A Survey of Optimal Control Allocation for Aerial Vehicle Control

Blaha, T.M.; Smeur, E.J.J.; Remes, B.D.W.

DOI
10.3390/act12070282

Publication date
2023

Document Version
Final published version

Published in
Actuators

Citation (APA)
Blaha, T. M., Smeur, E. J. J., & Remes, B. D. W. (2023). A Survey of Optimal Control Allocation for Aerial Vehicle Control. *Actuators*, *12*(7), Article 282. https://doi.org/10.3390/act12070282

actuators

Review

A Survey of Optimal Control Allocation for Aerial Vehicle Control

Till Martin Blaha *, **Ewoud Jan Jacob Smeur** and **Bart Diane Walter Remes**

Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1, 2629 HS Delft, The Netherlands
* Correspondence: t.m.blaha@tudelft.nl

Abstract: In vehicle control, control allocation is often used to abstract control variables from actuators, simplifying controller design and enhancing performance. Surveying available literature reveals that explicit solutions are restricted to strong assumptions on the actuators, or otherwise fail to exploit the capabilities of the actuator constellation. A remedy is to formulate hierarchical minimization problems that take into account the limits of the actuators at the expense of a longer computing time. In this paper, we compared the most common norms of the objective functions for linear or linearized plants, and show available numeric solver types. Such a comparison has not been found in the literature before and indicates that some combinations of linear and quadratic norms are not sufficiently researched. While the bulk of the review is restricted to control-affine plant models, some extensions to dynamic and nonlinear allocation problems are shown. For aerial vehicles, a trend toward linearized incremental control schemes is visible, which forms a compromise between real-time capabilities and the ability to resolve some nonlinearities common in these vehicles.

Keywords: control allocation; optimal control; hierarchical optimization; allocation schemes; flight control systems

check for updates

Citation: Blaha, T.M.; Smeur, E.J.J.; Remes, B.D.W. A Survey of Optimal Control Allocation for Aerial Vehicle Control. *Actuators* **2023**, *12*, 282. https://doi.org/10.3390/act12070282

Academic Editor: Keigo Watanabe

Received: 31 May 2023
Revised: 4 July 2023
Accepted: 6 July 2023
Published: 11 July 2023

1. Introduction

In many control applications, each of the plant's actuators may not directly affect a single state variable or derivative. In vehicle motion control, adding actuators can increase reliability through redundancy, enhance performance, or enable control over additional degrees of freedom. In novel high-performance aircraft configurations such as the Innovative Control Effectors demonstrator [1], but also in multirotor eVTOL aircraft [2], there are typically far more actuators than controlled degrees of freedom. Many of these actuators are axis-coupled, meaning that they affect the motion of more than one degree of freedom.

The design of feedback controllers for such plants can be simplified by choosing suitable controller outputs called 'pseudo-controls' $\boldsymbol{\nu}$, which are more closely related to the plant's state. This abstracted required control effort $\boldsymbol{\nu}_{ref}$ is then *allocated* over the available actuators in a separate step, resulting in the 'actuator commands' $\boldsymbol{u}$. Therefore, the controller itself does not need detailed information on the platform actuators. Depending on assumptions about the actuator suite and requirements on the solution, a number of control allocation techniques and applications are described in the literature.

Control allocation is relevant to many vehicles and has been applied to naval vessels for station keeping [3] and maneuvering [4], underwater vehicle control [5], the electronic stability control of road vehicles [6,7], high-performance aircraft [8,9], and unmanned air vehicles (UAVs) [10,11]. This work aims to provide motivation, derivation, capabilities, and limitations of several commonly used methods, focusing on aircraft. In particular, methods based on linear optimization programs are described with an extensive comparison of the possible distance norms in the objective function.

In Section 2 of this paper, the general plant models, some simplifying assumptions, and quality measures of control allocation methods are introduced. The body of the paper, Section 3, gives the results of the survey with a description and analysis of available methods, while Section 4 concludes the findings from these discussions.

2. Formulation, Assumptions, and Requirements

Pseudo-controls are often chosen as the lowest derivative of the state or output of a system, which contains the plant's input directly. Due to the algebraic relation between actuator states and pseudo-controls, this makes nonlinear inversion control laws (also known as feedback linearization) possible [12,13]. As an example of this, actuators in vehicle control commonly give rise to forces and moments on the vehicle, which makes for a natural choice of $\boldsymbol{\nu} = \left(\boldsymbol{a}^T,\ \dot{\boldsymbol{\omega}}^T\right)^T$, where $\boldsymbol{a}$ is the linear acceleration and $\dot{\boldsymbol{\omega}}$ is the angular acceleration required for the desired vehicle trajectory.

A model of the pseudo-controls $\boldsymbol{\nu} \in \mathbb{R}^d$ for d degrees of freedom is given below [8], which depends on the plant's state $\boldsymbol{x}$ and the state of the actuators $\boldsymbol{\delta}$ only (and not, for instance, their rates):

$$\begin{aligned} \boldsymbol{\nu} &= g(\boldsymbol{x}, \boldsymbol{\delta}) \\ \frac{\mathrm{d}^k\boldsymbol{\delta}}{\mathrm{d}t^k} &= f\left(\boldsymbol{\delta}, \dot{\boldsymbol{\delta}}, \ldots, \frac{\mathrm{d}^{k-1}\boldsymbol{\delta}}{\mathrm{d}t^{k-1}}, \boldsymbol{u}\right) \quad \boldsymbol{u} \in \mathcal{U}. \end{aligned} \tag{1}$$

The actuator state $\boldsymbol{\delta}$ is described as a dynamical system of order k, with the target actuator state $\boldsymbol{u} \in \mathcal{U} \subset \mathbb{R}^n$. The control allocation problem consists of finding an admissible $\boldsymbol{u}^* \in \mathcal{U}$ such that $\boldsymbol{\nu} = \boldsymbol{\nu}_{ref}$ for every pseudo-control reference input $\boldsymbol{\nu}_{ref} \in \Omega \subset R^d$. Here, Ω is known as the attainable moment set ('AMS') [14], which is a direct consequence of the restriction of $\boldsymbol{u}$. In case the desired $\boldsymbol{\nu}_{ref}$ is not in Ω ('infeasible'), then a command $\boldsymbol{u}^*$ should be returned that results in a $\boldsymbol{\nu}$ that is close to the reference in some sense. In these cases, it is desirable to be able to prioritize some pseudo-controls in case they are more relevant to vehicle stability.

Appropriate balancing between the individual actuators is another important quality of a solution $\boldsymbol{u}^*$ [15], although sometimes the opposite may be preferred (usage of only the most efficient actuators [16]). In any case, small changes in $\boldsymbol{\nu}_{ref}$ should not result in large changes in $\boldsymbol{\nu}$ or $\boldsymbol{u}^*$. Lastly, a suitable allocation scheme should return $\boldsymbol{u}^*$ within a reasonable amount of computational time in order to be real-time capable.

