perp}\|_2$ such that $\vec{u}_{\parallel} + \vec{u}_{\perp} \in \Omega$.

Recalling from [8] the general form of the PM algorithm optimal control allocation solutions for a piecewise linear segment:

$$\vec{u}_{opt} = \vec{u}_{\perp} + \vec{u}_{\|} = (\vec{c}_0 + \vec{c}_1) + P_{min}\vec{m}_{des}, \text{ where} \tag{14}$$

$$\vec{c}_0 \coloneqq \hat{B}\hat{B}_2^T \left(\hat{B}_2\hat{B}_2^T\right)^{-1} \vec{s}_2 \in \mathcal{N}(B) \tag{15}$$

$$\vec{c}_1 \coloneqq -\hat{B}\hat{B}_2^T \left(\hat{B}_2\hat{B}_2^T\right)^{-1} P_{min_2}\vec{m}_{des} \in \mathcal{N}(B) \tag{16}$$

$$P_{min_2} = B_2^T \left(B_1 B_1^T + B_2 B_2^T\right)^{-1} \tag{17}$$

where $\vec{u}_{\perp} = \vec{c}_0 + \vec{c}_1$ is the analytic solution to minimize $\|\vec{u}_{\perp}\|$. The PM algorithm uses a sequence of boundary objects (one for each piecewise linear solution segment) as moment magnitude increases for $\vec{m}_{des}$ along $\hat{m}_{des}$ to generate the minimal null-space component $\vec{u}_{\perp}$ for the minimal control solution $\vec{u}_{opt} = \vec{u}_{\|} + \vec{u}_{\perp} \in \Omega$ such that $\vec{m}_{des} = B\vec{u}_{\|}$. Equations (15,16) show that the null space component $\vec{u}_{\perp} = \vec{c}_0 + \vec{c}_1$ is strictly prescribed by the set of saturated effectors S_2 for the current piecewise linear segment. Unlike object-searching allocation algorithms that typically require searching all boundary objects under some selection criterion, the baseline Prediction Method method inherently generates the next boundary object as moment magnitude increases along $\hat{m}_{des}$ by following each piecewise linear segment to determine which effector saturates next. This next effector is added to the set of saturated effectors S_2 which is then used to generate the null-space component for the next piecewise linear segment of the minimal control solution. Following a similar line of development in [9], the AGI minimal control allocation solution for a piecewise linear solution segment was:

$$\vec{u}_{opt} = \vec{c}_0 + P_{aff}\vec{m}_{des} \tag{18}$$

$$\vec{c}_0 \coloneqq \hat{B}\hat{B}_2^T \left(\hat{B}_2\hat{B}_2^T\right)^{-1} \vec{s}_2 \in \mathcal{N}(B) \tag{19}$$

$$P_{aff} = \begin{bmatrix} P_{aff1} \\ P_{aff2} \end{bmatrix} = \begin{bmatrix} B_1^T \\ 0 \end{bmatrix} \left(\begin{bmatrix} B_1 & 0 \end{bmatrix} \begin{bmatrix} B_1^T \\ 0 \end{bmatrix} \right)^{-1} = \begin{bmatrix} B_1^T \left(B_1 B_1^T\right)^{-1} \\ 0 \end{bmatrix} \tag{20}$$

where P_{aff} of Eq. (20) is obtained by setting the columns of B corresponding to the saturated effectors (i.e., B_2) to $\vec{0}$.

The baseline PM method of sequentially determining boundary objects without searching has been shown to yield the correct optimal minimal control solutions in a large majority of $\vec{m}_{des}$ numerical test cases. However, it was shown in [21], that the baseline PM algorithm must allow a previously saturated effector to unsaturate (come off an upper or lower limit) to enable generation of minimal control optimal solutions for all $\vec{m}_{des} \in \Omega$. As a result, the Sequential Null Space Generalized Inverse (SNSGI) algorithm was introduced in [21]. It incorporated the Scalar Difference Quadratic (SDQ) evaluation into a null space object search routine with the baseline PM algorithm.

The null space object search routine for the SNSGI algorithm is performed for as many piecewise linear segments of the control solutions as necessary to yield the desired $\vec{m}_{des}$. For each linear segment, a series of boundary objects are generated from all strict saturation subset $S_{\underline{2}}$ combinations which differ from S_2 by one effector (i.e., $S_{\underline{2}} \subset S_2$). For example, if there are seven saturated effectors in S_2 for a given linear solution segment, then seven null space objects are searched for the strict subsets combinations of S_2. Comparison of the magnitudes of the associated allocation solutions $\vec{u} = \vec{u}_{\|} + \vec{u}_{\perp}$ for saturated effectors set S_2 and $\underline{\vec{u}} = \vec{u}_{\|} + \underline{\vec{u}}_{\perp}$ for saturation subset $S_{\underline{2}}$ was shown to only require comparison of the magnitudes of the null-space components. The SDQ was introduced as a simple null space component magnitude comparison function which both identifies and locates unsaturation occurrences. From [21], the SDQ is the scalar function:

$$\mathcal{F} = \left\|\vec{c}_{\underline{0}} + \vec{c}_{\underline{1}}\right\|_2^2 - \|\vec{c}_0 + \vec{c}_1\|_2^2 \tag{21}$$

Noting that $\vec{u}_{\|} = P_{min}\vec{m}_{des}$ can be expressed as a scaled unit vector where:

$$\hat{u}_{des} = \left(P_{min}\vec{m}_{des}\right) / \|P_{min}\vec{m}_{des}\|_2 \implies \tag{22}$$

$$P_{min}\vec{m}_{des} = a\hat{u}_{des}, \text{ where } a = \|P_{min}\vec{m}_{des}\|_2 \tag{23}$$

Now substituting only the saturated components of $a\hat{u}_2 = P_{min_2}\vec{m}_{des}$ into Eq. (16) and simplifying yields the SDQ as a function of unit vector scaling a:

$$\mathcal{F}(a) = \left\|\vec{c}_{\underline{0}} + \vec{c}_{\underline{1}}\right\|_2^2 - \|\vec{c}_0 + \vec{c}_1\|_2^2 = c_a a^2 + c_b a + c_c \in \Re \tag{24}$$

$$c_a = \left(\hat{u}_{\underline{2}}^T \left(\hat{B}_{\underline{2}}\hat{B}_{\underline{2}}^T\right)^{-1} \hat{u}_{\underline{2}} - \hat{u}_2^T \left(\hat{B}_2\hat{B}_2^T\right)^{-1} \hat{u}_2\right) \tag{25}$$

$$c_b = -2\left(\vec{s}_{\underline{2}}^T \left(\hat{B}_{\underline{2}}\hat{B}_{\underline{2}}^T\right)^{-1} \hat{u}_{\underline{2}} - \vec{s}_2^T \left(\hat{B}_2\hat{B}_2^T\right)^{-1} \hat{u}_2\right) \tag{26}$$

$$c_c = \vec{s}_{\underline{2}}^T \left(\hat{B}_{\underline{2}}\hat{B}_{\underline{2}}^T\right)^{-1} \vec{s}_{\underline{2}} - \vec{s}_2^T \left(\hat{B}_2\hat{B}_2^T\right)^{-1} \vec{s}_2 \tag{27}$$

The zero a_z of the SDQ coincides with the maximum of the quadratic where:

$$\partial\mathcal{F}(a) / \partial a = 0 \implies a_z = -c_b/2c_a \tag{28}$$

For the corner or equivalently each i_{th} iteration (for $i > 1$), the SNSGI algorithm creates i saturation subsets $S_{\underline{2}}$ by iteratively removing each effector of the current iteration's saturation indices set S_2^i to create $S_{\underline{2}} \subset S_2^i$ and performs the SDQ evaluation. It was noted in [21] that for the i_{th} piecewise linear allocation solution segment whose endpoints are denoted by the scaled unit vector gains $[a_i, a_{i+1}]$ (where $a_i = \|P_{min}\vec{m}_i\|$), that unsaturation is identified (occurrence necessary to maintain minimal control solutions) if $a_z \in (a_i, a_{i+1}]$. Moreover, the unsaturation occurrence is located at a_z within the solution segment.

In the event that unsaturation(s) occur, then at each linear solution segment corner, an unsaturation that occurs at the lowest moment magnitude is utilized to generate the next corner. Unsaturations for multiple effectors are treated as sequential unsaturations over very small changes in moment magnitude. Desired moments near $\vec{m}_{max}$, for $B \in \Re^{n\times m}$, may require up to $(m-n)$ null-space iterations to achieve the desired moment with a total of $N = \sum_{i=2}^{(m-n)} i$ SDQ unsaturation evaluations.

III. Modified Cascading Generalized Inverse Algorithm

In this section, the Cascading Generalized Inverse Algorithm as described in [3, 20] is discussed. Next a CGI specific Scalar Difference Quadratic is outlined. Implementing the SDQ into the CGI is discussed with a null space object search implementation for the i_{th} iteration where $1 < i \leq (m-n)$.

