

Control Allocation Algorithm for Hypersonic Glide Vehicles with Input Limitations

Johannes Autenrieb*

DLR, German Aerospace Center, 38108 Braunschweig, Germany

and

Patrick Gruhn[†]

DLR, German Aerospace Center, 51147 Cologne, Germany

<https://doi.org/10.2514/1.G009588>

Hypersonic glide vehicles (HGVs) operate in challenging flight regimes characterized by strong nonlinearities in actuation and stringent physical constraints. These include state-dependent actuator limitations, asymmetric control bounds, and thermal loads that vary with maneuvering conditions. This paper introduces an iterative control allocation method to address these challenges in real time. The proposed algorithm searches for control inputs that achieve the desired moment commands while respecting constraints on input magnitude and rate. For slender HGV configurations, thermal loads and drag generation are strongly correlated—lower drag typically results in reduced surface heating. By embedding drag-sensitive soft constraints, the method improves energy efficiency and implicitly reduces surface temperatures, lowering the vehicle's infrared signature. These features are particularly advantageous for long-range military operations that require low observability. The approach is demonstrated using the DLR's Generic Hypersonic Glide Vehicle 2 (GHGV-2) simulation model. The results confirm the method's effectiveness in maintaining control authority under realistic, constrained flight conditions.

Nomenclature

$\mathbf{B}$	= control effectiveness matrix, $\mathbb{R}^{3 \times 4}$
$\mathbf{B}_{\nu_x}, \mathbf{B}_{\nu_y}, \mathbf{B}_{\nu_z}$	= conditionalized rows of $\mathbf{B}$ associated with roll, pitch, and yaw
C_D	= aerodynamic drag coefficient
$\mathbf{D}$	= attainable moment set, $\mathbb{R}^3$
$\mathbf{h}(\mathbf{x})$	= output mapping, $\mathbb{R}^n \rightarrow \mathbb{R}^p$
J	= allocation cost, $J = \ \mathbf{u}\ _2^2$
$\mathbf{L} = [\mathbf{L}, \mathbf{M}, \mathbf{N}]^T$	= body-axis moment vector (roll, pitch, yaw), $\text{N} \cdot \text{m}$
$L_i \mathbf{h}$	= Lie derivative of function $\mathbf{h}$ along vector field $\mathbf{i}$, e.g., $\mathbf{f}, \mathbf{g}$
N_{iter}	= number of iterations in iterative dynamic control allocation loop
$\mathbf{P}$	= cost matrix in quadratic program
q	= dynamic pressure, N/m^2
r	= relative degree of the output
T	= time step, s
$\mathbf{u}$	= control input vector, $\mathbb{R}^4$
u_i	= control input for effector i , deg
$\dot{u}_i$	= rate of control input i , deg/s
$u_{\max,i}$	= upper magnitude bound for u_i , deg
$u_{\min,i}$	= lower magnitude bound for u_i , deg
$\dot{u}_{\max,i}$	= upper rate bound for u_i , deg/s
$\dot{u}_{\min,i}$	= lower rate bound for u_i , deg/s
$\mathbf{u}_s$	= desired steady-state control input vector
$\mathbf{u}_r$	= baseline control input from trajectory planning
$\boldsymbol{\nu}$	= virtual control input vector, $\mathbb{R}^3$

ν_x, ν_y, ν_z	= components of $\boldsymbol{\nu}$, corresponding to roll (L), pitch (M), and yaw (N)
$\mathbf{x}$	= state vector, $\mathbb{R}^n$
$\mathbf{y}$	= output vector, $\mathbb{R}^p$
α	= angle of attack, rad
β	= sideslip angle, rad
γ	= flight path angle, rad
$\Delta \mathbf{u}$	= control input correction vector
$\Delta \boldsymbol{\nu}$	= corrective virtual control command
$\Lambda(t)$	= time-dependent modulation of input constraints
ρ	= atmospheric air density, kg/m^3
$\boldsymbol{\sigma}$	= operating point vector, e.g., altitude and Mach number

Subscripts

c	= command or control
i	= index for control surface or effector
iter	= iteration index
max	= maximum bound
min	= minimum bound
r	= reference or baseline input
s	= steady-state solution

I. Introduction

IN RECENT years, hypersonic glide vehicles (HGVs) have received increasing attention in academic and industrial research and development [1]. These vehicles, designed to navigate through lower atmospheric layers at hypersonic speeds, present unique challenges that require precise control for optimal performance and guaranteed stability. Some HGV designs incorporate redundant control effectors, necessitating a control allocation system capable of distributing desired forces and moments among a redundant set of control effectors to achieve the desired vehicle response [2].

Traditional control allocation methods for overactuated aerial vehicles have evolved significantly [3]. For most aerospace applications, the pseudo-inverse-based control allocation (PICA) method [4], utilizing the Moore–Penrose inverse, is commonly used due to its simplicity. However, PICA does not adequately account for control input limitations. An alternative and more precise approach is to formulate the constrained control allocation problem as a

Received 4 September 2025; accepted for publication 13 February 2026; published online 19 June 2026. Copyright © 2026 by Deutsches Zentrum fuer Luft- und Raumfahrt e.V. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission. All requests for copying and permission to reprint should be submitted to CCC at www.copyright.com; employ the eISSN 1533-3884 to initiate your request. See also AIAA Rights and Permissions <https://aiaa.org/publications/publish-with-aiaa/rights-and-permissions/>.

*Research Scientist, Department of Flight Dynamics and Simulation, Institute of Flight Systems; johannes.autenrieb@dlr.de (Corresponding Author).

[†]Head of Missile Technologies Group, Department of Supersonic and Hypersonic Technologies, Institute of Aerodynamics and Flow Technology; patrick.gruhn@dlr.de.

quadratic programming (QPCA) problem and solve it using a QP solver [5,6]. This method can accurately handle asymmetric input constraints and coupling effects between control effectors. However, the computational intensity of QP solvers poses significant challenges for real-time implementation on systems with limited computational resources, such as embedded systems in hypersonic reentry vehicles [7]. The high computational demand makes it difficult to guarantee real-time performance, which is essential for the fast dynamics of HGVs. To address the need for a computationally less complex solution, Bordignon introduced the redistributed pseudo-inverse control allocation (RPICA) method [8]. RPICA iteratively approximates feasible solutions by redistributing residuals over the attainable moment set (AMS), offering a balance between computational efficiency and practicality. A scaled variant, redistributed scaled pseudo-inverse control allocation (RSPICA), was later proposed to maintain the directionality of the input vector in multiple-input, multiple-output (MIMO) systems under saturation conditions [9]. Despite advancements in computational simplicity, both RPICA and RSPICA face challenges when dealing with systems characterized by asymmetric control input limits and strong inter-directional coupling between control effectors. This issue is particularly evident in certain classes of HGVs where only positive control deflections are permissible. The inability of these methods to effectively handle asymmetric constraints and coupling effects limits their applicability in such scenarios.

Over the past few years, the German Aerospace Center (DLR) has undertaken a substantial research initiative focused on the development of the Generic Hypersonic Glide Vehicle 2 (GHGV-2). This research aims to comprehensively explore the capabilities and performance characteristics of strategic hypersonic glide vehicles, shedding light on their implications for understanding hypersonic threats [10,11]. The GHGV-2 adopts a waverider design to optimize lift-to-drag ratios at high Mach numbers [12]. The GHGV-2 is depicted in Fig. 1, with an external view shown in Fig. 1a and a sectional view highlighting relevant subsystems in Fig. 1b. Key subsystems include the thermal protection system (TPS), guidance, navigation, and control (GNC) system, battery, and actuators. During endoatmospheric flight, the GHGV-2 utilizes four integrated flaps as control effectors to regulate its attitude. Small propulsors are integrated in exoatmospheric conditions to generate the necessary moments for attitude adjustments.

The GHGV-2 is launched via a multistage rocket booster as part of the investigated mission profile. After ignition and acceleration, the vehicle separates from the launch system at an altitude of approximately 100 km, initiating a parabolic reentry trajectory. Once sufficient dynamic pressure is achieved during descent, the vehicle transitions to aerodynamic control using four surfaces, two located on the upper side and two on the lower side. Following reentry, the GHGV-2 tracks a predefined flight path angle to sustain its maximum lift-to-drag ratio en route to the target location. Due to sustained hypersonic speeds, the vehicle is exposed to significant thermal loads during this atmospheric glide phase. Consequently, any viable control allocation algorithm must generate the necessary control moments for maneuvering and account for thermal effects,

ideally minimizing heat loads to reduce the burden on the TPS and lower the infrared signature, a key consideration for military applications. Another critical challenge arises from state-dependent control input limitations stemming from the wide range of operating conditions. Increased dynamic pressure at lower altitudes and higher speeds restricts actuator limits compared to flight conditions involving lower speeds or air densities. This variability demands a control allocation approach explicitly incorporating state-dependent constraints to ensure robust actuator performance throughout the mission.

Given these complexities, developing a control allocation algorithm that is both computationally efficient for real-time implementation on embedded systems and capable of handling asymmetric control input limitations, actuator coupling, thermal load considerations, and state-dependent constraints is a significant challenge. This work presents an iterative control allocation approach explicitly tailored for hypersonic glide vehicles like the GHGV-2. We begin by introducing the general control allocation problem of overactuated systems and then discuss the specific control allocation problem for HGVs, highlighting the unique challenges posed by asymmetric control input limitations, significant coupling between control effectors, thermal load considerations, and state-dependent actuator constraints due to varying flight conditions. Based on the described problem formulation, we present our proposed method that addresses these challenges by directly incorporating state-dependent control input limits and thermal load considerations into the control allocation algorithm. It iteratively adjusts the control inputs to ensure that the required control moments are achieved while respecting all constraints and minimizing thermal stress on the control surfaces. The algorithm is designed to be computationally efficient, making it suitable for real-time implementation on the vehicle's embedded systems. We evaluate the effectiveness of the proposed approach through numerical simulations, demonstrating its advantages in handling the complex requirements of hypersonic glide vehicle control allocation. The results show that our method provides a reliable solution that balances computational efficiency with the ability to manage asymmetric constraints, actuator coupling, and thermal considerations.

II. General Control Allocation Problem

We define the following nonlinear control-affine system:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) + \mathbf{g}(\mathbf{x}(t))\mathbf{u}(t) \quad (1)$$

$$\mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t)) \quad (2)$$

where $\mathbf{x}(t) \in \mathbb{R}^n$, $\mathbf{u}(t) \in \mathbb{R}^m$, and $\mathbf{y}(t) \in \mathbb{R}^p$. The function $\mathbf{f}: \mathbb{R}^n \rightarrow \mathbb{R}^n$ represents a Lipschitz continuous system function, $\mathbf{h}: \mathbb{R}^n \rightarrow \mathbb{R}^p$ a Lipschitz continuous output function, and $\mathbf{g}(\mathbf{x}(t)) \in \mathbb{R}^{n \times m}$ is referred to as the control effectiveness matrix. In the field of flight control, inversion-based control methodologies, such as NDI, are commonly used to impose desired closed-loop behavior on the MIMO dynamics [13]. To do so, the time derivatives of the introduced nonlinear dynamics from Eqs. (1) and (2) are derived and reformulated using Lie derivative notation [14]:

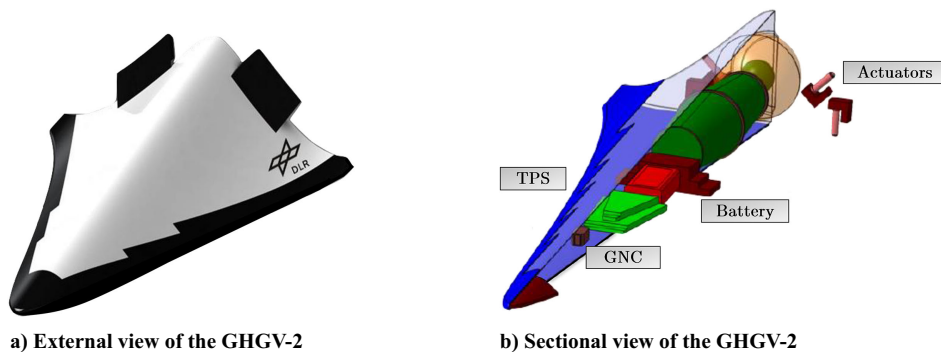


Fig. 1 The DLR Generic Hypersonic Glide Vehicle 2 concept.

$$\begin{aligned}
\dot{y}(t) &= \frac{\partial h}{\partial x} \dot{x}(t) = \frac{\partial h}{\partial x} [f(x(t)) + g(x(t))u(t)] \\
&= \frac{\partial h}{\partial x} f(x(t)) + \frac{\partial h}{\partial x} g(x(t))u(t) \\
&= L_f h(x(t)) + L_g h(x(t))u(t) \\
&= F(x(t)) + G(x(t))u(t)
\end{aligned} \quad (3)$$

A relationship between the input and the output can be established if $G(x(t)) \neq 0$. For cases in which $G(x(t)) = 0$, the relationship must be further derived until the system input affects the system output. The number of time derivations needed is called the relative degree and is represented by r [15]. For the sake of brevity, we consider here an example of a first-order system. For such a system with a relative degree of one, the NDI-based control law can be stated as

$$u(t) = G(x(t))^{-1}[\nu(t) - F(x(t))] \quad (4)$$

with $\nu(t) \in \mathbb{R}^o$ being a virtual control input vector created by a linear controller to impose the desired closed-loop characteristics on the controlled variables such that $\nu(t) = -Ky(t)$, with K being a Hurwitz gain matrix. It is evident that the inversion of the control effectiveness matrix $G(x(t))$, from now on for simplicity referred to as B , maps a desired $\nu(t)$ into the control space $\mathbb{R}^m$. In cases in which $m = n$ and $\text{rank}(B) = n$, this leads to a unique solution. In the case of overactuation, where $m > n$, the inversion of the matrix is not easily available, since the mapping into the control space is not unique and, mathematically speaking, is underdetermined. In the flight control literature, this problem is often referred to as the control allocation problem [16]. In general, the control allocation problem can be interpreted as a constrained optimization problem since magnitude and rate limitations on each actual available control effector $u_i(t)$ need to be considered. The rate limits are often defined as symmetric lower and upper bounds for each actuator, leading to the following rate saturation function for each actuator i :

$$R(\dot{u}_i(t)) = \begin{cases} \dot{u}_{\max,i}, & \text{if } \dot{u}_i(t) > \dot{u}_{\max,i}, \\ \dot{u}_{\min,i}, & \text{if } \dot{u}_i(t) < \dot{u}_{\min,i}, \\ \dot{u}_i(t), & \text{else} \end{cases} \quad (5)$$

Since a small and constant time step T for the control computation can be assumed, the rate limits can be interpreted as additional rate-dependent magnitude limits $\bar{u}_i(t)$ relative to the current deflection state $u_i(t)$, modeled as a zero-order hold limit:

$$\bar{u}_i(t) = u_i(t - T) + R(\dot{u}_i(t))T \quad (6)$$

Based on that, the magnitude limitations can be defined as individual lower and upper bounds for each effector i . The magnitude-constrained control inputs are represented using the following magnitude saturation function:

$$S(u_i(t), \dot{u}_i(t)) = \begin{cases} \min(u_{\max,i}, \bar{u}_i(t)), & \text{if } u_i(t) \geq \min(u_{\max,i}, \bar{u}_i(t)), \\ \max(u_{\min,i}, \bar{u}_i(t)), & \text{if } u_i(t) < \max(u_{\min,i}, \bar{u}_i(t)), \\ u_i(t), & \text{else} \end{cases} \quad (7)$$

This leads to the following formulation of the feasible set of control inputs U :

$$U := \{u(t) \in \mathbb{R}^m \mid \forall i \in [1, m]: \max(u_{\min,i}, \bar{u}_i(t)) \leq u_i(t) \leq \min(u_{\max,i}, \bar{u}_i(t))\} \quad (8)$$

The described admissible control input space for a static toy example with two magnitude- and rate-constrained control inputs u_1 and u_2 is given in Fig. 2.

