Proceedings of the 38th Chinese Control Conference
July 27-30, 2019, Guangzhou, China

A Fast Search Direct Allocation Method for Overactuated Systems

Tong Liu*, Hongliang Shen, Haizheng Li, Dongyu Liu
School of Automation, Beijing Institute of Technology, Beijing 100081, P. R. China
E-mail: liutong2002@bit.edu.cn

Abstract: Control allocation is used to distribute the total virtual control instruction among the redundant actuators. This paper focuses on improving the time performance of direct allocation method, and a fast search direct allocation method is presented. To reduce the on-line computations, an improved method of checking only the facets that located in the same octant as the desired moment is proposed. To shorten the search time, we propose a heuristic search method that determines search direction according to the positional relationship between the desired moment and the facet being checked. To verify the time reduction of the fast search direct allocation method, some numerical simulation experiments are taken, and we make the time requirement of the method analogous compared with modified direct allocation method, modified pseudoinverse redistribution method, and linear programming method solved by simplex method. The simulation results show that the computation time of the fast search direct allocation method reduces at least 60% compared with the three aforementioned methods. This method demonstrates the potential for future use in overdrive systems.

Key Words: control allocation, direct allocation, attainable moment set, heuristic search, overactuated system

1 Introduction

By mechanical design, in order to meet fault tolerance and control reconfiguration requirements, there may be more effectors than strictly needed to meet the motion objectives of a given application. The redundant actuators cause infinite commands that meet the control objective, so the control requirements cannot be generated directly by the actuators. Control allocation is used for distribution of total control effort among redundant actuators for satisfying control input constraints and some additional optimization criterion. In the area of attitude control of aircrafts, satellites, etc., where there is large number of control inputs to achieve fast maneuvers, control allocation plays an important role. Control allocation offers the advantage of a modular design for the over-actuated system where the high-level motion control algorithm can be designed without detailed knowledge about the effectors and actuators [1]. And important issues such as input saturation and rate constraints, power efficiency, tear-and-wear minimization can also be addressed in the control allocation algorithm.

The indicators that evaluate the control allocation algorithm are: allocation error between the desired moment (D-M) and the output moment generated by the actuators, memory space for storing algorithm off-line data, and allocation time used to calculate the control instruction according to the desired moment [2]. The basic control allocation methods, such as linear programming (LP), quadratic programming (QP), weighted pseudo-inverse (WPI), and redistributed weighted pseudo-inverse technique (RPI), use some performance measure as the weighted control input and optimize these performance measure to obtain the solution [3–5]. However, these methods cannot be used in situations where the actuators have rate constraints. Daisy chaining, generalized inverse, and pseudoinverse method occupy less memory space, and have superior time performance, but the existence of allocation error within attainable moment set (AMS) and limited allocation set restrict their application [6]. In order to expand allocation set and increase the probability of obtaining an optimal solution, [7] and [8] proposed modified pseudoinverse redistribution (MPIR) method and a sub-gradient optimization (SGO) method, but the allocation error may still exist. More improved control allocation methods are proposed in [9–13], but these methods have not optimized the solution using any additional performance criteria, and they might be used as supplements to other control methods to keep system functioning in case that some actuators failure. Direct allocation(DA) method uses the position constraint of the actuator to calculate AMS in advance, and the boundary of AMS is used to generate actuator command. By establishing a one-to-one relationship between the points on the AMS boundary and control instructions, the DA method can achieve error-free allocation within AMS. But searching for the correct AMS facet that is aligned with DM requires a lot of on-line computation time, which limits its application in the case of high real-time requirements [14].

This paper focuses on improving the time performance of DA method while ensuring no allocation error within AMS. In the process of searching the facet that is aligned with D-M, a new method of limiting the search area based on AMS vertex coordinates to determine the octant of AMS facet is proposed, which decreases the on-line computations greatly. Compared to traditional random search or sequential search methods, this paper proposed a heuristic search method that determines the search direction by the positional relationship between the DM and AMS facet, which performs very well whatever the number of control varies. The performance indicators of a certain type of aircraft are used in simulation experiments to illustrate the improvement of the method.

