

Article

# Attainable Moment Set Optimization to Support Configuration Design: A Required Moment Set Based Approach

Jiannan Zhang \*, Max Söpper and Florian Holzapfel 

Institute of Flight System Dynamic, Technical University of Munich, 85748 Garching bei München, Germany; max.soepper@tum.de (M.S.); florian.holzapfel@tum.de (F.H.)

\* Correspondence: jiannan.zhang@tum.de

**Featured Application:** The developed methods and framework can be used to support effectors configuring and design for novel electric-vertical-take-of-landing (eVTOL) aircrafts, which aims to optimize their control authority, and the results correspondingly provide an analytical basis for the flight control design.

**Abstract:** In this paper, we discuss an attainable moment set (AMS) optimization methodology considering a system's required moment set (RMS). The AMS describes the achievable moments from a system, given its input limits. An RMS, like an AMS, is a convex set in the moment space, describing the required moments for a system to meet the designed mission profile and disturbance rejection requirements. Given the configuration of a system, its mission requirements, and the derived RMS, the proposed optimization maximizes coverage of the AMS onto the RMS, thus ensuring the system possesses the guaranteed controllability to fulfill its required missions from a design level. To achieve this goal, the variables to optimize are chosen as effector settings, such as the installation position and angle of propellers and control surfaces, which effectively change the AMS without a vast impact on major design parameters, such as mass and moment of inertia. Since the optimization includes massive geometry operations of rays intersecting polyhedron, an efficient intersection solver is proposed to speed up the optimization process. The described method is applied to an electric-vertical-take-of-landing vehicle (eVTOL) with eight hover propellers, which delivers a highly improved coverage of the RMS compared to its initial design.

**Keywords:** attainable moment set; required moment set; configuration optimization; eVTOL



**Citation:** 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. <https://doi.org/10.3390/app11083685>

Academic Editor: Jérôme Morio

Received: 5 March 2021

Accepted: 15 April 2021

Published: 19 April 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Attainable moment set (AMS) is a powerful tool to understand a system's control authority. The concept was initially proposed in [1,2] to solve the direct control allocation problem from geometry intuition. It describes the reachable set of moments generated by a system given its inputs limits, e.g., limits of actuators' position and rate. For a linear time-invariant system, the AMS is a convex set defined in the  $\mathbb{R}^3$  moment space, as:

$$\mathcal{A}_M = \{m \mid m = Bu, \underline{u} \leq u \leq \bar{u}\}, \quad (1)$$

where  $m \in \mathbb{R}^k$  is the moment vector;  $u \in \mathbb{R}^m$  is the input vector ( $k$  and  $m$  are the number of moments and inputs, respectively);  $B \in \mathbb{R}^{k \times m}$  is the so-called effectiveness matrix representing each input's moment generation capability;  $\underline{u}$  and  $\bar{u}$  are respectively the lower and upper limits of the inputs. This concept can be extended to other quantities, such as generalized forces or pseudo controls, e.g., direct forces or angular accelerations, as long as the quantity is algebraically related to the system inputs [3].

The AMS has a big impact on the performance of a system from a control perspective. It poses limits on the maximum achievable time derivative of states, such as translational and angular accelerations, thus constraining maneuverability and agility of trajectories and

mission profiles, as well as the capability of disturbance rejections. Often a limited AMS poses risks on the safe operation of the system in off-nominal situations, e.g., effector or power system failures. As a consequence, a system with relatively big and redundant AMS means better safety margin against extreme cases.

What influences the AMS of a system is however fixed at the design phase by sizing and positioning of the effectors. Therefore, appropriately configuring the effectors at early design phase is critical to the system's performance. For conventional airplanes, the freedom of such effector designs is very limited given major design requirements such as stability and endurance. However, this freedom becomes largely expanded for novel aircraft architectures, such as electrical-vertical-take-off-landing (eVTOL) vehicles, which is often a hybrid system capable of both hover and high-speed flight, exploiting large number of redundant inputs in the form of distributed electric propulsion (DEP) for a bigger flight envelope and additional safety features. As a consequence, optimization can be utilized to find the best combination of effector settings without violating major design parameters such as mass and moment of inertia of the system.

Several studies have incorporated optimization to refactor settings of effectors or to directly search for the optimal AMS of the system. In [4], a trim analysis is performed for a conventional aircraft enhanced with DEP over wingspan, showing that the size of the vertical tail can be reduced by 45% by exploiting differential thrust without adversely affecting directional stability. The idea is further explored in [5] together with a  $\mathcal{H}_\infty$  control methodology in a sequential optimization process, which reduces the area of vertical tail by 60% and lowers the actuator bandwidth, while maintaining the overall control loop bandwidth. In [6], a propeller optimization is applied to a DEP enhanced airplane configuration to decide for the optimal propeller diameter, which shows up to 80% of take-off distance reduction compared to the two-propeller conventional counterpart. As for AMS optimization, a generic AMS framework is proposed in [7] to aid visualize AMSs of complex systems during design phase, which is then conceptually applied to the Max Launch Abort System escape vehicle to help decide the optimal orientation of the thrust nozzles. In [3], a trim routine is developed for an unconventional box-wing configuration to maximize local AMS at operation point. Although it does not directly relate to effector configuration of the system, it implements an optimization framework to exploit best control capability given the system redundancy.

The described methods in the literature aims to optimize a certain target within the design space of redundant inputs from a control's perspective. In this paper, we propose a different optimization framework which directly take into account performance requirements, including mission profiles, the desired control bandwidth of transient maneuver and the disturbance rejection ability, using the novel concept of Requirement Moment Set (RMS). Similar to an AMS, an RMS is a set in the moment space, often convex, which consists of all the moments required by a system to perform its predefined tasks. In order for a system to be capable of fulfilling its tasks, the AMS should fully enclose the RMS, and any violation of this enclosing means failure to meet the requirements otherwise. The RMS can be calculated from the predefined mission trajectory, the transient maneuver bandwidth as well as the desired ability to reject disturbances. The RMS seeks to aid the description of a system's control capability, i.e., AMS, by referencing its relative amplitude with respect to the mandatory moments driven by the mission requirements instead of purely examining the size of an AMS itself. It also provides a good insight into the requirements whether they are too ambitious to the overall vehicle design.

In this paper, we focus on formulation of the optimization problem and address the related algorithm to solve for the geometry coverage problem between the AMS and RMS as two polyhedral. To define the coverage problem, we propose a mathematical definition that represents the margin between the two convex polytopes, and the optimization directly targets to maximize this margin. Calculation of this margin requires finding intersections between rays extending from vertices of one polyhedron with the facets of another polyhedron. The margin needs to be evaluated in each optimization step such is

computationally demanding due to the large number of vertices on a precisely calculated RMS and additionally a large number of facets on the AMS of a highly redundant system. To handle this demanding computation, an efficient intersection solver is proposed to reduce the computation burden. The proposed methods are applied to an eVTOL with eight hover propellers to optimize for their installation angles. Derivation of the RMS is detailed in another submitted paper “Required Moment Sets: Enhanced Controllability Analysis for Nonlinear Aircraft Models” [8], which is directly used for this work as a priori information.

A brief structure of the paper is given here: Section 2 briefs on the theoretical background and approach overview; Section 3 mathematically formulates the optimization problem; Section 4 elaborates on the efficient intersection algorithm; Section 5 discuss the implementation details and the results; Section 6 briefly concludes the paper.

