



HAL
open science

Robust Transient Control of Reusable Liquid-Propellant Rocket Engines

Sergio Pérez-Roca, Julien Marzat, H el ene Piet-Lahanier, Nicolas Langlois, Marco Galeotta, Francois Farago, Serge Le Gonidec

► **To cite this version:**

Sergio P erez-Roca, Julien Marzat, H el ene Piet-Lahanier, Nicolas Langlois, Marco Galeotta, et al.. Robust Transient Control of Reusable Liquid-Propellant Rocket Engines. IAC 2019, Oct 2019, Washington, United States. <hal-02421543>

HAL Id: hal-02421543

<https://hal.science/hal-02421543v1>

Submitted on 20 Dec 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destin ee au d ep ot et  a la diffusion de documents scientifiques de niveau recherche, publi es ou non,  emanant des  tablissements d'enseignement et de recherche fran ais ou  trangers, des laboratoires publics ou priv es.



HAL Authorization

Robust Transient Control of Reusable Liquid-Propellant Rocket Engines

Sergio Pérez-Roca^{a,c,*}, Julien Marzat^a, Hélène Piet-Lahanier^a, Nicolas Langlois^b, Marco Galeotta^c, François Farago^c and Serge Le Gonidec^d

^a DTIS, ONERA, Université Paris-Saclay, Chemin de la Hunière, 91120 Palaiseau, France, {[sergio.perez_roca](mailto:sergio.perez_roca@onera.fr), [julien.marzat](mailto:julien.marzat@onera.fr), [helene.piet-lahanier](mailto:helene.piet-lahanier@onera.fr)}@onera.fr

^b Normandie Université, UNIROUEN, ESIGELEC, IRSEEM, Rouen, France, nicolas.langlois@esigelec.fr

^c CNES - Direction des Lanceurs, 52 Rue Jacques Hillairet, 75612 Paris, France, {[francois.farago](mailto:francois.farago@cnes.fr), [marco.galeotta](mailto:marco.galeotta@cnes.fr)}@cnes.fr

^d ArianeGroup SAS, Forêt de Vernon, 27208 Vernon, France, serge.le-gonidec@ariane.group

* Corresponding Author

Abstract

The current trend towards a more affordable access to space is generally materialising in reusable launchers and engines. From the control perspective, these reusable liquid-propellant rocket engines (LPRE) imply more demanding robustness requirements than expendable ones, mainly because of their multi-restart and thrust-modulation capabilities. Classically, the control system handles LPRE operation at a finite set of predefined points. That approach reduces their throttability domain to a restricted interval in which they are designed to be safe in nominal conditions. Moreover, the operation of their transient phases, which have a great impact on the duration of engine life, is not robust to the possible engine evolution. Hence, the goal of this work is to develop a control loop which is adapted to the whole set of operating phases, transient and steady-state, and which is robust to internal parametric variations. Several blocks have been assembled to constitute the control loop: engine simulation, reference generation and several controllers. First, simulators representative of the gas-generator-cycle (GG) *Vulcain 1* and *PROMETHEUS* engines were built. The purely thermodynamic modelling of the cycle was subsequently adapted to the control framework, obtaining a nonlinear state-space model. The available actuators are continuously controllable valves, binary igniters and binary starters. These actuators are related to discrete events in transient phases. Regarding the start-up operation, the igniter, starter and valves are activated during the first seconds. Up from the end of those activations, the whole system behaves in a fully continuous way. Hence, a different control strategy is proposed for each sub-phase. For the first and discrete sub-phase, a discrete optimisation of events timing is proposed, in which the time differences between events are adapted according to operation criteria and constraints. This trajectory planning, still under implementation, is to be performed off-line. The subsequent continuous sub-phase is feedback controlled to track pre-computed reference trajectories. Apart from the start-up, throttling scenarios also present a dedicated end-state-tracking algorithm. A model-based control method, Model Predictive Control, has been applied in a linearised manner with robustness guarantees to all these scenarios, in which a set of hard state and control constraints must be respected. Tracking of pressure (thrust) and mixture-ratio operating points within the design envelope is achieved in simulation along the continuous sub-phase while respecting constraints. Robustness to variations of the parameters, which are checked to be predominant according to analyses, is also demonstrated.

Keywords: Liquid-propellant rocket engines, Model Predictive Control, transient phases, robustness, reusability.

1. Introduction

The current context of launcher vehicles design is strongly linked to their reusability. From the automatic control perspective, this potential need for reusable liquid-propellant rocket engines (LPRE) implies stricter robustness requirements, related to their multi-restart and thrust-modulation capabilities. These demanding requirements stem from the possible internal perturbations caused by components faults or evolving parameters and from exogenous perturbations linked to the more complex mission profiles conceived for new launchers.

Multivariable-control developments of main-stage LPRE have attained a short throttling envelope (70%-120%) in test benches [1]. In real engines, the control system is generally designed to achieve the nominal operating point. One of the new features of the future European *PROMETHEUS* engine is the ability to throttle down to 30% of thrust [2]. Consequently, an enlarged controlled operating domain has to be feasible. Tracking and robustness must be kept at those low throttle levels, where physical phenomena are harder to anticipate.

The main control goal in these multivariable systems consists in tracking reference points in combustion-

chamber pressure and mixture ratio, which come from a high-level order according to launcher needs. Control-valves opening angles are changed so as to adjust engine's operating point while verifying a set of constraints. The majority of control studies in the literature employs linearised models about operating points for synthesising steady-state controllers, mostly based on PID (proportional, integral, derivative) techniques (such as [3]). In most papers, initial MIMO (Multi Input Multi Output) systems are decoupled into dominant SISO (Single Input Single Output) subsystems. Off-line optimisation studies have also been carried out [4]. Other works in the literature, incorporating more advanced techniques such as nonlinear [5], hybrid [6] or robust [7] ones, reinforce certain indicators of performance and robustness. However, there is an absence of publications which concern not only the steady state but also the demanding transient phases at the same level of performance and robustness, as reviewed in [8]. There is also a lack of method comparisons on a common benchmark, even simulated. Besides, only narrow throttling domains are feasible.

Sequential transient phases of engine operation (start-up and shutdown), are generally executed in open loop with narrow correction margins. The first sub-phase of the start-up transient is determined by a series of discrete activations. After all these commands, the second sub-phase of the transient, which is fully continuous, takes place until the steady state is reached. Open-loop (OL) control is normally applied due to the controllability and observability issues at very low mass flows [3]. In this paper, different control approaches are proposed for each sub-phase. Throttling transient scenarios are also treated differently.