The rest of the paper is organized as follows. Section II formulates the problem. Section III presents the method of determining the AMS search area, and the heuristic search method is described in detail in this section. The results of simulations are presented in Section IV, and Section V concludes the paper.

This work is supported by Yunnan Provincial Science and Technology Department Key Research Program (Engineering)under Grant 2018BA070.

2 Control Allocation Problem

As mentioned in the first section, the control allocation is very important for aircrafts with redundant actuators. In this paper, we discuss the flight attitude control problem of an aircraft that equipped with m actuators, $m > 3$. The linearized model of the aircraft can be expressed as:

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx \end{cases} \tag{1}$$

where $x \in R^5$ is the states of the aircraft, which generally include the angle of attack, the pitch rate, the angle of sideslip, the roll rate and the yaw rate. The output $y \in R^3$ includes the pitch rate, the roll rate and the yaw rate [15]. The forces and moments generated by actuators are specified by the matrix $B \in R^{5\times m}$. All the matrix A, B and C are determined by aircraft performance. $u \in R^m$ denote the control input vector. For each actuator has both minimum and maximum deflection positions, it can span a convex hull of control space in m-dimensional space:

$$\Omega \in \{u \,| u_{i,\min} \leq u_i \leq u_{i,\max}, i = 1, 2, ..., m\} \subset R^m$$

where Ω is denoted as the control subset, u_i is the i-th element of u, $u_{\min}$ and $u_{\max}$ are the minimum and maximum output limits of u_i respectively. In this problem, we are concerned with how to calculate the control input according to the control objective. So we consider the derivative of y:

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

where CB is referred as the controls effectiveness matrix. The linear transformation:

$$\tau = CBu \tag{3}$$

denotes control allocation rule, where τ is the result moment. A subset in 3-dimensional space, denoted as Φ, is formed when the controls generate moments through the mapping matrix CB:

$$\Phi = \{\tau \,| \tau = CBu, u \in \Omega\} \subset R^3 \tag{4}$$

where Φ is called the AMS consisting of all the moment vectors that are achievable within the control constraints. Since the linear mapping does not change the convexity of the set, Φ is also a convex set in the three-dimensional space. And we will make the following assumption (non-coplanar controls): Every 3×3 submatrix of CB is full rank. This assumption ensures that every point on the boundary of Φ is mapped by a unique control vector u in Ω, and such a system is called a non-coplanar system. In the rest of the paper, we consider the system as non-coplanar.

With these definitions, the DA problem can be stated as: Given a desired moment τ_c, finding a control input u_d in Ω that CBu_d have the same direction with τ_c , and CBu_d is closest to τ_c in magnitude, where τ_c is calculated by high-level control algorithm according to control objective.

3 The Proposed Method

In this section, we propose an improved method for the traditional direct control allocation called a fast search direct allocation method (FSDA). The main purpose of the improvement is to decrease the on-line computations of the algorithm and make it suitable for real-time application. The procedure of the FSDA method is summarized as follows:

Step 1: Calculate the AMS, save the vertices of each facet, and determine the search area.
Step 2: Randomly select an initial facet within the search area and denote it as f.
Step 3: Determine whether τ_c intersects f. If intersects, go to Step 5. If not, go to Step 4.
Step 4: Determine the search direction and denote the next facet in the search direction as f. Go back Step 3.
Step 5: Compute the control input u_d.

The FSDA method is split into off-line computations and on-line computations. Off-line computations construct the AMS and determine the search area. On-line computations search the facet that is aligned with τ_c, and calculate the command of actuators. To decrease on-line computations, we proposed a heuristic search method. It is expected that the FSDA method would guarantee no distribution error within AMS, and perform more efficiently than any other existing allocation method. The detailed proceeds of the FSDA method are described as follows.