## 2. Methodology Overview

Before we examine the details of the optimization formulation, an overview of the approach’s framework is introduced in this section. We start with a brief discussion of the theoretical background with the important concepts.

### 2.1. Theoretical Background

The most relevant concepts in this problem are the AMS and RMS, which are both defined in the moment space. Additionally, the AMS is derived from the admissible control set (ACS) and the effectiveness matrix. Recalling all relevant definitions from [8,9]:

- **Control Space:** A Cartesian coordinate system in the  $\mathbb{R}^m$  space, with one control effector as a variable on each axis.  $m$  is the number of effectors. Each point in this space represents a combination of effector positions.
- **Admissible Control Set: ( $\mathcal{A}_c$ ):** A set in the control space of all possible combinations of effector positions given their respective limits. Since the effectors are actuated independently, it figures a cuboid or hyper-cuboid in the  $\mathbb{R}^m$  control space.
- **Moment Space:** A Cartesian coordinate system in the  $\mathbb{R}^k$  space, with generalized forces on each axis as variables. The generalized forces do not necessarily have to be moments but also, e.g., angular accelerations, load factors or direct forces. The name “Moment Space” is more of a legacy concept at its initial definition.  $k$  is the number of generalized forces.
- **Attainable Moment Set: ( $\mathcal{A}_M$ ):** A set in the moment space of all possible combinations of generalized forces that can be produced from the effector combinations in the ACS. In this paper, we consider a linear mapping from ACS to AMS, with the so-called effectiveness matrix  $\mathbf{B} \in \mathbb{R}^{k \times m} : \mathcal{A}_c \rightarrow \mathcal{A}_M$ . Since  $\mathcal{A}_c$  is convex, this convexity is preserved by this linear transformation [10], making the AMS geometrically a convex polytope in the  $\mathbb{R}^k$  moment space. The  $\mathbf{B}$  matrix is inherently defined by the configuration especially the effectors, e.g., the moments produced by a propeller is dependent on its lever arm with respect to center of gravity and the angle of installation.
- **Required Moment Set:** A set in the moment space, incorporating all the combination of generalized forces desired to fulfill the prescribed system requirements and missions. These requirements and missions could include fulfillment of certain trajectory profiles, bandwidths of control reaction or disturbance rejections. The RMS is inherent from major system parameters such as mass and moment of inertia, as well as the prescribed requirements, and will remain unchanged once the concept of operations and preliminary design are fixed.

### 2.2. Calculation of AMS and RMS

The calculation of an AMS given an ACS and the  $\mathbf{B}$  matrix have been thoroughly discussed in [1,2,9]. Since the AMS is a convex set composed of the mapping of all the points from the ACS, it can be easily computed by evaluating the produced moments at all

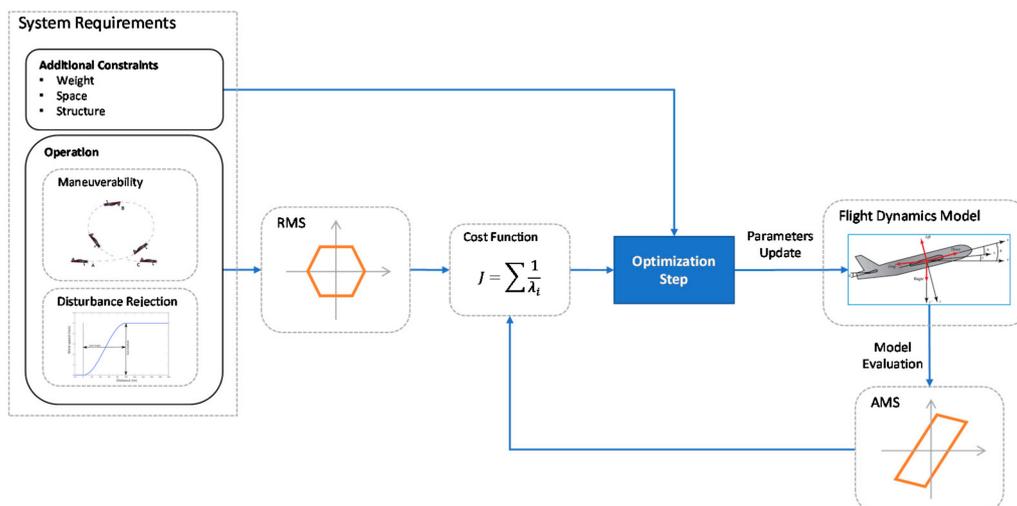
the ACS vertices (extremes of inputs) and then determine the convex hull as the exterior of the AMS by the readily developed *convhull* function in MATLAB® [9,11,12].

The calculation of RMS is elaborated in [8]. The idea is to evaluate disturbance perturbed dynamic models at each trim point and find the hull of the moments of all the perturbed models as the disturbance required moment set, and then generate a set of linearly spaced query directions to evaluate the maneuver required moments, which is assembled with the former one. The result is a convex polytope in the moment space with vertices on the query directions as an unconservative estimation of all the required moments.

### 2.3. The Optimization Framework

It is obvious that we can influence the AMS from a design level, especially with the effectors' settings, to manage its relative capacity with respect to the RMS as the RMS will remain primarily unchanged given the requirements. Intuitively, a good design should guarantee that the system's AMS completely encircles the RMS, such it is able to accomplish the tasks given an appropriately designed flight control system. This assignment seems trivial for conventional airplanes with three independent sets of control effectors impacting each axis in the moment space, however, when aerospace systems evolve in to modernized novel styles with multiple redundant and different types of effectors, the possibility of AMS optimization arises tremendously.

The framework to this optimization goal is depicted in Figure 1 as a block diagram. We start with formulating the AMS from some initial design parameters and evaluate the cost function. Given the requirement derived RMS, the cost function mathematically defines the coverage problem of the AMS as a polytope onto the RMS as another polytope. The optimizer then tries to reduce this cost function by updating some effector parameters considering certain system related constraints such as mass, space, and structure stiffness. With the updated parameters, we reformulate the AMS of the flight dynamics model and loop the process, until the parameters converge.



**Figure 1.** Workflow overview of the proposed optimization framework. The RMS of a system is inherent from the prescribed requirements, while the AMS can be optimized by configuring effectors. To guarantee the ability of mission fulfillment, the AMS should maximally enclose the RMS from design point of view, which is therefore the optimization goal.

The most important step in this process is to mathematically describe the coverage problem of two polytopes in the cost function. We will narrate in detail the formulation of this problem in the next chapter.

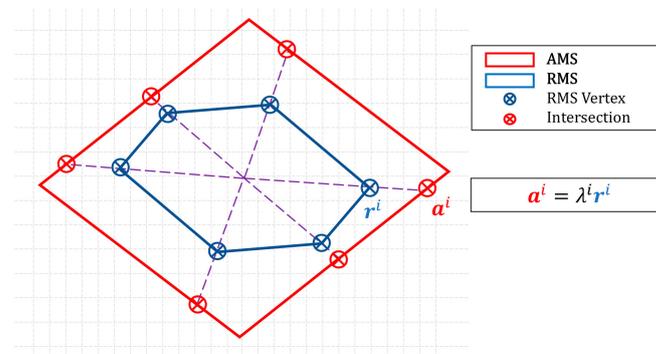
### 3. Optimization Formulation

Given two polytopes of the AMS and the RMS, mathematical definition of the coverage problem can be explained with the aid of an example draft in Figure 2. The red and blue polygons in the figure represent a 2D AMS and RMS, respectively. Due to the convex property, a complete coverage can be guaranteed when all the RMS vertices are enclosed by the AMS. For each RMS vertex, a margin factor is defined as

$$\lambda^i = \frac{\|a^i\|}{\|r^i\|}, \tag{2}$$

where,

- $r^i$  is the vector from origin to the  $i$ -th RMS vertex,
- $a^i$  is the intersection point of a ray extension from  $r^i$  onto the AMS's boundary,
- $\|\cdot\|$  is the 2-norm of a vector.