The main objective of this work is to control the transient phases of pump-fed LPRE. Concretely, tracking of combustion pressure (linked to thrust) and mixture ratio all along the transients is targeted. Simultaneously, a set of hard operational constraints has to be respected, related to mixture ratios, turbopumps rotational speeds and valves actuators angular velocities. The LPRE cycle studied is the gas-generator-cycle engine, but the method here is conceived to be applicable to other cycles. Control approaches in this paper are based on Model Predictive Control (MPC) techniques. MPC has been selected as the most adequate for this type of complex systems with hard constraints. Indeed, it is gaining popularity at academic and industrial levels and can incorporate robustness [9] or hybrid aspects, which are important for future work on this topic.

This paper is organised as follows. In Section 2, the modelling approach for simulation and control is described. In Section 3, the derived models are analysed, especially from the point of view of

sensitivity to parameters. Section 4 describes the different control strategies developed, mainly based on MPC techniques. The main results, concerning the selected scenarios, are presented and discussed in Section 5. Finally, Section 6 serves as a conclusion.

2. Modelling approach

The modelling approach considered in these studies was first described in [10] and revisited in [11,12]. Here it is again revisited and updated for considering different engine case-studies. Firstly it is relevant to clarify that several model structures are employed in the different blocks of the control loop in this paper. Concerning the plant to which the control is applied, a simulator of the real plant was constructed in the first place, instead of testing on a real engine or on a hardware-in-the-loop test bench. Indeed, a simple, dynamic and efficient way of modelling generic LPRE was sought, instead of using more accurate programmes or computations. An easy integration into *Simulink* was also preferred so as to easily test different control methods. Along these lines, a new *Simulink* library of rocket-engine components has been developed to build a simulator of the gas-generator-cycle engine. It has been named *T-RETM*, Toolbox for Rocket-Engine Transient Modelling.

This simulator, whose structure is built component-wise, contains the basic thermo-fluid-dynamics and mechanics of LPRE elements: mass, energy and momentum conservation equations. Simplified 0-D models of resistive components (valves, pipes, turbopumps) and capacitive ones (cavities, combustion chambers) have been developed. Subsequently, all components are joined in *Simulink* according to the selected engine. The engines considered in this paper, representative of *Vulcain 1* and *PROMETHEUS*, present a GG (gas generator) cycle. The former consists in a *LOX/LH₂* (liquid oxygen as oxidiser, liquid hydrogen as fuel) engine, while the latter presents the *LOX/LCH₄* mixture (liquid oxygen, liquid methane). In Fig. 1 the *Vulcain 1* cycle and main components are depicted and the main acronyms are summarised, while Fig. 2 is devoted to *PROMETHEUS*. The hot-gas flow necessary to drive turbines comes from a GG, a small combustion chamber that receives a small portion of the main propellant flow. The actuators considered in this paper are four continuously controllable valves (VCF, VCO, VGF, VGO) and a fifth one in the case of *Vulcain 1* (VGC). Apart from those, there are three discrete actuators: two binary chamber igniters (i_{CC} , i_{GG}) and one binary starter (i_{sta}). In *Vulcain 1* $i_{GG} \equiv i_{sta}$ can be assumed according to the sequence. That GG starter injects hot gas into the cavity during less than $1.5s$ so as to start driving turbines. *Vulcain 1* requires two turbopump shafts while *PROMETHEUS* only needs one, accounting for the density differences between propellants. This consists in the main contribution to the start-up transient phases of LPRE, because once

turbines start rotating, pumps can provide more flow to chambers, which increases combustion pressures and temperatures. These increases also lead to greater shaft speeds until a steady-state is achieved, at around three to four seconds after start.

Indeed, the activations of the aforementioned actuators, both the initial opening of valves and the ignitions, constitute the so-called start-up sequence. That consists in the first and discrete sub-phase of the start-up transient. Once all actuators are active, a continuous sub-phase takes place, where valves are continuously adjusted so as to reach a desired steady state, which is still far when the discrete sub-phase ends.

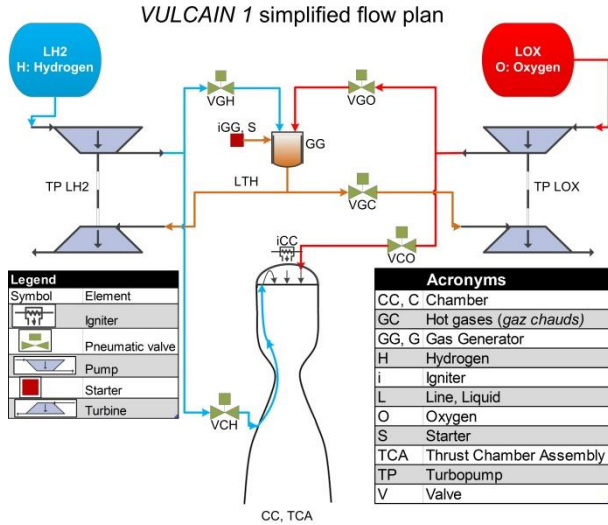


Fig. 1. *Vulcain 1* flow plan considered in model

Valves angles (α), which present a nonlinear but direct relation to sections (A), control the flows to the main combustion chamber (VCF and VCO), to the GG (VGF, VGO), and to the oxidiser turbine (VGC, in *Vulcain 1*). The latter is the main contributor to adjusting mixture ratio (MR), defined as the quotient between oxidiser (O) and fuel (F) mass flow rates $MR = \frac{\dot{m}_O}{\dot{m}_F}$. This ratio, a major behaviour indicator in LPRE, is established at three levels: at an engine's global level (MR_{PI}), taking pumped propellants into account; in the combustion chamber (MR_{CC}) and in the GG (MR_{GG}).

Since the beginning it was aimed at capturing in models the transient behaviour of the engine with the available engine parameters. No identification techniques from real data or precise multi-physical simulation platforms were available to the project. Thus, the developed simulator was then translated into a nonlinear state-space model by joining components equations symbolically. That is to say, a global expression of system's dynamic behaviour as a function of its state variables, internal parameters and control inputs was targeted.

This process was performed via the symbolic mathematical environment *Maple*, taking into account the internal definition of each LPRE component and of the global system (interconnection of components, as shown in Fig. 1 and 2).

At this stage, having already performed some simplifications with respect to the initial simulator, the model is referred to as *complex NLSS* (nonlinear state-space) or $f_c(\mathbf{x}, \mathbf{u})$. However, this model presented an excessive complexity for control design, since many ODEs (ordinary differential equation) extended over several pages. Thus, it was further reduced until attaining the here-called simplified NLSS such that $\dot{\mathbf{x}} = f_s(\mathbf{x}, \mathbf{u})$, more tractable for the derivation of control laws (more details in [10]).