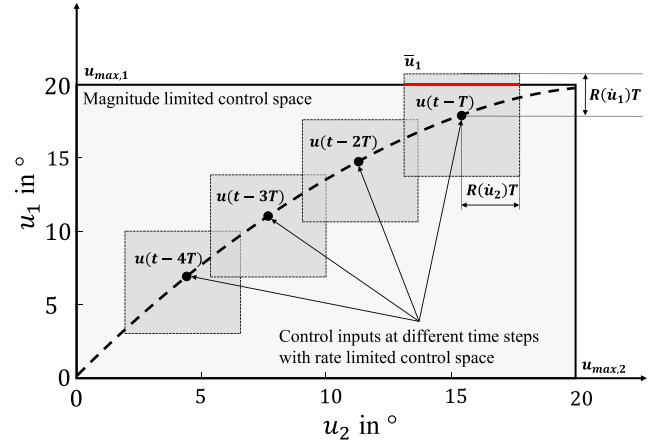


Fig. 2 Illustration of the discussed control allocation problem with considered magnitude and rate limits for a system with two control inputs u_1 and u_2 .

Based on the feasible set of control inputs, a feasible physical control volume D can be constructed as

$$D := \{\nu(t) \in \mathbb{R}^o \mid \nu(t) = Bu(t), u(t) \in U\} \quad (9)$$

In the context of flight control, the resulting convex set D is often referred to as the attainable moment set (AMS) [17]. An example of an AMS, in which dynamic effects coming from the rate limits are neglected, for a hypersonic glide vehicle with four control effectors ($m = 4$) is shown in Fig. 3.

The resulting general control allocation problem, considering magnitude and rate limits on the control input, can be formulated as the following constrained optimization problem:

$$\begin{aligned}
&\arg \min_{u(t) \in \mathbb{R}^m} \|u(t)\|_p \\
&\text{subject to } \nu(t) = Bu(t), \\
&\quad \max(u_{\min}, \bar{u}(t)) \leq u(t) \leq \min(u_{\max}, \bar{u}(t))
\end{aligned} \quad (10)$$

with p being a real number used to define the considered norm. The choice of the norm has a direct influence on the resulting control effort distribution and how the available actuator suite is used. In later sections of this work, the ℓ_2 -norm is adopted to promote a balanced distribution of control effort for the GHGV-2 and to enable

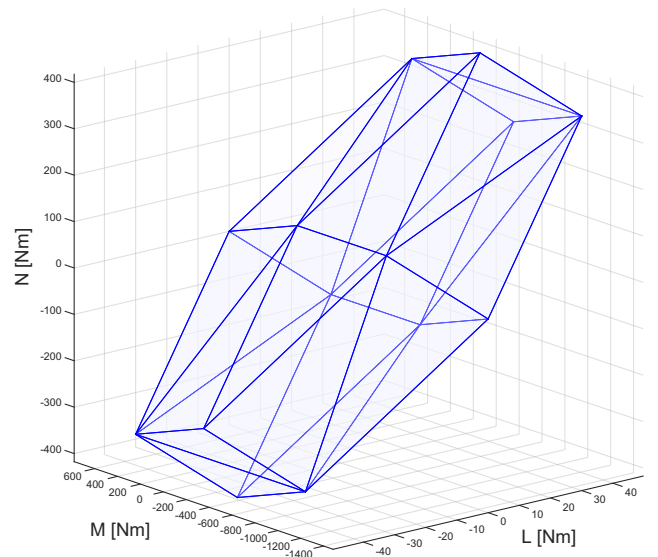


Fig. 3 An exemplary plot of an attainable moment set of the GHGV-2 for an operating point at $M = 8$ and $H = 30$ km.

an efficient least-squares formulation of the control allocation problem. Alternative formulations based on the ℓ_1 -norm or the ℓ_∞ -norm are, in many application cases, equally well suited to handle actuator limitations, coupling effects, and additional preference criteria, and have been successfully applied and tested in similar flight control applications [18]. Nevertheless, in order to provide a more comprehensive context for the choice of the ℓ_2 -norm adopted in this work, the influence of different norm selections on the solution of the control allocation problem is discussed in the following subsection.

A. Role of Norms in Control Allocation

By interpreting the optimization problem from a geometric perspective, different norms induce different level sets in the control input space, which interact with the admissible control input set $\mathcal{U}$ and the AMS $\mathbf{D}$ in distinct ways. As a result, the choice of norm implicitly encodes a preference for certain control distribution strategies and affects how a demanded virtual control input $\mathbf{v}(t)$ is distributed among the available actuators.

The minimization of the Euclidean ℓ_2 -norm corresponds to a least-squares formulation and leads to a quadratic programming (QP) control allocation problem. The ℓ_2 -norm promotes a smooth and balanced distribution of control effort across all available actuators and yields a unique solution whenever the feasible set is non-empty [19]. Due to its favorable numerical properties and balanced way of distributing the control effort among the available actuators, ℓ_2 -based control allocation has become a standard choice in many flight control applications [5,6]. In contrast, the ℓ_1 -norm promotes sparsity in the control input vector and tends to concentrate control effort on a reduced subset of actuators. This property can be advantageous in systems where certain effectors are preferred due to efficiency or authority. Control allocation problems based on the ℓ_1 -norm can be formulated as linear programs and efficiently solved using simplex-based or interior-point methods. Linear programming (LP) control allocation approaches have been successfully demonstrated in flight-tested applications, including reentry vehicles with actuator constraints [18,20,21]. The ℓ_∞ -norm minimizes the maximum absolute actuator deflection within the actuator suite and can

be interpreted as a worst-case input allocation strategy. This formulation is particularly suitable when limiting peak actuator usage is critical. Similar to the ℓ_1 -norm, ℓ_∞ -based control allocation problems can be cast as linear programs [16,22,23].

It is worth noting that, for low-dimensional allocation problems or operating regimes in which actuator constraints remain inactive, some of the listed norm choices may yield very similar or, in certain cases, identical solutions. Pronounced differences typically arise in strongly constrained or highly overactuated systems, where the geometry of the feasible control set interacts nontrivially with the selected norm.

To further illustrate the influence of the chosen norm on the solution of constrained control allocation problems, a simplified static example is considered and visualized in Fig. 4. The problem involves a system with a two-dimensional control input vector $\mathbf{u} = [u_1, u_2]^T$, whose admissible set is bounded by magnitude constraints $\mathbf{u}_{\min} \leq \mathbf{u} \leq \mathbf{u}_{\max}$ and indicated by the dashed red box in the figure. The control effectiveness matrix is defined as $\mathbf{B} = [1, -1]$, and the demanded virtual control input is set to $\mathbf{v} = 1$. The set of all control inputs satisfying the virtual control demand $\mathbf{v} = \mathbf{B}\mathbf{u}$ forms a straight line in the (u_1, u_2) -plane and is shown as the dashed black line in Fig. 4. Two representative cases are examined. In case 1, the magnitude bounds are chosen as $\mathbf{u}_{\min} = [-2, -2]^T$ and $\mathbf{u}_{\max} = [2, 2]^T$, such that the admissible region fully contains the unconstrained minima of the respective cost functions for the commanded $\mathbf{v}$ of the case. Consequently, the input constraints are inactive, and the solution is determined solely by the norm-induced objective. In case 2, the admissible region is deliberately reduced by selecting $\mathbf{u}_{\min} = [0, -1.5]^T$ and $\mathbf{u}_{\max} = [0.5, 1.5]^T$, such that only a limited segment of the solution line remains feasible. In this case, the input constraints are active and directly influence the resulting allocation. For each case, three objective functions of the form $J = \|\mathbf{u}\|_p$ are considered, where the choice of p corresponds to the ℓ_2 -, ℓ_1 -, or ℓ_∞ -norm. The corresponding optimization problems are formulated using the appropriate numerical frameworks for each norm, as discussed in the previous part of this section. In all cases, the equality constraint $\mathbf{v} = \mathbf{B}\mathbf{u}$ as well as the actuator magnitude limits

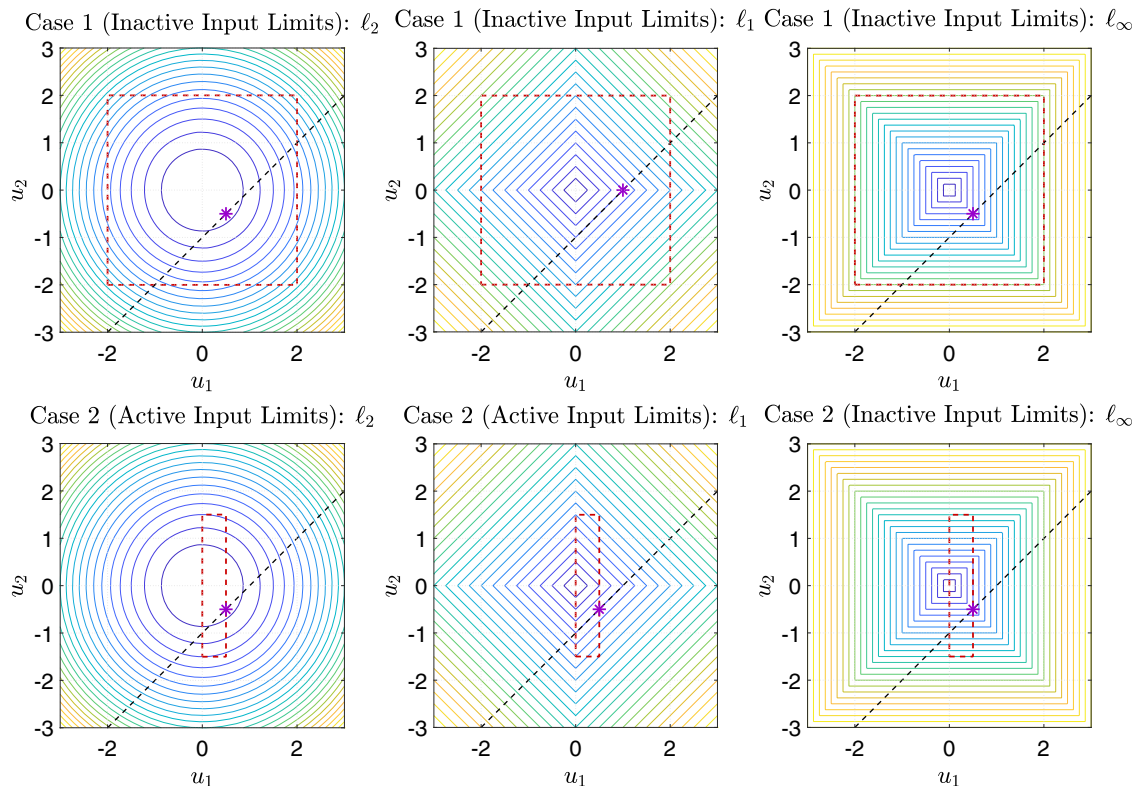


Fig. 4 Results of a comparative analysis illustrating the influence of different norm choices on control allocation with inactive and active input constraints.

are enforced as hard constraints within the optimization, as presented in Eq. (10). The resulting cost contours are visualized in Fig. 4, enabling a direct geometric interpretation of the norm-induced allocation behavior.

For the ℓ_2 -norm, the cost contours are circular, and the optimal solution corresponds to the feasible point on the solution line that minimizes the Euclidean distance to the origin. In case 1, this yields the solution $\mathbf{u} = [0.5, -0.5]^T$, resulting in a balanced distribution of control effort between the two inputs. In case 2, the unconstrained minimum lies on the boundary of the admissible set, and the QP-based implementation is able to identify this solution, again resulting in $\mathbf{u} = [0.5, -0.5]^T$. In contrast, the ℓ_1 -norm yields diamond-shaped cost contours, corresponding to squares that are rotated by 45 deg with respect to the coordinate axes in the (u_1, u_2) -plane. As a consequence, the associated optimization problem exhibits a sparsity-promoting behavior, tending to concentrate the control effort on a single actuator whenever this is compatible with the constraints. In case 1, the ℓ_1 -based allocation yields the solution $\mathbf{u} = [1, 0]^T$, where the entire control effort is assigned to u_1 while u_2 remains inactive. This behavior highlights the tendency of ℓ_1 -based approaches to reach the magnitude saturation of the most effective actuators at an earlier stage, which can be undesirable in applications where actuator saturation is problematic or where a balanced usage of available actuators is preferred. Although this effect can be reduced through appropriate conditioning and weighting of the optimization problem, as discussed in [18], the underlying numerical tendency remains inherent to the ℓ_1 -norm formulation. In case 2, the active input limits significantly restrict the admissible solution space, such that the unconstrained ℓ_1 -minimum becomes infeasible. Nevertheless, the LP-based allocation routine is able to identify the same feasible solution as obtained with the other norms, namely $\mathbf{u} = [0.5, -0.5]^T$. This illustrates that, as long as the solution space is nonempty, all considered optimization formulations are capable of identifying admissible solutions, even when the cost function alone does not yield a unique or well-conditioned minimum. The ℓ_∞ -norm leads, similar to the ℓ_1 -norm, to square-shaped cost contours but aligned with the coordinate axes. For the considered example, the resulting solution is identical to the ℓ_2 -based solution in both scenarios, yielding $\mathbf{u} = [0.5, -0.5]^T$ in case 1 and case 2. This behavior can be attributed to the well-conditioned structure of the considered control effectiveness matrix, where both control inputs contribute with comparable magnitude to the virtual control demand. Under such conditions, minimizing the peak control effort does not introduce a strong bias toward a particular actuator, and the ℓ_∞ - and ℓ_2 -based solutions coincide.

A notable distinction between the considered norm formulations arises from their smoothness properties. While the ℓ_2 -norm defines a smooth and continuously differentiable objective function, both the ℓ_1 - and ℓ_∞ -norms are nonsmooth and exhibit sharp edges in their level sets [19]. As demonstrated by the simplified example, these geometric properties directly influence the resulting allocation behavior. In particular, the ℓ_1 -norm exhibits a pronounced tendency to drive highly effective actuators into saturation, even in cases where alternative feasible solutions with more balanced actuator usage exist. Although this sparsity-promoting behavior can be beneficial in certain applications, it is generally undesirable in flight control systems where early saturation of individual actuators may reduce robustness or limit available control authority. The ℓ_∞ -norm, on the other hand, in some cases yields solutions similar to those obtained with the ℓ_2 -norm when the control effectiveness of each actuator is comparable, as observed in the presented example. However, due to its nonsmooth cost structure similar to ℓ_1 , ℓ_∞ -based formulations may still lead to nonsmooth allocation behavior and abrupt changes in the optimal solution when the active set changes [24,25]. In dynamic settings, such effects can result in rapid switching of actuator commands, input chattering, and frequent operation near rate limits, which is undesirable from both a performance and energy conservation perspective. While the inclusion of explicit rate constraints and adjustments to the numerical optimization structure can mitigate these effects to some extent, the

underlying tendency toward nonsmooth allocation remains inherent to the formulation.

Based on these considerations, an ℓ_2 -norm-based formulation is adopted in this work for the GHGV-2 application. The ℓ_2 -norm promotes a balanced distribution of control effort in accordance with actuator effectiveness and exhibits favorable smoothness properties in the presence of actuator constraints. Building on this formulation, a variety of control allocation algorithms that leverage the ℓ_2 -norm have been proposed and applied in the literature. These approaches differ in how the control allocation problem is structured and simplified and in how actuator limitations are addressed within the allocation process. The following subsection provides a brief overview of commonly used ℓ_2 -based control allocation algorithms and discusses their respective advantages and limitations in the context of hypersonic glide vehicle applications.

B. Common ℓ_2 -Based Control Allocation Algorithms

This section presents a brief overview of three commonly used algorithms in control allocation, highlighting their benefits and limitations in distributing virtual control input commands for the considered class of overactuated hypersonic systems.

1. Pseudo-Inverse-Based Control Allocation

The majority of the control allocation problems are solved by pseudo-inverse-based control allocation (PICA), using the Moore–Penrose pseudo-inverse, which implicitly solves the following unconstrained least-squares problem for the allocation of the incremental virtual control input vector ν :

$$\begin{aligned} \min_{\mathbf{u} \in \mathbb{R}^m} \quad & \|\mathbf{u}\|_2 \\ \text{subject to} \quad & \nu = \mathbf{B}\mathbf{u} \end{aligned} \quad (11)$$

with $\|\cdot\|_2$ being the ℓ_2 -norm. The defined optimization problem has the following closed-form solution:

$$\mathbf{u} = \mathbf{B}^+ \nu = \mathbf{B}^T [\mathbf{B}\mathbf{B}^T]^{-1} \nu \quad (12)$$

The major drawback of the generalized inverse approach is that the constraints on the control input $\mathbf{u}$ are neglected, leading to the computation of possibly infeasible control inputs.