2.1. Simplifying Assumptions on the Plant

Allocation can be significantly simplified by restricting the plant to control-affine models of the form

$$\boldsymbol{\nu} = g_x(\boldsymbol{x}) + g_\delta(\boldsymbol{x})\boldsymbol{\delta}\,. \tag{2}$$

If the plant dynamics $g_x(\boldsymbol{x})$ are known, then this can be rewritten with $\widetilde{\boldsymbol{\nu}} = \boldsymbol{\nu} - g_x(\boldsymbol{x})$, as common in nonlinear dynamic inversion control laws [12]. Similarly, $g_x(\boldsymbol{x})$ can be seen as a disturbance signal to be taken up by integral action in a PID controller for $\boldsymbol{\nu}_{ref}$, or eliminated using incremental control schemes [9,13]. Both terms in Equation (2) are potentially subject to knowledge of the state $\boldsymbol{x}$, which may require the use of estimators. This introduces additional sources of inaccuracies, next to modeling errors in the model(s) $g_x(\boldsymbol{x})$ and $g_\delta(\boldsymbol{x})$. For brevity, we will write $B \triangleq g_\delta(\boldsymbol{x})$, remembering that this 'effectiveness matrix' may depend on the plant state $\boldsymbol{x}$.

This generally excludes a large group of actuator constellations with nonlinearities in their effect on $\boldsymbol{\nu}$, or couplings between the actuators for which the second term has to be written $g_\delta(\boldsymbol{x},\ \boldsymbol{\delta})$. An exception is nonlinear actuators of the form $\boldsymbol{\nu} = g_x(\boldsymbol{x}) + g_\delta(\boldsymbol{x})\boldsymbol{\xi}(\boldsymbol{\delta})$, where ξ_i are invertible mappings $\mathbb{R}^1 \mapsto \mathbb{R}^1$ that are applied to the components of $\boldsymbol{\delta}$ separately. In these cases, linearity can be recovered by solving in terms of $\xi_i(\delta_i)$ and then applying ξ_i^{-1}. The bulk of the methods described in this work are restricted to these systems.

The thrust from rotors is often modeled as second-order polynomials that are strictly increasing between the actuator limits, and thus invertible. However, the effect of aerodynamic control surfaces is not strictly increasing when flow separates. Similarly, the friction generated at the tire–road interface will decrease when the slip angles or longitudinal slip ratio states are large. In these cases, no inverse mapping exists.

While most of the schemes presented in the following depend on this assumption, nonlinear control allocation is discussed in Section 3.5 along with the implications of linearization.

2.2. Assumptions on the Actuator Dynamical System

For the actuators themselves, one may assume that they can be modeled as a first-order lag for all actuators. This enables frequency-divided methods that prioritize actuators by their bandwidths. If these bandwidths are all negligibly fast compared to the plant dynamics such that $\boldsymbol{\delta} \approx \boldsymbol{u}$, then (together with a control-affine plant model) this results in a special case of Equation (1). Inverting this equation can be seen as the essence of static linear control allocation:

$$\boldsymbol{\nu} = B\boldsymbol{u} \quad \boldsymbol{u} \in \mathcal{U} \,. \tag{3}$$

Finally, actuators can often be seen as uncoupled from each other, and so it is common to define $\mathcal{U}$ as a box in R^n with lower and upper bounds $\underline{\mathrm{u}}$ and $\overline{\boldsymbol{u}}$:

$$\mathcal{U} = \{u_i \mid \underline{\mathrm{u}}_i \leq u_i \leq \overline{u}_i\} \quad \text{for all components} \quad i = \{1,\, 2,\, \ldots,\, n\} \,. \tag{4}$$

3. Evaluation of Methods

3.1. Ganging and Daisy Chaining

In the earliest flight control applications, the plant consisted mainly of linear axis-decoupled actuators (each $\boldsymbol{\delta}_i$ only influences one $\boldsymbol{\nu}_i$). This allows for the construction of a simple ganging matrix G with one non-zero entry in each row, such that the unconstrained solution would be $\boldsymbol{u}^*_{unc} = G\boldsymbol{\nu}$ [15]. This can exploit the entire (box-shaped) AMS and at least provides the optimal choice $\boldsymbol{u}^*_i = \min(\max(\boldsymbol{u}^*_{unc,i},\, \underline{\mathrm{u}}_i),\, \overline{\boldsymbol{u}}_i)\ \forall i$ whenever $\boldsymbol{\nu}_{ref}$ is infeasible, since each direction can be 'optimized' separately. Of course, the very restrictive assumptions on the actuators do not make this approach practical.

An expansion on ganging is daisy chaining, in which the columns of B are separated into different matrices of descending actuator priorities $B_1, B_2, \ldots, B_k$, with their ganging matrices G_k. If the solution to the most preferred actuators does not reach the desired pseudo-control ($\boldsymbol{\nu}_{ref} \neq B_1\boldsymbol{u}^*_1$), then the next actuator set is used for the remainder of $\boldsymbol{\nu}_{ref}$ as $\boldsymbol{u}^*_{2,unc} = G_2(\boldsymbol{\nu}_{ref} - B_1\boldsymbol{u}^*_1) \neq 0$. An example of this is the use of spoilers for roll control on traditional transport aircraft, which are only actuated for high acceleration demands, since their usage implies high lift and drag penalties. The simplicity of this scheme allows for stability analysis even in the case of actuator saturation [17].

3.2. Weighted Generalized Inverse

If we relax the axis-decoupling assumption, then G will not be sparse and cannot be derived by inspection. In addition, in most cases, there are more actuators n than controlled degrees of freedom d, such that there may be multiple solutions $\boldsymbol{u}$ that result in $\boldsymbol{\nu}_{ref}$. In those cases, a constrained minimization problem can be derived using a weighing matrix W_u and the p-norm, which minimizes a norm of the control effort while still maintaining the reference $\boldsymbol{\nu}$:

$$\begin{aligned} \min_{u} \quad & ||W_u^{\frac{1}{p}}\boldsymbol{u}||_p \\ \text{s.t.} \quad & \boldsymbol{\nu}_{ref} = B\boldsymbol{u} \,. \end{aligned} \tag{5}$$

The choice of $p = 2$ results in a convex optimization problem and well-balanced solutions $\boldsymbol{u}^*$ [18]. Furthermore, the solution can be solved explicitly as $\boldsymbol{u}^*_{unc} = G\boldsymbol{\nu}_{ref}$, with the weighted generalized inverse $G = B^+_{W_u} \triangleq W_u^{-1}B^T(BW_u^{-1}B^T)^{-1}$ [19]. This is especially useful if B is constant over various flight conditions, as the inverse can then be computed offline.