**Figure 2.** A 2D illustrative comparison between the AMS and the RMS. In 2D, both AMS and RMS reduce to convex polygons. In higher dimensions, they are both convex polyhedrons.

Note that this margin factor  $\lambda$  is a positive number and will be bigger than 1 if a vertex of RMS is bounded inside the AMS, and it is smaller than 1 if the vertex is beyond the AMS. The AMS's coverage over the RMS is therefore guaranteed if all the  $\lambda$  is bigger than 1.

Recall that we can influence the AMS (shape or capacity as a polytope) by changing the effector related parameters, therefore, we can optimize the AMS such that it fully encircles the RMS. To maximize overall coverage and punish vertices with small margins, the optimization problem can be defined as

$$\begin{aligned} \min_p J &= \sum_{i=1}^n \frac{1}{\lambda^i}, \\ \text{subject to, } &\begin{cases} \underline{p} \leq p \leq \bar{p} \\ c(p) \leq \bar{c} \end{cases}, \end{aligned} \tag{3}$$

where,

- $n$  is the total number of vertices on the RMS,
- $p$  is the vector of variables to optimize, with  $\underline{p}$  and  $\bar{p}$  the lower and upper bounds of  $p$ ,
- $c$  is the vector of some additional constraints, e.g., allowance of additional mass or space, with  $\bar{c}$  is the limits of  $c$ .

The key to this optimization problem is to calculate  $\lambda^i$  for each RMS vertex and evaluate the cost function. For problems in 3D or higher dimensions, calculating this intersection and finding  $\lambda$  is not at all trivial. Given the case of linear effectiveness, the AMS of a system with  $m$  inputs has  $2C_m^2$  facets, whereas the intersection is only located on one facet among all. To find the intersection, the most straightforward solution is to brutal search among all the facets. In a  $\mathbb{R}^3$  moment space, assume three vertices on one facet

of the AMS represented by 3D coordinates  $\vec{a}$ ,  $\vec{b}$ , and  $\vec{c}$ , and assume the query direction represented by the coordinate of an RMS vertex is  $\vec{d}$ . To solve for the margin factor  $\lambda$  of  $\vec{d}$  with respect to facet  $\{\vec{a}, \vec{b}, \vec{c}\}$ , the following set of equations needs to be solved:

$$\begin{aligned} \alpha_1 \vec{a} + \alpha_2 \vec{b} + \alpha_3 \vec{c} &= \lambda \vec{d} \\ \alpha_1 + \alpha_2 + \alpha_3 &= 1 \end{aligned} \quad (4)$$

It is a 4-by-4 linear system such that  $\alpha_{1,2,3}$  and  $\lambda$  can be solved for. The only valid solution is when  $\alpha_{1,2,3} \in [0, 1]$ , which means the intersection is encircled by the three vertices of the facet. Note that for each RMS vertex, this equation needs to be solved for all the  $2C_m^2$  facets of the AMS in the worst case, and in each optimization step all the RMS vertices need to be considered to evaluate the cost function. From [8], the RMS takes  $12 \times 12 \times 24 = 3456$  vertices to precisely represents its contour, and the AMS of a system with only 8 inputs has 56 facets. This means to evaluate the cost function defined in (3) for an 8-input system,  $3456 \times 56 = 193,536$  4-by-4 linear systems must be evaluated in each optimization step for the worst case.

Another solution to the intersection problem is to formulate it as an optimization problem by itself:

$$\text{subject to, } \begin{cases} \max \lambda, \\ \mathbf{B}\mathbf{u} = \lambda \mathbf{d} \\ \mathbf{u} \in \mathcal{A}_c, \lambda \mathbf{d} \in \mathcal{A}_M \end{cases} \quad (5)$$

This problem is also known as direct allocation in the control allocation domain, which can be transformed into a standard linear programming problem and solved by either simplex algorithm or interior-point algorithm [1,13,14]. However, this also means to solve for 3456 times linear programming problem for all the RMS vertices in each step of our “outer” optimization framework and is therefore not an acceptable means.

In this paper, we focus on the application in  $\mathbb{R}^3$  space of the optimization framework and propose an efficient 3D intersection solver to tackle the computation issue. As will be elaborated in the next chapter, the method starts from geometry intuition to locate the intersection facet by coordinate transformation and dimension reduction. Instead of solving a 4-by-4 linear system for each facet, the method solves only a point in triangle problem in  $\mathbb{R}^2$  space. Once the facet is determined, the intersection is easily located by “ray-tracing” solver often used in the computer graphics domain, which only involves vector dot products.

Given correct evaluations of the cost function, the optimization problem can be solved by surrogate optimization functions provided by MATLAB [11].

#### 4. Efficient Intersection Solver

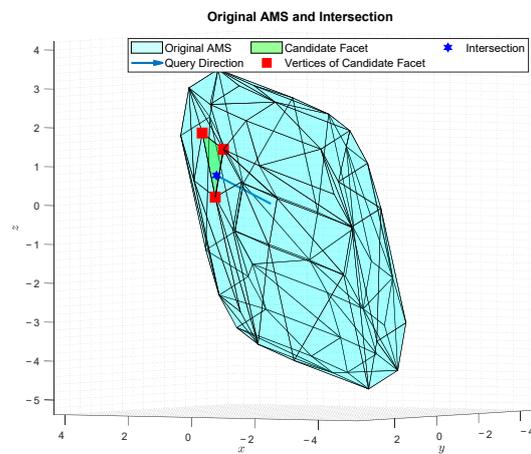
The problem of intersection is summarized as follows: for an AMS (a convex polyhedron) and a ray (starting from the origin) in 3D space, find the intersection point of the ray onto the AMS boundary. In our specific problem, the ray direction is defined by a vertex of the RMS. The difficult step is to find which facet the ray intersects with. Once the facet is found, the intersection point can be solved as a ray tracing problem [15].