2. Redistributed Pseudo-Inverse-Based Control Allocation

An alternative approach to solve the constrained control allocation problem was proposed by the authors in [26] and is called redistributed pseudo-inverse-based control allocation (RPICA). The core idea of the methodology is to cope with the problem of limited control input authority by iterating over the AMS until a feasible control input command is found. The algorithm starts by generating a first solution for a control vector candidate using the pseudo-inverse:

$$\mathbf{u}^{(1)} = \mathbf{B}^+ \nu^{(0)} \quad (13)$$

with $\nu^{(0)}$ being the initially desired commanded virtual control input vector from the controller. In the case where no elements of $\mathbf{u}^{(1)}$ are saturated and Eq. (13) delivers the desired virtual control input vector $\nu^{(0)}$, $\mathbf{u}^{(1)}$ is the final solution. In cases where at least one element of $\mathbf{u}^{(1)}$ conflicts with the available control limitations and the desired virtual control input vector is not generated, the process continues. The achievable control input is defined as

$$\nu^{(1)} = \mathbf{B}\mathbf{S}(\mathbf{u}^{(1)}) \quad (14)$$

with $\mathbf{S}(\mathbf{u}^{(1)})$ being a simplified version of the magnitude saturation function defined in Eq. (7), since in most of the formulations in the existing literature the rate limits are neglected. The control effectiveness matrix $\mathbf{B}$ is then sparsified such that the matrix columns that correspond to the saturated actuators are zeroed.

Also, the limits are adjusted to make sure that the final solution does not exceed the actual bounds:

$$\mathbf{u}_{\max}^{(j)} = \mathbf{u}_{\max}^{(j-1)} - \mathbf{u}^{(j)} \quad (15)$$

$$\mathbf{u}_{\min}^{(j)} = \mathbf{u}_{\min}^{(j-1)} - \mathbf{u}^{(j)} \quad (16)$$

Now, the remaining pseudo-control vector is computed as

$$\boldsymbol{\nu}^{(j)} = \boldsymbol{\nu}^{(j-1)} - \mathbf{B}_s \mathbf{S}(\mathbf{u}^{(j)}) \quad (17)$$

with $\mathbf{B}_s$ being the sparsified control effectiveness matrix. In the next iteration step, the remaining pseudo-control vector is allocated using $\mathbf{B}_s$ and the adjusted limits as defined in Eqs. (15) and (16). That procedure is continued until either the desired pseudo-control vector has been achieved or the maximum number of iterations has been reached. In both cases, the final control vector is the sum of the increments of all iteration steps.

For MIMO systems, it is preferable to keep the directionality of the unconstrained input vector $\mathbf{u}^{(j)}$. To make that possible, the authors in [27] proposed the redistributed scaling pseudo-inverse control allocation (RSPICA) algorithm. The saturated control input vector is scaled within the method to ensure perfect direction alignment with the desired pseudo-control vector. The used scalar a is chosen such that the element that exceeds its respective limit the most is saturated, while the others are not:

$$\mathbf{u}_a^{(j)} = \min \left(\underbrace{1, \frac{l(u_1^{(j-1)})}{u_1^{(j)}}, \frac{l(u_2^{(j-1)})}{u_2^{(j)}}, \dots, \frac{l(u_m^{(j-1)})}{u_m^{(j)}}}_{a} \right) \cdot \mathbf{u}^{(j)} \quad (18)$$

with

$$l(u_i^{(j-1)}) = \begin{cases} u_{\max,i}^{(j-1)}, & \text{if } u_i^{(j)} > 0, \\ u_{\min,i}^{(j-1)}, & \text{if } u_i^{(j)} < 0 \end{cases}$$

Both RPICA and its scaled variant, RSPICA, are advantageous for implementation in embedded systems with limited computational resources due to their simplicity. RSPICA, in particular, was designed to maintain input vector directionality in MIMO systems under saturation conditions by scaling the requested vector with a factor a , thereby ensuring that control input constraints are respected. However, despite these advancements, both methods have notable limitations. Their reliance on the ℓ_2 -based least-squares approach, inherent in the Moore–Penrose pseudo-inverse, can result in an inefficient use of the AMS when asymmetric input constraints and strong inter-directional control authority of the control effectors are present. Additionally, for systems with asymmetric input limits where the lower input bound is zero, RSPICA may produce zero-scaled solutions, resulting in trivial control commands. These challenges are particularly problematic for certain classes of hypersonic vehicles.

3. Quadratic-Programming-Based Control Allocation

The constrained control allocation problem can also be solved using a quadratic programming-based control allocation (QPCA) approach. Examples of this can be found in the work presented in [5,28]. To discuss this, we introduce the general formulation of a quadratic programming problem:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \frac{1}{2} \mathbf{x}^T \mathbf{P} \mathbf{x} + \mathbf{q}^T \mathbf{x} \\ \text{subject to} \quad & \mathbf{A} \mathbf{x} \leq \mathbf{b} \end{aligned} \quad (19)$$

with $\mathbf{A} \mathbf{x} \leq \mathbf{b}$ being a linear matrix inequality to constrain the solution of $\mathbf{x}$. In the context of control allocation, we often aim to solve the following constrained weighted least-squares problem:

$$\begin{aligned} \min_{\mathbf{u}} \quad & \frac{1}{2} \|\mathbf{B} \mathbf{u} - \boldsymbol{\nu}\|_{\mathbf{W}}^2 \\ \text{subject to} \quad & \mathbf{A}_u \mathbf{u} \leq \mathbf{b}_u \end{aligned} \quad (20)$$

with $\mathbf{A}_u \mathbf{u} \leq \mathbf{b}_u$ reflecting control limits, and $\|\mathbf{y}\|_{\mathbf{W}} = \sqrt{\mathbf{y}^T \mathbf{W} \mathbf{y}}$ being a weighted form of the Euclidean norm for a vector $\mathbf{y}$. We can introduce a conversion of the least-squares cost function to a quadratic programming objective by

$$\begin{aligned} \|\mathbf{B} \mathbf{u} - \boldsymbol{\nu}\|_{\mathbf{W}}^2 &= (\mathbf{B} \mathbf{u} - \boldsymbol{\nu})^T \mathbf{W} (\mathbf{B} \mathbf{u} - \boldsymbol{\nu}) \\ &= \mathbf{u}^T \mathbf{B}^T \mathbf{W} \mathbf{B} \mathbf{u} - 2 \mathbf{u}^T \mathbf{B}^T \mathbf{W} \boldsymbol{\nu} + \boldsymbol{\nu}^T \mathbf{W} \boldsymbol{\nu} \end{aligned} \quad (21)$$

Since the term $\boldsymbol{\nu}^T \mathbf{W} \boldsymbol{\nu}$ is constant and independent of $\mathbf{u}$, it can be discarded. This yields

$$\|\mathbf{B} \mathbf{u} - \boldsymbol{\nu}\|_{\mathbf{W}}^2 \propto \mathbf{u}^T \mathbf{B}^T \mathbf{W} \mathbf{B} \mathbf{u} - 2(\mathbf{B}^T \mathbf{W} \boldsymbol{\nu})^T \mathbf{u} \quad (22)$$

Multiplying by 1/2 gives the final quadratic form:

$$\frac{1}{2} \mathbf{u}^T \mathbf{B}^T \mathbf{W} \mathbf{B} \mathbf{u} - (\mathbf{B}^T \mathbf{W} \boldsymbol{\nu})^T \mathbf{u} \quad (23)$$

which corresponds to a quadratic programming problem with

$$\mathbf{P} = \mathbf{B}^T \mathbf{W} \mathbf{B}, \quad \mathbf{q} = -\mathbf{B}^T \mathbf{W} \boldsymbol{\nu} \quad (24)$$

The constrained QPCA problem can therefore be formulated as

$$\begin{aligned} \arg \min_{\mathbf{u} \in \mathbb{R}^m} \quad & \frac{1}{2} \mathbf{u}^T \mathbf{B}^T \mathbf{W} \mathbf{B} \mathbf{u} - (\mathbf{B}^T \mathbf{W} \boldsymbol{\nu})^T \mathbf{u} \\ \text{subject to} \quad & \mathbf{A}_u \mathbf{u} \leq \mathbf{b}_u \end{aligned} \quad (25)$$

with $\mathbf{A}_u = [\mathbf{I}_{m \times m}, -\mathbf{I}_{m \times m}]^T$ and $\mathbf{b}_u = [\min(u_{\max,i}, \bar{u}_i), -\max(u_{\min,i}, \bar{u}_i)]^T$. The formulated QPCA problem can be solved using a variety of methods, such as interior-point, active-set, or simplex methods [6]. The advantage of the method is that it is able to provide solutions to complex control allocation problems. However, a significant drawback of the approach is the high computational complexity and, as a result, the lack of real-time guarantees for certain embedded systems integrated into aerial vehicles, for example, the system integrated into the GHGV-2.

III. Problem Formulation

The considered GHGV-2 is modeled as a rigid body, and the nonlinear flight dynamics of the HGV are based on classical Newtonian mechanics. Figure 5 displays the components X , Y , Z

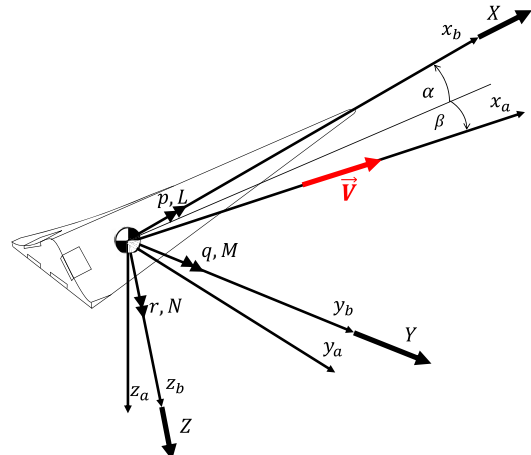


Fig. 5 Sketch of external forces and moments acting on the GHGV-2 concept.

of the total external force vector and the components L, M, N of the total external moment vector expressed in the body-fixed frame of the vehicle. As detailed in Sec. II, the application of suitable control allocation algorithms is essential for the overactuated GHGV-2 in both exoatmospheric and endoatmospheric flight phases.

Due to the brevity of this paper, we limit our discussion to the control allocation problem during the endoatmospheric flight phase. In this context, only the aerodynamic and gravitational forces and moment effects are relevant for the considered problem and the results presented later.

Figure 6 illustrates the available aerodynamic control surfaces and their corresponding deflections: upper-left fin u_1 , upper-right fin u_2 , lower-left fin u_3 , and lower-right fin u_4 [29]. This leads to the following control input vector:

$$\mathbf{u}(t) = \begin{bmatrix} u_1(t) & u_2(t) & u_3(t) & u_4(t) \end{bmatrix}^T \quad (26)$$

The aerodynamic coefficient derivatives with respect to the control surface deflections can be fairly well approximated by linear terms. To account for the dependencies on the current flight state $\mathbf{x}(t)$ and the operating point $\sigma(t)$, the control effectiveness matrix $\mathbf{B}(\mathbf{x}(t), \sigma(t))$ is continuously recomputed using an onboard plant model and sensor measurements. However, it is assumed to be constant within each time step. The control effectiveness matrix $\mathbf{B}(\mathbf{x}(t), \sigma(t))$ with respect to a virtual control command input vector $\nu(t) = [\nu_x(t), \nu_y(t), \nu_z(t)]^T$ is thus defined as

$$\mathbf{B}(\mathbf{x}(t), \sigma(t)) = \begin{bmatrix} \frac{\partial \nu_x}{\partial u_1} & \frac{\partial \nu_x}{\partial u_2} & \frac{\partial \nu_x}{\partial u_3} & \frac{\partial \nu_x}{\partial u_4} \\ \frac{\partial \nu_y}{\partial u_1} & \frac{\partial \nu_y}{\partial u_2} & \frac{\partial \nu_y}{\partial u_3} & \frac{\partial \nu_y}{\partial u_4} \\ \frac{\partial \nu_z}{\partial u_1} & \frac{\partial \nu_z}{\partial u_2} & \frac{\partial \nu_z}{\partial u_3} & \frac{\partial \nu_z}{\partial u_4} \end{bmatrix} \quad (27)$$

with $\mathbf{B}(\mathbf{x}(t), \sigma(t)) \in \mathbb{R}^{3 \times 4}$. As it is a nonsquare matrix, it cannot be inverted directly. A major challenge in hypersonic flight, often neglected in the literature, is the dependence of actuator magnitude and rate limits not only on the instantaneous flight state $\mathbf{x}(t)$ but also on the operating point $\sigma(t)$. During high-speed atmospheric operations, high dynamic pressure induces counterforces on actuators, limiting the achievable deflections and rates. Nonlinear aerodynamic effects further complicate matters, with asymmetric influences on upper and lower surfaces requiring distinct actuator constraints. Additionally, operating-point-dependent thermal loads must be considered to minimize heating, reduce the burden on the TPS, and lower the infrared signature of the vehicle during operations. These effects are illustrated in Fig. 7. In this figure, the red arrow $\mathbf{V}$ represents the resulting velocity vector of the vehicle as it moves at high speed through the atmosphere. The blue arrows depict the airflow vector fields around the vehicle, shaping the aerodynamic forces and heating effects on the control surfaces. Control surface i , modeled as a flapping plate, is characterized by a deflection angle u_i and a deflection rate $\dot{u}_i$. The red gradient along the control surface indicates aerodynamic heating generated by the

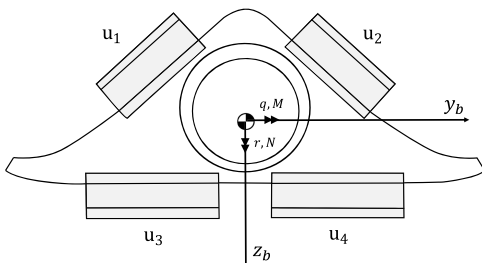


Fig. 6 Rear view of the GHGV-2 showing available control effectors during endoatmospheric operations.

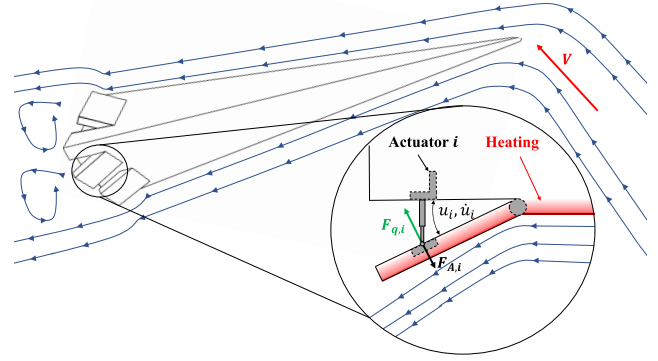


Fig. 7 Illustration of an overactuated hypersonic glide vehicle during high-speed atmospheric operations with high dynamic pressure and thermal loads.

vehicle's hypersonic speed, causing significant heat flux τ_i on the surface. These heating effects must be considered during operation to avoid material degradation and structural failure of the control surfaces and to minimize the infrared signature in defense scenarios. The green arrow $F_{q,i}$ represents the normal of the aerodynamic force acting on the surface, which is highly dependent on the deflection angle and the dynamic pressure $q = 0.5\rho V^2$ acting on the control surface area $S_{u,i}$, with ρ being the atmospheric density and V the resulting velocity. The actuator force $F_{A,i}$ is responsible for controlling the position of the surface, working against $F_{q,i}$, especially under high dynamic pressure conditions. Under these conditions, the actuator's ability to adjust the deflection angle and rate is sometimes restricted, leading to cases in which the deflection angle u_i is limited such that the resulting dynamic pressure force $F_{q,i}$ exceeds the maximum force that the actuator can provide.