There may be cases where we wish to minimize the difference to some preferred input $\boldsymbol{u}_d$, changing the objective function to $||W_u^{\frac{1}{p}}(\boldsymbol{u}-\boldsymbol{u}_d)||_p$. However, by introducing the variable $\boldsymbol{z} = \boldsymbol{u} - \boldsymbol{u}_d$ and minimizing over $\boldsymbol{z}$, it can be seen that the explicit solution only slightly changes to $\boldsymbol{u}^*_{unc} = \boldsymbol{u}_d + B^+_{W_u}(\boldsymbol{\nu}_{ref} - B\boldsymbol{u}_d)$, if $p = 2$ is maintained. As a final step, $\boldsymbol{u}^*_i = \min(\max(\boldsymbol{u}^*_{unc,i}, \underline{\mathrm{u}}_i), \bar{\boldsymbol{u}}_i)\ \forall i$ as for ganging, but now there are no optimality guarantees that $B\boldsymbol{u}^*$ is close to $\boldsymbol{\nu}_{ref}$ in any sense, which makes it unsuitable in the saturation regime. For example, it causes severe problems when the plant's control authority for one pseudo-control is much lower than for others (as shown for quadrotor yaw control in [11]).

3.3. Constrained Optimization

Since the turn of the century, control allocation methods became popular in research that incorporates the actuator limits as an inequality constraint in the optimization problem. A general way of formulating the problem is a sequential problem, which computes the set of all controls $\boldsymbol{u}$ that minimize the pseudo-control error $\boldsymbol{\nu}_{ref} - B\boldsymbol{u}$. If this set includes more than one element (for instance, because $n > d$, such that $\mathrm{rank}(B) < n$), then a second step balances the actuator effort by minimizing the distance to the preferred command $\boldsymbol{u}_d$:

$$\begin{aligned} &\min_{u\in\mathcal{S}} \quad ||W_u^{\frac{1}{p}}(\boldsymbol{u}-\boldsymbol{u}_d)||_p \\ \mathcal{S} = &\arg\min_{u\in\mathcal{U}} \quad ||W_v^{\frac{1}{p}}\left(v_{ref} - B\boldsymbol{u}\right)||_p \\ \mathcal{U} = \boldsymbol{u} &\quad \text{s.t.} \quad \underline{\mathrm{u}}_i \le \boldsymbol{u}_i \le \bar{u}_i \quad \forall i. \end{aligned} \tag{6}$$

This problem, found, for instance, in [20], is guaranteed to find a solution that satisfies $\boldsymbol{\nu}_{ref} = B\boldsymbol{u}$ if $\boldsymbol{\nu}_{ref}$ is within the capabilities of the platform, and it finds the closest approximation if it is not. The solution returned is always the most 'balanced' out of the available minimizers of the pseudo-control error. The desired vector $\boldsymbol{u}_d$ need not be static, but may be chosen to meet application needs: in aircraft control, minimal radar cross-section [21], most efficient trimmed flight command [20], or most efficient steady-state command $u\boldsymbol{u}_d = B^+_{W_{ss}}\boldsymbol{\nu}_{ref}$ [8] is possible.

For the quadratic case, $p = 2$, and there are positive definite weighing matrices W_u, W_v; this is a convex optimization problem with one unique solution. A solution algorithm was presented in [20] that turns the problem into two equality-constrained and bound-constrained least-squares problems, but it was found to converge slower than other alternatives. However, research in optimization with lexicographic problems such as (6) seems to have intensified in recent years, especially in the field of robot simulation and control, so potentially improved algorithms are available; see, for instance, [22–24].

3.3.1. Weighted Minimization

The sequential problem (6) can be approximated well by forming a linear combination of the two objectives (reducing error and reducing actuator cost) as a single objective. The allocation error is then prioritized using a small scalar multiplier γ, establishing a hierarchy similar to the 'Big M' method for equality constraints in linear programming. This method is also known for solving equality-constrained least-squares problems, such as (5) with $p = 2$ [sec. 5.1.5.] in [25]. The author warns of ill-conditioning with too small values of γ, which has to be chosen as a trade-off between numerical errors and approximation errors when the scalar chosen is too low. This is especially relevant since most embedded hardware on smaller UAV support only 32-bit floats.

The weighted minimization formulation is given by

$$\begin{aligned} &\min_{u} \quad ||W_\nu^{\frac{1}{p_\nu}}(B\boldsymbol{u} - \boldsymbol{\nu})||_{p_\nu} + ||\gamma^{\frac{1}{p_\nu}} W_u^{\frac{1}{p_u}}(\boldsymbol{u}-\boldsymbol{u}_d)||_{p_u} \\ &\text{s.t.} \quad \underline{\mathrm{u}}_i \le \boldsymbol{u}_i \le \bar{\boldsymbol{u}}_i \quad \forall i\,; \end{aligned} \tag{7}$$

compared with [20].

In the literature, different norms have been proposed in the context of control allocation since inception, with the most common ones being ℓ_1, ℓ_2, and ℓ_∞ [26,27]. Physical consideration and the selection of actuator state variables may motivate other norms for the actuator penalty: the power consumption of ship propellers is generally cubic in their speed [3], while the fuel-flow required in turbojet engines can be near-linear to the thrust demanded [28].

In general, ℓ_2 has the advantage that it provides a strictly convex objective function with a single optimal solution. The load is also naturally balanced between the actuators since large deviations in the distribution are avoided, even if the sum of the absolute errors of two possible distributions is equal (i.e., their ℓ_1 norms are equal). However, quadratic solvers generally require more operations to find the unique solutions, whereas ℓ_1 and ℓ_∞ have the advantage that they can be formulated as a linear program that can be solved with fast methods such as the simplex algorithm.