An example of the problem is shown in Figure 3a. The problem can be simplified by rotating the original coordinate system representing the moment space and correspondingly a dimension reduction. Given the query direction of the ray, a coordinate transformation by rotation can be applied such that the ray aligns with the  $x$ -axis in the transformed coordinate, as shown in Figure 3b. By doing so, the complexity of the problem is immediately reduced, due to the facts that:

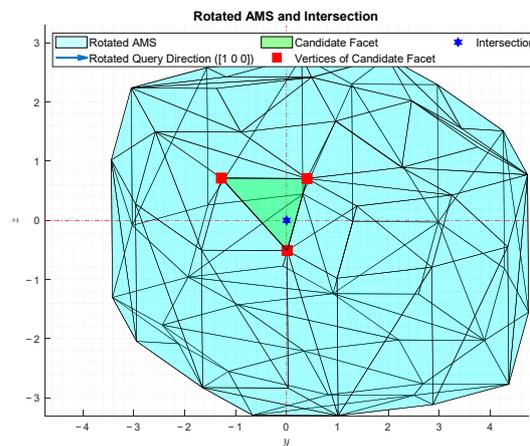
1. After rotation, only facets in the positive- $x$  half (rotated) space can be the intersection facet, meaning the number of search facets is reduced by approximately half.
2. The intersection facet contains the origin in the rotated  $y$ - $z$  plane, meaning the 3 vertices of the facet can only be of one of the following 3 cases:
  - lie in 3 different quadrants of the  $y$ - $z$  plane,
  - only lie in quadrants 1 and 3 in the  $y$ - $z$  plane,
  - only lie in quadrants 2 and 4 in the  $y$ - $z$  plane.

This further reduces the number of possible candidate facets.

3. The fixation for the intersection facet reduces to a 2D “point-in-triangle” problem—the three vertices of the intersection facet must enclose origin in the  $y$ - $z$  plane.
4. Once the facet is found, the final solution can be found by ray tracing.



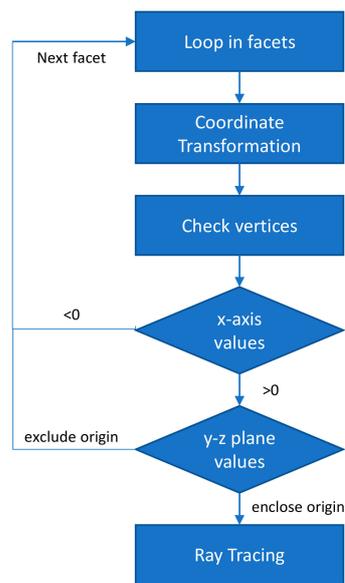
(a) Original AMS and Intersection



(b) Rotated AMS and Intersection

**Figure 3.** Comparison between the (a) original problem and (b) the rotated problem. To calculate the intersection, the priority task is to locate the intersection facet. In the rotated space as of (b), the query direction coincides with the  $x$ -axis. Therefore, the intersection facet can be located by solving a 2D “point-in-triangle” problem, that is, the facet which contains origin is the facet to solve for.

Accordingly, a process diagram to solve for the intersection point is given in Figure 4. Note that, although the loop to search for the facet is inevitable, the calculation in each search iteration is minor compared to solving a set of linear equations. The following sections in this chapter briefly introduce the aforementioned steps.



**Figure 4.** A brief process of the intersection solver. The process is to determine whether a facet is a candidate to solve for, and immediately loop to the next if it is not. This considerably speeds up the optimization compared to a brutal search.

#### 4.1. Coordinate Transformation

The coordinate transformation rotates the space such that the  $x$ -axis is aligned with the query direction. The rotation matrix  $R$  is solely defined by the query direction. Suppose the query direction is defined by a unit vector  $q$ , then applying the transformation to  $q$  gives:

$$Rq = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}. \tag{6}$$

This rotation matrix can be calculated by the Rodrigues’ rotation formula [16], which generates an orthogonal matrix. Applying this rotation to vertices of the AMS is equivalent to a canonical coordinate transformation. In the transformed system, the coordinate of  $q$  is  $[1; 0; 0]$ . Thereafter, the origin-in-triangle-check in the next section is operated within the transformed coordinates.

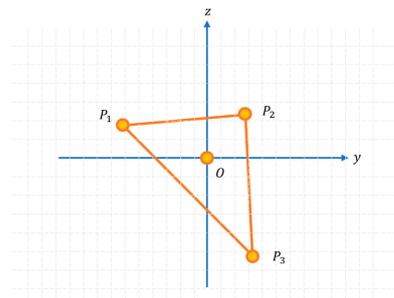
#### 4.2. Origin-in-Triangle Check

Once rotated, it is necessary to check if the origin is enclosed by the facet’s 3 vertices in the  $y$ - $z$  plane. For independent inputs, the ACS is a hyper-cuboid in  $\mathbb{R}^m$  control space, when mapped to  $\mathbb{R}^3$  moment space, some edges of the ACS mapped to the interior of the AMS through the null space of the  $B$  matrix, and parallel edges remains parallel through linear mapping. Therefore, the facets of AMS in  $\mathbb{R}^3$  space are parallelograms [2]. By using the MATLAB command *convhull*, each parallelogram facet is parameterized as two triangles in  $\mathbb{R}^3$  space, which directly serves our purpose of the origin-in-triangle check.

Keep in mind that only facets in the positive  $x$ -axis domain after rotation should be considered. As denoted in Figure 5, the easiest way to find if the origin is enclosed by triangle  $P_1P_2P_3$  is to calculate the barycentric coordinates of origin expressed by vector  $\vec{P_1P_2}$  and  $\vec{P_1P_3}$ , then check if both coordinates are positive and their sum is smaller than 1. Therefore, if coordinate of  $P_i$  in the  $y$ - $z$  plane is  $(y_i, z_i)$  ( $i = 1, 2, 3$ ), then the barycentric coordinate of origin is calculated by:

$$\begin{aligned} u &= \frac{y_1(z_1 - z_3) + z_1(y_3 - y_1)}{(y_2 - y_1)(z_3 - z_1) - (y_3 - y_1)(z_2 - z_1)} \\ v &= \frac{z_1(y_1 - y_2) + y_1(z_2 - z_1)}{(y_2 - y_1)(z_3 - z_1) - (y_3 - y_1)(z_2 - z_1)} \end{aligned}, \tag{7}$$

If  $u \geq 0$ ,  $v \geq 0$  and  $u + v \leq 1$ , then the origin is enclosed by triangle  $P_1P_2P_3$  in the  $y$ - $z$  plane.



**Figure 5.** D origin-in-triangle check. For a convex problem, there is only one intersection between the polyhedron and the query direction. Therefore, only the intersection facet will contain the origin in the rotated space thus can be determined by this check.

#### 4.3. Ray Tracing for Intersection

The ray tracing problem can be solved in the rotated space given the right facet, as shown in Figure 6. The task is to calculate intersection  $Q$  of the ray extending unit vector  $\vec{m} = [1; 0; 0]$  and the facet  $P_1P_2P_3$ . Assume  $\vec{n}$  is normal vector to facet  $P_1P_2P_3$ , and the length of vector  $\vec{OQ}$  is  $t$ . Then, it is obvious that:

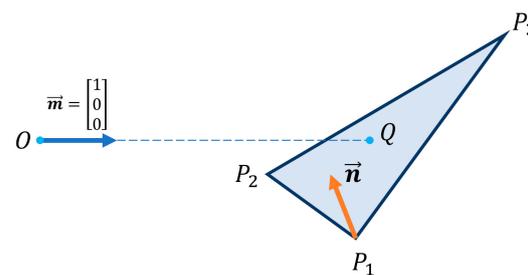
$$\vec{P_1Q} \cdot \vec{n} = 0, \tag{8}$$

therefore,

$$(\vec{P_1} - t\vec{m}) \cdot \vec{n} = 0, \tag{9}$$

such that,

$$t = \frac{\vec{P_1} \cdot \vec{n}}{\vec{m} \cdot \vec{n}}, \tag{10}$$



**Figure 6.** The ray tracing problem known as in the field of computer graphics, which provides a fast and efficient way of calculating the intersection between a query direction and a facet in space.