To address these high Mach number effects, we define the control allocation problem as a constrained optimization problem with state- and operating-point-dependent magnitude and rate limitations. The rate saturation function for each actuator i is defined as

$$R(\dot{u}_i(t), \mathbf{x}(t), \sigma(t)) = \begin{cases} \dot{u}_{\max,i}(\mathbf{x}(t), \sigma(t)), & \text{if } \dot{u}_i(t) > \dot{u}_{\max,i}(\mathbf{x}(t), \sigma(t)), \\ \dot{u}_{\min,i}(\mathbf{x}(t), \sigma(t)), & \text{if } \dot{u}_i(t) < \dot{u}_{\min,i}(\mathbf{x}(t), \sigma(t)), \\ \dot{u}_i(t), & \text{else} \end{cases} \quad (28)$$

By modeling a rate-dependent magnitude limit $\bar{u}_i$ relative to the current deflection state $\mathbf{u}(t)$, via Eq. (6), we can define a rate- and state-/operating-point-dependent magnitude saturation function:

$$S(u_i(t), \dot{u}_i(t), \mathbf{x}(t), \sigma(t)) = \begin{cases} \min(u_{\max,i}(\mathbf{x}(t), \sigma(t)), \bar{u}_i(t)), & \text{if } u_i(t) \geq \min(u_{\max,i}(\mathbf{x}(t), \sigma(t)), \bar{u}_i(t)), \\ \max(u_{\min,i}(\mathbf{x}(t), \sigma(t)), \bar{u}_i(t)), & \text{if } u_i(t) < \max(u_{\min,i}(\mathbf{x}(t), \sigma(t)), \bar{u}_i(t)), \\ u_i(t), & \text{else} \end{cases} \quad (29)$$

Leading to the following definition of the state- and operating-point-dependent feasible set of control inputs $\mathbf{U}(\mathbf{x}(t), \sigma(t))$:

$$\mathbf{U}(\mathbf{x}(t), \sigma(t)) := \{\mathbf{u}(t) \in \mathbb{R}^m \mid \forall i \in [1, m]: \max(u_{\min,i}(\mathbf{x}(t), \sigma(t)), \bar{u}_i(t)) \leq u_i(t) \leq \min(u_{\max,i}(\mathbf{x}(t), \sigma(t)), \bar{u}_i(t))\} \quad (30)$$

A control allocation problem is formulated that not only considers the current control distribution at each time step but also includes the previous time step as a soft constraint. This leads to the following constrained state- and operating-point-dependent optimization problem:

$$\begin{aligned}
& \min_{\mathbf{u}(t) \in \mathbb{R}^m} \quad \|\mathbf{W}_m(\mathbf{u}(t) - \mathbf{u}_s(t))\|_2^2 + \|\mathbf{W}_r(\mathbf{u}(t) - \mathbf{u}(t-T))\|_2^2 \\
& \text{subject to} \quad \boldsymbol{\nu}(t) = \mathbf{B}(\mathbf{x}(t), \boldsymbol{\sigma}(t))\mathbf{u}(t), \\
& \quad \mathbf{u}(t) \in \mathbf{U}(\mathbf{x}(t), \boldsymbol{\sigma}(t))
\end{aligned} \tag{31}$$

with $\mathbf{u}_s(t)$ denoting a desired steady-state control input for a given virtual control command $\boldsymbol{\nu}(t)$. The vector $\mathbf{u}_s(t)$ represents a preferred steady-state solution of the algebraic allocation relation $\boldsymbol{\nu}(t) = \mathbf{B}(\mathbf{x}(t), \boldsymbol{\sigma}(t))\mathbf{u}(t)$ and serves as a reference point for the solution process. At this stage, $\mathbf{u}_s(t)$ is introduced as an auxiliary design quantity and is neither required to be unique nor enforced as a hard constraint. Its purpose is to encode designer preferences regarding steady-state actuator usage, for example, with respect to load distribution or thermal considerations. The specific construction of $\mathbf{u}_s(t)$ and its integration within the subsequently proposed control allocation strategy are addressed in Sec. V.B. The weighting matrix $\mathbf{W}_m$ penalizes deviations of the computed control input from the preferred steady-state solution, whereas $\mathbf{W}_r$ penalizes variations between consecutive control inputs $\mathbf{u}(t)$ and $\mathbf{u}(t-T)$, thereby promoting smooth actuator behavior. Design guidelines and problem-specific choices for the weighting matrices $\mathbf{W}_m$ and $\mathbf{W}_r$ for the control allocation problem of the GHGV-2 are provided in Sec. V.A.

In summary, the problem addressed in this work is to determine, at each time instant, a control input $\mathbf{u}(t)$ for the system defined in Eqs. (1) and (2), in accordance with Eq. (31), such that the virtual control input demand $\mathbf{B}(\mathbf{x}(t), \boldsymbol{\sigma}(t))\mathbf{u}(t) = \boldsymbol{\nu}(t)$ is satisfied whenever the command lies within the state- and operating-point-dependent actuator limits. If the command is not feasible, the control input is determined so as to minimize the deviation between the requested and achievable virtual control input while respecting the input constraints defined in Eq. (30).

IV. Integrated Overall Control Allocation Concept

As discussed in Sec. III, during atmospheric flight, HGVs face significant challenges in maintaining stability and controllability due to thermal loads and limited aerodynamic authority [30]. These constraints complicate the design of effective control strategies and require careful management of aerodynamic and thermal effects. Owing to this complexity, the nominal atmospheric flight trajectory of the HGV is typically computed offline prior to the mission. During this offline optimization process, an optimal control problem is solved to obtain a feasible reference trajectory that satisfies prescribed initial and terminal conditions as well as physical constraints [31]. The resulting optimized state reference trajectory is bounded by aerodynamic flight corridor constraints, as illustrated in Fig. 8 [30]. Exceeding the upper altitude bound leads to insufficient aerodynamic control authority due to low air density, whereas

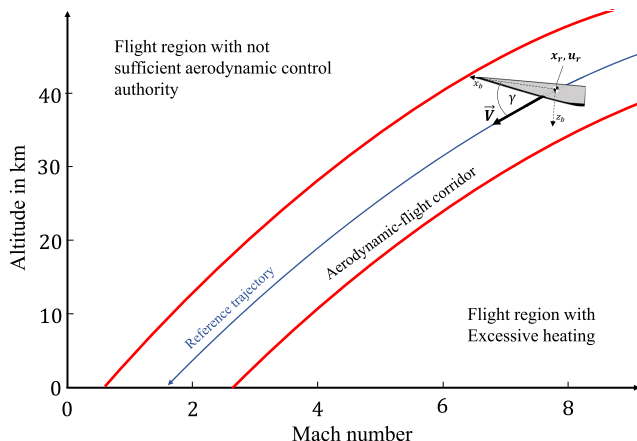


Fig. 8 Illustration of an operational flight envelope for a hypersonic glide vehicle, showing the reference trajectory bounded by upper and lower limits.

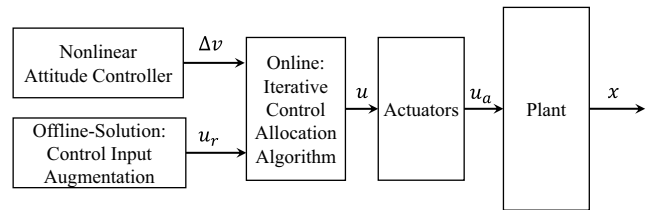


Fig. 9 Concept of the integrated control allocation system for the GHGV-2.

descending below the lower bound exposes the vehicle to excessive aerodynamic heating, potentially causing material degradation or structural failure. Based on the obtained state reference trajectory $\mathbf{x}_r(t)$, an offline baseline control input $\mathbf{u}_r(t)$ is computed using a computationally demanding but accurate QPCA method, as discussed in Sec. II.B.3, for the problem defined in Eq. (31). This offline solution incorporates all known hard and soft actuator constraints and is assumed to represent an optimal solution for a disturbance-free environment. The resulting control inputs are stored in lookup tables and serve as baseline commands during flight. Under nominal conditions, the baseline control input $\mathbf{u}_r(t)$ is sufficient to maintain the vehicle on its reference trajectory. However, due to model uncertainties, unmodeled dynamics, and external disturbances inherent in hypersonic flight, additional corrective control action is required. To address this issue, a nonlinear feedback controller generates a corrective virtual control input $\Delta\boldsymbol{\nu}(t)$ to compensate for deviations from the nominal trajectory; further information about the controller is provided in [29].

Assuming an affine structure of the control input, the total actuator command is expressed as

$$\mathbf{u}(t) = \mathbf{u}_r(t) + \Delta\mathbf{u}(t) \tag{32}$$

Here, $\Delta\mathbf{u}(t)$ represents an incremental control input required to realize the corrective virtual control command $\Delta\boldsymbol{\nu}(t)$ that is generated by the nonlinear feedback controller. Although the assumed affine nature of the control input formally allows the decomposition and superposition of the baseline command and the incremental correction input, the proposed iterative online control allocation algorithm does not allocate $\Delta\mathbf{u}(t)$ independently. Instead, it searches for the full control input $\mathbf{u}(t)$ while using the offline-optimized allocation solution $\mathbf{u}_r(t)$ as a baseline. This design choice leverages the fact that $\mathbf{u}_r(t)$ already represents a physically meaningful and near-feasible control input for the nominal operating case, thereby providing a well-conditioned starting point for the online allocation task. By incorporating $\mathbf{u}_r(t)$ directly into the allocation process, the search for a feasible control input is biased toward solutions that remain close to the nominal baseline, which reduces the likelihood of rapid input changes and convergence to poorly conditioned allocations. This is particularly important in scenarios with strongly asymmetric actuator effectiveness or state-dependent magnitude and rate constraints, where the admissible solution space may be highly restricted. The overall control allocation strategy is illustrated in Fig. 9. By combining an offline-optimized baseline solution with an online allocation mechanism operating on the full control input, the proposed architecture facilitates the real-time computation of feasible commands in the presence of disturbances and uncertainties. The design and implementation of the iterative online control allocation algorithm, which constitutes a central contribution of this work, are detailed in the following section.

V. Proposed Online Control Allocation Algorithm

In standard RPICA approaches, the problem of limited control input authority is addressed by iterating over the AMS until a feasible control input command is found [26]. During the iteration steps, the algorithm typically employs the Moore–Penrose pseudo-inverse. The main advantage of this approach is that a solution can often be obtained in a simple and computationally efficient manner, making it suitable for real-time implementation in many aerospace

applications. While it is, in principle, possible to account for actuator rate information indirectly through equivalent magnitude constraints, such considerations are typically neglected in the majority of practical RPICA implementations. Moreover, most RPICA formulations do not adequately represent asymmetric input or rate limitations, particularly when these constraints are intended to act as soft shaping mechanisms that influence the control distribution prior to constraint activation. For the class of hypersonic glide vehicles studied here, these aspects are of particular interest, rendering standard RPICA solutions suboptimal.

In the proposed iterative dynamic control allocation (IDCA) algorithm, a linear filter formulation, originally introduced in [28], is employed. Unlike the Moore–Penrose inverse, this approach explicitly accounts for both the current control distribution and the allocation of the previous time step as soft constraints [32]. This implicitly incorporates actuator rates into the allocation problem. The method provides an explicit solution to the following weighted least-squares problem:

$$\min_{\mathbf{u}(t) \in \mathbb{R}^m} \|\mathbf{W}_m(\mathbf{u}(t) - \mathbf{u}_s(t))\|_2^2 + \|\mathbf{W}_r(\mathbf{u}(t) - \mathbf{u}(t-T))\|_2^2 \quad (33)$$

subject to $\boldsymbol{\nu}(t) = \mathbf{B}\mathbf{u}(t)$

where $\boldsymbol{\nu}(t) = \mathbf{B}\mathbf{u}_r(t) + \Delta\boldsymbol{\nu}(t)$ is the virtual control input to be generated. The desired steady-state solution $\mathbf{u}_s(t)$ is specified in Sec. V.B. The algorithm is designed so that, if $\mathbf{u}_s(t)$ is feasible, the steady-state solution converges to $\mathbf{u}_s(t)$ or to a solution close to it. The weighting matrices $\mathbf{W}_m$ and $\mathbf{W}_r$ are symmetric and defined as functions of the state and control; their detailed design is discussed in Sec. V.A. Since $\mathbf{W}_m$ and $\mathbf{W}_r$ are symmetric, the combined weighting

$$\mathbf{W} = \sqrt{\mathbf{W}_m^2 + \mathbf{W}_r^2} \quad (34)$$

is nonsingular. This allows the reformulation of Eq. (33) into the simpler criterion

$$\min_{\mathbf{u}(t) \in \mathbb{R}^m} \|\mathbf{W}(\mathbf{u}(t) - \mathbf{u}_0(t))\|_2 \quad (35)$$

with

$$\mathbf{u}_0(t) = \mathbf{W}^{-2}(\mathbf{W}_m^2\mathbf{u}_s(t) + \mathbf{W}_r^2\mathbf{u}(t-T))$$

The closed-form solution of Eq. (33), expressed in the form of Eq. (35), can be written as [28]:

$$\mathbf{u}(t) = \mathbf{E}\mathbf{u}_s(t) + \mathbf{F}\mathbf{u}(t-T) + \mathbf{G}\boldsymbol{\nu}(t) \quad (36)$$

with

$$\mathbf{E} = (\mathbf{I} - \mathbf{G}\mathbf{B})\mathbf{W}^{-2}\mathbf{W}_m^2, \mathbf{F} = (\mathbf{I} - \mathbf{G}\mathbf{B})\mathbf{W}^{-2}\mathbf{W}_r^2, \mathbf{G} = \mathbf{W}^{-1}(\mathbf{B}\mathbf{W}^{-1})^+$$

The explicit solution of Eq. (36) is used within an iterative procedure to enforce actuator constraints. The algorithm is executed within a fixed controller cycle, so external variables such as $\mathbf{u}(t-T)$ and $\mathbf{x}(t)$ are treated as constant during the inner iterations. A closely related line of work was presented by Burken et al. [7], who introduced a fixed-point, quadratic-programming-based iteration scheme for control allocation. Their method shares structural similarities with the iterative procedures considered here, in that both refine the solution by reevaluating residual commands under actuator constraints. While the fixed-point scheme demonstrated promising performance for asymmetric reentry vehicle configurations and provided a rigorous QP-based foundation for handling magnitude constraints, actuator rate limits were only mentioned as a possible extension and not explicitly included. By contrast, the algorithm presented in the following integrates rate-dependent limits directly into the allocation problem, both as explicit constraints and as soft penalties in the weighted least-squares cost. Moreover,

the proposed formulation emphasizes practical implementation by illustrating how the weighting structure can incorporate operational considerations such as thermal load reduction and drag-sensitive allocation and by providing empirical evidence in the simulation results that the method can robustly handle strongly time-varying, state- and operating-point-dependent actuator constraints.

Figure 10 illustrates the IDCA process. The iterative procedure allows a maximum number of N iterations, with each step indexed by (j) . The actual number of iterations is denoted by $N_{\text{iter}} \leq N$. If a valid solution is found earlier, or if no further unsaturated actuators are available, the process terminates after N_{iter} iterations.

