Model-based robust transient control of reusable liquid-propellant rocket engines

Sergio Pérez Roca

université
PARIS-SACLAY

Model-based Robust Transient Control of Reusable Liquid-Propellant Rocket Engines

Thèse de doctorat de l'Université Paris-Saclay

École doctorale n° 580, Sciences et technologies de l'information et de la communication (STIC)
Spécialité de doctorat : Automatique
Unité de recherche : Université Paris-Saclay, ONERA, Traitement de l'information et systèmes, 91123, Palaiseau, France.
Référent : Faculté des Sciences d'Orsay

Thèse présentée et soutenue à Palaiseau, France, le 27/01/2020, par

Sergio PÉREZ ROCA

Composition du jury:

Cristina STOICA MANIU Professeure, CentraleSupélec (L2S)	Présidente
Michael OSCHWALD Professeur, DLR (Institut für Raumfahrtantriebe)	Rapporteur & Examinateur
Sami OTHMAN Maître de conférences, HDR, Université Lyon 1 (LAGEPP)	Rapporteur & Examinateur
Jean-Pierre CORRIOU Professeur Emérite, Université de Lorraine (ENSIC)	Examinateur
Mircea LAZAR Professeur associé, Dr., TU Eindhoven	Examinateur
Hélène PIET-LAHANIER Directrice de Recherche, ONERA (DTIS)	Directrice de thèse
Nicolas LANGLOIS Professeur, ESIGELEC (IRSEEM)	Co-directeur de thèse
Julien MARZAT Ingénieur de Recherche, HDR, ONERA (DTIS)	Co-encadrant & Examinateur
François FARAGO Ingénieur, Dr., CNES (DLA)	Invité, Co-encadrant
Marco GALEOTTA Ingénieur, CNES (DLA)	Invité, Co-encadrant
Serge LE GONIDEC Ingénieur, ArianeGroup SAS	Invité, Co-encadrant

Thèse de doctorat

NNT: 2020UPASS017

Funders:

Supervisors:

A Marina, a mi padre, a ma mare i a Melanie,

⋈

"Perquè hi haurà un dia que no podrem més, i llavors ho podrem tot",
Vicent Andrés Estellés.

⋈

"I got soul, but I'm not a soldier",
from *All These Things That I've Done* by The Killers.

Remerciements/Acknowledgements

Cette thèse n'aurait pu aboutir sans la collaboration et le soutien de plusieurs personnes de différents établissements et environnements.

Je voudrais tout d'abord remercier mes rapporteurs, Sami Othman et Michael Oschwald pour les constructifs rapports qu'ils ont écrit. Ich bedanke mich bei Prof. Oschwald für die qualifizierten Kommentare und ich freue mich, dass er meiner Doktorarbeit Relevanz beimisst. Je remercie aussi mes examinateurs, les Professeurs Cristina Stoica Maniu et Jean-Pierre Corriou de leurs intéressantes questions, and especially Mircea Lazar, who has made a great effort to carefully read my dissertation and to provide very constructive feedback, thereby helping to increase the quality of the final text.

Parmi ma "nombreuse" équipe d'encadrants, je remercie d'abord Elisa Cliquet Moreno, qui avait commencé à encadrer cette thèse avant son aventure japonaise; Serge Le Gonidec d'avoir lancé cette thèse; Nicolas Langlois de ses conseils et de m'avoir donné la possibilité d'aller à l'ESIGELEC-IRSEEM Rouen, où j'ai pu avoir des discussions multiculturelles et motivantes avec des doctorants et ingénieurs du même domaine; François Farago de sa gentillesse et sa prédisposition à m'écouter et m'aider quand j'en avais besoin; Hélène Piet-Lahanier de ses conseils scientifiques, de sa bonne humeur et de son aide avec les soucis administratifs qui ont apparu; Marco Galeotta de son aide immense pendant la phase de modélisation moteur, où j'ai pu profiter de sa grande expérience et savoir-faire, à part de sa bonne humeur et gentillesse (Grazie mille!); et Julien Marzat, qui a suivi mes pas de plus près, m'a proportionné des outils et m'a aidé et rassuré dans les prises de décisions.

Des personnes qui, n'étant pas encadrantes, ont aidé avec de petits gestes à faire des différents établissements liés à la thèse des endroits plus humains, sont Florence Marie (ONERA, probablement l'assistante la plus efficace et sympa du monde), Claire Quillien (ONERA), Bruno de la cafétéria (ONERA), Anne-Marie Cerrato (CNES), Alex Le Page (CNES), Bruno Vieille (CNES), Thérèse Toro (UPSud), Corinne Le Hong (ONERA), Virginie Wiels (ONERA), Gilles Duc (Supélec, pour sa gentillesse et son énorme travail en tête du pôle Automatique de l'école doctorale), Gérard Ordonneau (ONERA), Valérie Haemmerle et les autres ingénieurs de l'unité NGPA de l'ONERA-DTIS (et nos afterworks marrants), dont j'ai des remerciements spéciaux pour mon dernier co-bureau Prince Edorh, et pour Ioannis Sarras, qui s'est toujours intéressé à comment j'avançais et m'a fait plein de suggestions (même si enfin je n'ai pas pu toutes les appliquer).

On arrive à la partie la plus personnelle. Cette thèse a entraîné une montagne russe émotionnelle. Des questions sans réponse, des chemins barrés... En plus du grand challenge intellectuel qu'elle signifie, elle a entraîné aussi un grand challenge psychologique. Je me suis senti isolé, hybride (ni

d'ici ni de là-bas), parfois démotivé, mais j'ai réussi à trouver la motivation, grâce en partie à ma passion pour les moteurs fusée et les systèmes fluides, mais surtout grâce à toutes les personnes avec lesquelles j'ai partagé ces trois ans. Je remercie à tout le groupe de stagiaires et doctorants du DTIS, qui ont été ma famille en région parisienne en partageant beaucoup de soirées bars et jeux : Rodolphe Dubois, Vincent Chabridon, Iréna Robinson, Romain Rincé, Ric(c)ardo Bonalli, Jean-Lynce Gnanago, Denis de Oliveira, Julius Ibenthal, Antonello Venturino, Guillaume Hardouin, Baptiste Levasseur, Marcela Carvalho, Arnaud Denjoy, Antonella Florio, Thomas Perrin, Étienne Bertin (et ses blagues surréalistes), Nathan↔Michel (je vous assure que je l'ai vu), Camille Palmier (et sa force de cohésion du groupe), Enzo Iglésis (et ses faiblesses au baby-foot), Ali Hebbal (et notre passion commune pour Rafa Nadal), Julien Pelamatti (et son humeur incisive) y Esteban Restrepo Ochoa (amigo de navette y más chimba ;)). Notre bande de rock, faussement appelée "Les stagiaires d'Émilien", m'a beaucoup amusé et m'a fait découvrir des habilités cachées.

J'ai des remerciements particuliers pour deux déjà Docteurs qui m'ont poussé à finir cette thèse. Lorsque je pensais que cette thèse pouvait avec moi, là il y avait mon ancienne co-bureau Camille Sarotte (ou Sarette), pour m'encourager et me rassurer tout le temps. Un grand merci à elle pour son aide mathématique au tableau et de tous genres (même pour déplacer un lave-linge). Et là on arrive à Émilien Flayac. Quoi dire de ce personnage ? Mis à part son humeur presque offensive et ses douteuses connaissances d'espagnol, il m'a illuminé avec sa sagesse mathématique. En passant à mon bureau beaucoup de matins, il s'est petit à petit intéressé à ma thèse et a su poser les bonnes questions, ce qui lui a permis de me proposer des pistes de travail très pertinentes. On a même été co-auteurs d'un papier de conférence. Je lui dois beaucoup. En fait, le fonctionnement de la recherche devrait plus ressembler à ce que ce monsieur fait : échanger avec les autres de façon désintéressée et juste pour l'intérêt de la science.

Finalmente, quiero agradecer a los míos por el apoyo continuo durante estos tres años: a mis amigos Toni, Joaquín, Marcos, Luis, Álvaro, Edu, Loles, los Pablos, Juan, Juanpa, Jérôme, Louis, a les meues amigues Núria i Mayrén, bei der Familie Lipp, a mi familia Pérez i a la meua família Roca.

A mi padre Chema le agradezco su cariño y sus muestras de ánimo continuas (como "¡eres un crack!"). Y por último doy las gracias a las tres mujeres de mi vida y de mi corazón. A la meua germaneta Marina, por animarme siempre e incluso acompañarme a conferencias en otros continentes. Herzlichen Dank an Melanie, die trotz des Abstandes immer da war, wenn ich Aufmunterung und Beratung gebraucht habe. Jetzt beginnt eine neue Phase unserer Leben. Y a mi madre Carmen, porque es probablemente la mayor culpable de que haya llegado hasta aquí, habiendo puesto un foco enorme en mi educación. Gràcies Mare, por todo lo que me has dado y sigues dando.

Palaiseau, janvier 2020.

Résumé substantiel en français

La tendance actuelle vers un accès plus abordable à l'espace se matérialise généralement par des lanceurs et moteurs réutilisables. Du point de vue de la commande, ces moteurs fusée à propergol liquide (MFPL) réutilisables impliquent des spécifications de robustesse plus exigeantes que ceux à usage unique, principalement en raison de leurs capacités de redémarrage multiple et de modulation de poussée. Classiquement, le système de commande gère les opérations des MFPL autour d'un ensemble fini de points prédéfinis. Cette approche réduit leur domaine de modulation à un intervalle restreint dans lequel ils sont conçus pour être sûrs dans des conditions nominales. De plus, les phases transitoires, qui ont un impact important sur la durée de vie du moteur, ne sont pas exécutées de manière robuste.

Dans l'état de l'art de la commande des MFPL, la commande en boucle fermée a été réalisée autour de leurs états d'équilibre. Concrètement, des approches robustes face aux incertitudes et à certains scénarios défectueux sont présentes dans la littérature. La mitigation de l'endommagement de certains composants a également été considérée. Concernant les phases transitoires, seules des approches en boucle ouverte non robustes ont été développées. Un dénominateur commun de la plupart de ces approches est l'utilisation de modèles linéarisés du moteur sur leur point de consigne nominal. Cependant, ces stratégies de commande n'ont traité les exigeantes phases transitoires en boucle fermée ni permis la modulation continue dans un domaine de fonctionnement élargi.

Cette thèse a consisté à développer une boucle de régulation adaptée à l'ensemble des phases d'opération des MFPL à cycle de générateur de gaz (transitoire et régime permanent) et robuste aux variations paramétriques internes. Plusieurs blocs ont été développés pour constituer la boucle de régulation: simulation de moteur, génération de référence et plusieurs contrôleurs. Des simulateurs représentatifs du comportement thermo-fluido-dynamique des moteurs à cycle de générateur de gaz (GG) ont tout d'abord été construits. La raison du choix de ce type de cycle est que le moteur Européen réutilisable en développement, *PROMETHEUS*, sera construit sur cette base. La principale étude de cas pour la modélisation et le contrôle dans cette thèse a été le moteur *Vulcain 1*, tandis que *PROMETHEUS* a seulement été modélisé. Des modèles 0-D de chaque composant moteur principal ont été sélectionnés parmi la littérature ou développés. Ces sous-modèles ont ensuite été assemblés pour construire le simulateur transitoire basé sur des équations différentielles ordinaires (ODE), capables de capturer les phases transitoires, y compris le démarrage complet.

La modélisation purement thermodynamique du cycle a ensuite été adaptée au cadre de la commande dans le but d'appliquer des méthodes à base de modèle. Des modèles non-linéaires sous forme d'état des mêmes moteurs ont été dérivés. Pour ce faire, les modèles de composants définis

ont été symboliquement joints. C'est-à-dire que toutes les équations différentielles et algébriques-thermodynamiques ont été combinées causalement selon le circuit de flux du moteur. La manière dont les caractéristiques hybrides du système ont été traitées, impliquant des actionneurs continus et discrets, a été choisie selon la littérature de modélisation des systèmes hybrides. Les éléments discrets, concernant les actionneurs d'allumage et de démarrage, sont considérés comme faisant partie des entrées de commande et gouvernent certains termes dans les équations différentielles. Le reste des entrées de commande, de nature continue, sont les sections d'ouverture des vannes. Les variables d'état sont les vitesses de rotation des arbres, les pressions dans les cavités et les débits massiques à travers les vannes et tuyaux. Les mesures de l'ensemble du vecteur d'état sont supposées disponibles. L'adéquation de ces modèles transitoires à la synthèse des lois de commande, étant non-linéaires et non-affines par rapport aux entrées de commande, a ensuite été analysée. Des simplifications physiques et mathématiques ont dû être effectuées afin d'obtenir un modèle exploitable, qui peut également être linéarisé.

Ces modèles dérivés ont ensuite été utilisés pour synthétiser des contrôleurs. Leur forme non-linéaire a servi à générer des trajectoires de référence hors ligne en fonction de la poussée finale souhaitée et des rapports de mélange. Les objectifs de commande en ligne concernent non seulement le suivi multivariable, mais aussi la vérification des contraintes strictes et la robustesse aux variations paramétriques à un coût de calcul raisonnable. Constatant la complexité des modèles et afin de respecter tous ces objectifs, la commande prédictive (MPC) a été choisie comme méthode la plus appropriée. Deux approches de suivi principales via MPC avec retour d'état ont été proposées dans cette thèse pour les phases transitoires entièrement continues. Ces phases sont les opérations de modulation de poussée et la sous-phase continue du démarrage, qui a lieu une fois que tous les événements discrets qui la déterminent (les allumages et les ouvertures de vannes) se sont produits. Le suivi d'un état terminal pré-calculé est plus approprié pour les transitoires de modulation, où la référence peut varier rapidement. La sous-phase continue du transitoire de démarrage est contrôlée pour suivre les trajectoires de référence prédéfinies, qui sont plus coûteuses à calculer. Le calcul de la trajectoire est également effectué via une optimisation basée modèle.

La prise en compte en ligne de la dynamique non-linéaire a été explorée dans le développement de ces contrôleurs MPC. Cependant, la linéarisation des trajectoires s'est avérée plus adéquate pour la résolution en ligne du problème de contrôle optimal. Le suivi des points de fonctionnement en pression dans la chambre de combustion (liée à la poussée) et en rapport de mélange à l'intérieur du domaine de design est accompli en simulation tout en respectant les contraintes. La robustesse aux variations des paramètres, qui ont été identifiés comme prédominants selon les analyses de sensibilité, est achevée via un algorithme MPC robuste à un ensemble de scénarios prédéfinis. Concrètement, une formulation épigraphe du problème de MPC *minimax* est résolue simultanément pour tous les cas les plus significatifs de variations paramétriques internes. Des actions intégrales et des pénalités de région terminale sont aussi incluses pour maintenir/améliorer la performance en suivi. Une stratégie de gestion des entrées discrètes pendant la sous-phase discrète du démarrage a également été proposée, dans laquelle les différences de temps entre les événements d'activation pourraient être optimisées en tenant compte de l'évolution de la dynamique non-linéaire. Ce travail ouvre la voie à la validation expérimentale par des simulations hardware-in-the-loop ou des tests sur banc d'essai.

Contents

Nomenclature

α	Valve opening angle [°]
α_P	Auxiliary variable in QIH for terminal region constraint
$\boldsymbol{\tau}$	Time differences between events [s]
$\boldsymbol{\varepsilon}$	Margin for nonlinear dynamic constraints in trajectory planning
Δ	Uncertainty in dynamic matrices
$\dot{m}$	Mass flow [kg/s]
$\dot{w}$	Mass variation rate per unit of volume [$kg/s/m^3$]
η	Efficiency [-]
γ	Specific heat ratio [-] // Minimised scalar in epigraph scenario-based *minimax*
κ	Auxiliary variable in QIH
λ_{max}	Largest eigenvalue
$\mathbb{W}$	Perturbation set
$\mathbf{\Delta}_c$	Compact uncertainty set
$\mathbf{\Delta}_I$	Scenario-based non-compact uncertainty set
$\mathbf{R}_1$	Remainder of TAYLOR series
$\mathbf{U}$	Control-inputs trajectory
$\mathbf{X}$	State trajectory
μ	Mass fraction [-]
ν	Number of moles [$moles$]
Ω	Terminal-region set
ω	Rotational speed [rad/s] // Frequency $[rad/s]$
Φ	Heat transfer [W]
π	Pressure ratio [-]
ψ	Pressure-rise coefficient [m^2]
ρ, Rho	Density [kg/m^3]
$\tilde{}$	Overscript for dimensional variable
φ	Reduced mass flow coefficient [m^3]
$\varphi_{P,lim}$	Transition mass-flow coefficient [m^3]
$\vec{\mathbf{n}}$	Unit vector normal to surface
$\vec{\mathbf{v}}$	Flow velocity vector
ζ	Pressure-drop coefficient [-]
$_0$	Subscript for constant parameter // initial
$_{amb}$	Subscript for ambient
$_{cav}$	Subscript for pre-turbine cavity

c	Subscript for cavity/combustion/chamber // continuous // complex
d	Subscript for discrete
end	Subscript for E.MPC
eq	Subscript for equality
e	Subscript for nozzle exit
fp	Subscript for function of parameters
fu	Subscript for control-inputs dependent
fx	Subscript for state dependent
f	Subscript for formation
g	Subscript for gas
$ineq$	Subscript for inequality
inj	Subscript for injected
in	Subscript for inlet conditions
nom	Subscript for nominal
p,i	Subscript for combustion product i
pri	Subscript for primary
p	Subscript for varying parameter
r	Subscript for reference set point
sec	Subscript for secondary
sim	Subscript for simulator
sta	Subscript for starter
st	Subscript for stoichiometric
s	Subscript for simplified
th	Subscript for throat
$traj$	Subscript for T.MPC
t	Subscript for trajectory // time-dependent
vap	Subscript for vaporisation
z	Subscript for states with greater tracking relevance
A	Area // Valve section [m^2]
A, B, C, D	Standard state-space matrices
a_i	Butterfly-valve cross-section correlation coefficients
$A_{T,out}$, $AthLE$	Turbine outlet area [m^2]
B_w	State-space matrix epxressing influence of exogenous input
C	Torque coefficient [m^5]
C^*	Characteristic exhaust speed [m/s]
C_F	Thrust coefficient [-]
C_p	Specific heat at constant pressure [$J/K/kg$]
C_v	Specific heat at constant volume [$J/K/kg$]
$CoeCplP$	Pump torque polynomial coefficients
$CoeCplT$, $a_{i,T}$	Turbine torque correlation coefficients
$CoePrsP$	Pump outlet-pressure polynomial coefficients
$comp$	Chemical composition vector [1×6]
D	Variance in SOBOL analysis
e	Energy [J]

err	Steady-state (static) error
F	Thrust [N] // Stage cost
$f(\cdot)$	Nonlinear dynamic function
g_0	Standard gravity acceleration [m/s^2]
h	Enthalpy [J/kg] // Heat transfer coefficient (with subscript) [$W/m^2/K$]
H_∞	H infinity norm for robust control
I	Scenarios indices set
i	Igniter/starter activation flag // Perturbed scenario index
I_i	Identity matrix of order i
I_{sp}	Specific impulse [s]
I_{TP}, $JinTP$	Shaft angular inertia [$kg \cdot m^2$]
Ine	Component's inertia [$1/m$]
J	Cost function
j	Time step along prediction horizon
J_{perf}	Global performance criterion
K	Controller, gain
k	Discretised step // Thermal conductivity [$W/m/K$]
$k(\cdot,\cdot)$	Parametrised covariance function for Kriging
k_i	Butterfly-valve pressure-drop coefficient correlation coefficients
k_{hy}, Khy	Component's alternative resistance coefficient [$1/(m^4)$]
K_I	MPC integrator gain matrix
k_{res}, Res	Component's resistance coefficient [$1/(kgm)$]
km	Parameters conglomerate for $\dot{m}$ simplified NL ODE
kom	Parameters conglomerate for ω simplified NL ODE
kp	Parameters conglomerate for p simplified NL ODE
L, Lng	Length [m]
L_v	Vaporisation heat [J/kg]
$LCH4$	Liquid Methane
$LH2$	Liquid Hydrogen
LOX	Liquid Oxygen
M	Molecular weight [kg/mol]
m	Number of control inputs
m_γ, n_γ	Slope and ordinate at origin for C^* linearisation
MR	Mixture ratio [-]
N	Rotational speed [rpm]
n	Number of states
N_p	Prediction horizon in steps
N_R	Reduced rotational speed [-]
N_u	Control horizon in steps
os	Overshoot
P	Positive-definite matrix, solution of LYAPUNOV equation
p	Pressure [Pa, bar] // Parameter
Q	States weights matrix
q_{cool}	Evacuated heat from chamber by unit of surface [W/m^2]

R	Gas constant [$J/K/kg$] // Control-inputs weights matrix
$R(\cdot,\cdot)$	Parametrised correlation function for Kriging
Rad_T	Turbine radius [m]
S	MPC integrator-variables weight matrix // SOBOL index
s	Laplace variable
ST	Specific torque [-]
T	Temperature [K]
t	Time [s]
T_p	Prediction time horizon $[s]$
T_u	Control time horizon $[s]$
t_{wall}	Wall thickness [m]
Tq	Torque [Nm]
U	Allowable set for control inputs
u	Control input
V	Volume [m^3]
$V(\cdot)$	LYAPUNOV function
W	Work [J]
w	Perturbation
w_t	Exogenous input
X	Allowable set for states
x	State
y	Output
z	Integrator decision variable
ACC	Active Combustion Control
AG	ArianeGroup
ANN	Artificial Neural Network
CC, C	Combustion chamber
CCV	Chamber coolant valve
CFD	Computational Fluid Dynamics
CL	Closed Loop
CNES	Centre National d'Études Spatiales
CR	Cooling circuit
DES	Discrete-Event System
DF	Describing functions
DHA	Discrete Hybrid Automaton
DLR	Deutsches Zentrum für Luft- und Raumfahrt
E.MPC	End-state tracking MPC
EGO	Efficient Global Optimisation
EI	Expected Improvement
ESA	European Space Agency
F, fu	Fuel
FAST	Fourier Amplitude Sensitivity Analysis
FCV	Flow-control Valves
FDD	Fault Detection and Diagnosis

FDI	Fault Detection and Isolation
FPOV	Fuel-Preburner Oxidiser Valve
FTCS	Fault-tolerant Control Systems
GC	Hot gases (*gaz chauds*)
GG, G	Gas generator
H	Hydrogen
HIL	Hardware in the Loop
HMPC	Hybrid Model Predictive Control
HMS	Health Monitoring System
HPLoC	Heuristic Predictive Logic Controller
HPT	High-Pressure Turbine
I, Inj	Injector
ICS	Intelligent Control System
ISA	International Standard Atmosphere
ISFM	Engine Functional Simulation Platform (*French*)
JAXA	Japanese Aerospace Exploration Agency
KKT	Karush-Kuhn-Tucker
L	Liquid // Line
LEC	Life-Extending Control
LFT	Linear Fractional Transformation
LHS	Latin Hypercube Sampling
LMI	Linear Matrix Inequality
LPM	Lumped Parameter Method
LPRE	Liquid-Propellant Rocket Engines
LPV	Linear Parameter-Varying
LQG/LTR	Linear Quadratic Gaussian/Loop Transfer-function Recovery
LQR	Linear-quadratic regulator
LS	Least Squares
LTI	Linear Time Invariant
LVDT	Linear Variable Differential Transformer
M	Methane
MBFD	Model-based fault detection
MFV	Main Fuel Valve
MILP	Mixed Integer Linear Programming
MIMO	Multi-Input Multi-Output
MLD	Mixed Logic Dynamical
MOV	Main Oxidiser Valve
MPC	Model Predictive Control
NASA	National Aeronautics and Space Administration
NLP	Nonlinear Programming
NLSS	Nonlinear state space
NMPC	Nonlinear Model Predictive Control
O, ox	Oxidiser, oxygen
ODE	Ordinary Differential Equations

OL	Open Loop
OPOV	Oxidiser-Preburner Oxidiser Valve
P	Pump
PDE	Partial Differential Equations
PI	Pump inlet // Proportional-Integrator Controller
PID	Proportional-Integrator-Derivative Controller
PLC	Programmable Logic Controller
PN	Petri Net
PRBS	Pseudorandom Binary Sequence
PWA	Piecewise affine
PWM	Pulse-Width Modulation
QIH	Quasi-Infinite Horizon
RC	Reconfigurable Control
RML	Recursive Maximum Likelihood
RREC	Robust Rocket Engine Concept
RTD	Resistance Temperature Devices
RVDT	Rotary Variable Differential Transformer
SIDF	Sinusoidal-input Describing Function
SISO	Single-Input Single-Output
SQP	Sequential Quadratic Programming
SS	State space
SSME	Space Shuttle Main Engine
T	Turbine
T-MATS	Toolbox for Modelling and Analysis of Thermodynamic Systems
T-RETM	Toolbox for Rocket-Engine Transient Modelling
T.MPC	Trajectory-tracking MPC
TCV	Thrust Control Valve
TP	Turbopump
TVC	Thrust-Vector Control
V	Valve

Bold symbols, e.g. $\mathbf{x}$, denote vectors.

List of Figures

List of Tables

Part I

Introductory part

1 Introduction

The automatic control of liquid-propellant rocket engines (LPREs) is a specific field that presents multiple goals and constraints and that is usually simplified for adapting to the restrictive real space-engineering scenarios. Classically, the control system handles the operation of these devices at a finite set of predefined points. That is to say, engines are controlled around a set of steady-state operating points. This approach reduces their throttability domain to a restricted interval in which they are known to be fairly safe in nominal conditions. The main drawback of this strategy is the uncontrolled operation of transient phases, which have a great impact on the duration of engine life. Indeed, most engines are for single use only. In that sense, according to LE GONIDEC, [74] if reusable engines are to be built, their control systems need to be complexified. Let us bear in mind that the better the control system maintains the desired engine behaviour, the better its end-of-mission state will be, which is decisive for reusability. Indeed, facing the reusability feature implies stronger robustness requirements on that system [74]. Those demanding requirements mainly arise from the greater perturbations that can take place, of endogenous and exogenous nature. The additional multi-restart capability of reusable engines also toughens robustness specifications in the same sense. Thrust modulation or throttling is also a novel required capability of these engines. These contextual requirements are explained in greater detail throughout this introductory chapter.

In LPREs, the control references generally correspond to the two variables defining its operating envelope: combustion-chamber pressure (related to thrust) and mixture ratio (oxidiser to fuel ratio), which are usually controlled by means of adjustable valves. An overview of the functioning principles of these devices is provided in Section 2.1. In the following sections, the current context of launchers (Section 1.1) as well as the problematic of their associated reusable engines (Section 1.2) are discussed. Subsequently, an overview of the state of the art of LPREs (Section 1.3) as well as the explanation of the subject of study and outline of this thesis (Section 1.4) are provided. The list of publications is summarised in Section 1.5.

1.1 Context of current launcher programmes

The context of LPREs cannot be addressed without making reference to the vehicles they mainly propel. Current design trends of space launch vehicles are leaning towards reusable conceptions. Indeed, affordable access to space requires reliable and safe reusable transportation systems presenting maximum thrust-to-weight ratio and specific impulse. The economical and logistic advantages of reusable launchers have already been detected by several public and private agents around the globe. For instance, SOLTANI *et al* [150] point to cost reductions, return capability enhancement and less environmental impact.

Many conceptions have been formulated, such as launchers based on combined propulsion systems joining air-breathing and rocket engines (turbo-rockets, scramjet-rockets, etc.). However, the current developments and tests of reusable launchers still rely on rocket engines, as shown in Figure 1.1.1.

Figure 1.1.1: Main agents in the current context of reusable launchers

Namely, two US-American companies are currently taking the lead in launcher reusability: SPACEX [152] and BLUE ORIGIN [17]. The former is developing heavy launch vehicles: *Falcon 9*, with proven reusability of its first stage, and the already tested *Falcon Heavy*. Indeed, that company has identified the reuse of booster stages as a recurring cost reduction possibility [5]. BLUE ORIGIN, more focused on space tourism, has also recently demonstrated this functionality, but in this case their full small *New Shepard* launcher is reusable.

Japanese programmes are starting to deal with reusability by conceiving in-flight technology experiments and by mastering orbiter technology, as explained by BAIOCCO AND BONNAL [5]. Indeed, JAXA (Japanese Aerospace Exploration Agency) and Mitsubishi Heavy Industries [60] are developing a small reusable sounding rocket ($100kg$ to $100km$) propelled by four LOX/LH_2 (liquid oxygen liquid hydrogen) engines aiming at reaching a reusability target of 100 flights, thanks to both thrust control and health monitoring techniques. A recent Russian project, also indicated in reference [5], is the *Baikal* one [47], envisaging a reusable booster based on $LOX/kerosene$ propelled by the *RD-180* and *RD-191* engines.

With regard to the European developments, there have been some ESA (European Space Agency) projects for reusable spacecraft, such as *Hopper* [38], a cancelled spacecraft with horizontal take-off and landing. It would have presented a non-disposable primary stage and an expendable upper stage. Recently, an ESA future launcher preparatory programme has been assigned to Spain's PLD SPACE [123], in the domain of small-satellites launches. Furthermore, European projects for heavy launchers beyond *Ariane 6* (collaboration between ESA, CNES (Centre National d'Études Spatiales) and ArianeGroup), for which reusability was not considered profitable [5], are now contemplating this feature as a way of adapting to the evolving market. Some ideas for the design of the future *Ariane NEXT* might come from the research studies performed on the *Callisto* reusable launcher demonstrator [156], a collaboration between European industries and institutions and JAXA.

All the mentioned companies and institutions are tending to see launcher stages reusability as a way of reducing launch recurring costs. However, there are some limiting factors to consider too. First of all, if the booster is to fly back to Earth, additional propellant mass has to be loaded. Then, scheduled maintenance is a necessity in these vehicles, contributing to raise the costs. Critical components are inspected and replaced if deviations reach intolerable values. That was the case for the most famous *semi-reusable* launcher, the Space Shuttle, which required comprehensive maintenance and replacements after each mission, in particular for its engine.

1.2 Reusable rocket engines problematic: motivation of the thesis

The first example of reusable rocket engine which usually comes to mind is the Space Shuttle Main Engine (SSME), which propelled the US-American shuttle. Despite its successful operation, this staged-combustion LOX/LH_2 engine presented several endurance issues which led to a non-optimised and expensive maintenance [87]. These issues, most of them being of thermal nature, were partly related to the type of propellant chosen, being cryogenic and of low density. These are some of the reasons why the future European reusable engine *PROMETHEUS* is expected to run on methane (CH_4) [5]. This engine, currently under development, consists in a gas-generator (GG) cycle with a single turbopump shaft. It is expected to attain 950-$1000kN$ of thrust, and I_{sp} (specific impulse) between $326s$ and $366s$ [5], depending on the stage where it will be placed. A 5-time reuse and two to four ignitions per flight are envisaged.
In fact, both methane and kerosene propellant families, well mastered in industrial and aeronautical applications, present potential advantages for reusability, mainly due to the similar densities between fuel and oxidiser [5]. This feature permits lower rotation speeds and mono-shaft turbomachinery designs, a lower combustion chamber load and a high degree of similarities and synergies between the oxidiser and fuel lines. These characteristics lead to a lower unitary cost of the engine and to a higher level of robustness.
In addition, these new engines also require throttling capabilities so as to meet reusable launchers' mission objectives. As explained by CASIANO *et al* [27], LPRE designs generally consider constant-thrust operation, with small variations around the equilibrium point. But many scenarios exist where throttling would be necessary: planetary entry and descent, space rendezvous, orbital manoeuvring, hovering or hazard avoidance. The fact that throtteable LPREs are able to continuously adapt to the optimal (most fuel-economical) thrust curve ensures best vehicle performance, in comparison to discrete variations. In that reference [27], different throttling methods on US engines are presented such as special injectors (high-pressure drop, dual-manifold, variable area, etc.), and complex techniques (pulse modulation, multiple chambers, throat throttling). They enumerate several quantities and parameters which can be used to vary thrust: propellant flow rates, propellant types and composition, and nozzle exit and throat area. However, they also state that the most common way of tuning thrust is regulating the propellant flow through control valves. This is indeed the main strategy found in the literature, as it will be discussed throughout the dissertation. The main issues concerning down-throttling, that reference [27] points out, are: combustion and system instabilities (mainly caused by suboptimal injection), performance deterioration, excessive heat transfer and pump dynamics, which are typical problems in reusable engines.
This propulsive feature can be illustrated with examples. For instance, the low throttling capability of BLUE ORIGIN's *BE-3* (18%) [17] simplifies their vertical landing process. SPACEX's *Merlin 1D Vacuum* engine is said to throttle down to 39% [152]. Each engine of JAXA's reusable sounding rocket alone is said to be throtteable per se from 40% to 100% of thrust, enabling a vertical landing of the launch vehicle [60]. Their expander-bleed cycle is indeed required to be restartable, avoiding ignition problems related to moisture freezing. SSME presented a $[50\%, 109\%]$ interval for thrust and $[5.5, 6.5]$ for mixture ratio [142].
All the aforementioned new capabilities of new-generation LPREs (reusability, throttling, multi-restart)

render their operation more complex, and this might not be accomplished with simple control systems. Indeed, facing the reusability requirement implies stronger robustness requirements on the control system, which is in charge of ensuring a correct behaviour of the engine. Those demanding requirements mainly arise from the greater perturbations that can take place: increased endogenous perturbations due to possible components faults or evolving parameters and exogenous perturbations related to the mission profile, also more complex in reusable launchers. The multi-restart capability during flight also toughens robustness specifications in terms of ageing (and hence parameter variability) and varying initial conditions. Regarding throttling at test-benches, the classical linear multivariable control attains a reduced envelope around the steady-state nominal point, between 70% and 120% of thrust [74]. But most of the real flying engines are tuned to only attain the nominal thrust.
Therefore, larger controlled domains, considering the different regimes of operation of the engine, are a key field of improvement if these engines are to be safely throttled. All in all, it can be said that the potential need for reusable launchers and their associated engines represents a real challenge for control-system designers, which motivated this thesis.

1.3 Overview of achievements and limitations in the literature

In the literature, this problematic has been partially faced. Even if the first reference reviewed dates from the 1950s, a relatively low number of publications covering the control of LPREs, whether of reusable nature or not, have been published. This fact might stem from intentional confidentiality strategies by companies and agencies, since rocket engines are complex two-sided technological assets (with military and civil uses), generally needing wide technical know-how. But it could also mean that this bi-disciplinary topic has not been extensively tackled by the research community.
One of the most relevant articles published, by MUSGRAVE *et al* [109], developed a control loop targeted at reusable engines. This loop was rendered robust and reconfigurable in the event of some predefined failures. For synthesising their controller, they derived first a linearised model of a reusable engine. Their global approach allowed to operate the engine in a more reliable manner during steady-state stages. That is to say, the desired end-state was achieved and faults could be robustly counteracted. Nevertheless, neither the transient phases nor the explicit consideration of thermodynamic constraints were handled. Moreover, a non-exhaustive list of faults was taken into account.
Reference [55] remarks that thrust and mixture-ratio control in real flights can be achieved in open loop (OL) if a high accuracy is not required, or if off-line optimisation strategies are pertinent [37, 63]. However, closed-loop control is performed in the majority of publications. Most of authors have selected conventional PID-based (Proportional-Integrator-Derivative Controller) techniques for controlling around a given nominal point making use of linearised models. Concretely, PI (Proportional-Integrator) controllers are the preferred option [60, 65, 77, 115, 117, 154]. The only reference making use of closed-loop (CL) nonlinear techniques [89], considered damage modelling and control in their loop as other NASA reusable-engine publications have done [37, 87, 109], which seems appropriate for extending engine life. Other more complex approaches present in the literature, incorporating some hybrid [175] or robust techniques [71, 141], enhance certain aspects of performance and robustness in some engine cycles. Further references and their review are included in **Chapter 2**, which has led to the publication [121].

To summarise the state of the art, closed-loop control of LPREs has been achieved around their steady states. Concretely, robust approaches facing uncertainties and some faulty scenarios are present in the literature. Damage mitigation in certain components has also been considered. Regarding transient phases, only non-robust open-loop approaches have been developed. A common denominator of most of those approaches is the use of linearised thermodynamic models of the engine about their nominal set point. However, these control strategies have neither dealt with the demanding transient phases in a safe closed loop nor enabled robust throttling in an enlarged operation domain.

These unaddressed points in the literature justify this research study.

1.4 Subject and outline of the study

This study has consisted in developing a control loop for liquid-propellant rocket engines, which is adapted to the whole set of operating phases, transient and steady-state, and which is robust to internal parametric variations and modelling uncertainties at a fair computational cost.

The main methods and tools used or explored throughout the thesis are presented and discussed in **Chapter 3**.

The first step of this work was to develop the simulator block of the control loop, aiming at representing the transient thermo-fluid-dynamic behaviour of a gas-generator-cycle LPRE. The reason for the choice of this type of cycle is that European reusable engines under development (*PROMETHEUS*) will present it. For this sake, zero-dimensional models of each main engine component were selected from the literature or developed. Apart from the typical thermo-fluid-dynamic and mechanical conservation equations, no high-frequency transient phenomena, which could not be counteracted by the actuators of the cycle (valves), are considered in models. These sub-models were then assembled to build the transient simulator based on Ordinary Differential Equations (ODE). This assembly was aimed at representing the thermodynamics of the engine system as a whole while taking into account the inter-component interactions. Subsequently, the capacity of the resulting simulator to capture real engine's thermodynamic behaviour was verified, especially of the entire start-up transient. (**Chapter 4**).

The next stage consisted in deriving global mathematical expressions in the form of state-space models making use of the previously-built simulator. These models were also aimed at describing GG-engine's thermodynamic behaviour, but in this step a global explicit set of nonlinear ODEs, enabling the subsequent model-based control, was sought. For this purpose, the considered component models were symbolically joined. That is to say, all differential and algebraic equations defining components behaviour were causally combined according to engine's flow plan. Subsequently, several physical and mathematical simplifications were carried out so as to establish tractable nonlinear state-space models. The simplified nonlinear model was then linearised, and a comprehensive analysis of its response and characteristics was elaborated so as to verify its controllability and observability. Sensitivity analyses allowed to identify the most influential parameters and the worst-case scenario of parametric variations, relevant for the controller synthesis. The main contributions from this chapter and from Chapter 4 are included in [118]. The conference paper [120] contains a summary of models analysis. (**Chapter 5**).

These derived models were subsequently used for synthesising controllers. Firstly, their complex nonlinear form serves to compute off-line a full steady-state reference as a function of the desired final combustion-chamber pressure (related to thrust) and mixture ratios. On-line control goals have not only concerned multivariable tracking, but also hard-constraints verification and robustness to parametric variations at a fair computational cost. In a first step, reference linear controllers were elaborated with PID and Linear-quadratic Regulator (LQR) techniques. However, seeing the complexity of the models and in order to comply with all these goals, Model Predictive Control (MPC), a model and optimisation-based method, was selected as the most appropriate. End-state-tracking MPC controllers with integral action and parametric-robustness considerations (scenario-based robust-MPC) were synthesised, targetting the control of two types of transient phases. These phases are throttling operations and the continuous sub-phase of the start-up, which takes place once all the discrete events that determine it (ignitions and valve openings) have occurred. Valves sections are continuously adjusted for tracking the reference, especially in terms of pressure in the main chamber and injected mass flows. The resulting controllers were connected to the developed simulator, from which a full-state feedback is assumed. These contributions are presented in [119, 120]. (**Chapter 6**).

Regarding the start-up transients, another more precise tracking strategy was evaluated in a second place. For the continuous sub-phase of the start-up transient, the tracking of pre-defined reference trajectories in states and control inputs was implemented. The off-line computation is also performed via model-based optimisation. Although the controller in this case presents the same structure as the aforementioned MPC, some aspects such as the cost and dynamics were modified so as to perform trajectory tracking. The on-line consideration of nonlinear dynamics was explored in the development of these MPC controllers. However, linearisation about trajectories proved to be more adequate for the on-line resolution of the optimal control problem. Analogous considerations on robustness to parametric variations were included. A proposal to the management of discrete control inputs during the discrete sub-phase of the start-up has also been made, in which time differences between activation events could be optimised by taking into account the evolving nonlinear dynamics. These developments are contained in [120, 122]. (**Chapter 7**).

Conclusions and perspectives are drawn in **Chapter 8**.

1.5 Publications

The following list summarises the aforementioned publications.

Peer-Reviewed International Journal articles:

1) 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, vol. 107, pp. 63–84, May 2019. [121].

Peer-Reviewed International Conference papers:

2) 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), (Budapest, Hungary), pp. 58–67, July 2018. [118].

3) 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), Aug. 2019. [119].
4) 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*, in 15th European Workshop on Advanced Control and Diagnosis, ACD 2019 (to appear), (Bologna, Italy), Springer, Nov. 2019. [122].

Abstract-selection International Conference papers:

5) Pérez-Roca S., Marzat J., Piet-Lahanier H., Langlois N., Farago F., Galeotta M., and Le Gonidec S., *Robust Transient Control of Reusable Liquid-Propellant Rocket Engines*, in 70th International Astronautical Congress (IAC) (to appear), (Washington D.C., USA), International Astronautical Federation (IAF), Oct. 2019. [120].

An article summarising the main contributions to the modelling and control of LPREs has been submitted to IEEE Transactions on Aerospace and Electronic Systems.
Submitted Peer-Reviewed International Journal articles:

6) Pérez-Roca S., Marzat J., Piet-Lahanier H., Langlois N., Galeotta M., Farago F., and Le Gonidec S., *Model-based Robust Transient Control of Reusable Liquid-Propellant Rocket Engines*, submitted to IEEE Transactions on Aerospace and Electronic Systems, 2020.

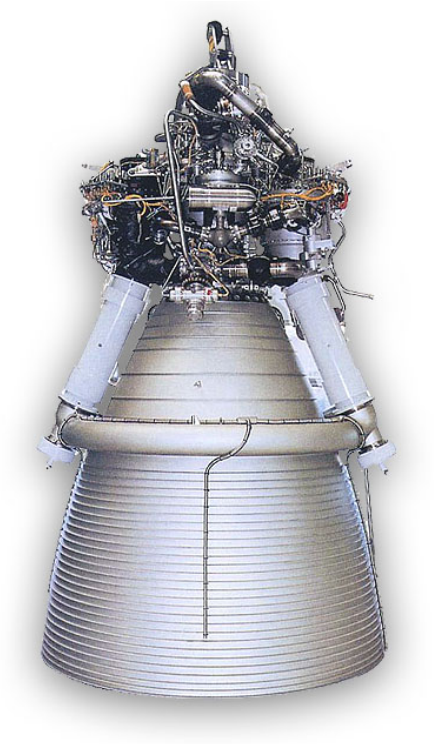
2 State of the art of LPREs control

In this chapter, the field of convergence between the liquid-propellant rocket-propulsion and automatic-control disciplines is reviewed. A comprehensive collection of academic works and some industrial developments are summarised and discussed, making the link to the current context of launcher reusability. The structure of this chapter is the following. Section 2.1 explains the basic functioning principles and design of LPREs, necessary for understanding the rest of the chapter. Section 2.2 serves as a background introduction to LPREs control systems: their purpose, their main parts and which variables are generally controlled through which inputs. Besides, the primary and secondary control loops are described, and also a series of advanced control-system conceptions applicable to reusable engines are introduced. The core sections, containing a detailed review of all articles, including academic and industrial contributions, are Review of control-oriented modelling and Review of control methods. Approaches are structured according to the different fields in automatic control: Identification and modelling, Sensors and actuators considerations, Analysis (contained in Section 2.3), and control techniques (in Section 2.4). The subsequent Summary and discussion section relates all modelling and control approaches and observations to one another. Section 2.6 contains the concluding remarks and the discussion of future control design trends. The content of this chapter has led to the publication of a survey journal article [121].

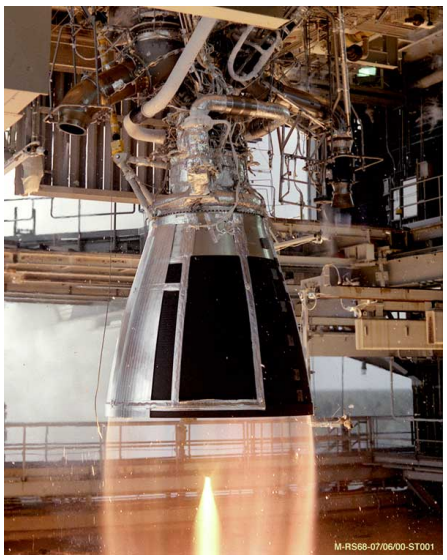
2.1 Rocket-engine generalities

Chemical rocket engines, mainly used to propel launch vehicles and spacecraft, can be classified in the first place according to the state in which the chemical propellant is stored: i.e. solid-propellant, liquid-propellant and hybrid (mixture of both) engines. Indeed, propellants nature determines to a great extent the conception of these systems. The chemical compounds furnished to the rocket engine undergo several thermodynamic transformations throughout their flow through the engine lines and components. The main phenomena are their combustion and expansion, the phenomena that ultimately produce thrust, the *raison d'être* of engines. As mentioned earlier, in this thesis only LPREs are covered.

LPREs are one of the most complex propulsion systems for many reasons. Very high thrust forces (F) can be generated (until the order of $10MN$), corresponding to power amounts of $3-4GW$ [56]. Extreme temperature (T) differences are observed between cryogenic tanks, which can be at $20K$ for liquid hydrogen; and combustion chamber core, which can surpass the $3500K$. These quantities are related to elevated combustion pressures (p_c) too, in excess of $200bar$ in the most powerful engines. Figure 2.1.1 depicts two examples of LPREs, the European *Vulcain 2* and the US-American *RS-68*. The basic static equations relating these main LPREs performance quantities are the following, in



(a) Vulcain 2



(b) RS-68 at test

Figure 2.1.1: LPREs examples

their ideal form [155]:

$$\dot{m}_c = \frac{p_c A_{th}}{C^*} = \frac{p_c A_{th}}{\sqrt{T_c}} \sqrt{\frac{\gamma}{R} \left(\frac{\gamma+1}{2} \right)^{-\frac{\gamma+1}{\gamma-1}}} \tag{2.1.1}$$

$$I_{sp} = \frac{1}{g_0} \sqrt{R T_c \frac{2\gamma}{\gamma-1} \left(1 - \left(\frac{p_e}{p_c} \right)^{\frac{\gamma-1}{\gamma}} \right)} \tag{2.1.2}$$

$$F = \dot{m}_c I_{sp} g_0 = C_F p_c A_{th} \tag{2.1.3}$$

where $\dot{m}_c$ is the ejected mass flow rate, A_{th} is the throat area, C^* is the characteristic exhaust speed, T_c is the combustion temperature, γ is the specific heat ratio, R is the gas constant of the mixture, g_0 is the standard gravity acceleration, p_e is the exhaust pressure, C_F is the thrust coefficient and I_{sp} is the specific impulse. The latter is the ratio of thrust to ejected mass flow, and hence represents the efficiency of the engine. Mixture ratio (MR), defined as the ratio between oxidiser and fuel mass flows, has an influence on the previous formulae via T_c and thermodynamic properties like γ and R. Several types of LPRE cycles have been conceived regarding the manner in which propellants are fed to the main thrust chamber (compound of combustion chamber and convergent-divergent nozzle), but the main elements in a LPRE cycle remain the same:

- Propellant tanks: cryogenic or not (low temperatures preferred for low-density propellants such as hydrogen).
- Pipe lines: ducts joining the different elements.
- Valves: control components guiding and/or regulating propellants flow through the system. They can for example feed combustion chambers, serve as by-pass elements, as safety relief devices, etc.
- Combustion chamber (CC): cavity where combustion occurs. Combustion can be either spontaneous (hypergolic propellants) or not (external ignition required). Even monopropellant mixtures can be burned.
- Nozzle: component whose convergent-divergent geometry provokes the expansion of a high-enthalpy flow and the consequent thrust generation due to the elevated mass flows and speeds attained.

The main factor influencing the cycle selection is propellants injection pressure, which determines the maximum pressure achievable in combustion chambers. Pressure-fed cycles can provide at most the same pressure at which propellants are stored. If pressure regulators are joined to tanks, a constant injection pressure can be furnished. Otherwise, a so-called blow-down operation with a decaying pressure is given.
If a turbopump subsystem is added to the engine, greater pressures than in tanks can be provided to chambers. Pump-fed cycles are therefore more powerful, since greater chamber pressures and hence thrust can be generated. However, they entail a higher complexity. A further classification can be made depending on the manner in which turbines are driven. These are connected to pumps via shafts and are necessary to increase input flow pressure. The most used pump-fed cycles are expander, gas-generator and staged-combustion cycles. The former directly uses propellant flow to drive turbines, but it must first flow through a regenerative cooling circuit around the combustion chamber. GG ones make use of the equally-named additional component to burn a low portion of propellants and feed its output to the turbine. That output flow is then exhausted or redirected to a downstream part of the thrust chamber. Figure 2.1.2 depicts that generic cycle. The staged-combustion is similar, but pre-burners output is directly fed to the main chamber to be burned together with more pumped propellant. Depending on the family of engine cycles, different start-up times are given. Pressure-fed ones are faster (3 to $15ms$) since they only include the operations of purge, valve opening, combustion initiation and pressure build-up, as explained in the book by SUTTON AND BIBLARZ [155]. In contrast, turbopump-fed cycles obviously require more time (1 to $5s$) for starting a GG or preburner and accelerating shafts. These cycles generally require more complex control systems, owing to the greater number of operations and components, and especially to the higher amount of flow-control valves. Therefore, the vast majority of control studies reviewed in this dissertation are devoted to pump-fed LPREs.
Transient phases, like start-up, are very accurately planned, fitting engine's characteristics and ignition type. Depending on the LPRE cycle, different sequences are planned. A typical start-up sequence for GG engines, such as the European *Vulcain 2*, begins by activating the starter related to GG and turbines, which increases pressure in GG so as to start driving the turbines. Next, with turbines turning faster, mass flows start to increase, allowing the opening of main chamber valves first and GG valves subsequently. The first chamber valve to be opened is normally the fuel one, due to two reasons: fuel must first flow through the regenerative circuit and a fuel-rich environment in the chamber is normally

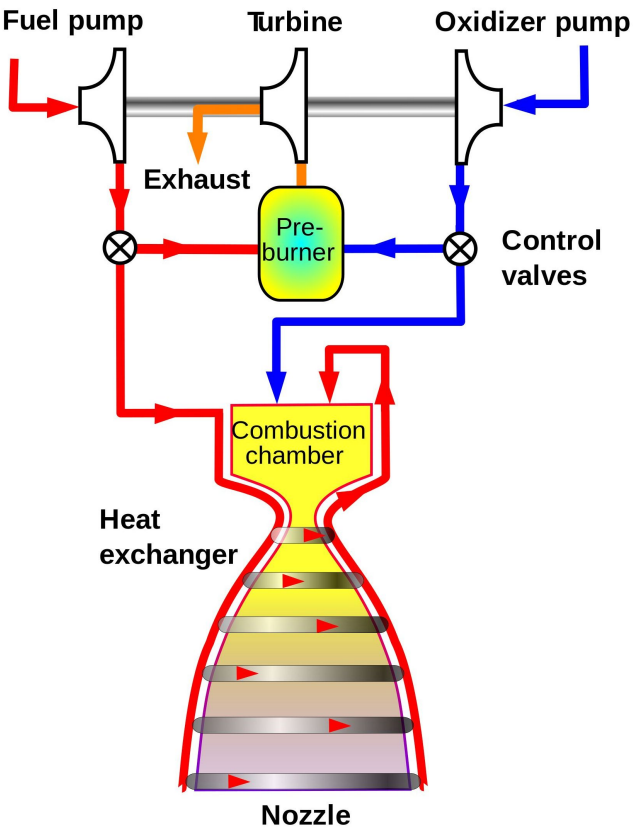


Figure 2.1.2: Schematic of a LPRE GG cycle by *Duk*, under CC BY-SA 3.0

desired. During this low-regime phase, a pyrotechnic igniter is employed to trigger combustion in the main chamber. The opening of GG valves provokes the most intense pressure-increase phase in the start-up.

Several characteristics have to be taken into account when dealing with start transients [55]. Energy and timing characteristics are vital and have to be evaluated and optimised. The influence of external initial conditions or perturbations, such as a start-up in vacuum, must also be assessed, with the aim of avoiding complex and problematic phenomena such as unwanted pressure drop, stall, surge, cavitation, changes in fluid density, etc. The type and quantity of control components and some operating constraints are determined by all these features.

However, the goal of attaining a smooth and reliable ignition, lacking of pressure and temperature peaks, is the main driver in the off-line and generally constant planning of the start-up sequence, adapted to each engine. For instance, ignition overpressure is due to excessive ignition delay times [96]. If ignition is delayed, too much unburned propellant cumulates in the chamber leading to deflagration combustion once ignition finally starts. These pressure peaks may cause back-flows into the feed lines, probably becoming catastrophic. Temperature peaks are not that critical at this phase since the engine normally comes from a chill-down phase.

The shutdown process also presents some constraints, primarily related to some mixture-ratio limits, but is generally simpler to conceive. Another transient phase is throttling from one operating point to another. Traditional LPREs have been designed for a constant thrust level, but current reusable designs tend to allow a set-point variation, generally forced by new launcher requirements. All these

transient phases might not be conceived as completely fixed sequences in the new engine designs. Further details on them are provided in Section 3.1.1.

2.2 LPREs control systems background

The main control problem in LPREs generally consists in tracking set-points in combustion-chamber pressure and mixture ratio, their main operating quantities (introduced in 2.1). This goal is attained via the adjustment of flow-control valves while complying with operating constraints.
Chamber-pressure and mixture-ratio control was achieved in test benches as early as in 1959 by OTTO AND FLAGE [117], by means of an analogue model-based PID approach on a regeneratively cooled hydrogen-fluorine rocket engine. Results during the start-up transient of this pressure-fed engine were successful with a maximum of $\pm 2\%$ deviation. Stoichiometric mixture ratios, which are generally related to maximum combustion temperatures, were avoided.
Before the spread of electronic devices for controlling engines, these used to be pneumatic and powered with helium gas. This gas is still employed for secondary tasks within the engine, although not for control. Simple timers were employed in the early engines to send actuation commands and later they developed into pressure ladder sequences.
In the last decades, automation in propulsion systems has evolved and comprises system engineering, control and health monitoring. Firstly, a multi-physic modelling approach is customarily carried out, joining chemical, hydraulic, electrical, mechanical, thermal and structural aspects. Each new engine design presents a different behaviour. Hence, each type of rocket engine cycle requires a specific control approach. A minority is naturally unstable, others are stable but with non-minimum-phase properties or present varying parameters [74]. In other words, the selection of the most adequate control system depends on system's requirements, accuracy, dynamic characteristics and reaction-time constraints, which are normally defined by mission's profile and by the whole architecture of the propulsion system. For instance, it is convenient to analyse the environment for taking into account all sources of perturbation (accelerations, propellant temperature variations, etc.).
As pointed out by LE GONIDEC 2017 [76], engine control systems rely on components such as hydro-mechanic devices or flow-rate controllers in order to improve performance in terms of accuracy, response time, perturbation rejection and reliability. The in-flight control should ensure a simplification of testing and flight preparation, allowing thrust modulation and versatility.
In a paper by BELLOWS *et al* from the ROCKETDYNE company [8], the need for a more reliable and coordinated control and health monitoring system for rocket engines is highlighted, already in 1984. A reference is made to SSME engine-mounted programmable control system, which managed redundancy, a severe environment and real-time constraints and offered flexibility. The actuators of this staged-combustion-cycle consist in a set of valves, and sensors are flowmeters and pressure transducers. The complexity of the LPREs control problem can already be inferred from the fact that only two actuators (control valves upstream from pre-burners) are used to control the system in a continuous way, while the rest of valves are open or closed according to high-level configuration orders. These signals are usually a part of sequential processes related to transients, often modelled as automata [73].
The paper by SEITZ AND SEARLE [142] explained how the first SSME control system was designed during the seventies. Apart from ensuring monitoring and fail-safe operation, the system could

perform repeatable start, steady and shutdown operations during 100 runs. At this staged-combustion engine, thrust and mixture ratio are controlled by correcting the power repartition between turbopumps, regulated by the preburner oxidiser valves (characteristic of staged-combustion cycles). During the start-up transient, CL control of thrust starts pretty early while mixture-ratio CL control starts later, as explained later in Section 2.3.3. Another loop monitors that certain temperature limits at turbine inlets are not reached by reducing the thrust command when necessary. At shutdown, CL is maintained until the minimum thrust level (50%) is attained.
The European company ArianeGroup have put a lot of effort since the late 80's on the control of rocket engines, as explained in [76]. In 1988, control was only applied to test benches, such as the PF52, PF50 (France) or P5 (Germany), in order to regulate tanks pressures and turbine speeds among others. The main purpose of this automation was to test *Vulcain* sub-systems such as gas generators and turbopumps. Since 1994, control started to be applied to actual rocket engines, so as to maintain their stability (some of them are non-minimum-phase systems) and performance. It was mainly devoted to control *Vulcain* engines in a monovariable way. For the older generation of engines, *HM7* and *Viking*, control systems simply consisted in hydro-mechanical loops. Due to the wish to improve the engine test performances (i.e. response time, reliability, accuracy, etc.), multivariable control was integrated in 2000 (3×3 for *Vulcain* and 2×2 for *Vinci*). New-generation engines will make use of more electric systems, as initiated by *Vinci* (partially electric).