3.1 Computation of the AMS and Determination of Search Area

The basic method of AMS construction is locking all the actuators at their extreme position in all possible combinations according to certain rules while allowing two actuators to traverse the range of their possible position. To further explain the procedure of calculation, we rewrite the control efficiency matrix CB to:

$$CB = [cb_1, cb_2, ..., cb_m]$$

where cb_i is the i-th column of CB:

$$cb_i = \begin{bmatrix} cb_{i,1} \\ cb_{i,2} \\ cb_{i,3} \end{bmatrix} \quad , \quad i = 1, 2, ..., m$$

Choosing any two column of CB, then a pair of parallelogram facets placed opposite to each other on the AMS can be decided. For example, we take cb_i and cb_j, and the corresponding two facets are defined as $f_{\max,ij}$ and $f_{\min,ij}$. The normal vector $n_{i,j}$ of $f_{\max,ij}$ is:

$$n_{\max,ij} = cb_i \times cb_j$$

The normal vector of $f_{\min,ij}$ is in the opposite position:

$$n_{\min,ij} = -cb_i \times cb_j$$

Let u_i and u_j traverse their range of values, and other control input variables are valued as follows:

$$k \in \{1, 2, ..., m\} \quad \text{and} \quad k \neq i, j$$

$$\Omega_{\max,ij} : \left\{ u \,\middle|\, u_k = \begin{cases} u_{\max,k}, & \begin{vmatrix} cb_i & cb_j & cb_k \end{vmatrix} > 0 \\ u_{\min,k}, & \begin{vmatrix} cb_i & cb_j & cb_k \end{vmatrix} < 0 \end{cases} \right\} \tag{5}$$

$$\Omega_{\min,ij} : \left\{ u \,\middle|\, u_k = \begin{cases} u_{\max,k}, & \begin{vmatrix} cb_i & cb_j & cb_k \end{vmatrix} < 0 \\ u_{\min,k}, & \begin{vmatrix} cb_i & cb_j & cb_k \end{vmatrix} > 0 \end{cases} \right\} \tag{6}$$

and

$$u_{\min,i} \leq u_i \leq u_{\max,i}$$
$$u_{\min,j} \leq u_j \leq u_{\max,j}$$

Then the two facets $f_{\max,ij}$ and $f_{\min,ij}$ corresponding to u_i and u_j can be determined:

$$f_{\max,ij} : \{v\,|v = CBu, u \in \Omega_{\max,ij}\} \tag{7}$$

$$f_{\min,ij} : \{v\,|v = CBu, u \in \Omega_{\min,ij}\} \tag{8}$$

Taking u_i and u_j to their maximum or minimum value respectively, we can get the four vertices of $f_{\max,ij}$ and $f_{\min,ij}$.

Since there are a total of $C_m^2 = m(m-1)$ combinations in any two columns of CB, the AMS has a total of $m(m-1)$ facets. Calculating each facet of the AMS according to the above method, and constructing the complete AMS. Numbering and saving each facet and each vertex of the AMS, and saving the number and coordinates of the four vertices of each facet. The AMS for a certain type of fighter aircraft is shown in Fig. 1.

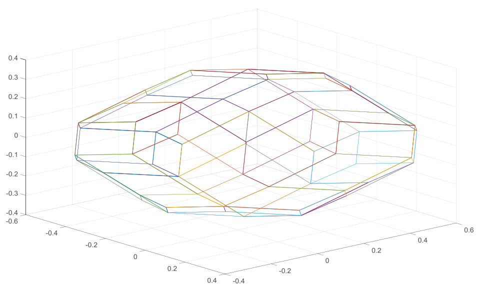

Fig. 1: An example of AMS for a certain type of fighter aircraft

After calculating the complete reachable set, we determine the octant of each facet according to the coordinates of the four vertices. If the four vertices of a facet are in the same octant, the facet is located in this octant. If the four vertices of a facet are in different octants, the facet across these octants. For given a desired moment τ_c, we can search for the facet that τ_c intersects within the octant where τ_c is located. Due to the reduced number of facets involved in the search process, the on-line calculation time is greatly reduced.

3.2 Computation of the Control Input

Since on the boundary of AMS, the moment vector is associated with a unique control vector, and the control input can be obtained by scaling the desired moment so that it reaches the boundary of AMS. Calculating the intersection point, we can invert the corresponding control vector. If the desired moment is larger than the one attainable in the given direction, the moment vector is scaled to the achievable value. If the desired moment is smaller, the control input associated with the maximum attainable moment is scaled to obtain the desired moment [15].

