

Dynamic Control Allocation through Kalman Filtering

¹Gianfranco Morani, ¹Francesco Nebula, ¹Federico Corraro and ²Marco Ariola

¹On-board Systems and ATM, Italian Aerospace Research Centre, Capua, Italy

²Dipartimento di Ingegneria, Università di Napoli Parthenope, Napoli, Italy

Article history

Received: 09-11-2018

Revised: 13-12-2018

Accepted: 08-02-2019

Corresponding Author:
Gianfranco Morani
Italian Aerospace Research
Centre, Capua, Italy
Email: g.morani@cira.it

Abstract: Control Allocation (CA) in aviation is the problem of distributing the commands among the available actuation means, in order to ensure the achievements of the moments requested by the flight control laws. CA plays a key role in fault-tolerant control systems since it gives robust performances and stability also in presence of faults to one or more aircraft actuators. In this study, a new algorithm is proposed for Control Allocation under both static and dynamic constraints. The proposed algorithm aims at overcoming the most common limitations of the existing algorithms, most of which do not account for actuator dynamics (i.e. they compute a control command that might not be compatible with aircraft performance limitations) and rely on iterative methods. The proposed approach does not need an iterative procedure because it rearranges the CA as a state observer problem in which observer states are the actual commands to actuators and observer measurements are the requested moments by the flight control laws. The observer is implemented through a Kalman Filter (KF), with the actuator dynamics as process model and the algebraic relationships between moments and commands as measurement model. The effectiveness of the proposed CA strategy has been shown through a numerical analysis. The numerical simulations showed that the control commands computed by Control Allocation algorithms guarantee moments that match the ones requested by the attitude control laws. This has been verified in nominal conditions (i.e. no actuator faults) but also in faulty ones, where one or more actuators are subject to malfunctioning and, furthermore, in simulations scenarios in which aggressive maneuvers led to the saturation of one or more control surfaces.

Keywords: Control Allocation, Fault-Tolerant Control, Kalman Filter

Introduction

Typically, the problem of designing attitude control laws is split in two sub-problems: closed-loop computation of the moments needed to regulate the vehicle attitude (Control laws) and computation of the commands for the aerodynamic surface actuators, which generate the required moments (Control Allocation) (Harkegard and Glad Torkel, 2005).

In modern aircrafts, the number of control surfaces is typically greater than the minimum required for the vehicle control around the three axes. This redundancy gives more flexibility in choosing the combination of control surfaces that gives the moments required to control the vehicle rotational dynamics. This degree of freedom can be then used in nominal conditions (i.e. when all the control surfaces are available) for minimizing the command effort, but also in faulty

conditions (i.e. when one or more actuators are subject to malfunctioning after a fault event) in order to still guarantee the required moments in spite of faulted actuators. In the latter case, the possibility of distributing the control effort among the “healthy” actuators may preserve vehicle performance and stability, when there is enough control redundancy to attain the requested moments.

The problem of Control Allocation is made even more complicated by the fact that the control actuators are limited in both position and rate. Therefore, the problem of finding a n -ple of control surfaces positions that satisfies the moments requested by the flight control laws is intrinsically a constrained problem.

In the last decades, a great effort has been spent in the design and development of control allocation algorithms based on different techniques. The simplest ones rely on the solution of unconstrained problems and subsequent modifications to account for control surfaces constraints

(Virnig and Bodden, 1994). However, most of the CA techniques try to solve a constrained optimization problem where constraints are imposed by actuator limitations. One of the most appreciated techniques is Direct Allocation (Durham, 1994; Durham and Bordignon, 1996), based on geometric approach where the set of attainable moments is defined and the control commands are computed. Other popular methods concern the use of linear programming, quadratic programming, redistributed pseudo-inverse, active set, etc (Kenneth *et al.*, 1995; Anthony and Marc, 2000; Durham and Bordignon, 1996).

As mentioned earlier, these approaches try to solve a constrained optimization problem where the objective function is usually the norm of the “moment error”, i.e. the difference between the requested and the actual moment vector. In case more than one solution can be achieved (i.e. there are more than one n -ple of control surfaces positions, which ensures a zero moment error), a secondary minimization objective can be set. For a detailed overview of the most common techniques, one may refer to Bodson (2002) and Petersen and Bodson (2006) and references therein.

The above-cited methods solve the control allocation problem by computing control commands that guarantee the achievement of a requested moment vector while taking into account some constraints arising from the actuators limitations (mainly actuator positions and rates). However, the presence of such constraints may not give the possibility to find an analytical solution to the problem. In that case, numerical solutions based on iterative methods can be used but this may lead to the well-known issues related to the convergence and stability of the solutions. Furthermore, the great majority of the mentioned approaches neglects the actuator dynamics, since an instantaneous relationship between control commands and moment vector is assumed. Therefore, the computed control commands may exhibit discontinuities that cannot be followed by the actuation system. To the best of our knowledge, the inclusion of actuators dynamics in CA problem is considered only in the approach described in Luo *et al.* (2007) where the authors present an algorithm based on model predictive control. However, the approach still relies on an iterative procedure.

In this study, a new approach for tackling the control allocation problem is presented. This approach uses a Kalman Filter (KF) for the computation of control commands and it is able to satisfy all the actuators constraints, both static and dynamic. Furthermore, this approach does not suffer from convergence issues since no iterative procedure is used.

