Optimal reconfigurable allocation of a multi-engine cluster for a reusable launch vehicle

Renato MURATA[⋆] [⋆⋆], *Louis THIOULOUSE*[⋆], *Julien MARZAT*[⋆] [†], *Hélène PIET-LAHANIER*[⋆],
Marco GALEOTTA[⋆⋆], François FARAGO[⋆⋆]
[⋆]*DTIS, ONERA, Université Paris-Saclay, 91123 Palaiseau, France*
renato.murata@onera.fr, louis.thioulouse@onera.fr, julien.marzat@onera.fr, helene.piet-lahanier@onera.fr
[⋆⋆]*CNES, Sous-Direction Techniques Systèmes de Transport Spatial, Paris, Île-de-France, 75612, France*
marco.galeotta@cnes.fr, francois.farago@cnes.fr
[†]Corresponding author

Abstract

This paper presents a control allocation algorithm that can be used to control a multi-engine cluster of a reusable launch vehicle. The allocation module is reconfigurable and can take into account three types of actuator faults (LoE, LIP, and float). A set of equations represents the relation between the force and position of one engine and the force and torque produced by the same engine at the launcher's center of mass. This relation in association with the actuator constraints has been used to build a nonlinear constrained optimization problem, on which an interior-point algorithm has been applied to find a solution. Simulations were performed under different scenarios using cluster configurations with three, seven, and nine engines. The results suggest that the allocation procedure is able to find relevant solutions in a large variety of nominal and faulty scenarios.

1. Introduction

The launcher market is highly competitive, and complete or partial recovery of launchers appears to be a promising way to reduce operating costs. The development of a Reusable Launch Vehicle (RLV) using a toss-back boost to return to the launch site implies new needs for thrust modulation and control, closed-loop dynamics, and engine monitoring. In order to address these new issues, multi-engine propulsion clusters are currently designed in Europe for the next generation of launchers. The multi-engine characteristics of the propulsion bay should offer more reliability and availability compared to a single-engine architecture [1].

The system considered consists of a cluster of multiple liquid-propellant rocket engines (LPRE). Each engine is modelled with a 2-DOF thrust vector control (TVC) and thrust-throttling capability. The problem addressed is the coordination of the individual engine set-points in terms of thrust module and gimbal angles so as to produce the overall force and torque references requested by trajectory management at the launcher level. This can be formulated as a constrained control allocation problem for an over-actuated mechanical system (see [2] for a review). Additionally, to meet fault tolerance requirements and improve the reliability of the propulsion bay, this control allocation module should take into account faults such as the complete loss of an engine, a decrease in thrust efficiency, or stuck gimballing axes. These faults can be identified by a dedicated Health Monitoring System (HMS), which is out of the scope of this work.

The use of control allocation algorithms to manage over-actuated systems is well documented in the literature. Since aircraft are typically over-actuated systems, with more control surfaces than variables to be controlled, control allocation algorithms were applied to aircraft models in [3, 4]. A survey dedicated to control allocation techniques to minimize the control effort for ships and underwater vehicles has also been proposed in [5]. In [6], a dynamic control allocation law was tested for an automotive vehicle yaw stabilization. Optimization-based control allocation is indeed not the only strategy to find a solution to such actuator redundancy problems, which is why it has been compared in [7] with an optimal control formulation for over-actuated systems. The use of control allocation algorithms to obtain fault-tolerance for linear systems subject to loss of effectiveness of actuators has also been investigated in [8] with application to a marine vessel. An advantage of using control allocation algorithms for fault-tolerance is that the controller structure does not have to be reconfigured in case of faults [9]. Another fault-tolerant control allocation strategy has been developed in [10] where unknown input observers (UIO) were used for fault diagnosis. UIOs have also been used for the detection and isolation of satellite thruster faults in [11] where the reconfiguration of the system is

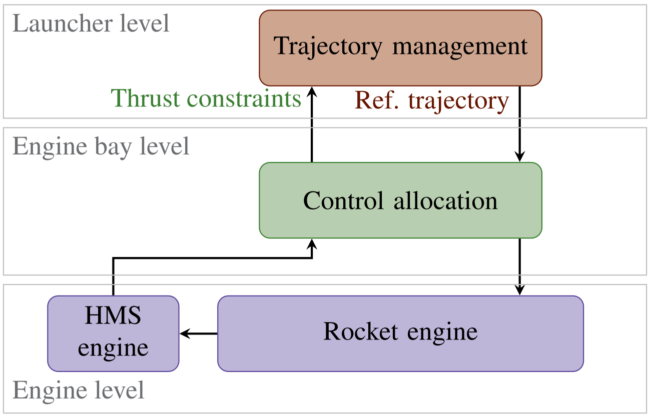

Figure 1: Possible functional architecture

then addressed by modifying the reference input to pre-compensate the fault and a nonlinear iterative control allocation procedure then computes the reference thrust for each actuator, following a system architecture similar to the one we consider.

The objective of the present paper is to formalize an optimization problem under constraints (minimum and maximum thrust levels, maximum gimbal angle) and select an appropriate optimization algorithm to design a generic reconfigurable allocation module for a cluster of multiple LPREs, and validate the behavior of this module with different cluster configurations on various scenarios. A description of the multi-engine propulsion cluster model is first given in Section 2, then the proposed optimization-based control allocation method is presented in Section 3. Section 4 reports the simulation results of the allocation module for various nominal and faulty scenarios. Finally, Section 5 summarizes the conclusions and perspectives of this work.

2. System description

A possible functional architecture of the multi-engine cluster is illustrated in Figure 1. The control allocation module receives its reference from the trajectory management module. Using the information on the health state of the engines coming from the engine-level HMS, the control allocation should meet the trajectory management requirements given the current engine cluster capabilities. The main objective is to minimize the error between the trajectory management reference at the launcher level and the engine cluster output at the engine level. An additional requirement for the allocation module is also to minimize the control effort, particularly the TVC action.

Figure 2 shows the coordinate system [12] used to describe the propulsive bay. The launcher reference frame is located at its center of mass at point O_B where $O_B \in \mathbb{R}^3$. Each engine $i \in \{1, \ldots, n\}$ from the cluster has its own reference frame and can produce a thrust of magnitude T_i, and $T = [T_1, \ldots, T_n]^T$. The engines can rotate $\delta_{q,i}$ degrees around the Y_i-axis in the X_iZ_i-plane and δ_r degrees about the Z_i-axis in the X_iY_i-plane. The 3D distance of the i-th engine from the launcher's center of mass O_B is denoted by α_i.

