Trajectory planning and tracking via MPC for transient control of liquid-propellant rocket engines

Sergio Pérez-Roca, Julien Marzat, Hélène Piet-Lahanier, Nicolas Langlois, Francois Farago, Marco Galeotta, Serge Le Gonidec

Trajectory planning and tracking via MPC for transient control of liquid-propellant rocket engines

Sergio Pérez-Roca[1,3], Julien Marzat[1], Hélène Piet-Lahanier[1], Nicolas Langlois[2], François Farago[3], Marco Galeotta[3], and Serge Le Gonidec[4]

[1] DTIS, ONERA, Université Paris-Saclay, Chemin de la Huniere, 91120 Palaiseau, France,
{sergio.perez_roca, julien.marzat, helene.piet-lahanier}@onera.fr,
[2] Normandie Université, UNIROUEN, ESIGELEC, IRSEEM, Rouen, France,
nicolas.langlois@esigelec.fr,
[3] CNES - Direction des Lanceurs, 52 Rue Jacques Hillairet, 75612 Paris, France,
{francois.farago, marco.galeotta}@cnes.fr,
[4] ArianeGroup SAS, Forêt de Vernon, 27208 Vernon, France,
serge.le-gonidec@ariane.group

Abstract. The new scenarios associated with launchers reusability force the enhancement of control algorithms for their liquid-propellant rocket engines. The transient phases of these engines are generally executed in open loop. The goal of this paper is to improve the control performance and robustness throughout the fully continuous phase of the start-up transient of a generic gas-generator cycle. The controller has to guarantee an accurate tracking in terms of combustion-chamber pressure and chambers mixture ratios, as well as to satisfy a set of hard operational constraints. The selected strategy comprises a nonlinear preprocessor and a linearised MPC (Model-Predictive Control) controller, making use of nonlinear state-space models of the engine. The former plans the reference trajectory of states and control, which is tracked by the latter. Control goals are attained with sufficient accuracy while verifying constraints within the desired throttling range. Robustness to internal parameters variations is considered in the MPC controller by means of an epigraph formulation of the *minimax* robust optimisation problem, where a finite set of parameter-variation scenarios is treated.

Keywords: Liquid-propellant rocket engines, model predictive and optimisation-based control, control of constrained systems, tracking, trajectory planning, robustness

1 Introduction

The current context of launcher vehicles design is strongly related to the reusability feature. From the automatic control perspective, this potential need for reusable liquid-propellant rocket engines (LPRE) is translated into stricter robustness requirements, mainly forced by their multi-restart and thrust-modulation

capabilities. These demanding requirements arise from the possible endogenous perturbations due to components faults or evolving parameters and from exogenous perturbations related to the more complex mission profiles envisaged by new launchers.

Multivariable-control studies of main-stage LPRE have attained a short throttling envelope (70%-120%) in test benches [7]. In flight, the control system is generally conceived to attain the nominal operating point. One of the design capabilities of the future European *PROMETHEUS* engine is to throttle down to 30% of thrust [1]. Hence, it seems necessary to enlarge the controlled operating domain. Tracking and robustness have to be maintained at those low throttle levels, where physical phenomena are more difficult to anticipate.

The main control problem in these multivariable systems is the tracking of setpoints in combustion-chamber pressure and mixture ratio, whose references stem from launcher needs. Control-valves opening angles are varied so as to adjust engine's operating point while respecting a series of constraints. Most of the control approaches in the literature make use of linearised models about operating points for developing steady-state controllers, commonly based on PID techniques (such as [14]). In general terms, initial MIMO (Multi Input Multi Output) systems are decoupled into dominant SISO (Single Input Single Output) subsystems. Other conventional closed-loop (CL) linear strategies considered multivariable state feedback [23]. Off-line optimisation studies have also been developed [5]. More advanced works identified in the literature, incorporating some nonlinear [9], hybrid [13] or robust [20] techniques, improve certain aspects of performance and robustness. Besides, regarding the treatment of component faults, some strategies for reconfiguration control have been proposed [13, 19].

To the best of our knowledge, there are no publications which deal with not only the steady state but also the demanding transient phases at the same level of performance and robustness [17]. Pre-defined sequences of engine operation (start-up and shutdown), are commonly performed in open loop with narrow adaptation margins. The first part of the start-up transient presents a succession of discrete events including valves openings and chambers ignitions. Once these commands have all been activated, the second part of the transient, which is completely continuous, takes place until the steady state is reached. The main reasons for carrying out open-loop (OL) control in the initial (discrete-event) phase, exposed in [14], are controllability and observability problems at very low mass flows. Transient control via valves becomes possible once all events have taken place. This observation has also been considered in this paper, where only the second part of the start-up transient, fully continuous, is controlled.

The main goal of this work is to control the start-up transient of a pump-fed LPRE. Concretely, accurate tracking of combustion pressure (related to thrust) and mixture ratio all along the transient is targeted. At the same time, a set of hard operational constraints has to be satisfied, mainly concerning mixture ratios, turbopumps rotational speeds and valves actuators angular velocities. The type of LPRE studied is a gas-generator-cycle engine, but the methodology is intended to be generally applicable to other cycles. As in [16], the control loop

presented in this paper is based on Model Predictive Control (MPC), which is accompanied by a preprocessor for reference-trajectory generation. The MPC method has been chosen as the most adequate for this type of complex systems with hard operational constraints, as introduced in the next sections. In fact, it is gaining popularity at academic and industrial levels and can take into account robustness [12] or hybrid aspects, which are relevant for future work on this topic. Regarding the generation of reference trajectories, the algorithm in this paper is also based on MPC, which is common in the literature [6, 21]. Other trajectory-planning methods have been proposed, especially in the field of robotic motion, but they are also mostly based on optimisation [2].
This paper is organised as follows. Section 2 serves as a recapitulation of modelling considerations. The state-space system considered in the subsequent sections is stated there. Section 3 explains the control strategy carried out, mainly based on MPC techniques. The main results are presented and analysed in Section 4. Finally, Section 5 serves as a conclusion.