2.2.1 Main control loops

As hinted in the previous paragraphs, LPREs control entails a long list of control subsystems or loops, in charge of dealing with certain variables or flight phases by acting on the controllable inputs (valves, starters or igniters) and by measuring engine's state. But all of them are related by a common management entity such a computer or an *orbiter*, as depicted for SSME in Figure 2.2.1 from [126]. In order to generate a controlled thrust F at a desired specific impulse I_{sp}, two level distinctions must be made according to [76]. At tank level, tank pressure must be controlled, and at engine level, chamber pressure and mixture ratio are the key variables. If the cycle contains a GG, more variables come into play. Indeed, the three main control loops in LPREs described in the book by HUZEL AND HUANG [55] are: thrust-level, propellant-utilisation and thrust-vector controls.
Thrust-level control: mainly governed by combustion pressure and hence by the total injected mass flow (as defined in (2.1.3)). This is the slowest loop, whose bandwidth is determined by thrust requirements. For constant-thrust engines, a tolerance of $\pm 3\%$ is typically achieved by adjusting orifice sizes and by opening and closing propellant valves without the need for CL. Should a greater precision be desired, regulators or controllers have to be used in order to compensate variable conditions. And if throttling is needed, especially during the last propelled phases, chamber pressure reduction has to be implemented either stepwise or continuously. As commented earlier, varying the mass flow is not the only way to modify pressure and its related thrust. When looking at (2.1.1) to (2.1.3), one can see that throat area and I_{sp} can also be varied, whereas C^* remains nearly constant for a given propellant combination. I_{sp} can be modified by changing the mixture ratio (coupling with MR loop, explained in the next paragraph). Nevertheless, as stated by TIMNAT [158], this variation generally leads to performance deterioration. Throat-area tuning is also problematic since complex mechanical or aerodynamic techniques would have to be implemented, such as pintles or secondary injections, which would alter the cycle design and behaviour and hence performance [158]. The schematic

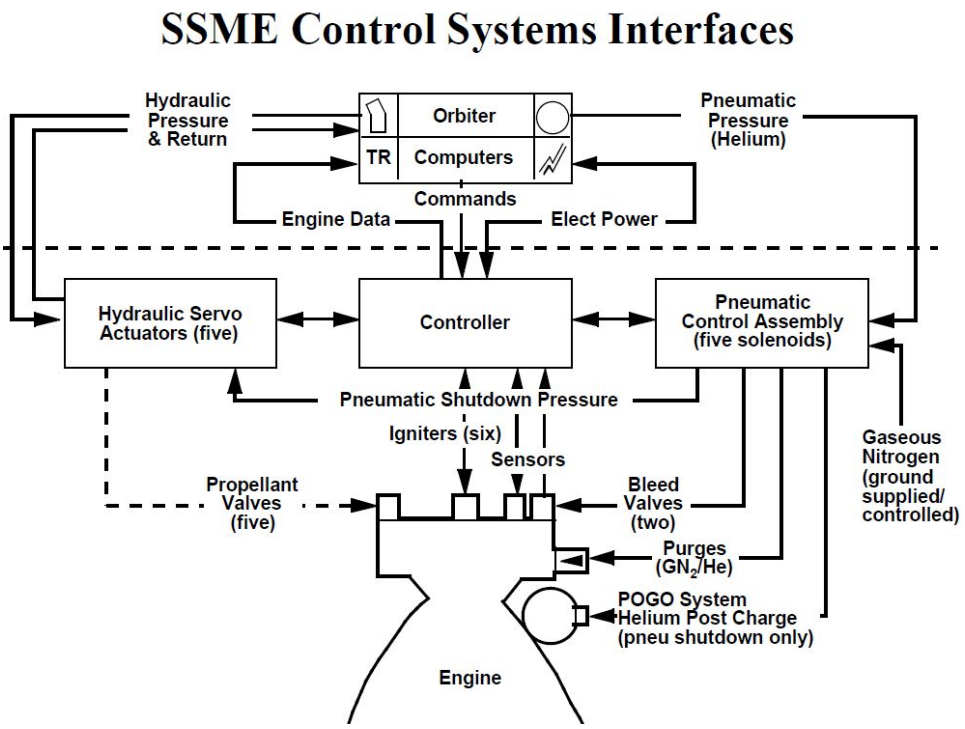


Figure 2.2.1: SSME control chart [126]

by [115], shown in Figure 2.2.2 is a good example of a closed thrust-level control loop using valves, again applied to the SSME. Feedback of pressures, temperatures and volumetric flows is used to compute valve control orders.

Propellant mixture ratio and propellant-utilisation control: this regulation is needed to attain optimum engine performance (maximum specific impulse) and to save propellant resources. As stated in [55], it can be performed either in closed or in open loop. Open-loop control, apart from requiring orifices calibration, can be further refined by weighing the loaded propellants and by installing adjustable orifices. This is acceptable in engines naturally presenting good MR (mixture ratio) tracking, such as single and first stages. In contrast, CL is needed at high-velocity-increment upper stages or restartable engines. In basic cases, MR can be controlled by adjusting the main oxidiser flow valve alone, typically a servo-controlled one.

Reference [88] highlights that this loop should be faster than the thrust-level one so as to minimise high deviations, but attention to their strong coupling must be paid. As indicated by [155], it is common that LPREs are controlled to a constant MR, selected according to performance. However, in some cases this command could vary in order to improve propellant management, especially during the end of engine's mission. This is intended to empty tanks completely and hence reduce the mass of the spacecraft. But it is also normal that mixture ratio undergoes great variations during start-up and shutdown phases, where one of the propellants may lead and hence the ratio is not meaningful. Besides, propellant lines' hydraulic resistances are usually different, resulting in different filling times in the start-up transient.

Thrust-vector control (TVC): an effective way of guiding a launcher is varying the direction of the generated thrust. Therefore, TVC is widely employed. Typical methods pointed out in [55] are gimballed thrust chamber, gimballed nozzle, jet vanes in the nozzle section, secondary injection into

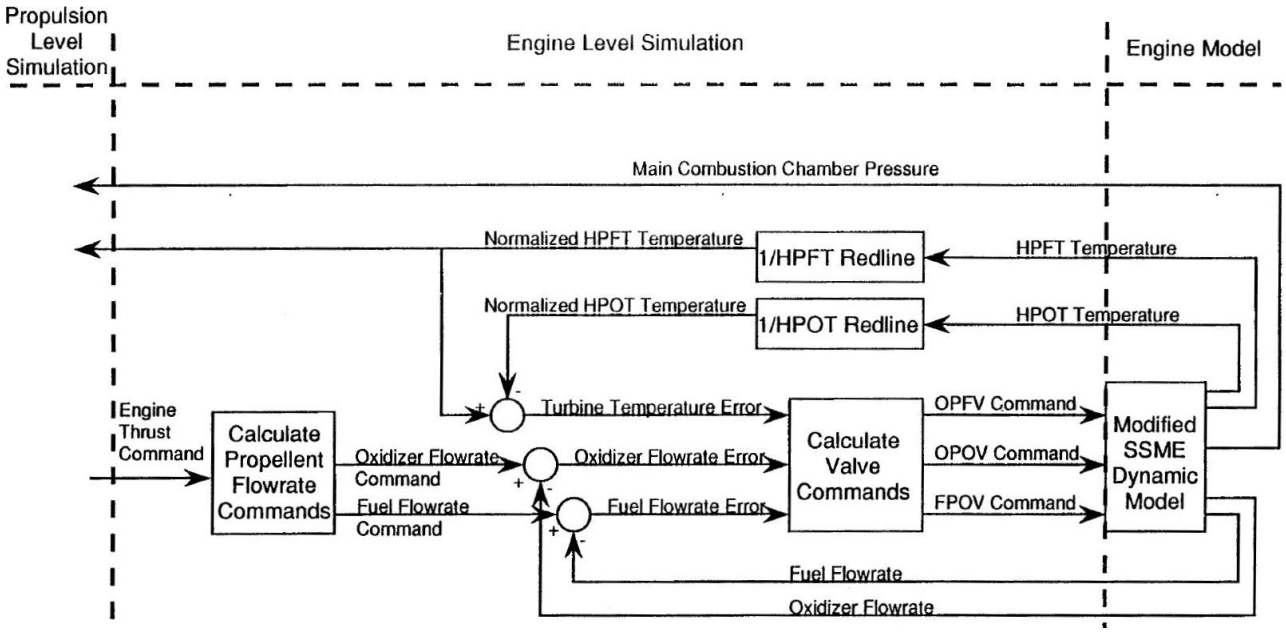


Figure 2.2.2: CL thrust-level control diagram, performed via valves (defined in Section 2.3.1) with turbopump-redlines verification by NEMETH *et al* 1991 [115]

the thrust chamber and auxiliary jets.
Nevertheless, this subsystem, serving as an interface with the vehicle, lies beyond the scope of this thesis. Other secondary but relevant loops explained in [55] and [155] are the following:
Duration control of main stage: includes tank low-level sensor and an accelerometer or similar device to send the cut-off signal.
Safety control: monitoring devices for combustion instabilities or over-temperature. An interruption of electrical power supply always triggers shutdown.
Tank pressurisation control: CL compatible with propellant-utilisation control and thrust control subsystems. Necessary for keeping the required pressure for a nominal operation of the engine.
Control calibration: proper settings of computer, devices, switches, etc.
Checkout and tests control: post-assembly and pre-firing checkouts, allowing operation simulation.
System start control: a sequential start control includes systems preconditioning (purging, chill-down), valves opening, start energy activation and introduction and ignition of propellants in the chamber. Other secondary events may appear such as some related to GGs or pre-burners. The start is either oxidiser-lead or fuel-lead in order to mitigate high-temperature peaks, depending on propellant combination and engine cycle. Between three to five seconds are normal durations of this phase. SSME start-up control strategy was in OL during the first $2.4s$ in thrust and $3.6s$ in mixture ratio, when the respective PI closed loops started [115]. Thanks to tests and simulations, nominal sequences are usually an off-line optimised trade-off between achieving the start of the motor and minimising damaging transients. But in reality the sequence of phenomena is very sensitive to external and internal conditions and characteristics. In OL, small changes might trigger longer transients or even an unsuccessful start, due to the highly nonlinear and coupled effects that take place.
System cut-off control: a quick and secure engine shutdown must be ensured. This shutdown concerns subsystems power, chamber flow rate and purges or flushes in the case of a test. A fuel-rich finalisation generally avoids high temperatures. Moreover, as indicated in [155], the amount of impulse

provided by this last phase may have to be reproducible if the application requires an accurate final vehicle speed.
A current research topic, which would represent another loop, is the precise control of the combustion phenomenon in order to avoid instabilities and enhance performance. Combustion instabilities of low and high frequencies arise when perturbations in velocity or pressure are enlarged by combustion heat transfer and produce acoustic energy periodically. That energy serves to amplify the perturbations, creating a self-sustained cycle. This poses important problems to the structure and performance of the propulsion system. The effects of these instabilities are mainly counteracted via passive methods implemented in LPRE combustion-chamber designs, such as baffles, resonators or acoustic liners. In contrast, Active Combustion Control (ACC) is a recent active methodology described by RTO/NATO [129] used in gas turbines for controlling combustion performance by means of dynamic hardware. This actuation hardware, which is generally pretty different to typical valves, is able to quickly adjust the combustion input in terms of injection timing instead of the traditional spatial distribution modification. Some research works concerning monopropellant rocket-engine combustion control via pintle devices or pressure regulators are [167, 175, 176]. However, that loop lies beyond the scope of this literature review, devoted to the control of the engine system as a whole.

2.2.2 Reusable LPREs control trends

It seems relevant to summarise some insights on advanced control-system conceptions applicable to reusable engines. As commented earlier, reusable engines would require these advanced concepts to a greater extent. The review by LORENZO *et al* 1995 from NASA [87] provided insight into the control technology research trends potentially applicable to reusable rocket engines in the context of NASA's SSME: intelligent control, multivariable control, life-extending control and the Robust Rocket Engine Concept. These concepts were aimed at mitigating durability issues in future reusable engines, and would still be applicable nowadays. The SSME presented indeed issues with turbine blades, bearings, thrust chamber liners and propellant ducts.

Intelligent Control System (ICS). This concept consists in a hierarchy of several control and diagnostic methods: life-extending control, reconfigurable control, real-time diagnostics, component-condition monitoring and engine prognostics among others. The mission-level control is the main director of the system, establishing thrust and mixture-ratio requirements. Then, the propulsion-level control must comply with propellant-utilisation requirements, regulate tank pressures and provide thrust and mixture-ratio commands to the engine level control.
In the ICS design by MUSGRAVE *et al* [109], model-based fault detection based on a modified model of SSME is successfully integrated and demonstrated in real-time. Different failure types are considered: freezing, limiting and leakage, but the focus is made on accommodating a frozen oxidiser valve at down-thrust and up-thrust manoeuvres. The selected valve is in fact the critical element determining the main combustion chamber mixture ratio. In the event of a fault, the new maximum thrust is calculated depending on the valve position and this is transmitted to the propulsion management system. The global damage rate can be estimated by monitoring the turbine discharge temperatures, relevantly representing engine health. If they exceed certain values, the engine is considered to be degraded. Hence, ICS would lengthen engine life and permit degraded performance even on a multi-engine system.

The NASA-Rocketdyne report on the Reusable Rocket Engine ICS functional framework by NEMETH *et al* [114, 115] deals with a rocket engine cluster of three devices as a whole, distributing thrust and mixture ratio among each engine, depending on health, efficiency and on the ageing state of each motor (estimated from measurements via damage models). Diagnostic logics are implemented to define the allowable reference point for each engine according to those criteria. Seven valves can be opened or closed but only two are fully controllable. Hence, different sets of valve commands can fulfil the control goal. At the same time, parameter ranges are respected via three secondary valves. The best combination is selected by the engine-level controller according to a cost function which depends on wear, risk of failure and performance losses. If redlines are reached, cut-off commands are sent to the valve sequence coordinator. The latter redefines the start sequence whenever there are failures so as to minimise the catastrophic risk.

Multivariable control. Single-loop control designs are often simple and do not present demanding requirements. However, multivariable control can provide a more accurate regulation. In fact, rocket engine control is intrinsically a multivariable problem, generally stable in OL and requiring mixture-ratio and combustion-pressure control by means of several valves, typically achieved in a linear way. For instance, the design by [109], making use of the Linear Quadratic Gaussian loop transfer-function recovery (LQG/LTR) and H_∞, has succeeded in stabilising the closed loops. The multivariable ICS by [115] studied the feasibility of CL control during the SSME start-up without success. The multivariable controller was aimed at rendering the system insensitive to parameter variations, perturbations or noise. But it was only effective after the transient, where it zeroes the error on turbopump discharge-pressures deviations, which is not a common strategy. These quantities influence the input flow to the main chamber and in the end determine thrust and mixture ratio. In fact, their commands are translated from thrust and mixture ratio errors.

Life-extending control (LEC). The amount of cumulated damage at critical points can be significantly reduced by an appropriate control during these periods. A life-extending control system is presented in the 2001 paper by LORENZO *et al* [89], aiming at accomplishing high performance and structural stability. The key approach is damage mitigation, performed in both turbines of SSME (O_2 and H_2) achieving positive transient results in the two outputs: chamber pressure and mixture ratio. Both linear and nonlinear control techniques are employed. In conclusion, after a small loss in performance, the engine life can be relevantly extended, reducing damages. However, their damage modelling does not take into account high-temperature effects.
Indeed, feeding damage rates back and using nonlinear optimisation are the keys to minimising damage (fatigue/fracture and creep) at critical phases, according to [87]. There is the need for nonlinear filters since the relation between local stress and damage rate is so. This way, the damage produced during transients is claimed to be divided by three at least. The drawbacks are increased computation and sensing.

Robust Rocket Engine Concept (RREC). Extending LEC, a multidisciplinary optimisation can be carried out to reduce even more damages during the critical transient phases (start-up and shut-down). This optimisation would concern the operating cycle, critical components like turbine blades, transient control parameters and endurance issues. This conception has only been suggested by LORENZO 1995 [86]. However, it would imply greater computation times and maybe dynamic response and propellant usage would be penalised.

No NASA follow-up studies putting in practice the aforementioned concepts on reusable engines have been traced in the literature. Nevertheless, outside NASA, some multivariable approaches on non-particularly reusable engines have been published; and the next sections present them, as well as previous works.

2.3 Review of control-oriented modelling

This section together with the following one, Review of control methods, are the core sections in this chapter, since they contain the detailed review of articles, including academic and industrial contributions. Approaches are structured according to the different fields in automatic control. This section covers identification and modelling, sensors and actuators considerations and system analysis. The greater abundance of research publications during the nineties and the early 2000's coincides with the operation of the SSME and the observation of its reusability problematic. Afterwards, not many recent papers have been published.

2.3.1 Identification and modelling

This subsection outlines model design and identification methods. Derivation of control laws for this type of systems is most often accomplished via model-based approaches. The main thermodynamic assumptions made are mentioned, but comprehensive explanations on thermo-fluid-dynamic modelling are not the main focus here. Section 3.1 and Chapters 4 and 5 are devoted to it.
The physics of rocket-engine components are generally described by means of thermo-fluid-dynamic and mechanical conservation equations. Terms are generally not developed in their full complexity since the most precise model is not the target. It is preferable to manipulate a tractable model for deriving control laws. The three flow equations are the typical mass (or continuity), momentum and energy conservation equations, described in Section 4.2.1. Regarding turbopump mechanics, the shaft usually presents a mechanical differential equation on its rotational speed. However, in the industry, models can also be built thanks to multi-physical simulation platforms, such as *CARINS* by CNES/ONERA [116]. Due to the progress in terms of modelling tools, linear state-space models can sometimes be automatically derived from multi-physical nonlinear models [76], which avoids in some cases the manual derivation of engine's equations.

It is relevant to mention the only open-source *MATLAB*® /*Simulink*® toolbox for modelling and analysis of thermodynamic systems (*T-MATS*) until the date [29]. *T-MATS* includes several thermodynamic and control modelling libraries, with focus on gas turbines. In other words, a complete system simulation can be set up by joining 0-D thermodynamic component models.
Concrete examples of publications using the *T-MATS* tool are for instance the work of ZINNECKER *et al* [179], who successfully simulated a twin-spool turbofan engine in a dynamic conception. SEOK *et al* [143] deal with a problem of integrated thrust and electrical power management in an aircraft by means of this toolbox.
The enhanced *T-MATS-Cantera* sub-library [70] allows the computation of precise fluid properties according to the thermodynamic state and chemical composition.

Linear identification

Instead of developing a model for control from thermodynamic equations, some authors have opted for identifying it from simulations or tests. The dynamic behaviour of the SSME was identified by DUYAR *et al* [44]. The open loop considers the opening angles of its main five valves as inputs, and six outputs apart from the typical two: pressure, temperature and speed of both HPT (high-pressure turbines). Preliminary information on the system's nonlinearities and bandwidths is obtained by exciting the open and closed loops. The fuel-preburner oxidiser valve (FPOV) and the oxidiser-preburner oxidiser valve (OPOV) are identified as the dominant ones, which allows the removal of the other valves from the open loop. Indeed, these dominant valves are regulated in all main SSME control publications [37, 109, 115, 142]. A pseudorandom binary sequence (PRBS) is used as perturbation signal for identification. It consists in a wideband long-duration signal switching from one to another value. A step signal with a 2%-amplitude PRBS is employed as the driving signal for the CL. Responses point to the presence of valves nonlinearities, but these can be isolated and even removed for identifying the main system.
By linearising the equations, the following transfer-function structure is taken as the base for parameter estimation between the j^{th} actuator and the i^{th} sensor:

$$H_{ij}(s) = C_i(sI - A)^{-1}B_j, \tag{2.3.1}$$

where A, B and C are standard state-space matrices and s is the Laplace variable. The recursive maximum likelihood method (RML) is the selected approach, determining the transfer function $H(s)$ coefficients from input and output data by subtracting the nominal values from the perturbed ones. The order of the model must be predefined. Indeed, the authors follow the *parsimony principle*, selecting the highest order at which the further error decrease is negligible, which in this case leads to four poles and four zeros. Comparisons with nonlinear simulations pointed to good representativeness around a limited-response region about the 100% thrust level, but the nonlinear effects of valve linkage backlash and valve stiction (static friction) were not considered, recommended to be included in future models.
Valves modelling in [117] was performed through identification after testing real devices. Dead times were considered between valves and sensors, capturing the time required for a pressure wave originated at the valve to arrive at the sensor. This causes a phase lag with no change in amplitude. The engine dynamic response is flat after $100Hz$. The equation of pressure variation is obtained in a simple way by combining static equations.
The least-squares (LS) method can also be used for model identification based on data coming from complex simulation platforms.

Linear thermodynamic modelling

Some approaches directly tackle linear models, defined in the frequency domain. The linear model by [132] handles the flow through pipelines. Gases are considered ideal, expansion is isentropic and regulators are ideal mechanical systems. In the feed lines, liquid propellant is treated as compressible but not in injectors. The combustion chamber is considered as a vessel with time delays for injection, mixing and combustion. The last phenomenon is treated as ideal, with perfect burning.

That model comes from previous work by the same research group [133, 134], in which a model block diagram in the frequency domain is established. The components described in those papers are the combustion chamber, the injector head, cooling jacket and pipelines, which are modelled linearly at the nominal operating regime.

Linearised models

The most common modelling-for-control approach in LPREs is linearising a nonlinear thermodynamic model about operating points. Regarding the reusable engine modelling used in [89] (defined in [124]), similar to the SSME, standard lumped parameter schemes have been applied for approximating the partial differential equations related to mass, momentum and energy conservation as first-order ones. Causal interconnections are defined to join all the engine's sub-elements, which results in a plant model of eighteen states, two control inputs and two outputs. However, the model is linearised around $p_c = 176bar$ and $MR = 6.02$ and reduced to a 13-state model via the HANKEL order reduction. Physically speaking, both turbopumps speeds are controlled by the corresponding preburner pressure, and so propellant flow is determined. The oxygen mass flow into the pre-burners is individually handled by two flow-control valves (FCV). In another paper by these authors [88], it is claimed that the linear models of that engine did not present relevant variations while throttling, suppressing the need for gain scheduling.

MCDERMOTT *et al* [102] obtained an analytical dynamic model of the small *Surveyor* engine, which is throtteable. This pressure-fed bipropellant engine covers a thrust range between 133 and $462N$. They linearised the model, which rendered it only accurate for small amplitude variations about the operating point. However, real CL results pointed to adequacy for large thrust deviations. The transfer function obtained in the end expresses thrust variation as a function of variations in fuel and oxidiser flow rates.

Although the work of BERGMANS AND MYERS [13] is devoted to a solid-propellant GG which runs an air turbo-rocket engine, it is relevant to comment on their decoupled SISO (single-input single-output) modelling approach. They derived first a nonlinear model of the GG and its outlet nozzle by means of the standard conservation equations (applied to solid propellants) and of sub-scale test data. The transfer function between valve area and outlet pressure is obtained after linearising the nonlinear model about a pressure which is not the steady-state one but is not far from it. This linearisation proves to be valid within a pressure range of around $35bar$.

The modelling approach by SOLTANI *et al* [150], even if devoted to fault detection and isolation (FDI), shares the same philosophy as control-oriented models. Their overall nonlinear model includes valves, pumps, a generator and the combustion chamber. It contains fourteen inputs, eighteen outputs, fourteen non-measurable variables and six continuous states. Twelve failure cases are considered, but the complexity of the system is too high to perform a reasonable monitoring. Hence, the structural analysis method is employed to divide a launcher propulsion system into simpler sub-elements in order to perform an easier and more specific FDI design. Its qualitative nature, abstracting system's behaviour, provides sub-components identification, residual design capability and reconfiguration possibilities.

Simple linearised models for valves, injector (including mechanical elements like masses, springs and dampers) and the combustion chamber are employed by ZHOU [178], allowing the definition of a relatively compact transfer function of the whole engine system. A constant combustion delay,

expressed in terms of injection and ignition delays (less than $5ms$), and ideal gases are considered.
In the report by LE FUR *et al* [71], the modelling and control of an expander-cycle rocket engine is tackled. Since their objective is analysing low frequencies, capacitance, inductance and combustion dynamics are not considered. The computation of an equilibrium point at some given valve gains is performed via a model of 19 static equations and unknowns: two turbopump speeds, four mass flows, five temperatures, seven pressures (CC, cooling inlet and outlet, turbine inlet and outlet among others) and mixture ratio. That system of equations is nonlinear and some equations are implicit. It is solved by defining a mesh of chamber-pressure and mixture-ratio values.
However, their simplified state-space representation only integrates seven states: rotation speeds, chamber pressure, the gains of the two control valves (turbine bypass) and their derivatives. Control inputs are the two corresponding opening positions. Some exogenous inputs are also considered: pump-inlet pressures and pressure drop coefficients of chamber valves (not controlled). Besides, there is a set of algebraic equations (eleven) related to some intermediate variables, states and exogenous inputs. Measured outputs are considered to be only rotation speeds, since they are the variables which can be directly controlled by turbine by-pass valves. All variables are normalised to their nominal values.
Linearisation around the previously computed equilibrium points is then performed on this model, since three of the equations are nonlinear. These equations are precisely the only ones not depending on control inputs. The presence of algebraic equations complicates the process, attaining a matrix of dimension 99.
Further examples of linearised models can be found in [141], [140], or in [175] and [176], where the CROCCO monopropellant combustion model [35] is used.

Describing functions

NASSIRHARAND AND KARIMI [111] succeeded in controlling the mixture ratio of a LPRE making use of a systematic describing-function method together with factorisation theory. The controller is designed for a linear plant coming from the chosen methodology. These results may allow the substitution of complex hydromechanical control valves by simple ones driven by microprocessor-based servomechanisms. However, these techniques are only applicable to engines whose main control loops (chamber pressure and mixture ratio) are decoupled, which is not the usual case.
Indeed, the idea behind describing functions (DF) is to represent nonlinear systems as linear time-invariant transfer functions which depend on the amplitude of the input signal. This generally translates into considering a set of linear systems, simplifying the initial problem at a said satisfactory robustness level. Concretely, sinusoidal-input DF (SIDF) models are employed in [111] for several reasons. Firstly, standard linear models do not represent correctly the amplitude dependency of the full plant. Other models such as random-input DF neither capture the dependency of the nonlinear state-space plant at the desired frequencies. Thus, a set of SIDF is considered as a safe base for a robust control without renouncing to performance. Besides, these functions are defined by just one parameter, the amplitude of excitation, reducing the design complexity. Basically, these models are obtained by considering a sinusoidal excitation of the plant and by computing the FOURIER integrals of the nonlinear equations of motion.

Nonlinear modelling

In general, rocket-engine models in their nonlinear form without linearisation have only been used for simulation and/or analysis, not for deriving control policies. In this sense, a performance model simulating *Vulcain*'s internal flow characteristics (pressure, temperature and flow rate) was developed by IFFLY AND BRIXHE within SNECMA Vernon (which became part of ArianeGroup) [57], with the collaboration of Techspace Aero. The main input data to the model are pump-inlet pressures and temperatures, geometric and thermal features and valve settings. Twelve states are considered: thrust chamber and GG mass flows, dump-cooling mass flow, pumps rotational speed, turbine-outlet pressures and temperatures and H_2-turbine mass flow.
Engine parameters in the previous model are obtained by data reconciliation with tests, that is to say, by estimation through generalised residual sum of squares (Gaussian assumption). Discrepancies of two origins are taken into account: test-measurement uncertainties and balance of physical equations. Operating ranges are also computed. These are the set of operating points reachable with a certain probability, considering elementary dispersions following Gaussian laws on engine parameters. The method employed is a time-efficient Monte Carlo scheme, taking engine's behaviour as linear. With this linear model, a covariance matrix can easily be calculated, allowing to draw an iso-probability locus at each operating point. By joining all these ellipses, a full flight envelope can be drawn. Operating limitations can also be calculated by introducing a constraint and then determining the valve configuration respecting this constraint. An accuracy of around 1% on the basic model parameters is attained.
SAINT-MARD *et al* [131] developed a nonlinear model of a turbopump-fed LPRE for simulation. On this model, also valid for transients, feasible working points are identified and requirements in terms of control-devices' OL accuracy are indicated. It presents some algebraic equations such as chemical reaction ones, including mass conservation, equilibrium equations and enthalpy conservation. In addition, it contains iterative loops for certain components like the nozzle. It is subject to operating constraints on mixture ratio and turbopumps' speeds. Concretely, mixture ratio must stay away from the stoichiometric value and critical shaft speeds must be crossed quickly so as to avoid eigenfrequencies and over-speed, which are common practices in LPREs.
ZHANG 1984 [174] applied the state-space framework to the analysis of the dynamic characteristics of a variable-thrust LPRE, obtaining results agreeing with experimental data. The engine concerned presents a pressure-fed cycle, whose injected flow in the chamber can be modified by varying the displacement of a spring-pintle compound. This is accomplished by means of a so-called variable-gain solenoid-valve hydraulic control system, used to adjust that pintle position with two valves. The considered delays in valves operation concern current-raising and piston-moving times. Seven states are selected for the nonlinear model: chamber pressure, gas mixture ratio, injected mass flows (two), and pintle's displacement and velocity. Combustion chamber follows a single-time-delay model, in which homogeneous mixing and ideal gases are assumed and droplets volume is ignored. CC differential equations concern mass conservation and mixture ratio differentiation, which is a rare approach similar to [96], a modelling reference without control purposes.
KOLCIO *et al* [66] developed a procedure to define dynamic models of liquid-phase systems, such as LPREs, for control and monitoring purposes. Partial differential equations of fluid flow through a control volume present the assumptions of quasi-one-dimensional, compressible and viscous fluid flow. Whenever the governing conservation equations (continuity, momentum and energy) render

the model too complex, empirical steady-state input-output maps are incorporated. These equations are then non-dimensionalised and this way a term expressing the speed of their dynamic response appears. Depending on its order of magnitude, transient behaviour can be neglected and hence model's order can be reduced. The same modelling procedure is applied to all fluid components, but a different discretisation is applied depending on the type of component. Discretisation is required to obtain nonlinear ODEs and reduce model's order, since spatial contributions are considered as linear. Individual component models, whose states consist in Mach numbers, pressures and temperatures, can be joined to obtain a global system model. Turbopumps are also considered, and present an empirical formulation. This modelling approach is said to be specially appropriate for simulating anomalies or faults, but no comments on control inputs or on any example of a global system model derivation are made.

The control modelling approach by YANG *et al* [168] can also be mentioned. A rocket engine is modelled in a nonlinear way as a feed-forward neural network made out of radial basis functions, adequate for real-time monitoring, diagnosis and control. Artificial Neural Networks (ANN) consist in an adaptive logic system based on the structure of the brain. They consist of several layers of *neurons* in which a series of weighted, interconnected additions serves to represent input-output relations. The weight of each interconnection is assigned during a training phase, in which the network is confronted to real or simulated data of the process that it is supposed to mimic. ANN may present problems in cases where inadequate actuation triggers dramatic failures, which are specially problematic during the training period, when the algorithm is learning the response to diverse actuations. Hence a pre-training of the network by means of simulation data is generally advantageous [129]. LE GONIDEC 2017 [75] also employs a nonlinear ANN representing the engine, although it is only used for estimation. Concretely, its static part is considered in the network, which is trained via databases and tests to provide correct mixture-ratio estimations.

The plant model by DAI AND RAY [37], in the same research group as LORENZO *et al* [88] (NASA Lewis/Glenn Research Centre), was obtained by translating the typical thermo-fluid-dynamic partial differential equations of conservation to first-order ODEs by means of standard lumped-parameter methods. Twenty states, two inputs (oxidiser valves) and two measured variables were attained. Structural and damage models are explained in greater detail in other companion papers [36, 124, 166]. This is the only nonlinear model found, on which a control algorithm has been designed (even if it is in OL), as explained later in Section 2.4.

2.3.2 Sensors and actuators considerations

Sensors. Many sensors are installed in LPREs, but their location and redundancy are usually limited by physical constraints or risk considerations. Thus, real engines and test benches often present slightly different sensor configurations. Important variables for [55] to be measured are the following. Measured states can be combustion temperatures, cavity pressures, turbopumps speed and some mass flows. Concerning control inputs, valve positions can be measured; and sometimes there are other physical parameters or quantities relevant to determine, such as fuel and oxidiser temperatures or vibrations. Classically, the sample rate is chosen in accordance with the measurement type and the application to supply (i.e. safety, monitoring or control).

For measuring valve position in terms of angles, the rotary variable differential transformer (RVDT) is usually employed, while in terms of linear motion, the linear variable differential transformer is used

(LVDT). Especially in reusable engines, vibrations should be monitored during operation so as to identify bearing wear or fault. Radial accelerometers placed on the pumps transmit to bandpass amplifiers which read the real-time signal. New bearing-wear sensors and real-time tracking filters are expected to furnish precise vibration measurements. According to [55], in order to mitigate the impact of random vibration, a narrow-band tracking filter is a good option since it measures synchronous and harmonic vibration.

Temperature sensors generally consist in thermocouples and in resistance temperature devices (RTD) [55]. Each RTD requires a bridge termination network with noise filtering and each thermocouple needs an additional gain. It is generally required that the sensors be robust and resistant to high temperatures and pressures, but the main CC temperature is usually too hot to be measured, and only GG or preburner temperatures are measured. This issue, together with the lack of some mass-flow measurements mentioned in the next paragraphs, can lead to partial observability, generally compensated by introducing estimators. Rotational speed sensors are generally of variable-reluctance type, comprising a permanent magnet and an independent pole piece surrounded by a coil winding of thin-filament magnet wire. These sensors can be used for control feedback but their main use is turbopump (TP) redlines monitoring.

The SSME presented 80 sensors collecting measurements at $50Hz$ including the redundant ones [142]. Those related to the control loop performance were tripled and those used for limits monitoring were doubled. Chamber pressure sensors were used to compute thrust, as logical, and volumetric-flowmeters were used to compute mixture ratio, after correcting with density calculations based on pressure and temperature.

Regarding the recommended sensors in [129], indications on their time-response features are given. They may present two distinct response time-scales, a faster one for feedback and a slower one for the possible adaptive filters. For dynamic pressure measurements, piezo-resistive or piezoelectric transducers are often used due to their high sensitivities and high natural frequencies, able to cover the large pressure range in LPREs.

In ArianeGroup's reference [65], the sensors required for their control loop are mentioned: flow-rate sensors are installed prior to the chamber, apart from the typical pressure sensor inside the chamber. A divider module, with protection against division by zero is used to compute the mixture ratio from the flow sensors. This is then sent as feedback, as well as chamber pressure measurement.

When dealing with test-bench control, ArianeGroup usually define mixture ratio at pump inlet, instead of the chamber-inlet one [76]. In this manner, the ratio between the pumped mass flows is established, yielding an expression of engine's global performance. This comes from the easier installation of flowmeters, typically of differential-pressure type (Venturis and orifice plates), at pump inlets in test benches. But in actual engines in flight, most mass flows are normally not measured. Hence, estimators have to be used (different solutions, for instance neural networks [75]), which vary from one engine to another. As an example, the number of sensors in the *Vinci* test bench is: two chamber pressure sensors and twenty-seven sensors devoted to computing mass flow from volumetric flow, temperatures and pressures (with redundancies). The estimation diagram by [115] is schematised in Figure 2.3.1, in which mixture ratio is estimated and thrust is calculated.

Sensor dynamics in [109] are of first order, representing the HPT discharge temperature sensors. In [117], strain-gage sensors, modelled as second-order systems, are used to measure pressures and engine thrust on a test bench.

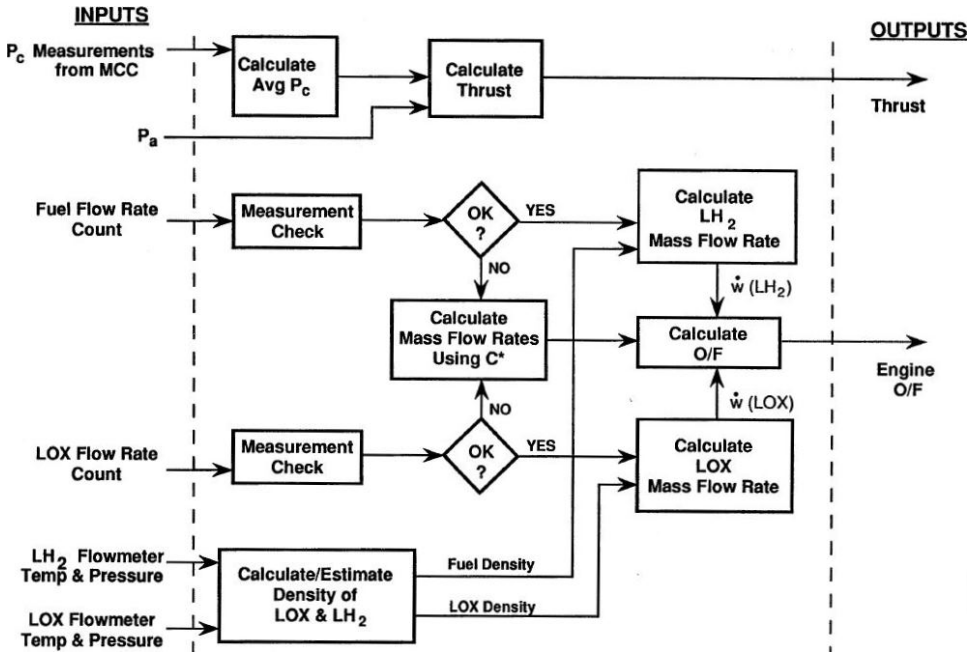


Figure 2.3.1: Mixture-ratio estimation and thrust calculation diagram by NEMETH *et al* 1991 [115], where O/F stands for MR, P_c for combustion pressure, MCC for main combustion chamber and P_a for ambient pressure

Actuators. The principal actuating devices in rocket engines, valves, can generally adjust propellant flow or activate a bypass duct. Their internal actuators varying their opening angle can make use of electric motors or of hydraulic or pneumatic power. As commented in [77], nowadays these servo-controlling actuators make use of sophisticated electronics in order to deliver fast actuation speeds which allow an effective control. The drawbacks of these tunable valves are large bandwidths and elevated energy consumption, forced by their internal servo-control loops. An alternative proposed in [77] is the use of bang-bang actuators not requiring expensive electronics. However, electrical actuators are gaining importance in valve-position control with respect to traditional pneumatic ones. The main reasons are that auxiliary helium-gas consumption and costs can be reduced and throttling efficiency is improved [33]. That is why the new hardware-in-the-loop (HIL) simulation platform by CNES and ArianeGroup [127] includes real valve-internal electric actuators. Since no real fluid lines are considered, the resisting torque on their hypothetical wet surface, equivalent to the real hydraulic one, is performed by another coupled actuator.

In [131] it is suggested to analyse the sensitivity functions relating mixture ratio and thrust with pressure drop and opening angles in control valves. The outcome of that analysis influences the selection of valve types (butterfly, shut-off, needle, ball, etc.), which must meet different kind of requirements, such as the characteristic relation between flow rate and pressure drop, actuation efforts, controllability, cost, weight, etc.

The SSME presented hydraulically-actuated valves, which in the event of a failure, were actuated by pneumatic elements rather than by the controller [142]. Reference [55] recommends that solenoids and drivers designs should include built-in fault detection checking voltage and/or current.

Actuator dynamics in [109] are defined by a servo motor with second-order dynamics, a first-order piston linkage and hysteresis (four states). Electrohydraulic servovalves of plug-type are used in [117].

Bypass-valves dynamics in [71] are modelled as a second-order system, whose gain depends on opening position (via a first-order polynomial).
Some examples of actuators highlighted by [129] to counteract combustion instabilities consist in compression drivers adding acoustic energy at certain frequencies, flow injectors adding secondary mass flow, large valves modulating the total mass flow, mechanical devices altering boundary layers, fluidic devices adding momentum to the flow and special electrodes initiating current-stabilised electric discharges.
Other family of actuators present in LPREs are igniters, used to initiate combustion in chambers. In European GG cycles, they are typically of pyrotechnic type, but spark-based ones have also been used elsewhere. They are generally not considered in control models since their effect consists in a discrete event only taking place during the start-up or re-ignition transients. Apart from the fact that not many transient control studies [115] have been carried out, this discrete actuator is not normally considered as a control input, mainly because the system would become hybrid.

2.3.3 Analysis

Concerning the stability analysis, as mentioned in [155], rocket engines operation tends to be naturally stable, mainly because the liquid flow system counteracts disturbances and off-design behaviour. The system naturally tends to regulate itself. Nevertheless, some internal natural resonances, mostly related to TP rotation, may present frequencies that could destabilise the system. Combustion instabilities may also arise in the absence of adequate passive dampers. Hence, the instability sources can be mostly avoided with an appropriate engine design.
Some stability-regions computation methods are proposed in [133] and [134]: the Mikhailov, Hermite-Biehler and Routh-Hurwitz criteria. Indeed, these authors point to the need for checking whether the typical rocket-engine transient phenomena interact with system's natural frequencies leading to instability. The main physical parameters influencing these low frequency couplings are, according to [134], pressure drop in the injector (connecting chamber and feed-lines oscillations), evaporation delay (to minimise) and combustion-chamber characteristic time or residence time. After joining combustion-chamber and injector equations in the frequency domain, an expression with these three main parameters can be obtained.
The Mikhailov criterion requires the construction of a frequency plot, where stability limits can be drawn. Therefore, this method can become cumbersome for large systems. The Hermite-Biehler theorem (Interlacing theorem) ensures necessary and sufficient conditions for the Hurwitz stability of a real polynomial.
The Routh-Hurwitz criterion, used in [133], can also be employed to compute the stability zones. It represents a mixture of both previous methods, since certain value ranges are selected for each parameter and then characteristic equations are solved. In all cases it is concluded that the system is stable within a wide enough region.
Both simulations and tests in [13] show the presence of reverse reactions, under- and overshooting in the transient of GG outlet pressure. Those initial reverse reactions, common in rocket engines, correspond to the non-minimum phase behaviour in the control terminology. This is mathematically expressed by the presence of a right-hand-plane zero in the transfer function, between valve area and outlet pressure in that case.
Regarding the essential controllability and observability analysis, NEMETH *et al* [115] describe the

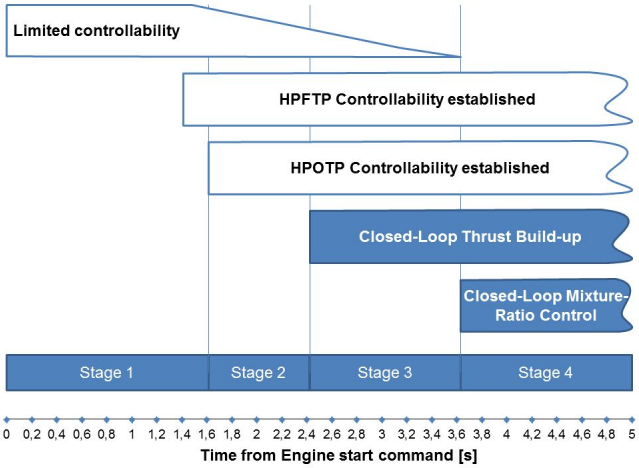


Figure 2.3.2: SSME start-up stages of controllability [115], HPFTP and HPOTP are high-pressure fuel and oxidiser TP

different behaviour stages of a LPRE during its start-up transient. Those authors clearly identify four stages of controllability and observability, schematised in Figure 2.3.2. The first stage comprises all discrete events (ignitions and valve openings), from the activation of the start command until the engine can be considered as a fully continuous system. The second stage consists in the transition phase between the end of discrete events and the start of CL control, where the main thrust build-up occurs. In the SSME case, stage 3 starts with combustion pressure CL control and the fourth one begins with mixture-ratio CL control. It is highlighted that the stages on which control should be improved the most are the first two, where the most damaging transient phenomena may occur. The SSME is said to be hardly observable or controllable during stage 1, as well as to contain model-plant mismatch. Different control alternatives were tested for controlling the start transient in [115], but those authors did not manage to obtain positive results due to these observability and controllability issues. In fact, they had to assume that these properties were met, although in reality it is very difficult to measure some quantities during the first stage of the transient, especially for oxygen. For instance, designing a flowmeter that detects small hydrogen flows at low pressures, which are very different from steady-state ones, is not a trivial task. In addition, those reduced flows at the early phases limit controllability. However, it would seem somewhat possible to measure and control on the hydrogen side because it remains monophasic (supercritical), while it seems impossible to those authors on the two-phase oxygen side. TP speeds are accurately measurable during start-up but are hardly controllable before their respective preburner is filled, which allows the management of the energy transmitted to the turbine. All these constraints led those authors to reject CL control during the discrete-events stage of start-up as a feasible approach. An ad-hoc approach expressed through an influence model was also tested, relating the deviations in observable states to the necessary valve opening counteraction. Nevertheless, no action is carried out on unobservable states, which in the end can result in greater damage after anomalies. Due to these negative results, [115] proposed to modify SSME configuration to allow observability and controllability during the discrete-event phase. For instance, lines could already be primed prior to start-up.

2.4 Review of control methods

With regard to the actual control techniques employed in the reviewed references, they can be classified into diverse categories: open-loop, conventional CL, nonlinear, robust, hybrid and reconfigurable control approaches.

2.4.1 Open-loop control

Open loops are generally preferred in space applications due to their simplicity. Since conventional rocket engines are naturally stable in flight, if neither high performance nor robustness levels are expected, OL can be a valid option. However, if high performance is sought or if the application is special, it may be a limited method since it is normally impossible for the input to know the transient response of the system. Even if there is a constant valve opening, the external conditions may alter the operating point.
An OL strategy applied to combustion [129] contains a controller accompanied by an actuator powerful enough to modify the combustion process. Common control actions are oscillatory inputs, which have achieved instability suppression, combustion enhancement and emission reduction.
Another OL (feed-forward) control algorithm within the project of LEC of SSME was developed by DAI AND RAY [37]. This approach, one of the most relevant in the literature, consists in an optimal policy to control the rocket engine along a nominal trajectory during its throttling transients. Off-line optimisation is performed on a plant model taking into account structural and damage dynamics as well as constraints of fatigue and creep damage, so as to extend critical components' life.
The outcome of the algorithm is an optimal control sequence with two functions: making a trade-off between performance and damage and identifying possible conflicting requirements posed by damage mitigation. This control sequence driving the system from an initial equilibrium state ($186bar$) to another one at a higher pressure ($207bar$), considers a quadratic cost functional: the square of the weighted sum of L_2-norms of states, outputs, control action (only two preburner oxidiser valves), deviations from final state and damage states derivatives. Nonlinear Programming (NLP) was used to optimise this cost functional of plant's performance without violating damage rate and accumulation constraints. The most important output to the authors is mixture ratio, which must be kept constant during the $300ms$-long operating-point shift. Time steps are gradually incremented from $\Delta t_1 = 3ms$ in order to represent better the initial quicker dynamics with a Runge-Kutta scheme, obtaining a total number of 37 steps. Results were very positive and pointed to a possible extension of main thrust chamber's life due to the imposition of those constraints.
KIFORENKO AND KHARITONOV [63] developed an off-line optimal control strategy for controlling the thrust of a LPRE. The control problem is considered from the perspective of the whole launcher vehicle, whose thrust needs to be optimised. This is indeed one of the crucial problems of rocket flight mechanics. Thrust is maximal when nozzle's outflow velocity is so, reached at an optimal mixture ratio, dependent on engine's parameters and assumed constant. A simple model of engine's global behaviour is chosen: an algebraic expression relating thrust to injected propellant mass flows into the main chamber links the controlled variable to controls. Differential equations consist in the general equations of motion of a launcher vehicle (centre of mass). These involve propellant mass consumption, dependent on mass flow rates; and acceleration, dependent on thrust. Rocket's full

mass evolution, including engine-system mass, is considered in that relation between acceleration and thrust.
Control actions consist in both main propellant mass flows, on which some constraints are imposed: a minimum and a maximum mixture ratio. Chamber pressure, also related to thrust via an algebraic formula, must not exceed an upper value. The admissible domain of controls is set according to those constraints. Fuel mass flow is expressed as a function of oxidiser mass flow via second-order polynomials. This way, only one control variable can be considered. The lower and upper value of mixture ratio multiplied by oxidiser mass flow are logically straight boundaries in the domain.
The optimal-control objective is reaching a maximum velocity within the allowable control domain, expressed in terms of oxidiser mass flow. An indirect method via PONTRYAGIN's optimality principle is employed, where the Hamiltonian function contains three adjoint functions multiplying acceleration, fuel mass flow and oxidiser mass flow respectively, all expressed as functions of the latter. Equations are rendered non-dimensional. At the initial instant, no optimal solution can be attained within the allowable domain, but control increases during operation and optimality conditions (maximum of Hamiltonian function) are met. Besides, only one control meets those conditions at each time instant in the appropriate intervals. Better results than traditional control are attained, but engine's internal transient behaviour is neglected.

2.4.2 Conventional CL control

The first step to enhance the robustness and performance of an OL is to add sensors, desirably to measure pressure or temperature fluctuations [129], and close the loop with a feedback. The frequency response of the closed-loop sensor should be higher than the operating frequency of the controller and actuator.
By implementing CL, the number of engine tests can be reduced, a steady state can be maintained and the overall launcher trajectory can be better predicted. Consequently, one can be less conservative in terms of propellant mass.
The analogue approach by OTTO AND FLAGE from NASA 1959 [117] considers a 3×2 MIMO (multi-input multi-output) system (chamber pressure, fuel flow and oxidiser flow related to two main valves). Their main objective was the classical one in control: attaining zero voltage difference between measurements and a reference, all being defined in volts within the loop. The MIMO is decoupled into three SISOs. The oxidiser control valve is in charge of controlling mixture ratio (first SISO), while the fuel control valve controls either the fuel flow itself or chamber pressure (two alternative SISOs). Indeed, the fuel-flow loop (based on fuel-flow feedback) is only active during the fuel-lead phase of the start-up, when there is no oxygen. Then, the fuel control valve switches over to participate in the chamber-pressure loop, and the oxidiser-flow loop is activated, as it can be seen in Figure 2.4.1. It is mentioned that the opposite decoupling is also possible, but less safe. In order to obtain propellant-flow measurements, pressure drop in each feed line is measured (with Venturis) and compared to references. Chamber pressure is also measured. So as to meet the maximum mixture-ratio requirement, the speed of the pressure loop was deliberately reduced. This way, the oxidiser valve could adapt quickly to changes in fuel flow and hence avoid ratio peaks. Controllers in the previous study are of PI type with small gain (less than 1), also during start-up. Step changes in mixture ratio reference are well tracked, with 15% overshoot. These set-point switches last around $1.5s$. Delays during start-up, especially in chamber pressure rise after valve opening, reduce control

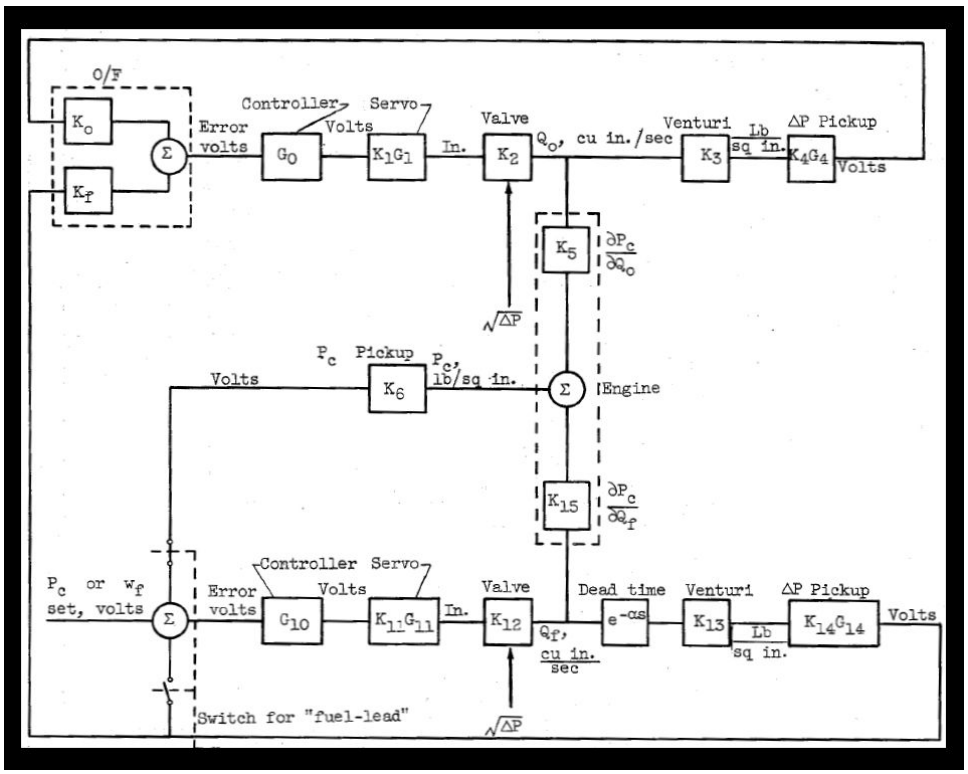


Figure 2.4.1: Control system schematic by OTTO AND FLAGE [117]

system's accuracy. Besides, a constant pressure reference is provided from the beginning, resulting in the aforementioned overshoot.

A simple and robust method to regulate a rocket engine with at least one combustion chamber is presented in ArianeGroup's work by KLEIN *et al* [65]. It is particularly applicable to expander cycles but also extendable to GG ones. The opening command to chamber feed valves is computed in OL, coming from an external order. The command to turbines' regulators device (bypass valves) is calculated in CL. The overall performance is satisfactory while keeping computational demand low.

A tracking filter (first or second order) is applied to the external reference of chamber pressure and mixture ratio, coming from predefined profiles, ground order or calculation by the electronic control unit according to flight data. With this filtered command, valve orders can be computed as fourth-order polynomial functions of the ratio between the filtered chamber pressure command and its maximum value allowed. But other alternatives are possible, such as artificial neural networks. The controller consists in a dual PI corrector, with a higher cut-off frequency than the tracking filter.

The previous reference is one of the multiple studies which ArianeGroup have carried out in the last decade concerning the control of expander-cycle engines [72, 75, 77]. In reference [77], combustion chamber pressure and mixture ratio are considered measured and are adjusted according to set-points, by means of a bang-bang control strategy applied to slow actuators. This way, nonlinearities are said to be introduced in the control strategy for avoiding over-demand on the electric motor. A minimum threshold on the set-point tracking error is set too. The continuous controller itself can simply be an integral action. However, the command sent to actuators, which consists in power supply pulses, is quantised (discretised) and presents variable durations.

In [72], the same control goal is targeted, but it is solved differently while complying with operating

limits. Firstly, the set-point signal is again filtered via a first-order filter. Then, a multivariable command with a predictive internal model is employed. This approach calculates the control necessary to reach a future objective while knowing a prediction of system behaviour within a given horizon. However, this linear off-line-tuned controller cannot be classified under the standard optimisation-based predictive control methods, such as MPC. In order to comply with engine limits, the control module tunes a control gain factor according to the lower of the differences between measured shafts rotation speeds and their operating limits. Measurements of that variable and of mass flows and chamber pressure, as well as the filtered set-point, are provided to the controller.

Reference [75] goes one step further and considers that mass flow measurements and hence mixture ratio measurements are not available. They are estimated from valve control signals and measured pressure. Indeed, as stated in Section 2.3.2, it is usually costly to insert mass-flowmeters in engine's lines. The engine is mainly controlled according to chamber-pressure reference, and then valves commands variations are adapted to ensure a desired mixture ratio.

In the state-space framework by [174], a particular variation of injected flow is considered, by means of a spring-pintle system whose position is altered by two valves. The feedback signal is the measured combustion pressure multiplied by a gain. The value of that gain is affected by a combination of effects: valve's displacement, orifice diameter, pressure difference and pulse width. They concluded that the three main control parameters which influence the dynamic response time and control precision of such an engine are: pulse width, the operating frequency of solenoid valves and the deviation from their critical operating points. Valve's frequency is the main contributor to the sampling frequency of the whole control circuit. It represents controller's ability to track the chamber-pressure reference. Its augmentation reduces overshoot. These parameters are tuned so as to obtain the most convenient engine behaviour. All in all, they include a term in the definition of valves' mass flow related to these parameters, utilising an empirical correlation to characteristic times equivalent to a first-order system.

A bipropellant variable-thrust rocket engine is also controlled by means of solenoid valves and variable-area injectors in the work of ZHOU [178]. The control system comprises a control circuit, two solenoid valves, two variable-section injectors and the thrust chamber. The control circuit contains operational amplifiers that receive the reference voltage signal and combustion-chamber pressure feedback signal, and output a deviation signal. Control action is exerted by valves, that indirectly let injectors undergo section variations according to pressure, by means of springs. This type of valves are pulse operational. Thus, control gain depends on a selected pulse width divided by the error voltage signal. Pulse frequency is set between 50 and $100Hz$. The compact transfer function modelled can be used to demonstrate stability for a given combination of parameters and control gain. Some issues regarding steady-state error at high thrust are mentioned.

The most recent publications dealing with CL control of LPREs come from the Japanese space agency JAXA. The *LE-X* engine by this agency is designed with high reliability and low-cost objectives and presents decoupled automatic control of mixture ratio and thrust by means of electric-actuator valves, enabling a smooth throttling, as presented by SUNAKAWA *et al* [154]. These valves are: thrust control valve (TCV), controlling thrust via the flow rate of turbine output gas; main oxidiser valve (MOV), controlling mixture ratio via the oxidiser pressure at chamber inlet; and main fuel valve (MFV), employed during throttling so as to maintain turbine inlet temperature at reasonable levels.

The control strategy was designed thanks to subscale valve tests and transient simulation. A two-degree-of-freedom control is applied: an estimation $P'(s)$ of the actual plant $P(s)$ is made. A non-described controller, whose denominator is $P'(s)$, is designed to determine the output transient

response. However, since $P'(s)$ is not perfect, a PI controller is included to correct the error. Ramp and step responses were tested at the three throttle levels (0%, 60% and 100%). Simulations presented no overshoot in thrust or mixture ratio but the chamber coolant temperature simulation at ramp response had it to an important extent, which may represent a big heat stress source. Thus, the step command was chosen for control. In order to verify and refine the control method, subscale valve tests were carried out with gaseous nitrogen.
The control algorithm for JAXA's current designs of reusable expander-bleed engines, by KAI *et al* [60], consists in a combination of simple feed-forward and feedback controls which achieves thrust tracking at steady-state. Three electrically-actuated ball-type valves are used for control. All three are of continuous nature and their effects on the controlled quantities are again decoupled in three SISO loops: on thrust, mixture ratio, and turbine inlet temperature, as in [154]. These three controllers present feed-forward at $100Hz$ according to a predefined valve position table for each level of thrust command. All three also receive feedback measurements of pressure, temperature and flow rates. The thrust loop contains a PI controller at $25Hz$ while mixture ratio and turbine inlet temperature only present threshold feedback control at $0.5Hz$. PI parameters were tuned according to step and frequency response test results.
Results after the start-up phase of the engine are positive, although with some overshoot. A constant steady-state thrust command is fed from the beginning. Those authors affirm that the engine can throttle between 21% and 109% at test bench with their controller, which is a major accomplishment in the current context seeing the high technical difficulty of down-throttling.

2.4.3 Nonlinear control

The only reference considering nonlinear techniques in CL is the mixed linear-nonlinear approach in [89], where two main loops are defined in the control system: an inner one for the performance controller linked to the linearised plant and an outer one for the nonlinear damage controller. This controller is designed on the combination of the performance controller and the plant model. Its main elements consist in a structural model for load conditions, a time-domain damage model, determining the damage rate based on the load estimations; and a damage controller, mitigating damage rate and accumulation at critical conditions, normally in transients.
On the inner loop, an H_∞ controller (optimal L_2 norm) is tuned. Its performance weights penalise outputs' tracking errors and control signal weights penalise valve motion, in order to minimise valve wear and to avoid large oscillations in the feedback control signal leading to valve saturation. The initial order of the controller is fifteen, but it could be reduced to five.
Damage modelling is carried out in continuous time and represents fatigue damage in the turbine blades, assumed to happen due to tensile loads and not to temperature. The outer control loop consists in a nested connection of the three elements mentioned before: structural estimator, nonlinear fatigue model and a linear dynamic filter as damage controller. The parameters of the latter are identified off-line by minimising a cost functional based on nonlinear optimisation, through Sequential Quadratic Programming (SQP). This functional is tuned first in a certain simulation, a representative pressure ramp-up, and comprises the effects of both reference tracking performance and blade damage. Interaction between the damage and performance controllers is mitigated by introducing the high-performance requirements in the cost functional and by the fact that both loops present different frequencies (higher for the outer one).

Results point to a slower pressure response with less overshoot due to the inclusion of the damage controller. Mixture ratio deviates excessively during the transient but reaches the desired steady-state value. Damage rate in the hydrogen turbine is somewhat too elevated, and checked to be higher in a more demanding simulation. But without the controller, results are absolutely not tolerable since damage in blades quickly gains two orders of magnitude during start-up, which reduces engine's life dramatically.