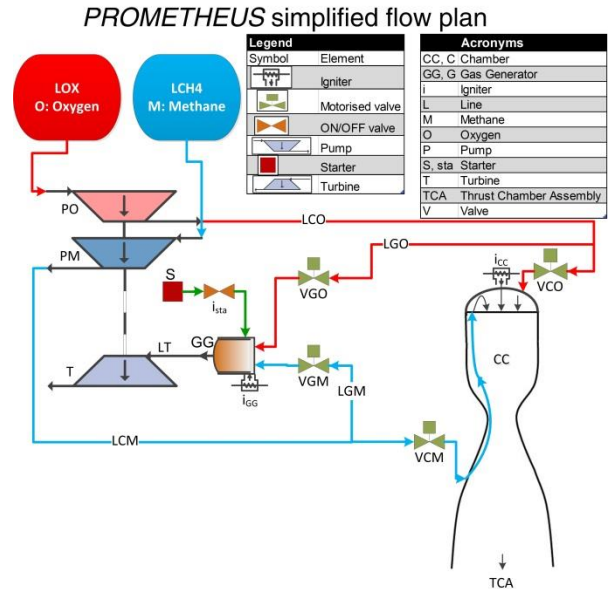


Fig. 2. *PROMETHEUS* flow plan considered in model

The number of states is n (12 in *Vulcain 1* (V.) and 9 in *PROMETHEUS* (P.)) and m is the number of control inputs (5 in V. and 4 in P.). Here, the state vector \mathbf{x} , of both NLSS, comprises turbopumps speeds, the several pressures in the system (combustion chamber, GG, and pre-turbine cavities), and mass flows streaming through valves and pre-turbine pipes. Thus, in V., the state vector is defined as:

$$\mathbf{x}_V = [\omega_H \ \omega_O \ p_{CC} \ p_{GG} \ p_{LTH} \ p_{VGC} \ \dot{m}_{LTH} \ \dot{m}_{VCF} \ \dot{m}_{VCO} \ \dot{m}_{VGF} \ \dot{m}_{VGO} \ \dot{m}_{VGC}]^T,$$

and for P. it is:

$$\mathbf{x}_P = [\omega_{PCC} \ p_{GG} \ p_{LT} \ \dot{m}_{LT} \ \dot{m}_{VCF} \ \dot{m}_{VCO} \ \dot{m}_{VGF} \ \dot{m}_{VGO}]^T,$$

The states with higher tracking importance are incorporated into a reduced state vector \mathbf{x}_z :

$$\mathbf{x}_z = [p_{CC} \ \dot{m}_{VCF} \ \dot{m}_{VCO} \ \dot{m}_{VGF} \ \dot{m}_{VGO}]^T.$$

The control input $\mathbf{u} = [\mathbf{u}_c \ \mathbf{u}_d]^T$ contains control inputs of continuous and discrete nature, which renders the model hybrid from the control perspective. The sections of control valves are the continuous ones:

$\mathbf{u}_c = [A_{VCF} A_{VCO} A_{VGF} A_{VGO} (A_{VGC} \text{ in } V.)]^T$, while igniters and starter activations consist in the discrete ones:

$$\mathbf{u}_d = [i_{cc} i_{GG} i_{sta}]^T.$$

Besides, all equations, states and control are rendered non-dimensional with respect to the nominal equilibrium values. An overview of the *Vulcain 1* state-space system can be found in [12]. Here, the form of the *PROMETHEUS* dynamic system $\dot{\mathbf{x}} = f_s(\mathbf{x}, \mathbf{u})$ is presented, where $a_i, b_i, \dots, k_i \in \mathbb{R}$ are internal-parameters conglomerates and w_t is an exogenous input corresponding to starter mass flow:

$$\begin{cases} \dot{x}_1 = (a_1 x_1^2 + b_1 x_1 x_4 + c_1 x_4 + x_1 (d_1 x_6 + e_1 x_7 + f_1 x_8 + g_1 x_9) + h_1 x_6^2 + i_1 x_7^2 + j_1 x_6 x_8 + k_1 x_7 x_9) u_7, \\ \dot{x}_2 = (a_2 x_6 + b_2 x_7) u_5 + c_2 x_6 + d_2 x_7 + e_2 x_2, \\ \dot{x}_3 = (a_3 x_8 + b_3 x_9) u_6 + c_3 x_8 + d_3 x_9 + e_3 x_5 + f_3 w_t, \\ \dot{x}_4 = a_4 x_4 + b_4 x_5, \\ \dot{x}_5 = a_5 (x_3 - x_4) + b_5 x_5^2, \\ \dot{x}_6 = \frac{(a_6 x_1^2 + b_6 x_6^2 + c_6 x_6 x_8 + d_6 x_2 + e_6) u_1^2 + f_6 x_6^2}{(g_6 u_1 + h_6) u_1}, \\ \dot{x}_7 = (a_7 x_1^2 + b_7 x_7^2 + c_7 x_1 x_7 + d_7 x_2 + e_7) u_2 + \frac{f_7 x_7^2}{u_2}, \\ \dot{x}_8 = (a_8 x_1^2 + b_8 x_6^2 + c_8 x_8^2 + d_8 x_6 x_8 + e_8 x_3 + f_8) u_3 + \frac{g_8 x_8^2}{u_3}, \\ \dot{x}_9 = (a_9 x_1^2 + b_9 x_7^2 + c_9 x_9^2 + d_9 x_1 x_7 + e_9 x_3 + f_9) u_4 + \frac{g_9 x_9^2}{u_4}. \end{cases} \quad (1)$$

It is clear that this system representation, even if it has been simplified, presents numerous nonlinearities and is non-affine with respect to control.

3. Models analysis

The analysis of the previous models is necessary in order to select and build the most appropriate control strategy. Several aspects can be analysed, ranging from the effect of simplifications, stability, measurability and observability, controllability, sensitivity to parameter variations, etc.

The different simplifications carried out obviously increase the modelling error slightly. It is specially present in mass flows, which can present a mismatch of 10 to 25% at each step of simplification (simulator, f_c , f_s , and linearised models). Errors in the rest of states remain below 10% at each step.

Stability is a general property of these GG LPRE models, which describe a system that compensates itself when a perturbation appears in the cycle due to its high coupling [8]. The possible destabilising phenomena in LPRE, such as combustion instabilities or frequency couplings, are not modelled in these state-space models for the sake of simplicity. Hence, when analysing linearised models about transient trajectories, all eigenvalues present negative real parts.

The state is assumed to be completely measurable in the real engine. This is a realistic assumption for ω and p . However, measuring some mass flows would be

problematic in terms of engine design. Mass flows are normally not measured in LPRE, but estimated through pressure, temperature and volumetric flow measurements. This estimation process is assumed perfect in this paper, but consists in a future block to add to the loop. In fact, during the discrete part of the transient, some observability issues can appear due to the low volumetric flows characteristic of that phase [3]. This, apart from other issues highlighted in Section 4, hinders the realisation of closed-loop (CL) control during the discrete sub-phase of transients.