2 Modelling

The modelling approach considered in this paper was first introduced in [15] and further explained and updated in [16]. Basically, a set of different models is used, characterised by diverse degrees of complexity according to their purposes. In order to represent the plant to be controlled, a simulator of the engine was first developed. In this simulator, constructed component-wise, the main thermo-fluid-dynamics of LPRE elements are modelled: mass, energy and momentum conservation equations. Besides, since the application considered in this paper, representative of the *Vulcain 1* engine, is based on a gas-generator (GG) cycle, turbopumps dynamics are also modelled via shaft mechanical conservation equations. In these cycles, the hot-gas flow required for spinning turbines comes from a GG, which is a small combustion chamber receiving a portion of the main propellant flow.
Fig. 1 depicts the main components of the engine and lists the main acronyms used. Actuators in this system are mainly five continuously-controllable valves (VCH, VCO, VGH, VGO and VGC). There are also two discrete actuators: one binary igniter (i_{CC}) and one binary starter (i_{GG}). Nevertheless, discrete inputs are deemed active in this paper since the goal is to manage the continuously-controlled part ot transients (up from $1.5s$ after start command). Valves opening angles (α), having a nonlinear and monotone relation to sections (A), control the flows to the main combustion chamber (VCH and VCO), to the GG (VGH, VGO), and to the oxidiser turbine (VGC). Their fully open position is 90°. The latter is the main contributor to determining mixture ratio (MR), defined as the quotient between oxidiser and fuel mass flow rates $MR = \dot{m}_{ox}/\dot{m}_{fu}$. This ratio is a key performance indicator in LPRE and is set at three levels: at an engine's global level (MR_{PI}), accounting for pumped propellants; in the combustion chamber (MR_{CC}) and in the GG (MR_{GG}).
The simulator was rewritten as a nonlinear state-space (NLSS) model by joining

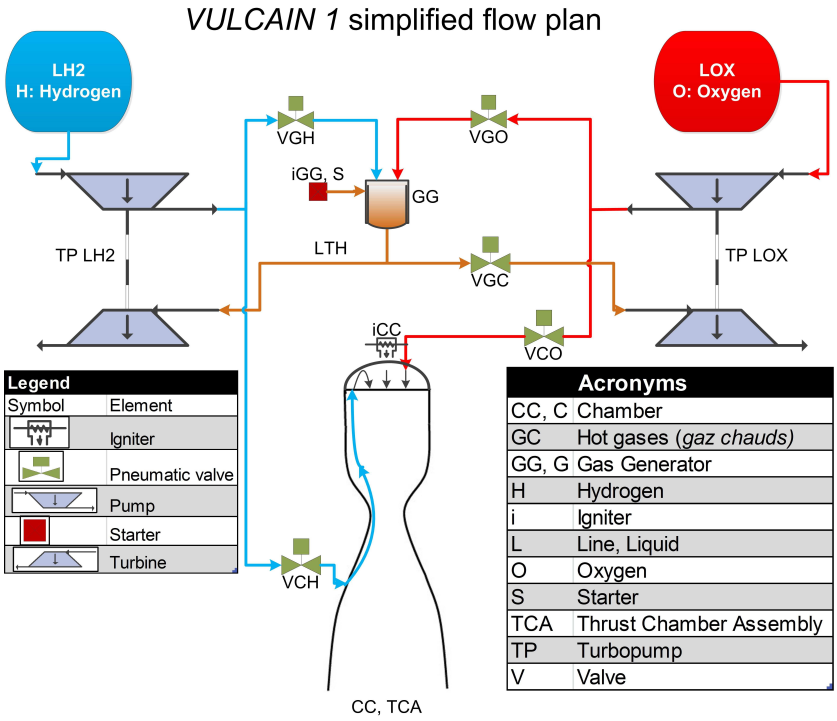

Fig. 1. *Vulcain 1* simplified flow plan with engine acronyms

components equations symbolically. As explained in [15, 16], throughout that translation several simplifications, comprising physical assumptions and mathematical rewriting, were carried out until obtaining the here-called *simplified NLSS* such that $\dot{\mathbf{x}} = f_s(\mathbf{x}, \mathbf{u})$.
The number of states is $n = 12$ and $m = 5$ is the number of control inputs. The state vector $\mathbf{x}$ contains the two turbopumps speeds ω_H and ω_O, the four pressures in the system (p_{CC} of combustion chamber, p_{GG} of the GG, p_{LTH} for hydrogen-turbine inlet cavity and p_{VGC} for oxygen-turbine inlet cavity) and six mass flows, including the ones streaming through control valves ($\dot{m}_{VCH}$, $\dot{m}_{VCO}$, $\dot{m}_{VGH}$, $\dot{m}_{VGO}$ and $\dot{m}_{VGC}$) and the one streaming through the hydrogen-turbine inlet pipe $\dot{m}_{LTH}$.

$$\mathbf{x} = [\omega_H \; \omega_O \; p_{CC} \; p_{GG} \; p_{LTH} \; p_{VGC} \; \dot{m}_{LTH} \; \dot{m}_{VCH} \; \dot{m}_{VCO} \; \dot{m}_{VGH} \; \dot{m}_{VGO} \; \dot{m}_{VGC}]^T. \quad (1)$$

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

$$\mathbf{x}_z = [p_{CC} \; \dot{m}_{VCH} \; \dot{m}_{VCO} \; \dot{m}_{VGH} \; \dot{m}_{VGO}]^T. \quad (2)$$

The control input $\mathbf{u}$ comprises the sections of the five control valves:

$$\mathbf{u} = [A_{VCH}\ A_{VCO}\ A_{VGH}\ A_{VGO}\ A_{VGC}]^T. \tag{3}$$

All equations, states and control have been rendered non-dimensional with respect to the nominal equilibrium values. The form of the dynamic system $\dot{\mathbf{x}} = f_s(\mathbf{x}, \mathbf{u})$ is the following, in which $a_i, b_i, ..., h_i \in \mathbb{R}^+$ are internal parameters and w_t is an exogenous input corresponding to starter mass flow:

$$\dot{x}_1 = -a_1 x_5 x_1 + b_1 x_5 + (-c_1 x_8 - d_1 x_{10})\, x_1 + e_1 {x_8}^2 + f_1 x_8 x_{10}, \tag{4}$$

$$\dot{x}_2 = -a_2 {x_2}^2 - b_2 x_6 x_2 + c_2 x_6 - (d_2 x_9 + e_2 x_{11})\, x_2 + f_2 {x_9}^2 + g_2 x_9 x_{11}, \tag{5}$$

$$\dot{x}_3 = a_3 x_8 + b_3 x_9 - c_3 x_3, \tag{6}$$

$$\dot{x}_4 = a_4 x_{10} + b_4 x_{11} - c_4\,(x_7 + x_{12}) + d_4 w_t, \tag{7}$$

$$\dot{x}_5 = a_5 x_7 - b_5 x_5, \tag{8}$$

$$\dot{x}_6 = a_6 x_{12} - b_6 x_6, \tag{9}$$

$$\dot{x}_7 = a_7\,(x_4 - x_5) - \frac{b_7 {x_7}^2}{x_4}, \tag{10}$$

$$\dot{x}_8 = \frac{\left(a_8 {x_1}^2 - b_8 {x_8}^2 - c_8 x_{10} x_8 - d_8 x_3 + e_8\right) {u_1}^2 - f_8 {x_8}^2}{(g_8 u_1 + h_8)\, u_1}, \tag{11}$$