It is also possible to use a different norm for each of the two terms (actuator usage penalty $W_u(\boldsymbol{u} - \boldsymbol{u}_d)$ and pseudo-control error $W_v(\boldsymbol{\nu}_{ref} - B\boldsymbol{u})$), resulting in nine different possible combinations. Their possible advantages and disadvantages along with possible solution algorithms are listed:

Pseudo-Control Error: ℓ_1—Actuator Penalty: ℓ_1

Used frequently in the literature (e.g., [16,29]) and favored for the availability of fast solution methods, but the problem does not have a unique optimum, especially when the allocation problem is feasible. This may lead to problems with certain solvers that first fully load some actuators while keeping others at their preferred state. According to [16], this may be regarded as an advantage; however, for UAVs with many similar or even identical actuators, it is highly undesirable to not use those similar actuators in similar ways. Solving relies on reformulating the ℓ_1-norms into a standard linear programming problem of the form

$$\begin{aligned} \min_{x} \quad & \boldsymbol{c}^T \boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{0} \leq \boldsymbol{x} \leq \boldsymbol{h}\,, \end{aligned} \tag{8}$$

which requires treating the positive and negative parts of the two ℓ_1-norms separately. The decision vector $\boldsymbol{x}$ is of dimension $2n + 2d$ [16] and there are d constraints:

$$\begin{aligned} \boldsymbol{x} &= \begin{pmatrix} \boldsymbol{u}^+ & \boldsymbol{u}^- & \boldsymbol{e}^+ & \boldsymbol{e}^- \end{pmatrix}^T \\ \boldsymbol{h} &= \begin{pmatrix} \overline{\boldsymbol{u}} - \boldsymbol{u}_d & \boldsymbol{u}_d - \underline{\mathrm{u}} & B\boldsymbol{u}_d - \boldsymbol{\nu}_{ref} & B\boldsymbol{u}_d - \boldsymbol{\nu}_{ref} \end{pmatrix}^T \\ \boldsymbol{c} &= \begin{pmatrix} \gamma \mathrm{diag}(W_u) & \gamma \mathrm{diag}(W_u) & \mathrm{diag}(W_\nu) & \mathrm{diag}(W_\nu) \end{pmatrix}^T \\ A &= \begin{pmatrix} -B & +B & +I_d & -I_d \end{pmatrix} \\ \boldsymbol{b} &= B\boldsymbol{u}_d - \boldsymbol{\nu}_{ref} \end{aligned} \tag{9}$$

This problem can be solved with, for instance, simplex [16] or interior point methods [18], and $\boldsymbol{u}^*$ can be recovered as $\boldsymbol{u}^* = \boldsymbol{u}_d + \boldsymbol{u}^+ - \boldsymbol{u}^-$. In terms of computational performance, the authors of [29] expected good scalability for larger platforms, but measured interior-point as around 2.5 times slower than Simplex for $n = 16$, and 3 times slower for $n = 11$. Additionally, IP also does not rely on jumping from corner to corner of the constraint region but rather takes a relatively smooth path from inside the feasible region, which may result in some more even balancing of the elements in $\boldsymbol{u}$ and may allow close-to-optimal premature termination [29].

If a solution to the similar problem is available (i.e., from the solution of a previous sample), then simplex methods can be 'warm-started' with that solution, potentially yielding much lower computational times. With interior point methods, it is harder to make use of that information [29], since a choice of an initial point too close to the boundary of the feasible region may result in slow convergence [30]. Some approaches exist [30,31], but have never been applied in control allocation research.

Pseudo-Control Error: ℓ_1—Actuator Penalty: ℓ_2

This combination was not found in any research so far. Since ℓ_2 is used for the rank(n) actuator penalty, the convexity of the objective function is still maintained, and a single unique solution is returned from an accurate solver. It is unclear whether the use of ℓ_1 significantly reduces the quality of the solution in the context of control allocation, for instance, because of scaling issues between the dissimilar norms of the terms.

Interior-point algorithms can be expected to solve this problem efficiently, since the matrix in the quadratic part of the objective function is diagonal and keeps solving the linear system for the step direction in the interior-point iterations efficiently, which may make it an interesting topic for future research.

Pseudo-Control Error: ℓ_1—Actuator Penalty: ℓ_∞

The authors of [15] used this combination of norms to address the issue that a simple ℓ_1 norm for the actuators does not incentivize balancing between similar actuators. By minimizing the maximum deflection (also called min-max), a load-balancing action is achieved, at least between the heavily loaded actuators.

The resulting system has $4n + 2d + 1$ variables and $2n + d$ constraints, which makes it slightly smaller than the resulting ℓ_∞-ℓ_∞ system. In [15], the scalar upper bound $\tilde{u}$ on the ℓ_∞-norm of the actuator balancing term and the slack variables $(\boldsymbol{s}^+, \boldsymbol{s}^-) > 0$, such that $\tilde{u} = \boldsymbol{u}^+ + \boldsymbol{s}^+$ and $\tilde{u} = \boldsymbol{u}^- + \boldsymbol{s}^-$, were introduced:

$$
\begin{aligned}
\boldsymbol{x} &= \begin{pmatrix} \boldsymbol{u}^+ & \boldsymbol{u}^- & \boldsymbol{s}^+ & \boldsymbol{s}^- & \tilde{u} & \boldsymbol{e}^+ & \boldsymbol{e}^- \end{pmatrix}^T \\
\boldsymbol{e}_{\max} &\triangleq B\boldsymbol{u}_d - \boldsymbol{v}_{ref} \\
\boldsymbol{h} &= \begin{pmatrix} \overline{\boldsymbol{u}} - \boldsymbol{u}_d & \boldsymbol{u}_d - \underline{\mathrm{u}} & \overline{\boldsymbol{u}} - \boldsymbol{u}_d & \boldsymbol{u}_d - \underline{\mathrm{u}} & \max\left(|\overline{\boldsymbol{u}} - \boldsymbol{u}_d|, |\boldsymbol{u}_d - \underline{\mathrm{u}}|\right) & \boldsymbol{e}_{\max} & \boldsymbol{e}_{\max} \end{pmatrix}^T \\
\boldsymbol{c} &= \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & \gamma \mathrm{diag}(W_u) & \gamma \mathrm{diag}(W_u) \end{pmatrix}^T \\
A &= \begin{pmatrix} -B & +B & 0 & 0 & 0 & +I_d & -I_d \\ 0 & 0 & I_n & 0 & -1 & I_n & 0 \\ 0 & 0 & 0 & I_n & -1 & 0 & I_n \end{pmatrix} \\
\boldsymbol{b} &= \begin{pmatrix} \boldsymbol{e}_{\max} & 0 & 0 \end{pmatrix}^T
\end{aligned}
\tag{10}
$$

Pseudo-Control Error: ℓ_2—Actuator Penalty: ℓ_1

This combination was not found in the literature. The solution is not unique, as with $\ell_1 - \ell_1$, but solvers are expected to be much slower since the problem is quadratic as opposed to linear. The same problems with unbalanced actuator values as with $\ell_1 - \ell_1$ are likely to occur. These two arguments combined lead to the conclusion to not further investigate this choice of norms.

Pseudo-Control Error: ℓ_2—Actuator Penalty: ℓ_2

This is the classic bound-constrained least-squares problem. It has been used with success in many works on control allocation dating back to 1994 [9,11,16,20,32]. It is a good candidate since its unconstrained version, Equation (5) (with the weighted generalized inverse as its analytical solution), has been shown to provide satisfactory solutions when actuator limits are not exceeded and it is widely used in commercial and academic aircraft and UAV control packages (PX4, The Paparazzi Autopilot, QCAT [33]). A comparison of solvers and approximations follows.