For a given facet, we have calculated its four vertices, then any moment vector on the facet can be linearly represented by the vector from the origin O to one vertex of this facet and the vectors from this vertex to the two adjacent vertices. A desired moment τ_c and a facet f is shown in Fig. 2, then we can write the equation $\tau_c = \alpha_1\overrightarrow{OA} + \alpha_2\overrightarrow{AB} + \alpha_3\overrightarrow{AD}$,

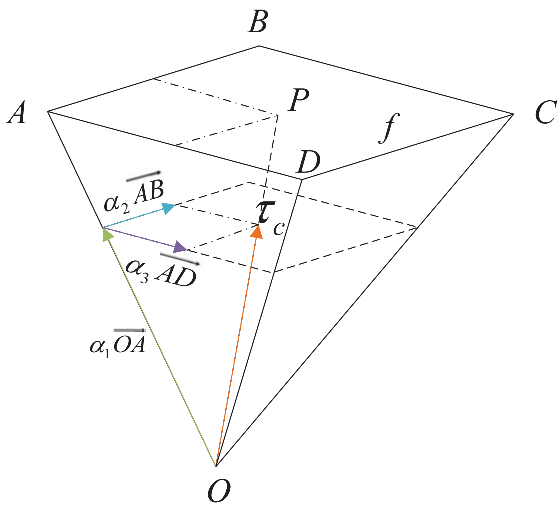

Fig. 2: Extended desired moment intersects the facet

or

$$\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} \overrightarrow{OA} & \overrightarrow{AB} & \overrightarrow{AD} \end{bmatrix}^{-1} \tau_c \tag{9}$$

Since we have restricted the search area, the angle between τ_c and the normal vector of f is less than $90°$, so $\alpha_1 > 0$. Let $\lambda_1 = 1/\alpha_1$, $\lambda_2 = \alpha_2/\alpha_1$, and $\lambda_3 = \alpha_3/\alpha_1$, we can determine whether τ_c intersects f according to the values of $(\lambda_1, \lambda_2, \lambda_3)$. Consider the following conditions:

$$\begin{cases} \lambda_1 > 0 \\ 0 < \lambda_2 \leq 1 \\ 0 < \lambda_3 \leq 1 \end{cases} \tag{10}$$

If these conditions are satisfied, τ_c or extended τ_c intersects f. If not, search for the next facet until the condition is satisfied. The specific search method will be described in detail in the next section. The control vectors associated with the vertices A, B, and D are u_A, u_B and u_D, then the control vector at the boundary is:

$$u_p = u_A + \lambda_2 u_{AB} + \lambda_3 u_{AD} \tag{11}$$

where $u_{AB} = u_B - u_A$, and $u_{AD} = u_D - u_A$.

If $\alpha_1 > 1$, the desired moment exceeds the maximum attainable moment and u_p is taken to control input. If $0 < \alpha_1 \leq 1$, the moment requirement is met by the control input $u_d = \alpha_1 u_p$.

3.3 A New Heuristic Search Method

When calculating the control input, if we use a correct facet, the solution can be obtained by solving a linear system of three equations in three unknowns and linearly combining the three control vectors. If the values of $(\lambda_1, \lambda_2, \lambda_3)$ do not satisfy the conditions (10), τ_c or extended τ_c will not intersect f. If all of the facets are tested sequentially for calculating the intersection, the procedure will be time consuming. To decrease on-line computations, a heuristic search method is proposed.

For the search procedure, we randomly select a facet f as the initial facet, and then determine whether the desired moment intersects f. If the desired moment does not intersect

f, determine the search direction according to the values of $(\alpha_1, \alpha_2, \alpha_3)$. Two typical situations are shown in Fig. 3.