2.4.4 Robust control

Robust control is another important field of study in this application with some uncertain parameters, varying during engine execution or from run to run. As observed by BARS *et al* [6], current innovative robust studies in general deal with probabilistic robustness. Concretely, randomised algorithms can bring about a reduction in the computational complexity of classical robust algorithms and in the conservativeness of H_∞ techniques. However, as explained in the next paragraphs, only the latter have been applied to LPREs, as well as KHARITONOV's theorem.

Robust control theory is applied to linear models of LPREs in the work of SANTANA *et al* [132], considering uncertainties in the plant coefficients. Robust stability with respect to parametric uncertainty is ensured by means of the generalised KHARITONOV's theorem (based on [16]) and it is concluded that robust control is capable of estimating the range of the required stabilising coefficients.

In fact, according to [132], conventional numerical methods for analysis are not adequate for uncertain models, requiring long computational times. Transfer functions with uncertain parameters would represent an easier solution. KHARITONOV's theorem involves interval polynomials $p(s)$, whose coefficients are defined as intervals (assumed independent). Every polynomial in the family of $p(s)$ is Hurwitz stable if its four extreme polynomials are so. These extreme or KHARITONOV polynomials present different combinations of extreme coefficient values. The application of this theorem to the characteristic polynomial of a plant with uncertain parameters like a rocket engine renders a necessary and sufficient condition for stability analysis.

Two uncertain parameters are selected for the robustness study in [132]: combustion chamber's time delay and time constant. These have the highest influence on the low-frequency combustion instabilities ($50 - 1000Hz$), generated by the interaction between the feed system and the combustion process. Hence, this frequency range was selected for the analysis. The delay can be estimated by the CROCCO's relation and the time constant represents the time that the gas remains in the chamber. BODE and NYQUIST envelopes were computed by means of the *PST* and the *RPC MATLAB®* toolboxes, confirming stability in the selected frequency range. The robust step response (in *PST*) demonstrated acceptable dynamic performance at the chosen parameter intervals ($[1, 5]ms$ for the delay and $[2, 5]ms$ for the constant).

The H_∞ robust-control approach, explained in 3.3.2, is applied by LE FUR *et al* [71] so as to throttle an expander-cycle rocket engine between 50 to 110% of nominal thrust in simulations, with a mixture ratio varying between 5.5 and 6.5. Simulation at different operating points (low and high chamber pressures) gives very dissimilar behaviours in the outputs, which leads those authors to affirm that a unique controller is difficult to synthesise. For instance, the solution for a given point destabilises the system at another regime. Hence, three different H_∞ controllers are synthesised at three distinct operating points. Loop shaping is performed via pre and post-compensation. Performance is set as a low-frequency constraint and robustness as a high-frequency one, as usual in this kind of techniques.

However, loop shaping does not guarantee CL stability; the gain slope in the vicinity of the cut-off frequency can become too steep. In the H_∞ framework by [71], this is corrected by means of a PID-based controller, that contains both a phase-lead controller to enlarge robustness margins by increasing the phase and avoiding high gain slopes, and an integrator to cancel the static error.
Several criteria are used to determine the controller and the weights of compensators: stability, input-perturbation rejection, steady-state error, actuators demand and bandwidth. Each one presents a different H_∞ -norm minimisation objective. Results show that it is more convenient to put the controller in the feedback loop so as to reduce overshoot and actuators saturation. Those authors propose gain scheduling for future steps, and affirm that a unique H_∞ controller would be feasible if some modifications were carried out on the engine, so as to have less different operating points.
The compromise between performance and robustness is again mentioned in the report by SAUDEMONT AND LE GONIDEC [141], who also developed a robust H_∞ control, this time on the *Vulcain* engines. Basically, greater gains are tuned at low frequencies to improve performance (settling time specification) and low gains are selected at higher ones to remain stable and robust.
It is a fact that the behaviour of the rocket engine varies as a function of the operating point (thrust and mixture ratio). Therefore, the system here consists in a Linear Parameter-Varying (LPV) system (introduced in Section 3.3.2), considering that some parameters vary with operation changes. Those uncertain parameters are expressed here by means of the Linear Fractional Transformation (LFT), used in the H_∞ synthesis. This expression of uncertainty makes use of parameters mean values and maximum amplitudes, multiplying the so-called uncertainty variables, defined between -1 and 1. The matrix model of this engine presents three inputs (GG input valves VGH and VGO and turbines flow distribution valve VGC) and three outputs (GG temperature, chamber pressure and global mixture ratio). The three output sensors, represented by first-order transfer functions with uncertain time constants, are also added to the model. A delay is also joined to their outlet, expressing the processing time required by the calculator, uncertain too. All these elements are expressed in superior LFT and linearly joined in a single matrix. For this sake, engine's matrix had to be decoupled by means of the structured uncertainty matrix. Mean values and maximum amplitudes are considered fixed. Hence, the resulting system is only dependent on the uncertainty matrix (a function of uncertainty variables), which is of dimension nine.
Regarding H_∞ design choices, the post-corrector or post-weight is chosen as the inverse of system's nominal matrix (without deviations), to compensate the static gain introduced by that matrix. The pre-weight is tuned as a set of integrators with different gains to ensure a bandwidth of $2rad/s$. The controller is obtained via the normalised coprime factorisation synthesis, available in the μ-synthesis *MATLAB*® toolbox. It is based on the problem of robust stabilisation of coprime factorisations and generally computes a high-order controller (eighth in this case). With this configuration the system is able to reach the reference signal and reject perturbations acceptably well.

2.4.5 Hybrid control

Systems consisting of both discrete and continuous characteristics at the same time are called hybrid systems. The basic phenomena in typical control problems are naturally continuous, normally defined by smooth functions. But if at some points, abrupt variations related to behaviour changes appear, discrete variables may come into play. This way, the different physical modes of the system can be modelled, usually in the form of piecewise continuous or affine systems. Other types of discrete

features can be discrete outputs (sensors), inputs (actuators) and discrete-event controllers. An example of discrete or discontinuous control can be gain scheduling. Hierarchical control structures such as the ICS proposed by [88] can be very appropriate for models with large uncertainties, noise and disturbances, according to [6]. These structures might comprise a set of controllers, estimators and generators, forcing the need for a switching logic to decide which ones are executed. Indeed, the adequate controller is selected according to some rules defined in the switching logic, related to parameter uncertainties ranges. This logic determines in the end the global stability and performance of the closed-loop, transforming the system into a hybrid one due to its discrete dynamics. For [6], these schemes are an attractive alternative to typical continuously tuned adaptive controllers because they reduce conservatism and enhance stabilisation and control performance, especially during transients. However, switching strategies are not the only hybrid approach; hybrid control is still an evolving field.

Only one reference concerning a subtype of hybrid control applied to an LPRE aspect has been identified. ZHENG *et al* [175] employed a variable-structure control for stabilising combustion in LPREs, which can be considered as an unstable time-delay system. These authors affirm, that if this system is stabilised by linear state feedback, robustness against parametric variations is not guaranteed. It is shown that a switching functional should be used in variable-structure controllers for systems with state or control-inputs delays. Basically, stable sliding modes are derived, in which the delayed system is transformed into a non-delayed one thanks to characteristic matrices.

The CROCCO monopropellant combustion model [35] is considered in that previous work, where two parameters can vary: reduced time lag and pressure's exponent (pressure dependence of combustion), difficult to measure. The former is the ratio between time lag and gas residence time in steady operations. The control variable taken, supposing a unique, regulated injected flow with no concrete valves to control, is the ratio between the pressure drop in the line before the injector and two times the drop in the injector. Results point to stability and better robustness results than with the linear feedback.

In a posterior paper, ZHENG AND FRANK [176] prove the robust stability conditions on general uncertain distributed delay-system and apply it to the same combustion model. Another stabilisation method based on LMI (Linear Matrix Inequality) is tested, said to be more efficient and not to require parameter tuning. This way, allowable parameter-varying ranges can be determined.

Furthermore, some elements related to hybrid control, like switching, may be extracted from the SSME reconfigurable control approach by MUSGRAVE *et al* [109], classified in the 2.4.6 category, but a concrete hybrid framework applied to LPRE systems (not only combustion) is missing.

2.4.6 Reconfigurable control

In general, conventional feedback control engineering may not be sufficient for modern complex systems in terms of performance. New strategies which tolerate failures while keeping stability and performance features are arising. According to ZHANG AND JIANG [173], thanks to fault-tolerant control systems (FTCS) reliability and availability can be increased, adapting to component malfunctions automatically. These systems involve Fault Detection and Diagnosis (FDD) and/or Fault Detection and Isolation (FDI) mechanisms. Research on these approaches was motivated by the world of civil aviation, with the aim of providing self-repairing capability for performing a safe landing. But nowadays it covers a wide range of applications, such as in aerospace, nuclear power, manufacturing,

etc. Owing to the complexity to the problem, FDD and Reconfigurable Control (RC) have historically been studied separately. That is the reason why some techniques for the former do not match the ones for the latter, since it is assumed that the counter-part is perfect. In this control section only the RC part is covered.

Traditionally, reconfigurable control has not been systematically used in propulsion systems, since not many engines contain redundant actuators. Instead, these are designed with a high degree of reliability. Nevertheless, in rocket engines, a fault in a main valve could imply a catastrophic failure. Reliability is a key aspect in reusable systems, and hence the Health Monitoring System (HMS) plays a leading role in the global system design. This system should diagnose faults difficult to detect, owing to uncertainties or sensor noise. As pointed out in [55], the interface engine-vehicle includes not only commands like start, stop and thrust-level modifications, but also health-monitoring data, critical to attain the required redundancy level to accomplish the mission objective.

MARCOS *et al* [97] present an architecture allowing the interaction between HMS and future control systems in LPREs, in the frame of the Future Launcher Preparatory Programme by ESA (2012). The tasks of diagnosis, prognosis and decision on the HMS side, and of management, reconfiguration and sequencing on the control side, are interrelated. The criticality of this interaction component is deemed as high as the one of HMS alone, and seems necessary for ensuring the development of reliable and robust fault-tolerant health-management systems. HMS information and engine status are managed in order to feed the controller with all abnormal behaviours. In fault cases, a gentle performance degradation (for instance a modification of mixture-ratio reference) could be attained with this architecture, instead of a brusque engine shutdown. For this sake, those authors envisage to use control techniques that allow fault accommodation or reconfiguration. Sequence reconfiguration via optimisation and engine parameters re-estimation are highlighted among others.

It is relevant to comment again on the work by MUSGRAVE *et al* [109]. Their reconfigurable controller comprises a command generator (synthesising the reference commands), several linear control designs and control blending. The valve fault accommodation process is a hybrid approach using gain scheduling and mode switching. The higher the number of accommodatable fault modes which the nominal robust controller cannot address, the higher the number of linear control alternatives. The authors consider that the nominal design cannot deal with the freezing of the FPOV, responsible for keeping a constant mixture ratio during throttling. In that event, the objective of the controller is to decouple MR from p_c using the remaining valves.

That main controller is synthesised using a robust method applicable to servo-compensators (LQG/LTR). This robust technique, explained in an earlier paper by MUSGRAVE [108], is capable of attaining good trajectory tracking and robustness to unmodelled dynamics and disturbances. However, it requires plenty of sensors for that purpose, and a trade-off has to be done between robustness and noise rejection. But this multivariable servo design is proven to be adequate for the SSME, having a stable open loop. A linear time-invariant system is considered. States are estimated via measurements and inputs using a Kalman-Bucy filter. The accommodation controller is obtained by eliminating the column related to the faulty valve and keeping the same dimensions for the gains. This design maintains safe execution over some degrees of degradation at an acceptable performance without gain scheduling.

Both the nominal controller and the linear alternatives (PI) are run in parallel during some time so as to induce a smooth transition. That transition duration is determined by the blending rate, $2Hz$ in that paper. The engine level coordinator is in charge of accomplishing MR and F requirements as close

as possible while keeping away from engine shutdown. The authors have defined a correlation for this sake, relating the maximum thrust possible for a certain position of a frozen FPOV without falling into disturbing MR-deviations. Mixture ratio is kept to 6.01.
The nominal transient demonstrates excellent tracking, although some degradation after the FPOV fault can be seen during accommodation. The maximum thrust is recomputed continuously as the MBFD (Model-based fault detection) scheme estimates the valve position at a high accuracy. However, there is a trade-off between the speed of convergence of the scheme and the need for quickly detecting an actuator fault. This interdependency results in some position estimate degradation (around 3.5%). A proposed solution may be delaying the thrust computation based on the valve position until the algorithm converges.
Another fault-tolerant strategy is carried out by SAROTTE *et al* [140] for the cooling circuit of a cryogenic-combustion test bench. This test bench, *Mascotte* by CNES/ONERA, was developed so as to investigate heat transfer in combustion chambers and jet separation in nozzles in realistic conditions comparable to the *Vulcain 2* engine. In that paper, an FDI scheme is designed for detecting and estimating actuator faults in the subsystem in the presence of disturbances, via unknown-input observers and cumulative-sum algorithms. The goal of the fault-tolerant reconfiguration control is to attain steady-state tracking while compensating the estimated fault and thereby maintaining system stability. For this sake, a transient nonlinear state-space model of the cooling circuit was derived, containing pressures and mass flows and assuming measurements of some of them. The control law, defined after linearising the system around the nominal steady state, contains two main terms: one for compensating the fault, whose magnitude is estimated in the FDI part; and a second term consisting in the reconfiguration element. The gain associated to the latter is chosen so as to stabilise the faulty system by means of LQR. This controller is hence proposed for an equivalent system where the unknown input mass flow is formulated as a function of the known state and inputs. This law was successfully validated in representative simulations.

Reconfigurable sequences. Generally, rocket engines do not readapt their start-up or shutdown sequences as a function of their thermodynamic state; the same pre-programmed one is executed. The engineering novelty presented by LE GONIDEC in ArianeGroup [73] enables the engine to correct those sequences whenever there are variations in its structural or system characteristics, in its thermodynamic state or in the environment. This calculation method selects the instants at which the different operations are carried out before the sequence starts. Computations are concretely performed right after each start or stop.
Those instants are optimised after modelling engine behaviour and including criteria for proper operation regarding the different discrete operations, for start-up and shutdown. The thermodynamic variables taken into account are regenerative circuit's initial temperature, heating coefficient (temperature difference outlet-inlet) and head loss (pressure difference outlet-inlet), since the application example consists in an expander cycle. Valve cross sections and opening durations are taken into account as dimensional parameters. Criteria for proper engine operation concern TP rotation speeds and accelerations (reversal too) and mixture ratio.
In order to calculate the starting cues or event instants, two variants are proposed. For both of them, a system of matrix equations relating the previous variables and the event vector is established. Concretely, the criterion is the dependent-variable vector, the actions are the unknowns and the effects of dimensional and thermodynamic variables are the independent term. The LS method is the first

possibility to be applied to that system, with a weighting factor concerning the criticality level of each criterion. The alternative method consists in developing a predictive module capable of foreseeing the proper operation criteria depending on the thermodynamic and dimensional parameters. Then, a fuzzy-logic module determines the action vector with that prediction.
Results show that similar chamber-pressure profiles are attained when starting from different initial temperatures. However, the build-up takes a little longer. Fuzzy logic, also described in [129], is a multi-valued logic enabling the system's response evaluation by means of fuzzy sets, determined either by making use of expert knowledge or by training. Its working principle is similar to neural networks, but here fuzzy functions and rules are used instead of weighted nodes. This method is very effective if the physical knowledge on the system is deep enough and even more if operational experience is given. Hence it is pretty applicable to complex systems difficult to model, where training requirements may be reduced.

LPRE-FTCS test bench. CNES and ArianeGroup have developed a HIL simulation platform that can include real actuators and controller hardware [127] (ROMET *et al*). It is called ISFM (from French, Engine Functional Simulation Platform) and is intended to allow the testing of control strategies with a certain link to real equipment. Adaptation to plenty of scenarios is possible, putting the focus on realistic failure cases to test FDI and RC algorithms. Qualification was performed on a real-time model of the *Vinci* engine, but other engines demonstrators are envisaged, such as for *PROMETHEUS*. The real-time models used come from more complex ones on which a reduction analysis has been performed, including physics simplification.

2.5 Summary and discussion

Throughout the previous sections, it has become clear that LPREs control involves many considerations, but is generally accomplished by controlling thrust (or combustion-chamber pressure) and propellants mixture ratio (either the global ratio or chamber's one) by adjusting the opening angle of a set of dominant valves while considering several constraints. Apart from these main variables to control, which are traditionally decoupled in two loops, there is a series of secondary loops that fulfil auxiliary tasks, also necessary for the correct operation of the engine (summarised in Section 2.2.1). Each engine and cycle presents its own peculiarities, concerning time constants, internal dependencies and sensitivities, which hinders the reuse of controllers among different devices.

The first and most important question to answer for conceiving a control algorithm for these complex systems is which control goals are expected. The whole strategy changes completely depending on whether it is aimed at keeping the engine state at one single level or several ones, whether operating-point throttling must be handled, whether transient control should be covered, whether system robustness is a major concern, etc. And generally this global goal is translated first into the choice of an appropriate model capturing the targeted dynamics. LPREs are naturally multivariable systems whose states mainly consist in pressures, temperatures, rotational speeds and mass flows. The state components must satisfy shafts mechanical equations and thermo-fluid-dynamic conservation equations on mass, energy and momentum, which induce high coupling between the variables. These differential equations, originally three-dimensional, can be firstly treated as zero-dimensional, as all reviewed authors do, via standard lumped parameters schemes or by directly

neglecting spatial contributions within components. The control of partial differential equations, an emerging field, has not been considered for LPREs, due to their high complexity and to the little interest of taking spatial effects into account. The behaviour of most components can be well described via zero-dimensional equations. Only in chambers would those contributions be relevant, but proved rather difficult to account for in the control design. The formal definition of these equations for the inter-connected components in a LPRE naturally leads to a system of nonlinear ODEs, unless linear identification techniques [44, 117] are used from the beginning. Another easier modelling alternative is defining equations already in the frequency domain in a linear way [132–134]. Describing functions have been used in one reference [111]; in fact most authors have linearised their nonlinear models about steady-state points [13, 71, 88, 89, 102, 109, 140, 141, 150, 175, 178], which limits representativeness to a narrow region around those points, as explicitly stated by [71] for instance. The only control-oriented nonlinear modelling approaches present in the literature dealt with engine performance simulations [57, 66, 131], state-space analysis [174], OL optimisation [36, 37, 166] and neural networks [75, 168]. These nonlinear models are representative in a wider region than linearised ones but imply a higher complexity. Nevertheless, some authors [13, 88] affirm that their linearised models are valid in relatively large domains, e.g. variations of $\pm 35 bar$ about the nominal point in [13].

Measured variables in LPREs mainly consist in pressures, TP rotational speeds, valves positions and some temperatures (not too elevated). Mass flows are generally only measured in test benches and not in actual flying engines. Hence, mixture-ratio estimators are common practice [55, 75, 115]. The more sensors are installed, the more precise control would be. Nevertheless, the inclusion of further components in an actual engine reduces its reliability from the engineering perspective, which is vital in space vehicles. In terms of model, most authors consider first-order transfer functions for sensors. Actuators are primarily valves and igniters. The former are deemed as either first-order or second-order systems with saturations and igniters are discrete elements. Anti-windup schemes have been applied to valves in CNES in order to mitigate their saturation. In fact, in expander-cycle LPREs, it is often the case that the controller forces the valves to be fully open to improve response time accounting for the intrinsic expander-cycle-related limitations. In other words, the slower operation dynamics of this type of engines, dictated by the heat exchange produced in the regenerative cooling circuit, limit the response time of the system.

Concerning the analysis of system characteristics, LPREs are generally naturally-stable systems, which to a certain extent simplifies control objectives by removing the need for stabilisation. Even so, some internal natural resonances, mostly related to TP rotation, may present frequencies that could destabilise the system [155]. Moreover, some engines present non-minimum phase behaviour, which entails initial reverse reactions [13]. The dangerous combustion instabilities can generally be mitigated via an appropriate passive-control design [55, 155]. Besides, controllability and observability conditions are rarely satisfied at very low operating points, especially during the sequential phase of start-up [115]. Adjusting mass flows through valves is hardly achievable, and mass-flowmeters (if installed) valid from very small flows to nominal values may become burdensome.

Regarding controller design strategies, the fact that on the whole LPRE stability is mostly naturally satisfied allows to focus on tracking and perturbation-rejection goals. Reference [55] points out that thrust and mixture-ratio control can be achieved in OL if no high accuracy is required, or if off-line

optimisation strategies are pertinent [37, 63]. The majority of references reviewed have closed the loop, which improves robustness and performance at the cost of installing sensors. Most authors have selected conventional PID-based techniques for controlling around a given nominal point. Concretely, PI controllers are the most used option worldwide, present in US-American, European and Japanese engines [60, 65, 77, 115, 117, 154], while CNES have also dealt with PID. In order to use these techniques in this multivariable application, the initial MIMOs are decoupled into SISO subsystems relating controlled variables to actuators, which are commonly two or three dominant valves. This decoupling is sometimes performed via feed-forward. The algorithm in [60], also including feed-forward for adapting to a larger operating domain, is said to achieve control from 21 to 109% of nominal thrust. Other conventional CL linear approaches considered multivariable state feedback [174, 178]. The only reference employing CL nonlinear techniques [89], factored damage modelling and control into their loop as other NASA reusable-engine works have done [37, 87, 109], which seems convenient for extending engine life. Nonlinear optimisation regarding damage was performed through SQP, while the inner engine loop was still linear. Other works dealt with the robustness problematic, primarily posed by varying parameters, by means of the generalised KHARITONOV theorem [132] or through H_∞ techniques [71, 141]. Only two sister papers have concerned a subtype of hybrid techniques (variable-structure sliding modes) [175, 176], but solely applied to combustion phenomena and not to the global LPRE system. One of the most relevant works, [109], combines some hybrid aspects, like switching, with reconfigurable control strategies, in order to accommodate faults of major valves in SSME. The LQG/LTR method is employed in their main controller. Indeed, the interaction of HMS and control, currently still pretty separate, is considered as a major development area by [97], since it would enhance engine robustness and reliability and hence reusability. The recent work of [140] proposes concrete solutions in this direction, applied to a LPRE subsystem. Besides, transient sequences like start-up or shutdown present discrete-events phases which are still performed in OL nowadays. The order of events is generally fixed by engine operation design, but the temporal separation between events is optimised in one reference [73] via LS, according to structural or system characteristics, the environment or the thermodynamic state. The total duration of transients is increased but robustness to different initial conditions is attained.
An efficient way of testing new control algorithms is to build a hardware-in-the-loop test bench, as [127], able to simulate adverse conditions and component faults. When more knowledge on algorithm performance is gained, tests can proceed in actual engine test benches, which then serve to validate the control strategy for future flights.

In order to relate all these different considerations in control design, it seems relevant to summarise and discuss now the complete combinations of design approaches in order to extract the predominant global strategies. Table 2.5.1 synthesises this information. In the goals column, whenever *control* is specified, it refers to the typical thrust and mixture-ratio control. As it can be seen in the table, few combined approaches of modelling and control are present in the literature. This is because many articles do not provide precise information on the model used for deriving the control law, and at the same time the goals of many other articles only concerned modelling.

As pointed out before, PID approaches are related to linear (or linearised) submodels, one of them coming from identification [117]. Further repeated strategies are the combination of linearised MIMO systems and robust techniques like H_∞ [71, 141], which is an effective way of treating model uncertainties from several origins at a concrete thrust level. Another analogy would be between [178]

Table 2.5.1: Chronological summary of authors' goals and approaches in modelling and control

		Approaches	
References	**Goal**	**Modelling**	**Control**
OTTO AND FLAGE 1959 [117]	**Transient and steady-state control** of regeneratively-cooled hydrogen-fluorine RE	**Linear identification** and static equations, decoupled SISOs	**PI**, transient switches
MCDERMOTT *et al* 1966 [102]	**Transient performance** of pressure-fed bipropellant *Surveyor*	**Linearised** MIMO	-
SEITZ AND SEARLE 1973 [142]	SSME **transient and steady-state control**	-	**CL** after events sequence
ZHOU 1982 [178]	**Steady-state control**	**Linearised** MIMO	**CL:** state feedback
ZHANG 1984 [174]	Pressure-fed cycle **state-space analysis**	**Nonlinear** MIMO	**CL:** state feedback
DUYAR *et al* 1990 [44]	SSME **identification**	**Linear identification**: RML	-
NEMETH *et al* 1991 [115], [114]	SSME **transient analysis and steady-state control**	-	**PI** after events sequence
KOLCIO *et al* 1994 [66]	**Transient modelling** for monitoring and control performance	**Nonlinear** MIMO	-
ZHENG *et al* 1995 [175]	Monopropellant **combustion stabilisation**	**Linearised**, unstable, time-delay combustion model (lower level of LPRE system) [35]	**Hybrid**: variable-structure sliding modes, robust stability conditions in [176]
MUSGRAVE *et al* 1996 [109]	SSME **reconfigurable** ICS with FDI	**Linearised** MIMO LTV	**LQG/LTR and PI** based on [108] with **gain scheduling** and mode switching
DAI AND RAY 1996 [37], [36], [166]	SSME damage-mitigating **throttling control**	**Nonlinear** MIMO	**OL** optimisation
BERGMANS AND MYERS 1997 [13]	**Modelling and analysis** of air turbo-rocket	**Linearised** SISOs	-
LE FUR *et al* 1997 [71]	Expander **robust steady-state** control	**Linearised** MIMO	**Robust:** H_∞
IFFLY AND BRIXHE 1999 [57]	*Vulcain* **transient performance** model	**Nonlinear** MIMO	-
SAINT-MARD *et al* 1999 [131]	**Transient performance** of LPREs, actuators analysis	**Nonlinear** MIMO	-
SANTANA *et al* 2000 [132]	**Steady-state modelling and uncertainty** analysis and management	**Linear** models based on [133] and [134]	**Robust:** Kharitonov's theorem

		Approaches	
References	**Goal**	**Modelling**	**Control**
KIFORENKO AND KHARITONOV 2000 [63]	**Thrust control** from launcher dynamic perspective (MR constraint)	**Linear** launcher dynamic equations, with algebraic relations to engine	**OL** optimisation
SAUDEMONT AND LE GONIDEC 2000 [141]	*Vulcain* **robust steady-state control**	**Linearised** MIMO, LPV, LFT	**Robust:** H_∞
LORENZO *et al* 2001 [89]	SSME **LEC** with damage considerations	**Linearised** MIMO	H_∞ and **nonlinear** SQP optimisation
YANG *et al* 2001 [168]	**Modelling** for real-time monitoring, diagnosis and control	**Nonlinear** feed-forward neural network	-
RTO/NATO 2002 [129]	Active **combustion control**	-	**OL**: powerful actuator
NASSIRHARAND AND KARIMI 2005 [111]	**MR steady-state control**	**Describing functions**	Factorisation theory
SUNAKAWA *et al* 2008 [154]	*LE-X* **steady-state control**	**Linear** decoupled SISOs	**PI**
LE GONIDEC 2011 [72]	**Steady-state control**	Predictive internal model	**CL:** state feedback
SOLTANI *et al* 2012 [150]	**FDI** of *Hopper* engine	**Linearised**: structural analysis	-
LE GONIDEC 2013 [73]	**Transient sequence adaptation**	Hybrid transient model	**Reconfigurable** sequences: LS, fuzzy and predictive techniques
KAI *et al* 2015 [60]	**Steady-state tracking**	**Non-model-based**, decoupled SISOs	Feed-forward and feedback (**PI** and thresholds).
LE GONIDEC AND FAYE 2015 [77]	**Steady-state control**	-	**PI**, bang-bang
KLEIN *et al* 2017 [65]	Expander **steady-state control**	-	**PI**, tracking filters, **OL** in transient
LE GONIDEC 2017 [75]	**Steady-state control** with **MR estimation**	**Linearised** MIMO, **nonlinear** neural network for estimation	**CL:** state feedback
SAROTTE *et al* 2018 [140]	**FDI** and **reconfiguration** of test-bench subsystem	**Linearised** MIMO	**Reconfigurable**, **LQR**

and [75], which perform state feedback on a linearised MIMO system. The rest of combinations are unique in the literature, which hinders the identification of trends. Nonetheless, it can be observed that more complex control techniques, such as LQG/LTR, or hybrid or robust strategies, are only associated with linear or linearised models. As commented earlier, nonlinear modelling contributions mainly targeted performance simulations, state-space analysis or OL optimisation [37], the latter being one of the most relevant studies reviewed (together with the hybrid reconfigurable approach [109]). These observed trends are logical in the sense that linear models allow more flexibility in control design at the price of a reduced validity domain. And this is precisely in direct relation to the goal of the study. Most studies have aimed at controlling engines within some pre-defined operating ranges, which matches very well a linearised model. The few articles facing transient control up from off-design pressure values [115, 117, 142] (until 1993) did not manage to obtain the same level of performance and robustness at all regions with their controllers, which were usually more adapted to the nominal region. Nonlinear models, and perhaps nonlinear control would be more appropriate for these phases. However, during the pre-defined (normally OL) sequence of discrete events in start-up and shutdown, there are generally controllability and observability issues due to the low mass flow rates, mainly remarked by [115]. That is the reason why continuous control starts after the end of that sequential phase. Further enhancement of the control of these transient phases will be a crucial area of improvement in LPREs control in the upcoming years, as summarised in the next concluding section.

2.6 Concluding remarks and answers provided in this thesis

This chapter has reviewed the different automatic control methods applied to liquid-propellant rocket engines, whose respective communities have traditionally remained relatively separate. Even though the total number of academic works and accessible industrial developments is relatively low, sufficient information is present in the literature to analyse the state of the art of this complex topic and its areas of improvement. The main control problem in these multivariable systems primarily consists in tracking set-points in combustion-chamber pressure and mixture ratio, whose references stem from launcher needs. Control-valves opening angles are adjusted in order to adapt engine's operating point while respecting some constraints. The different aspects in control systems, concerning modelling, sensors and actuators considerations, system analysis and the actual control techniques, have been reviewed and related to one another. From the comparison of the different approaches reviewed, the most common trend identified relies on linearised models about operating points for synthesising steady-state controllers, most of them based on PID techniques. In those cases, initial MIMO systems are considered decoupled into dominant SISO subsystems. Other more complex approaches present in the literature, incorporating some nonlinear, hybrid or robust techniques, enhance certain aspects of performance and robustness. However, no global approaches have been published that consider not only steady-state but also the demanding transient phases at the same level of performance and robustness. There is also a lack of method comparisons on a common benchmark, even simulated. Besides, only narrow throttling domains are feasible.

The potential need for reusable engines presents stronger robustness requirements than expendable ones due to 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 forced by new launchers.

The classical multivariable control of main-stage LPREs had attained a reduced throttling envelope (70%-120%) at test-benches [76]. In the future European *PROMETHEUS* engine, it is aimed at throttling down to 30% [5], and current designs by SpaceX [152], Blue Origin [17] and JAXA [60] are claimed to attain 39%, 18% and 40% respectively. Thus, an enlarged validity domain for reusability has to be conceived. At least, it becomes crucial to maintain tracking and robustness at those low throttle levels, where physical phenomena are more difficult to anticipate. The damaging combustion instabilities, which might be specifically controlled in simple engines (e.g. monopropellant pressure-fed [167, 175]), are more prone to appear in those cases. The management of this problem, apart from involving other design considerations, directly affects the control system. This system will have to ensure that the desired thrust level is robustly achieved. Indeed, one of the main conclusions of this literature review is the absence of fully nonlinear or hybrid frameworks, which may permit the control of a wider throttling domain. In this sense, possible solutions could be enhanced nonlinear approaches accounting for transient behaviour, gain-scheduled switched controllers, large off-line optimised-behaviour scheduling, optimal CL control, etc.

Pre-defined sequences (start-up/shutdown), traditionally managed in open loop with low correction margins, could be adapted according to the evolving system, depending on thermal issues, damages or on the mission. In this sense, the control design path proposed by [76] is hybrid control, blending conditional sequences with continuous control of thrust and mixture ratio.

However, as the conclusions in [115] on the control of transients indicate, controllability and observability issues may hinder that task. Transient control started to be plausible once all events had finished. Thus, those authors proposed to modify SSME configuration so as to enable these characteristics during the discrete-event phase. For instance, lines could already be primed prior to start-up, thereby creating certain mass flows relevant for valve control. If more actuators come into play, such as flow-control surfaces in chambers, the emerging field of partial-differential-equation control [128] may become attractive. This way, spatial effects might become controllable. Such multidisciplinary design choices, considering the control system as a key element in design loops, will definitely help improve LPREs operation in reusability scenarios.

Damage-rate management is one of the main contributions of NASA's SSME-related research [37, 89, 109]; and indeed it could be beneficial for new reusable engines. Recent studies at DLR (Deutsches Zentrum für Luft- und Raumfahrt) apply ANN to the prediction of heat transfer in regenerative cooling channels [165] and of fatigue life in combustion chambers [42]. Estimating or modelling the damage accumulation of the most exposed components in LPREs, like turbopumps, will certainly help to redefine control constraints and thus improve robustness, as shown in the aforementioned NASA papers. And in the event that such a component fails, reconfiguration strategies, perhaps making use of switching logics and hence some hybrid elements, will be convenient for ensuring the fail-safe operation of the launcher. A full interaction between control and HMS subsystems would become crucial in that case.

All these enhanced control aspects will not be computationally feasible in many cases without more powerful computers on-board. The authors in [141] highlight that the advantages of such a computer would enable a precise throttling and parameter corrections due to perturbations, which would translate into propellant savings. The problem is that it is generally not practical to install it in the launcher,

since it complexifies validation tests due to the inclusion of further reliability factors.
Nevertheless, it seems convenient to start deeming these improved control strategies as a real gain in reliability, contributing to robustify reusable rocket engines, as sought in this thesis.

Indeed, in this thesis some answers to these open questions are proposed. It has been sought to find a more advanced multi-objective solution for the main control problem in LPREs, which is the tracking of set-points in combustion-chamber pressure and mixture ratios. Concretely, the adopted solution tackles the robust control of GG-cycle LPREs along their whole thrust envelope, from their start-up transient until the selected thrust level. Indeed, robustness to internal parametric variations and transient constraint verification are additional control goals required by reusability.
The GG cycle has been selected as a representative LPRE benchmark, seeing that is present in the European engine *Vulcain 1* and more importantly, in the future reusable engine *PROMETHEUS*, from which control requirements are derived. Figure 2.6.1, representing the simplified flow plan of *Vulcain 1*, serves to define the considered benchmark. Only the core hydraulic system with the main actuators (valves and igniters) lies within the scope due to its dominant behaviour with respect to other subsystems, like purge lines. This cycle is explained in greater detail in Chapter 4.

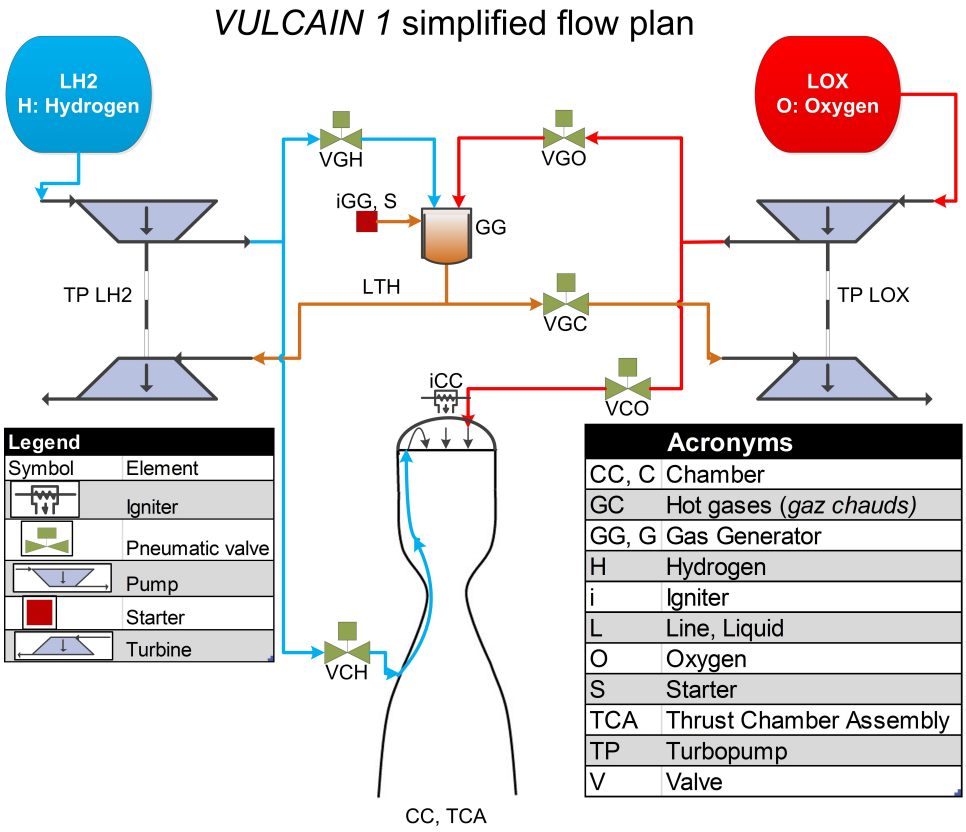


Figure 2.6.1: *Vulcain 1* simplified flow plan

Regarding the modelling of this benchmark, as explained in Chapter 5, nonlinear state-space models have been developed by considering the individual fluid-mechanic behaviour of engine components as well as the global transient behaviour. States consist in rotational speeds, pressures and mass flows. Discrete elements, concerning the igniter actuators, are considered as part of the control inputs.

The suitability of these transient models for the synthesis of control laws has been then studied. The models, in their nonlinear form, have been used to generate reference trajectories off-line as a function of the desired thrust and mixture ratios.
With regard to the on-line control, a model and optimisation-based method, Model Predictive Control (introduced in Section 3.3.1), has been deemed the most appropriate for this application, with hard operational constraints and multiple variables. The structure of the proposed loop is depicted in Figure 2.6.2, in which a full state feedback is assumed. A more detailed explanation of the loop can be found in Chapters 6 and 7.

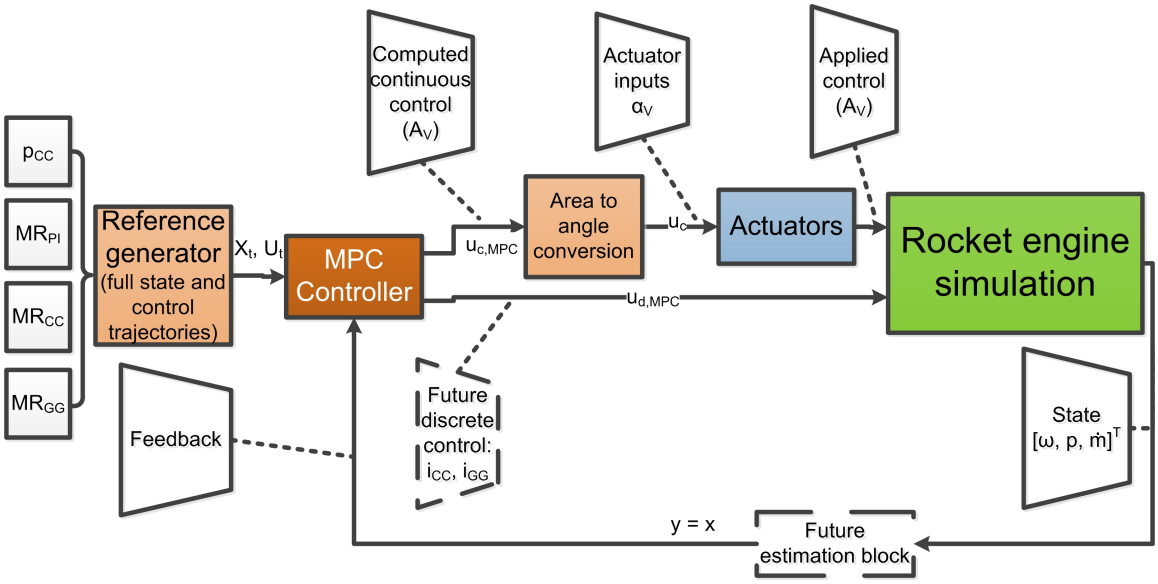


Figure 2.6.2: Proposed control-loop diagram

The on-line consideration of nonlinear dynamics has been explored in the development of that controller. However, linearisation about trajectories has proved to be more adequate for the on-line resolution of the optimal control problem. The control of the continuous sub-phase of the start-up transient (Chapter 7), as well as of throttling scenarios (Chapter 6), has been achieved with robustness considerations and constraints satisfaction. Proposals to the management of discrete inputs are also made in Chapter 7. This framework paves the way to the experimental validation in terms of hardware-in-the-loop simulations or further tests.

3 Methods and tools

In this literature-review chapter, diverse fields connected to the thesis subject are discussed. The previous chapter covered the convergence field between automatic control and LPREs; in other words, the state of the art of LPREs control. In contrast, in this section, the focus is put on methodological aspects and computational tools evaluated or used in the thesis. First, Section 3.1 provides an in-depth explanation of the transient modelling of LPREs, including the description of their behaviour and the main modelling and simulation approaches. Then, Section 3.2 serves to introduce the field of hybrid systems modelling for control purposes, linked to the dynamic model in this thesis. In Section 3.3, the different families of control techniques used in the thesis (predictive and robust) and envisaged for future work (hybrid) are discussed. Additional used tools, the SOBOL sensitivity analysis and Kriging-based techniques, are briefly presented in Section 3.4. Finally, summaries on each field are provided in 3.5.

3.1 Transient modelling of LPREs

Since this thesis is devoted to the control of LPREs during transient phases, it is paramount to well understand the natural behaviour of these systems under those conditions and the way they are modelled and simulated. Thus, those topics are reviewed in the following paragraphs.

3.1.1 Transient behaviour

The key phases, for which the control algorithms developed in this thesis are mainly conceived, are transients. The two main ones, taking place every time the engine is run, are start-up and shutdown. Further examples are related to throttling, calibration needs or operational deviations [55]. If it is aimed at maintaining performance level during these transients, components and operation must be designed in such a way that they can perform in a wide operating range, since extreme operating conditions are imposed on the components during these phases.
As explained in MANFLETTI's thesis [96], LPREs are tested under nominal conditions at different phases of their development so as to guarantee a maximum level of reliability. However, the most adverse conditions may occur during transients. Some transient phenomena related to these phases are combustion high-frequency instabilities, water-hammer effects, off-design turbopump operation, two-phase flows, etc., which may lead to temperature and pressure peaks, maybe detrimental to the engine. Some in-flight anomalies have indicated so, creating the need for transient behaviour modelling and testing.

Start-up transient

Consequently, transient processes are very accurately planned, fitting engine's characteristics and ignition type. Depending on the LPRE cycle, pressure-fed, gas generator, expander or staged combustion, different sequences are planned. GG-cycle engines such as *PROMETHEUS*, but especially cryogenic engines (*Vulcain 1, 2*), require a chill-down phase of 2 to 3 hours prior to the start-up so as to adapt the thermal state of components (except main valves and GG itself) in order to minimise the thermal shock effects arising from the cold fuel pumping [96]. Then, a typical start-up sequence for bi-turbopump engines such as *Vulcain 2* begins by activating the starter related to turbines, which feeds the GG sufficiently so as to allow its pyrotechnic ignition. Next, with turbines turning faster, the main chamber can be fed at a higher pressure than in the tanks. The first chamber valve to be opened is the fuel one, due to two reasons: fuel must first flow through the regenerative circuit and a fuel-rich environment in the chamber is desired in that case. Finally, another pyrotechnic igniter is employed to trigger combustion.
Another example of start-up sequence for a generic GG cycle, is provided by [55] (Figure 3.1.1):

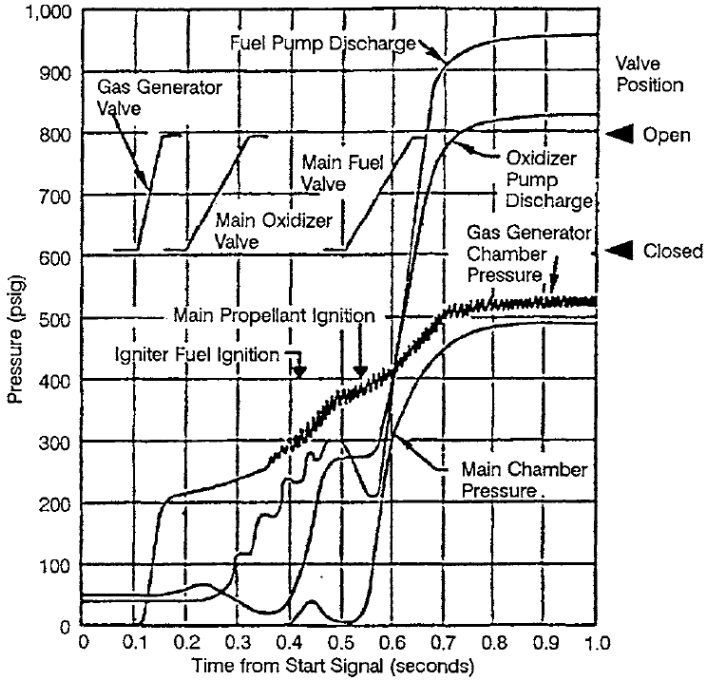


Figure 3.1.1: GG-cycle engine start-up sequence [55]

As indicated in the previous figure, several characteristics have to be taken into account when dealing with start transients [55]. A basic one is the pressure-build-up performance of GG, pumps and chambers, which translates into thrust build-up. Energy and timing characteristics are also vital and have to be evaluated and optimised. The influence of external initial conditions or perturbations, such as a start-up in vacuum, must also be assessed, with the aim of avoiding complex and problematic phenomena such as unwanted pressure drop, stall, surge, cavitation, changes in fluid density, etc. The type and quantity of control components is determined by all these features.

However, the goal of attaining a smooth and reliable ignition, lacking of pressure and temperature peaks, is the main driver in the careful planning of the start-up sequence, adapted to each engine. For instance, ignition overpressure is due to excessive ignition delay times [96]. If ignition is delayed, too much unburned propellant cumulates in the chamber leading to deflagration combustion once ignition finally starts. These pressure peaks may cause back-flows into the feed lines, probably becoming catastrophic. Temperature peaks are not that critical at this phase since the engine normally comes from a chill-down phase.

Other relevant factors indicated by KANMURI *et al* [61] are cooling characteristics of engine's components and turbopumps acceleration behaviour. They modelled transient phenomena with the objective of establishing an adequate start-up sequence for the Japanese staged-combustion engine *LE-7*. The hardest issues in their simulations turned out to be the estimation of the ignition timing and engine parameters, which in the end were simply taken from test data. Overshoot arose mainly when ignition (judged from mixture ratio estimation) was activated too early.

The most sensitive components to set-point variations are turbomechanical ones. The authors of [55] recommend the use of an active control system to adjust pump parameters to operate at the desired point. Otherwise, pumps run the risk of falling into stall during low-flow phases. For engines where throttling is required, centrifugal pumps are usually employed since they present less restrictive stall characteristics. Indeed, rocket-engine turbopumps must reach full power faster (less than $1s$) than in any other application domain [55], stemming from propellant-optimisation requirements. This complicates transient control.

It is also interesting to look at other types of cycles. In expander-cycle engines, such as *Vinci*, the energy to power turbines comes completely from the thrust-chamber regenerative circuit. Therefore, the first operation to carry out is the ignition of the main chamber, performed electrically in this engine [96]. Then, as temperature in the chamber increases, so does the fuel used to cool it, attaining a supercritical state and hence gaining velocity and pressure. Thanks to this, the reaction turbines can start to turn.

Moving on to staged-combustion engine examples, interesting information on SSME's transient behaviour is already open to the research community [126]. This engine also presents a preliminary phase comprising purge, thermal conditioning (chill-down around $1h$) and a second purge. Then, a similar start-up to expander cycles is conceived. Figure 3.1.2 depicts it, regulated by five valves.
Up from the start command ($t = 0s$), MFV is opened and the three major oxygen valves, MOV, FPOV and OPOV are regulated in such a way that the objective filling times in the two preburners and in the main chamber be attained [96]. FPOV and OPOV are also controlled in order to maintain mixture ratios at reasonable levels aiming at avoiding high temperature and pressure peaks. Indeed, pressure oscillations in the fuel system are monitored and serve to determine fuel preburner's ignition instant. It corresponds to the second decreasing oscillation. Next, the oxidiser preburner is ignited. $1.25s$ after start, the angular speed of the HP fuel TP is verified to guarantee that hydrogen can be pumped against the back-pressure generated in the main chamber due to the oxidiser filling. As turbines speed-up, CCV (chamber coolant valve) opening is reduced to 70% until $2.4s$, when the control system switches to CL and monitors chamber's pressure and commands OPOV, FPOV and CCV so as to pursue a nominal chamber pressure ramp until the desired operational point. The last

step is regulating FPOV so as to achieve the nominal mixture ratio. The whole start-up lasts around $5s$ in nominal conditions.

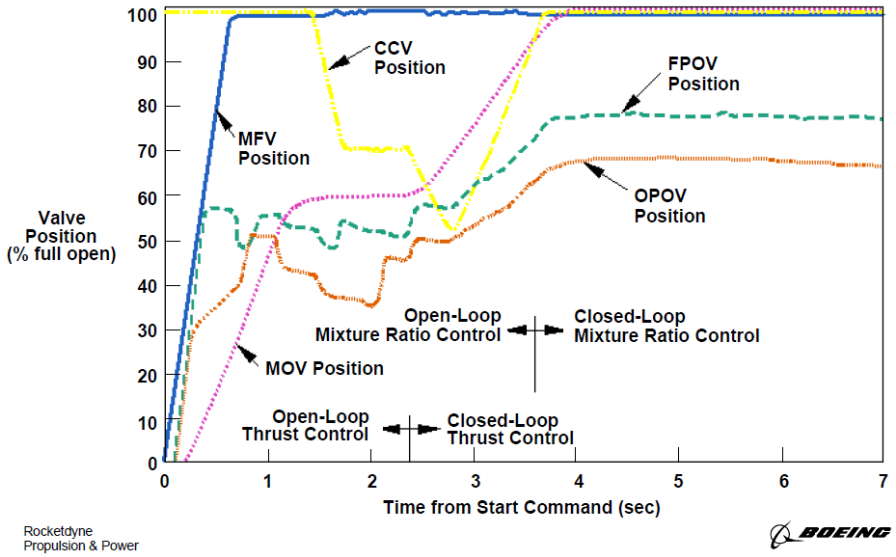


Figure 3.1.2: SSME start-up sequence [126]

Shutdown transient

For shutdown, similar considerations have to be taken into account, but in this case related to decay rates. An interesting ratio is defined in [96], the temperature ratio $T_R(t) = \sqrt{T_c(t)/T_{id}}$, which relates the time-dependent combustion temperature and the ideal steady-state one. If this ratio is below the unity, lower temperatures and therefore lower efficiencies are encountered. However, during transients, it is of interest if this ratio remains below one. Indeed, during ignition or shutdown, fluctuations in mixture ratio towards the stoichiometric one can trigger excessive temperatures. It is therefore paramount to carefully tune valve sequences in order to avoid this, especially for shutdown. Normally, oxidiser-side valves are closed first so as to diminish the mixture ratio and avoid the danger zone. Figure 3.1.3 depicts a shutdown example for the GG engine.
During shutdown, turbopump decelerates owing to friction losses and drag torque generated by pumps as they remove the remaining propellant from the ducts. According to [41], the most adverse phenomena occurring in this transient phase are pump cavitation and reverse flow. The former may happen after the fuel inlet valve closure due to the varying load. Then, this can lead to reverse flow into the cooling jacket (regenerative circuit), which may provoke extreme transients in pressure and mass flows.

During SSME's shutdown phase, all of its five valves are subject to shutdown schedules, depicted in Figure 3.1.4. These schedules are designed so as to guarantee a smooth and safe shutdown by establishing a fuel lag; in other words, the oxidiser leaves the combustion chamber before the fuel [126]. In this fashion a fuel-rich environment is ensured, which is related to cooler temperatures due to the lower mixture ratio. Let us recall that maximum temperatures are attained around the stoichiometric mixture ratio, and these are always lower the further from this point. Reference [41] also highlights that the fuel chamber valve could even increase its aperture so as to reduce turbines

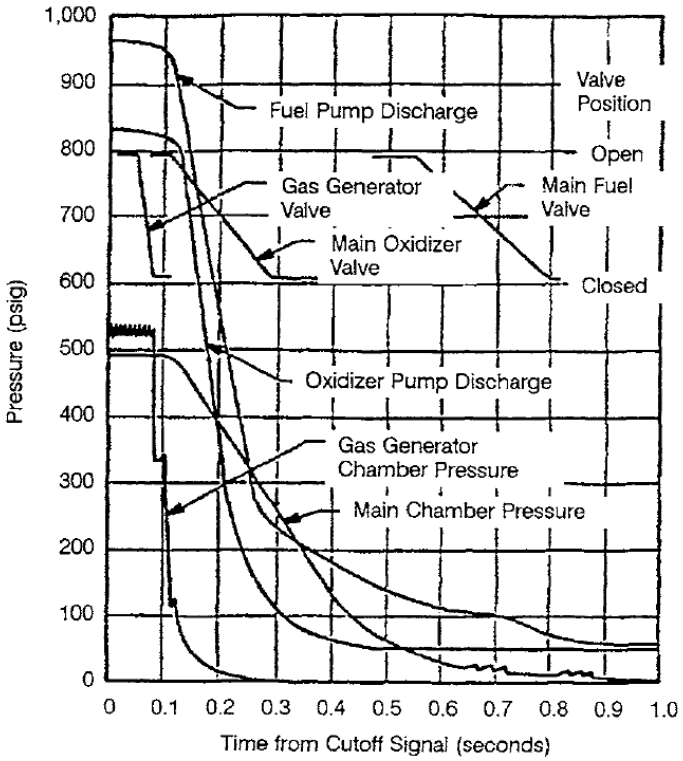


Figure 3.1.3: GG-cycle engine shutdown sequence [55]

power more quickly. This way the global propellant flow diminishes and surge effects in the oxygen pump can be avoided.
OPOV is the first valve to be progressively shut ($< 45\%/s$), in order not to provoke a too steep thrust decay, dangerous for spacecraft's structure. Then, FPOV and MOV are commanded to shut. However, their positions are regulated with the aim of preserving low mixture ratio and pressure decays while impeding undesired back-flows. CCV is also regulated so as to reduce heat loads in the chamber. When the main valves are already closed, MFV and CCV are kept open during an additional second so as to ensure a very fuel-rich shutdown.

SHAFIEY DEHAJ *et al* [144] highlight the fact that the thrust or pressure at the beginning of the process determines dramatically shutdown phases' durations and the residual impulse generated. The latter, the so-called cut-off impulse, is usually set as a minimisation goal due to launcher mission constraints. The overall sequence conception must also be selected in such a way that complete engine blow-up is avoided, due to excessive compression in propellant injection components.

From the previous paragraphs, it can be deduced that both transient operations, start-up and shutdown, consist of an initial sub-phase determined by some discrete events, and a subsequent one where a purely continuous behaviour is observed. The initial discrete events consist in the aforementioned ignitions, starter activation and valve openings or closings. Basically, these events modify the otherwise purely continuous behaviour of the system. The distinction between these two sub-phases is essential in this thesis, where different approaches to the control of each sub-phase are proposed.

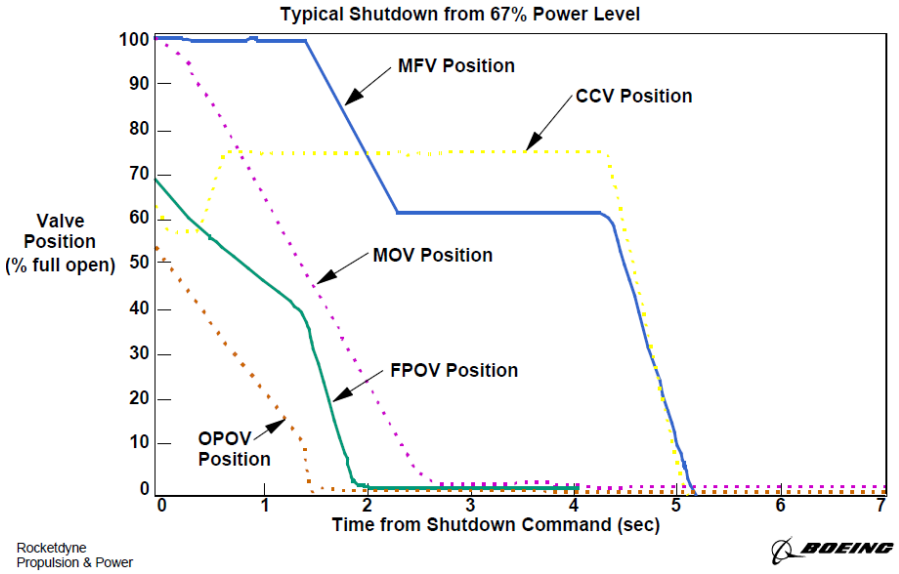


Figure 3.1.4: SSME shutdown sequence [126]

3.1.2 Transient modelling

The manner in which the aforementioned transient phases are modelled is a current research topic. Modelling of LPREs for control has been reviewed in Section 2.3.1, but here a wider overview of transient modelling paradigms and considerations for simulation purposes is provided.

A modelling and simulation tool of interest for this thesis is *CARINS*, a piece of software developed by ONERA, CNES and other laboratories since 2001. As presented by ORDONNEAU *et al* [116], the main aim of this tool, based on different open-source pieces of software, was the modelling and simulation of complex physical systems, especially of LPREs. For this sake, different libraries allowing the modelling of hydraulic, pneumatic, mechanic, thermal and control components are included and updated regularly. The ODEs related to components are based on mass, momentum and energy balance expressions. Some examples of complex LPREs phenomena modelled as PDE (Partial Differential Equations) are compressible flow, pressure waves, biphasic flows, heat transfer, injection (pulverisation, vaporisation).

The aim of MANFLETTI's thesis [96] was to develop a powerful simulation framework in which engine transients could be well predicted and at the same time computational times could remain short (in contrast to Computational Fluid Dynamics (CFD)). The concept of interaction between components in a downstream and upstream sense is the key of this modular ODE modelling approach. The numerical method used is the lumped parameter method (LPM) with EULER propagation, consisting in a 0-D modelling strategy in which a system is discretised into segments or lumps. Along these sections, system parameters are deemed to be uniform over volume. In this fashion, PDE are transformed into ODE, reducing complexity. This approach led to the development of *Moliere*, used for her simulations. ODEs, assumptions and boundary conditions related to modelling of the following LPRE components are provided: pipelines, dampers/gas accumulators, control devices (valves and throttles), injection domes/manifolds, turbopump assemblies and combustion devices.

A complete system simulation of the expander-cycle *Pratt&Whitney RL-10* is carried out by DI MATTEO *et al* [40] based on components from the ESPSS library in *EcosimPro*® (also presented within his dissertation [39]). Nonlinear interactions between components during transients are claimed to be well modelled in this multidisciplinary simulation tool. A 1-D discretisation is implemented in all the main components, including the thrust chamber (into 40 volumes). Valves remain 0-D.