Controllability can be verified after linearising f_s about the whole continuous start-up trajectory. In other words, all states can be controlled via valves during the continuous sub-phase. However, as commented for observability, controllability is not ensured during the discrete sub-phase, where there are valves that have to remain closed in order to execute a safe sequence. As a consequence, not all states can be controlled. Indeed, the first valves to open are CC-injection ones, which are considerably less influential in the control of the system than GG ones, which open at the end of the sequence, after ignitions and starter activation.

The previous models establish a complex but deterministic relation between control inputs and system states. No stochastic considerations are explicitly included, which could arise from the fact that some parameters or dynamics are not perfectly known a priori. This is an issue when modelling this kind of complex thermodynamic systems, since some of the parameters used in the differential equations are estimated from test data or tuned in simulations. In order to analyse the effect of parameters on the behaviour of the engine, a series of sensitivity analyses have been performed. In the following, only *Vulcain 1* will be used as case study due to the availability of a more consolidated set of internal parameters in contrast to *PROMETHEUS*, still under development. The list of considered engine parameters that might vary during operation or that are estimated for modelling (and hence not 100% certain) is summarised in Table 1.

In order to study the influence of each individual parameter on the transient behaviour of the engine, the Sobol sensitivity analysis [13] has been applied. This method is appropriate for large, highly-coupled nonlinear systems such as this one. In addition, it is a global sensitivity-analysis method, which allows to identify the relative contributions of each parameter independently as well as the combined variations to the variance in the output. Here, it is of interest to study the variations in terms of a global performance criterion J_{Sobol} , defined as the weighted addition of several simulated indicators:

$$J_{Sobol} = |\text{err}_{pcc}| + |\text{err}_{MRCC}| + |\text{err}_{MRGG}| + |\text{err}_{MRPI}| + 0.001 \cdot |A_{pcc}| + 0.01 \cdot |\text{os}_{pcc}|,$$

where err are static steady-state errors in the main operating quantities (contained in \mathbf{x}_z), os is overshoot and $A_{p_{CC}}$ is the surface between the perturbed p_{CC} start-up transient curve and the nominal one in open-loop (OL). The sampling of parameters variations according to Table 1 has been carried out via the Latin Hypercube sampling, recommended for calculating the Sobol global indices. In this fashion, a set of 1000 parameter-variation combinations is provided to a Kriging-based Sobol algorithm based on [14]. This algorithm creates a Kriging model based on the provided input-output data, which avoids the execution of costly simulations. After running 1 million Monte Carlo evaluations of that model, the most influential parameters, related to the higher Sobol global indices, can be identified. These are $AthCC$, $AthT$ and p_{tanks} , whose indices present an order of magnitude of 0.1 (normalised with respect to one), while $ResI$, $ResCR$ and η_{GG} indices are about 0.01. $AthCC$ and $AthT$ are considered due to possible inter-engine discrepancy or varying effective hydraulic sections during operation. Oscillations in p_{tanks} are also possible during operation, while $ResI$, $ResCR$ and η_{GG} are not certainly known, they are estimated for modelling. This shortlist of parameters, to which the model is more sensitive, is taken into account in the control algorithm (Section 4).

Table 1. List of *Vulcain 1* engine parameters considered in sensitivity analysis

Parameter	Definition	Considered variation range [%]
p_{tanks}	Tanks pressures, coupled with inlet densities and temperatures	± 10
$ResLC$	LC lines fluidic resistances	± 10
$ResI$	Injectors fluidic resistances	± 10
$ResCR$	Cooling-circuit fluidic resistance	± 10
$IneCR$	Cooling-circuit fluidic inertia	± 10
η_{CC}	CC combustion efficiency	± 5
η_{GG}	GG combustion efficiency	± 5
$ResLTH$	LTH line fluidic resistance	± 10
$IneLTH$	LTH line fluidic inertia	± 10
V_{cav}	Pre-turbine cavities volume	± 10
$AthCC$	CC throat section	± 5
$AthT$	Turbine inlet sections	± 5
$AthLE$	Turbine outlet sections	± 5

The effects of variations in the rest of parameters can be neglected according to the sensitivity analysis.

4. Control approaches

The control goals mainly concern reaching a desired end-state or following a predefined trajectory while complying with a set of hard constraints on \mathbf{x} and \mathbf{u} . This second goal is very important during transient phases, in order to avoid excessive mixture ratios (and hence temperatures), pressures or rotational speeds. Reusability requirements also point to this goal.

The proposed structure of the whole control loop is depicted in Fig. 3.

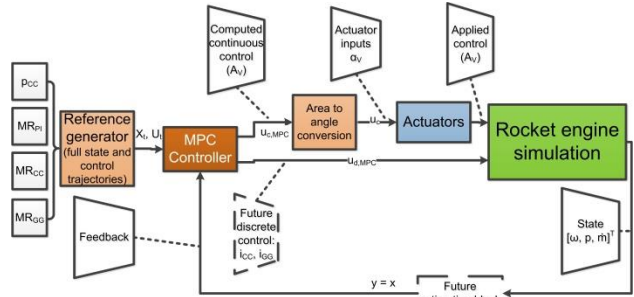


Fig. 3. Complete control loop, with areas of future work

To the right there is the plant block, where the LPRE simulator is placed. It simulates the behaviour of the engine via integration of the system of ODE as a function of its state, parameters and control inputs. The provided control inputs come from the controller, which computes the orders sent to valves (actuators block). These actuators are modelled as a separate entity via simplified second-order models, requiring an angular input instead of sections, which are considered in the controller. In fact, there is a nonlinear monotone relation between angles and sections, and hence it is more convenient to avoid that additional equation in the state-space models used in the controller. This controller receives the full state measurement from the simulator, since estimation is still considered perfect (future work). It also receives a reference to follow, which consists in pre-generated trajectories. Indeed, that reference is generated in a preprocessor connected to the controller.

4.1 Preprocessor

The role of the preprocessor is to generate in the first place a target reference ($\mathbf{x}_r, \mathbf{u}_r$) in terms of the full state and control vectors. In fact, the orders provided by the launcher to the engine control system only concern the final desired $p_{CC,r}$, $MR_{CC,r}$, $MR_{GG,r}$ and $MR_{PI,r}$. In order to obtain the whole \mathbf{x} used in the state-space models in this work, a calculation has to be performed. This is explained in [11]. Basically, system's ODE are equated to zero in order to obtain an equilibrium point that coincides with the provided input elements. It is paramount to obtain this full-state and control

equilibrium point because the control approach in this study is fully model-based, as explained in the next subsections.