To approximate the solution of (7), a redistributed generalized inverse may be used, which can be seen as a variation on daisy chaining with an online partitioning of B: when the unconstrained solution $\boldsymbol{u}^*_{unc}$ (computed via a generalized inverse) is not in $\mathcal{U}$, then the violating actuators are fixed at their minimum or maximum values, yielding $\boldsymbol{u}^1$. Those columns are removed from B and the remaining pseudo-control $\boldsymbol{\nu}_{ref} - B\boldsymbol{u}^1$ is re-allocated over the remaining columns B_2. The generalized inverses of B_k resulting from dropping columns can be efficiently computed from the full $B^+_{W_u}$ using the Sherman–Morrison–Woodbury formula [34]. Despite being used in works as recently as 2023 [35], it has been shown to provide unsatisfactory solutions that are sometimes far away from the optimum [20] or jump to very different allocation solutions with just small changes in $\boldsymbol{\nu}_{ref}$.

A fixed-point iteration scheme has been proven to converge to the solution [36], but has been found to be slow in practice [18,20].

Accurately solving this allocation problem online has only been considered computationally tractable with active-set algorithms since the early 2000s [20]. Contrary to [37], it has been used successfully in embedded UAV systems (e.g., [11]), albeit not for platforms with many actuators.

Alternatively, interior-point (IP) algorithms can be used to solve this allocation problem, which may have benefits for larger platforms, as their iterations scale better than linear with the number of variables [38]. When no prior knowledge on the solution is available (the solvers are 'cold-started'), then the cross-over point where IP starts to be faster than active-set seems to be around $n = 15$ [18]. When the solution to a similar problem is known (for instance, from an earlier sample), then active-set can be readily warm-started, which may improve the convergence speed drastically. Warm-starting interior-point methods is difficult, as explained in the ℓ_1-ℓ_1 case.

The authors recommend more research on suboptimal control allocation, where only a few iterations of active-set are performed at every discrete sample. This may have an effect similar to a low-pass filter, as every iteration guarantees a more optimal iterate. If the rate of change in the pseudo-control references $\boldsymbol{\nu}_{ref}$ remains small, then this delay may be negligible for some choice of iteration limit, thereby drastically improving the solution time upper bound.

Pseudo-Control Error: ℓ_2—Actuator Penalty: ℓ_∞

Not found in the literature. Load-balancing can be achieved with the ℓ_∞ norm as seen above, but also with the ℓ_2 norm. Combining the two does not seem beneficial over $\ell_2 - \ell_2$, as quadratic solvers need to be used regardless of the ℓ_∞ norm, which does not help to improve the speed of the algorithm.

Pseudo-Control Error: ℓ_∞—Actuator Penalty: ℓ_1

Not found in the literature. The same problems as $\ell_2 - \ell_1$ are expected, as it will lead to a larger system than $\ell_1 - \ell_1$. Therefore, it was chosen to not further investigate the choice of norms.

Pseudo-Control Error: ℓ_∞—Actuator Penalty: ℓ_2

Also not found in the literature, with equally uncertain advantages over $\ell_1 - \ell_2$. Additionally, reformulating ℓ_∞ norms to be solved with conventional optimization solvers always results in larger problems [15]. These two arguments combined lead to the conclusion to not further investigate the choice of norms.

Pseudo-Control Error: ℓ_∞—Actuator Penalty: ℓ_∞

Minimizing the supremum norm ℓ_∞ to balance control effort has been theorized in the literature for use in control [39] and, more specifically, also control allocation [15], where it was reformulated as a linear program in $4n + 4d + 2$ variables and $2n + 3d$ linear equality constraints. The reader is referred to that work for the definitions of the linear program variables.

This problem is thus larger, and solving it is slower than $\ell_1 - \ell_\infty$, whereas the authors of [15] note "no significant differences" when comparing the quality of the solutions $\ell_1 - \ell_\infty$ and $\ell_\infty - \ell_\infty$; however, they did not provide quantitative analysis.

3.4. Dynamic Control Allocation

If all actuators have similar dynamics (for instance, in a multi-rotor UAV), then the outer loop controller can simply take these into account when computing the pseudo-controls, and we can maintain the algebraic nature of the control allocation as shown above. An example of this for a quadrotor is the incremental scheme proposed in [11]. Sometimes, it may be preferred to use different actuators for different parts of the pseudo-control frequency spectrum: fast or effective actuators to respond quickly to transients, and slower but more energy-efficient actuators during steady-state.

This may be achieved by adding a derivative penalty term to the objective of (7), with a lower penalty for actuators to be used in higher-frequency transients. Conversely, the desired $\boldsymbol{u}_d$ can be chosen as some (unconstrained) command that is optimal in steady-state (i.e., when $\boldsymbol{\nu}_{ref}$ has been reached by the actuators). For instance, [8] recommends $\boldsymbol{u}_s = B^+_{W_{u_s}} \boldsymbol{\nu}_{ref}$, with a steady-state weighing matrix W_{u_s} that deprioritizes inefficient actuators.

If no saturation occurs, then $\boldsymbol{\nu}_{ref}$ is always achieved, and [8] shows that the allocation solution is given by a first-order lag $\boldsymbol{u}^*(t_k) = F\boldsymbol{u}(t_{k-1}) + G\boldsymbol{\nu}_{ref}(t_k)$. If the actuators possess different bandwidths, then the above methods can be used together with lead-lag compensation to match their bandwidths while still ensuring saturation is handled correctly [8]. The authors of [40] show how first-order actuator dynamics can be directly included in inversion-based controllers by choosing a higher-order derivative of the plant output as the definition of ν. While this explicitly compensates for the actuator dynamics, how this may interplay with control allocation methods or actuator saturation was not investigated.

For more complex actuator dynamics, it is an option to use model predictive control allocation [41], but it requires far more computational resources.

3.5. Nonlinear Control Allocation

If there are nonlinearities in the effect of the actuators on the plant, or even couplings between the actuators, then a control-affine form cannot be used. The system to be solved for $\boldsymbol{\delta}$ is then $\boldsymbol{\nu}_{ref} = g(\boldsymbol{x}, \boldsymbol{\delta})$. Of course, a direct nonlinear solution of the system is possible, whereas second-order solvers such as sequential quadratic programming or the Levenberg–Marquardt algorithm are preferred if ℓ_2 norms are used, and the Hessian of $g(\boldsymbol{x}, \boldsymbol{\delta})$ is cheap to compute [42]. This direct approach is also used in the continued work on the dual-axis thrust vectoring rotor platform of [43]. However, the resulting program may be nonconvex and therefore susceptible to local minima and generally slow to solve [4]. Offline nonlinear optimization has been used to approximate the allocation problem with a piecewise-linear function. Nevertheless, when applied to vehicle yaw control in [44], the piecewise-linear function required an order of 10^6 bytes for storage, even when allowing for a $\approx 20\%$ approximation error and only considering four actuators and four state variables.