In their analysis of NASA's ORION Multi-Purpose Crew Vehicle, VON GRAEVE *et al* [163] achieve good representativeness of water-hammer and priming effects thanks to an accurate *EcosimPro*® model. Pipes were deemed adiabatic to the environment, real fluid properties for propellants were considered and the homogeneous equilibrium model for two-phase flow was used. The discretisation of pipes was performed according to the rule of inertia, where the ratio $\frac{L/n}{A}$ must remain constant. L is the pipe length, n is the number of nodes and A is the cross section. The number of nodes was obtained from a trade-off analysis, confirming the representativeness of the highest pressure peaks.

BORONINE and FREY from ArianeGroup (AG) Ottobrunn [20] have developed in the last years an own versatile modular simulation tool for rocket engines, *SMART*, optimised for capturing transients. Cryogenic and hypergolic engines can be modelled, including all the typical rocket engine components as well as PID controllers. The main physical model for the one-dimensional flow discretised in a staggered grid presents the standard mass, momentum and energy conservation laws. Well-known approximations are applied to each component. RUNGE-KUTTA methods of second and third order are employed in the solver.

LIU and ZHANG [81] describe a modularisation technique based on the LPM, analogous to [96] and to the described software tools *EcosimPro*® and *SMART*. They call it pipe-volume modularisation disassembly, and conceive 21 different types of rocket-engine components. This method is claimed to be adaptive to complex elements such as valves, or turbopumps. Basically, ordinary one-dimensional compressible-fluid perfect-gas equations are considered for the three main classes modelled: pipes, volumes and valves. When priming occurs in a component, special equations are solved in the finite elements which are filled with liquid and gas at the same time at a certain instant. A set of rules is defined for components interconnection, which leads to a nonlinear system of equations for the whole assembly. A RUNGE-KUTTA propagation solver is employed.

A different basic numerical method for modelling is employed by RUTH *et al* [130]: the method of characteristics, said not to be limited by GIBBS phenomenon and to be able to capture discontinuities in velocity and pressure. This method was evaluated by [96], but she discarded it for being iterative, needing a solver and normally presenting slow convergence. In contrast, it does not require a time propagation method such as EULER or RUNGE-KUTTA.

A transient model for the *Vinci* expander engine is presented by DURTESTE [43] from *Snecma Moteurs* (currently AG). It is able to predict transient behaviour correctly when compared with test-campaigns data. A commercial modelling and simulation tool based on the Bond-graph theory and LPM is employed. Sub-models for each component are self-programmed thanks to the expertise in the company. In general, the standard conservation laws in the form of ODE are implemented. Since that article is devoted to an expander engine, the regenerative circuit dynamics play the leading role. Thus, variations of fluid properties are paramount, which leads to the choice of real-gas consideration as a function of the state variables.

Components modelling

As hinted in the previous paragraphs, the modelling of LPRE systems generally starts at the component level, where thermo-fluid-dynamic processes take place. However, for the sake of consistency, some general hypotheses shared among components have to be established when developing a model. For instance, the main modelling hypotheses assumed in the *Vulcain 1* performance model by IFFLY AND BRIXHE [57], very similar to the model developed in this thesis (in Chapter 4), are the following:

- Components are deemed zero-dimensional, uniform, steady and flow-averaged.
- Viscous effects are expressed as a pressure drop in lines, valves, injectors, etc.
- Chemical equilibrium is supposed.
- Thermodynamic tables are used for state equations of fluids.
- Perfect gases are assumed.

Focusing on each individual component, several internal assumptions can be made. But in general terms, they can be classified according to their physical nature into resistive and capacitive ones. Resistive components, such as valves or pipes, are modelled via the momentum-conservation equation (4.2.6). Capacitive elements, such as cavities or combustion chambers, require the energy-conservation equation (4.2.8) in order to determine their pressure and temperature evolution. The continuity equation (4.2.4) has to be verified everywhere in the cycle. Different degrees of complexity are considered depending on the application of the simulator.

Valves

Valves are modelled in [96] as a resistance in the flow field influencing pressure and mass flow. It is indicated that control valves present different opening curves depending on their type. There are linear, parabolic (ball valve), semi-linear (butterfly) and exponential curves. The correct behaviour has to be selected according to the application and desired effect along the transient. Hot-gas valves are modelled in [57] by means of the *Universal Gas Sizing Equation* [23] adapted for compressible fluids. For the rest of liquid-flow valves, their pressure drop is modelled as a simple function of valve displacement. Further details on valve modelling considerations are given in Section 4.2.2.

Injection

Injection of pumped propellants into combustion chambers is modelled in [96] by considering two phases via the LOCKHART-MARTINELLI correlations. Mass flow increases to its maximum value staying constant until the injection dome volume is completely filled, moment when it decreases a little to stabilise around the steady-state value. System pressures (including tanks, pipes, throttle and dome) undergo a similar behaviour, but with more oscillations in the feed lines. This behaviour must be checked, since the oscillations in the feed lines must not arrive to the dome or chamber, so as to avoid their coupling with combustion (chugging effect).

The injection dome model in [20] contemplates a liquid/vapour/gas bulk approach with liquid front propagation. In [57], injectors are simply represented by a pressure drop.

Combustion

The selected approach for transient combustion in [96], after discarding the considered too simplified *CEA-NASA*, presents ODEs related to pressure and temperature changes during the process, also conceived under an LPM. It is stated in that thesis that no perfect model accounting for all known phenomena has been successfully developed. The effects considered by this model are evaporation of incoming propellants, diffusion, mixing and chemical reactions, based on BELYAEV *et al* work [9]. Convective heat transfer in the combustion chamber is modelled by the complex BARTZ relations, estimating the hot-gas-side heat-transfer coefficient as an empirical function of chamber conditions and geometry.

Combustion modelling in [116] is based on the complex ARRHENIUS formalism. In [40], instead of using the simplified combustion modelling based on propellant mixture tables, a chemical equilibrium model based on GIBBS energy minimisation at each segment is employed, considering a non-adiabatic chamber. Transient conservation equations are obviously selected, also including a dynamic evolution equation of mixture ratio. The ignition flag is active when this ratio is within the permitted limits, then combustion can start after a considered ignition delay.

The combustion chamber in [20] is conceived as a single control volume in which a chemical equilibrium is solved. Heat transfer is modelled by means of the IEWLEW's boundary layer method together with empirical correlations. A propellant preparation time delay is introduced so as to capture the chugging oscillations, which can take place during start-up and shutdown and can be damaging. This delay accounts for droplet break-up and evaporation. Combustion gas data tables with linear interpolations are taken from NASA's GORDON and MCBRIDE *CEA* [53].

The CC is considered in [43] as a zero-dimensional thermo-pneumatic cavity with ideal mixture and without combustion delays. Gas properties at equilibrium are again generated with an in-house code. Besides, ignition is just considered as a function of thresholds in mixture ratio and mass fractions.

Combustion products in [144] are deemed ideal, without variations in the gas mixture constant and standard conservation equations are used. The igniter is of pyrotechnic type and is modelled in an isentropic way, considering conservation of mass in the igniter volume and gas dynamic relations. The resolution of the problem is also supported on the *CEA* model.

Turbomachinery

Turbopumps are typically designed by making use of existing databases of previous successful turbines and pumps, due to their complex mutual coupling and internal phenomena. BELYAEV *et al* approach [9] for pumps is again followed in [96]: pumps performance is described by their head, consumed power and efficiency, and their operation by rotational speed and volumetric flow. Equations synthesis departs from steady-state operation along a wide range of operation of volumetric flows and rotational speeds (cavitation-free), and then it is upgraded thanks to the insertion of transient elements: translational and rotational inertia of the liquid mass inside the pump, computable according to geometry.

Cavitation is a decisive phenomenon in pump's correct behaviour, since it usually leads to stall, erosion or low-frequency oscillations. It occurs if gas bubbles appear, when fluid pressure is beneath

vapour pressure, and then bubbles collapse when gas condenses as pressure increases again. A cavitation boundary on pump's map must be defined analytically and experimentally [96]. It is usually given at points with low pressure ratios and high mass flows.

Rocket turbines can run on liquids or gases depending on the design. They can be either impulsive or reactive, described by the degree of reaction, which expresses the ratio of static enthalpy drop in the rotor to the static enthalpy drop in the whole stage. Their power output depends on their efficiency, pressure gradient and mass flow. The latter is a function of the degree of reaction, which can be expressed as a function of v_R, the isentropic velocity ratio [96]. This ratio contains the actual blade velocity over isentropic velocity. Alterations in the degree of reaction, expressed as variations of v_R, can be triggered by pressure ratio fluctuations. In the end, power and torque can be accurately computed with this set of dependencies. But concerning torque calculation during start-up, indeterminacies appear in the equations for zero rotational speeds. Therefore, an alternative equation to the typical one is proposed in [96].

Turbopump modelling is performed in [40] by introducing *Pratt&Whitney* performance maps in the *EcosimPro*® components. These concern the head and resistive torque of pumps and two other maps for turbines: effective area as a function of velocity ratio at different pressure ratios and two-stage efficiency as a function of velocity ratio.

Turbopumps are treated generically in [20] (as in the Russian literature [9]), allowing the inclusion of gear boxes and multi-axes. Three differential equations serve to represent turbopumps: the balance between rotation speed and torque, the one between pump mass flow rate and momentum equation, and the one between turbine mass flow rate and its momentum. Cavitation effects are taken into account.

Performance maps for turbopumps in [43] come from component tests. Data are extrapolated to extreme operating points by means of an in-house quasi-steady one-dimensional code. Additional unsteady effects related to the hydraulic inertia of fluid inside the pump are added, similarly to [96].

In [57], characteristic curves for the specific torque are considered, which depend on reduced speed and pressure ratio. Torque is then computed by multiplying this specific torque by mass flow, radius and speed of sound. Pumps also present their characteristic curves as second order polynomials for pressure ratio and torque, analogously to [116].

3.1.3 Transient simulation specifically based on transient modelling

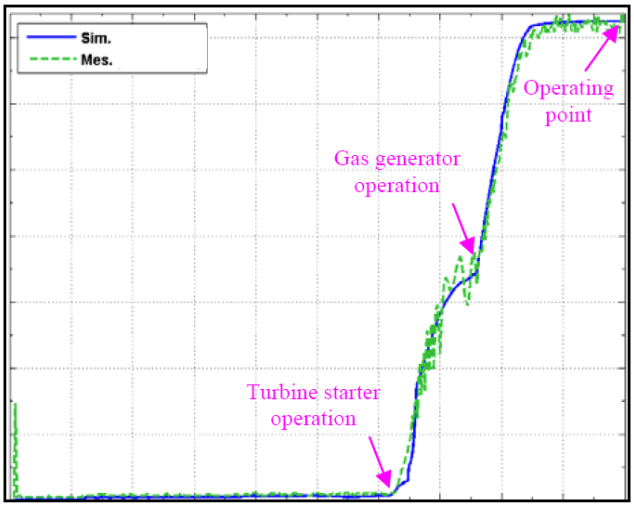
As pointed out by [40], simulations help to correctly tune predefined transient sequences so as to increase engine's safety and reliability and to reduce the number of experimental tests. Not only start-up is simulated in that reference, but also the preliminary phases (chill-down and pre-start), which according to those authors had not been simulated before. They aim at considering the most accurate initial conditions for the start-up.

An example of *CARINS* applicability to LPRE simulation is illustrated in [116], where the start-up transient of the GG engine *HM7B* is simulated. *CARINS* is capable of simulating this kind of discontinuous systems with quick events corresponding to ignitions and valve openings. Indeed, thanks to the inclusion of *Maxima* code, discontinuous functions (e.g. of logical nature) can be well

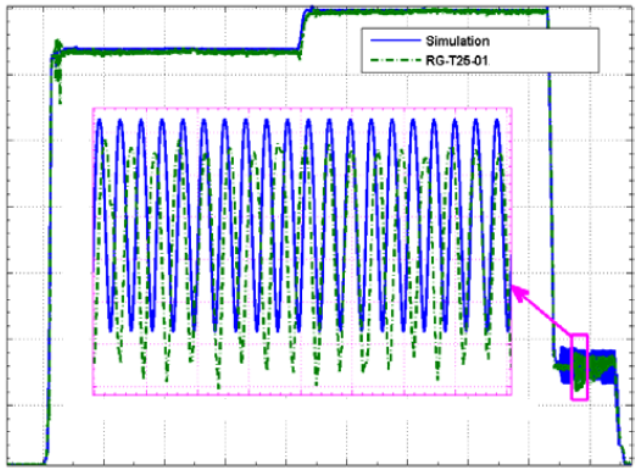
identified, allowing the enhancement of the system of equations around the commutation instants. The integration strategy is based on a state measurement at switching instants so as to obtain a piecewise-integrable differential system.

System simulations inspired on an existing test bench were carried out with the modelling approach in [96], with special emphasis on the simulation of LOX feed lines and of hard ignition combustion. Neither mass flow regulating devices nor turbomachinery are included. However, results can be extrapolated to general rocket engine systems, since the nature of some transient effects is common. Relevant water-hammer effects in the LOX lines are observed, produced once the liquid oxygen fills the lines downstream of the chamber valve and reaches the injection dome. Nevertheless, they are quickly damped, since a lot of energy is dissipated in the form of vibrations.

Satisfactory results are claimed to be obtained in [40] if compared to real ground tests of the same engine *RL-10*. Despite the many uncertainties that can appear in real cases (e.g. in valve discharge coefficients, running shaft torque, initial conditions), their model captures the main processes such as combustion, heat transfer, turbopump operation shift, valve manoeuvring, pressure drop, two-phase flows and also engine's time-to-accelerate. Cooling jacket wall temperature's importance to determine engine's start capability is highlighted.
The previous authors extended their work in another paper [41] so as to simulate shutdown transients. Satisfactory results, similar to tests, are again obtained, correctly capturing chamber pressure tail-off and oxygen mass flow decay. However, they point to some unresolved differences in terms of engine deceleration rates, probably arising from the different time scales of shutdown. These are shorter than for start-up. Besides, the phenomena of cavitation and blade-fluid interaction are not modelled.



(a) BORONINE and FREY pressure build-up result for GG engine using SMART [20]



(b) BORONINE and FREY chamber-pressure results for overall operation of an expander engine using SMART [20], with emphasis on chugging

Figure 3.1.5: BORONINE and FREY transient pressure results [20] (no axes labels). Tests measurements in green and simulations in blue.

Different engine cycles are simulated along transients in [20], such as a cryogenic GG, a cryogenic expander and a hypergolic pressure-fed one. Very accurate results seem to be obtained when compared with ground-test and flight measurement data. Priming, water hammer and ignition peaks are correctly predicted. An interesting example of pressure build-up in the GG engine, with the same

behaviour as in [96], is presented in Figure 3.1.5a. In addition, chamber pressure oscillations including chugging during shutdown are depicted in Figure 3.1.5b. Operating-point changes are also well predicted, as it can be seen in this second figure.

3.2 Hybrid systems modelling for control purposes

The fact that the transient phases of LPREs present some discrete events, which considerably alter the mainly continuous system dynamics, leads to the consideration of a hybrid model in this thesis (Chapter 5). Concretely, the predominantly continuous state-space models derived in this thesis contain certain discrete features, particularly in its control inputs. Thus, the field of hybrid system modelling has been explored.
Most of the common dynamical systems such as airplanes, computers or dishwashers could be modelled in a hybrid way. However, most of the literature is related to fully continuous or to fully discrete modelling, so as to simplify the task. The typical engineering solution to hybrid systems, considering primarily continuous or discrete models, usually employs *ad hoc* methods to put them in interaction, as highlighted by VAN DER SCHAFT AND SCHUMACHER [161].
First, it is worth defining the different discrete features which a hybrid system may present. Above all, the basic phenomena in typical control problems are naturally continuous, normally defined by smooth functions. But if at some points, abrupt variations related to behaviour changes appear, discrete variables must come into play. This way, the different physical modes of the system can be modelled, usually in the form of piecewise-continuous/affine systems [91]. Their switching, represented by variations in discrete variables, might sometimes be related to uncontrollable events [79]. Other types of discrete features can be discrete outputs (sensors), inputs (actuators) and controllers. Actuator state changes consist in controllable events, and an example of discrete or discontinuous control can be gain scheduling.
In fact, the definition of a hybrid system is so general that depending on the application field, different conceptions can be found [161]. In computer science, hybrid systems are mainly a discrete computer dealing with some continuous dynamics, also named as embedded system. In modelling and simulation, different operating modes can be contemplated, usually modelled as instantaneous and discrete transitions. Within the control community there are even more perspectives of looking at this type of systems: hierarchical systems, switching control schemes, relay control, discrete-event systems, etc. Even some nonlinear systems can only be stabilised with switching control.

Several ways of modelling hybrid systems, such as finite state automata or Petri nets, have been proposed in the literature. As mentioned in [91], different modelling techniques have arisen to capture various types of discrete dynamics. Hence, each tool has its own application field. Particular analysis and control synthesis procedures are also related to each of them.
A relevant modelling conception is the Mixed Logic Dynamical (MLD) hybrid framework. FERRARI-TRECATE *et al* [45] succeeded in controlling combined-cycle power plants by using this approach, which includes continuous and discrete dynamics, as well as logic rules. Indeed, some discrete features are inherent to these plants: turbines activation and deactivation and different start-up dynamics. They also relate this problem with an economic optimisation objective with operating constraints (minimum up and down times), solved by MPC.

The MLD framework, first defined by BEMPORAD AND MORARI [10], is chosen by the previous authors since it enables coordination and prioritisation between different agents and is integrable within on-line optimisation techniques. Three major steps are described within this framework. First, logic statements are associated with binary variables. The combination of these statements can be expressed as linear inequalities as functions of the binary variables. Then, if logic variables and linear functions are to be combined, auxiliary variables are defined. Finally, all the defined variables are integrated into a linear time-invariant discrete-time dynamic system. The continuous equations come from piecewise-affine input-output relations, considered as accurate for approximating nonlinear behaviours.
LUNZE *et al* 2009 [91] present how to obtain an MLD representation of a hybrid system by defining first a discrete hybrid automaton (DHA): a connection of a finite state machine and a switched linear dynamical system through an event generator and a mode selector. This DHA can be translated into MLD using mixed integer-linear equations and inequalities. Then, it can be integrated into a hybrid MPC problem, which operates switching linear dynamics, ON/OFF inputs and logic states under certain linear and logical constraints. It is solved by MILP (mixed integer linear programming).
Another interesting example modelled in MLD, explained in [91], is the hybrid control of a solar air conditioning plant [170]. This plant presents up to 13 different configuration modes with different operating dynamics, expressed in terms of discrete and continuous variables. Control actions consist in switched valves and pumps. Within the MLD formulation, configuration modes correspond to different binary variables, upon which the modelling of each component is based. This way, these variables can cancel or activate some terms in the physical equations. A similar approach to MLD are the event-flow formulae proposed by [161].

Some proposals deem separate entities for the continuous and discrete dynamics. According to [105], it always has to be guaranteed that the interaction between the continuous and discrete parts of the system is rigorously defined and that it does not present ambiguities.
ANTSAKLIS *et al* 1993 [4] conceive hybrid control as a part of intelligent autonomous control systems, where a continuous-state plant is controlled by means of a discrete-state supervisory controller. Indeed, that controller is a sequential machine considered as a Discrete-Event System (DES), which receives, handles and commands events represented by symbols. An interface between the plant and the controller is required, since it must guarantee a translation between continuous and discrete variables. Concretely, a generator converts plant outputs into symbols related to plant events, and an actuator is able to follow symbolic commands as constant operations at a certain level. An overview of this scheme can be seen in Figure 3.2.1. As indicated by [4, 79], MOSTERMAN [105] also points to a definition of binary switching operations related to continuous variables by means of intervals and thresholds. According to that author, a sequential control logic is required in order to cope with system's discontinuities, which demand memory savings, said to be implementable in Petri nets or finite state automata.

Petri nets (PN) are similar to automata, but they present emphasised paths and transitions. Their simplest version can be defined by $\langle \mathcal{P}, \mathcal{T}, \mathcal{A}, \mathbf{m_0} \rangle$, where $\mathcal{P}$ is a finite set of possible places, $\mathcal{T}$ is a finite set of transitions, $\mathcal{A}$ is the set of arcs and $\mathbf{m_0}$ the initial marking or state [92]. The main elements are places, represented by circles; and transitions, represented by bars or squares. These are connected through oriented arcs. Reference [12] points out some advantages of PN with respect to automata: they are graphically more intuitive, they allow the representation of all automata but not

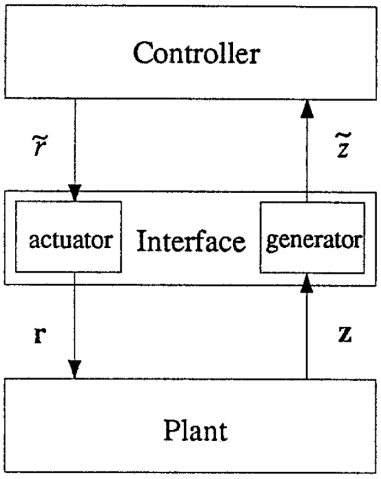


Figure 3.2.1: ANTSAKLIS *et al* hybrid scheme [4]

vice versa and structural information for analysis is easier to obtain. Their typical application field is control of manufacturing or biotechnological processes.
CHAMPAGNAT *et al* [28] summarise the different PN alternatives for modelling of hybrid systems. The simplest one consists in extending a continuous model with boolean variables, in order to express different configuration sequences. Then, there are the timed PN, in which the duration of each place and/or transition is specified. For this sake, good knowledge on the duration of all configurations is presupposed, assuming that they are always independent of past events.
An industrial batch process is analysed in their comparison. However, an interesting analogy to rocket engines can be made, since in these processes, fluids are normally deemed in a continuous way with a start and end event, undergoing thermal exchanges and transformations related to sequences. Within this example, different qualitative states are defined and a different state-space model is assigned to each of them.
In the same direction as [4, 79, 105], reference [28] proposes the combination of PN and the full differential equations, conceiving two interacting models. This way, there is no restriction to the continuous part. Transitions are triggered by means of predefined thresholds on the continuous plant, which are monitored. After each event, the integration of a new system begins, since the PN supervises the variable-structure plan.

As explained in Chapter 5, in this thesis a fairly simple consideration of discrete elements, some control inputs, is made. Inspiration was found on the MLD approaches [10, 91] to define the influence of inputs on the differential equations.

3.3 Predictive, robust and hybrid control techniques

In this thesis, predictive and robust control methods suitable for the LPRE problematic have been evaluated. The selected method in Chapters 6 and 7 is MPC, whose applications and robustness considerations are discussed. Other alternative robust approaches are summarised, such as H_∞ - or

LPV-based ones. The main optimisation solver used, *IPOPT*, is introduced. Some insights into the extension of MPC into hybrid MPC for future work on this topic are also provided.

3.3.1 Model Predictive Control (MPC)

The control strategies in this thesis are mainly based on MPC. 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 chapters. In fact, it is gaining popularity at academic and industrial levels and can take into account robustness [101] or hybrid aspects [91], which are relevant for future work on this topic. Linear MPC dates back from the 70's [101], while NMPC (nonlinear MPC) is more recent, from the 90's [48].

MPC lies on four basic foundations: internal model, reference trajectory (desired path for the output), optimal control sequence and the principle of receding horizon. The internal dynamic model of the system to control is used to predict its state behaviour $\mathbf{x}$ along a time horizon T_p. The output should converge to the desired reference trajectory within that prediction horizon. The objective is to obtain the control sequence $\mathbf{U}$ to minimise a cost function J related to the reference. The problem is discretised in practice according to a selected sampling period Δt, which is the same for state prediction and for the control sequence definition, yielding an amount of N_p steps. The control sequence presents a horizon T_u, N_u that may be lower or equal to state prediction one. The receding horizon concept implies that only the first computed control step (denoted here by $\mathbf{u}_k$) is factually applied to the system. The rest, mainly employed to predict system's behaviour, are either erased or used as a warm start for the next step, since the optimisation of the cost is carried out at each time step k. A visual representation of the method is provided in Figure 3.3.1.

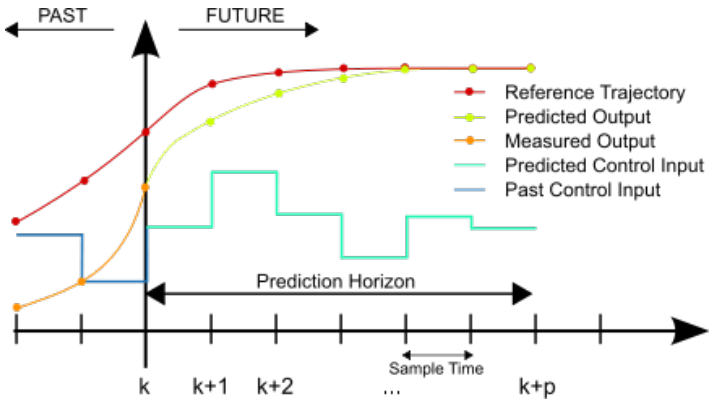


Figure 3.3.1: Schematic principle of MPC, *Martin Behrendt CC BY-SA 3.0*

The cost function J to minimise must convey the expression of the control goals. The most common definition of the cost is in quadratic form, penalising the deviation of states and control from a set point ($\mathbf{x}_r$,$\mathbf{u}_r$). Its continuous-time form is:

$$J(\mathbf{x}(t), \mathbf{u}(t)) = \int_t^{t+T_p} F(\mathbf{x}(\tau), \mathbf{u}(\tau)) d\tau, \tag{3.3.1}$$

where F is generally named as the stage cost [48]:

$$F(\mathbf{x}, \mathbf{u}) = (\mathbf{x} - \mathbf{x}_r)^T Q(\mathbf{x} - \mathbf{x}_r) + (\mathbf{u} - \mathbf{u}_r)^T R(\mathbf{u} - \mathbf{u}_r), \tag{3.3.2}$$

where $Q \in \mathbb{R}^{n \times n}$ and $R \in \mathbb{R}^{m \times m}$ are weight matrices, which serve to accentuate the relative importance of certain states or control inputs (n is the number of states and m is the number of control inputs). The general finite-horizon open-loop optimisation problem is the following:

$$\begin{aligned} \min_{\mathbf{u}} \quad & J(\mathbf{x}(t), \mathbf{u}(t)) \\ \text{s.t.} \quad & \mathbf{x} \in X && \forall t \in [t, t + T_p] \\ & \mathbf{u} \in U && \forall t \in [t, t + T_u] \\ & \dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)) && \forall t \in [t, t + T_p]. \end{aligned} \tag{3.3.3}$$

X and U are the allowable sets of states and control respectively, usually defined as constant box-bound constraints:

$$X := \{\mathbf{x} \in \mathbb{R}^n | \mathbf{x}_{min} \leq \mathbf{x} \leq \mathbf{x}_{max}\} \tag{3.3.4}$$

$$U := \{\mathbf{u} \in \mathbb{R}^m | \mathbf{u}_{min} \leq \mathbf{u} \leq \mathbf{u}_{max}\} \tag{3.3.5}$$

System dynamics is imposed in the constraints of the problem, via $\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}) \quad \forall t \in [t, t + T_p]$. This expression is a general nonlinear definition of the set of differential equations, which must be respected in predictions. Nevertheless, in practice, MPC is commonly discretised for real-time implementation. Throughout the dissertation, $\mathbf{X}$ and $\mathbf{U}$ are stacked vectors with future $\mathbf{x}$ and $\mathbf{u}$ at each time step j along a horizon N:

$$\begin{aligned} \mathbf{X} &= [\mathbf{x}_k, ..., \mathbf{x}_{k+j}, ..., \mathbf{x}_{k+N_p}]^T \\ \mathbf{U} &= [\mathbf{u}_k, ..., \mathbf{u}_{k+j}, ..., \mathbf{u}_{k+N_u}]^T \end{aligned} \tag{3.3.6}$$

Thus, $\mathbf{x}_k$ represents the initial state, which in this thesis is assumed to be measurable. The discretised form of the cost (3.3.1) can be expressed as:

$$J(\mathbf{X}, \mathbf{U}) = \left(\sum_{j=0}^{N_p} \mathbf{x}_{k+j}^T Q \mathbf{x}_{k+j} + \sum_{j=0}^{N_u} \mathbf{u}_{k+j}^T R \mathbf{u}_{k+j} \right) \Delta t. \tag{3.3.7}$$

Hence, the cost consists in the addition of the stage costs at each step along the whole horizon T_p or N_p. The first sum in (3.3.7) penalises the control error along N_p, calculated as the difference between the measured or estimated process output and the reference values, using the process model. The term related to control, as expressed in (3.3.7), penalises the difference with respect to a reference control. However, in the absence of that set point, control variations can be penalised alternatively. After obtaining the optimal value of J (by solving (3.3.3) via direct methods), generally subject to a set of constraints, just the first control signal, $\mathbf{u}_{MPC} \equiv \mathbf{u}_k$, is applied. At the next time step $k + 1$, the same process is repeated by shifting all variables one instant forward, yielding an implicit feedback. Thus, controllers are designed semi-automatically by selecting a set of significant control parameters. In this sense, the basic design parameters in MPC are the prediction horizon, the control horizon and the weighting coefficients. The minimum time step Δt may be defined by computational constraints.

This basic MPC problem can be extended in many senses. In the previous paragraphs, the notion of finite horizon has already been introduced. It is necessary for the on-line resolution of the optimisation problem. Ideally, an infinite horizon would minimise the cost and ensure a perfect prediction of the actual closed-loop behaviour (in the absence of model-plant mismatch and disturbances) [48]. Thus, in the applicable finite-horizon scheme as presented in (3.3.3), there are no explicit guarantees of CL stability. This is normally solved in the literature by adding adequate constraints or penalty terms, especially in nonlinear systems [48, 101]. In the work by CHEN AND ALLGÖWER [30], both an end-state inequality constraint and penalty term are defined in their quasi-infinite horizon (QIH) approach. Concretely, the last step in the prediction horizon is specially constrained to a terminal region Ω, which, together with the terminal penalty term E, have to be designed to enforce stability and recursive feasibility. In this sense, the MPC drives the system to that region, where a fictitious local controller K would perform the precise tracking at the end of the state prediction horizon, T_p. That controller is never used in practice (unless a control switch is implemented), because only the first computed control is transmitted to the plant. Hence, the real role of the fictitious feedback $\mathbf{u}(t+T_p) = K\mathbf{x}(t+T_p)$ is to compute Ω and E. This calculation is performed by linearising the model and by defining as a LYAPUNOV function $V(\mathbf{x}) = \mathbf{x}^T P\mathbf{x}$, whose P matrix is the solution of the following continuous-time LYAPUNOV equation:

$$(A_K + \kappa I)^T P + P(A_K + \kappa I) = -Q_K - K^T R_K K, \tag{3.3.8}$$

where A_K is the dynamic matrix in CL (with K), $\kappa \in \mathbb{R}^+$ (such that $\kappa < -\lambda_{max}(A_K)$) and Q_K and R_K are positive definite symmetric matrices $Q_K \in \mathbb{R}^{n\times n}$, $R_K \in \mathbb{R}^{m\times m}$. Then, E is defined as $E = V(\mathbf{x}(t+T_p))$, which is added to the stage cost in J. The terminal region constraint is expressed as $V(\mathbf{x}(t+T_p)) \leq \alpha_P$. More details about the calculation of α_P can be found in [30] and in Chapter 6. Basically, the verification of that inequality constraint ensures the reduction of the cost as

$$\dot{J}(\mathbf{x}, \mathbf{u}) < 0 \quad \forall \mathbf{x} \in \Omega, \tag{3.3.9}$$

which consists in a LYAPUNOV stability argument (introduced in Section 3.3.2), together with the fact that the cost is always positive. Terminal equality constraints are also a possible approach [101] but they are generally more constraining in a real-time implementation.

Applications

The applications of MPC can be found in multiple disciplines. It was originally applied to slow chemical systems (such as distillation columns [110]), where computational times were not problematic [11]. However, due to its advantages and to the evolution of processors, faster applications have been addressed. In this sense, the design approach can be strongly constrained according to the maximum time interval in which the controller has to provide an actuation signal.

For instance, Diesel engines have been controlled with MPC in [107, 159, 160]. In [160], so as to cope with the fast behaviour of the engine, the algorithm is simplified by relaxing control constraints via penalty functions and by linearising dynamics and interpolating between a set of operating points. In [107], the set of control decision variables is reduced by parametrising the control vector according to some pre-computed OL profiles. In this manner the computational effort diminishes, even if they

use nonlinear dynamic constraints. Reference [159] applies its self-adaptive predictive controller to a Diesel engine too. In that case, a proposal for an automatic optimal selection of MPC parameters, for linear systems and for a class of nonlinear ones, is made.

MPC has been applied to other thermodynamic systems, such as a cryogenic refrigerator in [19]. Some solutions for achieving fast MPC are addressed in that article. A variation of Δt is proposed during the linearised MPC execution. Moreover, the use of less efficient but faster optimisers is suggested for some phases. They also affirm that some unpredictable loads (thermal) can lead to constraint violations. Hence, they are systematically relaxed through a constraint-violation penalty term in the cost.

Furthermore, MPC-like schemes can not only be used on-line, but also serve to compute off-line optimal control. For instance, trajectory-planning or generation of reference trajectories for aerial vehicles is performed by off-line MPC in [68, 148]. Other trajectory-planning methods have been proposed, especially in the field of robotic motion, but they are also mostly based on optimisation [15].

Many more MPC applications can be found in the literature; some of them are commented in Section 3.3.4.

Robustness

The previous algorithms assume that the predictions obtained from the state-space models are identical to the real plant behaviour. However, if there is model-plant mismatch or perturbations of endogenous or exogenous nature, some deviations may be present in the final CL trajectories. Thus, some modifications of the original problem (3.3.3) have to be carried out in order to increase the robustness of the controller. The most common robust approaches rely on *minimax* optimisation or on tube-based invariant sets, as indicated by MAYNE *et al* [100, 101]. Other less common approaches summarised in [101] combine H_∞ with MPC.

The *minimax* approaches are well described in the thesis of LÖFBERG [82, 83]. The goal here is to consider the worst case arising from the effects of an endogenous or exogenous perturbation in the algorithm. Thus, there is a nested maximisation of the cost as a result of that disturbance:

$$\begin{aligned} \min_{\mathbf{u}} \max_{\mathbf{w}(t)} \quad & J(\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t)) & \\ \text{s.t.} \quad & \mathbf{x} \in X & \forall \mathbf{w} \in \mathbb{W}^n, \forall t \in [t, t+T_p] \\ & \mathbf{u} \in U & \forall \mathbf{w} \in \mathbb{W}^n, \forall t \in [t, t+T_u] \\ & \dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t)) & \forall \mathbf{w} \in \mathbb{W}^n, \forall t \in [t, t+T_p]. \end{aligned} \tag{3.3.10}$$

Nevertheless, solving this problem as defined in (3.3.10), with compact sets of possible perturbations $\mathbf{w} \in \mathbb{W}$, is generally not tractable on-line. Thus, it is often modified for enabling an efficient resolution. In [83], conservative approximations of *minimax* MPC via semidefinite relaxation are developed. Basically, all robustness and control-goal considerations are shifted to constraints in an epigraph formulation of the objective function. Indeed, the cost is simply a scalar $\gamma \in \mathbb{R}^+$ which constrains the

original quadratic cost for all possible perturbation values in compact sets. The discretised version of this modified problem is:

$$\begin{aligned} \min_{\mathbf{U},\gamma} \quad & \gamma \\ \text{s.t.} \quad & J(\mathbf{X},\mathbf{U},\mathbf{W}) \leq \gamma && \forall \mathbf{W} \in \mathbb{W}^n \\ & \mathbf{X} \in X && \forall \mathbf{W} \in \mathbb{W}^n \\ & \mathbf{U} \in U && \forall \mathbf{W} \in \mathbb{W}^n \\ & \mathbf{x}_{k+1+j} = g(\mathbf{x}_{k+j}, \mathbf{u}_{k+j}, \mathbf{w}_{k+j}) && \forall \mathbf{W} \in \mathbb{W}^n, \forall j \in [0, N_p] \end{aligned} \tag{3.3.11}$$

The problem is then translated into a semidefinite programme, in which constraints are expressed via LMI. The dynamics g is considered linear in the previous proposal (3.3.11). Different algorithm proposals are made for the inclusion of additive disturbances in the linearised dynamic equations and for the consideration of LFT systems with uncertain A and B matrices. The dynamics with additive perturbations $\mathbf{w}_k$, with their associated G matrix, is of the form:

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k + G\mathbf{w}_k, \tag{3.3.12}$$

while LFT dynamics presents their standard form. Another relevant article synthesising a robust *minimax* MPC controller consisting in a translation to an LMI programme is [67], with an infinite-horizon cost approach. Uncertainty is considered in the form of polytopes or of structured LTI (Linear Time-Invariant) systems.

Another usual simplification of the *minimax* programme is the consideration of a finite set of perturbed cases or scenarios, the scenario-based MPC [95]. In this manner, starting from (3.3.10), the maximisation or worst-case search is performed among the possible $\mathbf{w}_i$, where $i \in I = \{1, ..., \mathrm{i}\}$ represent the different finite scenarios. The papers by CALAFIORE *et al* [25, 26] are based on this philosophy and simultaneously manage a priori assigned probabilities of the occurrence of each scenario. This is claimed to be a valid alternative to deterministic MPC, because it remains a convex problem and can be efficiently solved. The fact of using statistical information on disturbances connects that approach to the stochastic MPC techniques, such as [14], where stochastic variables come into play.

Further approaches to consider uncertainties in robust *minimax* algorithms are summarised in the survey by BEMPORAD AND MORARI [11]. Apart from the polytopic uncertainty and structured feedback uncertainty considered in [67], the multi-plant description, in which different plants are used according to the disturbance, is proposed. The research direction concerning the construction of robustly invariant terminal sets is also highlighted. In this sense, the tube-based robust MPC [100] faces bounded state disturbances via the construction of an invariant region around the nominal trajectory, in which all possible state propagations are ensured to lie. The constraint sets on states and input are rendered tighter for guaranteeing robustness.

In order to reduce the computational times of *minimax* optimisation, which in some cases can surpass the allocated time, some authors propose different solutions. In [106], an explicit robust MPC scheme is described, in which off-line multi-parametric optimisation is carried out. Indeed, in explicit MPC,

the optimised control is precomputed according the different parametric variations. In that paper, a suboptimal solution of the [67] controller is obtained as a piecewise-affine function over a partition. Another approach, presented in [32], proposes the off-line computation of constraints restrictions suitable for rejecting bounded persistent bounded disturbances. These new constraints are a function of the $\mathbb{W}$ set and related dynamics.

Further extensions of MPC have been explored in the literature. The extension of MPC to hybrid systems is explained in Section 3.3.4, since it relates to the future work of the subject of this thesis.

3.3.2 Further robust control methods

Other robust control methods, discarded for this thesis as explained in Chapter 6, have been proposed in the literature. Their analysis is relevant so as to justify the selection of MPC as control approach, explained in that chapter. The most used are based on H_∞, LPV (Linear Parameter-Varying) systems and LYAPUNOV-based design. In general terms, robust approaches present larger stability margins and maintain performance levels in the presence of perturbations or modelling error. Normally, it is aimed at synthesising controllers from simplified models which still perform on the real plants, often too complicated to be accurately described by a set of linear differential equations.

As introduced in 2.4.4, the H_∞ method has been used in some LPRE control studies [71, 141]. Basically, this method seeks the minimisation of the infinity norm H_∞ of a system, which is defined as the superior bound of the singular values σ of an input-output transfer function $G(s)$ of the system across all frequencies ω [80]:

$$H_\infty = ||G(s)||_\infty = \sup_\omega \bar{\sigma}(G(j\omega)). \tag{3.3.13}$$

In this manner, the effects of disturbances are expressed in that norm, whose minimisation enforces the maximisation of the module margin in the system. That margin is defined as the minimal distance between the NYQUIST locus and the critical point. The singular values consist in a generalisation of the notion of gain to multivariable systems. So as to design a controller $K(s)$, the system is generally expressed in the so-called standard form, in which the relation between all exogenous and endogenous inputs and outputs is made clear. The global closed-loop transfer function is $F_l(P(s), K(s))$, where $P(s)$ concerns the transfer between exogenous variables. Exogenous inputs take into account disturbance, noise and reference signals. Exogenous outputs are expressions of performance and robustness criteria. Hence, an elaborate and complete model is required. The controller feeds the measured outputs $\mathbf{y}$ back to control inputs $\mathbf{u} = K(s)\mathbf{y}$. The goal is to minimise the H_∞ norm of the whole plant in standard form, so as to obtain an optimal $\hat{K}(s)$ [80, 162]:

$$\hat{K}(s) = arg\min_K ||F_l(P(s), K(s))||_\infty. \tag{3.3.14}$$

It is usually not viable to minimise simultaneously that norm on the whole frequency range. It appears that the performance level is only critical at low frequencies, while robustness versus modelling errors is a goal at higher frequencies. Thus, weighting functions are generally used in H_∞-based control loops. In addition, in order to obtain the standard-form representation, LFT is a usual way of representing uncertainties in the system, well suited for μ-analysis [141]. The resolution of (3.3.14)

can be carried out in different manners, such as via LMI-constrained or non-smooth optimisation, which can be time-consuming.
Approaches based on H_∞ have been successfully applied to aerospace systems which can be modelled with relatively short state vectors (under five), for instance to rotorcraft [49], satellite attitude control systems [103] or launcher flight control [162].

Robust techniques for LPV systems, which can also be combined with H_∞ as in [162], are also very present in the literature. This system representation provides a framework for considering the influence of parametric variations in linear dynamic equations. If $\mathbf{p}$ represents a parameters vector, this dynamics can be represented as [21]:

$$\dot{\mathbf{x}} = A(\mathbf{p})\mathbf{x} + B(\mathbf{p})\mathbf{u}, \tag{3.3.15}$$

$$\mathbf{y} = C(\mathbf{p})\mathbf{x} + D(\mathbf{p})\mathbf{u}. \tag{3.3.16}$$

Moreover, the parts depending on $\mathbf{p}$ do not necessarily correspond to real parametric variations, but they can also serve to simplify nonlinearities. For instance, one can have px instead of x^2. Polytopic or LFT representations can also be covered in this representation [21]. In order to control these systems, it is common to establish a minimisation goal, such as an $\mathbb{L}_2$ norm (Euclidean norm) in CL, and then solve a parametrised-LMI problem [22], where the control law explicitly depends on varying parameters. The resolution of that LMI-based problem can also become time-consuming.
This methodology has been applied to aerospace systems with relatively few states, such as aircraft flight-control systems [50], turbofan engines [51, 125] or spacecraft pointing [69].

Apart from these optimisation-based techniques, another perspective of robust control from the general theory of nonlinear systems is provided by KHALIL [62], based on the LYAPUNOV theory. LYAPUNOV functions are state-dependent expressions of some sort of potential energy in a system which should be reduced with time to demonstrate stability. If they are always positive (except at the origin) and their time derivative is negative semi-definite in a given state neighbourhood Ω, the system is said to be stable in Ω. The LYAPUNOV-based robust design approach [62] considers a general nonlinear system in the form:

$$\dot{\mathbf{x}} = f(\mathbf{x}) + B(\mathbf{x})\left[G(\mathbf{x})\mathbf{u} + \delta(t, \mathbf{x}, \mathbf{u})\right], \tag{3.3.17}$$

where G and δ are uncertain functions, including the modelling uncertainties of f and B. Different techniques are proposed to obtain a robust state feedback for the control-affine system (3.3.17) with matched uncertainties (at the same level as $\mathbf{u}$ in the state equation). One approach is Sliding-Mode control, where the system is driven to an equilibrium sliding surface or manifold. The control design has to ensure that the system remains there, which is accomplished via LYAPUNOV arguments. Another technique, the LYAPUNOV redesign, concerns the synthesis of a feedback law by making use first of a LYAPUNOV function of the nominal system. To this nominal law, an additional term which faces uncertainties is added, which is obtained via bound inequalities. These techniques lead to discontinuous controllers which may suffer from chattering, an excessively fast control switching. Applications to aerospace systems are for instance [34] for spacecraft attitude control and [145] for missile guidance.

Further robust techniques, highlighted in [1], are passivity-based control [62], robust saturations or variable-structure control [64].

3.3.3 Optimisation solver tool: *IPOPT*

Optimisation problems are solved in this thesis via the open-source software package *IPOPT* (Interior-Point Optimizer) [164]. Other tools for solving general constrained nonlinear programming (NLP) have been used during the algorithm development, like *MATLAB*® *fmincon* and *YALMIP* [84, 85]. *IPOPT* has been selected thanks to its efficient interior-point conception, which intrinsically respects barriers on decision variables; and to its coding flexibility, allowing the introduction of all user-defined gradients and Hessians in the calculation.
In that tool, large-scale NLP problems of the following form are considered [164]:

$$\begin{aligned} \min_{\mathbf{y}\in\mathbb{R}^N} \quad & f(\mathbf{y}) \\ \text{s.t.} \quad & \underline{\mathbf{g}} \leq \mathbf{g}(\mathbf{y}) \leq \overline{\mathbf{g}} \\ & \underline{\mathbf{y}} \leq \mathbf{y} \leq \overline{\mathbf{y}}, \end{aligned} \tag{3.3.18}$$

where $\mathbf{y} \in \mathbb{R}^N$ are optimisation variables, with lower and upper bounds $\underline{\mathbf{y}} \in (\mathbb{R} \cup -\infty)^N$ and $\overline{\mathbf{y}} \in (\mathbb{R} \cup \{\infty\})^N$ respectively; $f : \mathbb{R}^N \to \mathbb{R}$ is the cost function and $g : \mathbb{R}^N \to \mathbb{R}^c$ are general nonlinear constraints (an amount of c), with lower and upper bounds $\underline{\mathbf{g}} \in (\mathbb{R} \cup -\infty)^c$ and $\overline{\mathbf{g}} \in (\mathbb{R} \cup \{\infty\})^c$ respectively. The functions $f(\mathbf{y})$ and $g(\mathbf{y})$ can be convex or non-convex, but need to be twice continuously differentiable. Thus, only smooth problems are covered.
The resolution of this NLP is carried out via an interior-point line search filter method, which targets local solutions of (3.3.18). The barrier approach computes approximate solutions for a sequence of barrier problems. The algorithm also includes a feasibility restoration phase for the filter method, second-order corrections and inertia correction of the KKT (Karush-Kuhn-Tucker) matrix. Heuristics are also taken into account to speed up the computation.

IPOPT's straightforward *MATLAB*® interface presents the function syntax
$[\mathbf{y}, info] = ipopt(\mathbf{y}_0, funcs, options)$, whose arguments are:

- $\mathbf{y}_0$: vector of initial values for the primal variables, length N.
- *funcs*, a structure with the fields:
  - *funcs.objective*: cost function $f(\mathbf{y})$.
  - *funcs.gradient*: gradient of cost function.
  - *funcs.constraints*: constraints function, linear or nonlinear (if present).
  - *funcs.jacobian*: sparse Jacobian of constraints (if present).
  - *funcs.jacobianstructure*: sparse matrix indicating the structure of constraints Jacobian, filled with ones and zeros (if constraints present).
  - *funcs.hessian*: sparse Hessian of the Lagrangian L (needed in optimisation), which is defined as: $L(\mathbf{y}, \sigma, \lambda) = \sigma f(\mathbf{y}) + \lambda^T g(\mathbf{y})$, where $\sigma \in \mathbb{R}$ and $\lambda \in \mathbb{R}^c$ are optimisation's Lagrangian multipliers. The user-definition of the Hessian is optional. Otherwise the Hessian is approximated.

  - *funcs.hessianstructure* sparse structure of the Hessian of the Lagrangian L (optional, if the exact Hessian matrix is provided).
- *options*, a structure with the fields:
  - *options.lb*: decision-variables lower bound $\underline{\mathbf{y}}$.
  - *options.ub*: decision-variables upper bound $\overline{\mathbf{y}}$.
  - *options.cl*: constraints lower bound $\underline{\mathbf{g}}$.
  - *options.cu*: constraints upper bound $\overline{\mathbf{g}}$.
  - *options.ipopt*: series of options, regarding for instance tolerances and the Hessian consideration.

The consideration of sparse matrices and the fact that gradients, Jacobians and Hessians can be provided alleviate the computational burden, which is important in on-line MPC. Further interfaces in *C++* and other programming languages are available, which enlarges its versatility. Plenty of applications of *IPOPT* are present in the literature; for example, in the fields of robotic motion [104], chemical engineering [7] and energy converters [2].

3.3.4 Insights into hybrid control techniques

Matching the aforementioned model structures in 3.2, it seems interesting to provide an overview of the related hybrid control methods present in the literature, from which the hybrid algorithm proposed in 7.3 is inspired, and which could be useful for future extensions of the controllers in this thesis.
In the book by LUNZE *et al* [91], several control design trends for hybrid systems are indicated depending on the application: LYAPUNOV-based controllers and LMI for switched systems; optimal control for manufacturing; and MPC mixed with MLD for piecewise-affine systems (PWA). Other strategies are proposed in [161], such as pulse width modulation, sliding mode control, quadratic stabilisation of multi-modal linear systems and switching-control schemes (such as [146]). Concerning the latter family, it has been observed that some nonlinear continuous systems are controllable and stabilisable through these schemes, but they are not so by classical continuous feedback. In fact, switching logics are common to several control methods such as relay control, gain scheduling or fuzzy control [46]; either by considering plant evolution or controller switches from one regime to another. There exists the typical conception of switching logic in reconfigurable control, where a supervisor is in charge of selecting the controllers to be executed. But there is also the conception of a single controller box which generates a different control signal according to the symbolic instructions of the supervisor.
The most relevant hybrid-control trends, which could be envisaged for the continuation of the works developed in this thesis, are Hybrid MPC and MPC on Petri nets.

Hybrid MPC (HMPC)

Interesting applications of MPC on hybrid systems are explained in [91]. The studied plants are a supermarket refrigeration system [136], a multi-stage evaporator system [151] and a solar air conditioning plant, from which an analogy to rocket engines can be drawn, since these systems require that some temperatures and pressures remain between certain values while facing environmental

disturbances. Moreover, the optimisation criteria are minimum energy consumption and minimum wear of compressors for the refrigerator system, and minimal start-up time and minimal resource consumption for the evaporator.

Regarding the supermarket refrigerator control by SARABIA *et al* [136], nonlinear dynamics is switched-continuous and the inputs are discrete (ON/OFF). In other words, some equations are switched as a function of the inputs. Constraints on these inputs are treated linearly, but the process dynamics is deemed nonlinear and hence a mixed integer nonlinear programming can be solved at each sampling step within an NMPC approach. However, this can lead to cumbersome computations if executed on-line for a high-dimension system. Therefore, the solver has been simplified by translating binary decision variables into continuous variables by parametrising them in terms of pulses durations. If the optimiser works on the duration of the ON/OFF operations (real numbers), the solver becomes a continuous nonlinear programming problem, whose cost functions penalise temperature and pressure deviations from the objective ranges. This is executed on-line every sampling period following the sequential approach depicted in Figure 3.3.2:

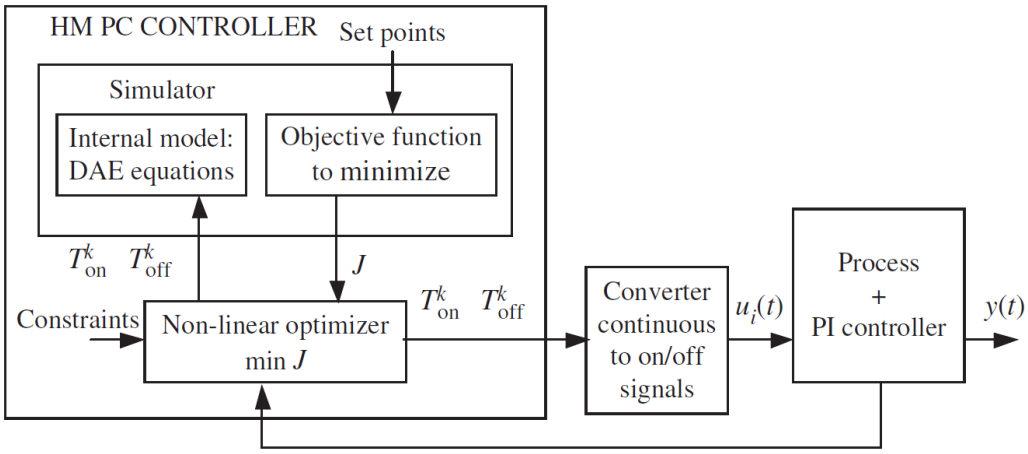


Figure 3.3.2: Hybrid NMPC implementation by SARABIA *et al* [136]

Concerning the evaporator system, its $3h$-long start-up has been optimised by SONNTAG *et al* [151] through finite-horizon formulation with nonlinear optimisation of the continuous variables and branch-and-bound search (decision tree branches exploration) over the discrete variables. This system changes of model structure after valves switching and shows different continuous dynamics at different phases. For complex systems such as this one, where only short horizons may be plausible, those authors recommend to base the choice of cost functions on heuristic knowledge of the optimal trajectories.

In order to control a solar air conditioning plant, a heuristic predictive logic controller (HPLoC) based on MPC and on knowledge of the process is implemented in [171]. Two loops are considered: the outer one determines the operating mode through the HPLoC and the inner one controls continuous variables via an MPC specific to the mode. The logic controller performs an integer optimisation problem through an analytic hierarchy process.

A supervisory hybrid model predictive control strategy for stabilising voltage in power networks has been proposed by NEGENBORN *et al* [112]. Their system is hybrid because discrete control variables and hierarchical loops are present. Indeed, regulation is performed at different levels corresponding to spatial and/or temporal partitioning. At the lowest level, controllers act directly on system's actuators. The higher levels monitor these lower levels so as to furnish reference operating points and constraints.

MPC on PN. The recent work by JULVEZ *et al* [59] proposes a hybrid framework related to the MLD systems formulation by [10]. Basically they transformed hybrid systems expressed as hybrid PN into an event-driven MLD and then applied event-driven MPC on it, attaining a MILP problem.
The introduced control actions change whenever a discrete event occurs, corresponding to mode switching in continuous time. The control input is thought as a piecewise-constant function with non-constant step duration. These actions enable transition firings. In the continuous cases an opening degree is provided, in analogy to a valve; and in the discrete ones a delay can be set.
That event-driven control conception aims at reducing model mismatch between discrete and continuous parts mainly caused by sampling. The resulting event-driven MLD (eMLD) obtained represents net's marking evolution between two events, after having translated the net into the standard piecewise-linear system, by means of inequalities and conditional statements. Then, MPC is applied on it for a selected number of events and positive results are attained in less computational time than standard MPC. The performance of continuous-time approaches is kept while using the lower computational times of discrete-time ones. Nevertheless, LPREs could be hardly expressed as PN, the start point of that approach.

The application of the aforementioned hybrid techniques to the control of LPREs is discussed in Section 7.3 and in the Conclusions and Perspectives Chapter.

3.4 Further tools: SOBOL sensitivity analysis and Kriging-based techniques

Other tools used in the thesis which are introduced here are the SOBOL sensitivity-analysis method and Kriging-based techniques.

SOBOL sensitivity-analysis method. In order to study the sensitivity of state-space models to internal parameter variations, which is a system-analysis step prior to control robustification, a sensitivity-analysis method was selected. It had to be appropriate for large-scale nonlinear systems such as the one presented in Chapters 4 and 5. In this sense, as explained in Chapter 5 the SOBOL sensitivity-analysis method [149] was chosen as an adequate tool for computing global parameter sensitivity indices, as recommended in [172, 177].
Due to its statistical approach, the SOBOL method can indeed capture the global effects of parameter variations on a given scalar criterion or output function $f(\mathbf{p})$. All parameters $\mathbf{p}$ are varied simultaneously over predefined spans, enabling the evaluation of the relative influence of each individual parameter, as well as their interactions, on the model output variance. The first step of the SOBOL analysis is to generate a series of sampled parameter sets $\mathbf{p} = [p_1, p_2, ..., p_s]$, rescaled to $[0, 1]$, with

their corresponding $f(\mathbf{p})$. To do so, a uniformly distributed quasi-random Monte Carlo integration is carried out. The $f(\mathbf{p})$ is considered as a random variable with mean f_0 and variance D:

$$f_0 = \int f(\mathbf{p}) d\mathbf{p}, \tag{3.4.1}$$

$$D = \int f(\mathbf{p})^2 d\mathbf{p} - f_0^2, \tag{3.4.2}$$

The basis of the method lies on the decomposition of D into contributions of single parameters variations, of second-order combinations, third-order, etc.:

$$D = \sum_{i=1}^{s} D_i + \sum_{i<j} D_{ij} + \sum_{i<j<l} D_{ijl} + ... + D_{1,2,...,s}, \tag{3.4.3}$$

where $D_{i_1...i_s} = \int f^2_{i_1...i_s}(p_{i_1}, ..., p_{i_s}) dp_{i_1}, ..., p_{i_s}$ consists in the variance of $f_{i_1...i_s}$. With these variances, the SOBOL sensitivity indices S can be computed as:

$$S_{i_1...i_s} = \frac{D_{i_1...i_s}}{D}, \tag{3.4.4}$$

according to the respective interactions order. The total-order sensitivity indices, capturing the global effects of a single parameter are then obtained as the addition of all the sensitivity indices as $S_{Ti} = S_i + S_{ij_{i\neq j}} + ... + S_{1...i...s}$.
Other comparable methods, such as FOURIER amplitude sensitivity analysis (FAST), follow the same basic algorithm while the multidimensional integration of sensitivity indices is different. In FAST, sinusoidal functions are used for pattern search instead of Monte Carlo integration.

Kriging-based techniques. In the case where multiple costly evaluations of a performance criterion $f(\mathbf{p})$ are required, either for SOBOL analysis or for any type of optimisation, it is of interest to substitute that simulation by a faster alternative. In this manner, Kriging models enable the immediate evaluation of an input-output relation via the prior processing of simulated data. In these models, the function f is interpolated by modelling it as a Gaussian process F [78]:

$$F(\mathbf{p}) = \mathbf{q}^T(\mathbf{p})\mathbf{b} + Z(\mathbf{p}), \tag{3.4.5}$$

where $\mathbf{q}(\mathbf{p})$ is a known regression function vector, $\mathbf{b}$ contains unknown regression coefficients to be estimated, and $Z(\cdot)$ consists in a zero-mean Gaussian process with a parametrised covariance function $k(\cdot, \cdot)$

$$k(Z(\mathbf{p}_i), Z(\mathbf{p}_j)) = \sigma_Z^2 R(\mathbf{p}_i, \mathbf{p}_j), \tag{3.4.6}$$

σ_Z^2 and $R(\cdot, \cdot)$ being the process variance and a parametric correlation function respectively. Both of them are either chosen or estimated from the data set. The Latin Hypercube Sampling (LHS) is a general method for generating random input parameter sets [172]. Different choices can be made for $R(\cdot, \cdot)$ [98]. Usually, the closer the samples are, the higher their correlation is assumed. Indeed, Kriging, also called Gaussian process regression, is the search for the best linearised unbiased predictor by exploiting the statistical information of the provided points. An advantage with respect to deterministic interpolation methods such as Splines, is the fact that the variance or accuracy of the prediction error can be computed.