Note that,  $P_1$  and  $\vec{n}$  are known given the facet.  $t$  is the length of the vector  $\vec{OQ}$ . Since the rotation transformation does not change magnitude,  $t$  is therefore  $\|a\|$  in Equation (2). The margin factor in Equation (2) is therefore calculated as:

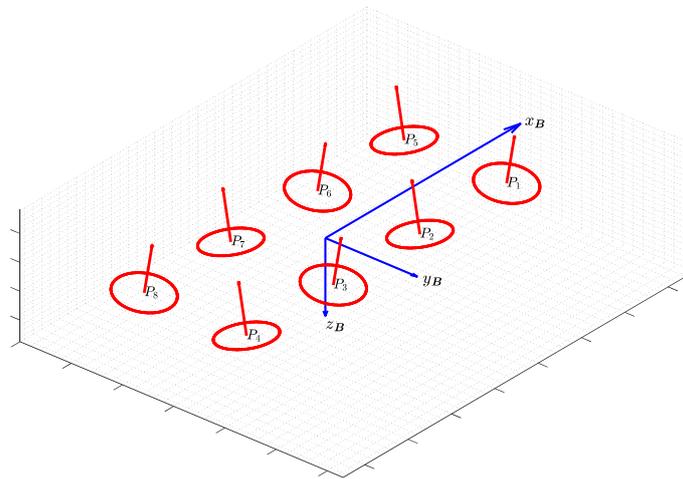
$$\lambda = \frac{t}{\|\vec{r}\|}. \tag{11}$$

We can repeat the process for each RMS vertex to calculate our final cost function in Equation (3). Note that, although the process contains several steps, each step requires minor computation power compared to solving a set of linear equations in higher dimensions, reducing the total amount of computation power needed.

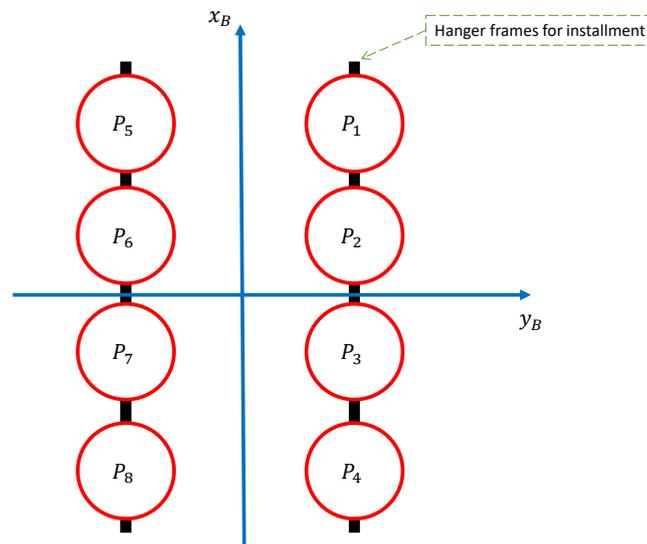
### 5. Implementation and Results

#### 5.1. The Plant Model and the Moment Sets

The proposed framework is applied to an eVTOL to optimize its eight hover propellers' installation angles. The outline of the propellers' arrangement is displayed below in Figure 7:



(a) Propeller Arrangement 3D View



(b) Propeller Arrangement Top View

**Figure 7.** Outline of propeller positions of the considered plant: (a) 3D view, (b) Top view. In (b), the black bars represent two hanger frames where the propeller are installed on. This leads to a natural constraint on the propeller rotation direction, which is only allowed to rotate around the  $x$ - $B$  axis.

The Cartesian coordinate in blue represent the body-fixed frame, denoted by  $B$ . The eight discs denoted as  $P_1$  to  $P_8$  sketch the eight propellers with an arrow representing the directions of forces they produce. The figure demonstrates the propellers respective relative positions and alignment in the  $B$ -frame. Since the propellers are installed on two hanger frames (black bars in Figure 7b) which are parallel to the  $x$ - $B$  axis, the propellers constrained to rotate around the  $y$ - $B$  axis. Therefore, all the propellers are tilted symmetrically around

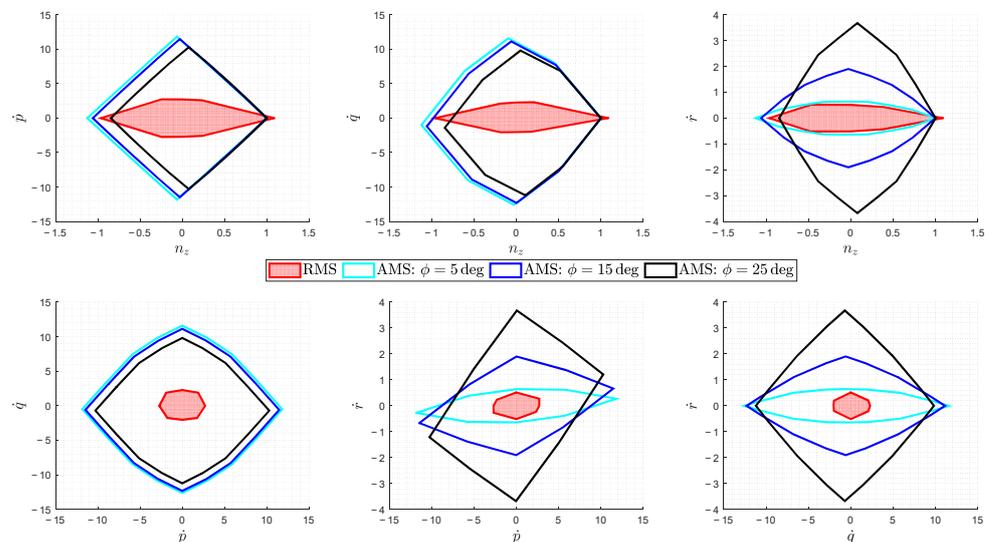
the  $x$ - $B$  axis which also accounts for the otherwise limited yaw authority. Note that the figure is only a conceptual illustration and does not represent actual sizing of the plant considered.

Generally, a linear relationship between the forces and moments produced and the square of rotational rate of the propellers can be assumed [17,18]. For hover control, the most relevant forces and moments are the vertical force along the  $z$ - $B$  direction and three moments. Here we choose the generalized forces of the vertical load factor and angular accelerations as the axis of the moment sets, since they are algebraically equivalent to the vertical force and moments, and they present a more straightforward relationship between the most often controlled variables, e.g., vertical speed and attitude. The mathematical model we consider here is therefore:

$$\begin{bmatrix} n_z \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix}_B = \mathbf{B}(\mathbf{p}) \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \vdots \\ \omega_3^2 \end{bmatrix}, \omega_i^2 \in [0, \omega_{\max}^2] \tag{12}$$

On the LHS,  $n_z$  is the vertical load factor;  $\dot{p}$ ,  $\dot{q}$ ,  $\dot{r}$  are the angular accelerations, subscript  $B$  denotes the quantities in the  $B$ -frame. On the RHS,  $\mathbf{B}$  is the effectiveness matrix, which is a function of the parameters,  $\mathbf{p}$  to be optimized.  $\omega_{1,\dots,8}^2$  are the square of rotation rate of the propellers.  $\omega_{\max}$  is the maximum rotational speed.