The following legacy Cascading Generalized Inverse algorithm iteration equations delineate the total moment, delta moment, total control and delta control vectors respectively for the i_{th} iteration for a given sequence of unsaturated and saturated effector indices sets [3].

$$\vec{m}_i = B\vec{u}_i \tag{29}$$

$$\vec{u}_i = \sum_{p=1}^{i} \Delta\vec{u}_p \tag{30}$$

$$\Delta\vec{u}_i = \left(k_i P_{T_i}\right)\Delta\vec{m}_i \tag{31}$$

$$\Delta\vec{m}_i = \vec{m}_{des} - \vec{m}_{i-1} \tag{32}$$

The vectors $\vec{m}_i$ and $\vec{u}_i$ are the moment and control allocation solution at the i_{th} corner or linear segment. P_{T_i} contains the generalized inverse P_{t_i} for the i_{th} iteration based on the number of unsaturated effectors in that iteration. Defining q as the number of saturated effectors, then the number of unsaturated effectors is $m-q$ where m is the number of effectors in $B \in \Re^{n\times m}$. If $m-q > n$ the P_{t_i} is the weighted generalized inverse computed using B_1. If $m-q = n$, then $P_{t_i} = B_1^{-1}$. Finally, if $m-q < n$ then P_{t_i} is the weighted least squares solution. Specifically from [3], the generalized inverse is found using:

$$P_{T_i} = \begin{bmatrix} P_{t_i} \in \Re^{(m-q)\times n} \\ \\ 0 \in \Re^{q\times n} \end{bmatrix} \quad P_{t_i} = \begin{cases} \left(W_p^T\right)^{-1} B_1^T \left(B_1 \left(W_P^T\right)^{-1} B_1^T\right)^{-1} & \text{for } (m-q) > n \\ B_1^{-1} & \text{for } (m-q) = n \\ \left(B_1^T W_d^T B_1\right)^{-1} B_1^T W_d^T & \text{for } (m-q) < n \end{cases} \tag{33}$$

The nominal version of the CGI algorithm applied to the minimal control problem starts using P_{min} to obtain $\Delta\vec{u}_1$. If any effectors in $\Delta\vec{u}_1$ exceed a effector position limit then a scaling factor k_1 is determined which brings the worst effector position to within limits. If no limits are exceeded then $k_1 = 1$. In [3], the authors utilize the scaling on P_{t_1}, however the scaling could equally be considered as scaling $\Delta\vec{m}_1$ as $\Delta\vec{u}_i = \left(k_i P_{T_i}\right)\Delta\vec{m}_i = P_{T_i}\left(k_i\Delta\vec{m}_i\right)$. In this fashion, the 1st scaling factor k_1 yields the moment along $\hat{m}_{des}$ associated with the end (or corner) of the first piecewise linear segment. Subsequent corners are obtained in a similar fashion with $\vec{m}_i$ yielding the moment at the i_{th} segment corner. Expressed in terms of Eqs. (30, 31) the minimal scaling factor k_i for the i_{th} iteration is found using:

$$k_i = \underset{k}{\operatorname{argmin}}\{k_j | \, j \in S_1\} \text{ where} \tag{34}$$

$$k_j = \begin{cases} \left(\vec{u}_{upr}(j) - \vec{u}_{i-1}(j)\right) / \Delta\vec{u}_i(j) & \left(\vec{u}_{i-1}(j) + \Delta\vec{u}_i(j)\right) > \vec{u}_{upr}(j) \\ \left(\vec{u}_{lwr}(j) - \vec{u}_{i-1}(j)\right) / \Delta\vec{u}_i(j) & \left(\vec{u}_{i-1}(j) + \Delta\vec{u}_i(j)\right) < \vec{u}_{lwr}(j) \end{cases} \tag{35}$$

Now implementation of unsaturation identification and location requires using the corner moment magnitudes $\vec{m}_i$ for comparison to the zero a_z of the SDQ. Since the SDQ is expressed as a function of the gain a for the unit vector $\hat{u}_{des}$, then applying the projection operator $\mathcal{P}_{B^T}$ (projects onto the subspace P_{min} containing $\vec{u}_{des}$) to the total control vector $\vec{u}_i$ at each corner retains only the $\vec{u}_{\|}$ component of the optimal allocation solution $\vec{u}_i = \vec{u}_{\|} + \vec{u}_{\perp}$. Then we obtain the unit vector gain for the i_{th} corner from the magnitude of $\vec{u}_{\|}$ using the corner moments $\vec{m}_i$ as:

$$a_i = \left\|\mathcal{P}_{P_{min}}\left(\vec{u}_i\right)\right\|_2 = \|P_{min} B\vec{u}_i\|_2 = \|P_{min}\vec{m}_i\|_2 \tag{36}$$

Next the mathematics of computing the zero a_z of the SDQ has options. The SDQ could be computed using Eqs.(25, 26, 28) but this broadly speaking requires computation of $\left(\hat{B}_2\hat{B}_2^T\right)^{-1}$ and $\left(\hat{B}_{\underline{2}}\hat{B}_{\underline{2}}^T\right)^{-1}$ which is an additional computational burden. However, the use of partitioned matrices in [21] substantially reduced this computational burden and eliminated the need to compute $\left(\hat{B}_{\underline{2}}\hat{B}_{\underline{2}}^T\right)^{-1}$ for a_z evaluation through the use of backward matrix partitioning. Alternatively, the zero of the SDQ could be computed using only P_{t_i} generated from the saturation set S_2 and a second generalized inverse $P_{t_{\underline{i}}}$ generated from the strict saturation subset $S_{\underline{2}}$ where $S_{\underline{2}} \subset S_2$. Specifically, using Eq. (71) from the Appendix of this paper then for the nominal CGI algorithm we have:

$$\begin{aligned} \vec{u}_{\perp_i} = \vec{c}_{0_i} + \vec{c}_{1_i} &= \left(P_{T_i} - P_{min}\right)\vec{m}_{des} + \left(I - P_{T_i}B\right)\sum_{p=1}^{i-1}\Delta\vec{u}_p \\ &= \left(P_{T_i} - P_{min}\right)\vec{m}_{des} + \left(I - P_{T_i}B\right)\vec{u}_{i-1} \end{aligned} \tag{37}$$

Recognizing that Eq. (37) is applicable to any moment $\vec{m}$ on the linear segment between $\vec{m}_i$ and $\vec{m}_{i+1}$, then expressing this moment as $\vec{m} = aB\hat{u}_{des}$ and substituting $\vec{m}$ in for $\vec{m}_{des}$ in Eq. (37) enables the computation of the Scalar Difference

Quadratic which after simplification yields:

$$\mathcal{F}(a) = \left\|\vec{c}_{\underline{0}} + \vec{c}_{\underline{1}}\right\|_2^2 - \|\vec{c}_0 + \vec{c}_1\|_2^2 = c_a a^2 + c_b a + c_c \in \mathfrak{R} \tag{38}$$

$$c_a = \hat{u}_{des}^T B^T \left(P_{\underline{T_i}}^T P_{\underline{T_i}} - P_{T_i}^T P_{T_i}\right) B\hat{u}_{des} \tag{39}$$

$$c_b = 2\hat{u}_{des}^T B^T \left(P_{\underline{T_i}}^T \left(I - P_{\underline{T_i}} B\right) - P_{T_i}^T \left(I - P_{T_i} B\right)\right) \vec{u}_{i-1} \tag{40}$$

$$c_c = \vec{u}_{i-1}^T \left(\left(P_{T_i} B\right)^T \left(I - P_{T_i} B\right) - \left(P_{\underline{T_i}} B\right)^T \left(I - P_{\underline{T_i}} B\right) + \left(P_{T_i} - P_{\underline{T_i}}\right) B\right) \vec{u}_{i-1} \tag{41}$$

$$\text{where } a_z = -c_b/2c_a \tag{42}$$

For each corner (starting at the $i = 2$), the MCGI algorithm creates i saturation subsets $S_{\underline{2}}$ by iteratively removing one effector to create $S_{\underline{2}} \subset S_2^i$ and performs the SDQ evaluation. As shown in [21], unsaturation is identified and located if $a_z \in (a_i, a_{i+1}]$. At each linear solution segment corner, the unsaturation that occurs at the lowest moment magnitude is utilized to generate the next corner by terminating the current linear segment at a_z, updating the saturation sets S_1, S_2 and resuming the CGI algorithm of Eqs. (29-31). Unsaturations for multiple effectors occur as sequential unsaturations over very small changes in moment magnitude. Desired moments near $\vec{m}_{max}$, for $B \in \mathfrak{R}^{n\times m}$, may require up to $(m-n)$ null-space iterations to achieve the desired moment with a total of $N = \sum_{i=2}^{(m-n)} i$ SDQ unsaturation evaluations.