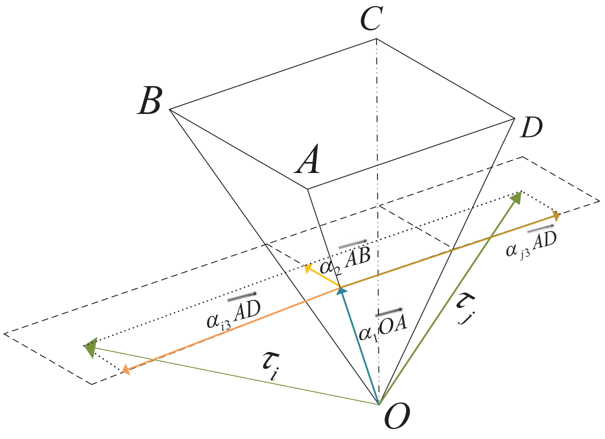

Fig. 3: The desired moment does not intersect the facet

Determine whether τ_i intersects the facet by solving the equation:

$$\tau_i = \alpha_1 \overrightarrow{OA} + \alpha_2 \overrightarrow{AB} + \alpha_{i3} \overrightarrow{AD}$$

In Fig. 3, since we have restricted the search area using the method as mentioned previously, $\alpha_1 > 0$ must be correct, and it is obvious that:

$$0 < \alpha_2/\alpha_1 < 1 \quad and \quad \alpha_{j3}/\alpha_1 > 1$$

The projection of τ_i along the direction of $\overrightarrow{AB}$ opposite to $\overrightarrow{AD}$, thus the facet adjacent to the side AB will be tested next. As for τ_j, the linear equation is:

$$\tau_j = \alpha_1 \overrightarrow{OA} + \alpha_2 \overrightarrow{AB} + \alpha_{j3} \overrightarrow{AD}$$

and we can obtain the conclusions:

$$0 < \alpha_2/\alpha_1 < 1 \quad and \quad \alpha_{j3}/\alpha_1 > 1$$

The extension of the facet along $\overrightarrow{AB}$ touches the extended τ_j, and the facet adjacent to side CD need to be tested next. Therefore, the search direction is determined, and the above method is used until the intersecting facet is found.

The procedure of heuristic search method is summarized as follows:

Step 1: Randomly select an initial facet that denoted as f, and mark the four vertices of f as A, B, C and D.

Step 2: Determine whether the desired moment τ_c intersects f by solving the equation:

$$\tau_c = \alpha_1 \overrightarrow{OA} + \alpha_2 \overrightarrow{AB} + \alpha_3 \overrightarrow{AD}$$

Step 3: If $0 \le \alpha_2/\alpha_1 \le 1$, go to Step 4. Else if $\alpha_2/\alpha_1 < 0$, denote the facet adjacent to the side AD as f, then go back Step 2. Else if $\alpha_2/\alpha_1 > 1$, denote the facet adjacent to the side BC as f, then go back Step 2.

Step 4: If $0 \le \alpha_3/\alpha_1 \le 1$, τ_c or extended τ_c intersects f, then go to Step 5. Else if $\alpha_3/\alpha_1 < 0$, denoted the facet adjacent to the side AB as f, then go back Step 2. Else if $\alpha_3/\alpha_1 > 1$, denoted the facet adjacent to the side CD as f, then go back Step 2.

Step 5: Calculate the control input.

According to the calculated control input, the actuators can control the magnitude and direction of forces generated by the individual effectors to achieve the motion control objectives.

4 Simulation Result

To verify the time reduction of FSDA method, some numerical simulation examples are given in this section. A certain type of aircraft is introduced to illustrate the method, and the control efficiency matrix of it is shown as follows:

$$CB^T = \begin{bmatrix} -0.1508 & 0.1508 & -0.1526 \\ -0.1508 & -0.1508 & 0.1526 \\ 0.1508 & -0.1508 & -0.1526 \\ 0.1508 & 0.1508 & 0.1526 \\ 0.4690 & 0 & -0.1526 \\ 0 & 0.4690 & 0.1526 \\ -0.4690 & 0 & -0.1526 \\ 0 & 0.4690 & 0.1526 \end{bmatrix}$$