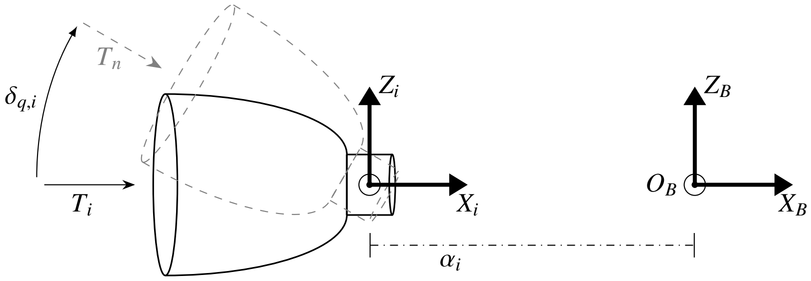

Figure 2: Nozzle and launcher coordinate system

The force T_i generated by the i-th engine can be expressed in the launcher's body frame as follows:

$$T_{B,i} = \begin{bmatrix} \cos\delta_{r,i}\cos\delta_{q,i} \\ \sin\delta_{r,i}\cos\delta_{q,i} \\ -\sin\delta_{q,i} \end{bmatrix} T_i \tag{1}$$

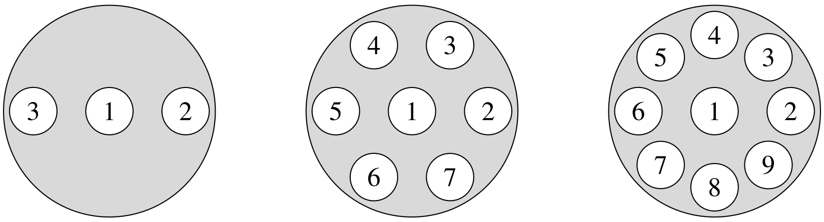

Figure 3: Cluster configurations with 3, 7, and 9 engines

The torque $M_{B,i}$ generated by the force T_i about the launcher's center of mass is calculated with the cross-product between the engine's position and the force vector $T_{B,i}$ as shown in equation (2).

$$M_{B,i} = \alpha_i \times T_{B,i} \tag{2}$$

Given a propulsion cluster with n engines, the standard configuration has one central engine longitudinally aligned with the launcher's center of mass. The other engines are evenly distributed around the main engine. This configuration allows the control of the roll angle without additional actuators. This paper analyzes three cluster configurations with respectively three, seven and nine engines (see Figure 3). In all configurations, each engine i can vary its thrust between $T_i \in [300, 1000]$ kN and the minimum and maximum values of the gimbal angles are restricted to $\delta_{r,i}, \delta_{q,i} \in [-5, 5]°$, with an extra constraint that forbids the maximum deflection (negative or positive) of both angles at the same time, i.e., $\sqrt{\delta_{r,i}^2 + \delta_{q,i}^2} \leq 5°$.

The control allocation module has to coordinate the engine cluster to produce the overall reference force T_{re} and torque M_{re} in the body frame of the launcher. Using equations (1) and (2), the contribution of the n engines can be combined to calculate the resultant force T_B and torque M_B at point O_B, as shown in Eq. (3)

$$\begin{aligned}
T_{Bx} &= \sum_{i=1}^{n} T_i \cos\delta_{r,i} \cos\delta_{q,i} & M_{Bx} &= \sum_{i=1}^{n} T_i(-\alpha_{y,i} \sin\delta_{q,i} - \alpha_{z,i} \sin\delta_{r,i} \cos\delta_{q,i}) \\
T_{By} &= \sum_{i=1}^{n} T_i \sin\delta_{r,i} \cos\delta_{q,i} & M_{By} &= \sum_{i=1}^{n} T_i(\alpha_{z,i} \cos\delta_{r,i} \cos\delta_{q,i} + \alpha_{z,i} \sin\delta_{q,i}) \\
T_{Bz} &= \sum_{i=1}^{n} -T_i \sin\delta_{q,i} & M_{Bz} &= \sum_{i=1}^{n} T_i(\alpha_{x,i} \sin\delta_{r,i} \cos\delta_{q,i} - \alpha_{y,i} \cos\delta_{r,i} \cos\delta_{q,i})
\end{aligned} \tag{3}$$

where $T_B = [T_{Bx}, T_{By}, T_{Bz}]^T$, $M_B = [M_{Bx}, M_{By}, M_{Bz}]^T$, and $\alpha_i = [\alpha_{x,i}, \alpha_{y,i}, \alpha_{z,i}]^T$.

With a force reference T_{re} and a torque reference M_{re} given by the trajectory management module, the control allocation must find a solution that satisfies equation (3) such that $T_{re} = T_B$ and $M_{re} = M_B$. We can define the error e_T as the difference between the force reference and the force output, $e_T = T_B - T_{re}$ and the error e_M as the difference between the torque reference and the torque output, $e_M = M_B - M_{re}$. The control allocation constraints for the given reference can be formalized as:

$$\begin{aligned}
e_{T,x} &= T_{Bx} - T_{re,x} & e_{M,x} &= M_{Bx} - M_{re,x} \\
e_{T,y} &= T_{By} - T_{re,y} & e_{M,y} &= M_{By} - M_{re,y} \\
e_{T,z} &= T_{Bz} - T_{re,z} & e_{M,z} &= M_{Bz} - M_{re,z}
\end{aligned} \tag{4}$$

where $T_{re} = [T_{re,x}, T_{re,y}, T_{re,z}]^T$, $M_{re} = [M_{re,x}, M_{re,y}, M_{re,z}]^T$, $e_T = [e_{T,x}, e_{T,y}, e_{T,z}]^T$ and $e_M = [e_{M,x}, e_{M,y}, e_{M,z}]^T$.