IV. Numerical Results

The performance of the Modified Cascading Generalized Inverse (MCGI) was evaluated against the Prediction Method, the Sequential Null Space Generalized Inverse, and the Cascading Generalized Inverse for the minimal control allocation problem in three ways. First the computational efficiency of the MCGI algorithm was compared to the other algorithms. The second evaluation examined each algorithm's ability to produce the desired moments throughout the AMS and to produce the associated optimal effector allocation. Finally, the impact of effector unsaturation is highlighted by comparing MCGI performance against the CGI algorithm for a single unsaturation case.

Algorithm timing was performed for a series of three control effectiveness matrices with increasing numbers of desired moments and effectors which yielded an increasing computational burden. The series of control effectiveness matrices were: $B_A \in \mathfrak{R}^{3\times10}$, $B_B \in \mathfrak{R}^{6\times20}$, and $B_C \in \mathfrak{R}^{9\times30}$. Equation (45) is the linearized control effectiveness matrix B_A for the NASA F-18 High Alpha Research Vehicle (HARV) [22] at an altitude of 10,000 ft, Mach = 0.3 and AoA = 12.5 deg. The B_B and B_C control effectiveness matrices were randomly generated robust rank matrices. The matrix B_B is a representative example of a modern aircraft as it provides six outcomes (e.g., three moments and three forces) for 20 effectors. The third matrix B_C is designed to challenge the algorithms and demonstrates solving for a larger under-determined linear system which is applicable to more fields than just aircraft control allocation. The impact of increasing the number of effectors and the number of desired outcomes challenged the algorithms in two ways.

Increasing the number of effectors m results in a general increase in the number of iterations required for a given $\vec{m}_{des}$. For example, moments near the boundary of the AMS may require $(m - n)$ iterations for $B \in \Re^{n\times m}$ to yield the desired moment. Conversely, increasing the number of desired moments n, raises the computational burden at each iteration to find the required matrix inverses. Increasing both m and n substantially increased the overall computational burden.

The computational timing comparisons utilized Matlab® m-files auto-coded into Matlab executables (mex) for all algorithms. The PM and SNSGI algorithms are developmental Matlab algorithms that produce the optimal control solution but also provide ancillary information such as the moments, control allocation and saturated control effectors at each iteration, as well as effector unsaturation and the inverse matrix condition number information [8, 21]. The CGI m-file is a basic algorithm which only provides the effector allocation and it is available online as a supplement to [23]. The online CGI algorithm was modified to enable auto-coding and to add LU matrix factorization required for inverse computation when $n > 3$ in B_B and B_C. Tables 1 and 2 show the computation time comparisons of B_A for interior moments and boundary moments respectively. These tables include the Prediction Method without partitioned inverses (NPI) and with the partitioned inverses (PI) where the partitioned inverses were shown in [21] to substantially increase computational efficiency. The tables present the SNSGI results without use of partitioned inverses (NPI) and with partitioned inverses (PI) but also evaluate SNSGI with partitioned inverse plus ill-conditioned inverse handling (PI, IC). Lastly, these tables present results for the legacy CGI algorithm and the MCGI. Tables 3 and 4 similarly evaluate the each algorithm's computational timing but with combined boundary and interior moment cases for B_B and B_C respectively. Figure 1 shows the mean computation timing comparisons of all algorithms for B_A, B_B, and B_C. Figure 1 shows a modest computational increase for unsaturation handling of MCGI over CGI for cases B_B and B_C. However, the MCGI exhibits lower computational cost than CGI for B_A cases, but this is due to the MCGI algorithm utilizing vector computations while the legacy CGI code uses individual vector components. Figure 1 shows the computational benefits of uses matrix inverse partitioning over both MCGI and CGI with increasing computational burden.

Next all algorithms were evaluated for moment and optimal allocation generation capability. For B_A, two groups of one thousand moment test cases each were selected with one group on the boundary of the AMS and the other group in the AMS interior. The boundary test cases were desirable since the use of robust rank control effectiveness matrices B guaranteed the existence of a unique optimal control allocation solution $\vec{u}_{opt}$ for each maximal boundary moment $\vec{m}_{max}$. The Direct Allocation (DA) algorithm is known to yield these unique allocation solution for boundary moments and moreover a Matlab® Direct Allocation m-file is available within the online supplement from [23] for the case of $n = 3$ (i.e B_A). Therefore, the online Direct Allocation supplement was used as the reference source for control allocation validation for the boundary moment test cases of B_A. The interior cases don't have known closed form unique allocation solutions, so control allocation validation was performed by using the SNSGI (PI, CI) control allocation solutions as the comparison reference with other algorithms. Moment validation evaluated each algorithm's ability to generate the desired boundary or interior moments respectively.

A moment error metric E_{m_i} of Eq. (43), compares an algorithm's moment output $\vec{m}_{alg_i}$ to desired moment $\vec{m}_{des_i}$ for each case i. Similarly, the signed control magnitude error metric E_{c_i} of Eq. (44), compares each algorithm's control output $\vec{u}_{alg_i}$ to a control reference $\vec{u}_{ref_i}$ where the reference was the unique DA boundary solution [6] and SNSGI (PI, IC) for the AMS interior.

$$E_{m_i} = \left\|\vec{m}_{des_i} - \vec{m}_{alg_i}\right\|_2 / \left\|\vec{m}_{des_i}\right\|_2 \tag{43}$$

$$E_{c_i} = \left(\left\|\vec{u}_{alg_i}\right\|_2 - \left\|\vec{u}_{ref_i}\right\|_2\right) / \left\|\vec{u}_{ref_i}\right\|_2 \tag{44}$$

Additionally, a binary (i.e., true or false) moment validation metric V_{m_i} determined if an algorithm's moment solution $\vec{m}_{alg_i}$ solution was equivalent to the desired moment $\vec{m}_{des_i}$ within a component-wise tolerance of 1e-10. Similarly, a control validation metric V_{u_i} compared each algorithm's allocation solution $\vec{u}_{alg_i}$ to the reference control solution $\vec{u}_{ref_i}$ within the component-wise tolerance 1e-10 (with $\vec{u}_{ref_i}$ chosen as $\vec{u}_{DA}$ for boundary cases and $\vec{u}_{SNSGI}$ for interior cases). The error metric statistics, mean (μ_m, μ_u) and standard deviation (σ_m, σ_u), for the boundary and interior cases of B_A are shown in performance Tables 5 and 6 respectively. The values of V_m and V_u (in red) denote the percentage of failed validation cases whereas V_m and V_u (in black) are successful cases for algorithms.

For matrices B_B and B_C, one thousand moment cases were generated by choosing one thousand random control vector permutations which each consisted of some combination of all effectors saturated at either an upper or lower limit. It is known [6], that control vectors $\vec{u}_{des_i}$ on the boundary of the ACS, when converted to moments $\vec{m}_{des_i} = B\vec{u}_{des_i}$, are either in the interior or on the boundary of the AMS. Therefore the one-thousand numeric test case for B_B and B_C contain both interior and boundary moments. Control allocation validation was performed using the SNSGI (PI, CI) control allocation solutions as the comparison reference with other algorithms. Tables 5-8, clearly show that effector unsaturation handling in SNSGI and MCGI result in substantially improved ability to achieve both the desired moment and minimal control solution over the algorithms without unsaturation handling (PM and CGI).

The upper plot of Fig. (2) shows the AMS for Eq. (45) and a desired moment series line of non-dimensionalized aerodynamic moment coefficients from the origin to $\vec{m}_{des}^T = \begin{bmatrix} 0.053 & 0.761 & 0.1 \end{bmatrix}$. The red portion of this line delineates those moments after unsaturation of effector six occurs. The lower subplot shows the control magnitude (%) reduction $(\|\vec{u}_{CGI}\|_2 - \|\vec{u}_{MCGI}\|_2) / \|\vec{u}_{MCGI}\|_2$ achieved for post-unsaturation moments only. MCGI is seen to have achieved an $\approx 6\%$ maximum control magnitude reduction over CGI.

$$B^T = \begin{bmatrix} -4.382E{-2} & -0.5330E{-2} & +1.100E{-2} \\ +4.382E{-2} & -0.5330E{-2} & -1.100E{-2} \\ -5.841E{-2} & -6.486E{-2} & +3.911E{-2} \\ +5.841E{-2} & -6.486E{-2} & -3.911E{-2} \\ +1.674E{-2} & +0.000E{+0} & -7.428E{-2} \\ -6.280E{-2} & +6.234E{-2} & +0.000E{+0} \\ +6.280E{-2} & +6.234E{-2} & +0.000E{+0} \\ +2.920E{-2} & +1.000E{-5} & +3.000E{-4} \\ +1.000E{-2} & +0.3353E{+0} & +1.000E{-5} \\ +1.000E{-2} & +1.000E{-5} & +1.485E{-1} \end{bmatrix}, \ \vec{u}_{upr} = \begin{bmatrix} 0.183 \\ 0.183 \\ 0.524 \\ 0.524 \\ 0.524 \\ 0.785 \\ 0.785 \\ 0.524 \\ 0.524 \\ 0.524 \end{bmatrix}, \ \vec{u}_{lwr} = \begin{bmatrix} -0.183 \\ -0.183 \\ -0.524 \\ -0.524 \\ -0.524 \\ -0.785 \\ -0.785 \\ -0.524 \\ -0.524 \\ -0.524 \end{bmatrix} \tag{45}$$

V. Conclusion

Analysis of the legacy CGI algorithm has demonstrated that it yields erroneous minimal control allocation solutions, fails to preserve moment direction and/or is unable to obtain desired moment for those cases which seek moment magnitudes greater than that at which unsaturation occurs along a specified moment direction $\hat{m}_{des}$. Often these cases occur relatively close to the maximal moment capability of the aircraft. Physically, situations which require near maximal moment generating capability are seldom utilized in nominal flight conditions, but when they do occur the ability to obtain these large moments may be critical. Examples of situations requiring maximal control moments are: the ability to recover from unusual attitudes (e.g. inverted spin), maintaining control during engine failure or during maximum performance maneuvers (e.g. low slow flight or high alpha maneuvers).