Once these $(\mathbf{x}_r, \mathbf{u}_r)$ are computed, if the controller has to deal with the continuous start-up transient, trajectories $(\mathbf{X}_t, \mathbf{U}_t)$ for states and control are also pre-generated off-line according to the initial conditions and to the desired \mathbf{x}_r . \mathbf{X} and \mathbf{U} are defined in general as the series of \mathbf{x} and \mathbf{u} at each time step k along a horizon N (valid throughout the whole paper):

$$\begin{aligned} \mathbf{X} &= [\mathbf{x}_1, \dots, \mathbf{x}_k, \dots, \mathbf{x}_N]^T \\ \mathbf{U} &= [\mathbf{u}_1, \dots, \mathbf{u}_k, \dots, \mathbf{u}_N]^T \end{aligned}$$

This trajectory planning is explained in [12], where an optimisation- and model-predictive-based method is proposed. This computation method can be regarded as an OL finite-horizon MPC scheme, typically used in trajectory planning [15]. A simple cost based on the current quadratic error with respect to $(\mathbf{x}_r, \mathbf{u}_r)$ is considered. The NLSS f_s is used for defining the nonlinear-dynamics constraints, more relevant during the initial build-up phases. Close to the steady state, linearised models are used, since modelling error is reduced. In addition, all important operating constraints can be imposed in the planning, which in the end provides a reference control trajectory that is safer than the constant OL openings.

Thus, the preprocessor serves to define the references to be tracked by the controller, either an end state or a trajectory.

4.2 MPC Controllers

Controllers in this study are based on MPC. This family of techniques is an optimisation-based control approach, where the control law is recomputed at each time step k with the goal of minimising a cost function. Basically, control inputs are computed according to the minimisation of the cost while respecting all dynamic and operating constraints along a discretised time horizon, in which the behaviour of the system is predicted with models. Only the first computed control of the horizon, corresponding to the first time step, is factually applied to the plant, since at the next step the whole process is repeated.

In this paper, different controllers are proposed according to the scenario that the engine is facing. As explained in previous sections, one of the goals of this work is to control start-up transients in CL. These transients contain a discrete and a continuous phase, which are treated differently here. The latter was considered in a first place so as to simplify the discrete elements of models, which are already active in that phase. Apart from the start-up, operating-point changes or throttling scenarios (fully continuous) are also targeted.

4.2.1 Continuous MPC controllers

Regarding the controllers for fully continuous scenarios, the tracking goal is different whether throttling or start-up is to be controlled.

Throttling: an operating-point change is performed here by tracking a new desired equilibrium $(\mathbf{x}_r, \mathbf{u}_r)$, without specifying a reference trajectory. The algorithm for this kind of scenarios can be taken from the previous paper [11], where it was originally presented as an also valid option for controlling the continuous start-up. The linear MPC algorithm considered there presented linear dynamic constraints in the form:

$$\Delta \mathbf{x}_{k+1} = A_d(\mathbf{x}_r, \mathbf{u}_r) \Delta \mathbf{x}_k + B_d(\mathbf{x}_r, \mathbf{u}_r) \Delta \mathbf{u}_k, \quad (2)$$

where A_d and B_d are the zero-hold-discretised and linearised state-space matrices. In (2) they are always evaluated at the end point to track. The goal is to find the control $\Delta \mathbf{u} = \mathbf{u} - \mathbf{u}_r = 0$ that drives the system to $\Delta \mathbf{x} = \mathbf{x} - \mathbf{x}_r = 0$. Hence, the cost to be minimised is then:

$$\begin{aligned} J_{end}(\mathbf{X}, \mathbf{U}, \mathbf{Z}) &= \\ & \left(\sum_{k=1}^{N_p} \Delta \mathbf{x}_k^T Q \Delta \mathbf{x}_k + \sum_{k=1}^{N_u} \Delta \mathbf{u}_k^T R \Delta \mathbf{u}_k + \right. \\ & \left. \sum_{k=1}^{N_p} \mathbf{z}_k^T S \mathbf{z}_k \right) \Delta t + \Delta \mathbf{x}_{N_p+1}^T P \Delta \mathbf{x}_{N_p+1}, \end{aligned} \quad (3)$$

where Q and R are weight matrices for state and control respectively. N_p and N_u are the states and control prediction horizons. S is related to the addition of integral error states, represented by \mathbf{z} , and the last term is related to an end-state penalty term, as in [16]. The MPC optimisation programme to solve at each time step is given by:

$$\min_{\mathbf{X}_i, \mathbf{U}, \mathbf{Z}_i, \gamma} \gamma \quad (4)$$

$$s. t. \quad J_{end}(\mathbf{X}_i, \mathbf{U}, \mathbf{Z}_i) \leq \gamma \quad \forall i \in I \quad (5)$$

$$\mathbf{X}_i \in X, \mathbf{U} \in U \quad \forall i \in I \quad (6)$$

$$A_{ineq} [\mathbf{X}_i \ \mathbf{U}]^T \leq \mathbf{b}_{ineq} \quad \forall i \in I \quad (7)$$

$$A_{i,eq} [\mathbf{X}_i \ \mathbf{U}]^T = \mathbf{b}_{i,eq} \quad \forall i \in I \quad (8)$$

$$\Delta \mathbf{x}_{i, N_p+1}^T P_i \Delta \mathbf{x}_{i, N_p+1} \leq \alpha_p \quad \forall i \in I \quad (9)$$

$$\begin{aligned} \mathbf{z}_{i, k+1} &= \mathbf{z}_{i, k} + \Delta t K_i \Delta \mathbf{x}_{z, i, k} \quad \forall i \in I, k \in \\ & [0, N_p]. \end{aligned} \quad (10)$$

The problem solved here is a minimisation of a scalar $\gamma \in \mathbb{R}^+$ which constrains the cost evaluated at a set of perturbed states \mathbf{X}_i . This consists in robust approach to MPC, based on [17,18], where a control \mathbf{u} is computed so as to face different perturbed dynamic propagations at the same time. Perturbations consist in internal parameter variations $\Delta_{i,k}$ inducing different A_d and B_d . I is the set of perturbed scenarios cases, in this case an amount of three plus the nominal scenario. In this sense, (2) is reformulated as:

$$\Delta \mathbf{x}_{i, k+1} = A_d(\mathbf{x}_r, \mathbf{u}_r, \Delta_{i, k}) \Delta \mathbf{x}_{i, k} + B_d(\mathbf{x}_r, \mathbf{u}_r, \Delta_{i, k}) \Delta \mathbf{u}_k.$$

In (6) the bounds of states X and control U are imposed, concerning the maximum thermo-fluid-dynamic values and valve opening limits. Inequality (7) serves to establish the minimum and maximum mixture-ratio constraints, vital to avoid high temperatures or

extinction. In addition, actuators speed constraints are also included there. In (8), linear dynamic constraints between states and control are required. Inequality (9) ensures end-state reachability [16] and (10) describes integrator's dynamics [19], where a gain K_i is defined so as to reduce error in \mathbf{x}_z . And all the constraints (5)-(10) have to be verified for all i .