In this manner, it can be used for optimising a set of parameters that influence a costly simulation, which can be considered as a black-box function. By generating an initial parameter set via LHS and then executing the simulation on those random sets, a Kriging model can be built. Reference [98] is an example of application to the automatic tuning of fault detection and isolation methods. Furthermore, the Kriging model can also be used for carrying out SOBOL analyses and avoiding costly evaluations at each Monte Carlo integration [58], as performed in Section 5.4.5.

3.5 Summaries

The next paragraphs serve to summarise the topics introduced along this chapter.

Transient modelling of LPREs. After reviewing the transient behaviour and modelling of LPREs, it seems clear that their main transient phases, start-up and shutdown, are complex physical processes. An accurate planning of these key operating phases has to be performed, with the main goal of safely attaining an end state. The most reliable techniques for well predicting their behaviour count on precise transient models, whose complexity varies substantially depending on the objective of the simulation. As seen in the literature, many different physical modelling approaches can be selected, there is not a single way of describing the behaviour of LPREs. General assumptions for the whole cycle, as well as intra-component hypotheses have to be carefully made depending on the kinds of phenomena to be captured. For instance, the inclusion of high-frequency combustion phenomena complexifies the modelling to a great extent. The more phenomena considered, the more engine parameters to be estimated. One of the most generalised approaches is the consideration of zero-dimensional modelling, neglecting spatial contributions, in order to obtain a set of ODEs instead of PDEs. All in all, the purpose of the simulation conditions the modelling considerations.
Start-up and shutdown consist of an initial sub-phase determined by some discrete events, and a subsequent one where a purely continuous behaviour is given. The distinction between these two sub-phases is paramount in this thesis, where different approaches to the control of each one are proposed in Chapters 6 and 7.

Predictive and robust control. Several techniques for the consideration of robustness in control design are present in the literature. The general goals are to maintain system performance in the event of perturbations and/or cope with modelling uncertainties when facing real plants. Approaches based on H_∞ and LPV systems can be designed with the help of LMI optimisation schemes. Nonlinear methods, such as sliding mode and LYAPUNOV redesign [62], are appropriate for deriving formal feedback laws for control-affine systems. The selected control method in this thesis, Model Predictive Control, can also take into account robustness to additive disturbances or to parametric variations. This predictive control method makes use of system models along predefined horizons in order to compute an on-line optimal control that respects state and input constraints. Stability terms, as well as robustness considerations, can be added to the scheme. Worst-case *minimax* optimisation as well as tube-based design are the main robust MPC approaches. For their resolution, LMI-based schemes have also been proposed, but they are not the only alternative to solve robust MPC. The concrete control-method selection and justification in accordance with the system studied in this thesis is explained in Chapter 6.

Hybrid systems and their control. The combination of continuous and discrete features for control is the general definition of hybrid control, which is still an evolving field. There are several ways of defining a hybrid system or hybrid control law. This definition is very dependent on the application, on the nature of discrete and continuous dynamics and their interaction. There are approaches concerning a global modelling of continuous and discrete dynamics within a common hybrid automaton [161] or state vector [52]. Other authors consider two separate entities that interact by means of a translation element [4]. For the discrete part, Petri nets are a widespread solution [28], which have been connected to different sets of continuous differential equations. Another approach, the MLD framework [10] includes continuous and discrete dynamics, as well as logic rules, associated to binary variables. A simplified conception of that one is performed in this thesis, where there are some discrete inputs and also the notion of an evolving system of equations, precisely triggered by these inputs. In other words, certain terms in the equations are cancelled or activated by binary variables. Indeed, this is inspired by some other hybrid thermodynamic benchmarks [45, 91]. According to [91], the most adequate control technique fitting this system representation is HMPC. This technique consists in optimising the future control actions of discrete and continuous nature according to a prediction of system's behaviour based on a hybrid model. Other applicable techniques are fuzzy logic [46] or switching control [146]. The strategy in [136] inspires the proposed hybrid algorithm in 7.3.

Part II

Developed approach

4 Construction and validation of a thermo-fluid-dynamic simulator representative of gas-generator-cycle LPREs

The objective of this chapter is basically to develop the block in the control loop (depicted in Figure 6.1.1) corresponding to the engine simulation or plant. A representative yet simple thermodynamic simulator of a GG LPRE in *Simulink*®, capturing its transient behaviour, was sought.

Due to the need for an enlarged operating domain, to quick variations during transients and to the natural nonlinearity of thermo-fluid-dynamic equations, models in this thesis are nonlinear. They also capture the varying dynamics corresponding to configuration changes driven by sequential-logical events (valves opening, main chamber ignitions or starter activations). This is important because it is during these transient phases that the main temperature and pressure peaks occur, and it is vital to avoid them in order to extend engine life.

The main target is a GG cycle because *PROMETHEUS*, the potential engine application of these studies, is of this type. However, since the modelling approach here begins at a component level, other engines and cycles can also be described. Even though the dynamic equations are defined inside each component (such as chambers or valves), due to the natural interdependencies within the cycle, a global set of differential equations is subsequently obtained.

Section 4.1 is devoted to introducing the GG-cycle engines *Vulcain 1*, the main case study in this thesis, as well as *PROMETHEUS*. Section 4.2 presents the thermo-fluid-dynamic modelling formulation in a component-wise way, generic to rocket engines. In that section, the cycle interconnection and simulation conditions are also explained. Finally, the behaviour of the obtained simulator is validated with respect to the expected one in Section 4.3.

4.1 Gas-generator-cycle (GG) engines description

The basic generalities of LPRE cycles have been introduced in Section 2.1. In this Section, more precise descriptions of the two engines modelled in this thesis, *Vulcain 1* and *PROMETHEUS*, are provided. Both of them present a GG cycle, and hence their working principle is similar. A schematic of this type of LPRE is shown in Figure 2.1.2. The *Vulcain 1* will be introduced first and will serve to explain the GG cycle, due to its classical layout. Then, the differences with respect to *PROMETHEUS* will be highlighted.

4.1.1 *Vulcain 1*

The main engine case study in this thesis is the retired *Vulcain* 1 (ArianeGroup 1996-2009) since it corresponds to a well-known gas-generator cycle in Europe from which test-campaigns data are available. The approximate steady-state operating data of this *Ariane 5* main stage engine are summarised in Table 4.1.1. For the sake of clarity, the real schematic of this engine is presented

Table 4.1.1: *Vulcain* 1 steady-state operating data

Variable	Value
Chamber mixture ratio	$[5.9, 6]$
Gas-generator mixture ratio	$[0.9, 1]$
Thrust (vacuum)	$1025kN$
Thrust (ground)	$815kN$
Chamber pressure	$100bar$
Gas-generator pressure	$87bar$
Chamber temperature	$3500K$
Gas-generator temperature	$1000K$
Specific impulse (vacuum)	$431.2s$
Turbopumps rotational speed (LOX)	$[11000, 14800]rpm$
Turbopumps rotational speed ($LH2$)	$[28500, 36000]rpm$

in Figure 4.1.1. In that figure the main components of the engine are depicted. Most importantly, it consists in a LOX/LH_2 (liquid oxygen as oxidiser, liquid hydrogen as fuel) engine, which forces the use of two different turbopumps to pump propellants from tanks due to their high density difference. The hot-gas flow necessary to drive turbines comes from a gas generator, which is a small combustion chamber that receives a small portion of propellants main flow. The actuators in this engine are five continuously controllable valves (VCH, VCO, VGH, VGO and VGC), one binary igniter (i_{CC}) and one binary starter (i_{GG}).
H stands for hydrogen, O for oxygen, CC or C for combustion chamber, GG or G for gas generator, V for valve, GC for hot gases, I or i for igniter/starter, L for line, T or turbine, PF for fuel pump and PO for oxidiser pump.
Valves angles (α) control the flows to the main combustion chamber (VCH and VCO), where thrust is produced thanks to the high pressure attained ($100bar$); to the gas generator (VGH, VGO), and to the oxidiser turbine (VGC). The latter consists in the main contribution in determining mixture ratio (MR), the quotient between oxidiser and fuel mass flow rates:

$$MR = \frac{\dot{m}_{ox}}{\dot{m}_{fu}}. \tag{4.1.1}$$

This ratio is defined 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 gas generator (MR_{GG}). Chamber's igniter i_{CC}, considered perfect here, enables combustion in that cavity and gas-generator's starter i_{GG} injects hot gas into that cavity during less than $1.5s$ so as to start driving turbines. This consists in the main contribution to start-up, because once turbines start rotating, pumps can provide more

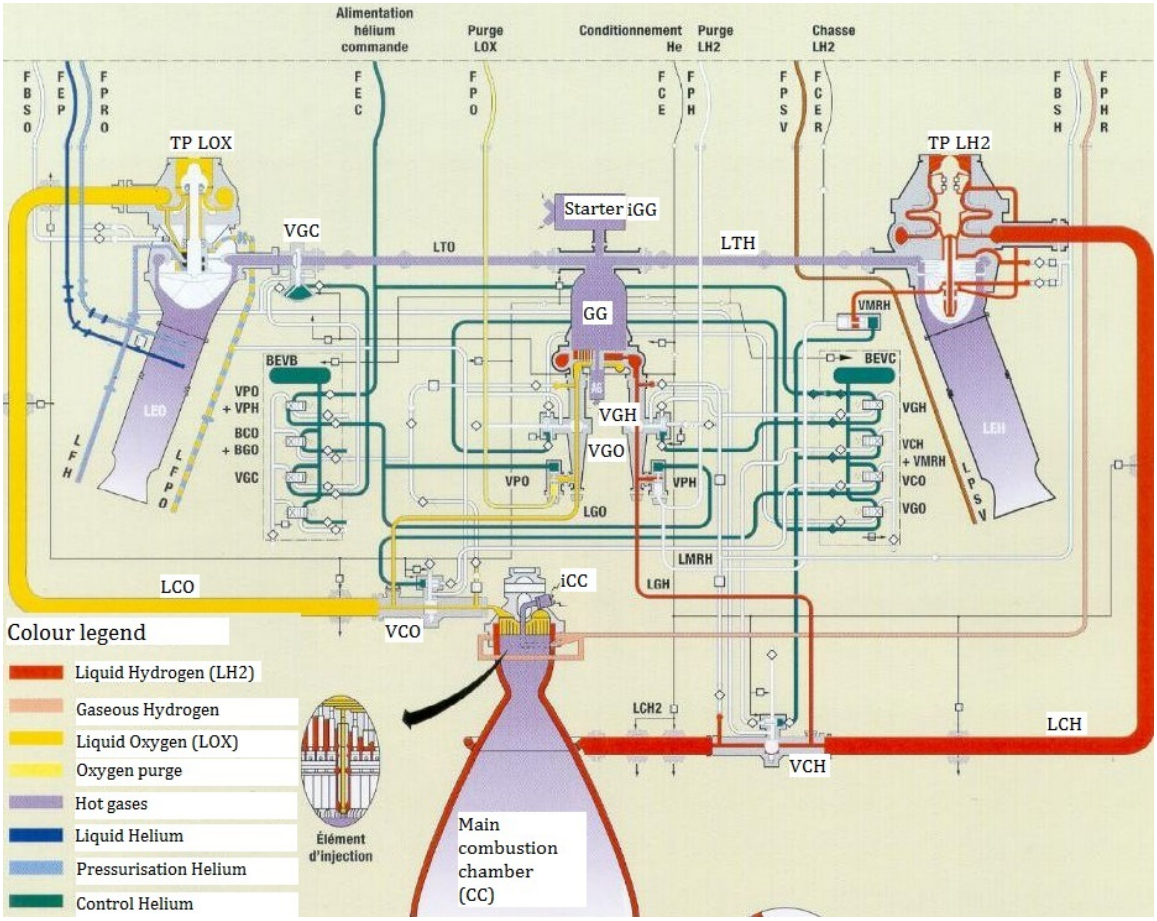


Figure 4.1.1: Schematic of the *Vulcain 1* engine by ArianeGroup and CNES

flow to chambers, which increases combustion pressures and temperatures. This build-up also leads to greater shaft speeds until a steady-state is achieved, at around three to four seconds after the start command.

Tanks contain propellants at assumed constant cryogenic conditions: $3bar$ and $21K$ in the hydrogen case and at $7bar$ and $90K$ in the oxygen one. Not all the elements depicted in Figure 4.1.1 are considered in the model. Only the core system lies within the scope because it represents the highly dominant behaviour of the engine. Hence, all subsystems such as Helium lines are ignored. It seems relevant to summarise the flow paths and elements considered in this thesis:

- Hydrogen line: the hydrogen leaving the tank is absorbed by pump PH, which pumps it to valves through a common pipe LCH. After this pipe, flow is split into the combustion-chamber valve VCH and the gas-generator valve VGH. In VCH, the resistance contribution of the cooling circuit, affecting fuel flow before the main chamber, is taken into account.
- Oxygen line: same path as hydrogen (oxygen tank, PO, LCO, then split into VCO and VGO), but it does not flow through the cooling circuit.
- Hot gases: the mixtures of oxygen and hydrogen are burned at independent ratios (determined by valves) in the two chambers CC and GG. The output flow of CC is discharged into the atmosphere through a converging-diverging nozzle. Nevertheless, the diverging part of the nozzle is not considered here since it does not present a direct impact on chamber pressure,

one of the main variables to control.
GG output is split into two lines, one for each turbine. The path to the hydrogen turbine TH is performed through the pipe LTH, and the path to the oxygen turbine TO crosses the valve VGC. Turbines exhaust is directly emitted to the atmosphere too. These turbines are obviously mechanically connected to their respective pumps by means of shafts.

4.1.2 *PROMETHEUS*

The new *PROMETHEUS* engine, conceived as a low-cost engine, is basically very similar to *Vulcain 1* in its structure (GG cycle) but presents two main differences. The most radical one is the propellant combination for which it is designed. It is LOX/LCH_4, liquid oxygen as oxidiser and liquid methane (M) as fuel, which changes several fuel-related engine parameters. Methane has been deemed attractive for space engines for a long time, but no important developments have been carried out until the last decade. Indeed it presents important life-cycle mission advantages with respect to conventional rocket propellants such as hydrogen or kerosene, as highlighted in the studies of NEILL *et al* [113]. For instance, the LOX/LCH_4 combination is pretty competitive in terms of bulk density impulse even though it presents a lower I_{sp} than LOX/LH_2. Moreover, it allows fuel and oxidiser storage at similar temperatures (also cryogenic), is non-toxic, cheap to produce (even if high purity levels could sensibly increase the production costs) and easy to handle.
In addition, the relatively low density difference between oxygen and methane allows the utilisation of a single turbopump shaft to pump propellants into chambers. Only one turbine is required, which is connected to the oxygen and methane pumps. This is the main structural difference between both considered engines, as it can be seen in Figure 4.1.2, depicting *PROMETHEUS* cycle. This difference simplifies the cycle interconnection from *Vulcain 1*, while the different fuel implies modifications in certain parameters such as tank pressure and temperature and fuel thermodynamic properties. Engine dimensions are also different and the main performance data are listed in Table 4.1.2 [5, 18].

Table 4.1.2: *PROMETHEUS* steady-state operating data

Variable	Value
Chamber mixture ratio	3.4
Gas-generator mixture ratio	$[0.23, 0.26]$
Thrust (vacuum)	$1000kN$
Chamber pressure	$100bar$
Specific impulse (vacuum)	$326s$ (first stage), $345s$ (upper stage)

The absence of a second turbine removes the need for a flow-distribution valve, VGC in *Vulcain 1*. The whole GG output flow is directed to the turbine. Thus, there are four controlled valves, two for the CC and two for the GG: VCM, VCO, VGM and VGO. The rest of the cycle is analogous to *Vulcain 1*, with the exception of the technological solution used in the starter. It produces an equivalent effect to *Vulcain 1*, but in this case it is a rechargeable volume which is filled with cycle's exhaust gases (from the GG/turbine). The moment of GG-combustion ignition $i_{GG} = 1$ and starter activation $i_{sta} = 1$ does

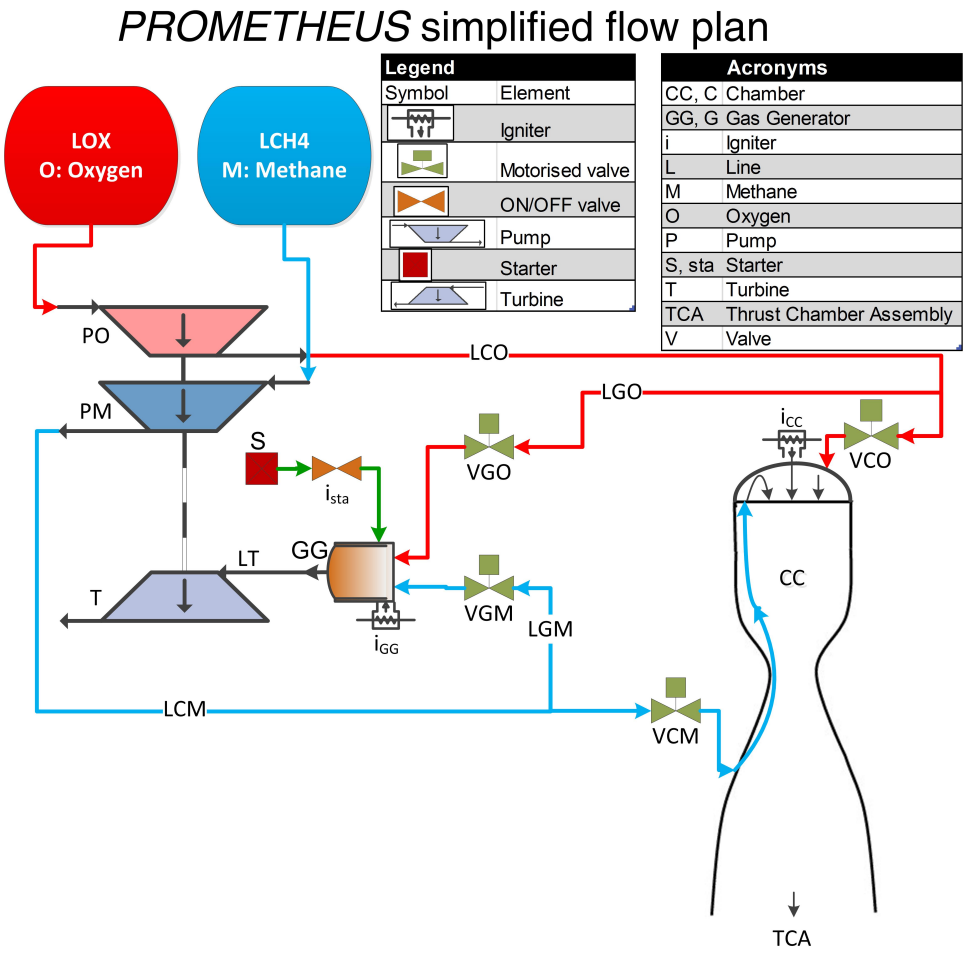


Figure 4.1.2: Schematic of the *PROMETHEUS* engine by ArianeGroup and CNES

not occur simultaneously in this engine, which forces the definition of an additional binary variable i_{sta}.

4.2 Thermo-fluid-dynamic modelling

A simple, dynamic and efficient way of modelling generic LPREs is sought, instead of using more accurate programmes or computations. An easy integration into *Simulink*® is also preferred so as to test different control methods. Along these lines, a new *Simulink*® library of rocket-engine components has been developed to build the simulator in this thesis. It has been named *T-RETM*, Toolbox for Rocket-Engine Transient Modelling. It is slightly based on components from *T-MATS Cantera* [70], which was not completely adequate for this problem due to some limitations. For instance, it is conceived for jet engines, does not present thermodynamic differential equations (only a mechanic ODE on shaft's speed) and requires relatively long computational times. Thus, a new transient 0-D simulation environment for rocket engines has been developed. The modelling approach here is component-wise, but there is a series of common assumptions shared among them.

4.2.1 Library's general aspects

The transmission of information between the interconnected 0-D components lies in a fluid flow vector or set of thermodynamic and chemical variables. Its twelve components are mass flow $\dot{m}$, total conditions (temperature T, pressure p, enthalpy h, density ρ), specific heat ratio γ and chemical composition $comp$. In the following, thermodynamic variables will be referenced to these total quantities even when omitting the adjective. These variables are defined at the input and output of components, since no intra-component spatial contributions are taken into account (0-D modelling), as in most transient modelling approaches introduced in Section 3.1 [81, 90, 96]. In general, gases are considered semi-perfect or thermally perfect, since the ideal-gas law is assumed but their caloric properties are temperature-dependent. Fluid's thermodynamic properties (specific heat at constant pressure C_p $(J/K/kg)$, gas constant R $(J/K/kg)$ and molecular weight M (kg/mol)) are obtained at each stage via weighted addition according to its chemical composition. Moreover, the effect of temperature on C_p is also considered by means of polynomials estimated via least-squares fitting, thanks to data obtained off-line from *Cantera* [70]. A degree of eight was selected as the best compromise for the representativeness of those polynomials. Concerning the use of the ideal-gas state equation

$$p = \rho R T, \tag{4.2.1}$$

it is used for all species at all temperature ranges with the exception of oxygen and methane. These two species present very different densities at their liquid state. Thus, fourth-degree polynomials are considered as a function of temperature if pressure is high enough (beyond the liquid-gas saturation line).

Fluids chemical composition is defined by means of a 1×6 vector, inspired by *Cantera*'s one [70]. It has been opted for considering the LOX/LH_2 and LOX/LCH_4 mixtures so as to be adaptable to the different European GG engines. Thus, composition (mass fraction μ) is defined in the whole library as:

$$comp = \left[\; \mu_{O_2}, \mu_{H_2}, \mu_{H_2O}, \mu_{CH_4}, \mu_{CO_2}, \mu_{N_2} \; \right]. \tag{4.2.2}$$

Other combustion-related species such as CO or radicals are not considered. These considerations serve to define the full flow vector at the outlet of a component once the necessary quantities are determined. For instance, information on two variables within the gas state equation (4.2.1) is needed, as well as on possible composition changes and on the output mass flow.

Transient effects considered and neglected

The following transient effects, which have a great impact on components modelling approaches, are taken into account: reverse flow and compressibility in chambers and turbines. Reverse flow is paramount in some valves like VGC and some pipes like LTH in *Vulcain 1*, serving to distribute the flow among turbines. Compressibility is present in the components where the flow is gaseous and at high temperature, like combustion chambers and turbines. These do not include injection valves, where the flow is considered cold and hence liquid (or supercritical in the hydrogen case). Thus, there is no limitation on valves choked mass flow.

Heat exchange has been studied after adding a simplified cooling circuit relating the main chamber

and its hydrogen input flow. The effect of rising hydrogen's temperature is relevant but the amount of heat extracted from the chamber has a tiny impact on its internal flame temperature, since the goal is to cool the wall.
Other effects which are also neglected are: water-hammer, turbomachines' stall and surge, combustion instabilities, chugging, cavitation and shockwaves, similarly to other simulators [43, 57]. The water-hammer effect does not seem relevant in pump-fed engines, where it is assumed to be damped. Stall and surge are generally dealt with by means of operating constraints. Combustion instabilities, the coupling effect between combustion and resonant acoustic waves, are beyond the scope of this model because their inclusion would highly complexify the modelling approach. Moreover, the frequencies of these phenomena are too high for the actuators to counteract possible instabilities, which can normally be avoided by an appropriate passive-control design [155]. Their active control would consist in an inner loop, which lies outside this thesis's goals. Thus, no combustion delays are explicitly considered either. Chugging physics involves interactions of lower frequency ($< 100Hz$) between different complex phenomena (combustion delays, varying hydraulic impedances, transient thermal flows, etc.) and hence it is also out of scope. Cavitation in liquid lines, which might be relevant in valves and pumps, is neglected for simplification and assumed to be avoided by operational constraints. Shockwaves within lines are also assumed to be avoided by engine's design. Indeed, in this study it is not necessary to model phenomena occurring at frequencies higher than $10Hz$, since actuators in the cycle (valves) would not be able to damp them. Inner loops with a different type of actuators would be required. Thus, no additional high-frequency phenomena are explicitly considered in this model. This does not mean that the natural dynamics captured in the plain thermo-fluid-dynamic equations of components (mentioned in the next paragraph) necessarily present lower frequencies than $10Hz$. They have their natural characteristic times determined by conservation equations. A frequency analysis of models is included in Section 5.4.

Conservation equations

The three main conservation equations used in this model have already been mentioned in the state of the art since they are the common ones for thermofluidic systems: mass (or continuity), momentum and energy conservation. In addition, turbopump shaft dynamics is captured via the mechanical acceleration equation. These basic equations serve to define the ordinary differential equations of the system.

The general mass-conservation equation is:

$$\frac{\partial}{\partial t}\int_V \rho dV + \int_A \rho(\vec{\mathbf{n}}\cdot\vec{\mathbf{v}})dA = 0, \tag{4.2.3}$$

where V is a control volume, A is a control area through which a flow streams, $\vec{\mathbf{n}}$ is the unit vector normal to that area and $\vec{\mathbf{v}}$ is the flow velocity vector. When applying this equation to capacitive elements like constant-volume cavities, a simplified expression describing the mass storage (or density variation) as a function of input and output mass flows can be written as:

$$V\frac{d\rho}{dt} = \dot{m}_{in} - \dot{m}_{out}. \tag{4.2.4}$$

The general momentum-conservation equation comes from the equilibrium of forces in a fluid line:

$$\frac{\partial}{\partial t}\int_V \vec{\mathbf{v}}\rho dV + \int_A \vec{\mathbf{v}}\rho(\vec{\mathbf{n}}\cdot\vec{\mathbf{v}})dA = 0. \tag{4.2.5}$$

Inside resistive elements like valves or pipes this equation can be expressed as [90, 96]:

$$\left(\frac{L}{A}\right)\frac{d\dot{m}}{dt} = p_{in} - p_{out} - k_{res}\dot{m}|\dot{m}|, \tag{4.2.6}$$

where L is length, and k_{res} is the corresponding resistance coefficient. The quotient L/A is the inertia of the element. The last main fluid-dynamic equation concerns energy (e) conservation, whose general form is:

$$\frac{\partial}{\partial t}\int_V e\rho dV + \int_A e\rho(\vec{\mathbf{n}}\cdot\vec{\mathbf{v}})dA = \Phi. \tag{4.2.7}$$

This equation is again applicable to capacitive elements in a simplified differential form [116]:

$$\frac{d}{dt}\left(\frac{pV}{\gamma-1}\right) = (h_{in}\dot{m}_{in} - h_{out}\dot{m}_{out}) + \Phi, \tag{4.2.8}$$

Φ being the heat transferred through the walls (received or sent). Regarding turbopump mechanics, the shaft usually presents a mechanical differential equation on its rotational speed ω:

$$\frac{d\omega}{dt} = \frac{Tq_T - Tq_P}{I_{TP}}, \tag{4.2.9}$$

where Tq_T is turbine's generated torque, Tq_P is pump's consumed torque and I_{TP} is shaft's angular inertia.

4.2.2 Components models formulation

The following paragraphs describe the equations, assumptions and conditions considered in the models of all main components developed in the *T-RETM* library, presented in order of fluid flow in the GG engine: tanks, pumps, pipes, splitters, valves, starter, cooling circuit, combustion chambers, cavities, turbines and shafts. As explained in the previous chapters, components can be classified according to their physical nature into resistive and capacitive ones. Resistive components, such as valves or pipes, are modelled via the momentum-conservation equation (4.2.6), in which a mass flow variation is computed at a given pressure gradient. Capacitive elements, such as cavities or combustion chambers, require the energy-conservation equation (4.2.8) in order to determine their pressure and temperature evolution at some given input and output mass flows. The continuity equation (4.2.4) has to be verified everywhere in the cycle. Depending on the component, different quantities in the *Flow out* vector are modified, following the assumptions made in Section 4.2.1.

Tanks (boundary conditions)

Tanks are considered as a very simple component (Figure 4.2.1), just used to define the input flow vector to the engine as a function of the specified parameters (pump inlet (PI) pressure and temperature and type of propellant) and of the required mass flow, which is the sum of the calculated

mass flows corresponding to line's downstream valves. Hence, they can be regarded as boundary conditions to the system.

- **Inputs**: required mass flow.
- **Outputs**: flow out.
- **Parameters**: pump inlet temperature (T_{PI}, T_PI [K]), pressure (p_{PI}, p_PI [bar]) and fluid composition ($comp_1$ to $comp_6$ as in (4.2.2)).

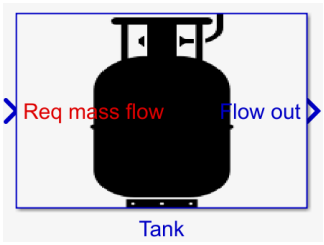


Figure 4.2.1: Tank component

Pumps

Pumps (Figure 4.2.2) modelling presents polynomials for computing output flow pressure and consumed torque, which depend on rotational speed, mass flow and thermodynamic properties.

- **Inputs**:
  - Flow in.
  - N: rotational speed [rpm].
- **Outputs**:
  - Flow out.
  - Tq_P: required torque [Nm] (negative).
- **Parameters**:
  - $CoePrsP = [a_{PrsP}, b_{PrsP}, c_{PrsP}]$: outlet-pressure polynomial coefficients.
  - $CoeCplP = [a_{CplP}, b_{CplP}, c_{CplP}]$: torque polynomial coefficients.
  - $\varphi_{P,lim}$, $PhiP\_lim$: transition mass-flow coefficient, indicating change of coefficients [m^3].
  - ρ_P, $RhoP$: density constant for polynomial [kg/m^3].

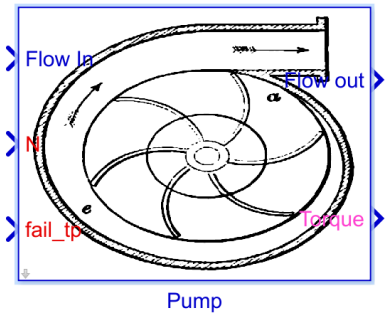


Figure 4.2.2: Pump component

These polynomials consist in parabolic correlations relating pump's self-similar coefficients. Those coefficients, defining pumps maps, are the reduced mass flow φ, pressure-rise ψ and torque C coefficients. They are defined as:

$$\varphi = \frac{\dot{m}_{in}}{\rho_P \omega} \quad [m^3] \tag{4.2.10}$$

$$\psi = \frac{\Delta p_P}{\rho_P \omega^2} \quad [m^2] \tag{4.2.11}$$

$$C = \frac{Tq_P}{\rho_P \omega^2} \quad [m^5]. \tag{4.2.12}$$

And the expressions are of the form:

$$\psi = a_{PrsP}\varphi^2 + b_{PrsP}\varphi + c_{PrsP} \tag{4.2.13}$$

$$C = a_{CplP}\varphi^2 + b_{CplP}\varphi + c_{CplP}, \tag{4.2.14}$$

from which the equation for outlet pressure $p_{P,out}$ can be extracted:

$$p_{P,out} = p_{P,in} + a_{PrsP}\frac{\dot{m}_{in}^2}{\rho_P} + b_{PrsP} \cdot \omega \cdot \dot{m}_{in} + c_{PrsP} \cdot \rho_P \cdot \omega^2, \tag{4.2.15}$$

where ω is rotational speed in rad/s and a_{PrsP}, b_{PrsP} and c_{PrsP} are the parameters provided by the turbopump designer. The associated torque (Tq_P) characteristic curve is:

$$Tq_P = -\left| a_{CplP}\frac{\dot{m}_{in}^2}{\rho_P} + b_{CplP} \cdot \omega \cdot \dot{m}_{in} + c_{CplP} \cdot \rho_P \cdot \omega^2 \right|, \tag{4.2.16}$$

where a_{CplP}, b_{CplP} and c_{CplP} are manufacturer's parameters. Equivalent expressions are proposed in [57, 96, 116]. Depending on the engine, these polynomials can slightly vary. For instance in *PROMETHEUS*, third-order expressions match better the behaviour. Besides, *Vulcain 1* coefficients vary depending on the mass-flow coefficient. Concretely, there are two sets of coefficients for the H pump. The numerical validity of these polynomials at very low rotational speeds, higher than $0.1rpm$, has been verified.

An important assumption made is the consideration that the inertial term of the liquid mass inside the pump is considered in line's overall inertial term, included in valve mass-flow differential equations (defined in 4.2.2). No leaks are factored in.

Output temperature $T_{P,out}$ (K) is computed by considering the balance between the consumed mechanical power and the provided energy to the flow. Seeing that the fluid is still liquid at this stage,

$$T_{P,out} = T_{P,in} + \frac{(1-\eta_P)Tq_P\omega}{C_{v,in}\dot{m}_{in}}, \tag{4.2.17}$$

where pump manometric efficiency η_P (between 0 and 1) can be estimated as [116]

$$\eta_P = \frac{\dot{m}_{in}\Delta p_P}{\rho_P Tq_P \omega}. \tag{4.2.18}$$

However, a constant mean $\eta_P = 0.3$ is used here since the previous expressions led to unphysically too elevated temperatures during the transient. The specific heat at constant volume C_v is used in order to deem the temperature increase as a variation in internal energy, more adapted to liquids than an enthalpy consideration.
All in all, this component serves to increase the pressure of stored propellants and direct them to the chambers injection lines, at the price of a required torque which has to be provided by the downstream turbine.

Pipes

A pipe (Figure 4.2.3) represents a pressure drop due to friction. Some pipes in the cycle, corresponding to pre-chambers lines, are considered static since their transient is irrelevant in the cycle simulation. In those cases, no inertial term is factored in since the whole inertial term of the injection lines is considered in the model of downstream valves. Mass flow is imposed and output pressure is simply computed with the pressure drop equation:

$$p_{out} = p_{in} - k_{res_{pipe}} \cdot \dot{m} \cdot |\dot{m}| \,. \tag{4.2.19}$$

In contrast, post-chambers pipes' transients are relevant to cycle's transient behaviour and hence they are considered dynamic (LTH in *Vulcain 1*). This means that output pressure is imposed and mass flow derivative is computed from the momentum-conservation equation (4.2.6) (also used later in valves), where $Ine_{pipe} = \frac{L_{pipe}}{A_{pipe}}$ is pipe's inertia:

$$\frac{d\dot{m}}{dt} = \frac{(p_{in} - p_{out} - k_{res_{pipe}} \cdot \dot{m} \cdot |\dot{m}|)}{Ine_{pipe}}. \tag{4.2.20}$$

In the library, these two modelling variants can be selected via the parameter *mode*.

- **Inputs**:
  - Flow in.
  - p_{back}: back pressure (outlet), only necessary in *mode* 2 $[bar]$.
- **Outputs**:
  - Flow out.
  - $\frac{d\dot{m}}{dt}$: mass flow derivative $[kg/s^2]$.
- **Parameters**:
  - $k_{res_{pipe}}$, $ResPipe$: resistance $[1/kg/m]$.
  - $k_{hy_{pipe}}$, $KhyPipe$: alternative resistance coefficient, according to the available data (used in *mode* 2) $[m^{-4}]$. To divide by inlet density so as to obtain $k_{res_{pipe}}$.
  - Ine_{pipe}, $InePipe$: fluidic inertia $[m^{-1}]$.
  - *mode*: if 1, output pressure is calculated for an imposed mass flow at input. If 2, output mass flow derivative is calculated as a function of inlet and back pressures.

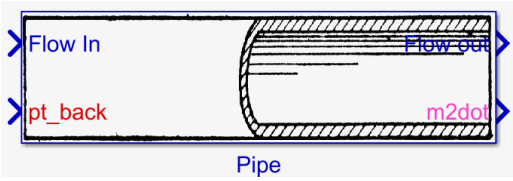


Figure 4.2.3: Pipe component

The forced output pressure in *mode* 2 comes from an additional non-reactive cavity (LTH cavity), which has to be placed before the hydrogen turbine for causality purposes. Basically, there has to be a capacitive element between resistive ones in order to obtain a causal system of equations.

Splitter

The splitter component (Figure 4.2.4) is a fairly simple one, allowing to divide the flow into two outputs.

- **Inputs**:
  - Flow in.
  - $\dot{m}_{sec}$: secondary mass flow [kg/s].
  - $\dot{m}_{pri}$: primary mass flow [kg/s].
- **Outputs**:
  - Secondary flow out.
  - Primary flow out.

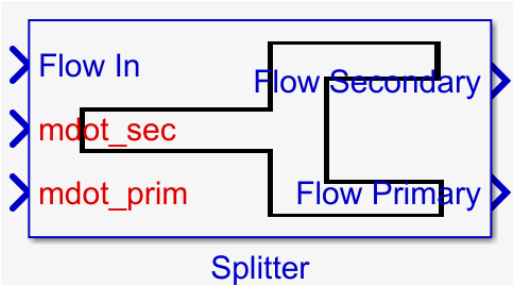


Figure 4.2.4: Splitter component

According to the provided mass flows indicating the new distribution, outlet flows are redefined with these new values while conserving temperature and pressure.

Injection valves

Valve components (Figure 4.2.5) are dynamic resistive elements and hence present a differential equation on their mass flow (4.2.6). For graphical simplification, they contain chamber injectors too, since they are the last elements before chambers.

- **Inputs**:
  - Flow in.
  - α_V, $ValveAngle$: angle [0 °closed, 90 °open].
  - $p_{back,c}$: back pressure, coming from combustion chambers [bar].
- **Outputs**:
  - Flow out.
  - $\frac{d\dot{m}}{dt}$: mass flow derivative [kg/s^2].
- **Parameters**:
  - L_V, $LngV$: length of line, for inertia consideration [m].
  - $EnvAngV$: valve's angles for interpolation [°].
  - $EnvKhyV$: resistance-coefficient correlation to angles [m^{-4}].
  - ρ_V, $RhoV$: inlet density constant for correlation [kg/m^3].
  - $k_{res_{inj}}$, $ResI$: downstream injector resistance [$1/kg/m$].
  - $[k_{res_{CR}}, Ine_{CR}]$, $ResCR\_IneCR$: cooling circuit resistance and inertia (in vector) ([$1/kg/m$],[m^{-1}]).

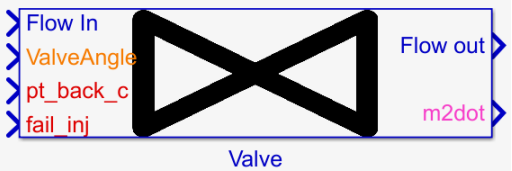


Figure 4.2.5: Valve component

An interpolation is performed so as to obtain a correlated value of valve's resistance coefficient k_{hy_V} according to the opening angle, as furnished by the manufacturer. Then, valve's area A_V and resistance k_{res_V} and line's overall inertia Ine_V can be computed. For obtaining A_V, the expression of liquid pressure drop is used, (the pressure-drop coefficient ζ is considered equal to one in the correlation):

$$k_{res_V} = \frac{k_{hy_V}}{\rho_V} + k_{res_{CR}} \tag{4.2.21}$$

$$k_{hy_V} = \frac{\zeta}{2A_V^2} \rightarrow A_V = \frac{1}{\sqrt{2k_{hy_V}}} \tag{4.2.22}$$

$$Ine_V = \frac{L_V}{A_V} + Ine_{CR}. \tag{4.2.23}$$

In the case of the main chamber fuel valve (VCH in *Vulcain 1*), it also presents the resistance and inertia of the cooling circuit through which fuel must flow ($k_{res_{CR}}$, Ine_{CR}). Indeed, fuels have a greater heating value than oxidisers. That is the reason why fuel is used in the regenerative cooling circuit to reduce the chamber wall temperature.
Injectors are taken into account so as to consider the real back pressure, which is slightly higher than the downstream chamber pressure (input value to the component). Indeed, it is necessary that

injectors produce a pressure drop so as to avoid reverse flow from the chamber into the injection lines, which is captured in the following equation:

$$p_{back} = p_{back,c} + k_{res_{inj}} \cdot \dot{m} \cdot |\dot{m}| \,. \tag{4.2.24}$$

Then, the mass flow differential equation can be obtained from the momentum conservation equation (4.2.6):

$$\frac{d\dot{m}}{dt} = \frac{(p_{in} - p_{back} - k_{res_V} \cdot \dot{m} \cdot |\dot{m}|)}{Ine_V}. \tag{4.2.25}$$

Reverse flow (with negative sign) is not allowed in these injection valves models since it is avoided by design in the real engine. Component's derivative output is set to zero if mass flow diminishes to zero. This modelling approach neglects the capacitive effect of chamber injection domes and cooling-circuit manifolds due to their low volume (fast dynamics).
This valve model only concerns its fluid side, that is to say, the valve behaviour in terms of fluid mechanics. Its mechanical behaviour, determined by its internal actuator, consists in a model block external to the engine simulator. That sub-system, representing the actuator dynamics related to the tracking of a commanded opening angle, is modelled as a simple second-order system. It feeds the valve fluid-model block with the predicted angle at each instant of time. More details on this connection can be found in Section 6.1.

Alternative butterfly valves

Butterfly valves (Figure 4.2.6) present basically the same dynamic formulation. There are only some differences in their resistance correlation to angle. In *Vulcain 1*, the VGC (hot gas valve) is of this kind and is used to establish the flow balance between turbines, and hence to tune the global mixture ratio. The flow directed to the oxidiser turbine is the one flowing through this valve. In order to avoid iterative loops and to ensure causality, an additional cavity has to be added downstream, whose pressure is considered as valve's outlet pressure.

- **Inputs**:
  - Flow in.
  - α_V, $ValveAngle$: angle [$0°$closed, $90°$open].
  - p_{back}: back pressure, coming from downstream cavity [bar].
- **Outputs**:
  - Flow out.
  - $\frac{d\dot{m}}{dt}$: mass flow derivative [kg/s].
- **Parameters**:
  - a_i, a_coeff: coefficients for correlation between effective cross section and opening angle.
  - k_i, k_coeff: valve's coefficients for correlation between pressure-drop coefficient and opening angle.
  - L_V, $LngV$: length of line, for inertia consideration [m].

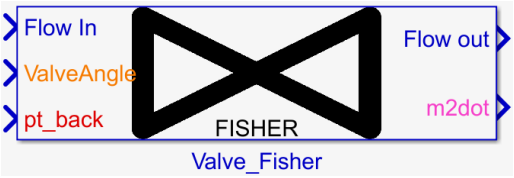


Figure 4.2.6: Butterfly valve component (*Fisher* VGC in *Vulcain 1*)

Valve section is defined as a polynomial function of the opening angle, according to the manufacturer:

$$A_V = (a_0 + \alpha_V^2(a_2 + \alpha_V^2(a_4 + \alpha_V^2 a_6))) \times 10^{-5}. \tag{4.2.26}$$

In these valves, an estimated expression of the pressure-loss coefficient ζ with respect to valve's angle is available [116]:

$$\zeta = (k_0 + k_1\alpha_V + k_2\alpha_V^2 + k_3\alpha_V^3) \times 10^4, \tag{4.2.27}$$

from which the resistance term can be computed:

$$k_{res} = \frac{\zeta}{2\rho_V A_V^2}. \tag{4.2.28}$$

Finally, mass-flow derivative is obtained analogously to (4.2.25), with the respective inertia. In contrast, reverse flow is allowed in this valve because in the cycle it serves to distribute the flow to turbines.

Starter

This simple yet vital component computes the GG starter output flow properties according to the current time instant within the start-up transient (Figure 4.2.7). Injected mass flow is a simple reproducible function of time since activation; and output temperature T (K), C_p and R are supposed constant. In *Vulcain 1*, the moment of GG-combustion ignition $i_{GG} = 1$ and starter activation is assumed the same. However, in *PROMETHEUS*, those two events occur at slightly different time instants.

- **Inputs**:
  - I_ON: combustion ignition boolean, coming from control inputs.
- **Outputs**:
  - $ignit$: ignition boolean, to transmit to chambers.
  - $starter\_chars$: starter properties array $[\dot{m}, T, C_p, R]_{sta}$.
- **Parameters**:
  - $EnvTim$: time instants for mass-flow correlation $[s]$.
  - $EnvQms$: mass-flow correlation to time $[kg/s]$.
  - Tmp: temperature $[K]$.
  - C_p: specific heat ratio at constant pressure $[J/kg/K]$.
  - Rgs: gas constant (R) $[J/kg/K]$.

Figure 4.2.7: Starter component

Cooling circuit (for GG cycles)

The cooling circuit (CR) is a simplified static component containing algebraic equations on heat transfer (Figure 4.2.8). It does not present a relevant impact on the global steady-state results in pressures and mixture ratios (presented in Section 4.3) but it alters slightly the shape of transient curves. No thermal dynamic equations have been included due to their assumed faster speed in comparison to fluidic time scales in GG cycles, which are captured in the considered conservation equations. Indeed, the resistive and inertial influence in the fluid-dynamic sense of the cooling line have been taken into account in the corresponding CC fuel valve.

- **Inputs**:
  - Flow in (fuel injection).
  - T_{hot}: hot temperature coming from combustion chamber, with unit delay [K].
  - $\dot{m}_{ox,CC}$: oxidiser mass flow injected into chamber [kg/s].
- **Outputs**:
  - Flow out.
  - q_{cool}: evacuated heat from chamber by unit of surface [W/m^2].
- **Parameters**:
  - $k_{h_{hot}}$: constant ratio between hot-side heat-transfer coefficient and total injected mass flow into chamber [$Ws/(m^2Kkg)$].
  - $k_{h_{cold}}$: constant ratio between cold-side heat-transfer coefficient and injected fuel mass flow into chamber [$Ws/(m^2Kkg)$].
  - t_{wall}: chamber wall thickness [m].
  - k_{wall}: wall thermal conductivity [$W/m/K$].

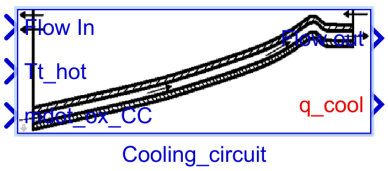


Figure 4.2.8: Cooling-circuit component

First of all, heat transfer coefficients from both hot gas and cold gas sides (h_{hot} and h_{cold}) are computed thanks to their ratios to mass flows, provided as parameters.

$$h_{hot} = k_{h_{hot}}(\dot{m}_{fu,CC} + \dot{m}_{ox,CC}), \tag{4.2.29}$$

$$h_{cold} = k_{h_{cold}}\dot{m}_{fu,CC}. \tag{4.2.30}$$

Wall temperature $T_{wall,hot}$ is saturated to $1000K$, which consists in the maximum desired temperature on the material:

$$T_{wall,hot} = min(T_{hot}, 1000). \tag{4.2.31}$$

This consists in a conservative hypothesis, in which the thermal dynamics is considered infinitely fast. The wall would acquire the hot gas temperature instantaneously and it is supposed not to overcome the $1000K$ by CR layout. And then, the heat evacuated from the chamber to the cooling circuit is computed, assuming forced convection:

$$q_{cool} = h_{hot}(T_{hot} - T_{wall,hot}). \tag{4.2.32}$$

Next, the global thermal resistance H of the three thermal elements in this problem (hot gas side, wall and cold gas side) is calculated. This permits the computation of fuel temperature when leaving the circuit (T_{cold}), since q_{cool} is overall the same, assuming similar exchange surfaces.

$$H = \frac{1}{\frac{1}{h_{hot}} + \frac{t_{wall}}{k_{wall}} + \frac{1}{h_{cold}}} \tag{4.2.33}$$

$$T_{cold} = T_{hot} - \frac{q_{cool}}{H}, \tag{4.2.34}$$

which slightly influences chamber's physics downstream. That is the reason why there is the need for a unit delay in the hot temperature input so as to avoid the occurrence of an algebraic loop in the simulator.
This simplified modelling approach is acceptable in a GG cycle, where only the injected fuel temperature plays a role in the system. Nevertheless, in expander engines, a dynamic model would be required since the outlet CR temperature determines the enthalpy available to turbines, as explained in Perspectives.

Combustion chambers

Not only the main CC but also the GG chamber are modelled via this dynamic component in which simplified combustion is simulated.

- **Inputs**:
  - Fuel flow in.
  - Oxidiser flow in.
  - $\dot{m}_{out}$: required output flow (fed from downstream in the case of GG, internally computed in CC) [kg/s].

  - $ignit$: ignition vector, necessarily including an ignition boolean in first place, and optionally appended starter properties ($[\dot{m}, T, C_p, R]_{sta}$) if a starter is connected to the chamber (as in GG).
  - p_c: integrated chamber pressure [bar].
- **Outputs**:
  - Flow out.
  - $\dot{p}$: chamber pressure derivative [bar/s].
  - T_c: chamber temperature [K].
  - MR, r: chamber gas mixture ratio.
- **Parameters**:
  - A_{th}: throat area [m^2].
  - V: volume [m^3].
  - η_c: combustion efficiency.

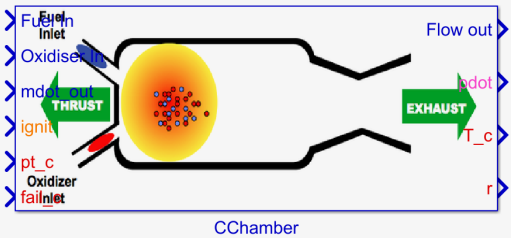


Figure 4.2.9: Combustion chamber component

Indeed, combustion chambers (Figure 4.2.9) contain a sub-component, the cavity, which may be reactive (combustion possible) or not. In the combustion chamber component, a reactive cavity with a given combustion efficiency is considered. The vaporisation and atomisation of injected fluids (liquids) are assumed perfect and without delay.

Cavities

This dynamic capacitive component (Figure 4.2.10) is sometimes necessary between resistive components (valves, turbomachines, pipes) to render global equations causal-implicit, since it captures the mass and energy filling of volumes. In the *Vulcain 1* model, apart from the reactive CC and GG cavities, two non-reactive cavities are included before turbines, after the LTH pipe and after the VGC valve. Gas pressure and density (and hence temperature) are assumed spatially uniform in the cavity, which already satisfies the conservation of momentum. Thus, that equation (4.2.6) is not used in this capacitive element.

- **Inputs**:
  - Fuel flow in (primary flow if only one input).
  - Oxidiser flow in (zero if only one input).
  - $\dot{m}_{out}$: required output flow (fed from downstream in the case of GG, internally computed in CC and non-reactive cavities) [kg/s].

  - $ignit$: ignition vector, necessarily including an ignition boolean in first place, and optionally appended starter properties ($[\dot{m}, T, C_p, R]_{sta}$) if a starter is connected to the chamber (as in GG).
  - p_c: integrated pressure [bar].
  - ρ_c: integrated density [kg/m^3].
- **Outputs**:
  - Flow out.
  - $\dot{p}$: pressure derivative [bar/s].
  - T_c: chamber temperature [K].
  - MR, r: chamber gas mixture ratio.
  - $\dot{\rho}$: density derivative [$kg/m^3/s$].
- **Parameters**:
  - $react\_flag$: 1 if reactive.
  - A_{th}: throat area (outlet) [m^2].
  - V: volume [m^3].
  - η_c: combustion efficiency (1.0 in non-reactive cavities).

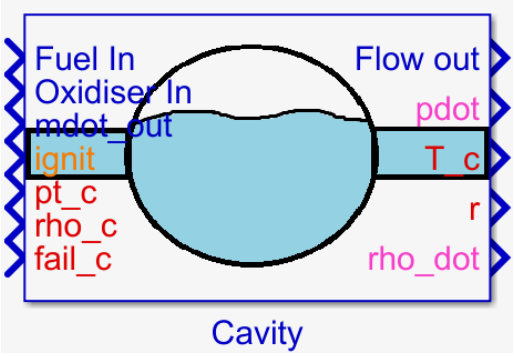


Figure 4.2.10: Cavity component

First of all, injected fuel and oxidiser (if present) are merged in a simple way through a mass-flow-weighted addition. Injected mass flow $\dot{m}_{inj}$ and mixture ratio MR are defined as functions of those injected species:

$$\dot{m}_{inj} = \dot{m}_{fu} + \dot{m}_{ox} \tag{4.2.35}$$

$$MR = \frac{\dot{m}_{ox}}{\dot{m}_{fu}}. \tag{4.2.36}$$

In the reactive case, the consumed and produced mass fractions of each species (μ_i) are computed statically as a function of mixture ratio, supposing that they stay constant at a given mixture ratio. This consists in a simplification of combustion kinetics with respect to [20, 96, 116], since it is considered here that the ARRHENIUS equations [153] do not provide a worthwhile precision augmentation while highly increasing complexity. In this proposed model, the excess fuel or oxidiser, present at non-stoichiometric mixture ratios, is taken into account so as to determine the outlet composition

($MR_{st} = \frac{\dot{m}_{ox,burned}}{\dot{m}_{fu,burned}}$ is the stoichiometric mixture ratio). The mass fractions expressions for the fuel and oxidiser species which are presented in the following lines correspond to the burned fractions of injected flow (μ_{burned}) and to the remaining fractions after combustion (μ_{out}).
If $MR \leq MR_{st}$, there is unburned fuel while all the oxidiser is assumed to be consumed:

$$\dot{m}_{fu,burned} = \frac{\dot{m}_{ox}}{MR_{st}} = \frac{\frac{MR}{MR+1}\dot{m}_{inj}}{MR_{st}} \rightarrow \mu_{fu,burned} = \frac{MR}{(MR+1)MR_{st}}, \tag{4.2.37}$$

$$\mu_{ox,burned} = \frac{MR}{MR+1}, \tag{4.2.38}$$

$$\dot{m}_{fu,out} = \dot{m}_{fu,inj} - \frac{\dot{m}_{ox}}{MR_{st}} = \left(\frac{1}{MR+1} - \frac{\frac{MR}{MR+1}}{MR_{st}}\right)\dot{m}_{inj} \rightarrow \mu_{fu,out} = \frac{1 - \frac{MR}{MR_{st}}}{MR+1}, \tag{4.2.39}$$

$$\mu_{ox,out} = 0. \tag{4.2.40}$$

Otherwise, if $MR > MR_{st}$, there is unburned oxidiser and the whole fuel injection is supposed to be consumed:

$$\mu_{fu,burned} = \frac{1}{MR+1}, \tag{4.2.41}$$

$$\dot{m}_{ox,burned} = MR_{st}\dot{m}_{fu} = MR_{st}\frac{1}{MR+1}\dot{m}_{inj} \rightarrow \mu_{ox,burned} = \frac{MR_{st}}{MR+1}, \tag{4.2.42}$$

$$\mu_{fu,out} = 0, \tag{4.2.43}$$

$$\dot{m}_{ox,out} = \dot{m}_{ox,inj} - MR_{st}\dot{m}_{fu} = \left(\frac{MR}{MR+1} - \frac{MR_{st}}{MR+1}\right)\dot{m}_{inj} \rightarrow \mu_{ox,out} = \frac{MR - MR_{st}}{MR+1}. \tag{4.2.44}$$

Combustion products mass fractions $\mu_{p,i}$ are computed according to the propellant combination, which can present different reactions. In the *Vulcain 1* case, LOX/LH_2 only presents water vapour as a product. Hence, it will represent 100% of burned reactants. In LOX/LCH_4 engines, such as *PROMETHEUS*, there is a proportion between water vapour and carbon dioxide as products. If ν represents the number of moles in the stoichiometric reaction and M molar masses, individual products mass fractions are:

$$\mu_{p,i} = \frac{\nu_{p,i}M_{p,i}}{\sum_i \nu_{p,i}M_{p,i}}(\mu_{fu,burned} + \mu_{ox,burned}), \tag{4.2.45}$$

after multiplying by the burned mass fraction, the portion of the flow converted into products.
The influence of the pyrotechnic starter on GG flow properties is expressed via simple weighted additions, serving to update GG's inlet C_p and R (μ_{sta} as starter's mass flow fraction):

$$\dot{m}_{in} = \dot{m}_{inj} + \dot{m}_{sta}, \tag{4.2.46}$$

$$\mu_{sta} = \frac{\dot{m}_{sta}}{\dot{m}_{in}}, \tag{4.2.47}$$

$$C_{p,c} = (1 - \mu_{sta})C_{p,in} + \mu_{sta}C_{p,sta}, \tag{4.2.48}$$

$$R_c = (1 - \mu_{sta})R_{in} + \mu_{sta}R_{sta}. \tag{4.2.49}$$

Then, cavity temperature T_c (of combustion or not) is computed by means of the gas equation of state, which depends on the provided pressure and density (coming from integration). It is corrected with

the starter influence if present. As a simplification, combustion efficiency η_c is considered in the gas equation too ($\eta_c = 1$ in non-reactive cavities):

$$T_{c,1} = \frac{p_c}{R_c \rho_c} \eta_c, \tag{4.2.50}$$

$$T_c = ((1 - \mu_{sta}) C_{p,in} T_{c,1} + \mu_{sta} C_{p,sta} T_{sta}) / C_{p,c}. \tag{4.2.51}$$

Efficiency serves here as an empirical tool to represent the thermal losses associated to dissociations, diffusion and three-dimensional effects (among others), which are not modelled and which would reduce temperature in reality. An alternative simplification option could have been correcting C_p first, which is dependent on temperature.

By knowing cavity's pressure (from input port) and temperature (from 4.2.50), the rest of thermodynamic and chemical components of the output flow vector can be updated.

The determination of output mass flow depends on the type of cavity. In the GG, mass flow through the outlet throat is taken from the input port $\dot{m}_{out}$. This input provides the sum of the integrated mass flows downstream (through VGC and LTH in *Vulcain 1*, LT in *PROMETHEUS*). However, it is saturated to the maximum choked flow [130] since it consists of hot gas.

$$\begin{aligned} C^* &= \sqrt{\frac{R_{out} T_c}{\gamma_{out}} \left(\frac{2}{\gamma_{out} + 1} \right)^{-\frac{\gamma_{out}+1}{\gamma_{out}-1}}}, \\ \dot{m}_{choked} &= \frac{p_c A_{th}}{C^*}, \\ \dot{m}_{out} &= min\left(\dot{m}_{out}, \dot{m}_{choked} \right), \end{aligned} \tag{4.2.52}$$

C^* being chamber's characteristic speed and A_{th} the throat area. In the CC and pre-turbine cavities the output flow is not imposed. Since outlet static pressure is initially the ambient one (p_{amb}), mass flow is computed subsonically until the throat chokes. In other words, the computation is subsonic until cavity's total pressure increases enough so as to obtain critical conditions at the throat. In the subsonic phase, the output mass flow is calculated as [130]:

$$\dot{m}_{out} = p_c A_{th} \sqrt{\frac{2\gamma_{out}}{(\gamma_{out} - 1) R_{out} T_c} \left(\left(\frac{p_{amb}}{p_c} \right)^{\frac{2}{\gamma_{out}}} - \left(\frac{p_{amb}}{p_c} \right)^{\frac{\gamma_{out}+1}{\gamma_{out}}} \right)}. \tag{4.2.53}$$

Otherwise, the choked equation (4.2.52) is employed.