This research applied the Scalar Difference Quadratic for unsaturation identification and location to the legacy Cascading Generalized Inverse method for Control Allocation. The ability to correctly identify and locate unsaturation resolved the three issues that prevented the CGI algorithm from obtaining optimal control allocation solutions throughout the entire Attainable Moment Set. Numerical case studies demonstrated that the Modified Cascading Generalized Inverse is now able to satisfy the idealized allocation algorithm requirements at a modest computational cost.

Table 1 Computation Time (μs) for Interior Moments, $B_A \in \Re^{3\times 10}$, 1000 Cases

Algorithm	Mean	Std Dev	Max	Min
PM (NPI)	+74.18	+31.66	+210.2	+50.25
PM (PI)	+72.59	+17.65	+148.1	+54.2
SNSGI (NPI)	+109.5	+119.6	+938.1	+48.99
SNSGI (PI)	+82.89	+47.59	+342.2	+48.04
SNSGI (PI, IC)	+81.96	+47.48	+360.5	+47.53
CGI	+59.94	+12.17	+121.3	+44.84
MCGI	+47.74	+13.75	+119.1	+35.05

Table 2 Computation Time (μs) for Boundary Moments, $B_A \in \Re^{3\times 10}$, 1000 Cases

Algorithm	Mean	Std Dev	Max	Min
PM (NPI)	+83.03	+38.24	+257	+51.79
PM (PI)	+79.03	+20	+164.4	+56.29
SNSGI (NPI)	+112.6	+121.8	+897.6	+49.34
SNSGI (PI)	+85.36	+49.08	+348.9	+48.39
SNSGI (PI, IC)	+82.31	+47.57	+361.8	+47.72
CGI	+81.08	+18.75	+149	+52.85
MCGI	+53.55	+15.26	+149	+37.18

Table 3 Computation Time (μs) for Boundary and Interior Moments, $B_B \in \Re^{6\times 20}$, 1000 Cases

Algorithm	Mean	Std Dev	Max	Min
PM (NPI)	+149.6	+73.23	+408.4	+52.1
PM (PI)	+107.3	+32.2	+214.1	+54.75
SNSGI (NPI)	+530.9	+726.2	+7956	+49.54
SNSGI (PI)	+255.8	+277.6	+2314	+52.85
SNSGI (PI, IC)	+261.9	+347.1	+5087	+52.58
CGI	+158.1	+39.05	+344.8	+73.97
MCGI	+320.3	+283.5	+1792	+37.03

Table 4 Computation Time (μs) for Boundary and Interior Moments, $B_C \in \Re^{9\times 30}$: 1000 Cases

Algorithm	Mean	Std Dev	Max	Min
PM (NPI)	+215.8	+127.4	+937.2	+59.56
PM (PI)	+150.4	+60.71	+468.7	+61.46
SNSGI (NPI)	+951.4	+1445	+1.714e+04	+54.99
SNSGI (PI)	+430	+614.1	+7456	+59.09
SNSGI (PI, IC)	+428.3	+623.6	+7786	+58.46
CGI	+249.1	+71.94	+661.5	+116.9
MCGI	+912.2	+925.4	+7655	+60.28

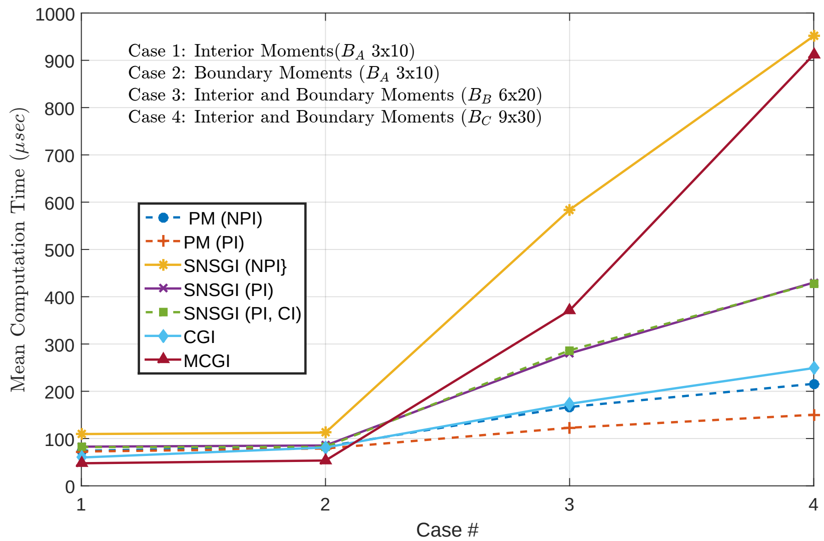

Fig. 1 Algorithm Mean Computation Time

Table 5 Boundary Validation Statistics, $B_A \in \Re^{3\times10}$, 1000 Cases

	$V_m(\%)$	$\mu_m(\%)$	$\sigma_m(\%)$	$V_u(\%)$	$\mu_u(\%)$	$\sigma_u(\%)$
PM (NPI)	+34.5	+0.4597	+0.7553	+39.6	-3.83	+5.16
PM (PI)	+35.5	+0.4468	+0.7485	+39.8	-3.772	+5.155
SNSGI (NPI)	+17.3	+2.99e-4	+6.662e-4	+24	-1.65	+3.516
SNSGI (PI)	+18.1	+2.97e-4	+6.505e-4	+24.5	-1.556	+3.448
SNSGI (PI, IC)	+0	+0	+0	+4	-5.922e-07	+1.317e-06
CGI	+34.3	+0.3191	+0.6327	+34.3	-8.074	+7.214
MCGI	+100	+1.312e-12	+8.316e-12	+100	-1.309e-10	+5.615e-10
DA	100	REF SOURCE *	-	100	REF SOURCE *	-

1) V_m, V_u: % validation cases. Failure (red) and success (black)
2) Mean (μ_m) and Std Dev (σ_m) of moment error metric E_m
3) Mean (μ_u) and Std Dev (σ_u) of control error metric E_u
* Unique DA solution used as reference control

Table 6 Interior Validation Statistics, $B_A \in \Re^{3\times10}$, 1000 Cases

	$V_m(\%)$	$\mu_m(\%)$	$\sigma_m(\%)$	$V_u(\%)$	$\mu_u(\%)$	$\sigma_u(\%)$
PM (NPI)	+1.1	+0.14	+0.3105	+3.7	+6.307	+6.484
PM (PI)	+1.1	+0.14	+0.3105	+3.7	+5.931	+6.519
SNSGI (NPI)	+0.6	+1.22e-05	+1.645e-05	+0.8	+0.942	+1.749
SNSGI (PI)	+0.5	+2.048e-05	+3.635e-05	+0.8	+0.1036	+4.645
SNSGI (PI, IC)	+100	+1.3e-12	+1.093e-11	100	REF SOURCE †	-
CGI	+1.7	+7.383e-2	+8.648e-2	+3.6	+3.704	+5.076
MCGI	+100	+2.338e-13	+4.632e-12	+99.8	+3.966e-11	+7.536e-10
MCGI	+0	+0	+0	+0.2	+8.081e-07	+5.621e-07
DUR-DA	+0	+0	+0	+91.4	+31.45	+17.05

See Table 5 notes
† SNSGI (PI, IC) solution (at equivalent moments) used as reference control