Linearized Control Allocation

Modeling $g(\boldsymbol{x}, \boldsymbol{\delta})$ as a piecewise-linear function in $\boldsymbol{\delta}$ has gained some popularity in recent years, as it can be solved exactly via mixed-integer linear or quadratic programs (MILP/MIQP). The introduced integer variables allow the branch-and-bound solver to select which segment of the piecewise-linear function governs $g(\boldsymbol{x}, \boldsymbol{\delta})$ at the solution. It has been applied to re-entry vehicles and UAVs, yielding more accurate control in simulation, but without meeting real-time constraints [45], or without mentioning them [10]. Application to autonomous underwater vehicles, at a low update rate of 10 Hz, has been successful [5].

If the derivatives of the nonlinear model can be evaluated cheaply, then a similar on-line approach exists, using a local affine model of the actuators [46] to return to the control-affine form discussed in Section 2.1:

$$v = g(x, \delta_0) + \left(\frac{\partial g}{\partial \delta}\right)_{x,\delta_0} \cdot (\delta - \delta_0) \quad (11)$$

Here, $\left(\frac{\partial g}{\partial \delta}\right)_{x,\delta_0}$ denotes the Jacobian of the control effectiveness $g(\boldsymbol{x}, \boldsymbol{\delta}_0)$ in $\boldsymbol{\delta}$ at the current operating point $(\boldsymbol{x},\ \boldsymbol{\delta}_0)$, which also captures coupling between the actuators (i.e., if the deflection of one actuator influences the effectiveness of another). However, if the solution is far from the operating point or the nonlinearities are large, linearization errors may become significant. The authors of [42] show that linearization errors can dominate other errors in the allocation by a factor of five, when controlling pitch maneuvers near the stall region of a high-performance aircraft with elevator and leading-edge slats. In addition, for the yaw stabilization of road vehicles, large errors were shown in [47].

Larger allocation errors seem to occur when the effect of an actuator on a pseudo-control is non-monotonous, as the derivative term may vanish or switch sign. An example of such actuators are tires [48], or aerodynamic surfaces that can stall. Indeed, in road vehicle control, linear(ized) control allocation seems widely rejected [44,47] and most research focuses on nonlinear optimization using nonlinear tire models.

Nevertheless, the linearized approach is especially useful in incremental control systems, where a command increment $\Delta\boldsymbol{\delta} = \boldsymbol{\delta} - \boldsymbol{\delta}_0$ is sought in response to a pseudo-control delta $\Delta\boldsymbol{v}_{ref}$, eliminating the dependency on $g(\boldsymbol{x}, \boldsymbol{\delta}_0)$. An early approach was presented for a marine vessel in [4], where finite differences were used to generate the gradients/Hessian and the result was identified as a sequential quadratic program of the original nonlinear problem, where each iteration is performed at one sample. The authors of [9] used an onboard spline model to rapidly compute accurate derivatives to incrementally control the challenging Lockheed ICE research object. However, motion restrictions and priority-scheduling had to be placed on the most nonlinear actuators to achieve global convergence with the linearized solver. A nearly identical scheme was presented in [35], where the authors instead used piecewise-linear functions as an approximation of $g(\boldsymbol{x}, \boldsymbol{\delta})$, but avoided the MILP by not enabling switching to different pieces during the solution.

3.6. Summary Table

Table 1 shows a brief overview of the capabilities and limitations of the above listed methods.

Table 1. Summary of the features of the investigated methods.

Method	System Model	Features and Limitations
Ganging	Control-affine	Every δ_i may affect one axis only, but for those systems guaranteed to find optimal solution. Very limited application. Redundant actuators move together.
Daisy Chaining	Control-affine	Allows for setting a static hierarchy of redundant actuators.
Generalized Inverse	Control-affine and equal bandwidth actuators	Allows for prioritization of actuators. Cannot deal with saturation. Single inverse operation, so fast computation. Unable to find a solution that satisfies $v_{ref} = B\boldsymbol{u}$ in cases where actuator saturation takes place.
Linear Optimal CA	Control-affine and equal bandwidth actuators	Allows for prioritization of actuators, and of control objectives when $v_{ref} = B\boldsymbol{u}$ has no solution due to actuator limits. If a solution $v_{ref} = B\boldsymbol{u}$ exists, it is found even in cases with saturating actuators. On embedded microprocessors, generally only real-time capable for smaller actuator counts ($n < 15$), ℓ_1 objective norms, or with suboptimal termination.
Dynamic CA [8]	Control-affine	Like constrained optimization, but allows for different actuator prioritization for transients.
Model Predictive CA	Depends on solver (linear or nonlinear)	Can deal with more complex models and constraints, but requires more computation time.
Direct Nonlinear CA	Nonlinear including coupled actuators	Generally slow to solve, especially for nonconvex problems.
Piecewise-linear CA	Linearized a priori in segments	Mixed-integer programming required. Can be faster than direct nonlinear, but still difficult to meet real-time demands.
Locally Affine CA	Linearized online	Solvable with the same algorithms as linear optimal CA. Linearization error may be large depending on the nonlinearities and the rate of change in the pseudo-control input.

4. Conclusions

Control allocation is required in many control systems, such that a considerable body of research exists. Closed-form solutions such as ganging and generalized inverses are used frequently, but require the dynamics of actuators to be fast and their effect on the plant to be linear, uncoupled, or even axis-separated. Furthermore, any static inversion-based control allocation scheme is unable to exploit the full capabilities (the AMS) of a set of actuators due to not considering their limits. Next to exploiting the full capabilities of existing platforms, some novel actuator design methodologies for VTOL aircraft configurations [49] rely on matching the AMS with a required moment set. For this, it would be instrumental to be able to actually achieve that AMS with the allocation strategy used.

For linear and fast actuators, it has been shown that redistributing the remaining control effort from saturated actuators is possible with iterative methods. Formulating a quadratic minimization is most common in the literature, and active-set solvers seem most popular and even real-time capable in some cases with small actuator counts. The redistributed generalized inverse approximation fails to provide results that are always close to the true solution, but research in suboptimal active-set solvers is lacking and may be very effective, especially if using 'warm-starts', with information from a previous sample.