The preliminary design tilts the propellers around the  $x$ - $B$  axis symmetrically, as can be seen in Figure 7. Assume the tilt angle is  $\phi$ , then the vector of preliminary parameters is  $\mathbf{p} = [\phi, -\phi, \phi, -\phi, -\phi, \phi, -\phi, \phi]$  for propellers  $P_1$  to  $P_8$ . Figure 8 shows the projections of the  $\mathbb{R}^4$  AMS and RMS into 6  $\mathbb{R}^2$  spaces, when  $\phi$  is  $5^\circ$ ,  $15^\circ$  and  $25^\circ$  respectively.



**Figure 8.** Comparison of RMS and AMSs when the symmetrically tilted angles of propellers is  $5^\circ$ ,  $15^\circ$  and  $25^\circ$  respectively.

Note that when the angle is small, the system shows very limited authority in yaw thus a very small margin with respect to the RMS. When the angle increases, the yaw authority is correspondingly enlarged but the achievable load factors and roll accelerations are noticeably degraded. For all the three cases, the pitch authority in the AMS is slightly changed and remains highly redundant compared to its counterpart in the RMS. Given

these observations, we choose to optimize a  $\mathbb{R}^3$  subset of the  $\mathbb{R}^4$  AMS, using a sub-model of Equation (12):

$$\begin{bmatrix} n_z \\ \dot{p} \\ \dot{r} \end{bmatrix}_B = \mathbf{B}_{\text{sub}}(\mathbf{p}) \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \vdots \\ \omega_3^2 \end{bmatrix}, \omega_i^2 \in [0, \omega_{\text{max}}^2], \tag{13}$$

where,  $\mathbf{B}_{\text{sub}}$  takes the first, second and fourth columns of the original  $\mathbf{B}$  matrix. Correspondingly, the RMS takes its  $\mathbb{R}^3$  projection with the same axes. With this simplification, the proposed intersection solver in Section 4 can be directly utilized.

### 5.2. Verification of Intersection Solver

Before showing the optimization results, the intersection solver proposed in Section 4 is verified against solutions of Equation (5), solved as a linear programming problem, using the solver provided in the “QCAT” toolbox [19]. The toolbox helps to translate the problem into a standard linear programming problem and then solve it using the inherit *linprog* function from MATLAB [11,20].

Totally 10,000 randomly generated cases evenly distributed in the  $\mathbb{R}^3$  space are compared for both solvers. The result is shown below in Figures 9 and 10. The calculated margin factors of the proposed method are mostly the same as the linear programming solution with a precision of  $\pm 2\%$ , but is on average 5–10 times faster as shown in the histogram of time consumptions.

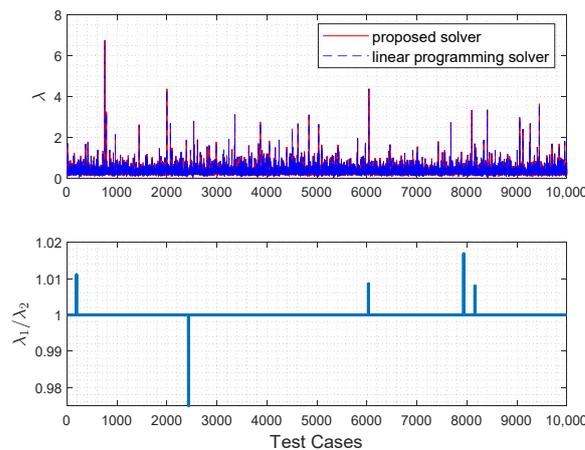


Figure 9. Comparison of margin factor  $\lambda$  calculated by both methods of 10,000 random cases.

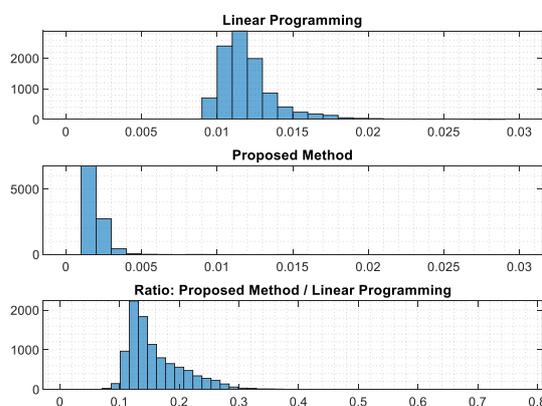
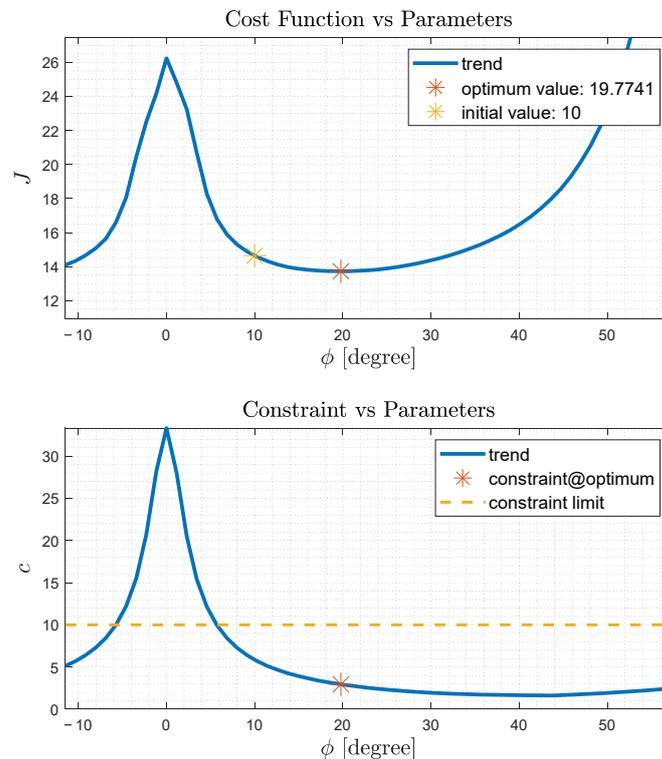


Figure 10. Histogram of time consumption of both methods.