Table 7 Validation Statistics, $B_B \in \Re^{6\times20}$, 1000 Cases

	$V_m(\%)$	$\mu_m(\%)$	$\sigma_m(\%)$	$V_u(\%)$	$\mu_u(\%)$	$\sigma_u(\%)$
PM (NPI)	+2.6	+1.941	+2.091	+11	+1.14	+2.767
PM (PI)	+2.6	+1.941	+2.091	+11	+1.14	+2.767
SNSGI (NPI)	+0	+0	+0	+0.1	+4.096e-08	+0
SNSGI (PI)	+0	+0	+0	+0	+0	+0
SNSGI (PI, IC)	+100	+1.123e-12	+2.671e-11	+100	REF SOURCE †	-
CGI	+2.7	+0.9216	+1.203	+11.3	+1.04	+2.66
MCGI	+100	+7.559e-13	+1.894e-11	+99.8	+1.354e-12	+4.833e-11
MCGI	+0	+0	+0	+0.2	+2.452	+3.467

See Table 5 notes
† SNSGI (PI, IC) solution (at equivalent moments) used as reference control

Table 8 Validation Statistics, $B_C \in \Re^{9\times30}$, 1000 Cases

	$V_m(\%)$	$\mu_m(\%)$	$\sigma_m(\%)$	$V_u(\%)$	$\mu_u(\%)$	$\sigma_u(\%)$
PM (NPI)	+0.8	+3.58	+2.721	+8.6	+1.204	+3.07
PM (PI)	+0.8	+3.58	+2.721	+8.6	+1.204	+3.07
SNSGI (NPI)	+0	+0	+0	+0.1	-1.038e-09	+0
SNSGI (PI)	+0	+0	+0	+0.1	+1.393e-08	+0
SNSGI (PI, IC)	+100	+1.112e-13	+3.867e-13	+100	REF SOURCE †	-
CGI	+0.8	+1.454	+1.077	+8.7	+1.218	+3.232
MCGI	+100	+5.759e-14	+7.775e-13	+99.9	+2.276e-13	+7.271e-12
MCGI	+0	+0	+0	+0.1	-1.025e-09	+0

See Table 5 notes
† SNSGI (PI, IC) solution (at equivalent moments) used as reference control

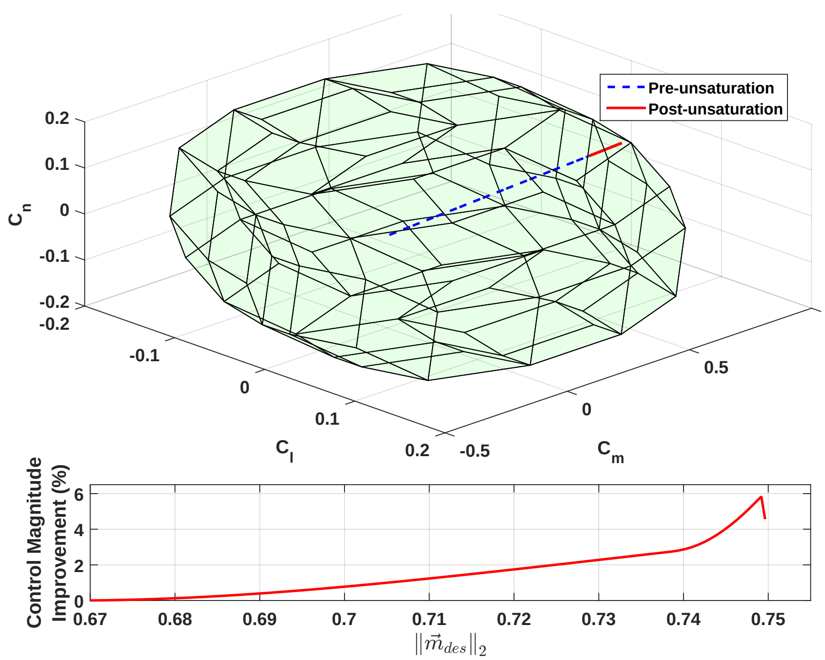

Fig. 2 Single Unsaturation Case MCGI to CGI Comparison

Appendix

The focus of this Appendix is to prove the equivalence of the control allocation solutions at each iteration for the baseline Prediction Method and the Cascading Generalized Inverse algorithm. Moreover, since the baseline Prediction Method was shown equivalent to the Affine Generalized Inverse algorithm [9], then through transitivity the Cascading Generalized Inverse algorithm is also equivalent to the Affine Generalized Inverse solution. The equivalence of these methods is predicated on utilizing the same sequence of unsaturated effector indices sets S_1^i and saturated effector indices sets S_2^i where at each iteration for $i \in \{a, b, \ldots\}$. The superscript denotes iteration order. For example, the sequence of saturated effector indices sets up to and including the fourth iteration is: S_2^a, S_2^b, S_2^c, S_2^d. The relationship between the saturated and unsaturated sets at each iteration is: $S_1^i \cup S_2^i = \{1, 2, \ldots, m\}$, $\forall i$ and $S_1^i \cap S_2^i = \emptyset$, $\forall i$.

The following Cascading Generalized Inverse iteration equations delineate the total moment, delta moment, total control and delta control vectors respectively for the i_{th} iteration for a given sequence of unsaturated and saturated effector indices sets [3].

$$\vec{m}_i = B\vec{u}_i \tag{46}$$

$$\vec{u}_i = \sum_{p=1}^{i} \Delta\vec{u}_p \tag{47}$$

$$\Delta\vec{u}_i = \left(k_i P_{T_i}\right) \Delta\vec{m}_i \tag{48}$$

$$\Delta\vec{m}_i = \vec{m}_{des} - \vec{m}_{i-1} \tag{49}$$

The general form of the corresponding minimal control baseline Prediction Method total control vector solution for a given desired moment $\vec{m}_{des}$ and linearized control effectiveness matrix B at the i_{th} iteration is [8]:

$$\vec{u}_{opt} = \vec{c}_0 + \vec{c}_1 + P_{min}\vec{m}_{des} \tag{50}$$

where

$$\vec{c}_0 = \hat{B}\hat{B}_2^T \left(\hat{B}_2\hat{B}_2^T\right)^{-1} \vec{s}_2, \text{ where } \vec{c}_0 \in \mathcal{N}(B) \tag{51}$$

$$\vec{c}_1 = -\hat{B}\hat{B}_2^T \left(\hat{B}_2\hat{B}_2^T\right)^{-1} P_{min_2}\vec{m}_{des}, \text{ where } \vec{c}_1 \in \mathcal{N}(B) \tag{52}$$

$$P_{min} = B^T \left(BB^T\right)^{-1} = \begin{bmatrix} P_{min_1} \\ P_{min_2} \end{bmatrix} \tag{53}$$

$$P_{min_2} = B_2^T \left(B_1B_1^T + B_2B_2^T\right)^{-1} \tag{54}$$

The Affine Generalized Inverse total control solution for the minimal control problem from [9] is:

$$\vec{u}_{opt} = \vec{c}_0 + P_{aff}\vec{m}_{des} \tag{55}$$

$$\vec{c}_0 = \hat{B}\hat{B}_2^T\left(\hat{B}_2\hat{B}_2^T\right)^{-1}\vec{s}_2, \text{ where } \vec{c}_0 \in \mathcal{N}\left(B\right) \tag{56}$$

Moreover, in [9], it was shown that:

$$\vec{c}_1 = \left(P_{aff} - P_{min}\right)\vec{m}_{des} = -\hat{B}\hat{B}_2^T\left(\hat{B}_2\hat{B}_2^T\right)^{-1}P_{min_2}\vec{m}_{des} \tag{57}$$

The generation of the generalized inverse matrix P_{aff} from Eq. (55) is identical to that of P_{t_i} from the Cascading Generalized Inverse method in Eq.(48). Specifically, given the unsaturated effector indices set S_1^i at the i_{th} iteration, then dividing B into unsaturated and saturated columns and setting the saturated columns to zero vectors then:

$$P_{aff_i} = P_{T_i} = \begin{bmatrix} B_1^T \\ 0 \end{bmatrix}\left(\begin{bmatrix} B_1 & 0 \end{bmatrix}\begin{bmatrix} B_1^T \\ 0 \end{bmatrix}\right)^{-1} = \begin{bmatrix} B_1\left(B_1B_1^T\right)^{-1} \\ 0 \end{bmatrix} \tag{58}$$

Lemma 0.1. *Given the Cascading Generalized Inverse definitions of the total moment, delta moment, total control and delta control in Eqs. (46-48) for the minimal control allocation problem, then the total control vector $\vec{u}_i$ for the i_{th} iteration can be divided into two orthogonal components $\vec{u}_i = \vec{u}_{\|_i} + \vec{u}_{\perp_i}$ where:*

$$\vec{u}_{\|_i} = P_{min}\vec{m}_i \tag{59}$$

$$\vec{u}_{\perp_i} = \left(P_{T_i} - P_{min}\right)\vec{m}_i + \left(I - P_{T_i}B\right)\sum_{p=1}^{i-1}\Delta\vec{u}_p \tag{60}$$