Next to the Euclidean norm ℓ_2, linear ℓ_1 or supremum ℓ_∞ norms are also possible. In a lexicographic minimization problem, where the pseudo-control error is prioritized over actuator balancing, using the ℓ_1 norm for the actuator balancing term is not sufficient for balancing, especially in conjunction with a simplex solver. Combining ℓ_2 for the pseudo-control error with norms other than ℓ_2 is also not recommended since the solver has to be quadratic and dense anyway, and so using ℓ_2 for the actuator balancing term will give the best results. Using ℓ_1 or ℓ_∞ for the pseudo-control error term and ℓ_2 for the balancing may result in solvers that can be optimized using the sparsity of the diagonal Hessian of the balancing term, and have not been researched to the authors' knowledge.

When the speed of the actuator dynamics is not negligible, then methods are available that prioritize the faster or more effective actuators in the transient phase and can be solved with the same algorithms. For mild nonlinearities common in some aerial and undersea vehicles, incremental methods have been proposed that make use of a local linearization and are therefore as real-time capable as linear optimization methods. However, these methods struggle to find accurate solutions when the effect of an actuator is a non-monotonous function over its range or depends heavily on the state of other actuators. In these cases, nonlinear solvers may be required, which are often not usable in real-time digital control.

Author Contributions: Conceptualization, T.M.B., E.J.J.S. and B.D.W.R.; investigation, T.M.B. and E.J.J.S.; writing—original draft preparation, T.M.B.; writing—review and editing, T.M.B. and E.J.J.S.; supervision, T.M.B., E.J.J.S. and B.D.W.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CA	Control Allocation
WGI	Weighted Generalized Inverse
UAV	Unmanned Aerial Vehicle
(e)VTOL	(electric) Vertical Take-off and Landing
AMS	Attainable Moment Set
MILP	Mixed-integer Linear Programming

References

1. Dorset, K.M.; Fears, S.P.; Houlden, H.P. *Innovative Control Effectors (ICE) Phase II*; Technical Report ADB232172; Lockheed Martin Tactical Aircraft: Bethesda, MD, USA, 1997.
2. Ducard, G.J.J.; Allenspach, M. Review of Designs and Flight Control Techniques of Hybrid and Convertible VTOL UAVs. *Aerosp. Sci. Technol.* **2021**, *118*, 107035. [CrossRef]
3. Fossen, T.I. *Handbook of Marine Craft Hydrodynamics and Motion Control*; John Wiley & Sons: Hoboken, NJ, USA, 2021.
4. Johansen, T.; Fossen, T.; Berge, S. Constrained Nonlinear Control Allocation with Singularity Avoidance Using Sequential Quadratic Programming. *IEEE Trans. Control. Syst. Technol.* **2004**, *12*, 211–216. [CrossRef]
5. Grechi, S.; Caiti, A. Comparison between Optimal Control Allocation with Mixed Quadratic & Linear Programming Techniques. *IFAC-PapersOnLine* **2016**, *49*, 147–152. [CrossRef]
6. Hac, A.; Doman, D.; Oppenheimer, M. *Unified Control of Brake- and Steer-by-Wire Systems Using Optimal Control Allocation Methods*; SAE Technical Paper 2006-01-0924; SAE International: Warrendale, PA, USA, 2006. [CrossRef]
7. Lee, S.; Yoon, Y.; Park, J.; Hong, W. Control Allocation of Rear Wheel Steering and Electronic Stability Control with Actuator Failure. In Proceedings of the 2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES), Madrid, Spain, 12–14 September 2018; pp. 1–8. [CrossRef]
8. Härkegård, O. Dynamic Control Allocation Using Constrained Quadratic Programming. *J. Guid. Control. Dyn.* **2004**, *27*, 1028–1034. [CrossRef]
9. Matamoros, I.; De Visser, C.C. Incremental Nonlinear Control Allocation for a Tailless Aircraft with Innovative Control Effectors. In *Proceedings of the 2018 AIAA Guidance, Navigation, and Control Conference, Chicago, IL, USA, 10 August–13 August 2009*; American Institute of Aeronautics and Astronautics: Kissimmee, FL, USA, 2018. [CrossRef]
10. Wang, P.; Zhou, Z. Piecewise Linear Control Allocation for Flying Wing UAV. In *Proceedings of the Second International Conference on Space Information Technology*; SPIE: London, UK, 2007; Volume 6795, pp. 216–222. [CrossRef]
11. Smeur, E.; Höppener, D.; de Wagter, C. Prioritized Control Allocation for Quadrotors Subject to Saturation. In Proceedings of the International Micro Air Vehicle Conference and Flight Competition 2017, Toulouse, France, 19 September 2017.
12. Snell, S.A.; Enns, D.F.; Garrard, W.L. Nonlinear Inversion Flight Control for a Supermaneuverable Aircraft. *J. Guid. Control. Dyn.* **1992**, *15*, 976–984. [CrossRef]
13. Sieberling, S.; Chu, Q.P.; Mulder, J.A. Robust Flight Control Using Incremental Nonlinear Dynamic Inversion and Angular Acceleration Prediction. *J. Guid. Control. Dyn.* **2010**, *33*, 1732–1742. [CrossRef]
14. Durham, W.; Bordignon, K.A.; Beck, R. *Aircraft Control Allocation*; Wiley: Chichester , UK, 2017.
15. Bodson, M.; Frost, S.A. Load Balancing in Control Allocation. *J. Guid. Control. Dyn.* **2011**, *34*, 380–387. [CrossRef]
16. Bodson, M. Evaluation of Optimization Methods for Control Allocation. *J. Guid. Control. Dyn.* **2002**, *25*, 703–711. [CrossRef]
17. Buffington, J.M.; Enns, D.F. Lyapunov Stability Analysis of Daisy Chain Control Allocation. *J. Guid. Control. Dyn.* **1996**, *19*, 1226–1230. [CrossRef]
18. Petersen, J.; Bodson, M. Constrained Quadratic Programming Techniques for Control Allocation. *IEEE Trans. Control. Syst. Technol.* **2006**, *14*, 91–98. [CrossRef]
19. Eldén, L. A Weighted Pseudoinverse, Generalized Singular Values, and Constrained Least Squares Problems. *BIT Numer. Math.* **1982**, *22*, 487–502. [CrossRef]
20. Härkegård, O. Efficient Active Set Algorithms for Solving Constrained Least Squares Problems in Aircraft Control Allocation. In Proceedings of the 41st IEEE Conference on Decision and Control, Las Vegas, NV, USA, 10–13 December 2002; IEEE: Las Vegas, NV, USA, 2002; Volume 2, pp. 1295–1300. [CrossRef]
21. Zhi, J.H.; Chen, Y.; Dong, X.M.; Xue, J.P.; Yi, J. Anti-Windup Weighted Pseudo-Inverse Control Allocation Method Based on Stealthy Characteristic. In Proceedings of the Chinese Control Conference, CCC, Hangzhou, China, 28–30 July 2015; Volume 2015, pp. 2715–2719. [CrossRef]
22. Pfeiffer, K.; Kheddar, A. Sequential Hierarchical Least-Squares Programming for Prioritized Non-Linear Optimal Control. *arXiv* **2023**, arXiv:2302.11891.
23. Pfeiffer, K.; Escande, A.; Gergondet, P.; Kheddar, A. The Hierarchical Newton's Method for Numerically Stable Prioritized Dynamic Control. *arXiv* **2023**, arXiv:2303.04428.
24. Dimitrov, D.; Sherikov, A.; Wieber, P.B. Efficient Resolution of Potentially Conflicting Linear Constraints in Robotics. *HAL* **2015**, hal-1183003.
25. Björck, Å. *Numerical Methods for Least Squares Problems*; SIAM: Philadelphia, PA, USA, 1996.
26. Bodson, M.; Pohlchuck, W.A. Command Limiting in Reconfigurable Flight Control. *J. Guid. Control. Dyn.* **1998**, *21*, 639–646. [CrossRef]
27. Pachter, M.; Chandler, P.R.; Mears, M. Reconfigurable Tracking Control with Saturation. *J. Guid. Control. Dyn.* **1995**, *18*. [CrossRef]
28. Bensel, A. *Characteristics of the Specific Fuel Consumption for Jet Engines*; Hamburg: Aircraft Design and Systems Group (AERO), Department of Automotive and Aeronautical Engineering, Hamburg University of Applied Sciences: Hamburg, Germany, 2018. [CrossRef]
29. Petersen, J.; Bodson, M. Interior-Point Algorithms for Control Allocation. *J. Guid. Control. Dyn.* **2005**, *28*, 471–480. [CrossRef]
30. Gondzio, J. Crash Start of Interior Point Methods. *Eur. J. Oper. Res.* **2016**, *255*, 308–314. [CrossRef]