$$\dot{x}_9 = \left(a_9 {x_2}^2 - b_9 x_9 x_2 - c_9 {x_9}^2 - d_9 x_3 + e_9\right) u_2 - \frac{f_9 {x_9}^2}{u_2}, \tag{12}$$

$$\dot{x}_{10} = \left(a_{10} {x_1}^2 - b_{10} {x_8}^2 - c_{10} x_{10} x_8 - d_{10} {x_{10}}^2 - e_{10} x_4 + f_{10}\right) u_3 - \frac{g_{10} {x_{10}}^2}{u_3}, \tag{13}$$

$$\dot{x}_{11} = \left(a_{11} {x_2}^2 - b_{11} x_9 x_2 - c_{11} {x_9}^2 - d_{11} {x_{11}}^2 - e_{11} x_4 + f_{11}\right) u_4 - \frac{g_{11} {x_{11}}^2}{u_4}, \tag{14}$$

$$\dot{x}_{12} = \left(x_4 - x_6 - \frac{a_{12} {x_{12}}^2}{x_4 {u_5}^2}\right) u_5. \tag{15}$$

It is noticeable that the system presents numerous nonlinearities and is non-affine with respect to control. Besides, full-state estimation is considered as perfect in this paper. This assumption is realistic for ω and p. Nevertheless, measuring some mass flows is generally inconvenient in practice. Hence, discussion on estimation will be relevant in future work.

3 Controller design

The synthesised controller is composed of two main blocks. First, a preprocessing block serves to generate state and control trajectories for arriving to a constant steady-state reference, $\mathbf{x}_r$, up from some given initial conditions. The main component is the MPC block, which receives those full-state and control reference trajectories and drives the system to them while meeting constraints. At the end

of the start-up transient, the state should track $\mathbf{x}_r$, with a particular interest in having a small tracking error in $\mathbf{x}_z$. Simultaneously, a set of hard constraints on $\mathbf{x}$ and $\mathbf{u}$ has to be satisfied throughout the transient. This second goal is major during these operation phases in order to avoid excessive temperatures, p or ω along the operation. The entire control loop is depicted in Fig.2.

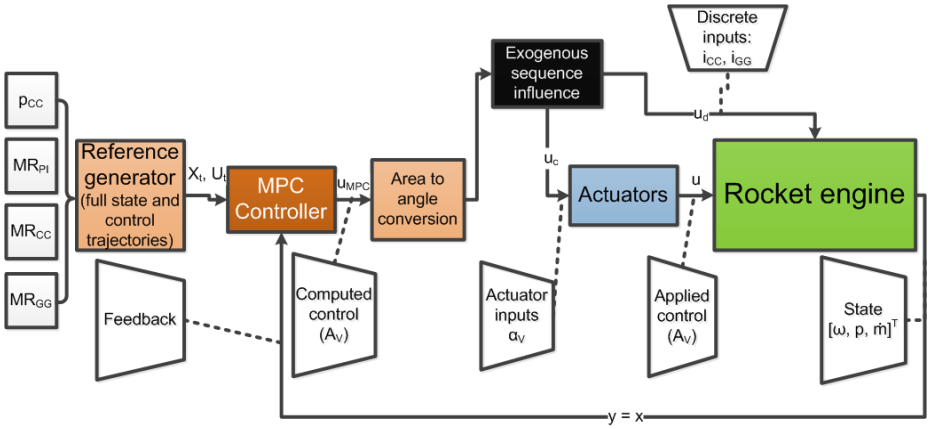

Fig. 2. Control-loop diagram

The remaining elements in that loop are the following. On the right-hand side the rocket engine is simulated (complex simulator), at an integration time step of $10^{-5}s$ in order to capture fast dynamics and to be robust to numerical stiffness. The inputs of that simulator and of the state-space model considered for control are valve sections $\mathbf{u}$. However, the actuators model (internal valve actuators) is expressed in terms of α. Consequently, a conversion block defining static and monotone nonlinear relations is required. The MPC controller provides valve sections that are then converted into angles. Valve actuators are considered as a separate element since they consist in an internal servo-loop, in which the angular position of the valve is regulated via a hydraulic or electrical actuator, modelled as a second-order system. This is a simplified modelling assumption since internal phenomena such as hysteresis and solid friction are not taken into account.

3.1 Preprocessor block: final-reference and trajectory generation

The preprocessing block serves as an off-line reference generator for the MPC controller. First of all, the end reference has to be constructed because the set of reference commands derived from launcher needs ($p_{CC,r}$, $MR_{PI,r}$, $MR_{CC,r}$ and $MR_{GG,r}$) is not sufficient to provide a full-state target equilibrium point $\mathbf{x}_r$ to the engine controller. This reference generation is presented in [16], in which

differential equations (4) to (15) are equated to zero. A steady-state reference in terms of control inputs $\mathbf{u}_r$ is also obtained in this calculation, since it is required for defining trajectories and for linearising the system.
Once these end targets ($\mathbf{x}_r$,$\mathbf{u}_r$) have been computed, a reference start-up trajectory ($\mathbf{X}_t$,$\mathbf{U}_t$) from some given $\mathbf{x}_0$ and $\mathbf{u}_0$ can be built. $\mathbf{X}$ 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):

$$\begin{aligned} \mathbf{X} &= [\mathbf{x}_1, ..., \mathbf{x}_k, ..., \mathbf{x}_{N_p}]^T, \\ \mathbf{U} &= [\mathbf{u}_1, ..., \mathbf{u}_k, ..., \mathbf{u}_{N_u}]^T. \end{aligned} \tag{16}$$

To do so, an optimisation- and model- predictive-based scheme is used. It can be regarded as an OL finite-horizon MPC scheme in which the prediction horizon is set to cover the duration of the start-up buildup transient, considered between $1.5s$, the end of the discrete sequential phase, and $2.5s$, the time when it is aimed at attaining the reference in the studied engine. This algorithm is based on the minimisation of a classical quadratic cost function J_{OL}, defined as:

$$J_{OL}(\mathbf{X}_t, \mathbf{U}_t) = \left(\sum_{k=1}^{N_{p,OL}} \mathbf{\Delta x}_k^T Q_{OL} \mathbf{\Delta x}_k + \sum_{k=1}^{N_{u,OL}} \mathbf{\Delta u}_k^T R_{OL} \mathbf{\Delta u}_k) \right) \Delta t, \tag{17}$$