KF is usually used for the state estimation of stochastic dynamic systems and it solves an optimization problem where the objective function is the covariance of the estimation error, which obviously depends on the process model, the measurement equations and the

covariance matrices of process and measurement errors. In the proposed CA strategy, the states estimated by KF are represented by the control commands. This is done by including the actuators dynamics in the process model while the requested moment vector is used as an external (virtual) measurement. In this way, actuator dynamics are taken into account by the process model thus making the computed control commands feasible for the actuators dynamics. At the same time, algebraic relationships between desired moments and control surfaces are taken into account in the measurement equations. Process and measurement covariance matrixes are finally used for obtaining a solution that is a good compromise between control effort and control accuracy (i.e. the capability to match the required moments), as it will be better explained later.

As mentioned earlier, the proposed control allocation strategy aims at overcoming the major limitations of the existing approaches. As a matter of fact, it allows to account for dynamic constraints, therefore the control commands are feasible, i.e. they can be followed by the actuation system. In addition, no iterative procedures are required to compute the control commands, thus avoiding the possible lack of convergence of the control allocation solution. This is of paramount importance to guarantee that the proposed algorithm can be suited for hard real-time applications such as the aircraft attitude control.

In the following sections, the problem of control allocation is introduced, the proposed algorithm is described and its effectiveness is shown by means of some numerical analyses.

Problem Overview

As explained earlier, Control Allocation (CA) allows an effective distribution of the requested control effort (the moments computed by flight control laws) among all the available effectors (Fig. 1). More in detail, linearized aircraft dynamics can be written as:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \quad (1)$$

where, $x \in \mathcal{R}^n$, $y \in \mathcal{R}^l$ and $u \in \mathcal{R}^m$ are the state, output and control commands respectively, $A \in \mathcal{R}^{n \times n}$, $B \in \mathcal{R}^{n \times m}$, $C \in \mathcal{R}^{l \times n}$ are the state, input and output matrices respectively. The goal of control allocation is then to find the control commands \bar{u} such that the requested *moment vector* M_{des} is achieved, i.e. $C B \bar{u} = M_{des} \in \mathcal{R}^l$.

With a sufficient number of control inputs, i.e. $m \geq l$ and without any constraint on the control authority, it is always possible to find a control input vector u for which $C B u = M_{des}$. However, considering that actuators are subject to position and rate limits, the solution might not exist. For this reason, the control allocation problem is usually set as a constrained optimization problem, that is:

$$\bar{u} = \min_{u_{\min} \leq u \leq u_{\max}} \|W_v(CBu - v_{des})\|_2 \quad (2)$$

where $\|\cdot\|_2$ is the L_2 -norm and W_v is a non-singular weighting matrix. In other words, the difference between the requested moments M_{des} and the actual ones CBu is minimized while taking into account the constraints on actuators positions and rates.

As mentioned earlier, the control allocation is even more important in case of actuator faults. Indeed, once a fault has been detected, CA shall be able to re-distribute the control authority among the healthy effectors. In that case, the input matrix B is replaced by the matrix B_{fault} . This matrix accounts for the fact that faulted surfaces cannot be used anymore to change the aircraft dynamics, i.e.:

$$\begin{aligned} (B^i)_{fault} &= (B^i)_{nom} \quad i \neq j \\ (B^i)_{fault} &= 0 \quad i = j \end{aligned} \quad (3)$$

where, B^i is the i -th column of the matrix B and j is the index of the faulted actuator.

Proposed Algorithm

In this section, the proposed algorithm for dynamic control allocation is described.

Kalman Filter Approach

As explained earlier, in the proposed CA strategy, the computation of control commands is performed by a Kalman Filter. KF is a well-established method for the optimal estimation of state variables in presence of process and measurement noises (Gelb, 1989) for details). A discrete linear system model is given:

$$\begin{aligned} x(t_{k+1}) &= Fx(t_k) + Gu(t_k) + w(t_k) \\ z(t_k) &= Hx(t_k) + v(t_k) \end{aligned} \quad (4)$$

where, $x \in \mathfrak{R}^h$ is the state vector, $u \in \mathfrak{R}^p$ is the input vector, $z \in \mathfrak{R}^m$ is the output vector, w and v are uncorrelated white Gaussian noises with zero mean and diagonal covariance matrices $Q \in \mathfrak{R}^{h \times h}$ and $R \in \mathfrak{R}^{m \times m}$ respectively.

KF then provides an optimal estimation of the state vector x according to the assigned statistics of process and measurement noises. As a matter of fact, Kalman filter minimizes the difference between the actual and estimated system state. Within the Control Allocation, this KF feature is exploited for estimating the control commands (which are the KF states) minimizing the error with respect to the desired control moments while satisfying the applicable constraints.

More in detail, the state of the Kalman filter will be made of both the commanded and the actual control surface positions, i.e.:

$$x = [\delta_{cmd} \quad \delta_{act}]^T \quad (5)$$

where, the superscript T indicates the transpose operator. Commanded positions are the ones given as reference to the actuation system while the actual positions are the ones that actuation system is able to obtain.

As explained before, the commanded deflections of control surfaces δ_{cmd} do not come from the control laws as the latter only compute the requested moment vector M_{des} . For this reason, they are considered as a part of the Kalman filter state and the following process equation is assigned to them:

$$\delta_{cmd}(t_{k+1}) = \delta_{cmd}(t_k) + w_1(t_k) \quad (6)$$

where, w_1 is the noise related to this process equation.

As it can be seen from the above equation, commanded deflections are modeled as a process with zero derivative. However, the covariance matrix of the related process noise is a design parameter, which allows modifying the control command dynamics to obtain the desired moment vector as it will be explained later.