We define the iterative residual virtual control command as $\boldsymbol{\nu}^{(j)}$. In the first iteration step, the control input $\mathbf{u}^{(1)}$, corresponding to the initial residual $\boldsymbol{\nu}^{(0)} = \boldsymbol{\nu}(t)$, is computed using Eq. (36). If $\mathbf{u}^{(1)}$ satisfies all actuator magnitude and rate constraints and realizes the desired virtual control input $\boldsymbol{\nu}(t)$, it is accepted as the final solution. Otherwise, the iterative redistribution procedure is continued. The achieved saturated virtual control in iteration step j is

$$\boldsymbol{\nu}^{(j)} = \mathbf{B}\mathbf{S}(\mathbf{u}^{(j)}, \dot{\mathbf{u}}^{(j)}, \mathbf{x}(t)) \quad (37)$$

where $\mathbf{S}(\cdot)$ is the state- and rate-dependent saturation function [cf. Eq. (29)], $\dot{\mathbf{u}}^{(j)} = (1/T)(\mathbf{u}^{(j)} - \mathbf{u}_\tau)$, and $\mathbf{u}_\tau = \mathbf{u}(t-T)$ is the control input of the last time step. If actuators saturate, the corresponding columns of $\mathbf{B}$ are set to zero, yielding the reduced matrix $\mathbf{B}_s$. Input limits are updated at each step to ensure feasibility:

$$\mathbf{u}_{\max}^{(j)} = \mathbf{u}_{\max}^{(j-1)} - \mathbf{u}^{(j)}, \quad \mathbf{u}_{\min}^{(j)} = \mathbf{u}_{\min}^{(j-1)} - \mathbf{u}^{(j)} \quad (38)$$

$$\dot{\mathbf{u}}_{\max}^{(j)} = \dot{\mathbf{u}}_{\max}^{(j-1)} - \frac{1}{T}(\mathbf{u}^{(j)} - \mathbf{u}_\tau), \quad \dot{\mathbf{u}}_{\min}^{(j)} = \dot{\mathbf{u}}_{\min}^{(j-1)} - \frac{1}{T}(\mathbf{u}^{(j)} - \mathbf{u}_\tau) \quad (39)$$

Iterative dynamic control allocation (IDCA)

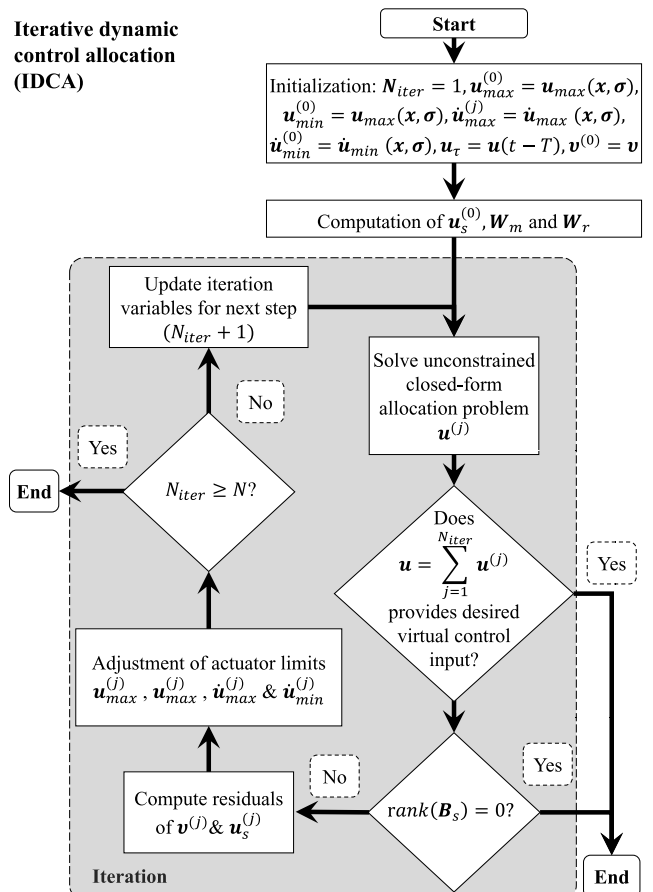


Fig. 10 Flowchart of the proposed IDCA algorithm.

A corresponding adjustment of the steady-state solution $\mathbf{u}_s$ is also applied. The residual virtual control vector is then updated as

$$\boldsymbol{\nu}^{(j)} = \boldsymbol{\nu}^{(j-1)} - \mathbf{B}_s \mathbf{S}(\mathbf{u}^{(j)}, \dot{\mathbf{u}}^{(j)}, \mathbf{x}(t)) \quad (40)$$

In the next iteration, $\boldsymbol{\nu}^{(j)}$ is reallocated using $\mathbf{B}_s$ and the updated limits. The procedure continues until the required virtual control is achieved, no unsaturated actuators remain, or N is reached. The final commanded control vector is

$$\mathbf{u}(t) = \sum_{j=1}^{N_{\text{iter}}} \mathbf{u}^{(j)} \quad (41)$$

For completeness, alongside the mathematical steps and the process visualization in Fig. 10, the algorithm is also summarized in pseudocode in Algorithm 1 to complement the formulation and facilitate implementation.

Algorithm 1: Iterative dynamic control allocation (IDCA)

```

1: Initialize:  $N_{\text{iter}} \leftarrow 1$ ,  $\mathbf{u}_{\text{max}}^{(0)} \leftarrow \mathbf{u}_{\text{max}}(\mathbf{x}, \boldsymbol{\sigma})$ ,  $\mathbf{u}_{\text{min}}^{(0)} \leftarrow \mathbf{u}_{\text{min}}(\mathbf{x}, \boldsymbol{\sigma})$ ,
 $\dot{\mathbf{u}}_{\text{max}}^{(0)} \leftarrow \dot{\mathbf{u}}_{\text{max}}(\mathbf{x}, \boldsymbol{\sigma})$ ,  $\dot{\mathbf{u}}_{\text{min}}^{(0)} \leftarrow \dot{\mathbf{u}}_{\text{min}}(\mathbf{x}, \boldsymbol{\sigma})$ ,  $\mathbf{u}_\tau \leftarrow \mathbf{u}(t - T)$ ,  $\mathbf{u}_s^{(0)} \leftarrow \mathbf{u}_s$ ,
 $\boldsymbol{\nu}^{(0)} \leftarrow \boldsymbol{\nu}$ 
2: Compute  $\mathbf{u}_s^{(0)}$ ,  $\mathbf{W}_m$ ,  $\mathbf{W}_r$ 
3: while  $N_{\text{iter}} < N$  do
4:   Compute candidate input  $\mathbf{u}^{(j)}$ 
5:   if  $\mathbf{u} = \sum_{j=1}^{N_{\text{iter}}} \mathbf{u}^{(j)}$  delivers desired  $\boldsymbol{\nu}$  then
6:     Terminate with:  $\mathbf{u}(t) = \sum_{j=1}^{N_{\text{iter}}} \mathbf{u}^{(j)}$ 
7:   end if
8:   Compute residuals of  $\boldsymbol{\nu}^{(j)}$  and  $\mathbf{u}_s^{(j)}$ 
9:   if  $\text{rank}(\mathbf{B}_s) = 0$  then
10:    Terminate with:  $\mathbf{u}(t) = \sum_{j=1}^{N_{\text{iter}}} \mathbf{u}^{(j)}$ 
11:   end if
12:   Update actuator limits  $\mathbf{u}_{\text{max}}^{(j)}$ ,  $\mathbf{u}_{\text{min}}^{(j)}$ ,  $\dot{\mathbf{u}}_{\text{max}}^{(j)}$ ,  $\dot{\mathbf{u}}_{\text{min}}^{(j)}$ 
13:   Update iteration counter  $N_{\text{iter}} \leftarrow N_{\text{iter}} + 1$ 
14: end while
15: Output: final control input  $\mathbf{u}(t)$ 

```

A. Control-Input-Based Weighting Matrices

The main motivation of the here-proposed iterative control allocation approach is to consider soft constraints on the distribution of the control efforts. To do so, a meaningful design of the weighting matrices $\mathbf{W}_m$ and $\mathbf{W}_r$ is leveraged. The magnitude weighting matrix $\mathbf{W}_m$ is designed to allocate weights to control effectors based on their current measured deflection state $u_i(t)$, as well as to use the weighting to prioritize control surfaces with lower thermal loads acting on them. The overall matrix $\mathbf{W}_m$ is defined as

$$\mathbf{W}_m = \begin{bmatrix} w_{m,1} & 0 & 0 & 0 \\ 0 & w_{m,2} & 0 & 0 \\ 0 & 0 & w_{m,3} & 0 \\ 0 & 0 & 0 & w_{m,4} \end{bmatrix} + \epsilon \mathbf{I} \quad (42)$$

with $\epsilon \in \mathbb{R}_{>0}$ being a small positive scalar and $\mathbf{I}$ a square identity matrix used to prevent the loss of needed symmetric matrix properties in the case that one or more weight elements go to zero. The elements $w_{m,i}(\mathbf{x}(t), u_i(t))$ represent, for each control surface i , a multiplicative weighting function designed to balance the two considered soft constraints in the allocation, as follows:

$$w_{m,i} = \underbrace{w_{m_D,i}(u_i(t))}_{\text{Weighting function for deflection magnitude}} \cdot \underbrace{w_{m_T,i}(\mathbf{x}(t), \boldsymbol{\sigma}(t))}_{\text{Weighting function for thermal loads}} \quad (43)$$

with $w_{m_D,i}(u_i(t))$ being the weighting function for the deflection magnitude and $w_{m_T,i}(\mathbf{x}(t), \boldsymbol{\sigma}(t))$ being the weighting function to address the effects of thermal loads for each effector i . The weighting function $w_{m_D,i}(u_i(t))$ is defined as

$$w_{m_D,i}(u_i(t)) = \frac{u_i(t)}{u_{\text{max},i}(\mathbf{x}(t), \boldsymbol{\sigma}(t))} \quad (44)$$

This formulation assigns higher weights to effectors with larger normalized deflection states, thereby discouraging further usage of actuators that are already operating close to their admissible limits. It is worth noting that any standard ℓ_2 -based control allocation, such as the Moore–Penrose pseudo-inverse, inherently promotes an even distribution of control effort. This follows directly from their objective of minimizing the Euclidean norm of the control input vector, even in the absence of explicit weighting. However, those unweighted formulations do not account for the instantaneous actuator state or for state-dependent variations in admissible control authority. As a result, actuators that are already highly deflected or close to saturation may continue to be utilized disproportionately, potentially leading to premature magnitude or rate saturation. By contrast, the proposed weighting explicitly incorporates the current actuator deflection relative to its state- and operating-point-dependent maximum admissible value. This enables a dynamic redistribution of control effort toward effectors with greater remaining authority, rather than enforcing uniform usage in an absolute sense. The effect becomes particularly important under asymmetric or state-dependent actuator limitations, such as those induced by varying aerodynamic and thermal loads in hypersonic flight, where different control surfaces may experience substantially different admissible ranges. In this way, the weighting extends the equalizing tendency of standard ℓ_2 -based allocation by making it aware of actuator availability.

To address the thermal effects via $w_{m_T,i}(\mathbf{x}(t))$ in a computationally efficient manner, we utilize insights from wind tunnel experiments conducted at the DLR Institute of Aerodynamics and Flow Technology [33]. Figure 11 shows an image of the wind tunnel test of the GHGV-2 at Mach 8.7, angle of attack $\alpha = 10$ deg, and sideslip angle $\beta = 0$ deg, with all flaps deflected by 20 deg.

The obtained results demonstrated that during endoatmospheric flight at high Mach numbers, thermal loads on the vehicle's lower side are more critical than those on the upper side due to increased heat flux. This phenomenon is attributed to the vehicle's operation at predominantly positive angles of attack, which alters the aerodynamic heating distribution and results in less severe thermal conditions on the upper part of the vehicle. Therefore, it is preferable to utilize the upper flaps as much as possible during the mission while minimizing deflection of the lower flaps. The results also confirmed that thermal loads on the flaps increase with deflection angle for both upper and lower control surfaces. It is important to note that this distribution of thermal loads between upper and lower surfaces depends on the angle of attack: at positive angles of attack, the lower flaps experience higher thermal loads, whereas at negative angles of attack, the upper flaps are more critically affected. Similarly, sideslip angles can cause imbalances between the left and right sides of the vehicle.

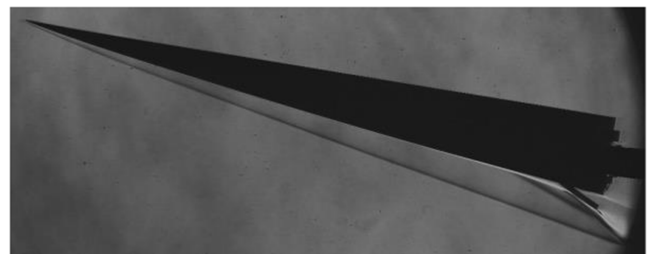


Fig. 11 Wind tunnel test of GHGV-2 with $M = 8.7$, $\alpha = 10$ deg, and $\beta = 0$ deg; all flaps deflected by 20 deg [33].

Given the complex nature of thermal modeling, directly incorporating heat flux considerations into the computation of $w_{m,i}(\mathbf{x}(t))$ was deemed impractical. However, in all relevant cases, we observed a strong positive correlation between aerodynamic drag and heat flux with increasing control deflection angles across all control surfaces. The extent of this correlation varies depending on effector position. This correlation allows us to use aerodynamic drag as a proxy for thermal loads in the weighting function. Assuming these influences can be approximated as affine, the thermal weighting for control surfaces is computed as

$$w_{m,i}(\mathbf{x}(t), \boldsymbol{\sigma}(t)) = \frac{C_{D,u_i}(\mathbf{x}(t))}{\max\{C_{D,u_1}(\mathbf{x}(t), \boldsymbol{\sigma}(t)), \dots, C_{D,u_m}(\mathbf{x}(t), \boldsymbol{\sigma}(t))\}} \quad (45)$$

where the denominator represents the maximum state-dependent aerodynamic drag coefficient among all control surfaces, and $C_{D,u_i}(\mathbf{x}(t), \boldsymbol{\sigma}(t))$ is the aerodynamic drag coefficient of control surface i . This formulation assigns higher weights to surfaces with higher drag contributions, thereby favoring allocation to surfaces with reduced thermal loads.

The rate weighting matrix $\mathbf{W}_r(\dot{\mathbf{u}}(t - T))$, designed to address the dynamic aspects of the control allocation problem by considering the rates of deflection, is defined as

$$\mathbf{W}_r = \begin{bmatrix} w_{r,1} & 0 & 0 & 0 \\ 0 & w_{r,2} & 0 & 0 \\ 0 & 0 & w_{r,3} & 0 \\ 0 & 0 & 0 & w_{r,4} \end{bmatrix} + \epsilon \mathbf{I} \quad (46)$$

where $w_{r,i}$ represents, for each control surface i , a rate-dependent weighting function defined as

$$w_{r,i} = \frac{|\dot{u}_i(t)|}{\dot{u}_{\text{crit},i}(\mathbf{x}(t), \boldsymbol{\sigma}(t))} \quad (47)$$

with the actuator rate at the initial time, when the allocation algorithm starts, given by

$$\dot{u}_i(t) = \frac{1}{T}(u_i(t) - u_i(t - T)) \quad (48)$$

The critical rate limit $\dot{u}_{\text{crit},i}(\mathbf{x}(t))$, representing the rate bound the actuator is approaching, is defined as

$$\dot{u}_{\text{crit},i}(\mathbf{x}(t)) = \begin{cases} |\dot{u}_{\text{max},i}(\mathbf{x}(t), \boldsymbol{\sigma}(t))|, & \dot{u}_i(t) \geq 0, \\ |\dot{u}_{\text{min},i}(\mathbf{x}(t), \boldsymbol{\sigma}(t))|, & \dot{u}_i(t) < 0 \end{cases} \quad (49)$$

The carefully crafted weighting matrices $\mathbf{W}_m$ and $\mathbf{W}_r$ contribute to the proposed control allocation algorithm's overall efficiency and stability, aligning with the discussed operational objectives. However, it is important to note that the proposed weighting functions $w_{m,i}$ and $w_{r,i}$ establish linear relationships with respect to the maximum and minimum values $u_{\text{max},i}(\mathbf{x}(t))$, $u_{\text{min},i}(\mathbf{x}(t))$, $\dot{u}_{\text{max},i}(\mathbf{x}(t))$, and $\dot{u}_{\text{min},i}(\mathbf{x}(t))$. Similarly, they assume an affine correlation between aerodynamic drag and thermal loads. While this linear approach ensures computational simplicity and efficiency, it inherently represents an idealized approximation. The proposed methodology simplifies the weighting function design but may not fully capture the nonlinear dependencies observed in practice.