Start-up control: if the start-up transient is to be executed, another approach is proposed, in which a set of planned trajectories $(\mathbf{X}_t, \mathbf{U}_t)$ are tracked, and not only a final point. Trajectory tracking is more convenient in this case where the system evolves in a highly nonlinear way and where trajectories can be pre-computed without tight computational-time limits. In a quick throttling request, there might not be enough time to generate adequate trajectories (around one minute in *MATLAB*), but only an end reference (about 3s). Since this is also a continuous phase of the start-up, which starts after the opening of the GG-injection valves, is controlled. In this case, the NLSS f_s model is linearised about trajectories at each k , which reduces the prediction error:

$$\Delta \mathbf{x}_{i,k+1} = A_d(\mathbf{x}_{t,k}, \mathbf{u}_{t,k}, \Delta i_k) \Delta \mathbf{x}_{i,k} + B_d(\mathbf{x}_{t,k}, \mathbf{u}_{t,k}, \Delta i_k) \Delta \mathbf{u}_k.$$

The cost function (3) is modified so as to penalise deviations from trajectories:

$$J_{traj}(\mathbf{X}, \mathbf{U}, \mathbf{Z}) = \left(\sum_{k=1}^{N_p} (\mathbf{x}_k - \mathbf{x}_{t,k})^T Q (\mathbf{x}_k - \mathbf{x}_{t,k}) + \sum_{k=1}^{N_u} (\mathbf{u}_k - \mathbf{u}_{t,k})^T R (\mathbf{u}_k - \mathbf{u}_{t,k}) + \sum_{k=1}^{N_p} \mathbf{z}_k^T S \mathbf{z}_k \right) \Delta t + (\mathbf{x}_{N_p+1} - \mathbf{x}_{t,N_p+1})^T P (\mathbf{x}_{N_p+1} - \mathbf{x}_{t,N_p+1}).$$

The minimisation problem in this case presents the same robust structure as in (4)-(10), but now it is related to trajectory tracking. This implies the cost function is substituted by J_{traj} and that deviations from trajectories are considered instead of final references. This method is presented in detail in [12].

4.2.2 Hybrid MPC controllers

Discrete sub-phase of start-up: concerning the discrete sub-phase of the start-up transient, a computation logic is proposed here. The problem to solve in this hybrid scenario is more complex than in the purely continuous ones. Indeed, the discrete events, which consist in valve openings, ignitions and starter activation, alter the dynamic behaviour of the engine. During the sequence, each valve (CC and GG injection ones) is forced to remain closed until a certain instant, when it can start influencing the system via \mathbf{u} . Ignition and starter activations are modelled here as discrete inputs which activate a set of terms in differential equations. All these events are executed at some precise instants which are precisely tuned for nominal conditions.

In this paper a strategy for optimising the time differences between events is proposed. Furthermore, the sections of the valves which are already open are

also computed. The goal is to robustify the sequence a priori and to gain controllability of valves, which otherwise are simply opened to pre-defined degrees. The nominal order of events is maintained, since it consists in a safe succession of actuations according to fuel and oxidiser properties.

An optimisation problem based on MPC principles is again envisaged. An OL finite-horizon scheme, pre-computed off-line as for trajectory generation, seems more appropriate than an on-line CL one. In on-line conditions, little observability and controllability are present during this phase, as explained in Section 3. In addition, computational times tend to be long due to the necessary inclusion of nonlinear dynamic constraints. Indeed, in this frame, the effects of events can be expressed via constraints. But these constraints need to be nonlinear because they must include a dependency on the additional decision variables $\boldsymbol{\tau}$, which incorporate the optimal time differences between events. In the case of *Vulcain I*, it is:

$$\boldsymbol{\tau} = [\tau_{VCF} \tau_{VCO} \tau_{iCC} \tau_{iGG} \tau_{VGF}].$$

The definition of these intervals is graphically shown in Fig. 4.

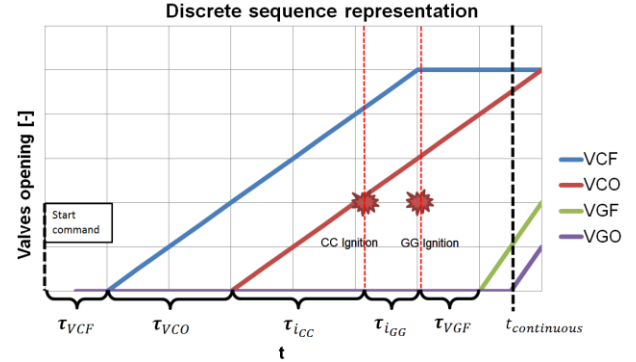


Fig. 4. Representation of considered time intervals during discrete sequence

The opening of the last valve (VGO) is considered at a fixed time, when the continuous phase starts. The goal is to attain a reference state $\mathbf{x}_{r,d}$ at that instant. Hence, a simple cost is used, only penalising the difference between the final step and that reference. The implicit dependencies on $\boldsymbol{\tau}$, expressed as nonlinear constraints, are built in the following way:

Algorithm 1. Definition of hybrid nonlinear dynamic constraints

Initialise \mathbf{X}_{NL} and \mathbf{U}_{off} as void matrices

for $k = 1, 2, \dots, N_p$ (11)

$t_k = t_0 + k\Delta t;$

$[\mathbf{u}_{k,off}, i_{CC}, i_{GG}, i_{sta}] = discrete_tree(\boldsymbol{\tau}, t_k);$

$\mathbf{x}_{k+1} = f_s(\mathbf{x}_{k+1}, \mathbf{u}_{k+1}, i_{CC}, i_{GG}, i_{sta})\Delta t + \mathbf{x}_k;$

Append \mathbf{x}_{k+1} to \mathbf{X}_{NL}

Append $\mathbf{u}_{k,off}$ to \mathbf{U}_{off}

end

The function *discrete_tree* provides the value of discrete inputs i according to the current time intervals and instant. If the current instant t_k surpasses the sum of the respective intervals, the corresponding event is activated. The vector \mathbf{U}_{off} reflects the additional constraints which have to be verified on control if valves are forced to be closed according to the timing. The system of equations f_s adapts according to the discrete inputs, and an implicit backward Euler scheme is proposed. In this manner, the changing dynamics are expressed via these implicit nonlinear constraints.