Instead, process equation for the actuated deflections δ_{act} is straightforward as they clearly depend on the actuator bandwidth. If we assume linear actuator dynamics, the resulting process can be written as:

$$\delta_{act}(t_{k+1}) = A_2\delta_{act}(t_k) + B_2\delta_{cmd}(t_k) + w_2(t_k) \quad (7)$$

where, w_2 is the noise related to this process equation.

Therefore the process model for the design of the CA KF will be the one of equation (4) with:

$$\begin{aligned} x &= [\delta_{cmd} \quad \delta_{act}]^T \\ F &= [I_n \quad A_2] \\ w &= [w_1 \quad w_2]^T \\ Q &= \begin{bmatrix} Q_1 & 0^{m \times m} \\ 0^{m \times m} & Q_2 \end{bmatrix} \end{aligned} \quad (8)$$

where, Q_1 and Q_2 are the covariance matrices (with m rows and m columns) of w_1 and w_2 , I_n is the $n \times n$ identity matrix with and $0^{m \times m}$ is a matrix with m rows and m columns, having all the elements equal to zero.

As mentioned earlier with reference to the system of equation (1), the control surface positions should guarantee the achievement of the requested moments, i.e. $CB\delta_{act} = M_{des}$. Therefore, they can be used as an external (virtual) measurement and included in the output equation of KF:

$$M_{des} = CB\delta_{act} + v \quad (9)$$

where, v is the measurement noise with a covariance matrix equal to R .

Hence, comparing equation (8) with equation (4), we have $z = M_{des}$ and $G = [0^{3 \times m} \ CB]$ where $0^{3 \times m}$ is a matrix with 3 rows (i.e. the 3 moment's components) and m columns having all the elements equal to zero.

Once the model equations have been set, the values of Q_1 , Q_2 and R matrices should be chosen (see equation (4)). It is worth recalling that Q_2 is the process noise related to the actuator dynamics, which are supposed to be known with a sufficient degree of accuracy. Furthermore, the limited actuators bandwidth represents a constraint to be satisfied by the control allocation algorithm. Hence the values of Q_2 matrix are expected to be chosen significantly smaller than the ones of Q_1 and R . In this way, the solution found by KF will be compatible with the limited bandwidth of actuators. On the other hand, Q_1 and R matrices can be seen as design parameters to be tuned, since control effort and accuracy in achieving the requested moments can be traded-off by selecting the covariance related to the actuator commands (equation (6)) and moments equation (9). Indeed, small values of R w.r.t. the ones of Q_1 give much more importance to the output equation (9), i.e. to the fact that obtained moments $CB\delta_{act}$ should be equal to the requested ones M_{des} . On the contrary, small values of Q_1 (w.r.t. the ones of R) give much more importance to the equation of commanded surface positions (equation 6), i.e. to the fact that control effort should be as small as possible (commanded positions are indeed constant according to equation 6).

Constraint Management

As it is clear from the previous section, dynamic constraints (i.e. limited bandwidth of actuators) are implicitly taken into account by adding the actuators dynamics in the process model and assigning a "small" covariance to the noise process. However, actuators (and thus the control surfaces) are subject to specific position and rate limits that become inequality constraints for the control allocation problem, i.e.:

$$\delta_{min} \leq \delta_{act}(t_k) \leq \delta_{max} \quad (10)$$

$$\delta(t_{k-1}) + \delta_{rate_min} T_c \leq \delta_{act}(t_k) \leq \delta(t_{k-1}) + \delta_{rate_max} T_c$$

where, T_c is the sample time.

In the proposed strategy, the inequality constraints are taken into account through the on-line update of equation (7). Indeed, for the current solution of CA problem $\delta_{act}(t_k)$, the constraints of equation (10) are checked. This equation can also be recast as:

$$\delta_{min_sat} \leq \delta_{act}(t_k) \leq \delta_{max_sat} \quad (11)$$

where, $\delta_{min_sat} = \max\{\delta_{min}, \delta_{act}(t_{k-1}) + \delta_{rate_min} T_c\}$ and $\delta_{max_sat} = \min\{\delta_{max}, \delta_{act}(t_{k-1}) + \delta_{rate_max} T_c\}$.

If the i -th component of $\delta_{act}(t_k)$ is outside the limits defined by equation (11), i.e. if $\delta_{act}^i > \delta_{max_sat}^i$ or $\delta_{act}^i < \delta_{min_sat}^i$ for some integer i in the range $[1, 2, \dots, m]$, the equation (7) is changed to force the i -th component to the constraint limit value. Once that the i -th component of δ_{act} has been fixed, the remaining components, i.e. the remaining control surfaces can be anyway used to achieve the requested moments. However, we have to distinguish between two different situations:

1. Constraint reached due to the position limits
2. Constraint reached due to the rate limits

In the first case, the i -th component of equation (7) is re-arranged as follows:

$$\delta_{act}^i(t_{k+1}) = A_2^{i,i} \delta_{act}^i(t_k) + B_2^{i,i} \delta_{lim}^i + w_2^i(t_k) \quad (12)$$

where, $A_2^{i,i}$ is the element in the i -th-row and i -th column of the matrix A_2 , $B_2^{i,i}$ is the element in the i -th-row and i -th column of the matrix B_2 , δ_{lim}^i is equal to $\delta_{min_sat}^i$ or $\delta_{max_sat}^i$ depending on whether the lower or upper position limit has been reached. In this way, the i -th component of the surface position vector δ_{act}^i will follow the value δ_{lim}^i instead of the commanded value δ_{cmd}^i .