31. Shahzad, A.; Kerrigan, E.C.; Constantinides, G.A. A Warm-Start Interior-Point Method for Predictive Control. In Proceedings of the UKACC International Conference on Control, Coventry, UK, 7–10 September 2010; pp. 1–6. [CrossRef]
32. Virnig, J.; Bodden, D. Multivariable Control Allocation and Control Law Conditioning When Control Effectors Limit. In *Guidance, Navigation, and Control and Co-located Conferences, Proceedings of the Guidance, Navigation, and Control Conference, Memphis, TN, USA, 17–22 June 1994*; American Institute of Aeronautics and Astronautics: Scottsdale, AZ, USA, 1994. [CrossRef]
33. Härkegård, O. Quadratic Programming Control Allocation Toolbox (QCAT). Version 1.2.1, 2008. Available online: http://research.harkegard.se/qcat/index.html (accessed on 4 November 2021).
34. Hager, W.W. Updating the Inverse of a Matrix. *SIAM Rev.* **1989**, *31*, 221–239. [CrossRef]
35. Rajput, J.; Khan, H.Z.I. Nonlinear Control Allocation Using a Piecewise Multilinear Representation. In *IEEE Transactions on Aerospace and Electronic Systems*; IEEE: New York, NY, USA, 2023; pp. 1–10. [CrossRef]
36. Burken, J.J.; Lu, P.; Wu, Z.; Bahm, C. Two Reconfigurable Flight-Control Design Methods: Robust Servomechanism and Control Allocation. *J. Guid. Control. Dyn.* **2001**, *24*, 482–493. [CrossRef]
37. Su, Y.; Yu, P.; Gerber, M.J.; Ruan, L.; Tsao, T.C. Nullspace-Based Control Allocation of Overactuated UAV Platforms. *IEEE Robot. Autom. Lett.* **2021**, *6*, 8094–8101. [CrossRef]
38. Nocedal, J.; Wright, S.J. *Numerical Optimization*; Springer Series in Operations Research and Financial Engineering; Springer: New York, NY, USA, 2006. [CrossRef]
39. Cadzow, J. Algorithm for the Minimum-Effort Problem. *IEEE Trans. Autom. Control* **1971**, *16*, 60–63. [CrossRef]
40. Steffensen, R.; Steinert, A.; Smeur, E.J.J. Nonlinear Dynamic Inversion with Actuator Dynamics: An Incremental Control Perspective. *J. Guid. Control. Dyn.* **2023**, *46*, 709–717. [CrossRef]
41. Chatrath, K. Vehicle Dynamics Control Using Control Allocation. Master's Thesis, Delft University of Technology, Delft, The Netherlands, 2019. [CrossRef]
42. Tol, H.; De Visser, C.; Van Kampen, E.J.; Chu, Q. Nonlinear Multivariate Spline Based Control Allocation for High Performance Aircraft. *J. Guid. Control. Dyn.* **2014**, 37, 1840–1862. [CrossRef]
43. Mancinelli, A.; Smeur, E.; Remes, B.; de Croon, G. Dual-Axis Tilting Rotor Quad-Plane Design, Simulations, Flight and Performance Comparison with a Conventional Quad-Plane Design. In Proceedings of the ICUAS, Dubrovnik, Croatia, 21–24 June 2022.
44. Tondel, P.; Johansen, T. Control Allocation for Yaw Stabilization in Automotive Vehicles Using Multiparametric Nonlinear Programming. In Proceedings of the American Control Conference, Portland, OR, USA, 8–10 June 2005; Volume 1, pp. 453–458. [CrossRef]
45. Bolender, M.A.; Doman, D.B. Nonlinear Control Allocation Using Piecewise Linear Functions. *J. Guid. Control. Dyn.* **2004**, *27*, 1017–1027. [CrossRef]
46. Oppenheimer, M.W.; Doman, D.B.; Bolender, M.A. Control Allocation for Over-actuated Systems. In Proceedings of the 2006 14th Mediterranean Conference on Control and Automation, Ancona, Italy, 28–30 June 2006; pp. 1–6. [CrossRef]
47. Lu, X.; Yuanlong, C.; Chi, J. A Nonlinear Control Allocation Algorithm for DYC in Distributed-Motor Drive Electric Vehicles Using S-SQP. In Proceedings of the The 26th Chinese Control and Decision Conference (2014 CCDC), Changsha, China, 31 May–2 June 2014; pp. 1530–1537. [CrossRef]
48. Pacejka, H.B. *Tire and Vehicle Dynamics*; SAE International: Warrendale, PA, USA, 2006.
49. Zhang, J.; Söpper, M.; Holzapfel, F. Attainable Moment Set Optimization to Support Configuration Design: A Required Moment Set Based Approach. *Appl. Sci.* **2021**, *11*, 3685. [CrossRef]