The force and torque generated by the engines and their impact on the launcher's center of mass are nonlinear due to the gimbal angles δ_r and δ_q and the coupling between variables. It is possible to simplify the relations presented in (3) under the small-angle approximation of the sine and cosine functions. Using this linearization method, the equations defined in (3) can be rewritten as defined in (5). However, this relation remains nonlinear due to the coupling between the thrust T_B and the gimbal angles δ_r and δ_q.

$$\begin{aligned}
T'_{Bx} &= \sum_{i=1}^{n} T_i & M'_{Bx} &= \sum_{i=1}^{n} T_i(-\alpha_{y,i}\delta_{q,i} - \alpha_{z,i}\delta_{r,i}) \\
T'_{By} &= \sum_{i=1}^{n} T_i\delta_{r,i} & M'_{By} &= \sum_{i=1}^{n} T_i(\alpha_{z,i} + \alpha_{z,i}\delta_{q,i}) \\
T'_{Bz} &= \sum_{i=1}^{n} -T_i\delta_{q,i} & M'_{Bz} &= \sum_{i=1}^{n} T_i(\alpha_{x,i}\delta_{r,i} - \alpha_{y,i})
\end{aligned} \tag{5}$$

If we consider that those angles are fixed, i.e. the engine cluster has no TVC action, then the coupling between variables disappears and the relations become linear. In this case, the angles $\delta_{r,i}$ and $\delta_{q,i}$ defined in equation (1) will be constant, and we can calculate a control effectiveness matrix B as

$$B = \begin{bmatrix} T'_{B,1} & \dots & T'_{B,n} \\ M'_{B,1} & \dots & M'_{B,n} \end{bmatrix}. \tag{6}$$

Note that, for some specific scenarios with no TVC action or all engines running at their nominal behavior, the need for optimal control allocation algorithms may be questionable. In those cases, it is possible to compute predefined solutions for all the possible forces and torques without relying on optimization algorithms. In this work, the focus is put on a more generic approach which can compute online a feasible solution for a large variety of cluster configurations and operating scenarios with healthy and faulty engines.

Three types of actuator failures are taken into account: Loss of Effectiveness (LoE), Float fault, and Lock-In-Place (LIP). A formal description of these faults can be found in the reference paper [13]. It is considered in the present work that all the faults are instantly detected and perfectly estimated by the HMS at the engine level.

- **Loss of effectiveness:** this fault affects only the thrust capability of the engine. It is less critical for the engine cluster because it does not decrease the overall number of actuators. If this fault occurs on the k-th engine, new constraints (7) will restrict the thrust T_k as:

$$T_k = [300, T_{k,f}] \quad \text{where} \quad T_{k,f} < 1000\,\text{kN}. \tag{7}$$

- **Float fault:** it can be seen as a total loss of one engine. If this fault occurs on the k-th engine, the thrust T_k is reduced to zero, and the gimbal angles $\delta_{r,k}$ and $\delta_{q,k}$ will not influence the engine cluster anymore. To take this fault into account, the relations described in (3) can be rewritten by removing the term corresponding to the faulty engine to obtain a cluster with $n-1$ active engines.

- **Lock-in-place:** the engine orientation is blocked in one position, and loses its TVC action. If the fault occurs on the k-th engine, the TVC angles $\delta_{r,k}$ and $\delta_{q,k}$ become constant. However, the thrust T_k will have an influence on the launcher, so the number of available engines is still n and the equations (3) will comprise new constant gimbal angle constraints (8) as

$$\delta_{r,k} = K_1 \quad \delta_{q,k} = K_2 \tag{8}$$

These faults will have an impact on the launcher, but from the control allocation point of view, the mapping (3) between the resulting forces acting in the engine cluster and the launcher center of mass will remain the same. The control allocation module will have to deal with more constraints for the loss of effectiveness and lock-in-place faults. If the float fault occurs, the control allocation module will lose one degree of freedom (equivalent to the loss of one engine) to resolve the allocation problem. One of the main advantages of the proposed optimization-based control allocation strategy for fault-tolerant systems is that the fault can be treated by adding more restrictions to the control allocation problem. The system reconfiguration is achieved by changing the individual references of the actuators, as previously considered in [8, 9, 11].

3. Optimal allocation method

The objective of the allocation method is to find the force T_i and gimbal angles $\delta_{r,i}$ and $\delta_{q,i}$, $\forall i \in [1, n]$ satisfying equation (3) such that the errors e_T and e_M defined in equation (4) are zero. This can be formulated as the search for a solution to a nonlinear constrained optimization problem. There are different ways to build the optimization problem. If the references T_{re} and M_{re} are always feasible, we can impose a constraint on the optimization problem as $e_T = 0$ and $e_M = 0$. However this is not possible if the references are unreachable, otherwise the optimization problem will not have any solution. In this case, one strategy is to minimize the errors e_T and e_M. The errors can be calculated using the complete version of the force and torque equations defined in Eq. (3) or the small-angle approximation of the sine and cosine version defined in Eq. (5). Other secondary criteria can also be incorporated, such as minimizing the magnitude of the TVC action δ_r and δ_q or the difference between the minimum and maximum thrust T_i among all engines. In this work, the optimization problem will be labelled as *explicit* if the references are feasible, otherwise it will be labelled as *implicit*.

3.1 Linearly constrained optimization problem

Under the assumptions of small-angles and constant TVC orientations, the control effectiveness matrix defined in Eq. (6) makes it possible to define an optimization problem with linear constraints, as defined in Table 1. In this case, the only input variable considered is the thrust T produced by each engine, while the gimbal angles are considered fixed. We want to minimize the difference between the thrust of each engine.

Table 1: Optimization problem with linear constraints

Variables	$x = \{T\}$
Objective function	$\min J_b(x) = \max(x) - \min(x)$
Constraints	$\begin{bmatrix} T_{re} \\ M_{re} \end{bmatrix} = Bx$ $T_{min} \leq T' \leq T_{max}$

3.2 Explicit constrained optimization problem

In the nonlinear case with gimbal angles active, three input variables are defined for each engine of the cluster, namely T, δ_r and δ_q, the dimension of the input vector is therefore equal to $3n$. If the references are feasible, the *explicit* optimization problem can be defined as presented in Table 2. The objective function $J_b(x)$ is defined to minimize the use of the TVC action, under equality constraints related to the fulfillment of the reference and inequalities related to actuator saturation.

Table 2: Explicit constrained optimization problem

Variables	$x = \{T, \delta_r, \delta_q\}$
Objective function	$\min J_b(x) = w_1 \|\|[\delta_r; \delta_q]\|\|_2$
Constraints	$e_T(x) = 0$ and $e_M(x) = 0$ $\sqrt{\delta_r^2 + \delta_q^2} \leq \delta_{max}$ $T_{min} \leq T \leq T_{max}$ $\delta_{r,min} \leq \delta_r \leq \delta_{r,max}$ $\delta_{q,min} \leq \delta_q \leq \delta_{q,max}$

3.3 Implicit constrained optimization problem

When the reference sent to the control allocation module is not feasible, the constraint $e_T(x) = e_M(x) = 0$ cannot be respected, therefore the previous optimization problem will not yield a solution. In the *implicit* constrained problem defined in Table 3, the errors e_T and e_M are instead incorporated into the objective function to search for an approximate feasible solution. The weight λ can be used to prioritize the minimization of either the thrust or the torque, while the weights w_1 and w_2 can be used to give more importance to a smaller solicitation of the TVC action or to get closer to the requested reference.