With these constraints, the following minimisation problem is to be solved, where increments are with respect to $(\mathbf{x}_{r,d}, \mathbf{u}_{r,d})$:

$$\min_{\mathbf{x}_d, \mathbf{u}_d, \boldsymbol{\tau}} \Delta \mathbf{x}_{Np+1}^T Q \Delta \mathbf{x}_{Np+1} \quad (12)$$

$$s. t. \quad \mathbf{X}_d \in X, \mathbf{U}_d \in U \quad (13)$$

$$A_{ineq,d}[\mathbf{U}_d] \leq \mathbf{b}_{ineq,d} \quad (14)$$

$$A_{eq,d}[\mathbf{X}_d \ \mathbf{U}_d]^T = \mathbf{b}_{eq,d} \quad (15)$$

$$\sum \boldsymbol{\tau} \leq t_{continuous} \quad (16)$$

$$|\mathbf{X}_d - \mathbf{X}_{NL}| = \mathbf{0} \quad (17)$$

$$\mathbf{U}_{off} = \mathbf{0} \quad (18)$$

Bounds are still considered in (13). In (14), in contrast to the previous (7), only actuators limits are constrained, since mixture ratios are not meaningful if some valves are closed. Equation (15) fixes the initial conditions of the problem, but not dynamics, which is constrained in (17) by running (11). The sum of all time differences must not exceed the beginning of the continuous phase $t_{continuous}$ (16). In (18) the corresponding valves are forced to be closed, depending on $\boldsymbol{\tau}$. The obtained $\mathbf{X}_d, \mathbf{U}_d$ and $\boldsymbol{\tau}$ would serve as reference trajectories for a robustified on-line tracking of the discrete sub-phase. The controller performing that tracking would have to take into account the same model-structure switches determined by the planned timing $\boldsymbol{\tau}$. That implies a varying number of available control inputs. Seeing the controllability and observability scarcities during this phase (mentioned in Section 3), only the tracking of \mathbf{U}_d trajectories would be conceivable. That is to say, only valve sections would be susceptible of being modified according to that reference, since not all the states in the system would be controllable nor observable. This approach remains to be numerically evaluated with extensive simulations.

5. Results and discussion

In order to solve the defined optimisation problems, the interior-point optimisation software *IPOPT* [20] has been employed within *MATLAB*. A general time step of $\Delta t = 10\text{ms}$ has been used in all cases, due to engine computer constraints. Computational times in *MATLAB* are about ten times longer than real time, not ruling out a future real-machine implementation.

5.1. Continuous control results

Regarding the results of continuous-control scenarios, different cases have been successfully simulated on the *Vulcain 1* engine. Firstly, continuous start-up control (trajectory tracking) results for different levels of desired end CC pressure (nominal 100%, 70% and 120%) are depicted in Fig. 5. The adjustable level of pressure or thrust at the end of the start-up is a required capability of new engines. Mixture ratios are kept constant to $MR_{CC,r} = 6, MR_{GG,r} = 1, MR_{PI,r} = 5.25$. Tracking is achieved with acceptable accuracy in p_{CC} for all cases (under 0.07% in nominal, under 0.95% in off-nominal) and in MR (under 0.32% in nominal, under 3.2% in off-nominal).

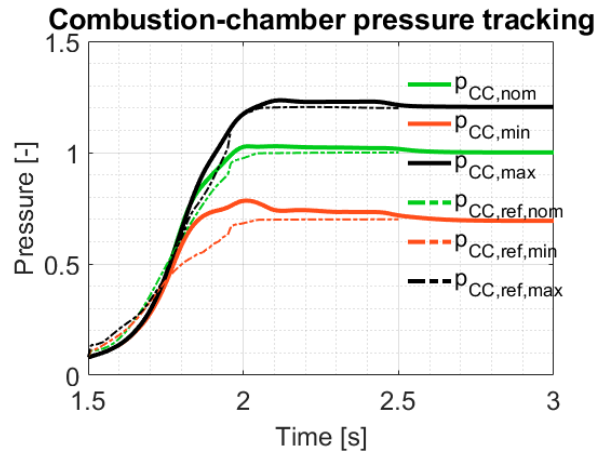


Fig. 5. Start-up control: p_{CC} tracking

At the same time, constraints are respected once mixture ratios become meaningful. Further performance indicators can be consulted in Table 2.

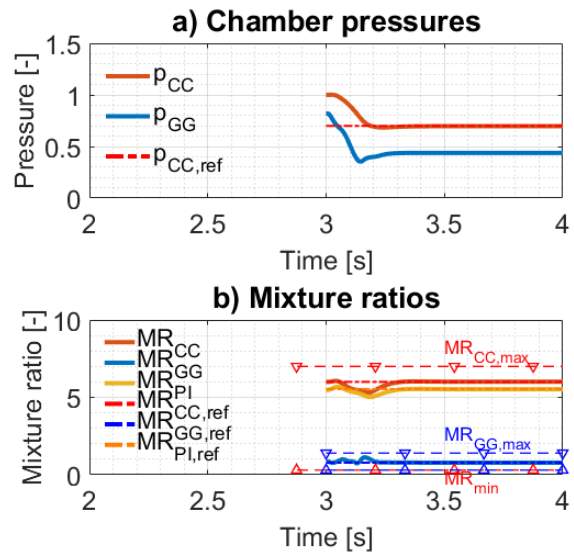


Fig. 6. Throttle-down CL scenario (100%-70% thrust), in terms of pressures (a) and mixture ratios (b)

When testing a throttling scenario, the resolution of (4)-(10) yields the results shown in Fig. 6. A throttle-down operation from 100% to 70% of thrust has been selected as a representative case. It is noticeable that the operation shift is safely performed (no constraints violated) within 0.34s.

As explained in Section 4, robustness considerations are included in the algorithms. Since some internal parameters can vary during operation, and others present uncertainty, the variations of the predominant parameters in Section 3 are also taken into account. Some robustness tests are presented in [12], in which variations of tanks pressure are well mitigated. The considered perturbed cases in that paper, are also considered here (+10% and -10% in p_{tanks}), since they are the most probable. The inclusion of the selected list of additional influential parameters in a third perturbed scenario also mitigates their variations in the simulator, as shown in the overall performance indicators summary in Table 2. That third scenario is referred to as the worst case, since it consists in the worst combination of those alterations. Table 2 collects results for start-up and throttling control in OL, CL and in perturbed scenarios.