where $\mathbf{\Delta x}_k = \mathbf{x}_k - \mathbf{x}_r$ and $\mathbf{\Delta u}_k = \mathbf{u}_k - \mathbf{u}_r$ are the variables to minimise, that is to say, the distances with respect to the reference equilibrium point. $Q_{OL} = I_n$ and R_{OL} are the weight matrices associated to states and control respectively. Diagonal terms in R_{OL} are set to 10^{10} so as to minimise control action. $N_{p,OL}$ and $N_{u,OL}$ are states and control prediction horizons, which in this case are taken equal to the horizon ($1s$) over the discretisation time $\Delta t = 10ms$, the latter selected to comply with controller's computational restrictions. Concerning the dynamics considered to predict the behaviour at each time step k, as announced in Section 2, the system is highly nonlinear. Hence, neglecting nonlinear dynamics can lead to relevant prediction errors, especially at points far from the equilibrium one. However, the main repercussion of imposing nonlinear constraints in optimisation problems is generally the loss of convexity of the optimised function and hence the increase in resolution complexity. The compromise chosen in this paper, related to the specific behaviour of the system, is the inclusion of nonlinear dynamic constraints until the system approaches its reference values to within 90%. This coincides approximately with the first half of the transient, where modelling errors of linearisation would be relevant if linear dynamics were used. Concretely, the aforementioned f_s is discretised via an Euler implicit scheme:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + f_s(\mathbf{x}_{k+1}, \mathbf{u}_{k+1})\Delta t, \quad k \in [1, N_{p,OL_{90\%}}]. \tag{18}$$

This scheme has been selected since it is the most stable among the first-order integration methods, required for lowering the complexity of the optimisation by reducing the interdependencies between decision variables. Once the pressure

p_{CC} (x_3 or $x_{z,1}$) attains its reference value, $\mathbf{x}_z$ states are forced to be equal to the end-reference values and only linear dynamics is imposed, in order to simplify the optimisation. A few time steps before the start of those end-state constraints, a smooth transition between nonlinear and linear dynamics is set via maximum slope constraints. Linear dynamics (A_c, B_c) stems from the linearisation of f_s about ($\mathbf{x}_r$, $\mathbf{u}_r$), which are then discretised via zero-order hold at Δt (A_d, B_d):

$$\mathbf{\Delta x}_{k+1} = A_d(\mathbf{x}_r, \mathbf{u}_r)\mathbf{\Delta x}_k + B_d(\mathbf{x}_r, \mathbf{u}_r)\mathbf{\Delta u}_k, \quad k \in [N_{p,OL_{90\%}} + 1, N_{p,OL} + 1]. \tag{19}$$

Having defined the different dynamics, the optimisation algorithm which is executed once for the whole horizon (OL trajectory planning) under constraints is the following:

$$\begin{aligned} \min_{\mathbf{X}_t, \mathbf{U}_t} \quad & J_{OL}(\mathbf{X}_t, \mathbf{U}_t) \\ \text{s.t.} \quad & \mathbf{X}_t \in X, \quad \mathbf{U}_t \in U \\ & A_{ineq}[\mathbf{X}_t \quad \mathbf{U}_t]^T \leq \mathbf{b}_{ineq} \\ & \mathbf{x}_{k+1} \leq \mathbf{x}_k + f_s(\mathbf{x}_{k+1}, \mathbf{u}_{k+1})\Delta t + \boldsymbol{\varepsilon}, \quad \forall k \in [1, N_{p,OL_{90\%}}] \\ & \mathbf{x}_{k+1} \geq \mathbf{x}_k + f_s(\mathbf{x}_{k+1}, \mathbf{u}_{k+1})\Delta t - \boldsymbol{\varepsilon}, \quad \forall k \in [1, N_{p,OL_{90\%}}] \\ & A_{eq}[\mathbf{\Delta X}_t \quad \mathbf{\Delta U}_t]^T = \mathbf{b}_{eq} \quad (\textit{including} \quad \mathbf{x}_{N_p+1} = \mathbf{x}_r). \end{aligned} \tag{20}$$

X and U are the allowable sets for states and control (compact subsets of $\mathbb{R}^{n(N_{p,OL}+1)}$ and $\mathbb{R}^{mN_{u,OL}}$ respectively). The set U for the first control $\mathbf{u}_1$ is specially constrained to comply with actuators capacity as in [10]:

$$\mathbf{u}_1 \in [\max(\underline{U}, \mathbf{u}_0 - \dot{\mathbf{u}}_{max}\Delta t), \min(\overline{U}, \mathbf{u}_0 + \dot{\mathbf{u}}_{max}\Delta t)], \tag{21}$$

where $\dot{\mathbf{u}}_{max}$ is the maximum sectional velocity of valves.
Regarding the rest of constraints, (20) contains linear inequality constraints (defined by A_{ineq} and $\mathbf{b}_{ineq}$), for satisfying MR and actuators sectional-velocity bounds, as well as for defining a monotonically increasing pressure buildup. Nonlinear dynamic constraints are not defined as strict equality constraints, but are treated as inequalities with a small margin $\boldsymbol{\varepsilon} = 10^{-2} \times \mathbf{1}_{n\times 1}$ (non-dimensional) so as to simplify the computation of a feasible optimal solution. Linear dynamics (19), initial conditions and end-state reaching are considered in the equality constraints (defined by A_{eq} and $\mathbf{b}_{eq}$). The end-state hard constraint forces the trajectory to precisely finish at the desired point. Fig. 3 depicts the set of generated trajectories for the different operating points (70%, 100%, 120% of thrust) in terms of combustion-chamber pressures. The interior-point optimisation software *IPOPT* [22] is used to solve (20). Due to the inclusion of nonlinear constraints in this quadratic-cost optimisation problem, the solution might not be global.

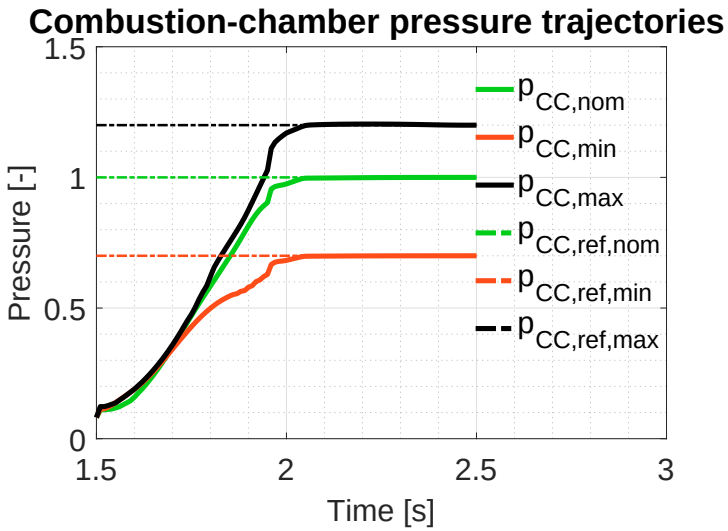

Fig. 3. Trajectory planning results in p_{CC} for $p_{CC,r} = 1$ (nominal), $p_{CC,r} = 0.7$ (minimum) and $p_{CC,r} = 1.2$ (maximum)

Fig. 4 presents the $\mathbf{u}$ trajectories for the nominal case.