3.4 Optimization algorithms

Different algorithms can be used to search for solutions to the above optimization problems. A popular algorithm to solve linear allocation problems is the Simplex. According to [14], this algorithm guarantees finite-time convergence

Table 3: Implicit constrained optimization problem

Variables	$x = \{T, \delta_r, \delta_q\}$
Objective function	$\min J_{em}(x) = w_1\|[\delta_r; \delta_q]\|_2 + w_2(\lambda\|e_T\|_2 + (1-\lambda)\|e_M\|_2)$
Constraints	$\sqrt{\delta_r^2 + \delta_q^2} \leq \delta_{max}$ $T_{min} \leq T \leq T_{max}$ $\delta_{r,min} \leq \delta_r \leq \delta_{r,max}$ $\delta_{q,min} \leq \delta_q \leq \delta_{q,max}$

and has a simple implementation. It follows an iterative procedure which explores the set of possible solutions inside a convex polyhedron. The algorithm goes from one edge of the polyhedron to a second one, with each edge being a solution to the problem. Each iteration calculates a feasible solution next to the previous one, such that the current solution is at least as good as the one found at the previous iteration. This method can be used to partially solve the linear case of the control allocation problem defined in Table 1. However, the objective function defined is nonlinear, so it can only yield a particular solution which respects the constraints. The interior-point method can be used to address general-purpose optimization problems, and it is specially well-suited for control allocation problems [2]. The starting point is considered inside a convex polyhedron of feasible solutions, and this point is then updated to find an optimal solution inside the polyhedron. An optimization algorithm based on the interior-point method has been designed in [15] for problems with nonlinear and non-convex objective functions and constraints. The algorithm requires all the constraints and objective functions to be smooth, with the first and second derivatives available. Due to its properties, this algorithm has been selected to solve the problems formulated in Tables 2 and 3. It can be noted that the constraint $\sqrt{\delta_r^2 + \delta_q^2} \leq \delta_{max}$ is nonlinear but convex, while the non-convex constraints of the optimization problems are related to the error terms e_T and e_M.

The MATLAB function *fmincon* can be used to implement the interior-point algorithm, a more detailed description of its implementation being available in [16–18]. To increase the speed and precision of the solver, the analytical expressions of the gradients of the objective function and the constraints have been provided. Otherwise, the solver should estimate the gradient via numerical finite differences. The Hessians can also be directly specified, which makes it possible to reduce the number of iterations if its approximation is not precise. It was chosen to let the solver calculate the Hessian with a quasi-Newton Broyden-Fletcher-Goldfarb-Shanno (BFGS) method. For the explicit constrained problem, stopping criteria have been defined in case of non-convergence of the algorithm (see Table 5).

4. Simulation results

To test the control allocation module, representative profiles of force and torque references have been defined, as displayed in Figure 4. The thrust reference is reachable for the configuration with nine engines, and has been scaled for the three and seven-engine configurations. The thrust reference $T_{re,x}$ is constant during 1s, followed by an exponential decay with time constant 0.7s. The forces on the y-axis $T_{re,y}$, z-axis $T_{re,z}$ and the torques $M_{re,y}$, and $M_{re,z}$ are zero for the first 0.5s, and then follow a sinusoidal profile until 4.5s. The torque roll $M_{re,x}$ has a step of duration 1s. The longitudinal thrust reference $T_{re,x}$ is feasible during the first 0.5s, where it requires all the engines to be at their maximum thrust capability with deflection angles δ_q and δ_r equal to zero. After 0.5s, the references are not feasible because the thrust and torque required on the other axes are beyond the physical constraints of the system. After 1s, the reference becomes feasible again thanks to the exponential decay of the longitudinal thrust component.
A summary of the test scenarios and the associated results results is presented in Table 4, and the parameter tuning of the interior-point optimization algorithm is reported in Table 5. To quantify the accuracy of the solution, it was defined that the error between the trajectory management reference and the engine cluster output cannot exceed 100N for the thrust and 100N m for the torque. This error limit is illustrated by the red line on the right side of the figures.

In Figure 5, the explicit control allocation algorithm is tested for the configuration with three engines. In this case, the algorithm could not converge to a local minimum during the phase where the reference is not feasible. It stopped after reaching a predefined maximum number of iterations. We can see that despite the non-feasibility of the reference, the solution is continuous all over the simulation. Figure 6, compared with Figure 5, highlights the difference between the explicit and implicit constrained problems. In the implicit case, the weighting parameter λ was used to prioritize the minimization of the torque error e_M over the thrust error e_T. In Figure 7, the effect of the linear approximation of the sine and cosine functions can be seen. The optimization algorithm converges faster, but the remaining error is higher compared to the scenario without the approximation (Figure 6).