With this outlet mass flow, and assuming that the gas volume V_g is equal to cavity's one (a more complex approach would be the use of a gas volume differential equation [96], but this effect is negligible), the mass-conservation equation (4.2.4) can be set:

$$\frac{d\rho_c}{dt} = \frac{\dot{m}_{in} - \dot{m}_{out}}{V_g}. \tag{4.2.54}$$

In order to obtain all the terms of the energy-conservation equation, the most characteristic one of this component, further computations have to be performed. As said before, species' production and consumption rates are assumed constant at a given mixture ratio (zero in non-reactive cavities). In the energy equation adapted to reactive cavities (4.2.57), there is a term representing mass variation

rates of each species per unit of volume ($\dot{w}_i$). This is expressed for each reacting (r) and produced (p) species as:

$$\dot{w}_{r,i} = -\frac{\mu_{burned,i}\dot{m}_{inj}}{V_g}, \tag{4.2.55}$$

$$\dot{w}_{p,i} = \frac{\mu_{out,i}\dot{m}_{inj}}{V_g}. \tag{4.2.56}$$

The energy equation expressed for cavity pressure is the following, considering additional combustion terms, one inlet (propellant mixing is neglected) and one outlet, and neglecting cooling and volume and efficiency variations [116]. The subscript i concerns all species while j only concerns the injected ones.

$$\begin{aligned}\frac{dp_c}{dt} = {} & \frac{(\gamma_{out}-1)}{V_g}\left(\frac{\gamma_{in}p_c}{(\gamma_{in}-1)\rho_{in}} + \frac{C_{v,in}p_c(\gamma_{in}-\gamma_{out})}{C_{v,out}\rho_c(\gamma_{out}-1)^2}\right)\dot{m}_{in} - \frac{p_c\gamma_{out}\eta_c}{V_g\rho_c}\dot{m}_{out} \\ & - ignit\left((\gamma_{out}-1)\eta_c\sum_i(\dot{w}_i h_{f,i}) - \frac{\gamma_{out}p_c}{(\gamma_{out}-1)\rho_c}\sum_i\left(\left(\frac{C_{p,i}}{C_{p,out}} - \frac{C_{v,i}}{C_{v,out}}\right)\dot{w}_i\right)\right. \\ & \left. + \frac{\gamma_{out}-1}{V_g}\sum_j(C_{p,out}T_{vap,j} - (L_{v,j} + C_{p,j}(T_{vap,j} - T_{in,j})))\dot{m}_{in,j}\right),\end{aligned} \tag{4.2.57}$$

where $h_{f,i}$ is the formation enthalpy of each species (J/kg), $T_{vap,j}$ is the vaporisation or boiling temperature of each species and $L_{v,j}$ (J/kg) is the vaporisation heat of each species (considered positive since endothermic). The first two terms, multiplying input and output flow respectively, correspond to the effect of introducing and removing flow from the cavity. Vaporisation heat influence (last term in (4.2.57)) has been observed to be relevant in simulations. The terms multiplied by $ignit$ are related to combustion and are only activated when the igniter is operative in reactive cavities. This is a modelling choice which is further justified in Chapter 5, since it is in relation with the control-oriented models developed there and the manner in which the effect of control inputs is considered. Another way to consider the effect of ignition in combustion consists in defining mixture-ratio and mass-fraction thresholds from which combustion can take place [43, 57].

Turbines

Turbines (Figure 4.2.11) are represented by a supersonic model valid above some given rotational speed and pressure ratio, similar to [57]. In GG cycles, supersonic turbines are often used since they allow to minimise the GG consumption, which increases performance. The enthalpy to drive the turbine is mainly obtained via a maximisation of the expansion factor rather than an increase in flow rate. This is precisely accomplished by reducing GG outlet section, and hence output flow.

- **Inputs**:
  - Flow in.
  - N: rotational speed [rpm].
  - $T_{T,out}$: output temperature from previous step [K].
- **Outputs**:

  - $T_{T,out}$: output temperature, which is fed back to inputs [K].
  - Tq_T: generated torque [Nm] (positive).
- **Parameters**:
  - $CoeCplT$: torque polynomial coefficients.
  - Rad_T: radius [m].
  - N_{min}: minimum rotational speed for polynomial validity [rpm].
  - $A_{T,out}$, $AthLE$: turbine outlet area [m^2].

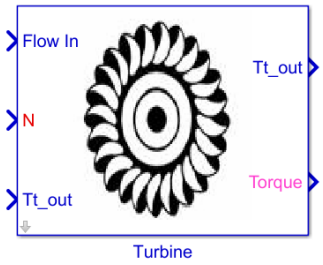


Figure 4.2.11: Turbine component

First, the reduced rotational speed N_R (non-dimensional, equivalent to a Mach number) is computed:

$$N_R = \frac{\omega Rad_T}{\sqrt{\gamma R T_{T,in}}}, \tag{4.2.58}$$

Rad_T being turbine's radius (m). Temperature is taken from the previous integration step by means of a unit delay, in order to avoid an iterative loop. With this temperature, outlet pressure $p_{T,out}$ can be computed by supposing a choked outlet nozzle [130], as in (4.2.52). Then, work W is calculated with the pressure ratio π_T (defined greater or equal than 1), all assuming that γ does not vary much from input to output [57]:

$$W = \dot{m}\frac{\sqrt{\gamma R T_{T,in}}}{\gamma - 1}(1 - \pi_T^{\frac{1-\gamma}{\gamma}})Rad_T. \tag{4.2.59}$$

Next, a regression model with eight coefficients ($a_{1,T}$ to $a_{8,T}$) and an auxiliary correlating variable ($Corr$) is applied to obtain the specific torque ST and then efficiency of a particular turbine:

$$Corr_{Vulcain\ 1} = \frac{0.4}{1 - \pi_T^{-0.28571}}, \quad Corr_{PROMETHEUS} = 1, \tag{4.2.60}$$

$$\begin{aligned} ST(N_R, \pi_T) = (a_{1,T} + N_R(a_{2,T} + N_R a_{3,T}) + \pi_T(a_{4,T} + \pi_T a_{5,T}) + a_{6,T}\pi_T N_R + a_{7,T}\ln(\pi_T) \\ + a_{8,T}\ln(N_R)) \cdot Corr. \end{aligned} \tag{4.2.61}$$

The previous correlation is not valid for too low pressure ratios or reduced rotational speeds. Empirical results point to a minimum ratio $\pi_{T,min}$ of 5 in *Vulcain 1* for using (4.2.61). Hence, for lower values a linear extrapolation is performed from $\pi_T = 1$ (by definition) and zero specific torque, as in [116].

$$ST = ST(N_R, \pi_{T,min})\frac{\pi_T - 1}{\pi_{T,min} - 1}. \tag{4.2.62}$$

Outlet temperature is easily obtained by the typical gas-turbine equation after having computed turbine's efficiency η_T:

$$\eta_T = ST \cdot N_R, \tag{4.2.63}$$

$$T_{T,out} = T_{T,in}(1 - \eta_T(1 - \pi_T^{\frac{1-\gamma}{\gamma}})). \tag{4.2.64}$$

Finally, generated torque Tq_T is calculated [57]:

$$Tq_T = ST \cdot W. \tag{4.2.65}$$

Shafts

The shaft (Figure 4.2.12) is a simple component containing the mechanical differential equation (4.2.9) on its rotational speed ω, which represents the torque balance between the pump and the turbine. No friction losses are considered. The shaft is only allowed to turn in one direction, only positive ω are allowed within the turbomachine components.

- **Inputs**:
  - Torque array (pump and turbine torques in the same vector, with sign) [Nm].
- **Outputs**:
  - $\dot{N}$: rotational acceleration [rpm/s].
- **Parameters**:
  - I_{TP}, $JinTP$: shaft's inertia [$kg \cdot m^2$].

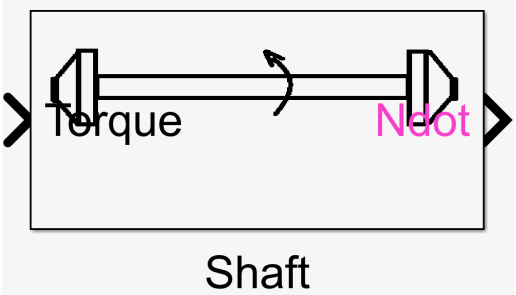


Figure 4.2.12: Shaft component

4.2.3 Cycle interconnection and simulation conditions

The structure of the GG cycle has been explained in Section 4.1. The previously described components of the developed *T-RETM* library have been connected following that layout, from which a simplified schematic has been depicted in Figure 2.6.1 for *Vulcain 1* and in Figure 4.1.2 for *PROMETHEUS*. The following explanations apply to the GG cycle, and hence to both engines. The resulting *Simulink*® diagram presents several layers. The inner one contains the flow structure, illustrated in Figure 4.2.13 for *Vulcain 1*. One of the most remarkable aspects of this structure is

the interdependency between upstream and downstream flows, very characteristic of these cyclic thermodynamic systems. The state of the flow downstream affects the upstream flow in most parts of the cycle. Choked throats are obviously not dependent on the downstream state, such as the CC and pre-turbine cavities outlets. However, the rest of lines in the cycle present that interdependency. This high degree of coupling is well captured in the calculation paradigm here. So as to understand the coupling logic, one can cut the cycle and extract the dependency chain. One can take into account first the pressure in the main cavities CC and GG, in order to define the evolution of mass flows through injection valves (also depending on the pumped-propellants pressure). An increase in pressure in the GG also triggers an increase in mass flow and pressure in its downstream line, which is connected to turbines. This leads to a higher shaft rotational speed, which serves to pump stored propellants at a higher pressure. The higher pumped pressure serves to augment the injected flow, which at the same time raises the pressure in chambers. This explains why the valve and dynamic pipe components require the back pressure as an input piece of data, or why the GG needs to have downstream information on its output flow. The mass flow extracted from tanks and pumped into chambers is the same as the one computed from the momentum-conservation equation in injection valves. The mechanical connection provided by the shaft also joins both ends of the cycle.

As explained throughout the components models, a set of differential equations is defined. Integrated variables or states comprise rotational speeds ω, pressures p, densities ρ and mass flows $\dot{m}$. Cavities, valves, certain pipes and shafts provide derivative values to the outer layer, where integration is performed. Only the density integration is carried out in the inner layer since it is not considered as a state in the subsequently derived state-space model, introduced in Chapter 5. The integration allows indeed to avoid the algebraic loops in the simulation, by providing integrated variables to components via a highly-coupled system of ODEs. Suitable methods to solve the integration are either with an automatic variable step and *ode23* or *ode45* schemes, or with a fixed step and *ode1* to *ode4*. In either ways a maximum time step of $10^{-5}s$ is required for obtaining accurate results.
Apart from considering the tank boundary conditions mentioned in Section 4.1, initial conditions for cavities pressures are at sea-level, as the start on ground is considered. Mass flows start very close to zero but not zero (10^{-15}), required for avoiding numerical initialisation issues. Due to similar reasons, shaft speeds must start at $2rpm$ for hydrogen $1.3rpm$ for oxygen, what already establishes the ratio between shafts in *Vulcain 1*. The same order of magnitude is used in *PROMETHEUS* turbopump initial condition. Initial conditions for the flow vector at fluid lines correspond to the presence of purging nitrogen after the chill-down phase ($101.325kPa$ and $20K$).
The component that serves to shift the system from this initial equilibrium to the nominal thrust is the starter. It consists in an external element which offers an initial stimulus to GG pressure during the start-up transient phase. Since the GG is the core of the cycle, the system evolves accordingly due to its coupling until attaining the steady state. The start-up phase also presents other events, explained in the next section 4.3.

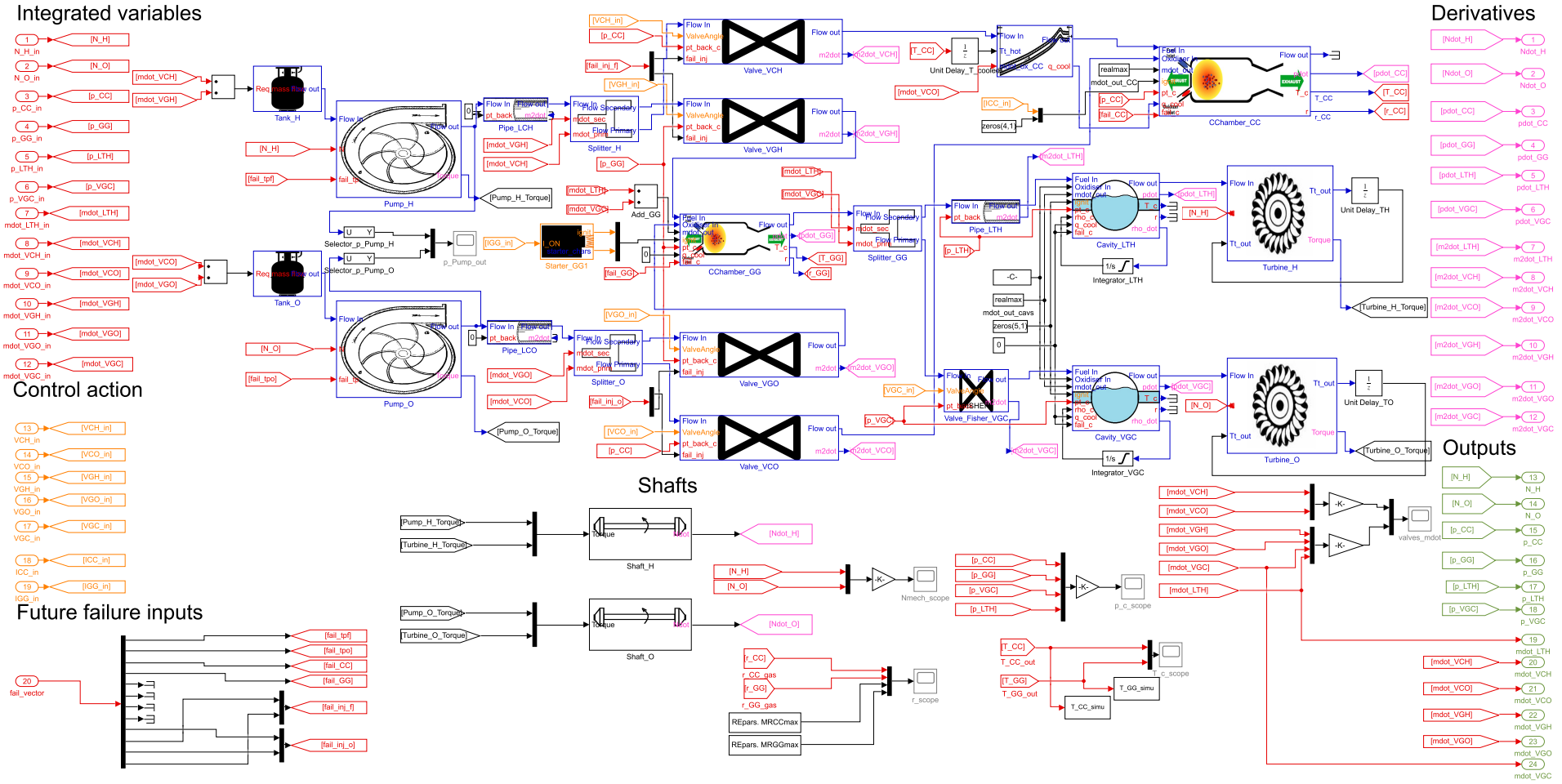


Figure 4.2.13: *Vulcain 1* simulator inner layer

4.3 Behaviour validation

Once the simulator was fully constructed, its behaviour was compared to the expected one. In the *Vulcain 1* case, the available parameters in this thesis have been received biased so as to alter the transient behaviour but obtain an approximately nominal equilibrium state after start-up. Hence, only a qualitative comparison of transient behaviour can be made. In contrast, a quantitative comparison of equilibrium states is possible. The key transient phase to simulate is the start-up of the engine, which is the main operation that the simulator has to capture. As presented in Section 3.1.1, these phases are executed as a series of discrete events at the beginning, concerning the opening of valves and the ignition of chambers. In the case of *Vulcain 1*, that event sequence is a pretty typical one for GG engines. In the simulations in this thesis, the sequence has been kept identical to the available biased nominal one at all cases:

1. VCH opens at $t_{VCH} = 0.1s$ (fuel-lead approach).
2. VCO opens at $t_{VCO} = 0.6s$.
3. Main chamber pyrotechnic ignition i_{CC} at $t_{i_{CC}} = 1s$, considered perfect and without delays.
4. Starter activates at $t_{i_{GG}} = 1.1s$. It provides full flow during $0.8s$ and half flow during $0.55s$ more.
5. VGH opens at $t_{VGH} = 1.4s$.
6. VGO opens at $t_{VGO} = 1.5s$.

Constant opening-angles commands are fed to valves, related to this engine's nominal steady-state: $\alpha_{VCH} = \alpha_{VCO} = 90°$, $\alpha_{VGH} = 72°$, $\alpha_{VGO} = 48°$ and $\alpha_{VGC} = 57°$. According to their internal-actuator dynamics (considered as second order), a certain time is required to reach that opening degree. CC valves present slower angular speeds than GG ones.
Figure 4.3.1 depicts the results in open loop during the first $4s$ of start-up operation. The figure shows results of normalised chambers pressures (a), mixture ratios (b), normalised mass flows through valves (c), chambers temperatures (d), valve opening angles (e) and normalised shaft rotational speeds (f). Normalisation is performed with respect to a nominal end state in each quantity: in pressures it is p_{CC}, in mass flows it is $\dot{m}_{VCH}$ for CC and $\dot{m}_{VGH}$ for GG flows, and in rotational speeds it is ω_H. Transient behaviour matches the expected results in terms of pressures, mass flows and rotational speeds, as validated by experts. Starter activation and the subsequent opening of GG valves can be clearly observed in the evolution of GG pressure. Besides, steady-state values, attained after $2.75s$ approximately, are within 1% close to design ones (Table 4.1.1), acceptable for an open-loop simulation. Mixture ratio starts to be meaningful up from $1.75s$ of simulation, where all valves are open and the starter is ending its contribution. From its definition, if the corresponding hydrogen flow becomes zero, the ratio tends to infinity, which happens during that interval after starter activation. These extreme ratios during the transient alter temperature calculation, whose transient evolution is not realistically predicted with this model, even though final values are correct. Not a high accuracy in combustion-temperature modelling has been implemented, as explained in 4.2.2. Thus, elevated peaks are attained at each respective ignition with overestimated slopes. However, it is certain that ignition processes represent abrupt discrete changes in system's evolution. A simplified approach to capture these events in the combustion pressure models (4.2.57) has been implemented in this chapter.

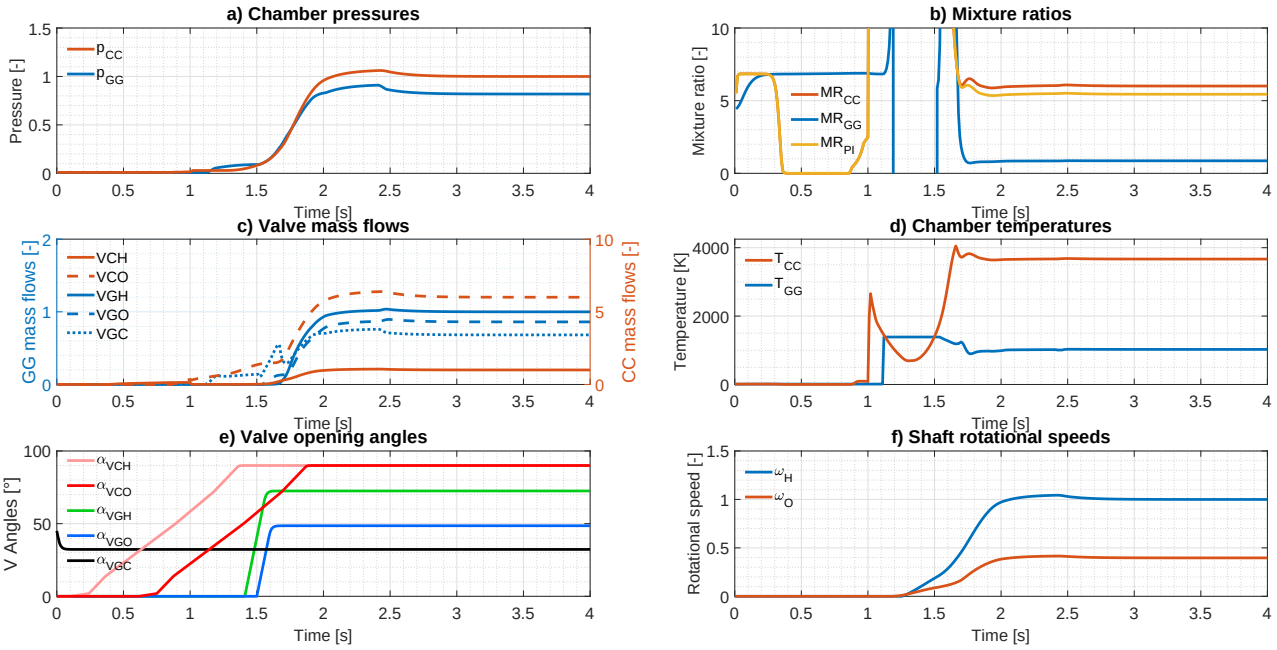


Figure 4.3.1: *Vulcain 1* T-RETM simulator results at start-up during $4s$

Some throttling scenarios have been simulated, in which valve's opening angles are varied so as to verify that the effects of OL control match the expected behaviour of the engine. The same simulation sequence has been set for the GG injection valves (VGH, VGO) and VGC: the first five seconds are nominal, then there are $7.5s$ of $\Delta\alpha = +15°$, $7.5s$ of $\Delta\alpha = -15°$ (with respect to nominal) and finally five more seconds of nominal behaviour. The effects of each type of valves (GG, VGC and CC) have been tested separately. Figure 4.3.2 depicts throttling by adjusting GG valves:

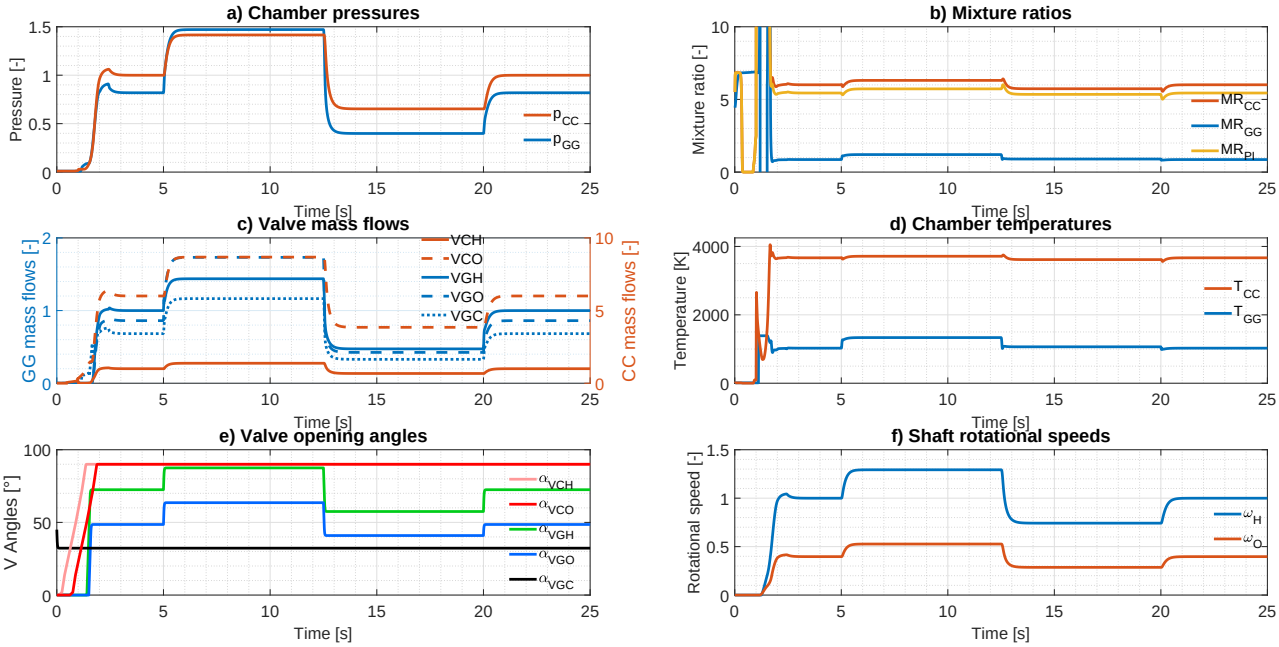


Figure 4.3.2: Throttling results by adjusting GG valves

It seems clear that these valves are directly related to chamber pressure and hence to thrust, one of the two variables to control. Figure 4.3.3 shows throttling results by adjusting the VGC valve:

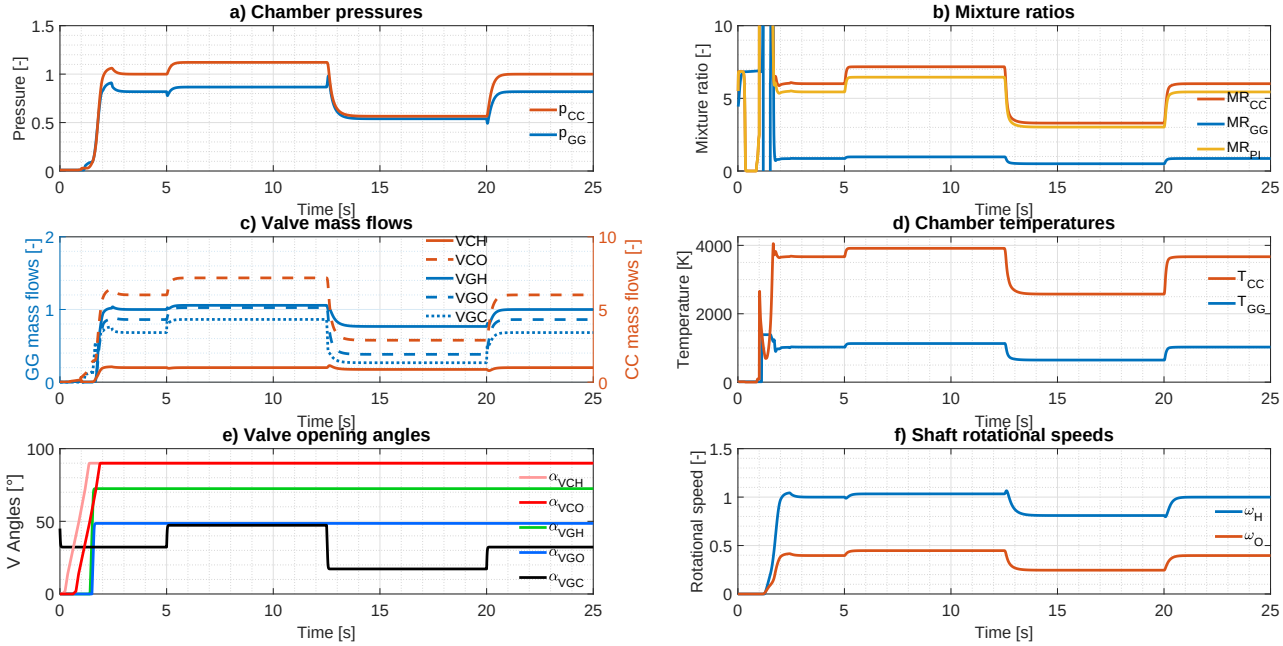


Figure 4.3.3: Throttling results by adjusting the VGC valve

As logical, this valve influences mixture ratio to a greater extent, because it is used to distribute the flow between turbines. Concerning CC valves, which are fully open in the nominal case, the throttling sequence has been 75% opening in the first abnormal phase and 25% in the second one. Figure 4.3.4 depicts throttling by adjusting these valves:

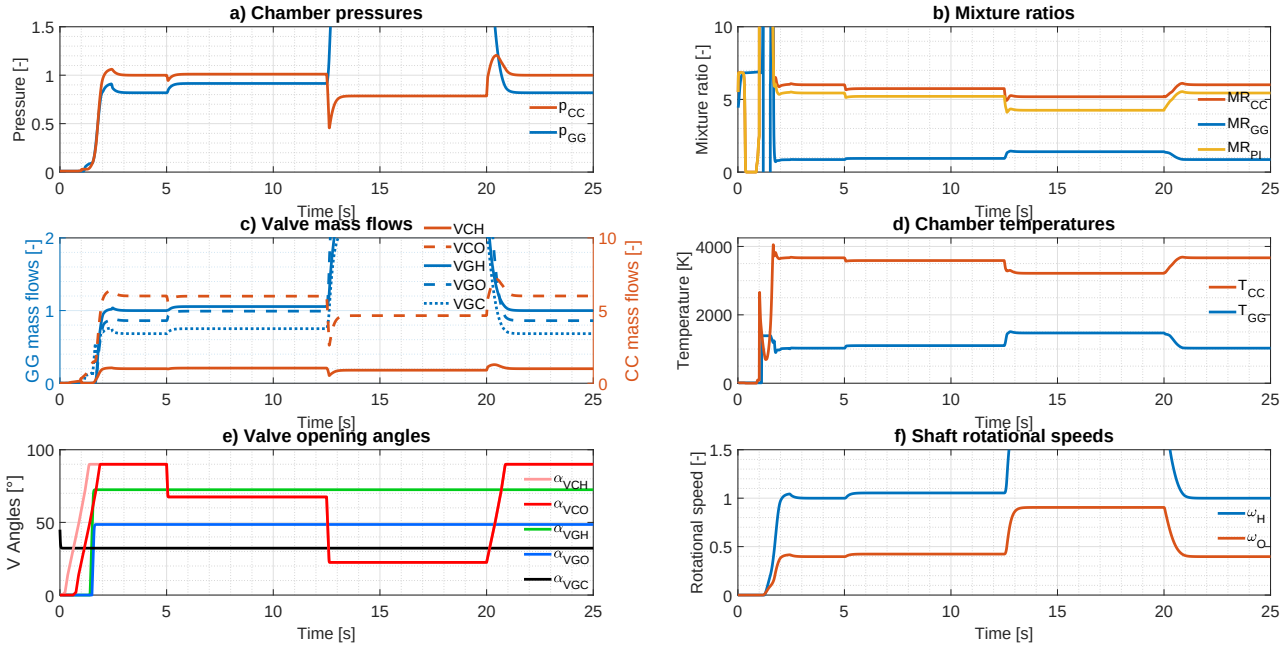


Figure 4.3.4: Throttling results by adjusting CC valves

In this case, closing the CC valves implies an increase in GG pressure and subsequently an increase in the CC pressure itself, which may seem illogical at first glance. However, it is physical since more pumped flow is directed towards the GG valves. This is captured by the model in the following way. First, CC valves' mass flows decrease. Hence, the pump is required to provide less output flow, but due to shaft's high inertia, the pump still presents a high rotational speed. By taking equation (4.2.15), and knowing that the coefficient associated to the square of mass flow is negative in this engine, it can be inferred that output pressure will be higher at a similar rotational speed if the mass flow to pump decreases. This effect provokes that both chamber pressures are higher for some incomplete openings. Hence, the throttling behaviour is qualitatively satisfactory, which will enable the simulation of off-design thrust levels acquired with the designed controllers in Chapters 6 and 7.

4.4 Summary

Along this chapter, a thermo-fluid-dynamic simulator representative of the gas-generator cycle LPRE has been constructed. The *Vulcain 1* engine has been the main case study, while a *PROMETHEUS* simulator has also been assembled. The approach is based on conservation equations on mass, momentum and energy. These typical thermodynamic equations have been applied to each basic component of a rocket engine, like valves, combustion chambers or turbopumps, while considering their specificities. A *Simulink*® library comprising these transient sub-models has been developed (*T-RETM*) with consistent fluid-dynamic assumptions. Apart from the plain conservation equations, no high-frequency transient phenomena, which could not be counteracted by the actuators of the cycle (valves), are considered in models. These components have then been joined to build a simulator of the engine. The fact of interconnecting all components while considering their interactions allows to define a set of global differential equations which are integrated in *Simulink*®. After its construction, the simulator has been validated qualitatively in terms of transient behaviour and throttling and quantitatively in terms of the nominal steady state. The entire start-up transient is satisfactorily predicted, with the exception of the less accurate temperature behaviour. In the next Chapter 5, the modelling approach used in this simulator is exploited in order to derive nonlinear state-space models of the GG engine. These state-space models and their analyses enable in the subsequent chapters to carry out model-based control.

5

Derivation and analysis of state-space models representative of gas-generator-cycle LPREs

The main goal of this chapter is to derive state-space models representative of gas-generator-cycle LPREs behaviour and valid for their model-based control. In other words, the target is a set of global nonlinear ODEs in which the thermodynamic evolution of the system depending on its state, internal parameters and control inputs is expressed. The thermo-fluid-dynamic model used in the simulator defined in the previous Chapter 4 is the start point for this derivation. In that simulator, the goal of the model was to precisely mimic the behaviour ot the GG LPRE. The manner in which that model was built, component-wise, led to a local definition of ODEs in each component. Interaction between components was expressed via a *Simulink*® diagram. In this chapter, the purpose of the model is different. In this case, it is aimed at achieving a global mathematical expression of the system as a whole, not just component-wise. Thus, the symbolic and causal interconnection of all component sub-models has been carried out.
A first set of simplifications has been applied along this translation process so as to obtain a mainly continuous state-space (SS) model representation, that captures the influence of some discrete inputs. In a second step, the obtained global SS has been further simplified in order to write a more tractable expression of the model for control purposes. This model has later been linearised, and a comprehensive analysis of its characteristics has been performed. The effects of simplifications and the properties of stability, controllability, measurability, observability and sensitivity to parameter variations have been analysed, confirming the suitability of the obtained SS models for the subsequent derivation of control laws. For the sake of clarity, these different modelling stages are listed:

- Simulator: from Chapter 4, intended to represent the real plant.
- Complex nonlinear SS model: first complete translation from simulator to the state-space formalism.
- Simplified nonlinear SS model: simplification of the complex model via mathematical and physical assumptions, in order to obtain tractable but representative differential equations.
- Linearised SS model: first-order TAYLOR series of the simplified nonlinear model.

This chapter is organised as follows. In Section 5.1 the translation process from simulator to state-space is explained. The outcome is presented and discussed in Section 5.2. The comparison of all modelling stages is carried out in Section 5.3 and their analysis is discussed in Section 5.4. The main contributions from this chapter, as well as from Chapter 4, led to the conference paper [118]. A summary of models analysis is included in the conference paper [120].

5.1 Translation of simulator into a nonlinear state-space model

The previous modelling approach in Chapter 4 consists in a component-wise thermodynamic modelling, conceived for simulation purposes. The goal of this second modelling phase is to obtain a global expression of the whole engine within a nonlinear SS model. For this sake, all components models have been joined by means of the symbolic computing environment *Maple*. In other words, adaptations of the equations derived in Chapter 4 have been associated according to cycle's flow schematic. The output quantities of every upstream component are fed to the inputs of every corresponding downstream one, always maintaining the symbolic calculation. In this manner, a general expression of engine's transient behaviour can be obtained, in which most of the former inter-component inputs and outputs do no longer appear in the final equations. This expression consists in a set of nonlinear ODEs which is a function of states (integrated variables), inputs (control action) and parameters (engine characteristics). For instance, pumps outlet pressures and temperatures do not appear because they are none of them. The initial structure of the GG model, based on the states considered in the simulator, is the following:

$$\dot{\mathbf{x}}_{sim} = f_{sim}(\mathbf{x}_{sim}, \mathbf{u}_c, \mathbf{u}_d, \mathbf{w}_t), \tag{5.1.1}$$

where $\mathbf{x}_{sim}$ are simulator states, $\mathbf{u}_c$ are continuous inputs (valves angles), $\mathbf{u}_d$ are discrete inputs (igniters and starters) and $\mathbf{w}_t$ is an exogenous time-varying input (starter mass flow $\dot{m}_{sta}$, following a profile 4.2.2). In the case of the *Vulcain 1* engine, these vectors represent the following variables:

$$\begin{bmatrix} \dot{\omega}_H \\ \dot{\omega}_O \\ \dot{p}_{CC} \\ \dot{p}_{GG} \\ \dot{p}_{LTH} \\ \dot{p}_{VGC} \\ \dot{\rho}_{CC} \\ \dot{\rho}_{GG} \\ \dot{\rho}_{LTH} \\ \dot{\rho}_{VGC} \\ \ddot{m}_{LTH} \\ \ddot{m}_{VCH} \\ \ddot{m}_{VCO} \\ \ddot{m}_{VGH} \\ \ddot{m}_{VGO} \\ \ddot{m}_{VGC} \end{bmatrix} = f_{sim} \left(\begin{bmatrix} \omega_H \\ \omega_O \\ p_{CC} \\ p_{GG} \\ p_{LTH} \\ p_{VGC} \\ \rho_{CC} \\ \rho_{GG} \\ \rho_{LTH} \\ \rho_{VGC} \\ \dot{m}_{LTH} \\ \dot{m}_{VCH} \\ \dot{m}_{VCO} \\ \dot{m}_{VGH} \\ \dot{m}_{VGO} \\ \dot{m}_{VGC} \end{bmatrix}, \begin{bmatrix} \alpha_{VCH} \\ \alpha_{VGH} \\ \alpha_{VCO} \\ \alpha_{VGO} \\ \alpha_{VGC} \end{bmatrix}, \begin{bmatrix} i_{CC} \\ i_{GG} \end{bmatrix}, \begin{bmatrix} \dot{m}_{sta} \end{bmatrix} \right), \tag{5.1.2}$$

which have been introduced in Chapter 4. Rotational speeds, cavity pressures and densities and pipe and valve mass flows are the states at the start point of the translation into a SS model, a total of sixteen in that engine.

At this phase, it was decided to discard some of the hybrid-modelling paradigms explained in Section 3.2. The reasons for this choice stem from the previous model structure, in which only two (or three in *PROMETHEUS*) discrete inputs $\mathbf{u}_d$ intervene in the system, and they are only switched once (from 0 to 1) for their activation during the start-up transient. Once chambers are ignited, the rest of time they are kept active and there is no interest in deactivating them for controlling the engine. Only in the case of a shutdown they are switched off, which is equivalent to combustion extinction. The real interest in terms of control is to know at what instant of time they should be activated for optimal and robustness purposes, as discussed in Section 7.3. In this sense, it has been opted for developing a hybrid-model conception similar to MLD [10, 91], in which binary variables directly intervene in differential equations for activating or cancelling certain terms. That modelling paradigm matches very well with components equations in Chapter 4, especially with the cavity pressure ODE (4.2.57), where combustion-related terms can be managed. Conversely, the structure of Petri Nets or automata would be conceptually further from LPREs behaviour, which is predominantly of continuous nature.

Along the symbolic component-connection process, some simplifications with respect to simulator's equations had to be performed so as to obtain globally defined expressions. That is to say, equations containing internal state-conditional or delayed-feedback statements distancing the model from the mainly continuous state-space formalism were modified. For the sake of simplicity, no additional discrete states were desired, which could have made up for those expressions; whereas discrete inputs are unavoidable. These simplifications, not altering to a relevant extent the transient behaviour of the system, consist in:

- All saturations in the previous thermodynamic equations, such as the one related to the choked mass flow (4.2.52), were eliminated. In consequence, the output throats of CC and pre-turbine cavities are always considered choked (not real during first instants of the start-up transient), the ideal-gas law yields densities (when required) throughout the whole cycle, and the GG output mass flow is fully determined by its downstream mass-flow states (no choked limitation). Similar considerations are made in the nonlinear model by [124].
- Turbines: in order to eliminate the problematic output-temperature feedback with delay, the assumption of a zero ΔT was checked to be acceptable. In addition, this simplification also helps reduce the complexity related to the low-regime extrapolation (4.2.62).
- Piecewise-continuous equations based on conditional statements, such as (4.2.37) and (4.2.41), can be expressed via the HEAVISIDE function. However, their introduction in a SS model is an avoidable source of complexity. Indeed, these expressions have been rewritten into single equations by means of smooth approximations of the HEAVISIDE function:

$$Heaviside(x) \approx \frac{1}{1+e^{-kx}}, \tag{5.1.3}$$

  with k arbitrarily big.

Another design goal of this translation consisted in obtaining differential equations on cavity temperatures instead of densities. Densities ODEs stem from the mass-conservation equations in cavities. However, densities are generally not measured in rocket engines. Consequently, they were re-expressed via the chain rule on the gas equation (4.2.1) as ODEs on temperature, which is generally measured. Nevertheless, this way of considering temperature ODE does not provide

more accuracy to the model. Indeed, temperature considerations in equations of Section 4.2.2 are too simplified for capturing their precise transient behaviour, as explained in Chapter 4. Much complex combustion models would have to be used, which is not of interest here. Thus, in the following, temperatures are just considered as correlated functions of the mixture ratios at the respective cavities ($T_{CC} = f_T(MR_{CC})$, $T_{GG} = f_T(MR_{GG})$, f_T being a polynomial depending on the propellant combination), and densities are statically computed via the cavities modified gas state equation (4.2.50). Furthermore, temperatures in pre-turbine cavities can be assumed equal to T_{GG} with little error, since those cavities receive the flow from the GG ($T_{LTH} \approx T_{GG}$, $T_{VGC} \approx T_{GG}$).

5.2 State-space models

Following this rewriting process, several SS models can be established according to the degree of complexity which is sought.

5.2.1 Complex nonlinear models

At this stage, having carried out the aforementioned simplifications with respect to the initial simulator, the state-space model is referred to as *complex NLSS* (nonlinear state-space) or $\dot{\mathbf{x}} = f_c(\mathbf{x}, \mathbf{u}_c, \mathbf{u}_d, \mathbf{w}_t)$. The structure of this model is the following in the case of *Vulcain 1*:

$$\begin{bmatrix} \dot{\omega}_H \\ \dot{\omega}_O \\ \dot{p}_{CC} \\ \dot{p}_{GG} \\ \dot{p}_{LTH} \\ \dot{p}_{VGC} \\ \ddot{m}_{LTH} \\ \ddot{m}_{VCH} \\ \ddot{m}_{VCO} \\ \ddot{m}_{VGH} \\ \ddot{m}_{VGO} \\ \ddot{m}_{VGC} \end{bmatrix} = f_{cV1} \left(\begin{bmatrix} \omega_H \\ \omega_O \\ p_{CC} \\ p_{GG} \\ p_{LTH} \\ p_{VGC} \\ \dot{m}_{LTH} \\ \dot{m}_{VCH} \\ \dot{m}_{VCO} \\ \dot{m}_{VGH} \\ \dot{m}_{VGO} \\ \dot{m}_{VGC} \end{bmatrix}, \begin{bmatrix} A_{VCH} \\ A_{VGH} \\ A_{VCO} \\ A_{VGO} \\ A_{VGC} \end{bmatrix}, \begin{bmatrix} i_{CC} \\ i_{GG} \end{bmatrix}, \begin{bmatrix} \dot{m}_{sta} \end{bmatrix} \right). \tag{5.2.1}$$

The number of states in that engine is $n = 12$ and $m = 5$ is the number of continuous control inputs. Here, the state vector $\mathbf{x}$ comprises 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 $\dot{m}_{LTH}$.

$$\mathbf{x} = [\omega_H \quad \omega_O \quad p_{CC} \quad p_{GG} \quad p_{LTH} \quad p_{VGC} \quad \dot{m}_{LTH} \quad \dot{m}_{VCH} \quad \dot{m}_{VCO} \quad \dot{m}_{VGH} \quad \dot{m}_{VGO} \quad \dot{m}_{VGC}]^T. \tag{5.2.2}$$

In these SS models, continuous inputs $\mathbf{u}_c$ are chosen as valve sections A, since they present a direct, nonlinear and monotone relation to angles (as highlighted in 4.2.2):

$$\mathbf{u}_c \equiv \mathbf{u} = [A_{VCH} \quad A_{VCO} \quad A_{VGH} \quad A_{VGO} \quad A_{VGC}]^T. \tag{5.2.3}$$

The reason for this choice is the reduction of nonlinearities, seeing that A enters in valves fluid-dynamic equations (4.2.25) directly. Discrete-inputs vector $\mathbf{u}_d \in \{0, 1\}$ consists of the activation of the CC igniter i_{CC} and of the GG starter/igniter i_{GG}. These activations must take place during the discrete sequential part of the start-up transient. The exogenous input $\mathbf{w}_t = \dot{m}_{sta}$ is also related to the sequence and is autonomously time-varying. It expresses the mass flow injected in the GG via the starter, which is known a priori all instants and influences GG thermodynamic parameters (concretely $R_{c,GG}$ via (4.2.46)). At steady state it is zero.

The precise form of this *Vulcain 1* system of differential equations is the following. The parameters and component abbreviations appearing in equations (5.2.4) to (5.2.15) have been introduced either in Chapter 4 or in this chapter. The subscript $_p$ denotes parameters considered varying (summarised in the upcoming Table 5.4.1), $_0$ stands for constant parameters, $_t$ for time dependency and $_{fp}$ for functions of other parameters. The terms with the subscript $_{fx}$ are dependent on the state and with $_{fu}$ on the input. The terms without these subscripts consist in states and inputs.

$$\begin{aligned}
\dot{\omega}_H = i_{GG} \Bigg[& 0.4 \left(a_{1TH_0} + RadTH_0 \omega_H \left(a_{2TH_0} + \frac{RadTH_0 a_{3TH_0} \omega_H}{\sqrt{\gamma_{outGG_{fp,x}} R_{outGG_{fp,x}} T_{GGfx}}} \right) \left(\sqrt{\gamma_{outGG_{fp,x}} R_{outGG_{fp,x}} T_{GGfx}} \right)^{-1} \right. \\
& + AthLEH_p \left(a_{4TH_0} + \frac{AthLEH_p a_{5TH_0}}{AthTH_p} \right) {AthTH_p}^{-1} + \frac{a_{6TH_0} AthLEH_p RadTH_0 \omega_H}{AthTH_p \sqrt{\gamma_{outGG_{fp,x}} R_{outGG_{fp,x}} T_{GGfx}}} \\
& \left. + a_{7TH_0} \ln \left(\frac{AthLEH_p}{AthTH_p} \right) + a_{8TH_0} \ln \left(\frac{RadTH_0 \omega_H}{\sqrt{\gamma_{outGG_{fp,x}} R_{outGG_{fp,x}} T_{GGfx}}} \right) \right) AthTH_p \gamma_{outGG_{fp,x}} RadTH_0 \left(1 \right. \\
& \left. - \left(\frac{AthLEH_p}{AthTH_p} \right)^{\frac{1-\gamma_{outGG_{fp,x}}}{\gamma_{outGG_{fp,x}}}} \right) \left(1 \right. \\
& \left. - \left(\frac{AthLEH_p}{AthTH_p} \right)^{-0.28571} \right)^{-1} \left(\sqrt{\left(2 \left(\gamma_{outGG_{fp,x}} + 1 \right)^{-1} \right)^{-\frac{\gamma_{outGG_{fp,x}}+1}{\gamma_{outGG_{fp,x}}-1}}} \right)^{-1} \left(\gamma_{outGG_{fp,x}} - 1 \right)^{-1} p_{LTH} \\
& - \frac{aCplPH_0 \dot{m}_{VCH}^2}{RhoPH_0} - 2 \frac{aCplPH_0 \dot{m}_{VCH} \dot{m}_{VGH}}{RhoPH_0} - \frac{aCplPH_0 \dot{m}_{VGH}^2}{RhoPH_0} - bCplPH_0 \omega_H \dot{m}_{VCH} - bCplPH_0 \omega_H \dot{m}_{VGH} \\
& \left. - cCplPH_0 RhoPH_0 {\omega_H}^2 \right] JinTPH_0{}^{-1},
\end{aligned} \tag{5.2.4}$$

$$\dot{\omega}_O = i_{GG}\left[0.4\left(a_{1TO_0} + RadTO_0\omega_O\left(a_{2\,TO_0} + \frac{RadTO_0 a_{3\,TO_0}\omega_O}{\sqrt{\gamma_{outGG_{fp,x}} R_{outGG_{fp,x}} T_{GGfx}}}\right)\left(\sqrt{\gamma_{outGG_{fp,x}} R_{outGG_{fp,x}} T_{GGfx}}\right)^{-1}\right.\right.$$
$$+ AthLEO_p\left(a_{4\,TO_0} + \frac{AthLEO_p a_{5\,TO_0}}{AthTO_p}\right) AthTO_p{}^{-1} + \frac{a_{6\,TO_0} AthLEO_p RadTO_0\omega_O}{AthTO_p\sqrt{\gamma_{outGG_{fp,x}} R_{outGG_{fp,x}} T_{GGfx}}}$$
$$\left.\left. + a_{7\,TO_0}\ln\left(\frac{AthLEO_p}{AthTO_p}\right) + a_{8\,TO_0}\ln\left(\frac{RadTO_0\omega_O}{\sqrt{\gamma_{outGG_{fp,x}} R_{outGG_{fp,x}} T_{GGfx}}}\right)\right) AthTO_p\gamma_{outGG_{fp,x}} RadTO_0\left(1\right.\right.$$
$$\left.\left. - \left(\frac{AthLEO_p}{AthTO_p}\right)^{\frac{1-\gamma_{outGG_{fp,x}}}{\gamma_{outGG_{fp,x}}}}\right)\left(1\right.\right.$$
$$\left.\left. - \left(\frac{AthLEO_p}{AthTO_p}\right)^{-0.28571}\right)^{-1}\left(\sqrt{\left(2\left(\gamma_{outGG_{fp,x}}+1\right)^{-1}\right)^{-\frac{\gamma_{outGG_{fp,x}}+1}{\gamma_{outGG_{fp,x}}-1}}}\right)^{-1}\left(\gamma_{outGG_{fp,x}}-1\right)^{-1} p_{VGC}$$
$$- \frac{aCplPO_0\dot{m}_{VCO}^2}{RhoPO_0} - 2\frac{aCplPO_0\dot{m}_{VCO}\dot{m}_{VGO}}{RhoPO_0} - \frac{aCplPO_0\dot{m}_{VGO}^2}{RhoPO_0} - bCplPO_0\omega_O\dot{m}_{VCO} - bCplPO_0\omega_O\dot{m}_{VGO}$$
$$\left. - cCplPO_0 RhoPO_0{\omega_O}^2\right] JinTPO_0{}^{-1}, \tag{5.2.5}$$

$$\dot{p}_{CC} \tag{5.2.6}$$
$$= \frac{\left(\gamma_{outCC_{fp,x}}-1\right)\left(\dot{m}_{VCH}+\dot{m}_{VCO}\right)}{V_{CC_0}}\left(\frac{\left(T_{PIH_p}\dot{m}_{VCH}+T_{PIO_p}\dot{m}_{VCO}\right)\left(R_{fu_0}\dot{m}_{VCH}+R_{ox_0}\dot{m}_{VCO}\right)\left(\gamma_{fu_0}\dot{m}_{VCH}+\gamma_{ox_0}\dot{m}_{VCO}\right)}{\left(\gamma_{fu_0}\dot{m}_{VCH}+\gamma_{ox_0}\dot{m}_{VCO}-\dot{m}_{VCH}-\dot{m}_{VCO}\right)\left(\dot{m}_{VCH}+\dot{m}_{VCO}\right)^2}\right.$$
$$\left. + \frac{T_{CCfx}R_{outCC_{fp,x}}\left(\gamma_{fu_0}\dot{m}_{VCH}+\gamma_{ox_0}\dot{m}_{VCO}-\gamma_{outCC_{fp,x}}\dot{m}_{VCH}-\gamma_{outCC_{fp,x}}\dot{m}_{VCO}\right)\left(R_{fu_0}\dot{m}_{VCH}+R_{ox_0}\dot{m}_{VCO}\right)}{\left(\gamma_{outCC_{fp,x}}-1\right)^2\eta_{CC_p}\left(Cp_{outCC_{fp,x}}-R_{outCC_{fp,x}}\right)\left(\dot{m}_{VCH}+\dot{m}_{VCO}\right)\left(\gamma_{fu_0}\dot{m}_{VCH}+\gamma_{ox_0}\dot{m}_{VCO}-\dot{m}_{VCH}-\dot{m}_{VCO}\right)}\right)$$
$$- {\gamma_{outCC_{fp,x}}}^{3/2} AthCH_p\sqrt{R_{outCC_{fp,x}}}\sqrt{T_{CCfx}}2^{\frac{\gamma_{outCC_{fp,x}}+1}{2\gamma_{outCC_{fp,x}}-2}}\left(\gamma_{outCC_{fp,x}}+1\right)^{-\frac{\gamma_{outCC_{fp,x}}+1}{2\gamma_{outCC_{fp,x}}-2}} {V_{CC_0}}^{-1} p_{CC}$$
$$- i_{CC}\left[\left(-\frac{\left(\gamma_{outCC_{fp,x}}-1\right)\eta_{CC_p}\mu_{oxinCC_{fx}}h_{fox_0}}{V_{CC_0}} - \frac{\left(\gamma_{outCC_{fp,x}}-1\right)\eta_{CC_p}\mu_{fuinCC_{fx}}h_{ffu_0}}{V_{CC_0}}\right.\right.$$
$$\left. + \frac{\left(\gamma_{outCC_{fp,x}}-1\right)\eta_{CC_p}\left(\mu_{fuinCC_{fx}}+\mu_{oxinCC_{fx}}\right)\mu_{p1_0}h_{fp1_0}}{V_{CC_0}}\right.$$
$$+ \frac{\gamma_{outCC_{fp,x}}R_{outCC_{fp,x}}T_{CCfx}\mu_{oxinCC_{fx}}}{\left(\gamma_{outCC_{fp,x}}-1\right)\eta_{CC_p}V_{CC_0}}\left(\frac{Cp_{oxCC_{fp,x}}}{Cp_{outCC_{fp,x}}} - \frac{Cp_{oxCC_{fp,x}}-R_{ox_0}}{Cp_{outCC_{fp,x}}-R_{outCC_{fp,x}}}\right)$$
$$+ \frac{\gamma_{outCC_{fp,x}}R_{outCC_{fp,x}}T_{CCfx}\mu_{fuinCC_{fx}}}{\left(\gamma_{outCC_{fp,x}}-1\right)\eta_{CC_p}V_{CC_0}}\left(\frac{Cp_{fuCC_{fp,x}}}{Cp_{outCC_{fp,x}}} - \frac{Cp_{fuCC_{fp,x}}-R_{fu_0}}{Cp_{outCC_{fp,x}}-R_{outCC_{fp,x}}}\right)$$
$$\left. - \frac{\gamma_{outCC_{fp,x}}R_{outCC_{fp,x}}T_{CCfx}\left(\mu_{fuinCC_{fx}}+\mu_{oxinCC_{fx}}\right)\mu_{p1_0}}{\left(\gamma_{outCC_{fp,x}}-1\right)\eta_{CC_p}V_{CC_0}}\left(\frac{Cp_{p1CC_{fp,x}}}{Cp_{outCC_{fp,x}}} - \frac{Cp_{p1CC_{fp,x}}-R_{p1_0}}{Cp_{outCC_{fp,x}}-R_{outCC_{fp,x}}}\right)\right)\left(\dot{m}_{VCH}+\dot{m}_{VCO}\right)$$
$$\left. - \frac{\left(\gamma_{outCC_{fp,x}}-1\right)\left(Cp_{outCC_{fp,x}}T_{vap,ox_0} - Lv_{ox_0} - Cp_{oxCC_{fp,x}}\left(T_{vap,ox_0}-T_{PIO_p}\right)\right)\dot{m}_{VCO}}{V_{CC_0}}\right],$$

$$\dot{p}_{GG} = \left(\gamma_{outGG_{fp,x}} - 1\right)\left(\dot{m}_{VGH} + \dot{m}_{VGO} + \dot{m}_{sta_t}\right) {V_{GG_0}}^{-1} \left(\frac{\left(T_{PIH_p}\dot{m}_{VGH} + T_{PIO_p}\dot{m}_{VGO}\right)\left(R_{fu_0}\dot{m}_{VGH} + R_{ox_0}\dot{m}_{VGO}\right)\left(\gamma_{fu_0}\dot{m}_{VGH} + \gamma_{ox_0}\dot{m}_{VGO}\right)}{\left(\gamma_{fu_0}\dot{m}_{VGH} + \gamma_{ox_0}\dot{m}_{VGO} - \dot{m}_{VGH} - \dot{m}_{VGO}\right)\left(\dot{m}_{VGH} + \dot{m}_{VGO}\right)^2} + \frac{T_{GGfx}R_{cGG_{fp,x,t}}\left(\gamma_{fu_0}\dot{m}_{VGH} + \gamma_{ox_0}\dot{m}_{VGO} - \gamma_{outGG_{fp,x}}\dot{m}_{VGH} - \gamma_{outGG_{fp,x}}\dot{m}_{VGO}\right)\left(R_{fu_0}\dot{m}_{VGH} + R_{ox_0}\dot{m}_{VGO}\right)}{\left(\gamma_{outGG_{fp,x}} - 1\right)^2 \eta_{GG_p}\left(Cp_{outGG_{fp,x}} - R_{outGG_{fp,x}}\right)\left(\dot{m}_{VGH} + \dot{m}_{VGO}\right)\left(\gamma_{fu_0}\dot{m}_{VGH} + \gamma_{ox_0}\dot{m}_{VGO} - \dot{m}_{VGH} - \dot{m}_{VGO}\right)} \right) - \frac{\gamma_{outGG_{fp,x}}R_{cGG_{fp,x,t}}T_{GGfx}}{V_{GG_0}}\left(\dot{m}_{LTH} + \dot{m}_{VGC}\right) - i_{GG}\left[\left(-\frac{\left(\gamma_{outGG_{fp,x}} - 1\right)\eta_{GG_p}\mu_{oxinGG_{fx}}h_{fox_0}}{V_{GG_0}} - \frac{\left(\gamma_{outGG_{fp,x}} - 1\right)\eta_{GG_p}\mu_{fuinGG_{fx}}h_{ffu_0}}{V_{GG_0}} + \frac{\left(\gamma_{outGG_{fp,x}} - 1\right)\eta_{GG_p}\left(\mu_{fuinGG_{fx}} + \mu_{oxinGG_{fx}}\right)\mu_{p1_0}h_{fp1_0}}{V_{GG_0}} + \frac{\gamma_{outGG_{fp,x}}R_{cGG_{fp,x,t}}T_{GGfx}\mu_{oxinGG_{fx}}}{\left(\gamma_{outGG_{fp,x}} - 1\right)\eta_{GG_p}V_{GG_0}}\left(\frac{Cp_{ox_{fp,x}}}{Cp_{outGG_{fp,x}}} - \frac{Cp_{ox_{fp,x}} - R_{ox_0}}{Cp_{outGG_{fp,x}} - R_{outGG_{fp,x}}}\right) + \frac{\gamma_{outGG_{fp,x}}R_{cGG_{fp,x,t}}T_{GGfx}\mu_{fuinGG_{fx}}}{\left(\gamma_{outGG_{fp,x}} - 1\right)\eta_{GG_p}V_{GG_0}}\left(\frac{Cp_{fu_{fp,x}}}{Cp_{outGG_{fp,x}}} - \frac{Cp_{fu_{fp,x}} - R_{fu_0}}{Cp_{outGG_{fp,x}} - R_{outGG_{fp,x}}}\right) - \frac{\gamma_{outGG_{fp,x}}R_{cGG_{fp,x,t}}T_{GGfx}\left(\mu_{fuinGG_{fx}} + \mu_{oxinGG_{fx}}\right)\mu_{p1_0}}{\left(\gamma_{outGG_{fp,x}} - 1\right)\eta_{GG_p}V_{GG_0}}\left(\frac{Cp_{p1_{fp,x}}}{Cp_{outGG_{fp,x}}} - \frac{Cp_{p1_{fp,x}} - R_{p1_0}}{Cp_{outGG_{fp,x}} - R_{outGG_{fp,x}}}\right)\right)\left(\dot{m}_{VGH} + \dot{m}_{VGO}\right) - \frac{\left(\gamma_{outGG_{fp,x}} - 1\right)\left(Cp_{outGG_{fp,x}}T_{vap,ox_0} - Lv_{ox_0} - Cp_{ox_{fp,x}}\left(T_{vap,ox_0} - T_{PIO_p}\right)\right)\dot{m}_{VGO}}{V_{GG_0}}\right], \tag{5.2.7}$$

$$\dot{p}_{LTH} = \frac{\gamma_{outGG_{fp,x}}R_{outGG_{fp,x}}T_{GGfx}}{V_{cav_p}}\dot{m}_{LTH} - \frac{{\gamma_{outGG_{fp,x}}}^{3/2}AthTH_p\sqrt{R_{outGG_{fp,x}}}\sqrt{T_{GGfx}}\,2^{\frac{\gamma_{outGG_{fp,x}}+1}{2\,\gamma_{outGG_{fp,x}}-2}}\left(\gamma_{outGG_{fp,x}} + 1\right)^{-\frac{\gamma_{outGG_{fp,x}}+1}{2\,\gamma_{outGG_{fp,x}}-2}}}{V_{cav_p}}p_{LTH}, \tag{5.2.8}$$

$$\dot{p}_{VGC} = \frac{\gamma_{outGG_{fp,x}}R_{outGG_{fp,x}}T_{GGfx}}{V_{cav_p}}\dot{m}_{VGC} - \frac{{\gamma_{outGG_{fp,x}}}^{3/2}AthTO_p\sqrt{R_{outGG_{fp,x}}}\sqrt{T_{GGfx}}\,2^{\frac{\gamma_{outGG_{fp,x}}+1}{2\,\gamma_{outGG_{fp,x}}-2}}\left(\gamma_{outGG_{fp,x}} + 1\right)^{-\frac{\gamma_{outGG_{fp,x}}+1}{2\,\gamma_{outGG_{fp,x}}-2}}}{V_{cav_p}}p_{VGC}, \tag{5.2.9}$$

$$\ddot{m}_{LTH} = \left(p_{GG} - p_{LTH} - \frac{KhyLTH_pR_{outGG_{fp,x}}T_{GGfx}}{p_{GG}}\dot{m}_{LTH}^2\right){InePipe_{LTH_p}}^{-1}, \tag{5.2.10}$$

$$\ddot{m}_{VCH} = \left[RhoPH_0cPrsPH_{fx}{\omega_H}^2 + \left(bPrsPH_{fx}\dot{m}_{VCH} + bPrsPH_{fx}\dot{m}_{VGH}\right)\omega_H + \left(-ResHIC_p - ResLCH_p + \frac{aPrsPH_{fx}}{RhoPH_0} - ResCR_p\right)\dot{m}_{VCH}^2 + \left(2\,\frac{aPrsPH_{fx}}{RhoPH_0} - 2\,ResLCH_p\right)\dot{m}_{VGH}\dot{m}_{VCH} + \left(-ResLCH_p + \frac{aPrsPH_{fx}}{RhoPH_0}\right)\dot{m}_{VGH}^2 + p_{PIH_p} - p_{CC} - \frac{\dot{m}_{VCH}^2}{2RhoVCH_0{A_{VCH}}^2}\right]\left(\frac{LngVCH_0}{A_{VCH}} + IneCR_p\right)^{-1}, \tag{5.2.11}$$

$$\ddot{m}_{VCO} = \left[\frac{RhoPO_0 cPrsPO_0}{LngVCO_0}{\omega_O}^2 + \left(\frac{bPrsPO_0\dot{m}_{VCO}}{LngVCO_0} + \frac{bPrsPO_0\dot{m}_{VGO}}{LngVCO_0}\right)\omega_O \right.$$
$$+ \left(-ResOIC_p - ResLCO_p + \frac{aPrsPO_0}{RhoPO_0}\right) {LngVCO_0}^{-1}\dot{m}_{VCO}^2$$
$$+ \left(2\,\frac{aPrsPO_0}{RhoPO_0} - 2\,ResLCO_p\right) {LngVCO_0}^{-1}\dot{m}_{VGO}\dot{m}_{VCO} + \left(-ResLCO_p + \frac{aPrsPO_0}{RhoPO_0}\right) {LngVCO_0}^{-1}\dot{m}_{VGO}^2$$
$$\left. + \frac{p_{PIO_p} - p_{CC}}{LngVCO_0}\right] A_{VCO} - \frac{\dot{m}_{VCO}^2}{2RhoVCO_0 LngVCO_0 A_{VCO}}, \tag{5.2.12}$$

$$\ddot{m}_{VGH} = \left[\frac{RhoPH_0 cPrsPH_{fx}}{LngVGH_0}{\omega_H}^2 + \left(\frac{bPrsPH_{fx}\dot{m}_{VCH}}{LngVGH_0} + \frac{bPrsPH_{fx}\dot{m}_{VGH}}{LngVGH_0}\right)\omega_H \right.$$
$$+ \left(-ResLCH_p + \frac{aPrsPH_{fx}}{RhoPH_0}\right) {LngVGH_0}^{-1}\dot{m}_{VCH}^2 + \left(2\,\frac{aPrsPH_{fx}}{RhoPH_0} - 2\,ResLCH_p\right) {LngVGH_0}^{-1}\dot{m}_{VGH}\dot{m}_{VCH}$$
$$\left. + \left(-ResHIG_p + \frac{aPrsPH_{fx}}{RhoPH_0} - ResLCH_p\right) {LngVGH_0}^{-1}\dot{m}_{VGH}^2 + \frac{p_{PIH_p} - p_{GG}}{LngVGH_0}\right] A_{VGH}$$
$$- \frac{\dot{m}_{VGH}^2}{2RhoVGH_0 LngVGH_0 A_{VGH}}, \tag{5.2.13}$$

$$\ddot{m}_{VGO} = \left[\frac{RhoPO_0 cPrsPO_0}{LngVGO_0}{\omega_O}^2 + \left(\frac{bPrsPO_0\dot{m}_{VCO}}{LngVGO_0} + \frac{bPrsPO_0\dot{m}_{VGO}}{LngVGO_0}\right)\omega_O \right.$$
$$+ \left(-ResLCO_p + \frac{aPrsPO_0}{RhoPO_0}\right) {LngVGO_0}^{-1}\dot{m}_{VCO}^2 + \left(2\,\frac{aPrsPO_0}{RhoPO_0} - 2\,ResLCO_p\right) {LngVGO_0}^{-1}\dot{m}_{VGO}\dot{m}_{VCO}$$
$$\left. + \left(-ResOIG_p + \frac{aPrsPO_0}{RhoPO_0} - ResLCO_p\right) {LngVGO_0}^{-1}\dot{m}_{VGO}^2 + \frac{p_{PIO_p} - p_{GG}}{LngVGO_0}\right] A_{VGO} - \frac{\dot{m}_{VGO}^2}{2RhoVGO_0 LngVGO_0 A_{VGO}}, \tag{5.2.14}$$

$$\ddot{m}_{VGC} = p_{GG} - p_{VGC} - \frac{\zeta VGC_{fu} R_{outGG_{fp,x}} T_{GGfx}}{2p_{GG}{A_{VGC}}^2}\dot{m}_{VGC}^2. \tag{5.2.15}$$

The complex state-space model corresponding to the *PROMETHEUS* engine, developed with the same methodology and assumptions, is available in Appendix A. That model presents a very similar structure to the *Vulcain 1*. Differences arise from the presence of a single turbine instead of two, and from the creation of two products (water and carbon dioxide) in the combustion reaction instead of one. The former leads to shorter state and input vectors, where $n = 9$ and $m = 4$; and the latter introduce more terms in cavity pressure ODEs (8.2.2) and (8.2.3). The differential equation on rotational speed (8.2.1) contains the contribution of both fuel and oxidiser pumps which are connected to the same shaft. Hence, in this engine, the state vector is:

$$\mathbf{x} = [\omega \quad p_{CC} \quad p_{GG} \quad p_{LT} \quad \dot{m}_{LT} \quad \dot{m}_{VCM} \quad \dot{m}_{VCO} \quad \dot{m}_{VGM} \quad \dot{m}_{VGO}]^T, \tag{5.2.16}$$

and the control inputs are:

$$\mathbf{u}_c \equiv \mathbf{u} = [A_{VCM} \quad A_{VCO} \quad A_{VGM} \quad A_{VGO}]^T, \tag{5.2.17}$$
$$\mathbf{u}_d = [i_{CC} \quad i_{GG} \quad i_{sta}]^T. \tag{5.2.18}$$

The absence of a second turbine eliminates the need for a flow-distribution valve (VGC in *Vulcain 1*). An additional discrete input i_{sta} has to be added since GG ignition and starter activation are not simultaneous, in contrast to *Vulcain 1*.