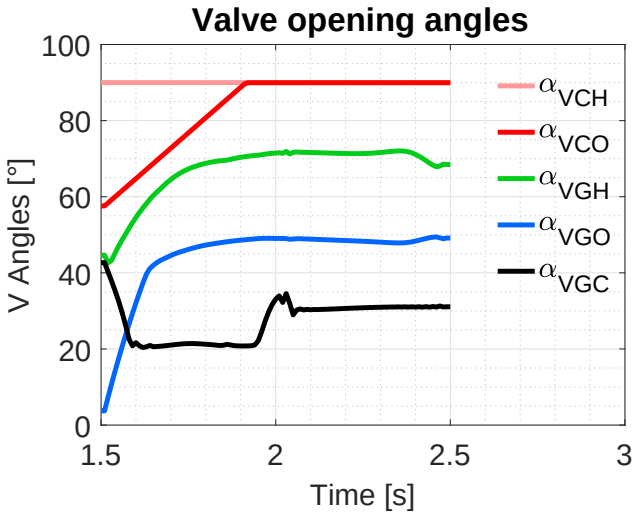

Fig. 4. Trajectory planning results in $\mathbf{u}$ for $p_{CC,r} = 1$ (nominal). Maximum opening angles are 90°.

Initial conditions are fixed according to the end of the discrete part of the transient (until $t = 1.5s$). The first control inputs (VCO and VCH), related to the main chamber, are pretty constrained in terms of operation, thereby imposing straight trajectories.

3.2 MPC algorithm for trajectory tracking

MPC predicts the future behaviour of the modelled system along a horizon, and optimises control inputs according to a cost function related to a reference trajectory or to an end state. In this paper, the strategy is to track the predefined trajectory ($\mathbf{X}_t$, $\mathbf{U}_t$). A similar algorithm, directly tracking an end state, was presented in [16]. However, tracking a trajectory enables a more precise control of transients, which is advantageous in this application. Dynamics is predicted in this controller in a linearised way, so as to enable a real-time control computation. The discrete-time matrices A_d and B_d are evaluated about each step in the trajectory, improving the prediction representativeness with respect to the use of single end-state matrices.

$$\mathbf{\Delta x}_{k+1} = A_d(\mathbf{x}_{t,k}, \mathbf{u}_{t,k})\mathbf{\Delta x}_k + B_d(\mathbf{x}_{t,k}, \mathbf{u}_{t,k})\mathbf{\Delta u}_k. \tag{22}$$

Thus, the goal of the controller is to drive $\mathbf{X}$ and $\mathbf{U}$ to $\mathbf{X}_t$ and $\mathbf{U}_t$. The matrix A_d is always stable along the trajectory, which is a particularity of these GG-cycle LPRE models. The approach selected here incorporates the quasi-infinite horizon (QIH) notions from [4]. That reference presents proofs for guaranteed stability and end-state reachability of MPC via the consideration of a terminal region and a fictitious local controller K acting at the end of the state prediction horizon, $N_p + 1$. The main role of that controller appears in the computation of the P matrix of a Lyapunov function $V(\mathbf{x}) = \mathbf{\Delta x}^T P \mathbf{\Delta x}$, by solving the following Lyapunov equation:

$$(A_K(\mathbf{x}_{t,N_p+1}, \mathbf{u}_{t,N_u}) + \kappa I)^T P + P(A_K(\mathbf{x}_{t,N_p+1}, \mathbf{u}_{t,N_u}) + \kappa I) = -Q_K - K^T R_K K. \tag{23}$$

In (23), the system considered is the composition of the end linear system with a simple LQR feedback controller, $A_K = A_c(\mathbf{x}_{t,N_p+1}, \mathbf{u}_{t,N_u}) + B_c(\mathbf{x}_{t,N_p+1}, \mathbf{u}_{t,N_u})K$, $\kappa \in \mathbb{R}^+$ (satisfying $\kappa < -\lambda_{max}(A_K)$) and Q_K and R_K are positive definite symmetric matrices $Q_K \in \mathbb{R}^{n\times n}$, $R_K \in \mathbb{R}^{m\times m}$. As explained in the aforementioned references, the first use of the computed $P \in \mathbb{R}^{n\times n}$ is to add a terminal-region penalty term in the MPC cost (24). It is used in constraints too, which are explained later. Besides, an integral action is added for increasing tracking precision on $\mathbf{x}_z$ (in a similar manner to [18]). Integrator decision variables are denoted by $\mathbf{z}$ (contained in $\mathbf{Z}$) and are also penalised in the cost through a weight matrix $S \in \mathbb{R}^{n_z\times n_z}$, whose diagonal is $[1, 0.1, 0.1, 0.1, 0.1]$. Thus, the MPC cost J to

minimise consists in:

$$J(\mathbf{X},\mathbf{U},\mathbf{Z}) = \left(\sum_{k=1}^{N_p} (\mathbf{x}_k - \mathbf{x}_{t,k})^T Q (\mathbf{x}_k - \mathbf{x}_{t,k}) + \sum_{k=1}^{N_u} (\mathbf{u}_k - \mathbf{u}_{t,k})^T R (\mathbf{u}_k - \mathbf{u}_{t,k}) + \sum_{k=1}^{N_p} \mathbf{z}_k^T S \mathbf{z}_k \right) \Delta t + (\mathbf{x}_{N_p+1} - \mathbf{x}_{t,N_p+1})^T P (\mathbf{x}_{N_p+1} - \mathbf{x}_{t,N_p+1}). \tag{24}$$

Basically, the integral and terminal costs are added to the traditional quadratic cost on states and controls, with a prediction horizon $N_p = 10$ steps ($0.1s$) and a control horizon $N_u = 5$. Implicitly, the last control $\mathbf{u}_{N_u}$ is used for $k \geq N_u$. Since these horizons are shorter than those used in trajectory generation (Section 3.1), the last step does not necessarily correspond to the end reference point. Longer horizons (more costly) did not enhance tracking or constraints satisfaction. Q and R are positive-definite symmetric weighting matrices $Q \in \mathbb{R}^{n \times n}$, $R \in \mathbb{R}^{m \times m}$, whose diagonals have been tuned off-line via Kriging-based black-box optimisation as in [11]. The criterion for that weight definition involves the minimisation of static error and overshoot in simulations.
Moreover, some robust considerations have been implemented. The minimisation of the previous J under constraints is not intrinsically robust. In fact, robustness to parameters and initial conditions variations and to modelling error is of interest in this application. In robust MPC approaches, the *minimax* optimisation, which minimises the worst-case scenario, is usually employed. A generic expression of this problem with uncertain dynamics is the following [8], in which Δ represents uncertainty in dynamic matrices $A_d(\cdot,\cdot,\Delta)$, $B_d(\cdot,\cdot,\Delta)$:

$$\begin{aligned} \min_{\mathbf{U}} \max_{\Delta} \quad & J(\mathbf{X},\mathbf{U}) \\ \text{s.t.} \quad & \mathbf{X} \in X \quad \forall \Delta \in \boldsymbol{\Delta}_c \\ & \mathbf{U} \in U \quad \forall \Delta \in \boldsymbol{\Delta}_c. \end{aligned} \tag{25}$$