Table 2. Control performance indicators results

Scenario	Nominal		Worst case		Nominal		Worst case	
	100% Start-up	Start-up	100% Start-up	Start-up	Throttling 100→70%	Throttling 100→70%	Throttling 100→70%	Throttling 100→70%
Indicator	OL	CL	OL	CL	OL	CL	OL	CL
Settling time ($p_{CC,r} \pm 1\%$) [s]	2.76	2.54	-	2.51	-	0.34	-	-
Over-shoot (% in p_{CC})	6.29	2.84	4.28	2.67	3.19	2.29	5.24	2.57
p_{CC} static error (%)	0.21	0.064	1.75	0.48	3.19	0.498	5.24	1.12
MR_{CC} static error (%)	0.18	0.32	1.25	0.44	2.72	0.18	4.61	0.82
MR_{GG} static error (%)	1.41	0.069	1.62	1.21	1.37	0.05	2.68	0.58
MR_{PI} static error (%)	1.41	0.022	2.97	0.056	2.98	0.29	4.9	1.38

Results point to a considerable performance improvement in all indicators with respect to the original OL, especially in perturbed scenarios. Robustness to parametric variations is noticeable in those cases.

6. Conclusions

The evolving design requirements of reusable launchers and their associated liquid-propellant rocket engines impose more demanding robustness requirements on control systems. Transient phases have traditionally been carried out in OL, which does not ensure robustness to possible parameter variations in the engine. These variations are more probable in reusable engines owing to their longer lives and executions. In this work, approaches to control the transient phases of a GG-cycle LPRE have been proposed. The main goal is to track combustion-chamber pressure and mixture ratios while respecting engine constraints. By making use of state-space models of these engines, strategies based on Model Predictive Control (MPC) have been developed for the different transient scenarios that the engine faces. An end-state-tracking algorithm for throttling operations, as well as a trajectory-tracking one for the start-up transient, have been synthesised with successful results. Robustness to the set of more influential parameters in the model is considered in the controller and demonstrated.

6.1. Perspectives

Several improvement areas can be mentioned. Full-state measurements are assumed perfect, which in reality has to be solved via mass-flow estimators. Discrete inputs to the system, involved in the start-up sequence, are also to be controlled in order to robustify that sequence. The aforementioned algorithm (11)-(18) is a proposal which mimics in a relatively precise way the physics of the complex problem which is controlling the discrete sub-phase of the LPRE start-up. Due to the numerous nonlinearities, coupling and implicitness of constraints, the optimisation problem becomes highly non-convex. The software *IPOPT* is not well suited for these computations. Thus, this approach remains to be numerically evaluated with extensive simulations. General non-convex optimisation solvers are under study, as well as other ways of expressing the problem. Furthermore, more realistic simulations, such as hardware-in-the-loop ones, are also to be carried out.

References

- [1] Le Gonidec S. Automatic & Control applications in the European space propulsion domain. From need expression to preparation for an uncertain future. ACD2016 Airbus Safran Launchers, Lille, France, 2016.
- [2] Baiocco P. and Bonnal C. Technology demonstration for reusable launchers. Acta Astronautica, 120:43-58, March 2016.
- [3] Nemeth E., Anderson R., Ols J., and Olsasky M. Reusable rocket engine intelligent control system framework design, phase 2. Technical Report NASA Contractor Report 187213, Rockwell

- International, Canoga Park, California, September 1991.
- [4] Dai X. and Ray A. Damage-Mitigating Control of a Reusable Rocket Engine: Part II-Formulation of an Optimal Policy. *Journal of Dynamic Systems, Measurement, and Control*, 118(3):409-415, September 1996.
- [5] Lorenzo C.F., Ray A., and Holmes M.S. Nonlinear control of a reusable rocket engine for life extension. *Journal of Propulsion and Power*, 17(5):998-1004, 2001.
- [6] Musgrave J.L., Guo T.H., Wong E., and Duyar A. Real-time accommodation of actuator faults on a reusable rocket engine. *IEEE transactions on control systems technology*, 5(1):100-109, 1996.
- [7] Saudemont R. and Le Gonidec S. Study of a robust control law based on H infinity for the Vulcain rocket engine - Etude d'une commande robuste a base de commande H infinity pour le moteur Vulcain. ArianeGroup internal report, ArianeGroup, ESTACA, Vernon, France, 2000.
- [8] Pérez-Roca S., Marzat J., Piet-Lahanier H., Langlois N., Farago F., Galeotta M., and Le Gonidec S. A survey of automatic control methods for liquid-propellant rocket engines. *Progress in Aerospace Sciences*, 107:63-84, May 2019.
- [9] Mayne D.Q., Rawlings J.B., Rao C.V., and Sckaert P.O.M. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789-814, June 2000.
- [10] Pérez-Roca S., Langlois N., Marzat J., Piet-Lahanier H., Galeotta M., Farago F., and Le Gonidec S. Derivation and Analysis of a State-Space Model for Transient Control of Liquid-Propellant Rocket Engines. In 2018 9th International Conference on Mechanical and Aerospace Engineering (ICMAE), pages 58-67, Budapest, Hungary, July 2018.
- [11] Pérez-Roca S., Marzat J., Flayac E., Piet-Lahanier H., Langlois N., Farago F., Galeotta M., and Le Gonidec S. An MPC Approach to Transient Control of Liquid-Propellant Rocket Engines. In 21st IFAC Symposium on Automatic Control in Aerospace - ACA 2019, Cranfield, UK, August 2019.
- [12] Pérez-Roca S., Marzat J., Piet-Lahanier H., Langlois N., Farago F., Galeotta M., and Le Gonidec S. Trajectory planning and tracking via MPC for transient control of liquid-propellant rocket engines. Accepted in 15th European Workshop on Advanced Control and Diagnosis, ACD 2019, to appear, Bologna, Italy, November 2019.
- [13] Sobol I.M., Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates, *Mathematics and Computers in Simulation*, vol. 55, no. 1, pp. 271–280, Feb. 2001.
- [14] Iooss B., Lemaitre P. A review on global sensitivity analysis methods. In *Uncertainty management in simulation-optimization of complex systems*, pages 101-122, Springer, 2015.
- [15] Betts J.T. Survey of Numerical Methods for Trajectory Optimization. *Journal of Guidance, Control, and Dynamics*, 21(2):193-207, 1998.
- [16] Chen H. and Allgoewer F. A Quasi-Infinite Horizon Nonlinear Model Predictive Control Scheme with Guaranteed Stability. *Automatica*, 34(10):1205-1217, 1998.
- [17] Calafiore G.C. and Fagiano L. Robust Model Predictive Control via Scenario Optimization. *IEEE Transactions on Automatic Control*, 58(1):219-224, January 2013.
- [18] Lofberg J. Minimax Approaches to Robust Model Predictive Control. Number No. 812 in *Linköping Studies in Science and Technology Dissertations*. UniTryck, Linköping University Electronic Press, April 2003.
- [19] Santos L.O., Afonso P.A.F.N.A., Castro J.A.A.M., Oliveira N.M.C., and Biegler L.T. On-line implementation of nonlinear MPC: an experimental case study. *Control Engineering Practice*, 9(8):847-857, August 2001.
- [20] Waechter A. and Biegler L.T. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25-57, March 2006.