In the second case (rate limits reached), the i -th component of equation (7) is set as:

$$\delta_{act}^i(t_{k+1}) = \delta_{act}^i(t_k) + T_c \delta_{rate_lim}^i + w_2^i(t_k) \quad (13)$$

where, $\delta_{rate_lim}^i$ is equal to $\delta_{rate_min}^i$ or $\delta_{rate_max}^i$ depending on whether the lower or upper rate limit is reached. In this way, the i -th component of the surface position vector δ_{act}^i will have a constant derivative equal to $\delta_{rate_lim}^i$.

It is worth recalling that the equations reproducing the actuator dynamics (i.e. equation 7) has a process noise with a very small covariance Q_2 (with respect to both Q_1 and R). Clearly, this is true also for Equation (12) and (13). In this way, control surface position will be "forced" to the constraints, which will then become active equality constraints. In constrained optimization problems, equality constraints are closely related to Lagrange multipliers, which give a measure of how much the cost function changes when the constraint is relaxed (see Harkegard, 2002). More in detail, negative Lagrange multipliers imply that relaxing the constraints increases the cost function. For this reason, before executing the steps of the Equation 12 and 13, the CA logic checks the sign of Lagrange multipliers. Indeed, in case the Lagrange multiplier is negative, the equality

constraint is relaxed (i.e. it is no more an active constraint) and all the n control surfaces can be used again to achieve the target moment vector M_{des} .

Lagrange multipliers can be computed as follows:

$$\lambda = C_{constr}(CB)^T(M_{des} - CB\delta_{act}) \quad (14)$$

where, the superscript T indicates the transpose operator and C_{constr} is the matrix describing the inequality constraints of equation (11), re-casted as follows:

$$C_{constr}\delta_{act} \geq \delta_{sat} \quad (15)$$

with:

$$C_{constr} = \begin{bmatrix} I^{n \times n} \\ -I^{n \times n} \end{bmatrix} \quad (16)$$

$$\delta_{sat} = \begin{bmatrix} \delta_{min_sat} \\ -\delta_{min_sat} \end{bmatrix}$$

A similar approach is used by Active Set method proposed in Harkegard (2002) to account for the presence of inequality constraints. Indeed, in the Active Set Method, saturated inputs are used as active (equality) constraints and the problem of control allocation is solved in an iterative way. However, Kalman filter solves it dynamically, giving the advantage of no convergence issues at each time step; on the other hand, the achievement of the requested moment vector is not instantaneous but it follows Kalman filter dynamics.

Numerical Analysis

In order to show the effectiveness of the proposed CA algorithm, some numerical analyses have been carried out using a 6DOF model of the flight test bed High Altitude Performance Demonstrator (HAPD), an unmanned vehicle of the High Altitude Long Endurance (HALE) category designed by Italian Aerospace Research Centre (Imperatore and Vecchione, 2009).

For what concerns the vehicle control surfaces, the HAPD vehicle has a set of redundant control surfaces that are listed below (see Fig. 1):

- Three elevators pairs (Inboard, Midboard and Outboard), with $\pm 25^\circ$ saturations;
- Two ailerons pairs (Inboard and Outboard ailerons), with $\pm 25^\circ$ saturations;
- Two rudders (Upper and Lower), with $\pm 25^\circ$ saturations.

Moreover, it is equipped with eight independent engines, commanded by the auto-throttle subsystem (not managed by the Control Allocation algorithm).

Simulation Model

The simulation model is made of:

- HAPD 6 DOF model (including external environment, actuators and sensors, see Baraniello *et al.*, 2011; Cicala *et al.*, 2011)
- Flight Control Laws
- Control Allocation subsystem