B. Designer-Preference-Based Steady-State Solution

The proposed IDCA approach allows it to consider a desired steady-state control input vector $\mathbf{u}_s(t)$. This section discusses how a good guess for $\mathbf{u}_s(t)$ is obtained based on an a priori known baseline control input $\mathbf{u}_r(t)$ and designer preferences in the control effector usage of the hypersonic glide vehicle. Even though the computed

$\mathbf{u}_s(t)$ might not always provide a perfect solution to the requested virtual control command or, in some cases, might not be physically feasible under certain constraints, it serves as a desirable reference point or target for the redistributed control allocation algorithm. As discussed in Sec. IV, in a scenario where no external disturbances or model uncertainties are present, the baseline control input $\mathbf{u}_r(t)$ would be sufficient to maintain the HGV on its reference trajectory. However, given the inherent uncertainties and potential disturbances in hypersonic flight, the control system also requires an attitude feedback controller to provide additional corrections. This controller computes a virtual control adjustment $\Delta\mathbf{v}(t)$, which compensates for deviations from the nominal trajectory. Assuming an affine nature for the control input, the desired steady-state control input can be computed as

$$\mathbf{u}_s(t) = \mathbf{u}_r(t) + \Delta\mathbf{u}(t) \quad (50)$$

where $\Delta\mathbf{u}(t)$ represents the correction determined through the proposed online control allocation algorithm. The control input $\Delta\mathbf{u}(t)$ is generated using the following request-dependent pseudo-inverse-based approach:

$$\Delta\mathbf{u}(t) = \begin{bmatrix} \Delta u_1(t) \\ \Delta u_2(t) \\ \Delta u_3(t) \\ \Delta u_4(t) \end{bmatrix} = \mathbf{B}_C^+(\Delta\mathbf{v}(t))\Delta\mathbf{v}(t), \quad (51)$$

$$\text{with } \mathbf{B}_C(\Delta\mathbf{v}(t)) = \begin{bmatrix} \mathbf{B}_{v_x}(\Delta v_x(t)) \\ \mathbf{B}_{v_y}(\Delta v_y(t)) \\ \mathbf{B}_{v_z}(\Delta v_z(t)) \end{bmatrix}$$

Here $\mathbf{B}_C(\Delta\mathbf{v}(t))$ represents a conditionalized control effectiveness matrix. The submatrices $\mathbf{B}_{v_x}(\Delta v_x(t))$, $\mathbf{B}_{v_y}(\Delta v_y(t))$, and $\mathbf{B}_{v_z}(\Delta v_z(t))$ correspond to the three virtual control channels. The requested corrective virtual control command vector is

$$\Delta\mathbf{v}(t) = [\Delta v_x(t), \Delta v_y(t), \Delta v_z(t)]^T$$

which provides information about the required virtual control moments along the body axes.[‡] Based on flight dynamic properties and engineering preferences, certain elements of $\mathbf{B}_{v_x}(\Delta v_x(t))$, $\mathbf{B}_{v_y}(\Delta v_y(t))$, and $\mathbf{B}_{v_z}(\Delta v_z(t))$ are sparsified to force an allocation on desired control effectors.

Figure 12 illustrates how the appropriate control surfaces are identified for an example of a virtual pitch command $\Delta v_y(t)$. For

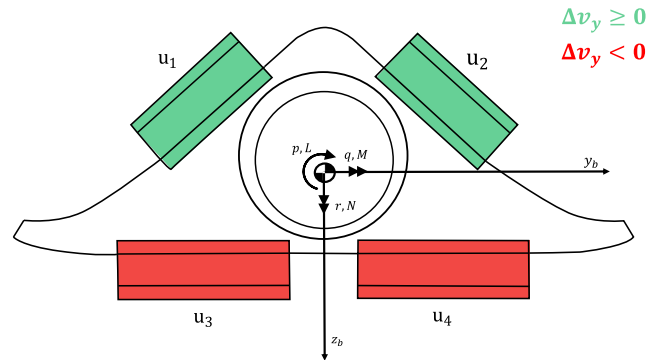


Fig. 12 Corresponding control surface selection for positive and negative incremental pitch moment Δv_y commands.

[‡]The abstract virtual controls v_x, v_y, v_z correspond to the physical roll, pitch, and yaw moments L, M, N in the body-fixed frame. We retain the abstract notation here to emphasize the generality of the formulation.

a positive pitch command $\Delta\nu_y(t) > 0$, both the upper flaps u_1 and u_2 are identified as appropriate. Conversely, for a negative pitch command $\Delta\nu_y(t) < 0$, u_3 and u_4 make the most sense from a flight dynamic perspective. This leads to the following sign-based conditionalization of $\mathbf{B}_{\nu_y}(\Delta\nu_y(t))$ in the pitch-related row:

$$\mathbf{B}_{\nu_y}(\Delta\nu_y(t)) = \begin{cases} \begin{pmatrix} \frac{\partial\nu_y}{\partial u_1} & \frac{\partial\nu_y}{\partial u_2} & 0 & 0 \end{pmatrix} & \text{if } \Delta\nu_y(t) \geq 0, \\ \begin{pmatrix} 0 & 0 & \frac{\partial\nu_y}{\partial u_3} & \frac{\partial\nu_y}{\partial u_4} \end{pmatrix} & \text{if } \Delta\nu_y(t) < 0 \end{cases} \quad (52)$$

A similar approach is taken for the roll component $\Delta\nu_x(t)$ and the yaw component $\Delta\nu_z(t)$. The roll component $\mathbf{B}_{\nu_x}(\Delta\nu_x(t))$ is conditionalized in the following way:

$$\mathbf{B}_{\nu_x}(\Delta\nu_x(t)) = \begin{cases} \begin{pmatrix} 0 & \frac{\partial\nu_x}{\partial u_2} & \frac{\partial\nu_x}{\partial u_3} & 0 \end{pmatrix} & \text{if } \Delta\nu_x(t) \geq 0, \\ \begin{pmatrix} \frac{\partial\nu_x}{\partial u_1} & 0 & 0 & \frac{\partial\nu_x}{\partial u_4} \end{pmatrix} & \text{if } \Delta\nu_x(t) < 0 \end{cases} \quad (53)$$

The yaw-related row $\mathbf{B}_{\nu_z}(\Delta\nu_z(t))$ is defined as

$$\mathbf{B}_{\nu_z}(\Delta\nu_z(t)) = \begin{cases} \begin{pmatrix} 0 & \frac{\partial\nu_z}{\partial u_2} & 0 & \frac{\partial\nu_z}{\partial u_4} \end{pmatrix} & \text{if } \Delta\nu_z(t) \geq 0, \\ \begin{pmatrix} \frac{\partial\nu_z}{\partial u_1} & 0 & \frac{\partial\nu_z}{\partial u_3} & 0 \end{pmatrix} & \text{if } \Delta\nu_z(t) < 0 \end{cases} \quad (54)$$

The resulting control input $\mathbf{u}_s(t)$ constitutes a reasonable and operationally meaningful steady-state reference. It is important to emphasize that $\mathbf{u}_s(t)$ is neither enforced as a hard constraint nor required to exactly satisfy $\nu(t) = \mathbf{B}\mathbf{u}(t)$. Instead, it serves as a designer-preferred initial target for the subsequent redistributed dynamic control allocation, as shown by the closed-form solution in Eq. (36). The actual control input $\mathbf{u}(t)$ is obtained through an iterative redistribution process that explicitly accounts for feasibility, actuator limits, and the full control effectiveness matrix. If the reference $\mathbf{u}_s(t)$ is infeasible, the allocation naturally deviates from it and searches for a nearby admissible solution. From this perspective, the conditionalization can be interpreted as a structured initial guess that biases the solution toward desirable actuator usage patterns. Nevertheless, it must be noted that the induced sparsification implicitly introduces an allocation characteristic related to an ℓ_1 -type control allocation pattern. Such behavior is often advantageous in aerospace applications, as it promotes selective usage of the most effective control surfaces. However, as discussed in Sec. II.A, purely ℓ_1 -based control allocation formulations tend to drive individual actuators toward their magnitude and rate limits,

which can result in frequent constraint activation. In the proposed approach, this tendency is mitigated by embedding the designer-preferred steady-state solution as a soft reference within the proposed ℓ_2 -based dynamic control allocation approach. In this way, the sparsity-inducing properties of the steady-state preference are retained without being enforced as hard constraints, while the ℓ_2 -based dynamic control allocation law preserves compatibility with the full, nonsimplified control effectiveness model as well as actuator magnitude and rate limitations.

VI. Simulation Results

A simulation-based analysis was carried out to examine the proposed IDCA algorithm. The flight dynamics model and the applied aerodynamic data set used for the time simulations are provided in a MATLAB/Simulink environment developed for the control design of hypersonic flight vehicles; see [10] for more details.

A. Comparison Analysis on Static Moment Commands

As discussed in Sec. II.B, the presented PICA and RPICA algorithms are not necessarily capable of handling asymmetrical control input limitations. To illustrate this limitation, a simplified example of a constrained control allocation problem is considered and visualized in Fig. 13. The setup involves two control inputs $\mathbf{u} = [u_1, u_2]^T$, where the admissible set is bounded by asymmetrical magnitude constraints $\mathbf{u}_{\min} = [0, 0]^T$ and $\mathbf{u}_{\max} = [1.5, 1.5]^T$. In this stationary example, rate constraints are neglected. The control effectiveness matrix is defined as $\mathbf{B} = [0.5, -0.5]$, and the demanded virtual control input is given by $\nu = 0.5$.

The set of all solutions to $\nu = \mathbf{B}\mathbf{u}$ forms a straight line in the (u_1, u_2) -plane, shown as the dashed black line in Fig. 13. The admissible region for $\mathbf{u}$ is marked by the dashed red box, representing the input bounds. Thus, the set of feasible solutions corresponds to the intersection between this line and the red box. Geometrically, every point on the black line segment that lies inside the red box is a feasible solution. Among these, the optimal solution is defined as the feasible point that minimizes the cost $J = \|\mathbf{u}\|_2^2$, which corresponds to the point closest to the origin. In this case, the unique optimal solution is $\mathbf{u} = [1, 0]^T$.

As shown in the figure, the standard PICA algorithm computes the solution $\mathbf{u} = [0.5, -0.5]^T$ (magenta star), which satisfies the equation $\nu = \mathbf{B}\mathbf{u}$ but lies outside the admissible region and is therefore physically not feasible. After saturation, the corrected control input becomes $\mathbf{u} = [0.5, 0]^T$ (violet star). This input is admissible but no longer reproduces the commanded virtual input, resulting in a large allocation error. By contrast, RPICA, QPCA, and the proposed IDCA algorithm are capable of incorporating the input limits directly and computing the admissible and optimal solution $\mathbf{u} = [1, 0]^T$ (blue star). While RPICA is sufficient in this simple case, its performance deteriorates for higher-dimensional actuator systems with strong asymmetrical limits as present in the GHGV-2.

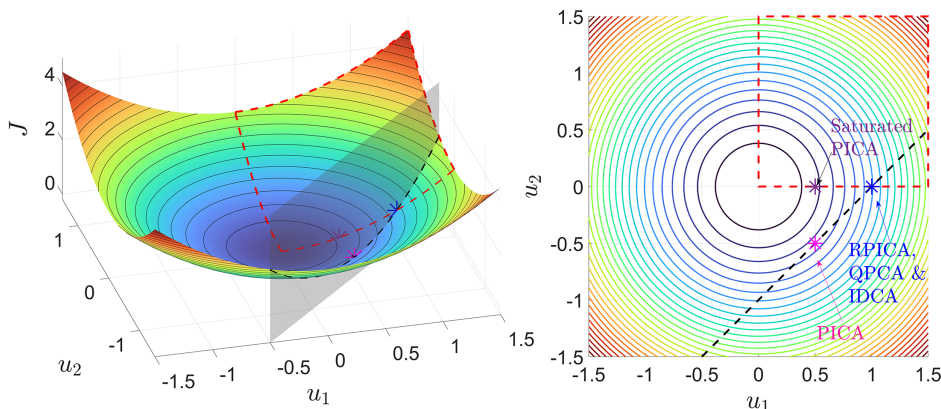


Fig. 13 Results of a simple constrained control allocation problem with two inputs and asymmetrical magnitude limits.

QPCA is generally robust but computationally expensive, motivating the proposed IDCA algorithm that retains real-time feasibility while achieving comparable accuracy.

Based on this simple analysis, the algorithms are next evaluated at a representative operating point of the GHGV-2. The vehicle is trimmed for flight at Mach 8 and an altitude of 30 km, where the four aerodynamic control surfaces of the HGV, u_1 to u_4 , are available. The admissible bounds are defined as

$$\mathbf{u}_{\min} = \begin{bmatrix} 0 \text{ deg} & 0 \text{ deg} & 0 \text{ deg} & 0 \text{ deg} \end{bmatrix}^T, \\ \mathbf{u}_{\max} = \begin{bmatrix} 20 \text{ deg} & 20 \text{ deg} & 20 \text{ deg} & 20 \text{ deg} \end{bmatrix}^T$$

and the control effectiveness matrix at this operating point is

$$\mathbf{B} = \begin{pmatrix} -20.01 & 20.01 & 93.94 & -93.94 \\ 126.7 & 126.7 & -501.4 & -501.4 \\ -127.5 & 127.5 & -45.72 & 46.72 \end{pmatrix} \quad (55)$$

A stationary virtual control command $\nu = [-400, 800, -2000]^T \text{ N} \cdot \text{m}$ is prescribed, which lies within the AMS and is therefore physically realizable without violating the actuator magnitude limits. Since the command is stationary, rate constraints are not considered in this case.

The results are shown in Fig. 14. PICA generates a solution that formally satisfies the allocation equation but violates the lower input bounds for several surfaces, making it physically infeasible. Saturated PICA respects the limits but concentrates almost the entire effort on u_1 , failing to reproduce the commanded moment. RPICA

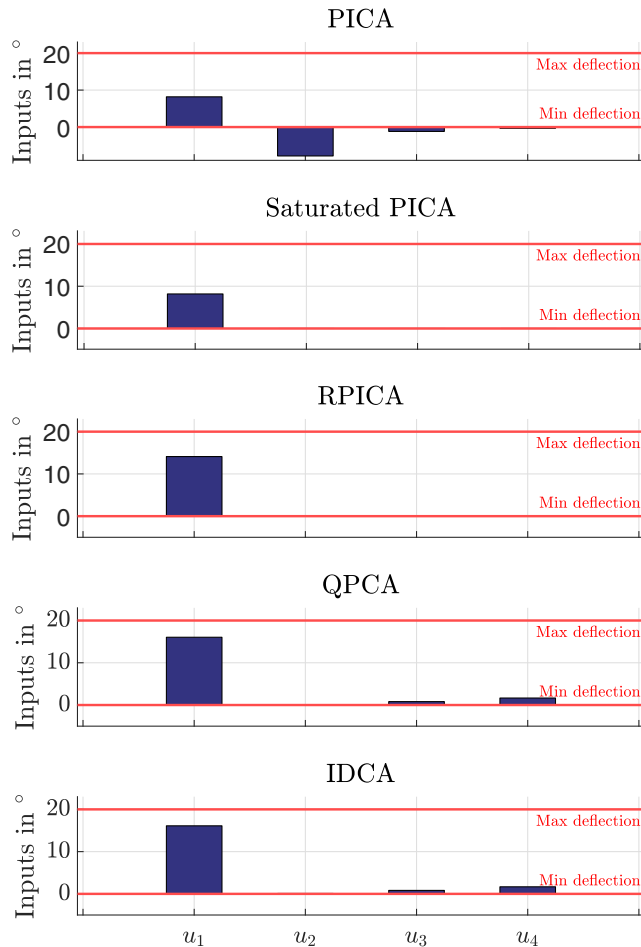


Fig. 14 Obtained control deflections for different allocation algorithms under a moment command within the AMS of the GHGV-2.

Table 1 Results for the GHGV-2 stationary allocation case

Method	Cost $J = \ \mathbf{u}\ _2, \text{ N} \cdot \text{m}$	Error norm $\ \nu - \mathbf{B}\mathbf{u}\ _2, \text{ N} \cdot \text{m}$	Computation time, s
PICA	11.3749	1.3270×10^{-12}	1.5760×10^{-4}
Saturated PICA	8.1773	1.0140×10^3	1.5760×10^{-4}
RPICA	14.1015	1.0140×10^3	3.9210×10^{-4}
QPCA	16.1473	2.2737×10^{-13}	0.3911
IDCA	16.2146	3.2155×10^{-13}	5.5550×10^{-4}

also respects the bounds but suffers from premature saturation in three surfaces during the first iteration, leaving only u_1 to carry the residual, which again results in a large allocation error. In contrast, QPCA and IDCA both provide admissible deflections that satisfy the command. Their allocations are nearly identical, with QPCA solving a quadratic optimization problem and IDCA iterating dynamically over weighted residuals.