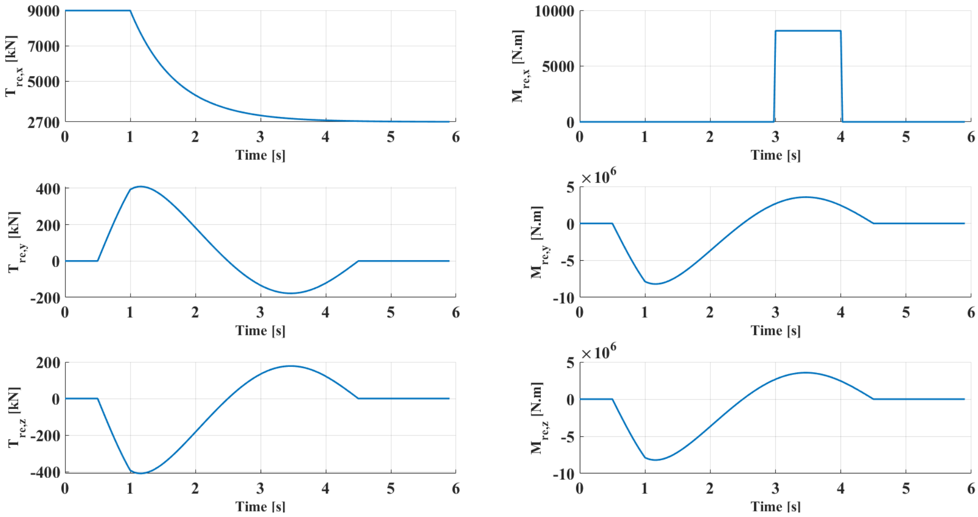

Figure 4: Force and torque references for allocation performance evaluation

Table 4: Summary of the simulated evaluation scenarios for the allocation module

Scenario displayed on	Nb. of engines n	Engine Fault status	Exp. Const.	Small-angle approx.	Nb. it. (mean)	Conv. time in ms (mean)	Total conv.
Fig. 5	3	Nominal	Yes	No	6	11	No
Fig. 6	3	Nominal	No	No	39	75	Yes
Fig. 7	3	Nominal	No	Yes	28	46	Yes
Fig. 8	7	LoE (1 engine)	No	No	36	93	Yes
Fig. 9	9	Float (1 engine)	No	No	19	53	Yes
Fig. 10	9	LIP (1 engine)	No	No	14	39	Yes

Table 5: Tuning parameters of the optimization algorithm for the different evaluation scenarios

Scenario displayed on	w_1	w_2	λ	Max func. evals	Max iter.	Constraint tol.	Optimality tol.	Step tol.
Fig. 5	10^5	/	/	15 000	500	10^{-8}	10^{-12}	10^{-6}
Fig. 6	$1.8 \cdot 10^6$	1	0.65	60 000	2 000	10^{-12}	10^{-12}	10^{-12}
Fig. 7	$3.8 \cdot 10^5$	1	0.65	60 000	2 000	10^{-12}	10^{-12}	10^{-12}
Fig. 8	$3.1 \cdot 10^7$	1	0.01	60 000	2 000	10^{-6}	10^{-12}	10^{-7}
Fig. 9	10^7	1	0.1	60 000	2 000	10^{-5}	10^{-12}	10^{-7}
Fig. 10	$2 \cdot 10^7$	1	10^{-5}	60 000	2 000	$2 \cdot 10^{-5}$	10^{-5}	10^{-7}

Simulations with faulty cases were made with the cluster configurations comprising seven and nine engines. The considered faulty scenarios are active from the beginning of the simulation and affect one non-central engine. In Figure 8, a loss of effectiveness was considered, decreasing the maximum thrust of one engine to 600kN. As the LoE affected a non-central engine, it generates an undesirable torque in the body frame of the launcher. The control allocation module uses the TVC action to compensate for this asymmetry, with the gimbal angles not precisely at 0 when the simulation begins. The same effect is more visible in Figure 9 where a float fault (engine loss) was simulated. A nonzero value of gimbal angles is obtained to compensate for the imbalance generated by the fault. Figure 10 shows the performance of the control allocation module for a locking-in-place (LIP) of both TVC gimbal angles δ_r and δ_q of a non-central engine at angle zero. The compensation for this error is also obtained with the gimbal angles of the other engines, saturated to their respective minimum and maximum values.

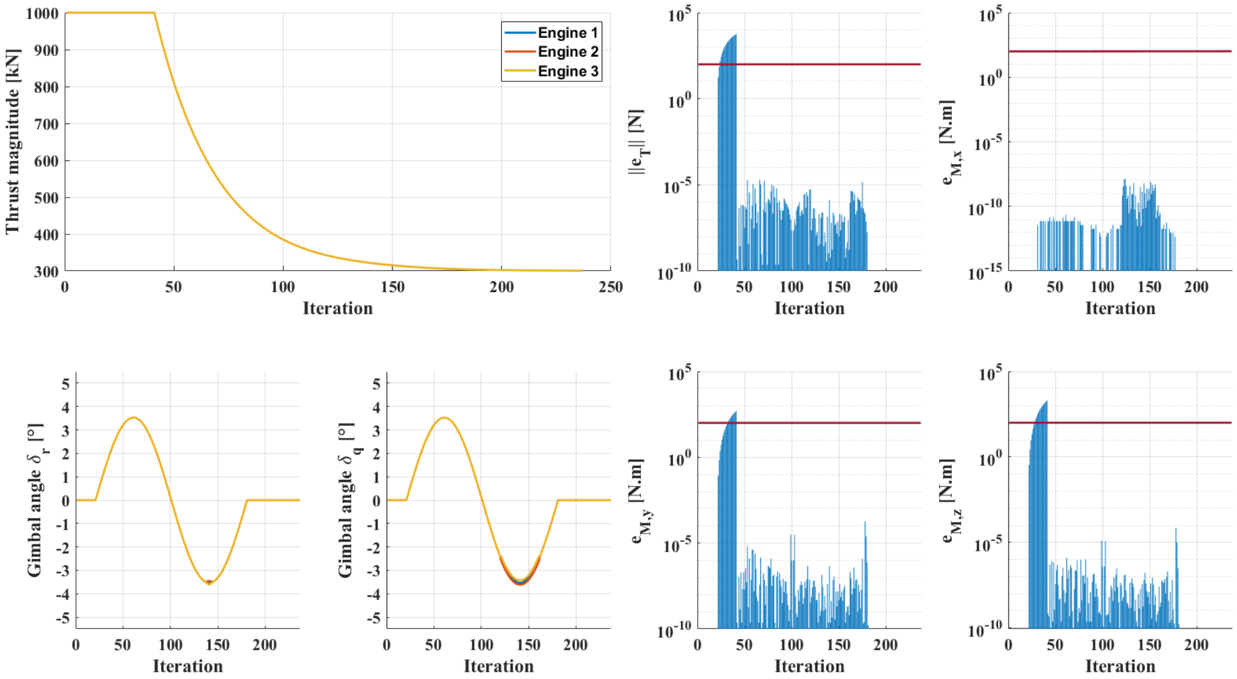

Figure 5: Control allocation result – explicit constrained problem with 3 nominal engines