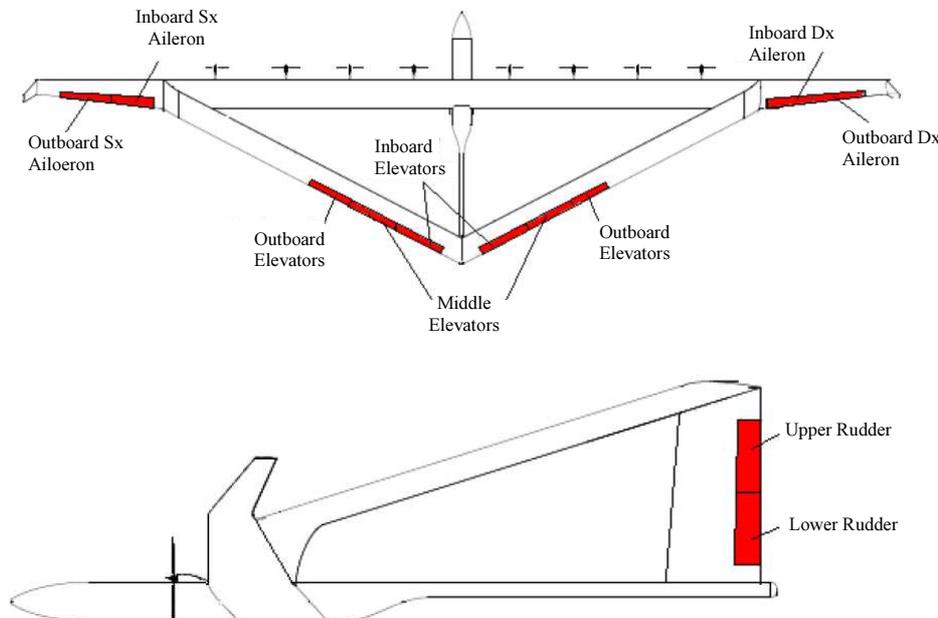


Fig. 1: HAPD control surfaces

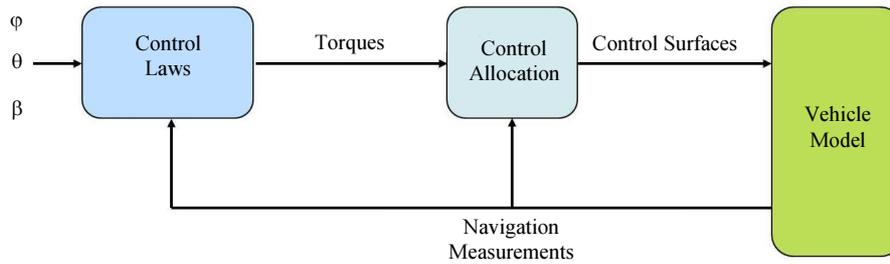


Fig. 2: HAPD control laws architecture

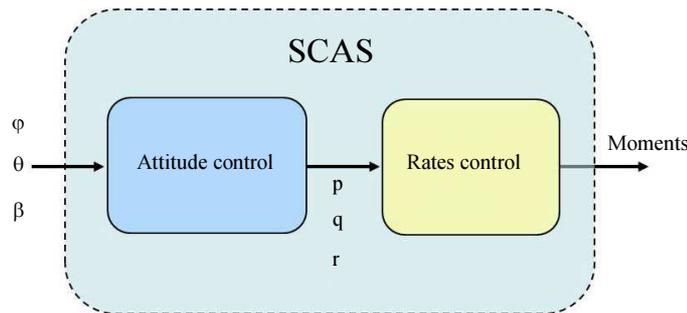


Fig. 3: Stability and Control Augmentation System (SCAS) architecture

Actuators dynamics have been modeled as a first order linear system with a bandwidth equal to 5 Hz. Rate limits are equal to 60 deg/s.

As we can see in the Fig. 2, Control Laws compute the moments required to guarantee the desired attitude dynamics.

Control Laws module (Fig. 3) is composed by two macro-elements, the inner loop to control the angular rate components p , q and r and the outer loop to control the angles φ , θ , β . Both the modules are based on Adaptive Model Following Control strategy (Sollazzo *et al.*, 2009; Poderico *et al.*, 2014 for further details).

The reference commands for the Control Laws are the roll angle φ the pitch angle θ and the sideslip angle β . With reference to the inner loop, that controls the angular rates, the following variables are defined:

$$\begin{aligned} x &= [u \ v \ w \ p \ q \ r] \\ y &= [p \ q \ r] \\ u &= [v_x \ v_y \ v_z] \\ r_{ref} &= [p_{ref} \ q_{ref} \ r_{ref}] \end{aligned} \tag{17}$$

where, u , v , w are the components of the true air speed in body frame, p , q , r are the roll, pitch and yaw rate respectively and v_x , v_y , v_z are the requested moments along the three axes of the body frame. It is worth noting that, considering the system reported in Equation 17, the moments have the dimension of angular accelerations (rad/s^2). Finally, the subscript ref indicates the reference value.

The generation of control surfaces commands is a task of the Control Allocation module. It receives the requested moments v_x , v_y , v_z from the Control Laws and provides the control commands $\delta_1, \delta_2, \dots, \delta_{12}$, i.e. six elevators (three pairs, Inboard, Midboard and Outboard respectively), four ailerons (two pairs Inboard and Outboard) and two rudders (Upper and Lower).

Simulation Scenarios

The numerical analyses concern both nominal and fault scenarios where the fault of one or more control actuators is simulated. It is worth specifying that, in fault scenarios, it is assumed that the information about failed actuators are provided by a FDI system that is anyway beyond the scope of this paper. Hence, at each time instant, CA is assumed to know which are the healthy surfaces.

A first simulation has been carried out by considering a nominal situation (i.e. no fault). Initial conditions have been set as follows:

$$\begin{aligned} h_0 &= 500m \\ V_0 &= 25m/s \\ p_0 &= 0 \text{ deg/s} \\ q_0 &= 0 \text{ deg/s} \\ r_0 &= 0 \text{ deg/s} \\ \varphi_0 &= 0 \text{ deg} \\ \theta_0 &= -2 \text{ deg} \\ \beta_0 &= 0 \text{ deg} \end{aligned} \tag{18}$$

Step commands for roll and pitch angles have been given to Stability and Control Augmentation System (SCAS), while reference for sideslip angle is zero.

In Fig. 5, a comparison between the moments requested by the control laws (labeled as “ref”) and the ones commanded by the control allocation logic (labeled as “actual”) is reported. As it can be seen from the figure, the difference between reference and actual moments is quite negligible. This ensures the tracking of the reference commands as shown in Fig. 4.

In order to show the effectiveness of the proposed CA algorithm in presence of an actuator fault, the fault of outboard ailerons and inboard elevators during the same maneuvers has been simulated. As it is shown in the Fig. 8 and Fig. 9, where the control commands are reported, the outboard ailerons and inboard elevators are no more used by the CA algorithm after the fault event. This is

possible since CA uses the remaining control surfaces to obtain the requested moments (Fig. 7). This allows the tracking of the referenc commands, as shown in Fig. 6.