Nevertheless, solving (25) for large compact uncertainty sets $\boldsymbol{\Delta}_c$ is too computationally costly for this application. Thus, only a finite set of uncertain scenarios are considered (inspired from [3]); and an equivalent formulation to the *minimax* in [8] is solved. Precisely, a scalar $\gamma \in \mathbb{R}^+$ is minimised via an epigraph formulation. That γ constrains the previous cost J defined in (24) evaluated at some perturbed states propagations $\mathbf{X}_i$:

$$\begin{aligned} \mathbf{X}_i &= [\mathbf{x}_{i,1}, ..., \mathbf{x}_{i,k}, ..., \mathbf{x}_{i,N_p+1}]^T, \quad i \in I \\ \boldsymbol{\Delta}\mathbf{x}_{i,k+1} &= A_d(\mathbf{x}_{t,k}, \mathbf{u}_{t,k}, \Delta_{i,k}) \boldsymbol{\Delta}\mathbf{x}_{i,k} + B_d(\mathbf{x}_{t,k}, \mathbf{u}_{t,k}, \Delta_{i,k}) \boldsymbol{\Delta}\mathbf{u}_k, \quad k \in [0, N_p + 1], \end{aligned} \tag{26}$$

where $\Delta_{i,k}$ are certain selected internal parameter variations belonging to $\boldsymbol{\Delta} = \{\Delta_{i,k}, i \in I, k \in [0, N_p + 1]\}$. I is a finite set which indexes the perturbed cases. The epigraph formulation shifts robustness considerations into the constraints

list, thereby not requiring maximisation. Only a smooth convex nonlinear programme (NLP) is solved. That minimisation problem, in which decision variables are extended to account for all $\mathbf{X}_i$, is stated below:

$$\begin{aligned} \min_{\mathbf{X}_i,\mathbf{U},\mathbf{Z}_i,\gamma} \quad & \gamma \\ \text{s.t.} \quad & J(\mathbf{X}_i,\mathbf{U},\mathbf{Z}_i) \leq \gamma \quad \forall i \in I \\ & \mathbf{X}_i \in X, \quad \mathbf{U} \in U \quad \forall i \in I \\ & A_{ineq}[\mathbf{X}_i \quad \mathbf{U}]^T \leq \mathbf{b}_{ineq} \quad \forall i \in I \\ & A_{i,eq}[\mathbf{X}_i \quad \mathbf{U}]^T = \mathbf{b}_{i,eq} \quad \forall i \in I \\ & (\mathbf{x}_{i,N_p+1} - \mathbf{x}_{t,N_p+1})^T P_i (\mathbf{x}_{i,N_p+1} - \mathbf{x}_{t,N_p+1}) \leq \alpha_P \quad \forall i \in I \\ & \mathbf{z}_{i,k+1} = \mathbf{z}_{i,k} + \Delta t K_I (\mathbf{x}_{z,i,k} - \mathbf{x}_{z,t,k}) \quad \forall i \in I, k \in [0, N_p]. \end{aligned} \tag{27}$$

As in Section 3.1, the set U for the first control $\mathbf{u}_{MPC} \equiv \mathbf{u}_1$ is specially constrained as in (21), but in this closed-loop case $\mathbf{u}_0$ is the previous-step control (warm start is performed). In (27) there are also equality constraints (defined by $A_{i,eq}$ and $\mathbf{b}_{i,eq}$) for the different linear dynamics along the whole horizon (26) and the same linear inequality constraints as in (20) applied to all $\mathbf{X}_i$. Concerning the terminal region, a constant α_P refers to the neighbourhood in which the Lyapunov term of J is constrained in a nonlinear way (further details on the QIH method in [4]). A specific P_i is used in each perturbed case since matrices are different. The series of Δ_i, where $I = \{1, 2, 3\}$, are taken constant for all k and represent the following parameter-variation cases:

- $i = 1$: nominal parameters.
- $i = 2$: the main varying parameters are increased by 10%: interdependent tanks pressures and temperatures.
- $i = 3$: the main varying parameters are decreased by 10%: interdependent tanks pressures and temperatures.

These cases consist in the worst ones which the controller must face in terms of tanks parameters variations, corresponding to the bounds of $\mathbf{\Delta}$. It can be checked in simulations that these would be the extreme dynamics-perturbing cases related to the compact uncertainty set $\mathbf{\Delta}_c$, for which a costly *minimax* would be necessary. Since the resulting $\mathbf{U}$ computed in (27) is forced to comply with all these uncertainty scenarios and all propagated perturbed states must verify all constraints, the robustness of the controller is increased. The last set of constraints in (27) consists in the integrator dynamics [18], where K_I is a gain matrix computed off-line in the same manner as Q and R.

4 Analysis of results

The interior-point optimisation software *IPOPT* [22] has been employed due to the smoothness and convexity of the NLP (27). Simulations of the aforementioned control loop within the *MATLAB* environment are executed from 1.5s

until $3s$ after the engine-start command, which is the time window in which continuous control is applied in the start-up transient.
Mixture-ratios initial values commence very far from the allowable area, owing to the low initial mass flows that alter the definition of quotients. In fact, chambers are still not physically ignited during the first instants (even if igniters have been activated). Hence, MR are not representative quantities at the very beginning of the start-up. Fig. 5 comprises the results of p_{CC} tracking at three non-dimensional pressure levels: $p_{CC,r} = 1$ (nominal), $p_{CC,r} = 0.7$ (minimum for this engine) and $p_{CC,r} = 1.2$ (maximum). At all three points, the final reference mixture ratios remain equal $MR_{CC,r} = 6$, $MR_{GG,r} = 1$ and $MR_{PI,r} = 5.25$, as usual during start transients. MR tracking in the nominal case is shown in Fig. 6 (with end-reference curves for enhancing visualisation).