The deflection position limits of each control surface are:

$$u_{\max}^T = \begin{bmatrix} -4.189 & -4.189 & -5.236 & -5.236 \\ & -5.236 & -1.396 & -1.396 & -5.236 \end{bmatrix}$$

$$u_{\min}^T = \begin{bmatrix} 1.833 & 1.833 & 5.236 & 5.236 \\ & 5.236 & 7.854 & 7.854 & 5.236 \end{bmatrix}$$

The control efficiency matrix satisfies the assumption that each 3×3 sub-matrix is invertible. The result of the simulation are obtained on a Intel Core i5-4570 machine running at 3.20GHz and using implementations of the algorithms as m-files in Matlab 2017b. Simulation method is: randomly generate a virtual control instruction $\tau_c = \begin{bmatrix} C_l & C_m & C_n \end{bmatrix}$,where C_l, C_m and C_n are the three directions of the virtual control instruction, and calculate the control input by four different methods. The number of simulations is 1000. Timing results are obtained using the tic/toc commands and rough comparison is provided between the FSDA , modified direct allocation method (MDA) [2], MPIR [7] and DA (SIMP) [5]. The simulation results are shown in Table 1:

Table 1: Comparison between FSDA, MDA, MPIR and DA(SIMP)

	FSDA	MDA	MPIR	DA(SIMP)
Minimum time/ms	< 1	< 1	< 1	1
Maximum time/ms	2	17	11.25	53
Average time/ms	0.218	1.1	1.28	4.21
number of errors	0	9	0	0
Average number of facets be checked	4.746	2.528	8	28.276

Table 1 shows the on-line computation time, amount of errors and the average number of facets to be checked before the correct facet is found of the four facets during the test.

From the table we can see that the average time of FSDA reduces by 80.18% compared with MDA, 82.97% compared with MPIR, 94.82% compared with DA(SIMP). The maximum calculating time of FSDA reduces by 88.24% compared with MDA, 82.22% compared with MPIR, 96.23% compared with DA(SIMP). In the system with a high real-time requirement, there is strict limit on the maximum calculating time, and the FSDA method performs better than other three methods. The average number of facets to be checked of FSDA is 4.746, and the number of MDA is 2.528. But MDA costs more time finding the facets near the virtual instruction. And MDA has an error rate of nearly 1%, while this problem does not exist in FSDA. The comparison of the properties between these methods indicates that FSDA has better time performance and is capable of being an alternative to other efficient allocation method.

In another simulation, we use the same aircraft model as the previous simulation and randomly generate a set of virtual control commands that continuously change in three-dimensional space. Real-time calculate the control inputs based on virtual control instructions using FSDA. The instruction dynamic tracking result is shown in Fig. 4. From the picture, we can see that the control input follows the virtual control command very well, and in a few cases, there are deviations in the C_l and C_n directions. The error occurs because the virtual instructions lay outside AMS, and the virtual control instruction cannot be implemented regardless of how the actuators deflect. This problem can be avoided in the design of the control surface and the design of the control law. And FDA well keeps the properties of AMS-based methods (direction preservation, without allocation errors in AMS).

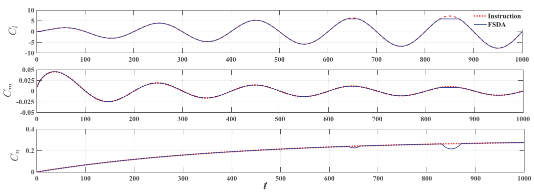

Fig. 4: Instruction dynamic tracking result

Fig. 5 shows an additional characteristic of the method. We test the average time of on-line computations with an increasing number of actuators. Note that FSDA makes the average time stay at nearly the same level with the increasing number of actuators. The three other methods are gradually increasing time to get the results when the number of actuators increases. When the number of actuators is 5, the average time of FSDA reduces 61.85% compared with MDA, 70.45% compared with MPIR, 90.17% compared with DA(SIMP), and as the number of actuators increases, the ratios increase. When the number of actuators is 10, the average time of FSDA reduces 85.24% compared with MDA, 87.47% compared with MPIR, 96.17% compared with DA(SIMP). The results indicate that FSDA is more real-time in performance than the other three methods in control allocation problems with a larger number of actuators.