Finally, the effectiveness of the proposed CA algorithm in managing the actuators position limits is shown. To this end, a demanding roll maneuver is executed (i.e. a step of 45 deg).

One of the two rudder commands reaches the saturation value (25 degrees), therefore the rudder command is not used for several seconds, as shown in Fig. 11. However, the control effort is effectively re-distributed among the remaining control surfaces thus ensuring that the requested moments are anyway achieved (Fig. 10). As explained earlier, the command is kept at its saturation value (at about 12 sec) because the Lagrange multipliers are non-negative (Fig. 12).

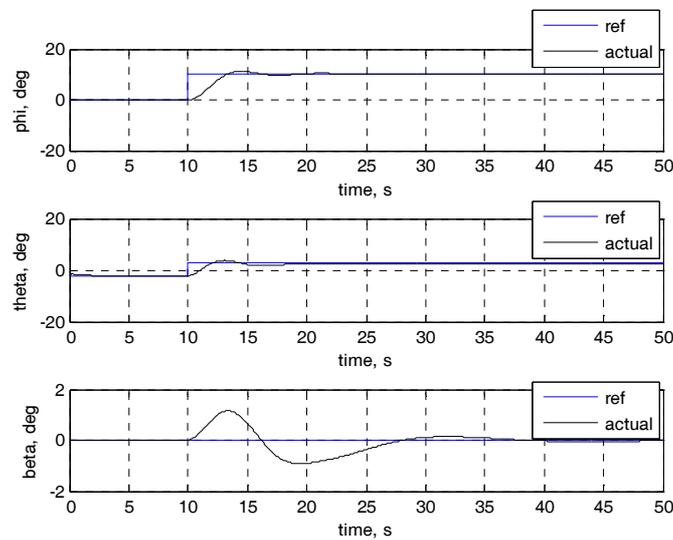


Fig. 4: Reference vs actual outputs (nominal scenario)

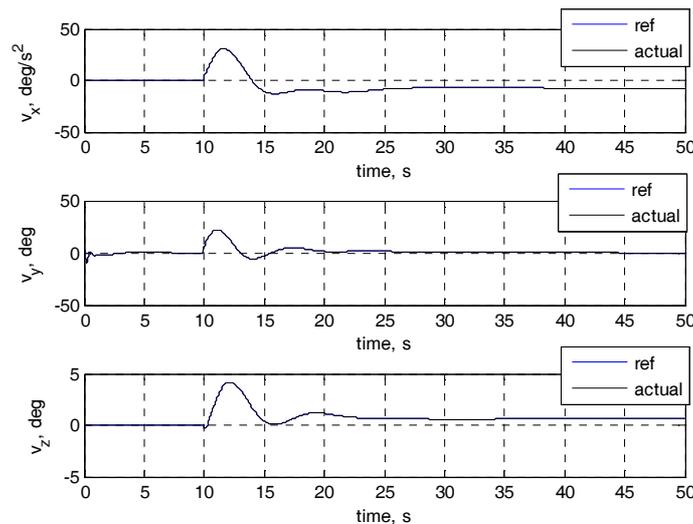


Fig. 5: Requested vs Actual Moments (nominal scenario)

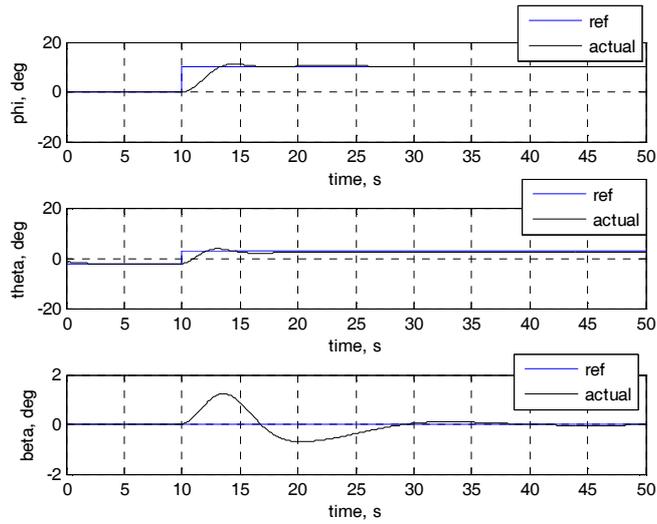


Fig. 6: Reference vs actual outputs (Outboard Ailerons and Inboard Elevators Locked at 11.5 sec)

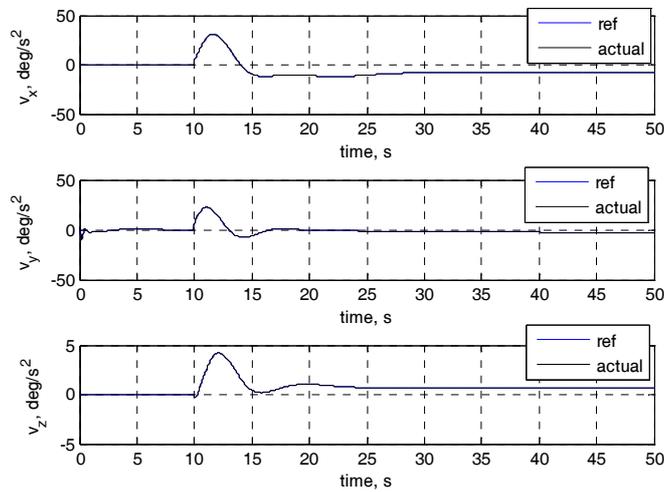


Fig. 7: Reference vs Actual moments (Outboard Ailerons and Inboard Elevators Locked at 11.5 sec)