Table 1 provides the allocation cost $J = \|\mathbf{u}\|_2$, the allocation error $\|\nu - \mathbf{B}\mathbf{u}\|_2$, and the computation time. PICA exhibits the lowest cost but yields an invalid solution, while saturated PICA and RPICA produce high allocation errors despite respecting the bounds. QPCA and IDCA achieve nearly identical, minimal error norms, but QPCA requires several orders of magnitude longer computation time. IDCA achieves the same level of accuracy at real-time-capable runtimes, only marginally slower than PICA and RPICA.

To further assess robustness, a Monte Carlo analysis with $N = 1000$ random virtual control commands is conducted. The mean command vector is chosen as $\nu_m = [-100, 300, -500]^T$, with each component modeled as a Gaussian random variable $\nu_i \sim \mathcal{N}(\nu_{m,i}, \sigma_i^2)$. The standard deviations are set to $\sigma_i = (2/3)|\nu_{m,i}|$ for $i = 1, 2, 3$, such that the $\pm 3\sigma$ range spans approximately twice the mean value of each component. Random commands are then generated according to $\nu = \nu_m + \sigma \odot \mathbf{z}$, where $\mathbf{z}$ is a vector of independent standard normal variables and $\odot$ denotes elementwise multiplication. A total of $N = 1000$ such samples is generated, ensuring a broad coverage of possible commands both near and within the AMS. The actuator limits and the control effectiveness matrix remain unchanged from the stationary case.

Figure 15 shows the cost distributions. IDCA produces slightly higher costs than QPCA and RPICA on average, but all results remain within physically reasonable ranges. PICA frequently exhibits lower costs, although many of its solutions are inadmissible.

The corresponding allocation error distributions in Fig. 16 reveal that PICA, QPCA, and IDCA achieve consistently low errors, whereas saturated PICA and RPICA produce significantly larger

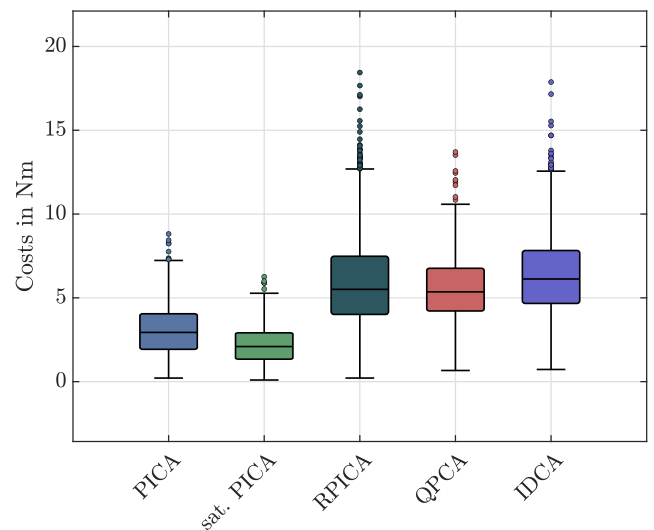


Fig. 15 Distribution of allocation costs $J = \|\mathbf{u}\|_2$ over $N = 1000$ Monte Carlo samples.

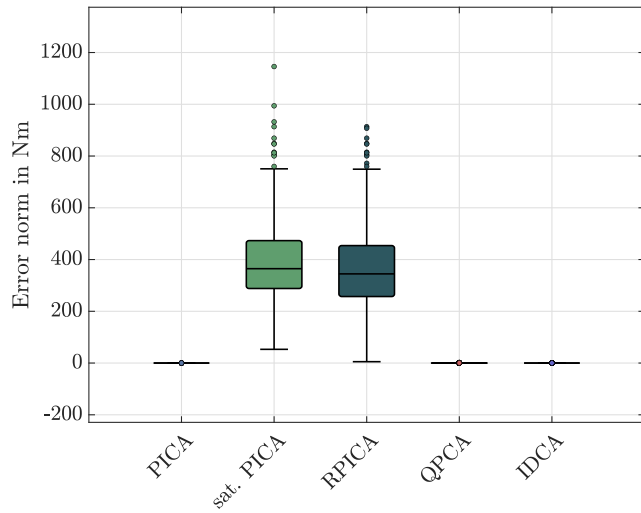


Fig. 16 Distribution of allocation error $\|\nu - Bu\|_2$ over $N = 1000$ Monte Carlo samples.

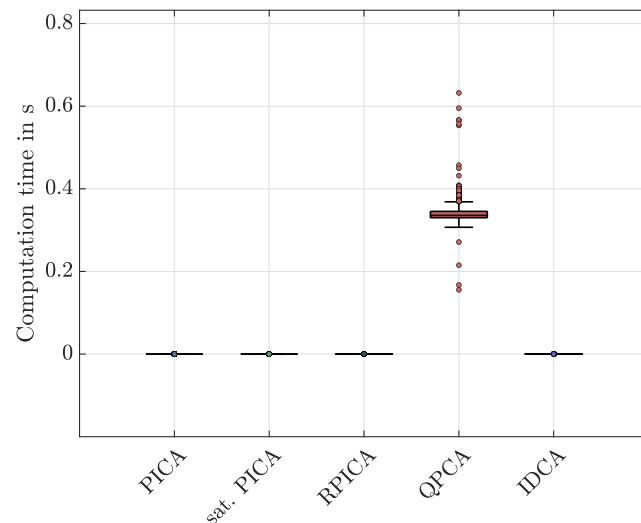


Fig. 17 Computation times across $N = 1000$ Monte Carlo samples.

errors, often exceeding several hundred N·m. This confirms that PICA neglects input limits, while QPCA and IDCA generate realizable allocations.

Finally, Fig. 17 compares computation times. QPCA requires orders of magnitude longer runtimes due to solving a quadratic program at every iteration, while PICA, saturated PICA, and RPICA remain computationally fast but unreliable. IDCA achieves runtimes in the millisecond range, slightly above RPICA, but still compatible with real-time requirements.

Overall, the stationary and Monte Carlo analyses confirm that while PICA produces formally correct but inadmissible solutions, and RPICA respects constraints but distributes residuals poorly, QPCA and IDCA achieve accurate, realizable allocations. The decisive advantage of IDCA is that it matches the accuracy of QPCA at real-time-capable runtimes, demonstrating its suitability for onboard implementation in the GHGV-2.

B. Time-Varying Moment Commands on Proposed Algorithm

For this assessment, the same control effectiveness matrix B as in Eq. (55) is considered. The GHGV-2 is again trimmed at Mach 8 and an altitude of 30 km, representing the same operating condition as before. In this analysis, however, the magnitude limits of the control surfaces are modeled as time-varying functions in order to capture the dynamic effects of varying aerodynamic loads at high

Mach numbers. It is assumed that the influence of dynamic pressure leads to a reduction in the achievable upper magnitude bound of the actuators, while the lower magnitude bound remains at zero for all control surfaces, as it is not significantly affected by external influences due to the direction of the resulting aerodynamic forces. To model this effect, the upper bounds are expressed as a sinusoidal modulation:

$$u_{\max}(t) = u_{\max, \text{full}} \cos(\Lambda(t))$$

where $u_{\max, \text{full}} = 20$ deg denotes the technically achievable maximum flap deflection and $\Lambda(t)$ is a freely chosen modulation function.

In addition to time-varying magnitude bounds, rate constraints are considered as well. The upper and lower bounds of the rates, $|\dot{u}_{\max}|$ and $|\dot{u}_{\min}|$, are modeled as time-dependent functions. The upper rate limit decreases from 20 deg/s to 10 deg/s over the simulation, reflecting the reduced ability of actuators to achieve high positive rates under increasing aerodynamic loads. Conversely, the lower rate limit is assumed to increase from -20 deg/s to -30 deg/s, because the aerodynamic forces facilitate the retraction of the control surfaces. This asymmetry captures the physical reality that positive rate changes become more restricted, while negative rate changes become easier to achieve as the flight condition evolves.

The allocation module is tested with a sinusoidal virtual control input command $\nu(t) = [\nu_x(t), \nu_y(t), \nu_z(t)]^T$. The chosen command alternates between feasible and infeasible values with respect to the AMS, enabling an evaluation of the algorithm's performance under both realizable and saturated conditions. Figure 18 compares the commanded virtual control inputs (green) with the moments generated by the IDCA algorithm (blue). A generally close tracking is observed, particularly for ν_z , while noticeable deviations appear in ν_x and ν_y whenever actuator magnitude or rate saturation is encountered. During unsaturated phases the algorithm achieves high accuracy, with nearly perfect matching of the requested commands.

Figure 19 shows the error between the requested and generated commands, defined as

$$\Delta\nu(t) = \nu(t) - Bu(t)$$

The results confirm that the error converges toward zero in non-saturated phases, while in saturated regimes distinct deviations occur. The error magnitudes align with the observations from Fig. 18, where ν_y exhibits the largest deviations during phases of strong demand. This correlation indicates that actuator constraints directly determine the achievable accuracy of the allocation.

The actuator magnitudes $u_1, \dots, u_4$ and their time-varying bounds are shown in Fig. 20. The trajectories clearly demonstrate that all commands remain within the admissible set. In the initial phase, a nearly linear increase is visible for u_1 and u_2 , which indicates active rate saturation: since the maximum rate is reached, the magnitude grows linearly until the limit changes. Comparing with Fig. 21 confirms this behavior, as $\dot{u}_1$ and $\dot{u}_2$ remain pinned at their respective limits during these phases. The corresponding deviations in $\Delta\nu_x$ and $\Delta\nu_y$ highlight the direct causal link between actuator rate constraints and tracking error.

The role of the control surfaces in producing the commanded virtual inputs is also evident. For example, to generate positive ν_y , the algorithm retracts the surfaces that would produce negative contributions (u_3, u_4) while driving u_1, u_2 upward. Conversely, for negative ν_y , the roles switch. A similar logic applies for roll moments ν_x , where surfaces u_2, u_3 dominate the positive direction, while u_1, u_4 dominate the negative. This role distribution is consistent with the control effectiveness structure in Eq. (55) and explains why certain surfaces repeatedly saturate depending on the sign of the demanded moment. The algorithm implicitly prioritizes the dominant axis when strong demands occur, reducing error in that channel even if moderate errors appear in the others. For instance, in phases of high-pitch demand, the allocation accepts deviations in ν_x to minimize the error in ν_y . Such behavior could be

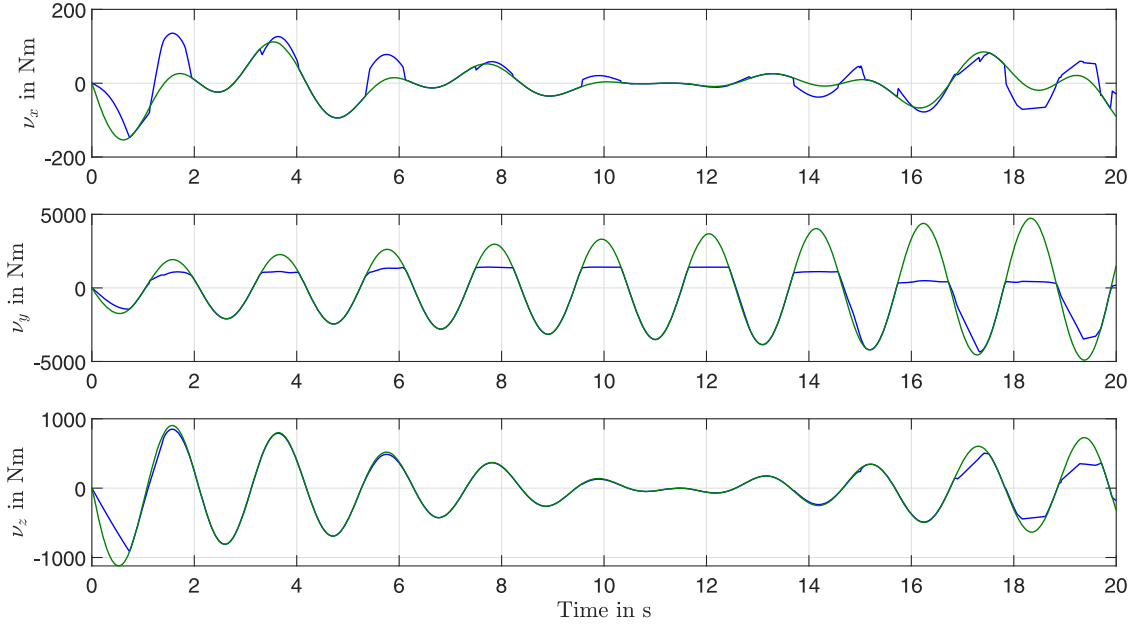


Fig. 18 Results for the virtual control commands ν_x , ν_y , and ν_z . Green: commanded input. Blue: generated input after allocation.

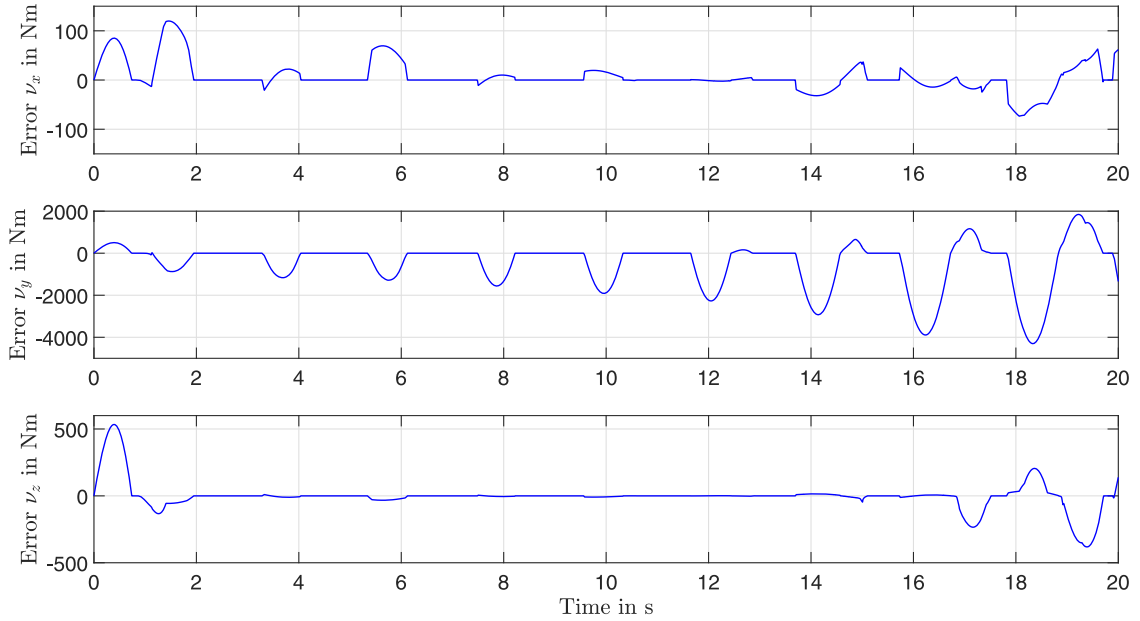


Fig. 19 Error between commanded and generated virtual control inputs ν_x , ν_y , and ν_z . Blue lines: allocation error time series.

adapted with weighting factors to impose mission-specific priorities, though in the present setup the distribution already provides a reasonable compromise.

The time histories of the rates in Fig. 21 further illustrate this behavior. Whenever a magnitude limit is reached, the corresponding rate collapses to zero, consistent with the physical limit of the actuator. This produces linear evolution in the magnitude traces and explains the sharp transitions observed in the error plots. Spikes in the rate signals occur when one surface saturates and the remaining effort is redistributed across the other actuators, causing rapid but coordinated adjustments. These interactions confirm the coupled nature of the allocation: constraints in one actuator channel propagate into others as the residual command is redistributed.