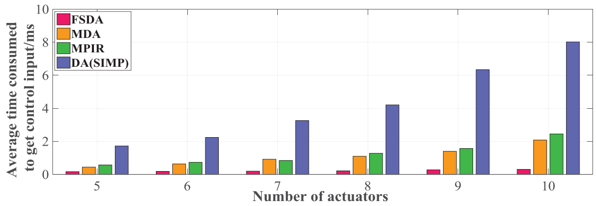

Fig. 5: Average time of on-line computation

5 Conclusions

This paper proposes a more efficiency direct allocation method called fast search direct allocation method. After constructing the AMS, we determine which octant each facet of the AMS is located according to the coordinates of its four vertices, and save as off-line data. Then we can check only the facets that located in the same octant as the desired moment to reduce on-line computation. To shorten the search time, we propose a heuristic search method that determines search direction according to the positional relationship between the desired moment and the facet being checked. The simulation results show that the average time of FSDA reduces at least 60% compared with MDA, MPIR, and DA(SIMP), and FSDA performs better in real-time performance in highly redundant overactuated systems. Additionally, FSDA keeps the properties of AMS-based methods(direction preservation, without allocation error in AMS). The method proposed in this paper can be used for flight control of aircraft with multiple control surfaces as well as other automatic control systems with redundant actuators.

References

[1] Johansen T A, Fossen T I. Control allocationłA survey. *Automatica*, 49(5):1087-1103, 2013.

[2] Tang S, Zhang S, Zhang Y. A Modified Direct Allocation Algorithm with Application to Redundant Actuators. *Chinese Journal of Aeronautics*, 24(3):299-308, 2011.

[3] D.ENNS. Control Allocation Approaches. in *Proceedings of AIAA Guidance Navigation and Control*, 1998: 98-108.

[4] Ola Harkegard. Efficient Active set Algorithms for Solving Constrained Least Squares Problems in Aircraft Control Allocaiton. in *Proceedings of IEEE Conference on Decision and Control*, 2002: 1295-1300.

[5] Bodson, Marc. Evaluation of Optimization Methods for Control Allocation. *Journal of Guidance, Control, and Dynamics*, 25(4):703-711, 2002.

[6] Durham W. Constrained Control Allocaiton: three moment problem. *Journal of Guidance, Control, and Dynamics*, 17(2): 330-336, 1994.

[7] Jin J. Modified Pseudoinverse Redistribution Methods for Redundant Controls Allocation. *Journal of Guidance, Control, and Dynamics*, 28(5): 1076-1079, 2005.

[8] Servidia P A, Snchez Pea R. Spacecraft thruster control allocation problems. *IEEE Transactions on Automatic Control*, 50(2): 245-249, 2005.

[9] Cameron D, Princen N. Control allocation challenges and requirements for the blended wing body. in *Proceedings of AIAA Guidance, Navigation and Control Conference*, 2000: 4539.

[10] Casavola A, Garone E. Adaptive fault tolerant actuator allocation for overactuated plants. in *Proceedings of the American Control Conference*, 2007: 3985-3990.

[11] Harkegard O, Glad T. Resolving Actuator Redundancy-

optimal Control vs. Control Allocation. *Automatica*, 41: 137-144, 2005.
[12] Demenkov M. Reconfigurable direct control allocation for overactuated systems. *IFAC Proceedings Volumes*, 44(1): 4696-4700, 2011.
[13] Kishore W C A, Sen S, Ray G. Disturbance Rejection and Control Allocation of Over-Actuated systems. in *Proceedings of IEEE International Conference on Industrial Technology*, 2006: 1054-1059.
[14] Naskar A K, Patra S, Sen S. A reconfigurable Direct Control Allocation method. in *Procssdings of Control Conference IEEE*, 2013: 1463-1468.
[15] John A.M., Petersen and Marc Bodson. Fast Implementation of Direct Allocation with Extension to Coplanar Controls. *Journal of Guidance, Control, and Dynamics*, 2002, 25(3):464-473.