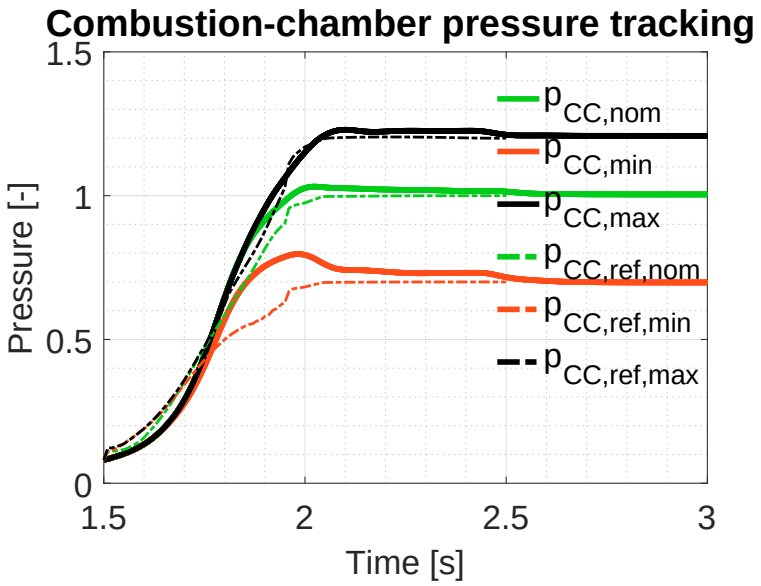

Fig. 5. Tracking results in p_{CC} for $p_{CC,r} = 1$ (nominal), $p_{CC,r} = 0.7$ (minimum) and $p_{CC,r} = 1.2$ (maximum)

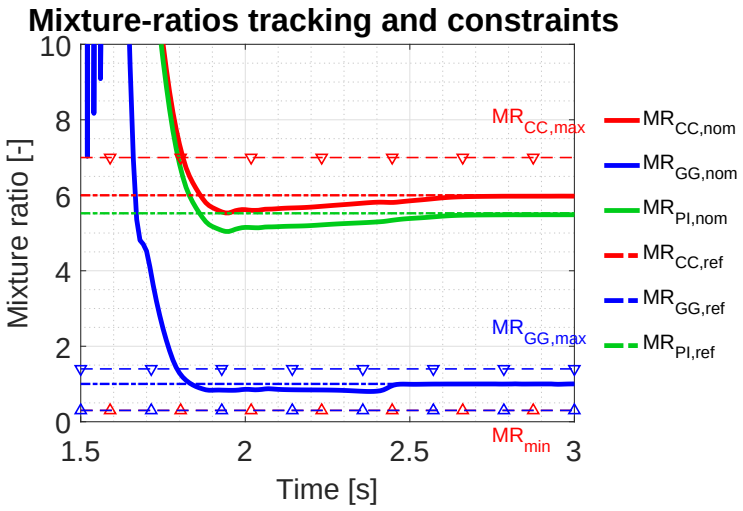

Fig. 6. Tracking results in MR for $p_{CC,r} = 1$ (nominal)

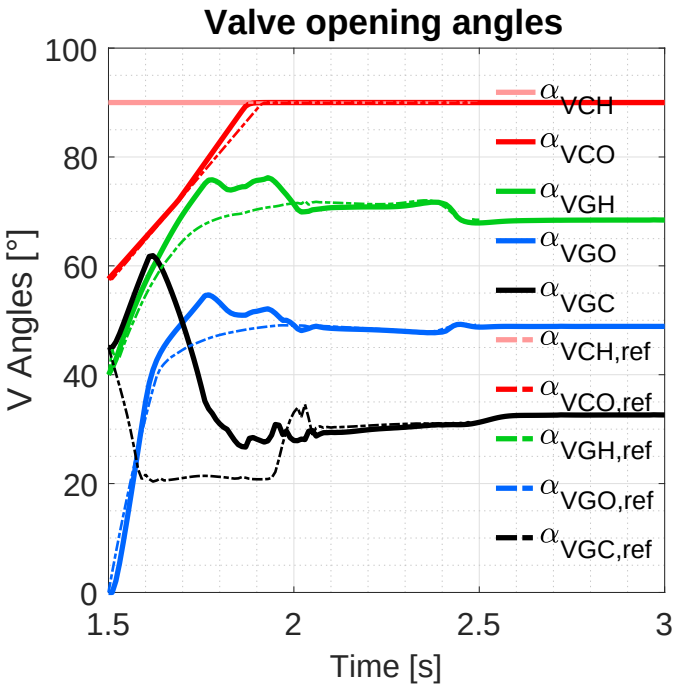

Fig. 7. Control results for $p_{CC,r} = 1$ (nominal)

Tracking is accomplished with acceptable accuracy in p_{CC} for all cases (under 0.12% in nominal, under 0.9% in off-nominal) and in MR (under 0.2% in nominal, under 3% in off-nominal). Simultaneously, constraints are respected up from the time when it is considered feasible and acceptable to respect them in practice ($1.9s$). The presence of a slight overshoot (2.6% in nominal) before the final convergence is generated by the nonlinear influence of the GG-starter input mass flow, which is not accounted for in the linearised model. It is more elevated in the minimum case since the relative impact of the starter is greater. Planned trajectories from Fig. 3 do not present overshoot in order to counteract this aspect of real engine behaviour. Fig. 7 shows the computed control as well as the planned trajectories. Some differences are present, especially during the first $0.5s$ in α_{VGC}, due to the more realistic initial mixture ratios in the simulator, which lie outside the bounds considered in trajectory generation. Even though a nonlinear model is used in the planning of those first instants, modelling errors are still relevant due to simplifications [15, 16].

The controller achieves that tracking performance even if random initial conditions (after the sequential transient phase) are set, while constraints-satisfaction times vary by some hundredths of seconds. Computational times in *MATLAB* are generally ten times longer than real time, not ruling out a future real-machine implementation.

4.1 Comparison with open loop and alternative MPC controller

Table 1 contains the comparison in terms of some performance indicators between this trajectory-tracking MPC (T.MPC) proposal, the end-state tracking (E.MPC, with different robustness considerations) presented in [16], and OL simulations.

Table 1. Performance-indicators comparison between this MPC proposal, end-state MPC [16] and OL at the three selected operating points

Operating point	**Nominal**			**Minimum**			**Maximum**		
Indicator	**OL**	**E.MPC**	**T.MPC**	**OL**	**E.MPC**	**T.MPC**	**OL**	**E.MPC**	**T.MPC**
Settling time ($p_{CC,r} \pm 1\%$) [s]	2.8	**2.51**	2.54	2.67	**2.55**	2.57	2.69	2.53	**2.51**
Overshoot (% in p_{CC})	6.31	5.04	**2.6**	15.1	11.46	**11.29**	3.34	4.04	**2.57**
Constraints verification [s]	1.81	**1.8**	1.82	1.83	**1.76**	1.78	**1.77**	1.81	1.84

The nominal OL corresponds to engine's original input command, which is precisely tuned for the standard case, as traditionally done in flight-ready engines. The minimum and maximum OL commands have been calculated via the preprocessor explained in Section 3.1, maintaining the same control as in the reference $\mathbf{u}_r$ along the transient. It seems clear that the strategy in this paper results in a higher level of performance than OL and that it reduces overshoot with respect to the alternative approach in [16]. Indeed, the real benefit of this CL MPC control is more noticeable at off-nominal points, where it is more problematic to accomplish multivariable tracking at high performance while meeting constraints during the transient. Furthermore, even if the nominal pressure and MR are targeted, the precomputed $\mathbf{u}_r$ would not always drive the system in OL to the same state $\mathbf{x}_r$. As explained in the next subsection, robustness is also an advantage.