Proof. This proof will show that the orthogonal components $\vec{u}_{\|_i}$ and $\vec{u}_{\perp_i}$ are contained in the orthogonal subspaces generated by the range space of the column vectors of the transpose of the controls effectiveness matrix B^T and the column vectors of the null space of B basis matrix $\hat{B}$ respectively. We know from linear algebra that any control vector $\vec{u}$ is the sum of the control vectors projections onto $\mathcal{P}_{B^T}\left(\vec{u}\right)$ and $\mathcal{P}_{\mathcal{N}(B)}\left(\vec{u}\right)$, so then $\vec{u} = \mathcal{P}_{B^T}\left(\vec{u}\right) + \mathcal{P}_{\mathcal{N}(B)}\left(\vec{u}\right)$. Given a total control vector $\vec{u}_i \in \Re^m$ generated at the i_{th} segment by the Cascading Generalized Inverse method, then application of the projection operator $\mathcal{P}_{B^T}\ (\vec{u}_i) = B^T\ \left(BB^T\right)^{-1} B\vec{u}_i$ together with Eqs.(46-47) yields:

$$\mathcal{P}_{B^T}\ (\vec{u}_i) = B^T\left(BB^T\right)^{-1} B\vec{u}_i = P_{min}B\vec{u}_i \tag{61}$$

$$= P_{min}B\left(\Delta\vec{u}_i + \sum_{p=1}^{i-1}\Delta\vec{u}_p\right) \tag{62}$$

$$= P_{min}B\left(k_iP_{T_i}\right)\left(\vec{m}_{des} - B\sum_{p=1}^{i-1}\Delta\vec{u}_p\right) + P_{min}B\sum_{p=1}^{i-1}\Delta\vec{u}_p \tag{63}$$

Now since we are showing the division of the total control vector $\vec{u}_i$ for the i-th iteration (i.e., linear segment), then without loss of generality, we can choose the desired moment such that it is achieved within the i-th iteration $\vec{m}_{des} = \vec{m}_i = B\sum_{p=1}^{i}\Delta\vec{u}_p$ or equivalently $\Delta\vec{m}_{i+1} = \vec{0}$. This is permissible as the magnitude of $\vec{m}_{des}$ does not change the $\Delta\vec{u}_i$ solutions for the current iteration or any prior iteration in which the moment $\vec{m}_{des}$ is not satisfied. Specifically, the scaling factor scales $\Delta\vec{m}$ resulting in identical $\Delta\vec{u}_p$. For example given two desired moments $\vec{m}_a$ and $\vec{m}_b$ which differ in magnitude but have identical directions and which are not satisfied during the i-th iteration, then the delta control solution in an iteration is $\Delta\vec{u} = P_t\ (k_a\Delta\vec{m}_a) = P_t\ (k_b\Delta\vec{m}_b)$. Moreover, for the Cascading Generalized Inverse method, if desired total moment is achieved during the i-th iteration then no scaling is required (i.e., $k_i = 1$.) Finally, for any generalized inverse (particularly the minimal control P_{min}), we have $BP_{min} = I$ and therefore for the iteration in which the desired moment is achieved then $B\left(k_iP_{T_i}\right) = I$. Equation (63) is simplified to yield the total control parallel component as:

$$\vec{u}_{\parallel_i} = \mathcal{P}_{B^T}\ (\vec{u}_i) = P_{min}\left(\vec{m}_{des} - B\sum_{p=1}^{i-1}\Delta\vec{u}_p\right) + P_{min}B\sum_{p=1}^{i-1}\Delta\vec{u}_p, \implies \vec{u}_{\parallel_i} = P_{min}\vec{m}_{des} = P_{min}\vec{m}_i \tag{64}$$

Now analyzing the perpendicular component by projecting onto the subspace spanning the null space of B, we have:

$$\vec{u}_{\perp_i} = \mathcal{P}_{\mathcal{N}(B)}\ (\vec{u}_i) = \mathcal{P}_{(I-P_{min}B)}\ (\vec{u}_i) \tag{65}$$

$$= (I - P_{min}B)\left(\Delta\vec{u}_i + \sum_{p=1}^{i-1}\Delta\vec{u}_p\right) \tag{66}$$

$$= (I - P_{min}B)\left(k_iP_{T_i}\left(\vec{m}_{des} - B\sum_{p=1}^{i-1}\Delta\vec{u}_p\right)\right) + (I - P_{min}B)\sum_{p=1}^{i-1}\Delta\vec{u}_p \tag{67}$$

Again, without loss of generality, we assume the direction of $\vec{m}_{des}$ is preserved but the magnitude is chosen such that the moment is achieved during the i_{th} iteration and therefore $k_i = 1$. Note that the above expressions hold for each point including the end points. At each corner j where $(j < i)$ then $k_j = 1$ and $\vec{u}_{\perp_j}$ is defined as above with $\vec{m}_{des} = \vec{m}_j$.

Additionally, for any generalized inverse we have $BP_{min} = I$ and $BP_{T_i} = I$ since:

$$BP_{t_i} = \begin{bmatrix} B_1 & B_2 \end{bmatrix} \begin{bmatrix} P_{t1_i} \\ 0 \end{bmatrix} = B_1 B_1^T \left(B_1 B_1^T\right)^{-1} + 0 = I \tag{68}$$

Now Eq. (67) is expanded and simplified:

$$\vec{u}_{\perp_i} = P_{t_i}\vec{m}_{des} - P_{t_i}B\sum_{p=1}^{i-1}\Delta\vec{u}_p - P_{min}\vec{m}_{des} + P_{min}B\sum_{p=1}^{i-1}\Delta\vec{u}_p + \sum_{p=1}^{i-1}\Delta\vec{u}_p - P_{min}B\sum_{p=1}^{i-1}\Delta\vec{u}_p \tag{69}$$

$$\vec{u}_{\perp_i} = \left(P_{t_i} - P_{min}\right)\vec{m}_{des} + \left(I - P_{t_i}B\right)\sum_{p=1}^{i-1}\Delta\vec{u}_p \tag{70}$$

where $\vec{m}_{des} = \vec{m}_i$. Thus Eq.(64) and Eq.(70) shown the division of the Cascading Generalized Inverse total control vector $\vec{u}_i$ as claimed. □

Theorem 1. *Given a linearized control effectiveness matrix $B \in \Re^{n\times m}$, a desired outcome (moment) $\vec{m}_{des}$, upper ($\vec{u}_{upr}$) and lower ($\vec{u}_{lwr}$) effector position limits, then the Prediction Method Eqs. (50- 52) and Cascading Generalized Inverse Eq. (47) solutions at the i_{th} iteration to the minimal control allocation problem $\vec{u}_{opt} = \text{argmin}_{\vec{u}} \left\{\vec{u}^T\vec{u} \mid \vec{m}_{des} = B\vec{u},\, \vec{u} \in \Omega\right\}$ are equivalent for a given sequence of unsaturation and saturation sets S_1^p and S_2^p where $p = 1, 2, \ldots, i$, $\forall i \leq (m - n)$.*

Proof. The optimal allocation solutions to the minimal control allocation problem for a given desired moment $\vec{m}_{des}$ along a moment direction $\hat{m}_{des} = \vec{m}_{des}/\|\vec{m}_{des}\|_2$ are known to be piecewise linear and continuous. Both the PM and the CGI solutions for the i_{th} iteration where $i \leq (m - n)$ are obtained by finding a sequence of effector indices sets (saturated and unsaturated) and solving for the corresponding control solution at the "corners" of each piecewise linear segment. This iteration process is typically continued until the desired moment is achieved or the $i = (m - n)$ iteration is performed but is unable to completely satisfy the desired moment. As the PM and CGI algorithms solution methods differ after $(m - n)$ iterations, this proof is limited to showing equivalent solutions for moments which can be satisfied within $(m - n)$ iterations. This proof specifically shows the equivalence of PM and CGI solutions for the iteration during which the desired moment is achieved. However, iterative application of this proof to a sequentially increasing set of desired moment magnitudes (one for each iterative segment) along the desired moment direction $\hat{m}_{des}$ shows that the PM and CGI solutions are equivalent for all iterations up to and including $i = (m - n)$.