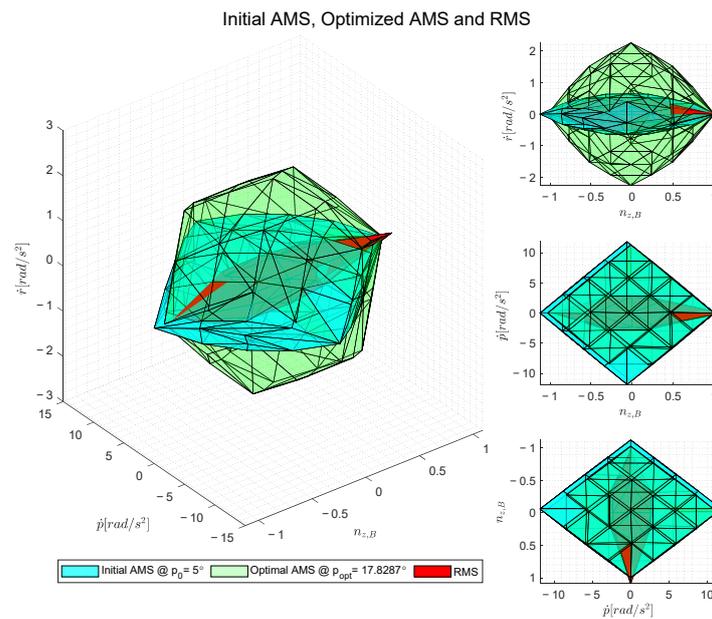
### 5.3. Single Variable Optimization

To verify the optimization framework, we firstly take equally distributed angles, with only one variable  $\phi$  to parameterize the problem as in Section 5.1, such that  $\mathbf{p} = [\phi, -\phi, \phi, -\phi, -\phi, \phi, -\phi, \phi]$ . The signs here represent the rotation direction of each propeller. This single variable setup helps us to verify the optimization results by plotting the relationship between the cost function and the variable  $\phi$ . If implemented correctly, the optimization should generate the minimum value on the cost function curve. Some settings for the optimization, the variable is limited between  $[-10^\circ, 50^\circ]$ , whose initial value is set to 5 degrees. The constraint is chosen as the condition number of the  $\mathbf{B}$  matrix, which must be smaller than a threshold such that the resultant system is NOT ill-shaped in its controllability.

In Figure 11, the trend between cost function and constraint with respect to the optimization variable  $\phi$  is shown. The initial value and optimal value calculated by the optimization framework is marked by yellow and red astray symbols. Clearly, the value calculated by the optimization achieves the global minimum of the cost function and resides well inside the constraint limit. Figure 12 shows a direct comparison between the initial AMS, the optimal AMS, and the RMS in the  $\mathbb{R}^3$  space. The coverage margin of the AMS over the RMS is largely increased, especially in the yaw axis where the initial margin was relatively low. The optimal AMS sacrifice a little performance on the roll axis compared to the initial one, but the resultant margin is still rather high. Note that, the system seems unable to handle the positive corner of the load-factor axis despite the optimal configuration. This reminds of a preliminary design issue, or too ambitious requirements presumed.

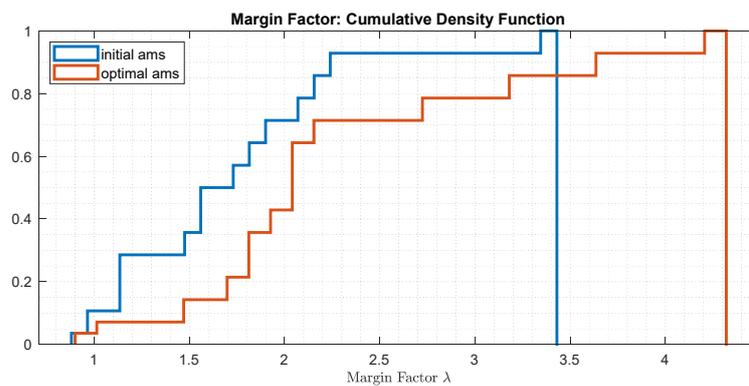


**Figure 11.** Relation of cost function and constraints with the optimization variable, each is a one-to-one mapping since we have only one optimization variable. This figure shows that the optimizer successfully locates the optimal point which also fulfills the constraints.



**Figure 12.** Comparison of RMS (red), initial AMS (cyan), and optimal AMS (green) of the single variable optimization. As can be seen, the optimal AMS largely increase yaw capability without minor sacrifice of the other two axes.

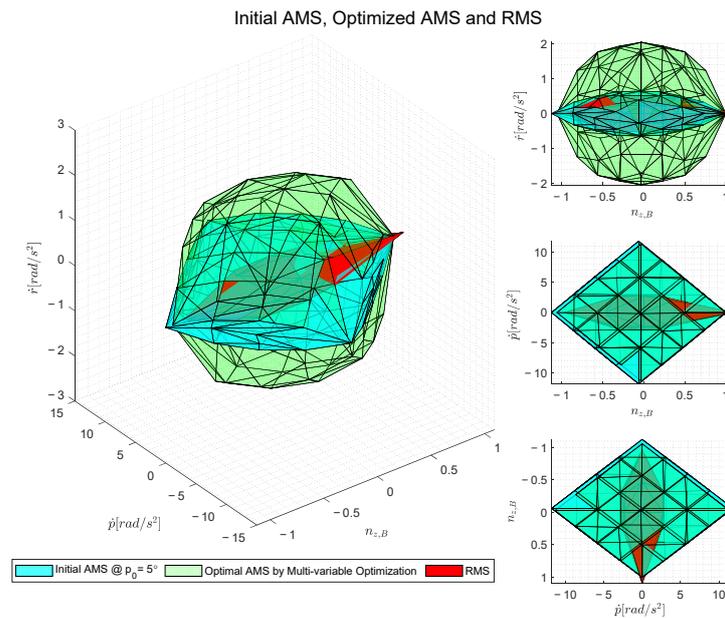
Figure 13 compares further the margin factor cumulative density function. For the initial and optimal AMS, the margin factors of all RMS vertices are gathered and counted to formulate the cumulative function. The curve of optimal AMS is clearly shifted right-ward, indicating an increased probability of safety margin. For the optimal AMS, only less than 10% of the margin factor is smaller than 1.5, whereas the number for initial AMS is more than 30%. Additionally, the biggest margin is increased from 3.4 to 4.3, with a marginal increase in the smallest value.



**Figure 13.** Comparison of margin factor CDF of all the query direction (RMS vertices) between initial and optimal AMS. The overall margin factor of the optimized AMS is elevated compared to the initial one.

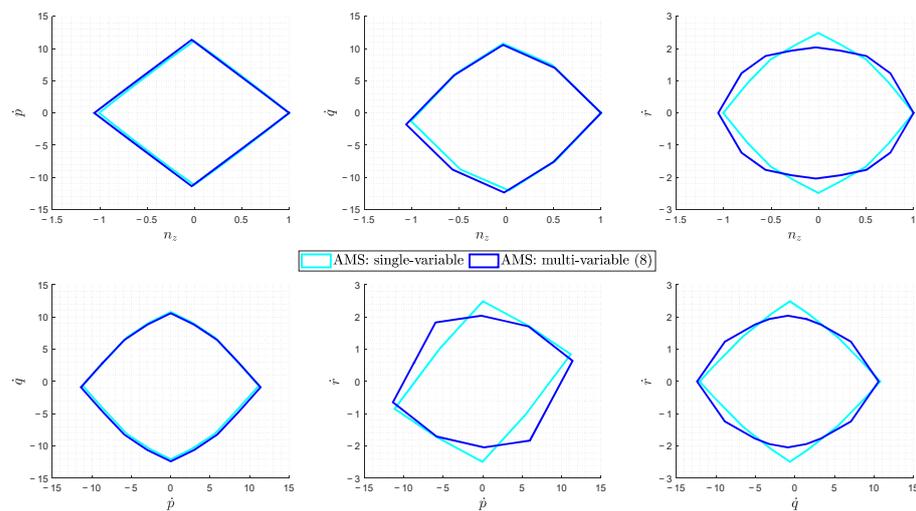
#### 5.4. Multivariable Optimization

Since the framework is verified by the single variable case, we can use it to directly optimize all the 8 angles independently. The parameter vector is therefore  $p = [\phi_1, -\phi_2, \phi_3, -\phi_4, -\phi_5, \phi_6, -\phi_7, \phi_8]$ , where  $\phi_{1-8}$  are the absolute values of the angles and the sign represent the rotation direction. The Figure 14 shows the multi-variable optimized AMS compared to the initial AMS and RMS.