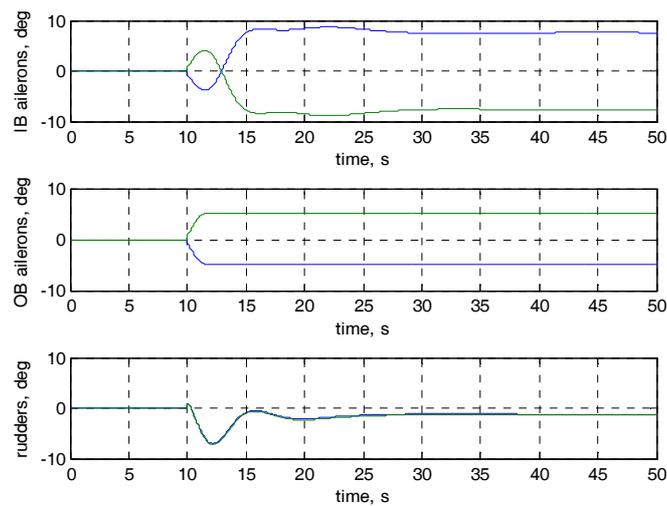


Fig. 8: Latero-Directional Control commands (Outboard Ailerons and Inboard Elevator Locked at 11.5 sec)

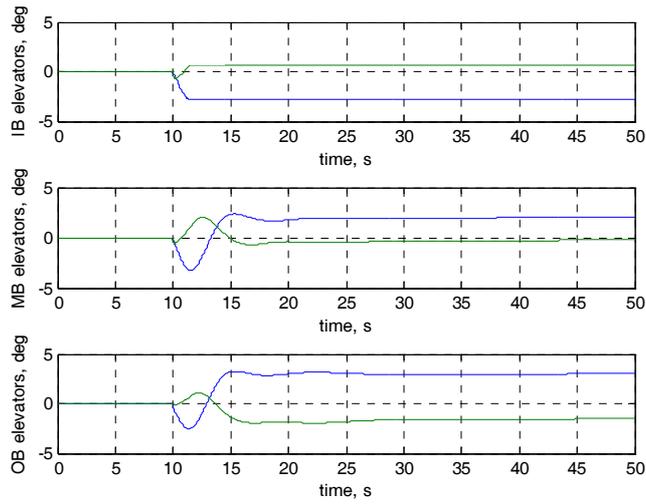


Fig. 9. Longitudinal Control commands (Outboard Ailerons and Inboard Elevator Locked at 11.5 sec)

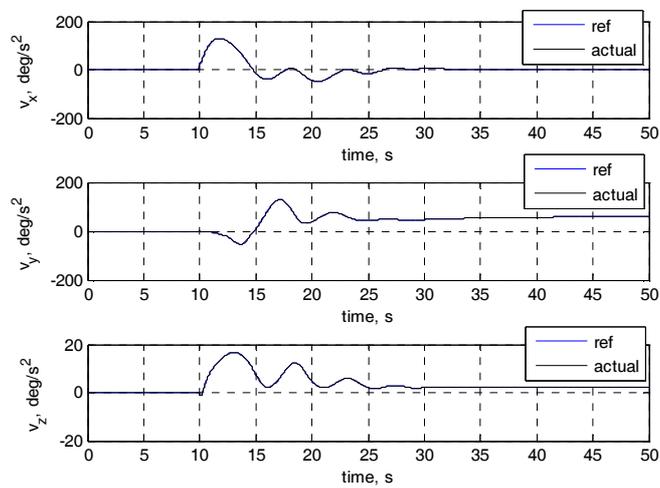


Fig. 10: Reference vs Actual moments (aggressive roll maneuver)

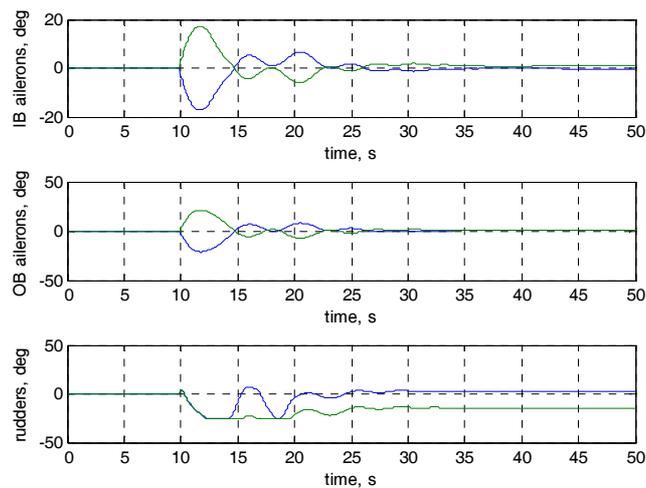


Fig. 11: Latero-Directional Control commands (aggressive roll maneuver)