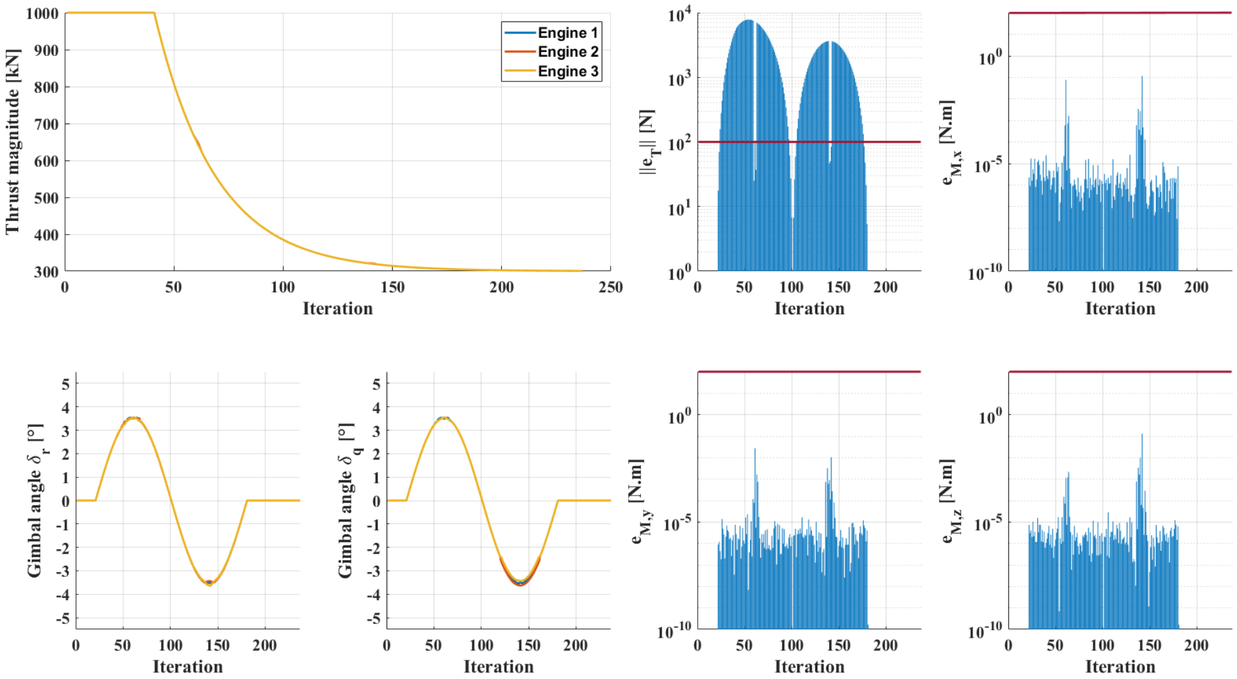

Figure 6: Control allocation result – implicit constrained problem with 3 nominal engines

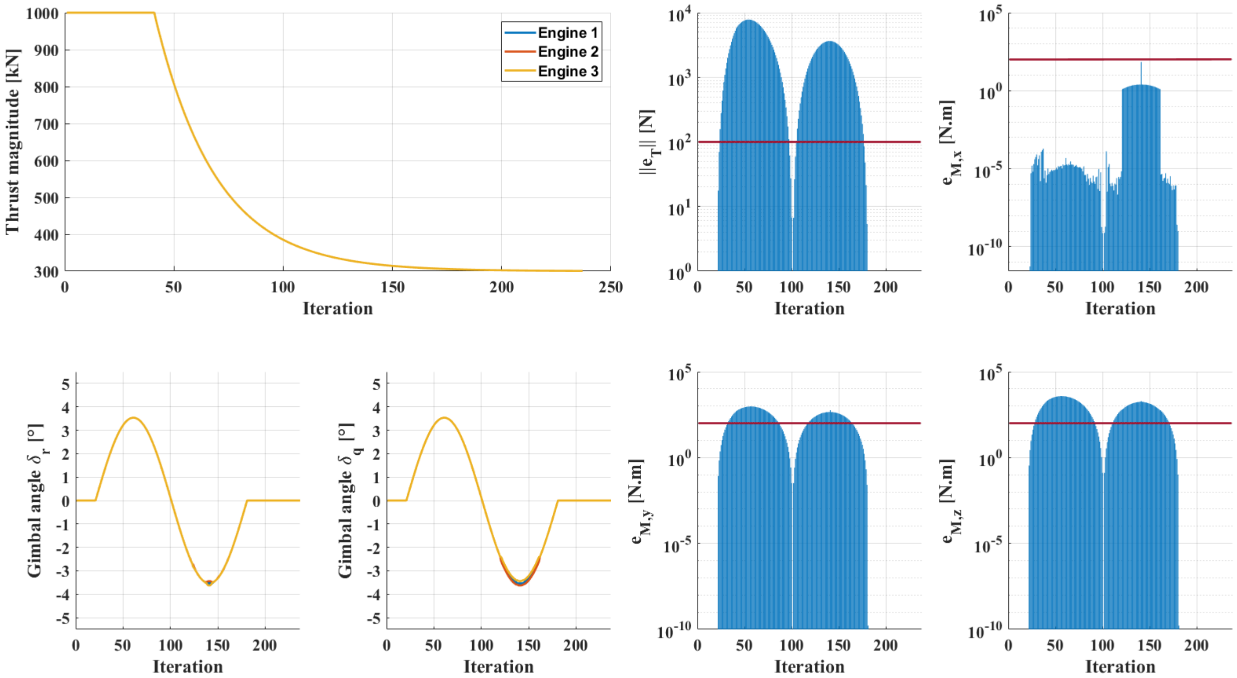

Figure 7: Control allocation result – implicit constrained problem with 3 nominal engines and linear approximation of the sine/cosine functions