4.2 Robustness analysis

In Section 3.2, the different perturbed cases have been listed. The system has to be robust to those scenarios in which some internal parameters can vary in real operation. These scenarios have been simulated by intentionally altering those parameters in the simulator. The performance indicators obtained with the different approaches for the nominal thrust are shown in Table 2.

Table 2. Performance-indicators comparison between this MPC proposal, end-state MPC [16] and OL at the perturbed scenarios

Perturbed cases	**Nominal**			**Increased by 10%**			**Decreased by 10%**		
Indicator	**OL**	**E.MPC**	**T.MPC**	**OL**	**E.MPC**	**T.MPC**	**OL**	**E.MPC**	**T.MPC**
Settling time ($p_{CC,r} \pm 1\%$) $[s]$	2.8	**2.51**	2.54	2.7	2.6	**2.55**	2.66	**2.52**	2.53
Overshoot (% in p_{CC})	6.31	5.04	**2.6**	5.72	4.79	**2.67**	5.3	5.45	**2.6**
Constraints verification $[s]$	1.81	**1.8**	1.82	**1.79**	**1.79**	1.82	**1.8**	**1.8**	1.82
p_{CC} static error (%)	0.25	0.26	**0.12**	0.35	0.36	**0.22**	0.64	0.17	**0.04**
MR_{CC} static error (%)	0.17	**0.01**	0.13	3.07	**0.19**	0.3	3.48	0.17	**0.05**
MR_{GG} static error (%)	1.39	**0.05**	0.12	0.6	**0.0008**	0.09	0.83	**0.1**	0.16
MR_{PI} static error (%)	1.43	0.3	**0.19**	3.24	0.12	**0.01**	3.65	0.48	**0.37**

Results point to the moderately greater robustness to parameters variations of the approach presented in this paper in comparison to [16]; and considerably greater with respect to OL. Comparing both CL approaches, the enhancements of T.MPC in terms of static errors and settling time are not elevated. The noticeable reduction of overshoot is the major advantage.

5 Conclusion

The control of the transient phases of liquid-propellant rocket engines is normally carried out in open loop due to its highly nonlinear and hybrid behaviour. In this paper, the control of the fully continuous part of the start-up of a gas-generator-cycle LPRE has been treated. Continuous valves are the considered actuators for regulating pressure in the main chamber and mass-flow mixture ratios. The tracking of these quantities, states in the model used, is accomplished via an MPC controller, which at the same time verifies a set of hard operational constraints. Concretely, what the controller tracks are pre-generated trajectories of states and control. These are computed off-line in a preprocessor which takes into account a full-state end reference constructed according to launcher commands, via a nonlinear state-space model of the engine. The linearised MPC controller with integral action achieves trajectory tracking with sufficient accuracy and constraints are satisfied during the transient. Robustness considerations are also included in the controller so as to face a set of internal-parameter variation scenarios, plausible in real engine operation. The computationally costly nested *minimax* optimisation, commonly used in robust MPC approaches, has been rewritten as the minimisation of a scalar cost. The controller demonstrates robustness against the predefined worst-case scenarios, since performance is not degraded. In future work, other ways of posing this robustness consideration globally, for instance via the estimation of perturbations, will be investigated. The use of discrete actuators in control may also be a relevant factor to consider. A more extensive validation study with respect to parametric variations will be carried out.

References

1. Baiocco P. and Bonnal C. Technology demonstration for reusable launchers. *Acta Astronautica*, 120:43–58, March 2016.
2. Betts J.T. Survey of Numerical Methods for Trajectory Optimization. *Journal of Guidance, Control, and Dynamics*, 21(2):193–207, 1998.
3. Calafiore G.C. and Fagiano L. Robust Model Predictive Control via Scenario Optimization. *IEEE Transactions on Automatic Control*, 58(1):219–224, January 2013.
4. Chen H. and Allgoewer F. A Quasi-Infinite Horizon Nonlinear Model Predictive Control Scheme with Guaranteed Stability. *Automatica*, 34(10):1205–1217, 1998.
5. 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. - NASA.

6. Kuwata Y., Schouwenaars T., Richards A., and How J. Robust Constrained Receding Horizon Control for Trajectory Planning. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, San Francisco, USA, August 2005. American Institute of Aeronautics and Astronautics.
7. 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.
8. Loefberg J. *Minimax Approaches to Robust Model Predictive Control*. Number No. 812 in Linkoeping Studies in Science and Technology Dissertations. UniTryck, Linkoeping University Electronic Press, April 2003. Google-Books-ID: 96VYBQAAQBAJ.
9. 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. - NASA.
10. Luo Y., Serrani A, Yurkovich S., Doman D.B., and Oppenheimer M.W. Model predictive dynamic control allocation with actuator dynamics. In *Proceedings of the 2004 American Control Conference*, volume 2, pages 1695–1700, Boston, USA, 2004.
11. Marzat J., Walter E., Piet-Lahanier H., and Damongeot F. Automatic tuning via Kriging-based optimization of methods for fault detection and isolation. In *2010 Conference on Control and Fault-Tolerant Systems (SysTol)*, pages 505–510, Nice, France, October 2010.
12. Mayne D.Q., Rawlings J.B., Rao C.V., and Scokaert P.O.M. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, June 2000.
13. 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. - NASA.
14. 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.
15. 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.
16. 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.
17. 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.
18. 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.
19. Sarotte C., Marzat J., Piet-Lahanier H., Iannetti A., Galeotta M., and Ordonneau G. Actuator Fault Tolerant System For Cryogenic Combustion Bench Cooling Circuit. In *10th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS 2018*, volume 51, pages 592–599, Warsaw, Poland, 2018. Elsevier. - ONERA/CNES.

20. 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.
21. Singh L. and Fuller J. Trajectory generation for a UAV in urban terrain, using nonlinear MPC. In *Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148)*, volume 3, pages 2301–2308 vol.3, Arlington, VA, USA, June 2001.
22. 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.
23. Zhang Y.L. State-space analysis of the dynamic characteristics of a variable thrust liquid propellant rocket engine. *Acta Astronautica*, 11(7):535–541, July 1984.