Given the CGI solution from Eq. (47) at the i_{th} iteration that satisfies the desired moment (i.e., $\vec{m}_i = \vec{m}_{des}$), then Lemma 0.1 shows the partition of the total allocation solution such that:

$$\vec{u}_i = \vec{u}_{\|_i} + \vec{u}_{\perp_i} = P_{min}\vec{m}_i + \left(P_{t_i} - P_{min}\right)\vec{m}_i + \left(I - P_{t_i}B\right)\sum_{p=1}^{i-1}\Delta\vec{u}_p \tag{71}$$

Recalling that P_{T_i} and P_{aff_i} from Eq. (58) are equivalent for identical saturation sets S_1^i and S_2^i, then comparison of Eq. (71) to Eqs. (50,52,57) shows that the CGI solution contains both the $\vec{u}_{\|_i}$ and $\vec{c}_1$ components of the PM solution. It only remains to show that:

$$\vec{c}_0 = \hat{B}\hat{B}_2^T \left(\hat{B}_2\hat{B}_2^T\right)^{-1} \vec{s}_2 \iff \left(I - P_{T_i}B\right)\sum_{p=1}^{i-1} \Delta\vec{u}_p \tag{72}$$

In the sequel, for simplicity we will drop the i notation subscript when refering the the i_{th} iteration quantities but will retain it for quantities associated with previous iterations. We express $\vec{c}_0$ as unsaturated and saturated components (based on the i_{th} iterations sets S_1 and S_2) such that $\vec{c}_0 = \begin{bmatrix} \vec{c}_{0_1}^T & \vec{c}_{0_2}^T \end{bmatrix}^T$ and similarly the columns of B such that $B = \begin{bmatrix} B_1 & B_2 \end{bmatrix}$. Now recalling the generalized inverses P_t and P_{aff} for identical effector indices sets:

$$P_{aff} = \begin{bmatrix} B_1^T \\ 0 \end{bmatrix} \left(\begin{bmatrix} B1 & 0 \end{bmatrix} \begin{bmatrix} B_1^T \\ 0 \end{bmatrix} \right)^{-1} = \begin{bmatrix} B_1^T \left(B_1 B_1^T\right)^{-1} \\ 0 \end{bmatrix} = \begin{bmatrix} P_{aff_1} \\ 0 \end{bmatrix} = P_T \tag{73}$$

Then expressing the CGI portion of Eq. (72) using the unsaturated and saturated components yields:

$$\begin{bmatrix} \vec{a}_1 \\ \vec{a}_2 \end{bmatrix} = \left(I - P_{T_i}B\right)\sum_{p=1}^{i-1} \Delta\vec{u}_p = \left(\begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} - \begin{bmatrix} P_{aff_1} \\ 0 \end{bmatrix} \begin{bmatrix} B_1 & B_2 \end{bmatrix} \right) \sum_{p=1}^{i-1} \Delta\vec{u}_p = \begin{bmatrix} \left(I - P_{aff_1}B_1\right) & -P_{aff_1}B_2 \\ 0 & I \end{bmatrix} \sum_{p=1}^{i-1} \Delta\vec{u}_p \tag{74}$$

Now recalling from [3] that the delta control vectors $\Delta\vec{u}_1, \Delta\vec{u}_2, \ldots$ is a sequence of vectors where one effector is saturated and held constant for each iteration. Moreover, due to the nature of the P_{aff} where $P_{aff_2} = 0$, then subsequent delta control vectors don't contribute to the saturated controls. For example if the j_{th} component of the first delta control vector is saturated at the upper position limit, then $\Delta\vec{u}_1(j) = \vec{u}_{upr}(j)$ then $\Delta\vec{u}_2(j) = \Delta\vec{u}_3(j) = \cdots = 0$. Therefore we can express the sum of the $1 \ldots (i-1)$ delta control vectors in unsaturated and saturated form as:

$$\sum_{p=1}^{i-1} \Delta\vec{u}_p = \begin{bmatrix} \sum_{p=1}^{i-1} \Delta\vec{u}_{p_1} \\ \sum_{p=1}^{i-1} \Delta\vec{u}_{p_2} \end{bmatrix} = \begin{bmatrix} \vec{s}_1 \\ \vec{s}_2 \end{bmatrix} \tag{75}$$

where $\vec{s}_2$ is the vector effector positions corresponding to the active position limits for the saturated effector indices set S_2. Similarly, the $\vec{s}_1$ vector is the sum of the delta control vector positions for the unsaturated effectors associated with the set S_1. Substitution of Eq. (75) into Eq. (74) shows the unsaturated and saturated portions of $\left(I - P_{T_i}B\right)\sum_{p=1}^{i-1} \Delta\vec{u}_p$

are (respectively):

$$\vec{a}_1 = \left(I - P_{aff_1} B_1\right) \vec{s}_1 - P_{aff_1} B_2 \vec{s}_2 \tag{76}$$

$$\vec{a}_2 = \vec{s}_2 \tag{77}$$

Now expressing the PM $\vec{c}_0$ from Eq. (56) into unsaturated and saturated vector form yields:

$$\vec{c}_0 = \begin{bmatrix} \vec{c}_{0_1} \\ \vec{c}_{0_2} \end{bmatrix} = \begin{bmatrix} \hat{B}_1 \\ \hat{B}_2 \end{bmatrix} \hat{B}_2^T \left(\hat{B}_2 \hat{B}_2^T\right)^{-1} \vec{s}_2 = \begin{bmatrix} \hat{B}_1 \hat{B}_2^T \left(\hat{B}_2 \hat{B}_2^T\right)^{-1} \vec{s}_2 \\ \vec{s}_2 \end{bmatrix} \tag{78}$$

Equations (77, 78) show that the saturated components of the PM and CGI algorithm are identical, $\vec{c}_{0_2} = \vec{a}_2$. Therefore, it remains to show equivalence of the unsaturated components:

$$\left(I - P_{aff_1} B_1\right) \sum_{p=1}^{i-1} \Delta \vec{u}_{p_1} - P_{aff_1} B_2 \vec{s}_2 = \hat{B}_1 \hat{B}_2^T \left(\hat{B}_2 \hat{B}_2^T\right)^{-1} \vec{s}_2 \tag{79}$$

First we will show:

$$\left(I - P_{aff_1} B_1\right) \sum_{p=1}^{i-1} \Delta \vec{u}_p = \vec{0} \tag{80}$$

where $\Delta \vec{u}_p = \left[P_{\mathrm{aff}_{1,p}}\right]_i \left(k_p \Delta \vec{m}_p\right)$, $\forall p \in \{1, 2, \ldots, i-1\}$. The notation $[x]_i$ denotes retaining the rows of x associated with the unsaturated effectors (S_1^i) of iteration i. We utilize the notation P_i to denote the P_{aff_1} for the i_{th} iteration so:

$$\left(I - P_{aff_1} B_1\right) \sum_{p=1}^{i-1} \Delta \vec{u}_p = \left(I - P_i B_1\right) \left([P_1]_i \, k_1 \Delta \vec{m}_1 + [P_2]_i \, k_2 \Delta \vec{m}_2 + \cdots + [P_{i-1}]_i \, k_1 \Delta \vec{m}_1\right) \tag{81}$$

To expand, note B_1 for $p < i$ can be expressed as $B_{1_p} = \begin{bmatrix} B_1 & B_{1,2p} \end{bmatrix}$ where B_1 corresponds with the i_{th} iteration, then

$$P_p = \begin{bmatrix} B_1^T \left(B_1 B_1^T + B_{1,2_p} B_{1,2_p}^T\right)^{-1} \\ B_{1,2_p}^T \left(B_1 B_1^T + B_{1,2_p} B_{1,2,_p}^T\right)^{-1} \end{bmatrix}, \forall p \in \{1, 2, \ldots, i-1\} \tag{82}$$

$$\left[P_p\right]_i = B_1^T \left(B_1 B_1^T + B_{1,2_p} B_{1,2_p}^T\right)^{-1} \tag{83}$$

$$P_i = B_1^T \left(B_1 B_1^T\right)^{-1} \tag{84}$$

Note B_1 for each iteration $p < i$ can be expressed as $B_{1_p} = \begin{bmatrix} B_1 & B_{1,2p} \end{bmatrix}$ where B_1 corresponds to the i_{th} iteration.

Examining $(I - P_i B_1) P_p$ we see that:

$$(I - P_i B_1) P_p = \left(I - B_1^T \left(B_1 B_1^T\right)^{-1} B_1\right) B_1^T \left(B_1 B_1^T + B_{1,2_p} B_{1,2_p}^T\right)^{-1} \tag{85}$$
$$= B_1^T \left(B_1 B_1^T + B_{1,2_p} B_{1,2_p}^T\right) - B_1^T \left(B_1 B_1^T\right)^{-1} \left(B_1 B_1^T\right) \left(B_1 B_1^T + B_{1,2_p} B_{1,2_p}^T\right) = 0,\ \forall p \in \{1, 2, \ldots, i-1\}$$

and thus Eq. (80) is identically zero. Now referring back to Eq. (79), it only remains to show that:

$$\left(P_{af\,f1} B_2 + \hat{B}_1 \hat{B}_2^T \left(\hat{B}_2 \hat{B}_2^T\right)^{-1}\right) \vec{s}_2 = 0 \tag{86}$$

Now we want Eq. (86) to hold $\forall \vec{s}_2 \in \Re^k$ with $k = \#$ of saturated effectors. We tranform the linear system into an equivalent one by using the invertible matrix $\left(\hat{B}_2 \hat{B}_2^T\right)$ and defining the vector $\vec{v}$ such that $\vec{s}_2 = \left(\hat{B}_2 \hat{B}_2^T\right) \vec{v}$ and substituting this into the left side of Eq. (86), using $P_{af\,f\,1} = B_1^T \left(B_1 B_1^T\right)^{-1}$ and transposing yields:

$$\vec{v}^T \left(\hat{B}_2 \hat{B}_2^T B_2^T \left(B_1 B_1^T\right)^{-1} B_1 + \hat{B}_2 \hat{B}_1^T\right) \tag{87}$$