Overall, the simulation results present a consistent and physically plausible picture. The time-varying magnitude and rate bounds are respected at all times, the algorithm provides accurate tracking in feasible regions, and deviations are directly attributable to actuator limits. The prioritization behavior ensures that dominant demands

are satisfied first, yielding a robust tradeoff across channels. Importantly, even under dynamic and asymmetric constraints, the IDCA algorithm demonstrates reliable real-time operation and robust allocation, confirming its applicability for hypersonic flight scenarios where actuator characteristics vary significantly over time.

VII. Conclusions

This paper presents an iterative dynamic control allocation approach tailored to hypersonic glide vehicles and evaluates it in the context of asymmetric input limitations, actuator coupling, and state-dependent operating conditions. In contrast to pseudo-inverse-based and redistributed pseudo-inverse methods, the proposed formulation explicitly accounts for asymmetric magnitude and rate constraints and avoids degenerate or infeasible solutions under strong actuator coupling. Compared to quadratic-programming-based control allocation approaches, the method significantly reduces computational complexity while retaining the ability to

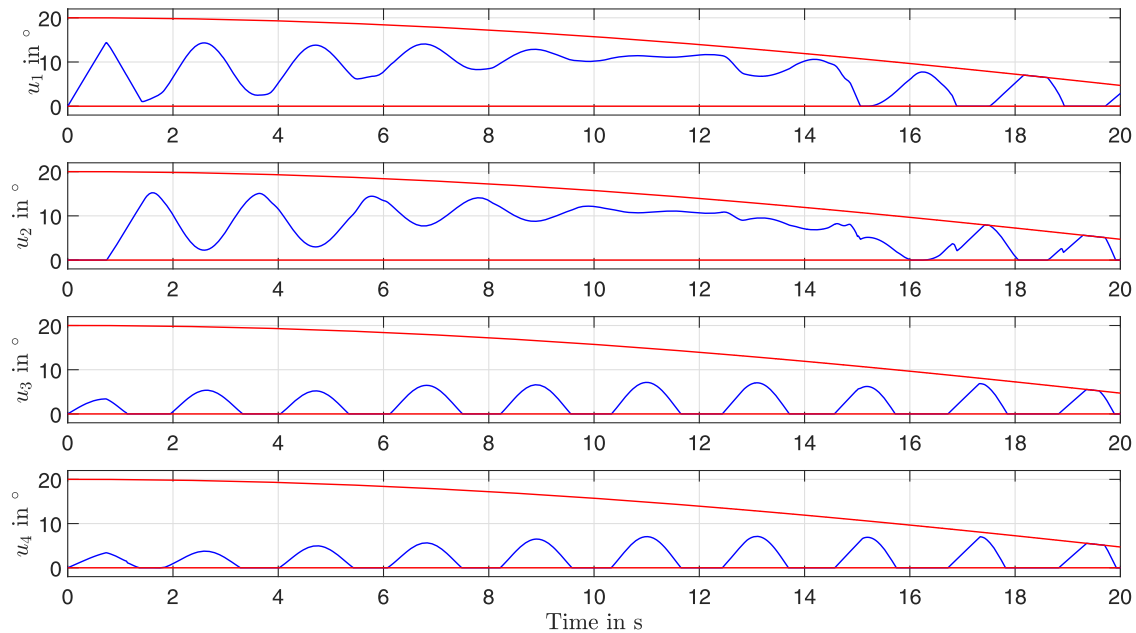


Fig. 20 Computed control input magnitudes with time-varying magnitude constraints. Blue: generated deflections. Red: admissible bounds.

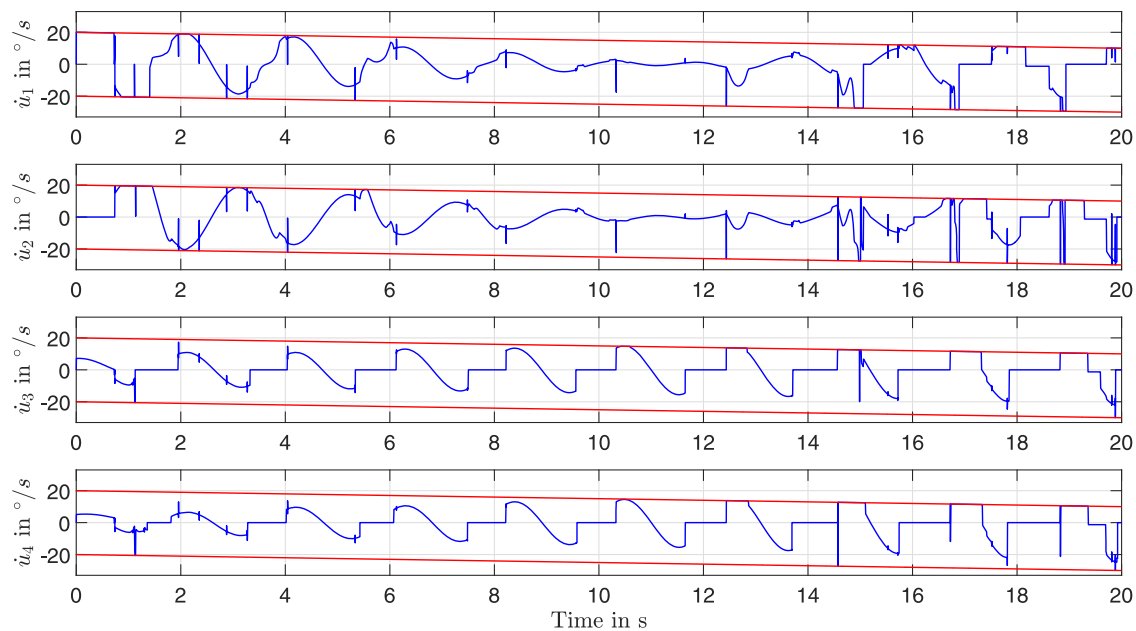


Fig. 21 Computed control input rates with time-varying rate constraints. Blue: generated rates. Red: admissible bounds.

shape control distribution through soft constraints, making it suitable for real-time implementation in inner-loop flight control systems.

A key aspect of the proposed approach lies in the formulation of the control allocation problem itself. By appropriately designing state- and actuator-dependent weighting functions, the allocation process accounts for the instantaneous magnitude and rate states of the actuators as well as their directional control effectiveness. This enables a more balanced distribution of control effort and reduces the tendency of individual actuators to saturate, thereby preserving control authority under constrained conditions. In addition, the weighting function design provides a systematic mechanism to address thermal load effects. The use of drag-sensitive weighting functions allows thermal considerations to be incorporated implicitly into the allocation process without introducing additional hard constraints. By influencing control distribution based on drag generation characteristics, thermal loads and associated energy losses

can be reduced while maintaining feasibility and compatibility with an iterative, approximate solution strategy.

Simulation results using the GHGV-2 model demonstrate that the proposed approach reliably respects actuator magnitude and rate limits, maintains control authority under realistic flight conditions, and offers a practical and computationally efficient alternative to existing state-of-the-art control allocation techniques for hypersonic applications.

References

- [1] Slayer, K. M., "Hypersonic Weapons: Background and Issues for Congress," Congressional Research Service TR R45811, Washington D.C., 2023, <https://crsreports.congress.gov/product/pdf/R/R45811/35>.
- [2] Autenrieb, J., "Data Fusion-Based Incremental Nonlinear Model Following Control Design for a Hypersonic Waverider Configuration," *AIAA SciTech 2023 Forum*, AIAA Paper 2023-1997, 2023, <https://doi.org/10.2514/6.2023-1997>

- [3] Oppenheimer, M. W., Doman, D. B., and Bolender, M. A., "Control Allocation for Over-Actuated Systems," *2006 14th Mediterranean Conference on Control and Automation*, Mediterranean Control Assoc. (MCA), 2006, pp. 1–6.
<https://doi.org/10.1109/MED.2006.328750>
- [4] Bordignon, K., and Bessolo, J., "Control Allocation for the X-35B," *Biennial International Powered Lift Conference and Exhibit*, AIAA Paper 2002-6020, 2022.
<https://doi.org/10.2514/6.2002-6020>
- [5] Johansen, T., Fossen, T., and Berge, S., "Constrained Nonlinear Control Allocation with Singularity Avoidance Using Sequential Quadratic Programming," *IEEE Transactions on Control Systems Technology*, Vol. 12, No. 1, 2004, pp. 211–216.
<https://doi.org/10.1109/TCST.2003.821952>
- [6] Härkegård, O., "Dynamic Control Allocation Using Constrained Quadratic Programming," *Journal of Guidance Control and Dynamics*, Vol. 27, No. 6, 2004, pp. 1028–1034.
<https://doi.org/10.2514/1.11607>
- [7] Burken, J. J., Lu, P., Wu, Z., and Bahm, C., "Two Reconfigurable Flight-Control Design Methods: Robust Servomechanism and Control Allocation," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 3, 2001, pp. 482–493.
<https://doi.org/10.2514/2.4769>
- [8] Bordignon, K. A., "Constrained Control Allocation for Systems with Redundant Control Effectors," Ph.D. Thesis, Virginia Polytechnic Inst. and State Univ., Blacksburg, VA, 1996.
- [9] Hafner, S., Steinert, A., and Holzapfel, F., "Connecting Sequential Least Squares Active Set, Exact Redistributed, and Redistributed Scaled Pseudoinverse Control Allocation," *2025 European Control Conference (ECC)*, European Control Assoc. (EUCA), 2025, pp. 1118–1124.
<https://doi.org/10.23919/ECC65951.2025.11187134>
- [10] Autenrieb, J., Fezans, N., Gruhn, P., and Klevanski, J., *Towards a Control-Centric Modelling and Simulation-Framework for Hypersonic Glide Vehicles*, German Aeronautics and Space Congress (DLRK), Bremen, Germany, 2021.
- [11] Autenrieb, J., and Fezans, N., "Nonlinear Model Following Control Design for a Hypersonic Waverider Configuration," *CEAS EuroGNC "Conference on Guidance, Navigation and Control"*, Council of European Aerospace Societies (CEAS), May 2022.
- [12] Gruhn, P., "Design and Analysis of a Hypersonic Glide Vehicle (Original German Title: Auslegung Und Analyse Eines Hypersonischen Gleitflugkörpers)," *Conference on Applied Research for Defense and Security in Germany*, German Assoc. for Defence Technology (DWT), 2020.
- [13] Holzapfel, F., "Nonlinear Adaptive Control of an Unmanned Aerial Vehicle (Original German Title: Nichtlineare Adaptive Regelung Eines Unbemannten Fluggerätes)," Ph.D. Thesis, Technische Universität München, München, Germany, 2004, <https://mediatum.ub.tum.de/?id=601905>.
- [14] Lombaerts, T., and Looye, G., "Design and Flight Testing of Nonlinear Autoflight Control Laws," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2012-4982, 2012.
<https://doi.org/10.2514/6.2012-4982>
- [15] Bhardwaj, P., Raab, S. A., and Holzapfel, F., "Higher Order Reference Model for Continuous Dynamic Inversion Control," *AIAA Scitech 2021 Forum*, AIAA Paper 2021-1130, 2021.
<https://doi.org/10.2514/6.2021-1130>
- [16] Petersen, J. A. M., and Bodson, M., "Interior-Point Algorithms for Control Allocation," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 3, 2005, pp. 471–480.
<https://doi.org/10.2514/1.5937>
- [17] Pfeifle, O., and Fichter, W., "Cascaded Incremental Nonlinear Dynamic Inversion for Three-Dimensional Spline-Tracking with Wind Compensation," *Journal of Guidance, Control, and Dynamics*, Vol. 44, No. 8, 2021, pp. 1559–1571.
<https://doi.org/10.2514/1.G005785>
- [18] Schierman, J. D., Ward, D. G., Hull, J. R., Gandhi, N., Oppenheimer, M., and Doman, D. B., "Integrated Adaptive Guidance and Control for Re-Entry Vehicles with Flight Test Results," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 6, 2004, pp. 975–988.
<https://doi.org/10.2514/1.10344>
- [19] Härkegård, O., "Backstepping and Control Allocation with Applications to Flight Control," Ph.D. Thesis, Linköping Universitet, Linköping, Sweden, 2003, <http://www.diva-portal.org/smash/get/diva2:243491/FULLTEXT01.pdf>.
- [20] Ikeda, Y., and Hood, M., "An Application of L1 Optimization to Control Allocation," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, AIAA Paper 2000-4566, 2000.
<https://doi.org/10.2514/6.2000-4566>
- [21] Bolender, M. A., and Doman, D. B., "Nonlinear Control Allocation Using Piecewise Linear Functions," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 6, 2004, pp. 1017–1027.
<https://doi.org/10.2514/1.9546>
- [22] Bodson, M., and Frost, S. A., "Load Balancing in Control Allocation," *Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 2, 2011, pp. 380–387.
<https://doi.org/10.2514/1.51952>
- [23] Huang, X., and Duan, G., "Dynamic Infinity-Norm Constrained Control Allocation for Attitude Tracking Control of Overactuated Combined Spacecraft," *IET Control Theory & Applications*, Vol. 13, No. 11, 2019, pp. 1692–1703.
<https://doi.org/10.1049/iet-cta.2018.5707>
- [24] Frost, S., Bodson, M., and Acosta, D., "Sensitivity Analysis of Linear Programming and Quadratic Programming Algorithms for Control Allocation," *AIAA Infotech@Aerospace Conference*, AIAA Paper 2009-1936, 2009.
<https://doi.org/10.2514/6.2009-1936>
- [25] Frost, S. A., and Bodson, M., "Resource Balancing Control Allocation," *Proceedings of the 2010 American Control Conference*, Inst. of Electrical and Electronics Engineers, New York, 2010, pp. 1326–1331.
<https://doi.org/10.1109/ACC.2010.5530448>
- [26] Virnig, J., and Bodden, D., "Multivariable Control Allocation and Control Law Conditioning When Control Effectors Limit," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 1994-3609, 1994.
<https://doi.org/10.2514/6.1994-3609>
- [27] Zhang, J., Bhardwaj, P., Raab, S. A., Saboo, S., and Holzapfel, F., "Control Allocation Framework for a Tilt-Rotor Vertical Take-Off and Landing Transition Aircraft Configuration," *AIAA Aviation Forum*, AIAA Paper 2018-3480, 2018.
<https://doi.org/10.2514/6.2018-3480>
- [28] Härkegård, O., "Dynamic Control Allocation Using Constrained Quadratic Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 6, 2004, pp. 1028–1034.
<https://doi.org/10.2514/1.11607>
- [29] Autenrieb, J., and Fezans, N., "Flight Control Design for a Hypersonic Waverider Configuration: A Non-Linear Model Following Control Approach," *CEAS Space Journal*, Vol. 17, No. 1, 2024, pp. 1–24.
<https://doi.org/10.1007/s12567-024-00544-0>
- [30] Li, G., Zhang, H., and Tang, G., "Flight-Corridor Analysis for Hypersonic Glide Vehicles," *Journal of Aerospace Engineering*, Vol. 30, No. 1, 2017, Paper 06016005.
[https://doi.org/10.1061/\(ASCE\)AS.1943-5525.0000667](https://doi.org/10.1061/(ASCE)AS.1943-5525.0000667)
- [31] An, K., Guo, Z.-Y., Xu, X.-P., and Huang, W., "A Framework of Trajectory Design and Optimization for the Hypersonic Gliding Vehicle," *Aerospace Science and Technology*, Vol. 106, Nov. 2020, Paper 106110.
<https://doi.org/10.1016/j.ast.2020.106110>
- [32] Sadien, E., Roos, C., Birouche, A., Carton, M., Grimault, C., Romana, L. E., and Basset, M., "A Simple and Efficient Control Allocation Scheme for On-Ground Aircraft Runway Centerline Tracking," *Control Engineering Practice*, Vol. 95, Feb. 2020, Paper 104228.
<https://doi.org/10.1016/j.conengprac.2019.104228>
- [33] Hohn, O., and Gruhn, P., "Thermal Investigations on a Wind Tunnel Model of a Hypersonic Missile (Original German Title: Thermaluntersuchungen an Einem Windkanalmodell Eines Hypersonischen Flugkörpers)," *Conference on Applied Research for Defense and Security in Germany*, German Assoc. for Defence Technology (DWT), 2018.

M. W. Oppenheimer
Associate Editor