In the following, the letter F (fuel) is used as an umbrella acronym to consider the equivalent component names depending on the fuel, e.g. VGH or VGM (VGF), and avoid repetitions.

These GG-LPRE state-space system representations with time-varying coefficients and continuous and discrete inputs for control purposes are one of the main contributions of this thesis, as reviewed in Section 2.3. The most usual modelling approaches had concerned linear or linearised models coming from identification or simplified fluid mechanics. The closest control-oriented nonlinear modelling approach is the staged-combustion LPRE model in [124], used in the articles [36, 37, 89, 166]. That approach also considered causal component interconnections but concerned time-invariant coefficients and only continuous inputs.

5.2.2 Simplified nonlinear models

Up from this point, this resulting f_c system was analysed and checked to still be far too complex for deriving nonlinear control laws. It is noticeable that differential equations present numerous high-order and coupled nonlinearities, with many interdependencies and varying terms. Hence, a further list of simplifications was implemented until achieving the here-called *simplified NLSS* such that $\dot{\mathbf{x}} = f_s(\mathbf{x}, \mathbf{u}_c, \mathbf{u}_d, \mathbf{w}_t)$, more tractable for its manipulation. This processing slightly reduces model fidelity during transients but eases the development of control algorithms, which will have to cope with this model mismatch. These modifications are:

a) Flow thermodynamic properties of cavities (C_p, R, γ, μ) formally depend on state-dependent variables such as mass-flow ratios (MR) and/or temperatures. This fact complexifies and couples to a great extent the model, introducing high-order nonlinearities. Thus, their values are considered constant. The ones depending on MR, (R, γ and μ at every cavity inlet and outlet) are evaluated at the corresponding desired end MR. This allows the removal of HEAVISIDE functions in chambers mass fractions computations (4.2.37) to (4.2.41). Temperature-dependent ones (C_p) are averaged along the start-up transient. Distinctions according to the chamber, CC or GG, are made due to the relevant temperature and mixture differences.

b) This choice of considering constant MR and T for thermodynamic-properties evaluation in order to reduce state dependencies presents some drawbacks. The main one is the alteration of the starter influence on the GG via $\mathbf{w}_t$, which becomes too powerful if the same considerations as in (4.2.46) are made. Hence, in this simplified model, the weighted influence of the starter mass-flow fraction μ_{sta} has to be considered in GG input and output thermodynamic properties T_{inGG}, R_{inGG}, γ_{inGG}, Cp_{inGG}, R_{outGG} and Cp_{outGG}. In addition, in the calculation of μ_{sta} itself, a constant injected mass fraction from valves is considered to avoid state coupling (with GG mass flows). These increased time dependencies, seeing that $\mathbf{w}_t$ is known, are less inconvenient than maintaining the set of state dependencies which are removed by applying simplification a).

c) VGC pressure-drop coefficient ζ_{VGC} is considered equal to its design value, so as to avoid its polynomial dependency on the angular input (4.2.27).

d) The original hydrogen-pump pressure coefficients a_{PrsPH}, b_{PrsPH} and c_{PrsPH} vary according to a condition on states (explained in 4.2.2) in *Vulcain 1*. Hence, they are recalculated here so as to cover the whole operating range and to avoid the conditional statement.

e) The expressions of chamber's characteristic speeds C^* in choked mass-flow equations (4.2.52) and of turbine specific torque (4.2.61) are rewritten into simplified first-order expressions via least-squares regression so as to avoid their highly nonlinear terms. In the case of the C^* terms dependent on γ, the obtained coefficients are named m_γ (slope) and n_γ (ordinate at origin).

f) After an empirical study of the relative size of all the remaining terms in equations along transient simulations, the ones presenting three orders of magnitude less than the rest were neglected. These correspond in general to terms associated to the squares of GG mass flows (small), to vaporisation heat in cavities, and to some state products in valves equations (5.2.11) to (5.2.14).

The same amount of states and inputs has been maintained since all of them are relevant to system's dynamics, having similar time scales and coupling effects. Indeed, the whole system evolves simultaneously in terms of states, as seen in simulations from Chapter 4. The cost of these simplifications is the increase in modelling error, which is investigated in Section 5.3.

The equations in the simplified NLSS for *Vulcain 1*, in which states and inputs are denoted by x_i and u_i for ease of visualisation, are presented below. The parameters $< k... >$ are positive conglomerates of engine parameters which serve to express the simplified system of ODEs in a more compact manner. The definition of these parameters as well as the respective ODEs for *PROMETHEUS*, can be found in Appendix B. Since they are only dependent on parameters and on the exogenous input influence, they are either constant or time-dependent. With the aforementioned simplifications, no state or control dependencies are present within those parameter composites.

$$\dot{x}_1 = \left(-\frac{kom_{H2_{fp,t}} x_5 x_1}{\sqrt{T_{GG_{fx}}}} + kom_{H4_{fp,t}} x_5 - kom_{H3_{fp}} \left(x_8 + x_{10}\right) x_1 - kom_{H5_{fp}} \left({x_8}^2 + x_8 x_{10}\right) \right) i_{GG}, \tag{5.2.19}$$

$$\dot{x}_2 = \left(kom_{O1_{fp}} {x_2}^2 - \frac{kom_{O2_{fp,t}} x_6 x_2}{\sqrt{T_{GG_{fx}}}} + kom_{O4_{fp,t}} x_6 - kom_{O3_{fp}} \left(x_9 + x_{11}\right) x_2 - kom_{O5_{fp}} \left({x_9}^2 + x_9 x_{11}\right) \right) i_{GG}, \tag{5.2.20}$$

$$\begin{aligned} \dot{x}_3 = {} & \left(\left(kp_{CC4_{fp}} - kp_{CC1_{fp}} T_{CC_{fx}} \right) x_8 + \left(kp_{CC4_{fp}} - kp_{CC1_{fp}} T_{CC_{fx}} \right) x_9 \right) i_{CC} \\ & + \left(kp_{CC5_{fp}} + kp_{CC2_{fp}} T_{CC_{fx}} \right) x_8 + \left(kp_{CC5_{fp}} + kp_{CC2_{fp}} T_{CC_{fx}} \right) x_9 - kp_{CC3_{fp}} \sqrt{T_{CC_{fx}}} x_3, \end{aligned} \tag{5.2.21}$$

$$\begin{aligned} \dot{x}_4 = {} & \left(\left(kp_{GG5_{fp}} - kp_{GG1_{fp,t}} T_{GG_{fx}} \right) x_{10} + \left(kp_{GG5_{fp}} - kp_{GG1_{fp,t}} T_{GG_{fx}} \right) x_{11} \right) i_{GG} + \left(kp_{GG6_{fp,t}} + kp_{GG2_{fp,t}} T_{GG_{fx}} \right) x_{10} \\ & + \left(kp_{GG6_{fp,t}} + kp_{GG2_{fp,t}} T_{GG_{fx}} \right) x_{11} - kp_{GG4_{fp,t}} T_{GG_{fx}} \left(x_7 + x_{12}\right) + \left(kp_{GG6_{fp,t}} + kp_{GG2_{fp,t}} T_{GG_{fx}} \right) w_t, \end{aligned} \tag{5.2.22}$$

$$\dot{x}_5 = kp_{LTH1_{fp,t}} T_{GG_{fx}} x_7 - kp_{LTH2_{fp,t}} \sqrt{T_{GG_{fx}}} x_5, \tag{5.2.23}$$

$$\dot{x}_6 = kp_{VGC1_{fp,t}} T_{GG_{fx}} x_{12} - kp_{VGC2_{fp,t}} \sqrt{T_{GG_{fx}}} x_6, \tag{5.2.24}$$

$$\dot{x}_7 = \frac{x_4 - x_5}{InePipe_{LTH_p}} - \frac{km_{LTH_{fp,t}} T_{GG_{fx}} {x_7}^2}{x_4}, \tag{5.2.25}$$

$$\dot{x}_8 = \frac{\left(km_{VCH3_{fp}} {x_1}^2 - km_{VCH4_{fp}} {x_8}^2 - km_{VCH5_{fp}} x_{10} x_8 - km_{VCH6_{fp}} x_3 + km_{VCH7_{fp}}\right) {u_1}^2 - km_{VCH8_{fp}} {x_8}^2}{\left(km_{VCH1_{fp}} u_1 + km_{VCH2_{fp}}\right) u_1}, \tag{5.2.26}$$

$$\dot{x}_9 = \left(km_{VCO1_{fp}} {x_2}^2 - km_{VCO2_{fp}} x_9 x_2 - km_{VCO3_{fp}} {x_9}^2 - km_{VCO4_{fp}} x_3 + km_{VCO5_{fp}}\right) u_2 - \frac{km_{VCO6_{fp}} {x_9}^2}{u_2}, \tag{5.2.27}$$

$$\dot{x}_{10} = \left(km_{VGH1_{fp}} {x_1}^2 - km_{VGH2_{fp}} {x_8}^2 - km_{VGH3_{fp}} x_{10} x_8 - km_{VGH4_{fp}} {x_{10}}^2 - km_{VGH5_{fp}} x_4 + km_{VGH6_{fp}}\right) u_3 - \frac{km_{VGH7_{fp}} {x_{10}}^2}{u_3}, \tag{5.2.28}$$

$$\dot{x}_{11} = \left(km_{VGO1_{fp}} {x_2}^2 - km_{VGO2_{fp}} x_9 x_2 - km_{VGO3_{fp}} {x_9}^2 - km_{VGO4_{fp}} {x_{11}}^2 - km_{VGO5_{fp}} x_4 + km_{VGO6_{fp}}\right) u_4 - \frac{km_{VGO7_{fp}} {x_{11}}^2}{u_4}, \tag{5.2.29}$$

$$\dot{x}_{12} = \left(x_4 - x_6 - \frac{\zeta VGC_{f0} R_{outGG_{fp,t}} T_{GG_{fx}} {x_{12}}^2}{2 x_4 {u_5}^2}\right) u_5. \tag{5.2.30}$$

An easier analysis of system's structure can be performed on this f_s rather than on f_c. If T_{CC} and T_{GG} are considered as varying parameters, calculated as external functions of the current state; and discrete inputs (i_{CC}, i_{GG}) are active, the pressure ODE (5.2.21) to (5.2.24) become linear with respect to states and to the exogenous input $\mathbf{w_t}$. Rotational speed equations (5.2.19) to (5.2.20) remain nonlinear due to the presence of non-negligible state products and quadratic terms. The pipe mass-flow ODE (5.2.25) also presents a division by a state. The influence of control inputs is only present in valve mass-flow equations (5.2.26) to (5.2.30). Each valve section u_i directly influences its respective mass-flow equation exclusively. Some of these equations can be split into affine and non-affine terms with respect to control in the form $\dot{x}_i = g(\mathbf{x})u_i + h(\mathbf{x}, u_i)$. Those equations are (5.2.27) to (5.2.30), where the respective $g(\mathbf{x})$ are quadratic functions of the state and $h(\mathbf{x}, \mathbf{u})$ present quadratic terms on state and control inputs in the denominator. Equation (5.2.26) is particularly complex since no affine part can be extracted. Hence, this simplified SS is nonlinear in state and control and non-affine in control.

In all the SS models derived here, every differential equation, state and control input has been rendered non-dimensional with respect to the nominal equilibrium values and to a unitary time scale. The purpose of the non-dimensionalisation was to homogenise the orders of magnitude of all quantities, which in dimensional terms can present large differences, from 10^7 for pressures in Pa, to 10^{-5} of sections in m^2. Keeping equations dimensional was detrimental for the conditioning of linearised matrices, derived later in this chapter. Hence, in the following, all states and inputs are deemed dimensionless.

5.2.3 Linearised models

These simplified nonlinear models are then linearised (first-order TAYLOR series via *Maple*) with respect to $\mathbf{x}$ and $\mathbf{u}$ to obtain the following system representation:

$$\dot{\mathbf{x}} = A_c(\mathbf{x}_r, \mathbf{u}_r, \mathbf{w}_{t,r})\boldsymbol{\Delta}\mathbf{x} + B_c(\mathbf{x}_r, \mathbf{u}_r, \mathbf{w}_{t,r})\boldsymbol{\Delta}\mathbf{u} + B_w(\mathbf{x}_r, \mathbf{u}_r, \mathbf{w}_{t,r})\mathbf{w}_t, \tag{5.2.31}$$

where A_c and B_c are system's dynamic matrices evaluated at some given state and control vectors $(\mathbf{x}_r, \mathbf{u}_r)$ and time-dependent exogenous input $\mathbf{w}_{t,r}$. B_w represents the influence of that input, the starter mass flow, on the p_{GG} equation. The $\boldsymbol{\Delta}\mathbf{x}$ and $\boldsymbol{\Delta}\mathbf{u}$ represent the deviations in $\mathbf{x}$ and $\mathbf{u}$ with respect to those points. The complete symbolic expressions of these matrices and their evaluation about the nominal steady state for *Vulcain 1*, as well as the most relevant corresponding transfer functions, are included in Appendix C. In those expressions, it can be observed that 102 elements out of 144 in the A_c matrix are zero. Diagonal terms are non-zero with the exception of the p_{GG} (x_4) state. Regarding B_c, only a diagonal expressing the relations between valve mass flows and their respective sections is non-zero. In B_w, only the element related to p_{GG} is non-zero.

5.3 Models comparison in OL simulations

In order to assess the accuracy of these derived state-space models, OL-simulation comparisons among the different stages of nonlinear modelling (simulator, f_c and f_s) are carried out. In the following, only *Vulcain 1* is used as case study due to the availability of a more consolidated set of internal parameters in contrast to *PROMETHEUS*, still under development. The start-up transient phase, or first four seconds of engine operation, is selected as a comparison time frame. The transient and steady-state behaviour representativeness with respect to simulator results, from the previous Figure 4.3.1, is the main comparison criterion. The same integration method, *ode3* with a fixed time step of $10^{-5}s$, is used in all cases. In Figures 5.3.1 and 5.3.2 the results of f_c and f_s are shown respectively, in terms of normalised chamber pressure (a), mixture ratios (b), normalised mass flows (c), chamber temperatures (d), valve opening angles (e) and normalised shaft rotational speeds (f). The complex model (Figure 5.3.1) is close to the simulator in almost all variables, attaining less than 1% of error in steady-state values. The exception is p_{GG} which is under-quantified by 8%. Transient behaviour is accurately predicted in all states except for mass flows. Indeed, modelling error is specially present in those variables, which can present a mismatch until 15% during the transient after each step of simplification (Table 5.3.1). Transient error in the other states are generally below 12% at each step.

Regarding the simplified NLSS (Figure 5.3.2), the influence of the aforementioned simplifications results in some differences in final values and especially in transient values. Concretely, pressures are somewhat over-quantified during the initial part of transients but reach accurate final values in p_{CC} (under 1% error). Mixture ratios are slightly over-quantified in their final value ($+7\%$), resulting from shifts in mass flows (between -5% and $+5\%$). As explained before, temperature transient behaviour is not well predicted with these models. It is considered as a function of the MR, which are themselves not well defined during the first part of the transient.

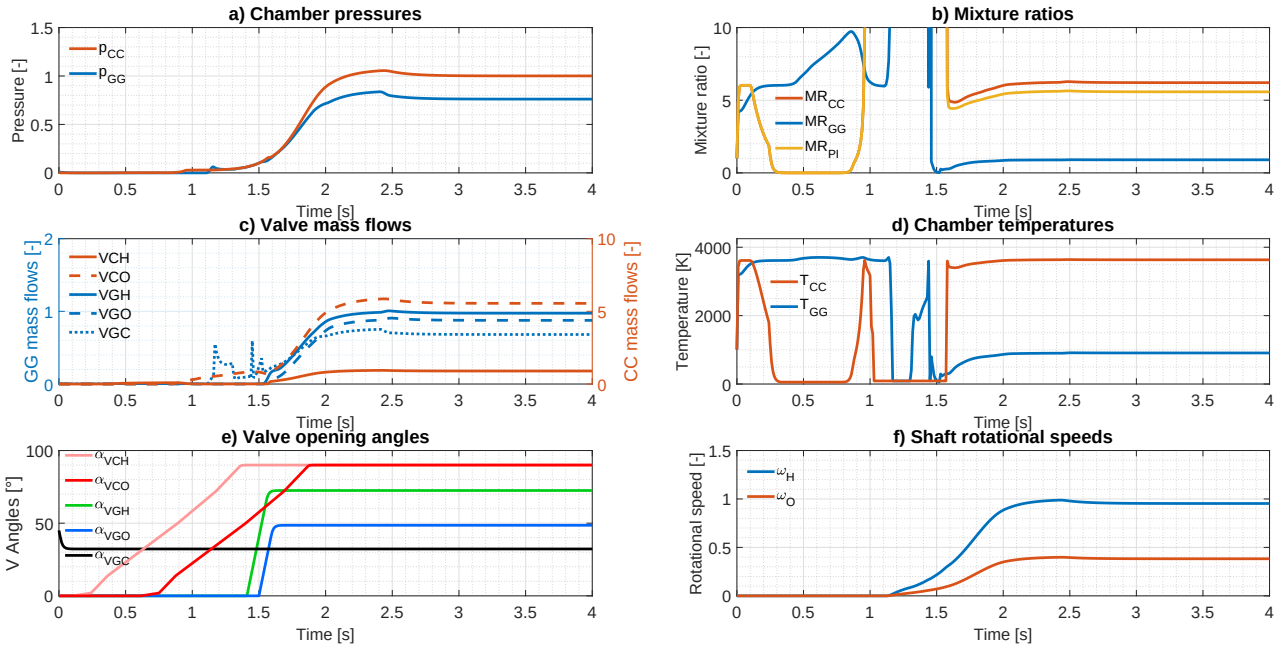


Figure 5.3.1: *Vulcain 1* complex NLSS results at start-up during $4s$

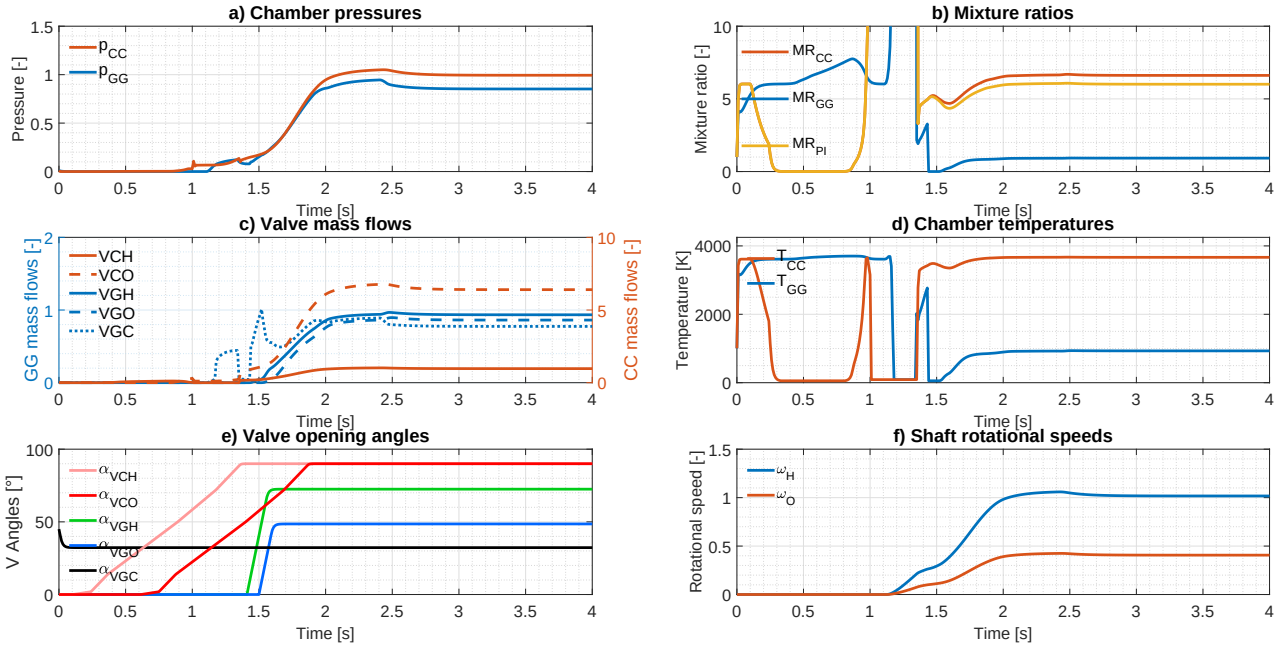


Figure 5.3.2: *Vulcain 1* simplified NLSS results at start-up during $4s$

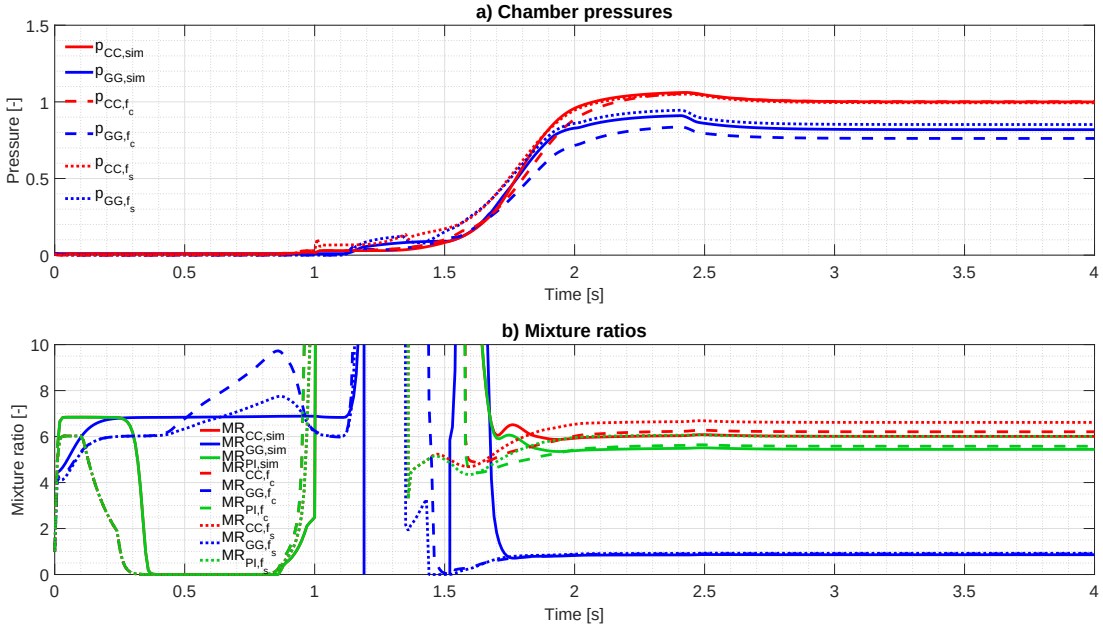


Figure 5.3.3: *Vulcain 1* nonlinear modelling comparison between simulator, f_c and f_s at start-up during $4s$

In order to visualise these modelling discrepancies between the different stages, a graphical comparison of the three steps in terms of chambers pressures and mixture ratios is displayed in Figure 5.3.3. It is noticeable that pressures maximum slopes in Figure 5.3.3.(a) are slightly underrated and that p_{GG} steady-state error is non-negligible. However, p_{CC} is predicted within 1% accuracy and slope changes and time scales are well synchronised. Regarding mixture ratios in (b), the aforementioned steady-state errors in MR_{CC} and MR_{PI} are visible, whereas in MR_{GG} it is very low. MR transient-behaviour trends are more difficult to predict with the implemented simplifications. Even so, the definition of MR prior to the continuous sub-phase is neither clear nor relevant, since some valves are closed and chambers are still not completely physically ignited. Thus, it can be affirmed that these SS models are representative enough for pursuing model-based control of transient chamber pressure and near-steady (end-of-start-up) mixture ratios. A summary of the maximum modelling discrepancies among models along the continuous phase of the start-up is presented in Table 5.3.1.

Table 5.3.1: Maximum modelling errors among models along the continuous phase of the *Vulcain 1* start-up [%]

Variables	ω_H	ω_O	p_{CC}	p_{GG}	$\dot{m}_{VCF}$	$\dot{m}_{VCO}$	$\dot{m}_{VGF}$	$\dot{m}_{VGO}$	$\dot{m}_{VGC}$	MR_{CC}	MR_{GG}	MR_{PI}
Simulator→ f_c error	5.36	4.78	1.61	8.47	11.25	9.3	3.19	-0.87	1.19	-1.76	-3.9	-0.99
$f_c \rightarrow f_s$ error	-6.2	-6.1	0.96	-11.8	-7.74	-15.03	5.55	3.42	-13.18	-6.44	-2.26	-7.59

With regard to the linearised model, its discrepancy with respect to the simplified NLSS, from which it is obtained, is another matter of analysis. This can be formally computed for all $\mathbf{x}$ and $\mathbf{u}$ via the TAYLOR formula with integral reminder for multivariable functions, which expresses the difference

between the original nonlinear system and its k-order TAYLOR series [147]. The general expression of this formula is:

$$f(\mathbf{z}_r+\boldsymbol{\Delta}\mathbf{z}) = f(\mathbf{z}_r)+Df(\mathbf{z}_r)\boldsymbol{\Delta}\mathbf{z}+...+\frac{1}{k!}D^k f(\mathbf{z}_r)\boldsymbol{\Delta}\mathbf{z}^k+\int_0^1 \frac{(1-t)^k}{k!}D^{k+1}f(\mathbf{z}_r+t\boldsymbol{\Delta}\mathbf{z})\boldsymbol{\Delta}\mathbf{z}^{k+1}dt, \quad (5.3.1)$$

where f is a function of class C^{k+1} defined in an open set $U \in \mathbb{R}^n$ to which the equilibrium point $\mathbf{z}_r$ and segment $[\mathbf{z}_r, \mathbf{z}_r + \boldsymbol{\Delta}\mathbf{z}]$ belong, t is an auxiliary variable and the derivatives $D^k f(\mathbf{z}_r)\boldsymbol{\Delta}\mathbf{z}^k$ are defined as:

$$D^k f(\mathbf{z}_r)\boldsymbol{\Delta}\mathbf{z}^k = \sum_{i,...,j=1}^{n} \frac{\partial^k f}{\partial z_i ... \partial z_j}(\mathbf{z}_r)\Delta z_i ... \Delta z_j. \quad (5.3.2)$$

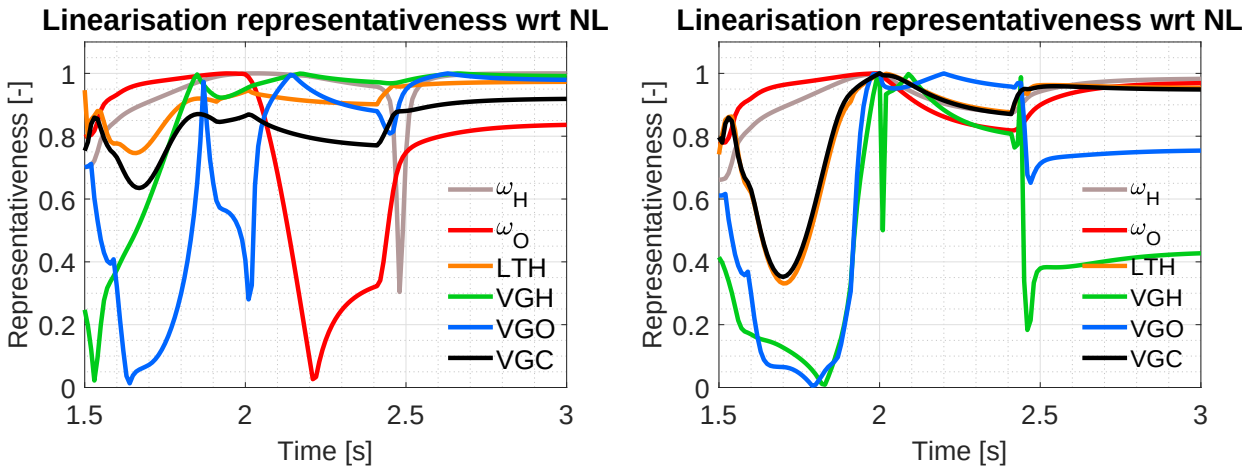
Since the linearisation is of first order, k is equal to 1. Thus, the remainder $\mathbf{R}_1$ (the last term in 5.3.1) expressed for the simplified NLSS $\dot{\mathbf{x}} = f_s(\mathbf{z})$, in which $\mathbf{z} = [\mathbf{x}, \mathbf{u}, \mathbf{w}_t]^T$, yields [147]:

$$\mathbf{R}_1(\mathbf{z}) = \int_0^1 (1-t) \sum_{i,...,j=1}^{n+m+1} \left(\frac{\partial^2 f_s}{\partial z_i ... \partial z_j}(\mathbf{z}_r + t\boldsymbol{\Delta}\mathbf{z})\Delta z_i \Delta z_j \right) dt. \quad (5.3.3)$$

This is computed for all the differential equations (5.2.19) to (5.2.30), yielding cumbersome expressions for rotational speeds and mass flows, but zero for pressures, whose simplified ODEs are linear with respect to $\mathbf{x}$, $\mathbf{u}$ and $\mathbf{w}_t$. In order to evaluate the discrepancies in ω and $\dot{m}$, the obtained expressions are numerically evaluated along the continuous start-up transient in comparison with the nominal steady-state linearisation. The relative weight of the first-order terms with respect to the remainder terms $\mathbf{R}_1$ is depicted in Figure 5.3.4, as an indicator of representativeness:

$$Representativeness = \frac{|A_c(\mathbf{x}_r, \mathbf{u}_r, \mathbf{w}_{t,r})\boldsymbol{\Delta}\mathbf{x} + B_c(\mathbf{x}_r, \mathbf{u}_r, \mathbf{w}_{t,r})\boldsymbol{\Delta}\mathbf{u} + B_w(\mathbf{x}_r, \mathbf{u}_r, \mathbf{w}_{t,r})\mathbf{w}_t|}{|A_c(\mathbf{x}_r, \mathbf{u}_r, \mathbf{w}_{t,r})\boldsymbol{\Delta}\mathbf{x} + B_c(\mathbf{x}_r, \mathbf{u}_r, \mathbf{w}_{t,r})\boldsymbol{\Delta}\mathbf{u} + B_w(\mathbf{x}_r, \mathbf{u}_r, \mathbf{w}_{t,r})\mathbf{w}_t| + |\mathbf{R}_1(\mathbf{x}, \mathbf{u}, \mathbf{w}_t)|} \quad (5.3.4)$$

The evaluation of the $\mathbf{R}_1$ expressions for $\dot{m}_{VCF}$ and $\dot{m}_{VCO}$ is not well defined during transients and hence not depicted. Results for the nominal steady state (Figure 5.3.4a) indicate that static errors are within 17% while transient errors can become very elevated locally. Indeed, the depicted ODEs are highly nonlinear. If a linearisation at the middle of the transient is performed (Figure 5.3.4b), better representativeness is attained in that vicinity (under 10% at $t = 2s$), as logical.



(a) About nominal steady state

(b) About $t = 2s$, mid-transient

Figure 5.3.4: *Vulcain 1* linearisation representativeness with respect to f_s during the continuous phase of start-up

5.4 Models analysis for control purposes

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 step and frequency response, stability, measurability, observability, controllability and sensitivity to parameter variations.
The first aspect to introduce is the fact that the derived nonlinear state-space models do not present an analytical solution to their equilibrium point $\dot{\mathbf{x}} = \mathbf{0}$ even in their simplified form. Knowledge on the desired equilibrium point on some states is used to numerically reconstruct a full-state equilibrium vector in Chapter 6, as a part of the controller.

5.4.1 Step and frequency-domain response

In order to assess the response of the system in the time and frequency domains, the step response and BODE plots of the steady-state linearised model are depicted in Figures 5.4.1 (step), 8.2.1 (BODE magnitude) and 8.2.2 (BODE phase). The latter two figures are included in Appendix D.

Here, the different influences of each control input on each state can be observed. It appears that the main contribution to states variation comes from the three last inputs, corresponding to the GG valves VGF, VGO and VGC. These are hence the main control inputs to adjust for shifting the system, as shown in Chapter 4. Regarding the frequency response, most of the 60 input-output relations present cut-off frequencies between 1 and $10Hz$. Some exceptions, arriving to 100 or $1000Hz$, mainly concern the VGC valve and mass flows, which can fluctuate at faster rates.

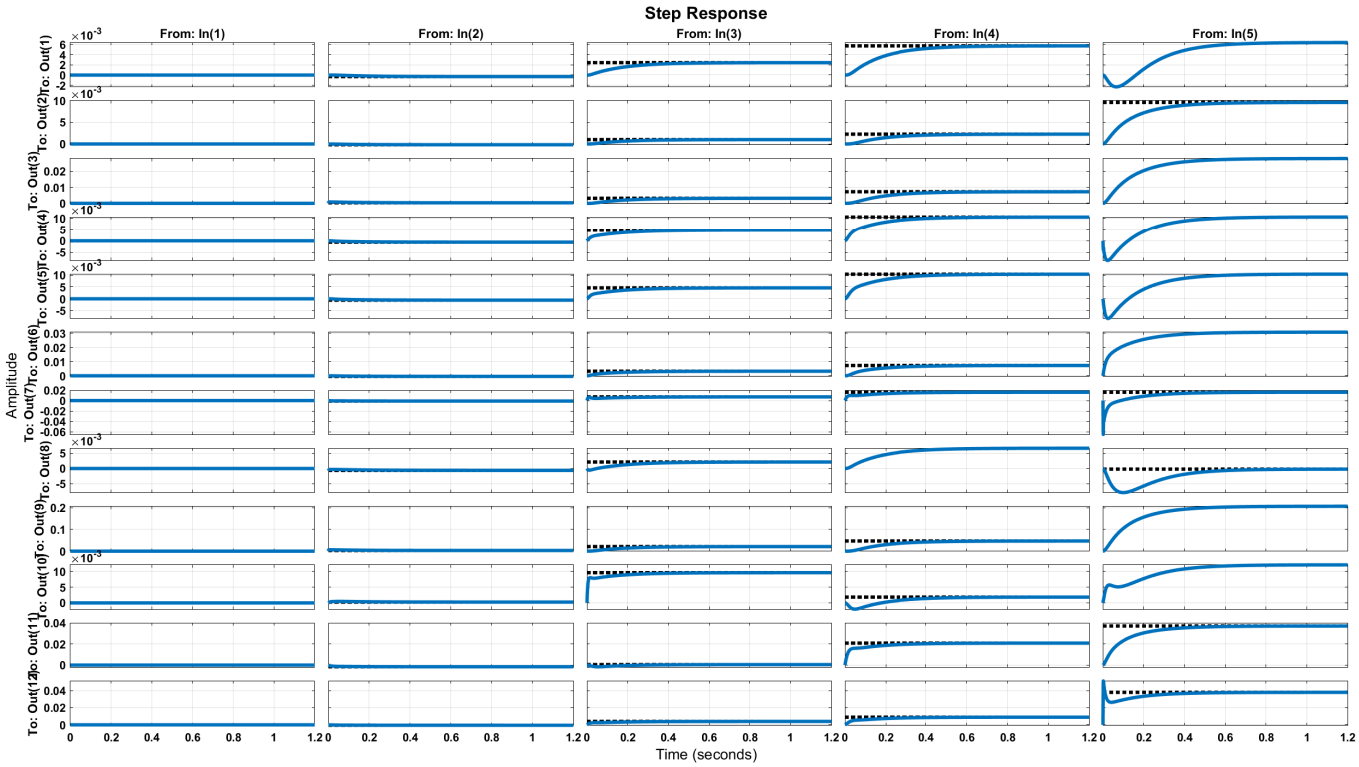


Figure 5.4.1: *Vulcain 1* multivariable step response at steady-state

5.4.2 Stability

Stability is an inherent property of these GG-LPRE models, which describe a system that self-compensates disturbance in the flow cycle due to its high coupling [155]. The possible destabilising high-frequency phenomena in LPREs, such as combustion instabilities, are not modelled in these SS models for the sake of simplicity (Section 4.2.1). Hence, when analysing linearised models about transient trajectories, all eigenvalues present strictly negative real parts. This holds for the different stages, as shown by the poles of the linear model evaluated at the nominal steady-state in the *Vulcain 1* engine:

$$10^4 \times [-9.3555 + 0.0000i, \quad -0.3747 + 0.0000i, \quad -0.1961 + 0.1325i, \quad -0.1961 - 0.1325i,$$
$$-0.0553 + 0.1906i, \quad -0.0553 - 0.1906i, \quad -0.0242 + 0.0000i, \quad -0.0006 + 0.0000i,$$
$$-0.0019 + 0.0000i, \quad -0.0063 + 0.0028i, \quad -0.0063 - 0.0028i, \quad -0.0071 + 0.0000i].$$

Before the steady state, at the start of the continuous sub-phase of the start-up transient, the poles also indicate stability:

$$10^4 \times [-6.3349 + 0.0000i, \quad -0.2754 + 0.0000i, \quad -0.1079 + 0.1277i, \quad -0.1079 - 0.1277i,$$
$$-0.0574 + 0.0000i, \quad -0.0277 + 0.0285i, \quad -0.0277 - 0.0285i, \quad -0.0184 + 0.0035i,$$
$$-0.0184 - 0.0035i, \quad -0.0002 + 0.0000i, \quad -0.0009 + 0.0001i, \quad -0.0009 - 0.0001i].$$

However, at the very initial point of the start-up transient, the closed position of valves leads to control inputs equal to zero. Since controls are present in the denominators of the previous equations (5.2.26) to (5.2.30), indeterminacy is obtained in the poles computation. If a small lower bound $\epsilon = 10^{-12}$ is assumed in the non-dimensional inputs, the following poles are obtained:

$$10^4 \times [-1.3075 + 6451.8965i, \quad -1.3075 - 6451.8965i, \quad -439.6956 + 476.317i, \quad -439.6956 - 476.317i,$$
$$-12.8691 + 897.0535i, \quad -12.8691 - 897.0535i, \quad -24.7533 + 0i, \quad -0.016 + 0.0182i,$$
$$-0.01598 - 0.01823i, \quad -0.02154 + 0i, \quad -7.6475 \times 10^{-7} + 0i, \quad -6.8105 \times 10^{-6} + 0i].$$

In this case, the system is marginally stable since some poles are very close to zero. Moreover, oscillations can become elevated accounting for the large complex parts of half of the poles. Nevertheless, the system does not remain in that state in the subsequent instants. Dominant poles correspond in all cases to valve mass flows.

Remark:
Formal stability proofs on the nonlinear system via the LYAPUNOV theory [62] (introduced in Section 3.3.2) have not been accomplished in this thesis. There is not a standard methodology for constructing such a function for a given system, but several methods and strategies have been proposed [62]. Even if a standard quadratic LYAPUNOV function is defined ($V(\mathbf{x}) = \mathbf{\Delta x}^T P \mathbf{\Delta x}$, P being a positive-definite matrix), no formal proof of $\dot{V}(\mathbf{x}) \leq 0$ has been derived due to the high amount of nonlinear and coupled terms present in the twelve-state models derived in this chapter. For these analyses, the system has been shifted so that its nominal equilibrium point corresponds to the origin. KHALIL vanishing-perturbation theory [62], in which a part of the nonlinear terms in the differential equations are considered as perturbations $g(\mathbf{x}, \mathbf{u})$ with respect to the rest of so-called nominal terms $f(\mathbf{x}, \mathbf{u})$ such that

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) + g(\mathbf{x}, \mathbf{u}), \tag{5.4.1}$$

has been employed for attempting to find a suitable $V(\mathbf{x})$. Different choices can be made for f and g. Here, the positive terms are considered as perturbations and the stabilising negative terms should counteract them and drive the state to the origin so as to verify the local-stability criteria $V(\mathbf{x}) > 0$ and $\dot{V}(\mathbf{x}) \leq 0$ ($\dot{V}(\mathbf{x}) < 0$ for global asymptotic stability). However, neither that method nor the generalised KRASOVSKII theorem, in which $V(\mathbf{x})$ is defined as $V(\mathbf{x}) = \dot{\mathbf{x}}^T P \dot{\mathbf{x}}$, have delivered conclusive formal results. Only some numerical evaluations have been obtained when combining both approaches. In other words, a $V(\mathbf{x}) = \dot{\mathbf{x}}^T P \dot{\mathbf{x}}$ has been computed by considering only the stabilising negative terms

in (5.2.19)-(5.2.30). The matrix P is computed in the following manner by expanding the generalised KRASOVSKII theorem [62], in which control and exogenous-input dependencies are omitted:

$$
\begin{aligned}
V(\mathbf{x}) &= f^T(\mathbf{x}) P f(\mathbf{x}), \\
\dot{V}(\mathbf{x}) &= f^T(\mathbf{x}) P \dot{f}(\mathbf{x}) + \dot{f}^T(\mathbf{x}) P f(\mathbf{x}) = \\
&= f^T(\mathbf{x}) P \left(\frac{\partial f}{\partial \mathbf{x}}\right)^T \dot{\mathbf{x}} + \left[\left(\frac{\partial f}{\partial \mathbf{x}}\right)^T \dot{\mathbf{x}}\right]^T P f(\mathbf{x}), \quad G \equiv \frac{\partial f}{\partial \mathbf{x}} \rightarrow \\
\dot{V}(\mathbf{x}) &= f^T(\mathbf{x}) P G^T f(\mathbf{x}) + f^T(\mathbf{x}) G P f(\mathbf{x}) = \\
&= f^T(\mathbf{x})(P G^T + G P + Q - Q) f(\mathbf{x}) = \\
&= f^T(\mathbf{x})(P G^T + G P + Q) f(\mathbf{x}) - f^T(\mathbf{x}) Q f(\mathbf{x}),
\end{aligned}
\tag{5.4.2}
$$

where P and Q are positive-definite matrices. The second term in the last line is negative definite due to the choice of Q. Thus, a P that renders $(PG^T + GP + Q)$ negative semi-definite is to be found to prove $\dot{V}(\mathbf{x}) \leq 0$ in some neighbourhood. This is numerically evaluated with a varying $P(\mathbf{x})$

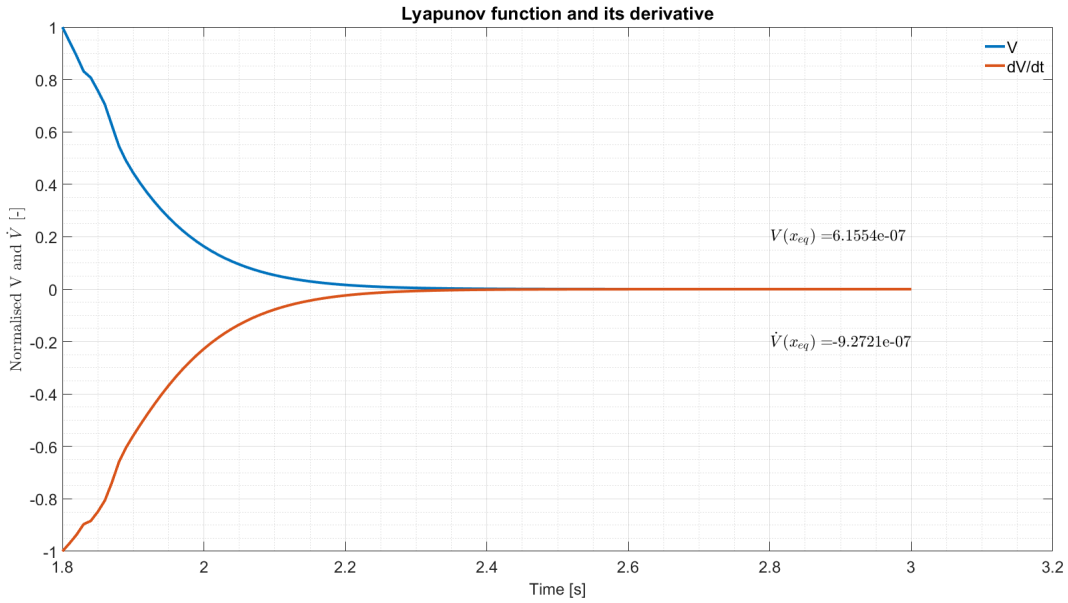


Figure 5.4.2: Numerical evaluations of KRASOVSKII-based LYAPUNOV function during start-up

computed at each simulation time step via *MATLAB®*'s *lyap* utility as a function of the gradient G, with Q taken as an identity matrix. Stability is numerically confirmed at the final phase of the nominal start-up transient, as shown in Figure 5.4.2. The evaluation of the proposed V provides positive, decreasing values. Before that time instant $(1.8s)$, LYAPUNOV stability conditions are not verified in nominal start-up simulations.

5.4.3 Measurability and observability

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 LPREs, but estimated through pressure, temperature and

volumetric flow measurements. This estimation process is assumed perfect in this thesis, but consists in a future block to add to the loop. Some previous works in ArianeGroup on mass-flow estimation have been summarised in Section 2.3.2. In fact, during the discrete part of the transient, some observability issues can appear due to the low volumetric flows characteristic of that phase [115]. This, apart from other issues highlighted in Chapter 7, hinders the realisation of CL control during the discrete sub-phase of transients.
During the fully continuous phases, the linearised models derived in this chapter locally fulfil the observability criteria for time-invariant systems. This holds even if no mass-flow measurements are given (6 out of 12 states in *Vulcain 1*), provided that control inputs are known. Thus, linear mass-flow estimators could be developed by using these models. In this case, the matrix C_c relating states to measured outputs $\mathbf{y}$ for *Vulcain 1* would be:

$$\mathbf{y} = \begin{bmatrix} \omega_H \\ \omega_O \\ p_{CC} \\ p_{GG} \\ p_{LTH} \\ p_{VGC} \end{bmatrix} = C_c\mathbf{x} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{x}. \tag{5.4.3}$$

5.4.4 Controllability

Controllability criteria can be verified after linearising f_s about the whole continuous start-up trajectory, considering the system as locally time invariant. In other words, all states can be controlled via valves during the entire continuous sub-phase, once all events have been activated. Even though the control inputs only appear directly in the mass-flow differential equations (5.2.26) to (5.2.30), all states are indirectly accessible from control through coupling. It is interesting to detail this coupling, which can be easily determined by analysing the terms in linearised models, where there are many zero elements. The combustion-chamber pressure p_{CC}, the main state defining thrust and hence the main variable to control, presents a clear relation to the injected mass flows in the chamber, $\dot{m}_{VCF}$ and $\dot{m}_{VCO}$. Nevertheless, the adjustment of CC injection valves during fully continuous phases is not convenient for system's operation shift, as analysed in Section 4.3. Therefore, other ways of accessing to those mass flows have to be found. Indeed, they are also highly influenced by rotational speeds ω, which are affected by pre-turbine pressures p_{LTH}/p_{LT} (and eventually p_{VGC}). These are mostly determined by their inlet mass flows, these being highly influenced by the pressure in the GG p_{GG}, the core of the system. Obviously, this pressure is primarily altered through its injected mass flows $\dot{m}_{VGF}$ and $\dot{m}_{VGO}$, which can be modified by adjusting GG injection valves. In fact, controllability criteria are met even if only the GG valves, VGF, VGO and VGC are considered as control inputs. This main coupling path in *Vulcain 1* is summarised in (5.4.4) for the sake of visualisation:

$$\{A_{VGF}, A_{VGO}\} \rightarrow \{\dot{m}_{VGF}, \dot{m}_{VGO}\} \rightarrow \{p_{GG}\} \rightarrow \{\dot{m}_{LTH}, \dot{m}_{VGC}\} \rightarrow \{p_{LTH}, p_{VGC}\} \rightarrow \\ \rightarrow \{\omega_H, \omega_O\} \rightarrow \{\dot{m}_{VCF}, \dot{m}_{VCO}\} \rightarrow \{p_{CC}\}. \tag{5.4.4}$$

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 (as shown in Chapter 4), which open at the end of the sequence, after ignitions and starter activation.

5.4.5 Sensitivity to parameters

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 engine parameters used in the differential equations are estimated from test data or tuned in simulations. Parameters in this thesis have been provided by ArianeGroup and CNES for the nominal case. In order to analyse the effect of their variations on the behaviour of the engine, a series of sensitivity analyses have been performed. The list of considered varying engine parameters is summarised in Table 5.4.1. There can be different sources of deviation. Some parameters might vary during operation or between different engine runs. Others are estimated for modelling and hence are not 100% certain. Inter-engine material discrepancy can also be a source of variation.

Parameter	**Definition**	**Considered variation range** [%]	**Variation source**
p_{PI}	Tanks pressures, coupled with inlet densities and temperatures	± 10	Oscillation during/between operations
$ResLC$	LC lines fluidic resistances	± 10	Modelling uncertainty
$ResI$	Injectors fluidic resistances	± 10	Modelling uncertainty
$ResCR$	Cooling-circuit fluidic resistance	± 10	Modelling uncertainty
$IneCR$	Cooling-circuit fluidic inertia	± 10	Modelling uncertainty
η_{CC}	CC combustion efficiency	± 5	Modelling uncertainty, oscillation during/between operations
η_{GG}	GG combustion efficiency	± 5	Modelling uncertainty, oscillation during/between operations
$ResLTH$	LTH line fluidic resistance	± 10	Modelling uncertainty
$IneLTH$	LTH line fluidic inertia	± 10	Modelling uncertainty
V_{cav}	Pre-turbine cavities volume	± 10	Modelling uncertainty
$AthCC$	CC throat section	± 1	Inter-engine discrepancy, alteration during/between operations.
$AthT$	Turbine inlet sections	± 1	Inter-engine discrepancy, alteration during/between operations.
$AthLE$	Turbine outlet sections	± 1	Inter-engine discrepancy, alteration during/between operations.

Table 5.4.1: List of *Vulcain 1* engine parameters considered in sensitivity analysis

Indeed, tanks pressure and combustion efficiencies might oscillate during flight or between engine executions. Fluidic resistances, inertias and combustion efficiencies cannot be directly determined;

they have to be estimated from tests or simulations. Orifices in the cycle may present varying effective hydraulic sections during operation and/or inter-engine discrepancies.
In order to study the influence of each individual parameter on the transient behaviour of the engine, the SOBOL sensitivity analysis [149] has been applied. This method, introduced in Section 3.4, 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. In fact, a criterion which determines the analysis-method selection is the capability to calculate local or global sensitivity indices. The Jacobian of system's differential equations with respect to parameters can indicate local sensitivities, since they can only provide numerical values when evaluating at a concrete state and time instant. In this sense, the SOBOL method provides global indices since it covers the whole behavioural spectrum due to its statistical approach.
Here, it is of interest to study the variations in terms of a global performance criterion J_{perf}, defined as the weighted addition of several simulated indicators:

$$J_{perf} = |err_{pCC}|+|err_{MRCC}|+|err_{MRGG}|+|err_{MRPI}|+0.001 \cdot |A_{pCC}|+0.01 \cdot |os_{pCC}|, \tag{5.4.5}$$

where err are static steady-state errors in the main quantities that determine engine operation (contained in the $\mathbf{x}_z$ vector defined in Chapter 6), os is overshoot (in %) and A_{pCC} represents the difference in surface between the perturbed p_{CC} start-up transient curve and the nominal desired one (in %). Thus, the lower the criterion the better the performance. The sampling of parameters variations according to Table 5.4.1 has been carried out via LHS, recommended for calculating the SOBOL global indices [172, 177]. In this fashion, a set of 1000 parameter-variation combinations, with their corresponding output J_{perf} (computed with the simulator model from Chapter 4), is provided to a Kriging-based SOBOL algorithm based on [58]. This algorithm creates a Kriging model (introduced in Section 3.4) based on the provided input-output data, which avoids the execution of further costly simulations. After running 1 million Monte Carlo evaluations of that less expensive model, the most influential parameters, related to the higher SOBOL global indices, can be identified. These are the CC throat section ($AthCC$), the turbine inlet sections ($AthT$) and tanks pressures (p_{PI}), whose indices present an order of magnitude of 0.1 (normalised with respect to one). Injectors fluidic resistances ($ResI$, concretely $ResOIC$ and $ResOIG$), the cooling-circuit fluidic resistance ($ResCR$) and GG combustion efficiency η_{GG} indices are one order of magnitude lower (0.01). This shortlist of parameters, to which the model is more sensitive, is taken into account in the control algorithm (Chapters 6 and 7). The effects of variations in the rest of parameters can be neglected according to the sensitivity analysis. All these results are graphically summarised in Figure 5.4.3.