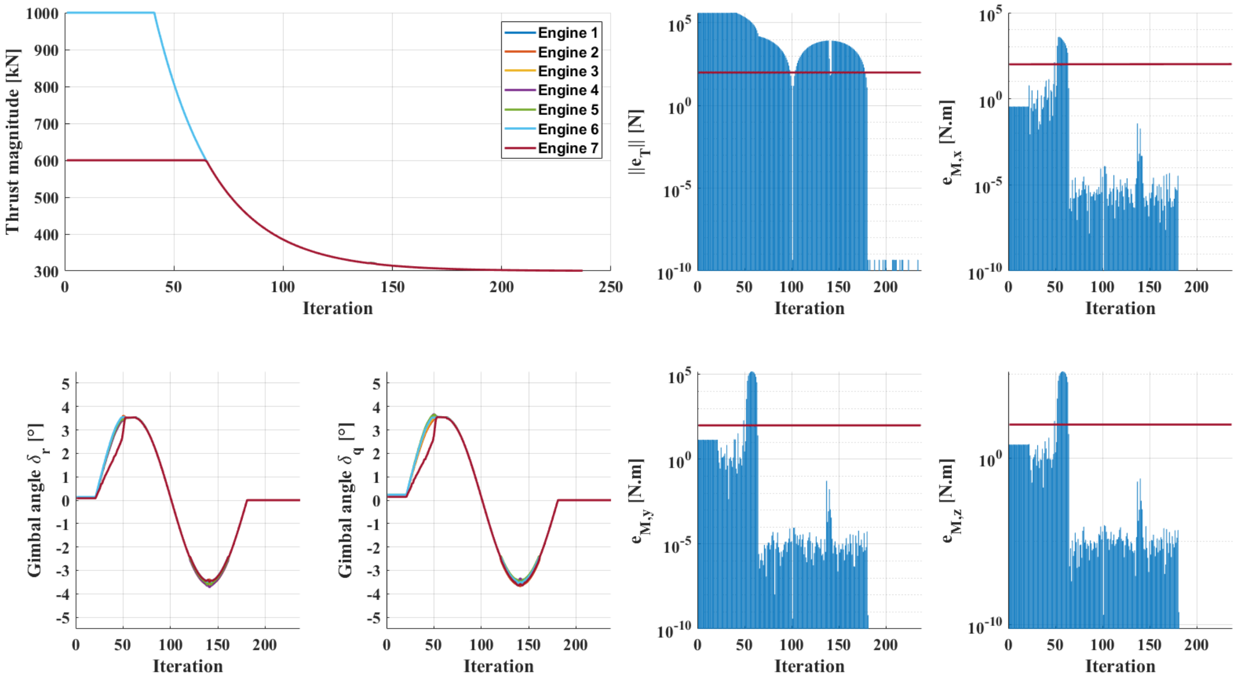

Figure 8: Control allocation result – implicit constrained problem with 1 engine under LoE and 6 nominal engines

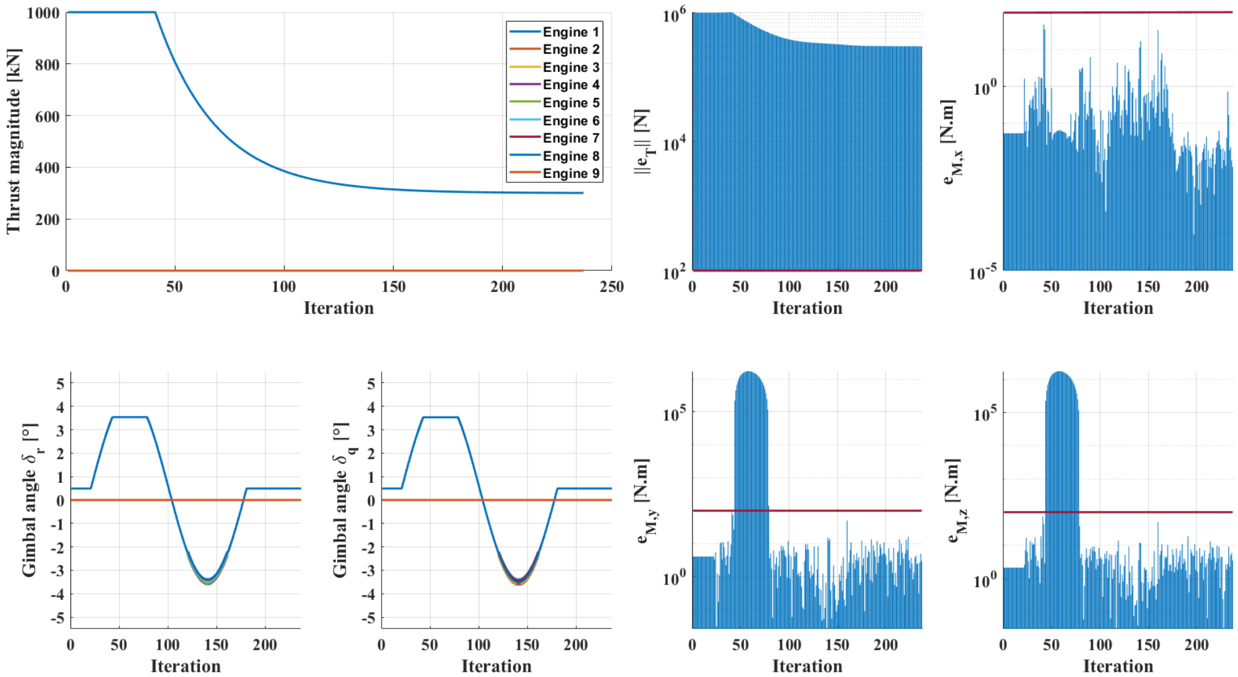

Figure 9: Control allocation result – implicit constrained problem with 8 nominal engines and 1 under float fault