**Figure 14.** Comparison of optimal AMS, initial AMS, and RMS of the multivariable optimization setup.

A comparison between the single variable and the multi-variable optimized AMS are shown below. The corresponding optimal angles, optimal cost function and nonlinear constraint value are for the initial system, single-variable optimized system and multi-variable optimized system are summarized in the table below. Note that the multi-variable optimization achieves a more even distributed space between the axes, especially in the  $n_z - \dot{r}$  plane on the upper-right subplot of Figure 15. The multi-variable optimization is able to achieve a smaller cost function compared to the single-variable case as shown by Table 1.



**Figure 15.** Comparison of single-variable optimized AMS and multi-variable optimized AMS.

**Table 1.** Comparison of optimized angle, cost function and nonlinear constraint of initial, single-variable optimized and multi-variable optimized AMS.

	Initial State	Single-Variable Optimization	Multi-Variable Optimization
Parameter Vector (Angles in degree)	5	19.8	27.8
	−5	−19.8	−4.7
	5	19.8	0.3
	−5	−19.8	11
	−5	−19.8	−27.8
	5	19.8	4.7
	−5	−19.8	−0.3
	5	19.8	−11
Cost Function	17.2	13.7	13.2
Nonlinear Constraint (Condition Number)	11.25	2.9	5.4

## 6. Conclusions

In this paper, an AMS optimization framework to support configuration design is proposed. The optimization considers the RMS of a system which imprints the requirements information such as the disturbance rejection and maneuverability. Given the RMS as a polyhedral, the optimization aims to maximize the overall coverage of the AMS over the RMS, such that the system obtains a larger safety margin, and the prescribed requirements can be fulfilled to its best extent. An efficient intersect solver is proposed due to the high computation resources needed for computing the complex geometry margin between the AMS and the RMS. The framework is applied to an eVTOL with eight rotors for their installation angles. It is firstly verified by a single variable optimization setup where the relationship between cost function and the variable can be plotted clearly. It is then utilized to the multi-variable case to directly optimize for all the eight propeller angles. Optimization results show that the resultant AMS largely improves the coverage margin especially in the low-authority axes without sacrificing overall performance. The framework helps to largely increase the design freedom with enhanced performance.

**Author Contributions:** Conceptualization J.Z. and M.S.; methodology J.Z. and M.S.; software J.Z.; validation J.Z. and M.S.; investigation J.Z.; writing-original draft preparation J.Z.; writing-review and editing, J.Z. and M.S.; visualization J.Z.; supervision F.H.; project administration F.H.; funding acquisition, F.H. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data sharing not applicable.

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

## References

1. Durham, W.C. Constrained control allocation. *J. Guid. Control Dyn.* **1993**, *16*, 717–725. [[CrossRef](#)]
2. Durham, W.C. Constrained control allocation—Three-moment problem. *J. Guid. Control Dyn.* **1994**, *17*, 330–336. [[CrossRef](#)]
3. Varriale, C.; Voskuil, M.; Veldhuis, L.L. Trim for Maximum Control Authority Using the Attainable Moment Set. In *AIAA Scitech 2020 Forum*, 2020/01/05; American Institute of Aeronautics and Astronautics: Reston, VA, USA, 2020; ISBN 9781624105951.
4. Nguyen Van, E.; Troillard, P.; Jézégou, J.; Alazard, D.; Pastor, P.; Döll, C. Reduction of Vertical Tail Using Differential Thrust: Influence on Flight Control and Certification. In *Proceedings of the Advanced Aircraft Efficiency in a Global Air Transport System (AEGATS'18)*, Toulouse, France, 23–25 October 2018; pp. 1–8.
5. Van, E.N.; Alazard, D.; Döll, C.; Pastor, P. Co-design of aircraft vertical tail and control laws using distributed electric propulsion. *IFAC PapersOnLine* **2019**, *52*, 514–519. [[CrossRef](#)]

6. Moore, K.R.; Ning, A. Distributed Electric Propulsion Effects on Existing Aircraft through Multidisciplinary Optimization. In *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*; American Institute of Aeronautics and Astronautics: Reston, VA, USA, 2018; ISBN 978-1-62410-532-6.
7. Pei, J.; Bassett, G.; Grisham, J.; Finch, P.; Toniolo, M.; Miller, L.; Bandu, P. Generic Control Allocation Toolbox for Preliminary Vehicle Design. In *2018 Modeling and Simulation Technologies Conference, Reston, Virginia*; American Institute of Aeronautics and Astronautics: Reston, VA, USA, 2018; ISBN 9781624105517.
8. Söpper, M.; Zhang, J.; Holzapfel, F. Required Moment Sets: Enhanced Controllability Analysis for Nonlinear Aircraft Models. *Appl. Sci.* **2021**, submitted.
9. Durham, W.; Bordignon, K.A.; Beck, R. *Aircraft Control Allocation*; Wiley: Chichester, UK, 2017; ISBN 9781118827796.
10. Bordignon, K.A. *Constrained Control Allocation for Systems with Redundant Control Effectors*; Virginia Tech: Blacksburg, VA, USA, 1996.
11. The Mathworks, Inc. *MATLAB Version: 9.8.0.1451342 (R2020a)*; The Mathworks, Inc.: Natick, MA, USA, 2020.
12. Barber, C.B.; Dobkin, D.P.; Huhdanpaa, H. The quickhull algorithm for convex hulls. *ACM Trans. Math. Softw.* **1996**, *22*, 469–483. [[CrossRef](#)]
13. Johansen, T.A.; Fossen, T.I. Control allocation—A Survey. *Automatica* **2013**, *49*, 1087–1103. [[CrossRef](#)]
14. Bodson, M. Evaluation of Optimization Methods for Control Allocation. *J. Guid. Control Dyn.* **2002**, *25*, 703–711. [[CrossRef](#)]
15. Foley, J.D.; Van, F.D.; van Dam, A.; Feiner, S.K.; Hughes, J.F.; Angel, E.; Hughes, J. *Computer Graphics: Principles and Practice*; Addison-Wesley Professional: Boston, MA, USA, 1996; ISBN 9780201848403.
16. Mebius, J.E. Derivation of the Euler-Rodrigues formula for three-dimensional rotations from the general formula for four-dimensional rotations. *arXiv* **2007**, arXiv:math/0701759.
17. Mahony, R.; Kumar, V.; Corke, P. Multicopter Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor. *IEEE Robot. Autom. Mag.* **2012**, *19*, 20–32. [[CrossRef](#)]
18. Riether, F. *Agile Quadrotor Maneuvering Using Tensor-Decomposition-Based Globally Optimal Control and Onboard Visual-Inertial Estimation*; Massachusetts Institute of Technology: Cambridge, MA, USA, 2016.
19. Härkegård, O. Quadratic Programming Control Allocation Toolbox for Matlab (Version 1.2.1). 2004. Available online: <http://research.harkegard.se/qcat/> (accessed on 1 February 2021).
20. Dantzig, G.; Orden, A.; Wolfe, P. The generalized simplex method for minimizing a linear form under linear inequality restraints. *Pac. J. Math.* **1955**, *5*, 183–195. [[CrossRef](#)]