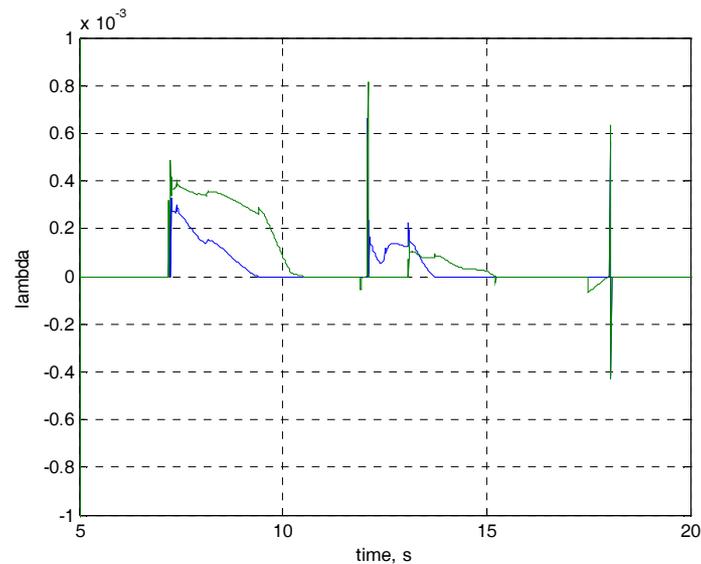


Fig. 12: Lagrange Multipliers (aggressive roll maneuver)

Conclusion

In this study, a new algorithm for Control Allocation has been proposed. It is based on a Kalman filter approach and presents some advantages with respect to the existing methods.

Indeed, the algorithm allows solving the control allocation problem by taking into account not only position and rate limits of the actuators but also the actuator dynamics thus ensuring that the computed solution is feasible, i.e. it can be followed by the actuation system. Furthermore, this approach does not suffer from the well-known problems of convergence and stability of numerical methods based on iterative methods.

The effectiveness of the proposed Control Allocation strategy has been shown in both nominal and fault scenarios. To this end, a numerical analysis has been carried out by using 6DOF vehicle model of HAPD vehicle (a Flying Test Bed developed by Italian Aerospace Research Centre). Simulations have shown good capabilities of ensuring robust performance and stability also in presence of a fault affecting the control surfaces.

Author's Contributions

G. Morani and F. Nebula: Have been involved in the simulation set-up, in the execution of the numerical verifications.

All the authors contributed to the methodological aspects and to the definition of the Control Allocation algorithms, as well as to the writing of the manuscript.

Ethics

No ethical issues may arise after the publication of this manuscript.

References

- Anthony, P. and S. Marc, 2000. A closed-loop comparison of control allocation, Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit, Aug. 14-17, Dever, CO, USA. DOI: 10.2514/6.2000-4538
- Baraniello, V.R., M. Cicala and L. Cicala, 2011. An algorithm for real time estimation of the flexible UAV structural motions using a video-based system. Proceedings of the 14th International Conference on Information Fusion, Jul. 5-8, IEEE Xplore press, Chicago, Illinois, USA.
- Bodson, M., 2002. Evaluation of optimization methods for control allocation. *J. Guidance, Control Dynamics*, 25: 703-711. DOI: 10.2514/2.4937
- Cicala, M., V.R. Baraniello and A. Sollazzo, 2011. A Load Alleviation Application of an Elastic Motion Estimator for a Flexible UAV. Proceedings of the 19th Mediterranean Conference on Control & Automation (MED), Jun. 20-23, IEEE Xplore press, Corfu, Greece. DOI: 10.1109/MED.2011.5983114
- Durham, W.C. and K.A. Bordignon, 1996. Multiple control effector rate limiting. *J. Guidance Control Dynamics*, 19: 30-37. DOI: 10.2514/3.21576
- Durham, W.C., 1994. Attainable moments for the constrained control allocation problem. *J. Guidance Control Dynamics*, 17: 1371-1373. DOI: 10.2514/3.21360
- Gelb, A., 1989. Applied Optimal Estimation. The M.I.T. Press.
- Harkegard, O. and S. Glad Torkel, 2005. Resolving actuator redundancy-optimal control vs. control allocation. *Automatica*, 41: 137-144. DOI: 10.1016/j.automatica.2004.09.007

- Harkegard, O., 2002. Efficient active set algorithms for solving constrained least squares problems in aircraft control allocation. Proceedings of the Proceedings of the 41st IEEE Conference on Decision and Control, Dec. 10-13, IEEE Xplore press, Las Vegas, Nevada, USA.
DOI: 10.1109/CDC.2002.1184694
- Imperatore, B. and L. Vecchione, 2009. A flexible wing unmanned aerial research system. SAE Technical Paper 2009-01-3127.
- Kenneth, A.B. and W.C. Durham, 1995. Closed-form solutions to constrained control allocation problem. J. Guidance Control Dynamics, 18: 1000-1007.
DOI: 10.2514/3.21497
- Luo, Y., A. Serrani, S. Yurkovich, M.W. Oppenheimer and D.B. Doman, 2007. Model-predictive dynamic control allocation scheme for reentry vehicles. J. Guidance Control Dynamics, 30: 100-113.
DOI: 10.2514/1.25473
- Petersen, J.A.M. and M. Bodson, 2006. Constrained quadratic programming techniques for control allocation. IEEE Transactions Control Systems Technol., 14: 91-98.
DOI: 10.1109/TCST.2005.860516
- Poderico, M., G. Morani and F. Corrado, 2014. Fault detection isolation and reconfiguration algorithms for atmospheric reentry. Proceedings of the 22nd Mediterranean Conference on Control and Automation, Jun. 2014.
- Sollazzo, A., G. Morani and A. Giovannini, 2009. An adaptive fault-tolerant FCS for a large transport aircraft. Springer-Verlag.
- Virnig, J. and D. Bodden, 1994. Multivariable control allocation and control law conditioning when control effectors limit. Guidance, Navigation and Control Conference, Scottsdale, AZ, USA.
DOI: 10.2514/6.1994-3609.