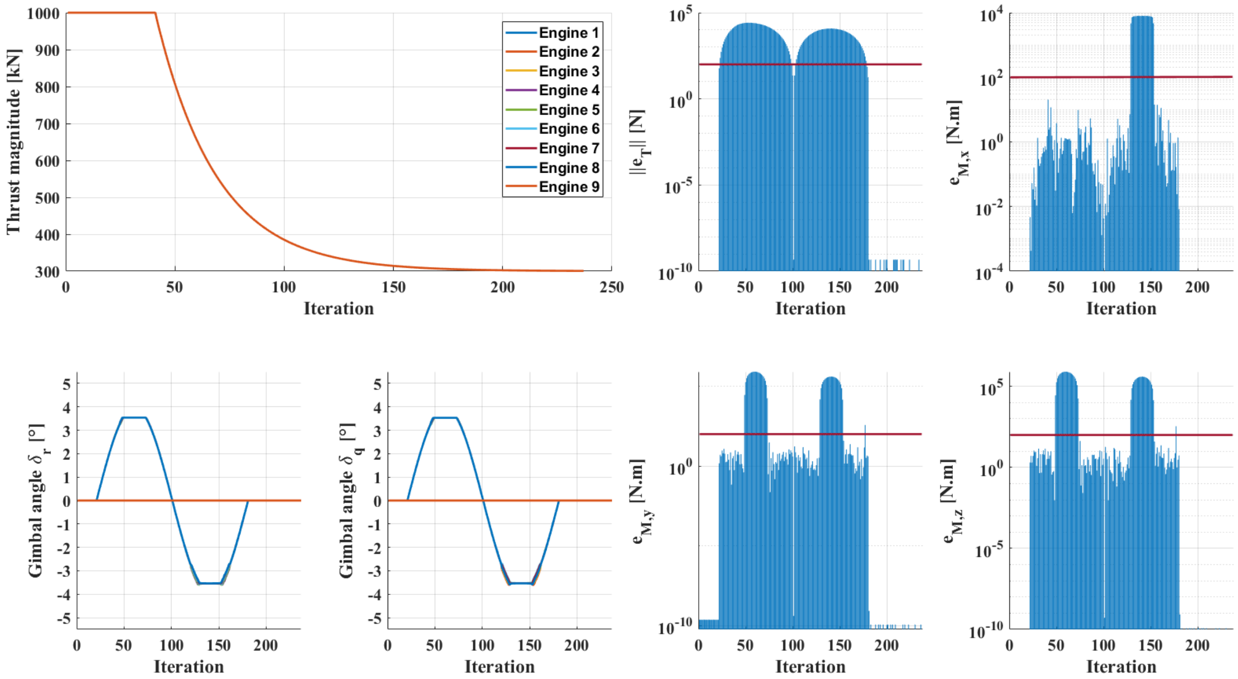

Figure 10: Control allocation result – implicit constrained problem with 8 nominal engines and 1 locked-in-place

5. Conclusions and Perspectives

The application of an optimization-based control allocation method to generate individual references for a multi-engine cluster has been described in this paper. The allocation module is capable of treating different types of faults, namely loss-of-effectiveness of the thrust, floating (total loss of the thrust) and lock-in-place of gimbals.

Based on the launcher geometry and the constraints on the individual thrusts and orientations, three optimization problems have been defined. A linearly constrained optimization problem was first built with simplified versions of the equations. The nonlinear equations were then used to build two alternative optimization problems with respectively explicit and implicit constraints. The linearly constrained problem can be partially solved with the Simplex algorithm, while the interior-point method was used to search solutions for the two other nonlinear and non-convex optimization problems. The reconfiguration of the system in case of faults is addressed by modifying the constraints in the optimization problems. The control allocation algorithm was tested for cluster configurations comprising respectively three, seven, and nine engines. The simulation results show that the control allocation approach was able to take into account the three types of faults. The implicit constrained problem took more time to converge when compared to the explicit one, but it offers more flexibility and tuning options.

Some additional topics could be considered for future work. First, the allocation module was not tested with simultaneous faults even if this is a usual assumption in the design of fault-tolerant systems. The nonlinear and non-convex equations could be rewritten to obtain a convex optimization problem and apply dedicated techniques. The development of a dedicated engine HMS capable of detecting and estimating the faults considered in this paper is necessary for the functioning of the allocation module. The interaction between the allocation and the trajectory management modules could also speed up the optimization if the references are not feasible at all time instants. Finally, the dynamic model of the actuators and their interference could be taken into account in the model of the system.

Acknowledgments

This work is part of the PIC CNES/ONERA on Reusable Launch Vehicles (RLV).

References

[1] Stephane Colas, Serge Le Gonidec, Philippe Saunois, Martine Ganet, Antoine Remy, and Vincent Leboeuf. A point of view about the control of a reusable engine cluster. In *Proceedings of the 8th European Conference for Aeronautics and Space Sciences (EUCASS)*, 2019.

[2] Tor Arne Johansen and Thor I Fossen. Control allocation—A survey. *Automatica*, 49(5):1087–1103, 2013.

[3] Youmin Zhang, V Sivasubramaniam Suresh, Bin Jiang, and Didier Theilliol. Reconfigurable control allocation against aircraft control effector failures. In *2007 IEEE International Conference on Control Applications*, pages 1197–1202. IEEE, 2007.

[4] James Buffington. Tailless aircraft control allocation. In *AIAA Guidance, Navigation, and Control Conference*, 1997.

[5] Thor I Fossen and Tor Arne Johansen. A survey of control allocation methods for ships and underwater vehicles. In *14th Mediterranean Conference on Control and Automation*, 2006.

[6] Johannes Tjonnas and Tor Arne Johansen. Stabilization of automotive vehicles using active steering and adaptive brake control allocation. *IEEE Transactions on Control Systems Technology*, 18(3):545–558, 2009.

[7] Ola Härkegård and S. Torkel Glad. Resolving actuator redundancy—-optimal control vs. control allocation. *Automatica*, 41(1):137–144, 2005.

[8] Alessandro Casavola and Emanuele Garone. Fault-tolerant adaptive control allocation schemes for overactuated systems. *International Journal of Robust and Nonlinear Control*, 20(17):1958–1980, 2010.

[9] Halim Alwi and Christopher Edwards. Fault tolerant control using sliding modes with on-line control allocation. *Automatica*, 44(7):1859–1866, 2008.

[10] Andrea Cristofaro and Tor Arne Johansen. Fault tolerant control allocation using unknown input observers. *Automatica*, 50(7):1891–1897, 2014.

[11] Antoine Abauzit and Julien Marzat. A multiple-observer scheme for fault detection, isolation and recovery of satellite thrusters. In *Advances in Aerospace Guidance, Navigation and Control*, pages 199–214. Springer, 2013.

[12] Carlo Alberto Pascucci, Michael Szmuk, and Behçet Açikmeşe. Optimal control allocation for a multi-engine overactuated spacecraft. In *IEEE Aerospace Conference*, 2017.

[13] Jovan D Bošković and Raman K Mehra. Failure detection, identification and reconfiguration in flight control. In *Fault Diagnosis and Fault Tolerance for Mechatronic Systems: Recent Advances*, pages 129–167. Springer, 2003.

[14] Marc Bodson. Evaluation of optimization methods for control allocation. *Journal of Guidance, Control, and Dynamics*, 25(4):703–711, 2002.

[15] Richard H Byrd, Mary E Hribar, and Jorge Nocedal. An interior point algorithm for large-scale nonlinear programming. *SIAM Journal on Optimization*, 9(4):877–900, 1999.

[16] Jorge Nocedal, Figen Öztoprak, and Richard A Waltz. An interior point method for nonlinear programming with infeasibility detection capabilities. *Optimization Methods and Software*, 29(4):837–854, 2014.

[17] John E Dennis Jr and Robert B Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations*. SIAM, 1996.

[18] Michael JD Powell. Algorithms for nonlinear constraints that use Lagrangian functions. *Mathematical programming*, 14(1):224–248, 1978.