Now we will show that $\left(\hat{B}_2 \hat{B}_2^T B_2^T \left(B_1 B_1^T\right)^{-1} B_1 + \hat{B}_2 \hat{B}_1^T\right)$ is the zero matrix by proving that the right null space of this matrix is the entire subspace. Since B is robust full rank, then B_1^T is full rank and from linear algebra the columns of $\begin{bmatrix} B_1^T & \mathcal{N}(B_1) \end{bmatrix}$ span the entire space. Therefore we create any arbitrary vector $\vec{w} = \begin{bmatrix} B_1^T & \mathcal{N}(B_1) \end{bmatrix} \vec{x}$ and thus:

$$0 = \vec{v}^T \left(\hat{B}_2 \hat{B}_2^T B_2^T \left(B_1 B_1^T\right)^{-1} B_1 + \hat{B}_2 \hat{B}_1^T\right) \begin{bmatrix} B_1^T & \mathcal{N}(B_1) \end{bmatrix} \vec{x},\ \forall \vec{x} \tag{88}$$

Now since the product of the control effectiveness matrix and its null space basis is zero then:

$$0 = B \hat{B}_v = \begin{bmatrix} B_1 & B_2 \end{bmatrix} \begin{bmatrix} \hat{B}_1 \\ \hat{B}_2 \end{bmatrix} \implies -B_2 \hat{B}_2 = B_1 \hat{B}_1 \implies -\hat{B}_2^T B_2^T = \hat{B}_1^T B_1^T \tag{89}$$

and therefore distributing B_1^T in Eq. (88) and using Eq. (89) simplifies to:

$$\left(\hat{B}_2 \hat{B}_2^T B_2^T + \hat{B}_2 \hat{B}_1^T B_1^T\right) = \left(-\hat{B}_2 \hat{B}_1^T B_1^T + \hat{B}_2 \hat{B}_1^T B_1^T\right) = 0 \tag{90}$$

When distributing $\mathcal{N}(B_1)$ in Eq. (88), the first term is identically zero as $B_1 \mathcal{N}(B_1) = 0$. Now since the $\begin{bmatrix} \mathcal{N}(B_1) \\ \vec{0} \end{bmatrix} \in$

span $\mathcal{N}(B)$, then from the singular value decomposition of $B = U\Lambda V^T$, where $VV^T = I$ with $V = \begin{bmatrix} V_a & \hat{B} \end{bmatrix}$:

$$\begin{bmatrix} \mathcal{N}(B_1) \\ \vec{0} \end{bmatrix} = \left(V_a V_a^T + \hat{B}\hat{B}^T\right) \begin{bmatrix} \mathcal{N}(B_1) \\ \vec{0} \end{bmatrix} = \begin{bmatrix} \hat{B}_1\hat{B}_1^T & \hat{B}_1\hat{B}_2^T \\ \hat{B}_2\hat{B}_1^T & \hat{B}_2\hat{B}_2^T \end{bmatrix} \begin{bmatrix} \mathcal{N}(B_1) \\ \vec{0} \end{bmatrix} \implies \hat{B}_2\hat{B}_1^T \mathcal{N}(B_1) = 0 \tag{91}$$

Therefore Eq. (88) holds for all $\vec{x}$ and thus $\vec{c}_{0_2} = \hat{B}_1\hat{B}_2^T \left(\hat{B}_2\hat{B}^T\right)^{-1} \vec{s}_2 = -P_{aff1}B_2\vec{s}_2$, so the PM and CGI minimal control allocation solutions are equivalent for the i_{th} iteration where $i \leq (m-n)$ as claimed. □

Funding Sources

This work was supported with funding from NASA.

References

[1] Harris, J. J., and Stanford, J. R., "F-35 Flight Control Law Design, Development and Verification," *AIAA AVIATION Forum*, AIAA Paper 2018-3516, 2018. doi:10.2514/6.2018-3516.

[2] Denham, J. W., "Project MAGIC CARPET: Advanced Controls and Displays for Precision Carrier Landings," *AIAA SciTech Forum*, AIAA Paper 2016-1770, 2016. doi:10.2514/6.2016-1770.

[3] Bordignon, K., and Bessolo, J., "Control Allocation For The X-35B," *2002 Biennial International Powered Lift Conference and Exhibit*, AIAA Paper AIAA 2002-6020, 2002. doi:10.2514/6.2002-6020.

[4] B., D. J., Lallman, F. J., and Bundick, W. T., "Integrated Reconfigurable Control Allocation," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, AIAA Paper 2001-4083, 2001. doi:10.2514/6.2001-4083.

[5] Frost, S. A., Bodson, M., Burken, J. J., Jutte, C. V., Taylor, B. R., and Trinh, K. V., "Flight Control with Optimal Control Allocation Incorporating Structural Load Feedback," *Journal of Aerospace Information Systems*, Vol. 12, No. 12, 2015, pp. 825–834. doi:10.2514/1.I010278.

[6] Durham, W. C., "Constrained Control Allocation," *Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 4, 1994, pp. 717, 725. doi:10.2514/3.21201.

[7] Jin, J., "Modified Pseudoinverse Redistribution Methods for Redundant Controls Allocation," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 5, 2005, pp. 1076–1079. doi:10.2514/1.14992.

[8] Acheson, M. J., "Optimal Control Prediction Method for Control Allocation," *AIAA Aviation and Aeronautics Forum and Exposition (AIAA AVIATION 2018)*, AIAA Paper 2018-3897, Atlanta, GA, 2018. doi:10.2514/6.2018-3897.

[9] Acheson, M. J., "Affine Generalized Inverse for Optimal Control Allocation," *AIAA Aviation and Aeronautics Forum and Exposition (AIAA AVIATION 2018)*, AIAA Paper 2018-3902, Atlanta, GA, 2018. doi:10.2514/6.2018-3902.

[10] Harkegard, O., “Efficient Active Set Algorithms for Solving Constrained Least Squares Problems in Aircraft Control Allocation,” *Conference on Decision and Control*, IEEE, 2002.

[11] Harkegard, O., “Dynamic Control Allocation Using Constrained Quadratic Programming,” *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 6, 2004, pp. 1028–1034.

[12] Petersen, J. A. M., “Interior-Point Algorithms for Control Allocation,” *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 3, 2005, pp. 471–480.

[13] Petersen, J. A. M., and Bodson, M., “Constrained Quadratic Programming Techniques for Control Allocation,” *Conference on Decision and Control*, IEEE Paper 01271666, 2003.

[14] Bordignon, K. A., and Durham, W. C., “Null-Space Augmented Solutions to Constrained Control Allocation Problems,” *Guidance, Navigation, and Control Conference*, AIAA Paper 1995-3209, 1995. doi:10.2514/6.1995-3209.

[15] Demenkov, M. N., “Reconfigurable linear control allocation via generalized bisection,” *Guidance, Navigation, and Control Conference*, AIAA Paper 2005-6341, 2005. doi:10.2514/6.2005-6341.

[16] Cameron, D., and Princen, N., “Control Allocation Challenges and Requirements for the Blended Wing Body,” *Guidance, Navigation, and Control Conference*, AIAA Paper 2000-4539, 2000. doi:10.2514/6.2000-4539.

[17] Page, A. B., and Steinberg, M. L., “A Closed-Loop Comparison of Control Allocation Methods,” *Guidance, Navigation, and Control Conference*, AIAA Paper 2000-4538, 2000. doi:10.2514/6.2000-4538.

[18] Bodson, M., “Evaluation of Optimization Methods for Control Allocation,” *AIAA Guidance, Navigation, and Control Conference and Exhibit*, AIAA Paper 2001-4223, 2001. doi:10.2514/6.2001-4223.

[19] Enns, D., “Control Allocation Approaches,” *AIAA Guidance, Navigation, and Control Conference and Exhibit*, AIAA, Boston, MA, 1998, pp. 98,108.

[20] Bordignon, K. A., *Constrained Control Allocation for Systems with Redundant Control Effectors*, Ph.D. Dissertation, Virginia Polytechnic Institute and State University, Blacksburg, VA, 1996.

[21] Acheson, M. J., “Computational Efficiency Enhancements for Optimal Control Allocation,” *AIAA Aviation and Aeronautics Forum and Exposition (AIAA AVIATION 2019)*, AIAA Paper 2019-3462, Dallas, TX, 2019. doi:10.2514/6.2019-3462.

[22] Glaze, M., and Durham, W. C., “Design and Implementation of a Control Allocation Toolbox for Matlab V5.0,” AIAA Paper 1998-4436, 1998. doi:10.2514/6.1998-4436.

[23] Durham, W., Bordignon, K. A., and Beck, R., *Aircraft Control Allocation*, AIAA Aerospace Series, Wiley, West Sussex, United Kingdom, 2017.