These results are physically coherent. Orifice sections, either throats or inlets, are indeed very precisely tuned in cycles according to engine design. Even little variations in their values can trigger relevant mass flow deviations, altering the whole operation. Tanks pressures are crucial boundary conditions to the system, determining the maximum attainable pressures in chambers. GG combustion efficiency variations entail repercussions on the whole system, the GG being its core. This study also highlights the importance of well estimating fluidic resistances, which influence cycle dynamics in locations with significant pressure drop like injectors.

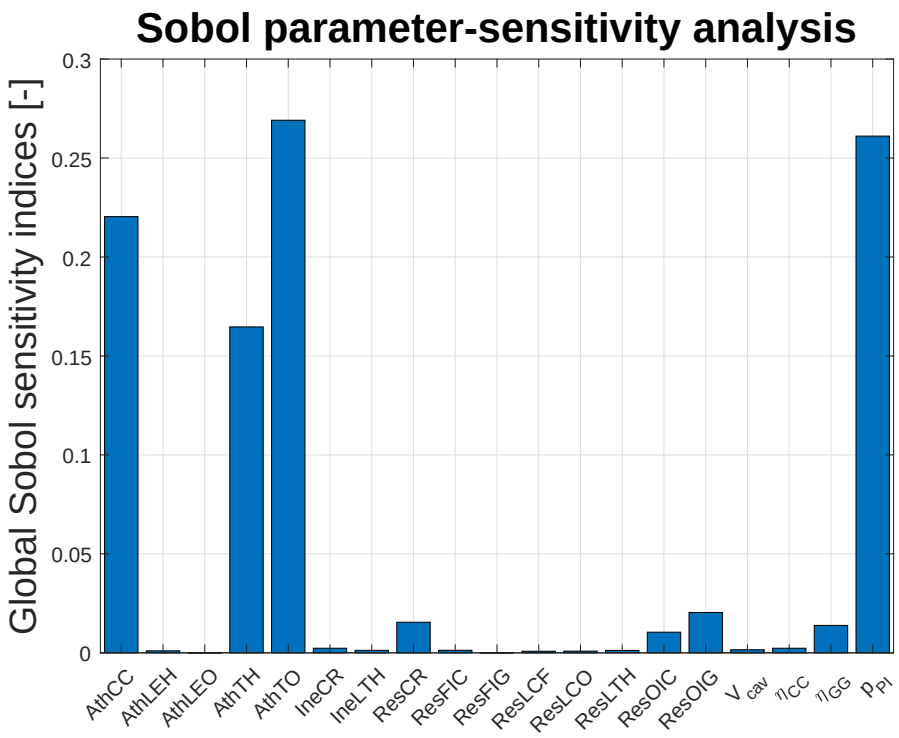


Figure 5.4.3: SOBOL global sensitivity indices of *Vulcain 1* varying parameters

5.4.6 Worst-case parameter-varying scenario search

Building on the previous sensitivity analysis, it is of interest to find extreme parameter-varying cases that would degrade the performance criterion (5.4.5) to a higher extent. This information can be later used in robust control design. From the practical experience of LPRE system behaviour in simulations and in flight, it is known that p_{PI} can more probably vary in reality, especially in the new reusability scenarios. Thus, a higher priority is put on them in the following. Furthermore, a worst-case scenario of variations of the previous shortlist of influential parameters has also been computed. A Kriging-based black-box minimisation based on [98] of the negative performance criterion (5.4.5), as a function of the possible combinations of parameters variations, has been carried out:

$$\hat{\mathbf{p}} = \arg\min_{\mathbf{p}\in\mathbb{P}} -J_{perf}(\mathbf{p}), \tag{5.4.6}$$

where $\mathbf{p}$ is the parameters vector assembling the aforementioned shortlist, $\hat{\mathbf{p}}$ is the resulting parameters combination minimising the cost and $\mathbb{P}$ are the allowed intervals defined in Table 5.4.1. It is a black-box problem because the indirect relation $J_{perf}(\mathbf{p})$ is obtained from simulations in a first step. This optimisation is performed via Efficient Global Optimisation (EGO) schemes, which use a Kriging model (introduced in Section 3.4) of the criterion built after a certain amount of simulations. Preliminary samplings in this subsection are all also performed via LHS, which provides a set of randomly scattered input data to be simulated. Parameters are continuously varied within the intervals in Table 5.4.1 and the EGO algorithm, which accounts for the statistical information provided by Kriging (variance of prediction error), reduces the cost via the maximisation of the Expected Improvement (EI). EI expresses the interest of a supplementary evaluation of the inputs depending on previous results

and on the Kriging prediction of the mean and variance. Maximising EI implies a trade-off between local search and the exploration of unknown areas, which makes EI suitable for global optimisation. The value of EI decreases progressively until a threshold of 10^{-8} in this case. In this manner, the worst combination of deviations leading to the poorest performance can be estimated. The outcome of this search, which practically attains the bounds of intervals, is presented in Table 5.4.2. It corresponds to a degradation of 106% in J_{perf} with respect to the nominal OL.

Parameter	**Definition**	**Variation direction in considered interval (+/-)**
p_{PI}	Tanks pressures, coupled with inlet densities and temperatures	-
$ResOI$	Injectors fluidic resistances (of oxidiser lines)	-
$ResCR$	Cooling-circuit fluidic resistance	-
η_{GG}	GG combustion efficiency	-
$AthCC$	CC throat section	+
$AthTH$	Fuel-turbine inlet section	+
$AthTO$	Oxidiser-turbine inlet section	-

Table 5.4.2: Worst-case combination of parameter variations in *Vulcain 1*

Some of these results can be explained according to engine physics. It is indeed more difficult for the system to attain nominal values if tank pressures are lower than expected. Lower flows and hence cavity pressures are given. The reduction in efficiency of the GG is also a common factor leading to poorer performance. The section of CC throat, if augmented, physically leads to a slower increase in CC pressure. Therefore, this worst-case scenario will be used in the controller design in Chapters 6 and 7.

5.5 Summary

The main goal of this chapter has been to derive global mathematical expressions of the GG-cycle thermodynamic behaviour in the form of state-space models. A translation from the simulator in the previous chapter has been performed via a symbolic causal interconnection of all components. A mainly continuous modelling approach has been selected, inspired by the MLD hybrid-modelling paradigm [10, 91], in which the discrete inputs of the system manage certain terms in the differential equations. Several steps of simplifications have been carried out from the initially obtained complex nonlinear state-space model in order to derive a more tractable expression of the model for control purposes. These GG-LPRE state-space system representations for control purposes with time-varying coefficients and discrete inputs consist in one of the main contributions of this thesis, as reviewed in Section 2.3. The *Vulcain 1* engine has served as the main case study, while *PROMETHEUS* has also been modelled. The simplified nonlinear model has then been linearised, and a comprehensive analysis of its response and characteristics has been elaborated. The effects of simplifications, entailing modelling error, and the properties of stability, controllability, measurability, observability and sensitivity to parameter variations have been analysed, confirming the suitability of the obtained state-space models for the subsequent derivation of control laws. Models are stable and controllable

during the continuous sub-phase of the start-up transient and full-state measurements are assumed to be available. The most influential parameters and the worst-case scenario of parametric variations have been identified. All this information will be used in the development of control algorithms in Chapters 6 and 7. The main contributions from this chapter and from Chapter 4 are included in [118]. The conference paper [120] contains a summary of models analysis.

6 Synthesis of MPC controllers for end-state tracking in continuous GG-LPRE transients

This chapter is devoted to the synthesis of model-based controllers for GG-LPREs making use of the state-space models derived in the previous Chapter 5. The goal in this first controller-design chapter is to drive the system towards a pre-calculated terminal state, without imposing a concrete trajectory but respecting a set of hard constraints. That terminal state is related to the desired set point of the GG engine in terms of combustion-chamber pressure and mixture ratios in chambers and at pump inlets. The controller aims at reaching that final reference by adjusting the available continuous control inputs, which are valve sections. Thus, the objective is to control the engine along fully continuous transient phases, such as throttling scenarios. In other words, it is aimed here at tracking a change of operating point while verifying state and control constraints. The robustness of the solution against internal parametric variations is also required. All these goals are to be attained at acceptable computational times.
Several incremental steps have been performed in terms of controller design, always following the same control-loop structure, defined in Section 6.1. In the first place, simple linear controllers dealing with steady-state tracking have been developed. The use of PID or LQR techniques (Section 6.3) did not completely fulfil the control goal, since constraints are not taken into account in the design, and hence not respected in some scenarios. This reason, together with the need for a multivariable tracking on a complex system, motivated the choice of MPC as a suitable control method. The multivariable tracking reference is computed off-line as a function of launcher needs, as explained in Section 6.2. MPC controllers for end-state tracking (E.MPC) based on the linearised models have then been synthesised, including the required set of hard state and control constraints. Further justifications for this method selection and design characteristics, including robustness to parametric variations, are included in Section 6.4. The corresponding results when applied to throttling scenarios and continuous start-up phases are presented and analysed in 6.5. These developments are included in the conference papers [119] and [120].

6.1 Control-loop structure

The main control diagram considered in this thesis is depicted in Figure 6.1.1. The designed controllers, either MPC or the here-called *reference* ones (PID, LQR, pole placement), receive a full state and control reference ($\mathbf{x}_r$, $\mathbf{u}_r$) from an off-line preprocessor block and compute the corresponding control on-line. The remaining elements in that diagram are the following. To the right there is the simulation of the rocket engine, performed via the integration of the simulator developed in Chapter 4. Its initial conditions, when simulating the continuous part of the start-up transient, are taken as the outcome of the nominal discrete sub-phase ($t = 1.5s$ in *Vulcain 1*). Control algorithms are tested on that model, the most complex one considered in this thesis, for the sake of plant representativeness. The inputs of the state-space model used for control are valve sections $\mathbf{u}_c$ (5.2.3), as explained in Chapter 5.

However, the actuators mechanical model (internal valve actuators) requires an input in terms of opening angles α. That is the reason why there is a conversion block, characterised by the static and monotone nonlinear functions defined in Section 4.2.2. The controller provides valve sections that are then translated into angles. The cause for considering valve actuators as a separate entity is the fact that they represent an internal servo-loop, in which the angular position of the valve is tuned by means of 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. The exogenous sequence influence modifies discrete control inputs according to the current stage of the transient phase. But during the totally continuous part of the start-up, considered in this control approach, that influence is not relevant because the discrete inputs are always activated.

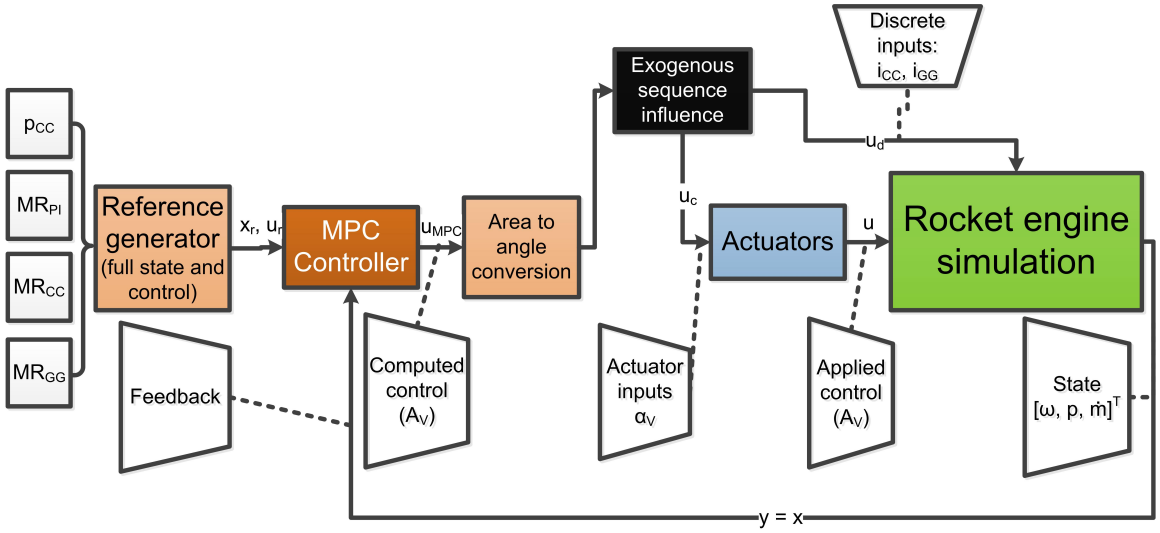


Figure 6.1.1: Control-loop diagram for end-state tracking

6.2 Preprocessing: reference generator

The preprocessing block serves as an off-line reference generator for multivariable controllers. Indeed, the set of steady-state reference commands derived from launcher needs and considered in this thesis, a total of four, concern thrust or p_{CC} (interrelated according to equations (2.1.1) to (2.1.3)) and mixture ratios in chambers and at PI. As said before, $\mathbf{x}$ presents twelve states in the state-space models developed in Chapter 5 for the *Vulcain 1* case (5.2.2) and nine for *PROMETHEUS* (5.2.16). Therefore, the indicated command is not sufficient to provide a complete equilibrium point to an engine controller based on these models. Hence, a way of restoring full state and control reference vectors from those data is necessary. In addition, without these $(\mathbf{x}_r, \mathbf{u}_r)$, the posterior controllers would not attain the tracking goal with high precision. This is mainly due to the fact that f_s is linearised about $(\mathbf{x}_r, \mathbf{u}_r)$, as explained in Section 6.4.

The four reference inputs are $p_{CC,r}$, $MR_{PI,r}$, $MR_{CC,r}$ and $MR_{GG,r}$. Moreover, the last three, in contrast to p_{CC}, do not directly correspond to states in the model. They establish relations between $\tilde{\dot{m}}_i$. In the following, the presence of a tilde ($\tilde{}$) on top of a quantity means that it is dimensional and its absence means the contrary. In the first place, thanks to the selected pressure $p_{CC,r}$ and $MR_{CC,r}$, the

choked-flow static equation (4.2.52) can provide injected mass flows into the CC $\dot{m}_{VCF,r}$ and $\dot{m}_{VCO,r}$, seeing that the characteristic velocity C^* depends on MR. Then, the rest of states at equilibrium, concerning rotational speeds, pressures in GG and pre-turbine cavities and the remaining mass flows, as well as the control inputs $\mathbf{u}_r$ making that state possible, are to be computed. Discrete control inputs $\mathbf{u}_d$ are considered active since a running steady-state is sought. The first two valve sections in $\mathbf{u}_c$, corresponding to CC valves, are chosen to be opened at their nominal position (100%). This choice stems from the throttling analysis in Section 4.3, in which it was pointed out that the throttability of the engine was directly related to the GG valves, which can govern the whole state. The tuning of CC valves is not convenient in this kind of engines, even if off-nominal operating points are targeted. All the remaining equilibrium variables are determined by solving the following overdetermined system of nonlinear equations:

$$\begin{cases} \dot{\mathbf{x}} = f_c(\mathbf{x}_r, \mathbf{u}_r) = \mathbf{0} \quad \setminus (\dot{p}_{CC} = 0) \\ \frac{\tilde{\dot{m}}_{VCO} + \tilde{\dot{m}}_{VGO}}{\dot{m}_{VCF} + \dot{m}_{VGF}} = MR_{PI,r} \\ \frac{\dot{m}_{VGO}}{\dot{m}_{VGF}} = MR_{GG,r} \\ \tilde{\dot{m}}_{VGF} + \tilde{\dot{m}}_{VGO} = \tilde{\dot{m}}_{LT(H)} + \tilde{\dot{m}}_{VGC}. \end{cases} \tag{6.2.1}$$

The first set of equations forces the ODEs of the complex NLSS (5.2.4)-(5.2.15) in *Vulcain 1* to be at equilibrium, the second and the third ones determine the indicated MR and the last one enforces the equilibrium of $\sum \tilde{\dot{m}}_i$ in the GG. The ODE for $\dot{p}_{CC}$ (5.2.6) is removed since it is completely dependent on the reference inputs, not providing any additional information. This amount of equations, $n+2$ (14), is necessary to retrieve a physical solution of the $n-3$ (9) unknown states and $m-2$ (3) unknown continuous inputs. The resolution is performed numerically via nonlinear least squares due to the unavailability of an analytic solution of the system, of either f_c or f_s. The complex model has been chosen to increase the accuracy with respect to the simulator and hence the real engine. This computation lasts around 2-$3s$ by means of the *MATLAB*® *lsqnonlin* function with a tolerance of 10^{-10}. The reference values for $p_{CC,r}$ and $MR_{CC,r}$ obtained in the resulting $\mathbf{x}_r$ are exact via this procedure. However, there is some relative error in the result with respect to the input $MR_{PI,r}$ and $MR_{GG,r}$. Concretely, in $MR_{PI,r}$ it is about 5% at all points and in $MR_{GG,r}$ it is 0.3% in the nominal point and about 20% in off-nominal ones due to its normally low value, 1.

In fact, the $\mathbf{u}_r$ computed by means of the aforementioned method can be regarded as an OL command. Especially in the case of off-nominal points, this calculation provides a reference OL control in the event that closed-loop would not be available in the engine. In flight, it is usually the case that ad-hoc parameters are tuned to achieve an alternative thrust level. Thus, this method can be useful for performing a 120%-thrust start-up for instance, as depicted in Figure 6.2.1, even though robustness is not guaranteed at all. Performance indicators of these OL controllers in different scenarios are included in the comparison Tables 6.5.1, 6.5.3 and 6.5.4.

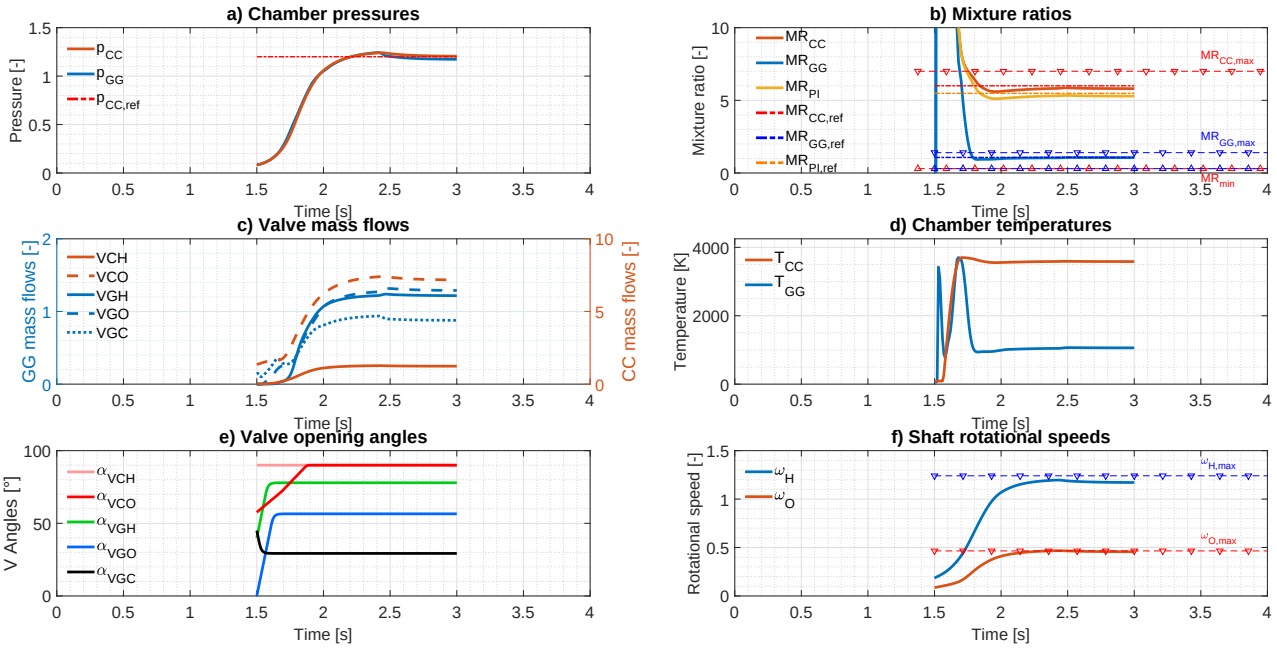


Figure 6.2.1: OL-control results for a 120% continuous start-up in *Vulcain 1*

6.3 Synthesis of reference linear controllers: PID and LQR

The first steps towards a controller implementation concerned the synthesis of simple reference linear controllers, making use of the linearised model of the engine about the desired pre-computed end point $(\mathbf{x}_r, \mathbf{u}_r)$. The analysis of these models has been presented in Section 5.4. The goal of the elaboration of these first controllers was to obtain baseline results for a subsequent comparison with more advanced methods like MPC.

Firstly, a set of PID controllers were developed so as to track a constant reference in p_{CC} by tuning the GG valves. This classical technique has been identified in Chapter 2 as the most used in the literature of LPREs control. However, seeing that LPREs are highly multivariable systems, one would have to decouple them into numerous SISO and design the corresponding PID controllers. Here, this has been performed for p_{CC}, the main variable to track. Control contributions from GG injection valves (VGF and VGO) are considered, which leads to two different PID controllers. The corresponding transfer functions are included in Appendix C. The impact of VGC can be mainly observed in MR; thus, it is not used here to control CC pressure. Actuators models (second-order) have been appended to each of the control inputs, so as to consider their dynamics in the PID design. By using the *MATLAB® pidtune* function (with a reference-tracking design focus) on the resulting terminal state-space matrices and simulating the obtained controllers to attain a nominal thrust (100%) after start-up, the static error in p_{CC} is of the order of 0.001% (Figure 6.3.1). Nevertheless, in Figure 6.3.1, it can also be seen that due to the high overshoot, constraints on rotational speeds and mixture ratios, defined in detail later in this chapter (Section 6.4), are highly violated (triangular symbols). In addition, considerable static error in the MR references is present, as expected. Actuators are also saturated during most of the transient and settling times are long.

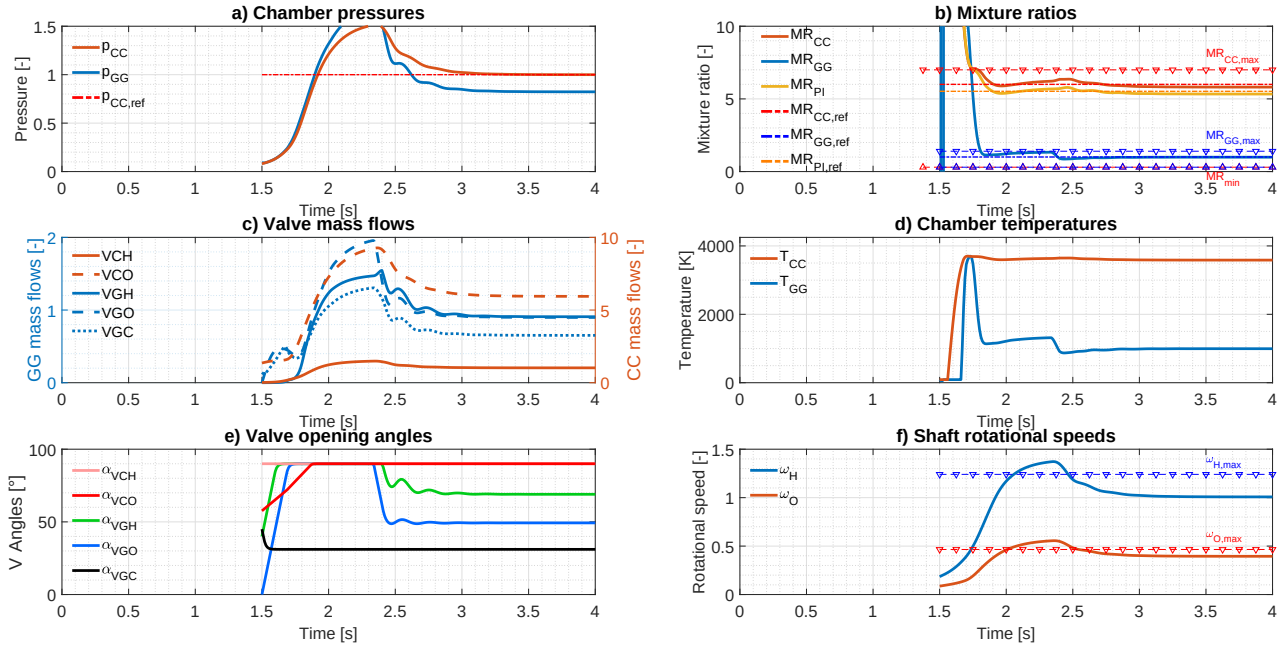


Figure 6.3.1: PID-control results for a nominal continuous start-up in *Vulcain 1*

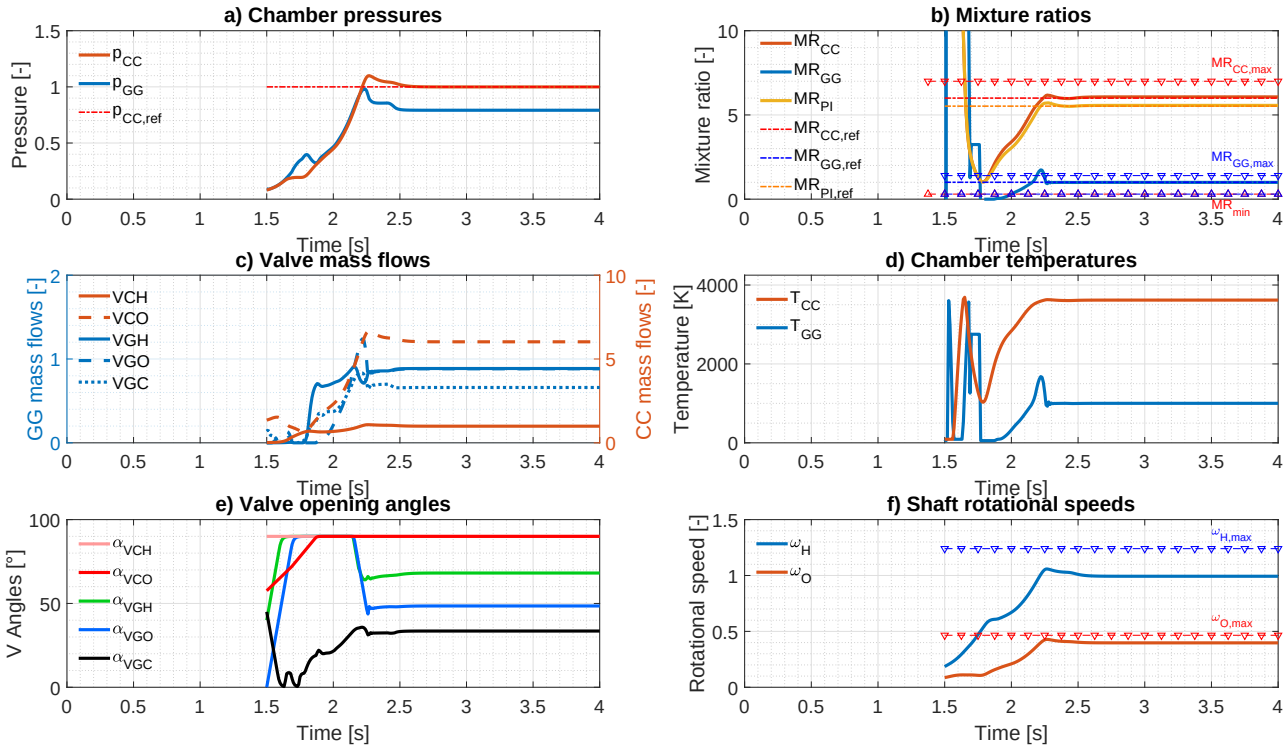


Figure 6.3.2: LQR-control results for a nominal continuous start-up in *Vulcain 1*

Regarding the conventional LQR techniques, an invariant controller has also been developed with the *MATLAB® lqr* command. In this case, the multivariable tracking goal has been requested at a selected state-weight matrix $Q_{lqr} = 10^2 \times I_n$ and control-weight matrix $R_{lqr} = I_m$, to give more relevance to state-error reduction (I_i being the identity matrix of order i). With this method, overshoot is reduced below 15%, which allows to respect the ω constraints in the nominal start-up, but not the MR_{GG} constraints, which are slightly violated (Figure 6.3.2). However, if a start-up until 120% of thrust is targeted, rotational-speeds constraints are also surpassed, as discussed later in the comparison with MPC in Section 6.5. In Figure 6.3.2 it can also be seen that static error in MR is reduced (below 1.3%) at the price of a higher error in p_{CC} (0.88%). With these multivariable methods, no actuator internal models are considered in the linear state-space models, due to the absence of measurements of their internal states.

The performance indicators of PID and LQR control are included in Table 6.5.2 in Section 6.5. Other conventional control methods, such as modal control comprising pole placement and state feedback with integrators, have also been tested without better results.

6.4 MPC controller design

6.4.1 Justification of method selection

As observed in the unsatisfactory results obtained with the reference control methods, a more advanced method is necessary to meet the control goals in this application. After the analysis of the state-space models derived in Chapter 5, and following the methodological literature review from Chapter 3, the main selected control method in this thesis has been Model Predictive Control. Indeed, the complexity of the models, being highly nonlinear and non-affine in control; the relatively high number of states to be controlled by few inputs and the need for a strict verification of hard constraints during transients motivated the selection of this approach. According to the MPC survey [101], this family of methods is very suitable for controlling systems in which the computation of a satisfactory off-line control law is too difficult or practically impossible. Moreover, it can be extended to include robustness considerations, as introduced in Section 3.3.1. In fact, with the alternative robust control methods presented in Section 3.3.2, not all of these control goals can be attained simultaneously at a reasonable computational time, provided that a control law can factually be synthesised. Regarding the H_∞-minimisation approaches, state and input constraints are not directly considered in the synthesis and multivariable tracking goals might be attained with low accuracy, even though robust solutions can be computed. Long computational times may also be a drawback of H_∞, as well as for LPV techniques. The resolution of large LMI-constrained optimisation problems, related to those methods, has been checked to be too computationally costly for this application. With regard to LYAPUNOV-based techniques, the definition of a valid LYAPUNOV function on this system has only been achieved for the linearised model, as explained later in this section. This obstacle, as well as the non-affinity of the model with respect to control, hinders the application of the sliding-mode and LYAPUNOV redesign techniques. Feedback linearisation [62], which could be helpful to simplify the control problem, also requires affinity in control inputs.
The versatility of the controller with respect to different engines and cycles was also a design goal.

The MPC approach developed in this chapter and in Chapter 7 is easy to adapt to distinct engines, provided that a similar modelling approach (as in Chapter 5) is carried out. The availability of symbolic nonlinear and linearised models describing the whole operating span of LPREs is a valuable asset which can be exploited with this model-based control approach. Furthermore, MPC can be extended to HMPC by considering the discrete elements into the optimisation, as reviewed in Section 3.3.4. The first steps towards a hybrid proposal for this system are included in Section 7.3.

6.4.2 MPC algorithm design

Control goals

The main goal of the MPC controller designed here is to drive the state towards a desired reference $\mathbf{x}_r$ at the end of the transient, with a special focus on having a small tracking error in p_{CC} and in the three different MR, at the CC, GG and PI. The states determining these quantities, and hence with greater tracking relevance, are incorporated into a reduced state vector $\mathbf{x}_z$ of length n_z:

$$\mathbf{x}_z = [p_{CC} \quad \dot{m}_{VCF} \quad \dot{m}_{VCO} \quad \dot{m}_{VGF} \quad \dot{m}_{VGO}]^T. \tag{6.4.1}$$

At the same time, a set of hard constraints on $\mathbf{x}$ and $\mathbf{u}$ has to be met throughout the transient. This objective is somewhat more important than tracking in order to avoid excessive temperatures or rotational speeds during engine's operation, which could lead to catastrophic failures or to prevent the reuse of certain parts. The concrete series of state and input constraints defining the allowable sets X and U and further restrictions are the following.
State constraints:

- Rotational speeds ω, or x_1 (and x_2 in *Vulcain 1*), must remain under an upper bound specific to each shaft. In *Vulcain 1*, these bounds are $\overline{\omega_H} = 40000rpm$ and $\overline{\omega_O} = 15000rpm$, and in *PROMETHEUS* $\overline{\omega} = 28000rpm$, in dimensional terms. These limits can be easily surpassed during transients when targeting high-thrust operating points.
- CC mixture ratio $MR_{CC} = \frac{\dot{m}_{VCO}}{\dot{m}_{VCF}}$, once its value starts to be physically meaningful (singular at zero flows), must remain within $[0.3, 7]$ in *Vulcain 1* and within $[0.1, 3.6]$ in *PROMETHEUS*. Lower bounds must be respected so as to avoid combustion extinction and upper bounds are directly related with the maximum allowable CC temperature (around $3700K$ in *Vulcain* 1). This translates into inequality constraints in the optimisation problem. These restrictions can be readily exceeded during transients if countermeasures are not set and only tracking goals are demanded.
- Analogous considerations are made for the GG mixture ratio $MR_{GG} = \frac{\dot{m}_{VGO}}{\dot{m}_{VGF}}$, the secondary combustion chamber, with stricter allowable intervals $[0.3, 1.4]$ and $[0.1, 0.3]$. The greatest temperature that can be attained in this chamber is around $1400K$ in *Vulcain 1*.
- For the rest of states, there are no formal specifications since excessive values are rarely attained. However, defining bounds helps the optimisation in MPC to reach an optimal solution faster. Hence, all maximum pressures are considered as 115% of $p_{CC,r}$. Maximum mass flows

are taken as 5 times the reference $\dot{m}_{VGF,r}$ for GG flows, 25 times that reference for $\dot{m}_{VCF}$, and 50 times for $\dot{m}_{VCO}$.

Input constraints:

- Valve sections A_V or $\mathbf{u}_c$ are generally limited to their physical opening, which in non-dimensional terms is $[0, 1]$. The particular VGC valve in *Vulcain 1*, u_5, is specially constrained to be opened at least at 5% so as to avoid extreme effects on MR. This valve is indeed never fully closed since it determines the flow ratio between turbines. In low-thrust computations there is a also 10% lower bound in VGO to avoid extinction in the GG. Concerning the CC injection valves, VCF and VCO, it has been mentioned in Section 6.3 that their tuning during fully continuous phases is not convenient, following the analyses from Chapter 4. Therefore, they are kept open at their nominal opening profile. This is expressed through hard bounds in u_1 and u_2, which eases the search of an optimal valid $\mathbf{u}_c$. System's controllability is guaranteed with u_3, u_4 and u_5 in *Vulcain 1*, the GG valves, as demonstrated in the SS-analysis Section 5.4. In hybrid phases, this is not necessarily true, as explained in Section 7.3.
- The rate of variation of valve sections or *sectional speed*, $\dot{\mathbf{u}}_c$, is also constrained according to actuators limitations $\dot{\mathbf{u}}_{c,max}$. In non-dimensional terms, GG-valves sectional speed is limited to 5 full sections per second, whereas CC-valves one is limited to 0.89 sections per second (according to specifications). The implementation of these constraints on the derivative of inputs in the optimisation problem is realised via inequality constraints (6.4.13).

Concrete reference trajectories are not imposed in the approach in this chapter, in contrast with Chapter 7. If start-up transients are to be controlled, the duration of this phase until reaching the reference is required to range between 2 and $4s$, which allows the system to cope with possible perturbations or uncertainties while complying with constraints. Indeed, robustness to parametric variations according to Table 5.4.1 is also a goal to be met by the controller. Apart from these vital transients, the throttling operations are also targetted with this algorithm. As a reminder, control inputs $\mathbf{u}$ only denote continuous inputs $\mathbf{u}_c$, since $\mathbf{u}_d$ are considered as active (equal to one).

Dynamics

As introduced in Section 3.3.1, MPC predicts the future system behaviour along a horizon, and optimises control inputs according to a cost function generally related to a reference trajectory or to an end state. Here, the dynamic model used in the state-feedback MPC controller is considered as a linearisation of the continuous-time nonlinear system f_s (5.2.19)-(5.2.30), as defined in 5.2.31, about the previously computed $(\mathbf{x}_r, \mathbf{u}_r)$ from (6.2.1). The MPC problem is initially considered in continuous time, as in (3.3.3). However, in order to solve it numerically, it is discretised via zero-order hold (exact discretisation) at $\Delta t = 10ms$, which is a constraint on the available controller computation frequency:

$$\mathbf{\Delta x}_{k+1} = A_d(\mathbf{x}_r, \mathbf{u}_r)\mathbf{\Delta x}_k + B_d(\mathbf{x}_r, \mathbf{u}_r)\mathbf{\Delta u}_k, \tag{6.4.2}$$

where k denotes the current time step. Thus, in linear terms, the goal of the controller is to find the set of $\mathbf{\Delta u} = \mathbf{u} - \mathbf{u}_r$ that drives the state to $\mathbf{\Delta x} = \mathbf{x} - \mathbf{x}_r = \mathbf{0}$. The matrix A_d is stable for all the physically feasible $\mathbf{x}_r$, which is a particularity of these GG-cycle LPRE models, highlighted in Section 5.4. Nonlinear models have also been tested for determining dynamic relationships. However, the on-line

resolution of the optimisation problem defined later in this section was not practical, requiring longer computational times ($\sim \times 10$) and usually leading to unfeasible and sub-optimal solutions. Therefore, linear dynamics has been implemented. Since the exogenous input $\mathbf{w}_t$ is zero at final equilibrium points, it has been omitted in the dynamic equations in this chapter. Besides, internal-actuators characteristics are directly considered in constraints.

Quasi-infinite horizon

The approach carried out is partially based on the QIH approach by [30], because it presents proofs for guaranteed CL stability and recursive feasibility of continuous-time MPC by incorporating the notion of a terminal region Ω. As introduced in Section 3.3.1, in [30] a terminal-state nonlinear inequality constraint and a penalty term are defined to enforce those features. The last step in the prediction horizon is constrained to Ω, where a local, simple LQR controller K is designed. By considering the related feedback $\mathbf{u}(t+T_p) = K\mathbf{x}(t+T_p)$, the P matrix of the LYAPUNOV function $V(\mathbf{x}) = \mathbf{\Delta x}^T P \mathbf{\Delta x}$ can be computed via the following continuous-time LYAPUNOV equation, previously stated in (3.3.8):

$$(A_K + \kappa I)^T P + P(A_K + \kappa I) = -Q_K - K^T R_K K. \tag{6.4.3}$$

In (6.4.3), the compound of the continuous-time linear system with K is considered, $A_K = A_c + B_c K$ (where A_c is the continuous counterpart of A_d) and $\kappa \in \mathbb{R}^+$ (satisfying $\kappa < -\lambda_{max}(A_K)$). Q_K and R_K are positive-definite symmetric weight matrices $Q_K \in \mathbb{R}^{n\times n}$, $R_K \in \mathbb{R}^{m\times m}$, whose diagonal values are of the order of 0.1, with the exception of $R_{K1,1}$ and $R_{K2,2}$ which are set to 10^{-10} due to the irrelevance of u_1 and u_2 (as shown in Section 4.3). The computed $P \in \mathbb{R}^{n\times n}$ serves to add a supplementary terminal-region term $E = V(\mathbf{x}(t+T_p))$ in the MPC cost, as proposed in [30]. The terminal-region constraint is expressed as $V(\mathbf{x}_{k+N_p}) \leq \alpha_P$ in the discretised problem. That α_P is iteratively calculated as the largest possible value that yields a non-negative result in the following non-convex nonlinear programme [30]:

$$\begin{aligned} \max_{\mathbf{x}} \quad & \mathbf{\Delta x}^T P \phi(\mathbf{x}) - \kappa \mathbf{\Delta x}^T P \mathbf{\Delta x} \\ & \text{s.t.} \quad \mathbf{\Delta x}^T P \mathbf{\Delta x} \leq \alpha_P, \end{aligned} \tag{6.4.4}$$

where $\phi(\mathbf{x}) = \mathbf{0}$ in this linear-dynamics case. In this application, $\kappa = 0.95$ and $\alpha_P = 0.2$.

Remarks:
Other options for the LYAPUNOV functions used in QIH, based on the nonlinear state-space model (5.2.19)-(5.2.30) (from Section 5.4.2), have also been implemented. Nevertheless, the optimisation problem could not be solved at acceptable computational times and feasible solutions were only found near the terminal region.
Besides, as stated in [94], when the MPC problem is discretised in practice, as introduced in the next paragraphs, the aforementioned QIH approach does not present a formal stability guarantee. In discrete-time it can only be said that those stabilising ingredients promote CL stability and feasibility, unless a discrete-time LYAPUNOV equation is solved.

Integral action

Furthermore, an integral action is also included to enforce a more precise tracking on $\mathbf{x}_z$, based on [135]. According to [110], the CL performance of model-based control is highly dependent on model representativeness. Modelling errors and disturbances can trigger steady-state error. Those authors highlight two general methods to avoid this in MPC:

a) Augment the decision variables with states representing tracking-error integration, like [135]. Drawbacks are additional computational cost and the possible need for anti-windup for the integral term.
b) Augment the linear MPC process model to consider a constant step disturbance, as in [110]. Disturbance is estimated and supposed constant in the future. To do so, general state-space models for estimating output and state disturbance have to be designed. Then, the steady-state target is shifted to adapt to it.

In this thesis, option a) has been selected in the absence of disturbance models and seeing that no relevant computational penalties were induced. The additional integral decision variables are denoted by $\mathbf{z}$ and present a corresponding weight matrix $S \in \mathbb{R}^{n_z \times n_z}$ in the cost, whose diagonal is $[1, 0.1, 0.1, 0.1, 0.1]$. The dynamics of these $\mathbf{z}$ variables is defined in constraints (6.4.16), where the constant gain matrix $K_{I,end} \in \mathbb{R}^{n_z \times n}$ determines the velocity of their response [135]. If this gain is set too large, unwanted oscillations in the related states can appear.

Cost definition

Thus, the MPC cost in continuous time is defined in a first step as:

$$J_{end}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{z}(t)) = \int_t^{t+T_p} \left(\boldsymbol{\Delta}\mathbf{x}(t)^T Q_{end} \boldsymbol{\Delta}\mathbf{x}(t) + \boldsymbol{\Delta}\mathbf{u}(t)^T R_{end} \boldsymbol{\Delta}\mathbf{u}(t) + \mathbf{z}(t)^T S \mathbf{z}(t) \right) d\tau + V(\mathbf{x}(t+T_p)), \tag{6.4.5}$$

which in the discretised problem yields, with a shortened control horizon:

$$J_{end}(\mathbf{X}, \mathbf{U}, \mathbf{Z}) = \left(\sum_{j=0}^{N_p-1} \boldsymbol{\Delta}\mathbf{x}_{k+j}^T Q_{end} \boldsymbol{\Delta}\mathbf{x}_{k+j} + \sum_{j=0}^{N_u} \boldsymbol{\Delta}\mathbf{u}_{k+j}^T R_{end} \boldsymbol{\Delta}\mathbf{u}_{k+j} + \sum_{j=0}^{N_p-1} \mathbf{z}_{k+j}^T S \mathbf{z}_{k+j} \right) \Delta t + V(\mathbf{x}_{k+N_p}). \tag{6.4.6}$$

This definition consists in the traditional quadratic cost on states and controls (3.3.7) plus the added integral and terminal costs, with a prediction horizon $N_p = 10$ steps $(0.1s)$ and a control horizon $N_u = 5$. Implicitly, the last control $\mathbf{u}_{k+N_u}$ is used for $j \geq N_u$, j representing the time steps along the horizon. Further extensions of these horizons did not improve the solutions in terms of tracking or constraints satisfaction. Let us recall from (3.3.6) that $\mathbf{X}$, $\mathbf{U}$ and $\mathbf{Z}$ are the succession of predicted/computed $\mathbf{x}$, $\mathbf{u}$ and $\mathbf{z}$ at each j along their respective horizon. Q_{end} and R_{end} are positive-definite symmetric weighting matrices $Q_{end} \in \mathbb{R}^{n \times n}$, $R_{end} \in \mathbb{R}^{m \times m}$, whose diagonals have been computed off-line via

Kriging-based black-box optimisation as in [98]. Their impact on performance indicators is major. An analogous procedure to the search of a worst-case scenario from Section 5.4.6 has been employed here. In this optimal weight selection, a Kriging model, introduced in Section 3.4, is built upon the following input-output data. Inputs consist in the diagonal terms of Q_{end} and R_{end} and the chosen output is the simulated performance criterion (5.4.5) already defined in the SOBOL analyses from Section 5.4.5. This criterion concerns static errors in $\mathbf{x}_z$, and overshoot and surface difference with respect to the nominal transient in terms of p_{CC}. The simulations required to quantify this criterion are executed on the whole CL during the nominal start-up transient. An EGO scheme based on expected improvement is then used on the resulting Kriging model to minimise that criterion by varying the inputs at a threshold of $EI = 10^{-6}$. For this sake, about 500 iterations, implying around 60 hours of calculation, are required. The resulting values are specified in Section 6.5.

Robustness considerations

Moreover, some robust considerations have been implemented. The minimisation of the previous J_{end} under constraints is not intrinsically robust. Indeed, robustness against parameters and initial conditions variations, perturbations and modelling error is very important in this application. Robust MPC approaches generally make use of the *minimax* optimisation, which minimises the worst-case scenario of endogenous or exogenous perturbations. Other approaches, summarised in Section 3.3.1, are the tube-based MPC [100] or explicit robust MPC [106]. The robust *minimax* approach has been selected due to the elevated complexity of building invariant sets along trajectories (tube-based) or off-line state partitions (explicit MPC) seeing the highly multivariable nature of this system. A generic expression of the continuous *minimax* MPC problem has been defined in (3.3.10). The same problem, expressed in discretised version and with uncertain dynamics is the following [83], in which Δ represents uncertainty in dynamic matrices $A_d(\cdot,\cdot,\Delta)$, $B_d(\cdot,\cdot,\Delta)$:

$$\begin{aligned} \min_{\mathbf{U}} \max_{\Delta} \quad & J(\mathbf{X},\mathbf{U}) \\ \text{s.t.} \quad & \mathbf{X} \in X \quad \forall \Delta \in \boldsymbol{\Delta}_c \\ & \mathbf{U} \in U \quad \forall \Delta \in \boldsymbol{\Delta}_c \end{aligned} \tag{6.4.7}$$

In this manner, internal parameter variations affecting the dynamic matrices would be taken into account by the programme, which is a priority in this thesis. However, solving (6.4.7) for compact uncertainty sets $\boldsymbol{\Delta}_c$ is generally not tractable on-line, and especially too costly in this application. The resolution of this problem via LMI constrains in the frame of a semidefinite relaxation programme (3.3.11) by [83] has been attempted here with *YALMIP*, leading to too long computational times (100 times the real time) without improved robustness. Hence, it has been opted for choosing a finite discrete set of uncertain scenarios (in a similar manner to [26]), the so-called scenario-based MPC [95]. In addition, the equivalent epigraph formulation based on [83], stated in (3.3.11), has been implemented so as to avoid the maximisation. Concretely, this formulation allows to pose the problem as a minimisation of a scalar $\gamma \in \mathbb{R}^+$, which is equivalent to performing scenario-based *minimax*. In

this approach, that γ constrains the J_{end} (6.4.6) of the nominal non-robust problem (3.3.3) evaluated at several perturbed states propagations $\mathbf{X}_i$:

$$\begin{aligned} \mathbf{X}_i &= [\mathbf{x}_{i,k}, ..., \mathbf{x}_{i,k+j}, ..., \mathbf{x}_{i,k+N_p}]^T, \quad i \in I, \\ \mathbf{\Delta x}_{i,k+1} &= A_d(\mathbf{x}_r, \mathbf{u}_r, \Delta_{i,k})\mathbf{\Delta x}_{i,k} + B_d(\mathbf{x}_r, \mathbf{u}_r, \Delta_{i,k})\mathbf{\Delta u}_k, \quad i \in I, \end{aligned} \tag{6.4.8}$$

where $\Delta_{i,k}$ are certain selected internal parameter variations belonging to the non-compact $\mathbf{\Delta}_I = \{\Delta_{i,k}, i \in I\}$. I is a finite set, which serves to index the considered perturbed cases. The epigraph formulation permits to entirely shift the robustness considerations into the list of constraints, thereby not requiring maximisation. Therefore, only a smooth convex NLP is required, which is more computationally tractable than (6.4.7).
The series of Δ_i, where $I = \{1, 2, 3, 4\}$, are taken constant for all k and j and represent the following parameter-variation cases, selected according to the sensitivity and worst-case analysis from Sections 5.4.5 and 5.4.6:

- $i = 1$: nominal parameters.
- $i = 2$: the main varying parameters are increased by 10%: interdependent tanks pressures p_{PI} and temperatures T_{PI}.
- $i = 3$: the main varying parameters are decreased by 10%: interdependent tanks pressures p_{PI} and temperatures T_{PI}.
- $i = 4$: worst-case variations combination defined in Table 5.4.2.

Cases 2 and 3 consist in the most probable scenarios of endogenous perturbations which the controller should be able to face, particularly within the reusability context. From the analysis Section 5.4.6, case 4 is an off-line estimation of the extreme dynamics-perturbing scenario corresponding to the compact uncertainty set $\mathbf{\Delta}_c$ (related to Table 5.4.1), for which a costly *minimax* would be necessary.

Remark:
An alternative definition of the disturbance, which has been proposed in the frame of this thesis [119], is an additive disturbance. This representation was aimed at capturing exogenous perturbations to the system, affecting dynamics in the following manner:

$$\mathbf{\Delta x}_{i,k+1} = A_d(\mathbf{x}_r, \mathbf{u}_r)\mathbf{\Delta x}_{i,k} + B_d(\mathbf{x}_r, \mathbf{u}_r)\mathbf{\Delta u}_k + \mathbf{w}_{i,k}, \tag{6.4.9}$$

where $\mathbf{w}_{i,k}$ are again certain selected perturbation vectors belonging to $\mathbb{W} = \{\mathbf{w}_{i,k}, i \in I\}$. The different $\mathbf{w}_{i,k}$ represent the various ways in which the system could evolve after an unknown perturbation in the state, and hence it was proposed to estimate them by analysing the modes of the system (eigenvectors of A_c). I corresponded to a subset of the eigenvectors related to the closest eigenvalues to zero (to the instability condition). In this manner, the structural information of A_c was used to define the most unfavourable disturbance scenarios, similarly to [169]. However, this consideration was discarded since no formal definition of the external perturbations that the controller should be able to face has been available to the thesis. Robustness to parametric variations is more founded on concrete expected requirements of the control loop developed in this thesis.

Optimisation problem

All in all, the discretised minimisation problem proposed here, in which decision variables are extended to consider all $\mathbf{X}_i$, is described below:

$$\min_{\mathbf{X}_i,\mathbf{U},\mathbf{Z}_i,\gamma} \quad \gamma \tag{6.4.10}$$

$$\text{s.t.} \quad J_{end}(\mathbf{X}_i,\mathbf{U},\mathbf{Z}_i) \leq \gamma \qquad \forall i \in I \tag{6.4.11}$$

$$\mathbf{X}_i \in X, \quad \mathbf{U} \in U \qquad \forall i \in I \tag{6.4.12}$$

$$A_{ineq}[\mathbf{X}_i \quad \mathbf{U}]^T \leq \mathbf{b}_{ineq} \qquad \forall i \in I \tag{6.4.13}$$

$$A_{i,eq,end}[\mathbf{X}_i \quad \mathbf{U}]^T = \mathbf{b}_{i,eq,end} \qquad \forall i \in I \tag{6.4.14}$$

$$\boldsymbol{\Delta}\mathbf{x}_{i,k+N_p}^T P_i \boldsymbol{\Delta}\mathbf{x}_{i,k+N_p} \leq \alpha_P \qquad \forall i \in I \tag{6.4.15}$$

$$\mathbf{z}_{i,k+1+j} = \mathbf{z}_{i,k+j} + \Delta t K_{I,end} \boldsymbol{\Delta}\mathbf{x}_{z,i,k+j} \qquad \forall i \in I, j \in [0, N_p - 1]. \tag{6.4.16}$$

X and U in (6.4.12) are the allowable sets for states and control (compact subsets of $\mathbb{R}^{n(N_p+1)}$ and $\mathbb{R}^{mN_u}$ respectively), comprising the previously defined bound constraints. The set U for the first control $\mathbf{u}_{MPC} \equiv \mathbf{u}_k$, the one which is really applied to the plant, is specially bounded to comply with actuators capacity [93]:

$$\mathbf{u}_{MPC} \in [\max(\underline{U}, \mathbf{u}_0 - \dot{\mathbf{u}}_{max}\Delta t), \min(\overline{U}, \mathbf{u}_0 + \dot{\mathbf{u}}_{max}\Delta t)], \tag{6.4.17}$$

where $\mathbf{u}_0$ is the previous-step control (warm start is performed in control variables), $\dot{\mathbf{u}}_{max}$ is the maximum sectional velocity of valves and $\underline{U}$ and $\overline{U}$ are the inferior and superior bounds of U. Thanks to saving the full $\mathbf{U}_0$ computed in the previous step, from which only the first control is transmitted to the plant, an initial guess for the whole $\mathbf{X}_i$ can be made by propagating dynamics (6.4.8) from $\mathbf{x}_k \equiv \mathbf{x}_0$. That initial state $\mathbf{x}_k$ corresponds to the measurement of simulator's state at each time step, fed back to the controller (as in Figure 6.1.1). The initial value of γ is set large enough (10^5).
Regarding the rest of constraints, (6.4.10) contains equality constraints (6.4.14) (defined by $A_{i,eq,end}$ and $\mathbf{b}_{i,eq,end}$) for determining initial conditions and the different linear dynamics (6.4.8) along the whole horizon. Linear inequality constraints (6.4.13) (defined by A_{ineq} and $\mathbf{b}_{ineq}$) are set for complying with MR and actuators sectional-velocity limits at all $\mathbf{X}_i$. Indeed, the vital mixture ratios limits defined earlier establish linear inequalities between states; and actuators constraints are defined similarly to (6.4.17). The difference in control inputs between two time steps must not exceed $\pm\dot{\mathbf{u}}_{max}\Delta t$.
Inequalities (6.4.15) represent the nonlinear terminal-region constraints computed in (6.4.4) [30], where a different P_i is used according to each perturbed scenario i. The last constraints (6.4.16) correspond to the integrator dynamics [135], where $\mathbf{z}_{i,k}$ (at $j = 0$) are initialised to zero the $K_{I,end}$ matrix has been computed off-line in the same manner as Q_{end} and R_{end}. The resulting values are specified in Section 6.5. Performance indicators are very sensitive to these gains.
It is important to emphasise the fact that the resulting $\mathbf{U}$ obtained in (6.4.10)-(6.4.16) has been confronted to all these scenarios and that all propagated perturbed states must comply with all constraints, thereby improving the robustness of the controller. This approach with equality constraints within an uncertain problem is only valid because of the finite choice of $\Delta_{i,k}$ or $\mathbf{w}_{i,k}$.

6.5 CL simulations and analysis of results

Simulations of the control loop with end-state-tracking MPC (E.MPC) proposed in this chapter (Figure 6.1.1) are run at a maximum integration time step of $\Delta t_{sim} = 10^{-5}s$, as required by the developed engine simulator (Chapter 4). However, the maximum computation frequency allowed to the controller is $100Hz$, i.e. control orders are sent every $\Delta t = 10ms$. This piecewise-constant control is implemented in *Simulink*® via rate-transition blocks. Integration is performed with that fixed step and *ode3* (BOGACKI-SHAMPINE scheme), seeing that higher-order and variable-step methods led to identical results (as mentioned in Section 4.3) despite the associated longer simulation times.
The interior-point optimisation software *IPOPT* [164], presented in Section 3.3.3, has been used to solve this smooth convex NLP (6.4.10) within the *MATLAB*® environment. As mentioned in that section, this tool has been selected instead of *MATLAB*® *fmincon* or *YALMIP* [84, 85] owing to its efficient interior-point conception, which intrinsically respects barriers on decision variables; and to its coding flexibility, allowing the introduction of user-defined gradients, Jacobians and Hessians in the calculation. This, together with the consideration of sparse matrices, alleviate the computational burden, vital in this application.
Optimality tolerance is set to 10^{-6} and constraints-violation tolerance is set to 5×10^{-2}. The gradient of the cost γ as well as the Jacobian (with respect to decision variables) of all constraints other than constant bounds, (6.4.11), (6.4.14), (6.4.13), (6.4.15), (6.4.16), are provided to the *IPOPT* function. Hessians are approximated via the limited-memory quasi-Newton method. This is more convenient in this case where the resulting dense Hessian would increase computation time.
In this manner, the resulting computational times in *MATLAB*® are of the order of ten times longer than real time, which does not rule out a future real-machine implementation. The number of decision variables considered in the selected MPC design is $length(I) \times (N_p + 1) \times n + (N_u + 1) \times m + length(I) \times N_p \times n_z + 1$ (for γ)$= 759$. Apart from the individual bound constraints on those variables, the number of equality and inequality constraints combined is 1904. An overview of the implementation of this loop in *Simulink*® is provided in Figure 6.5.1.

Two main continuous-control scenarios are simulated: the continuous sub-phase of the start-up transient as well as the throttling operation, which consists in a change of operating point. As mentioned in the previous chapters, the control of these transient phases is one of the main motivations of this thesis. The MPC-parameters tuning results for the former scenario via the aforementioned Kriging-based optimisation are:

- $Q_{end} = diag(1790.12, 7839.51, 5370.37, 308.64, 185.19, 6728.4, 4259.26, 6851.85, 7222.22, 9814.85, 9814.81, 7222.22)$.
- $R_{end} = diag(0, 0, 185.19, 925.93, 9444.44)$.
- $K_{I,end} = diag(90.74, 42.59, 1.85, 98.15, 94.44)$.

And for throttling scenarios:

- $Q_{end} = diag(8950.62, 7469.14, 3888.89, 6728.4, 2901.23, 7098.77, 925.93, 8950.62, 4506.17, 3395.06, 50009691.36)$.
- $R_{end} = diag(0, 0, 3271.60, 3395.06, 9074.07)$.

- $K_{I,end} = diag(0.62, 11.73, 31.48, 27.78, 98.15)$.

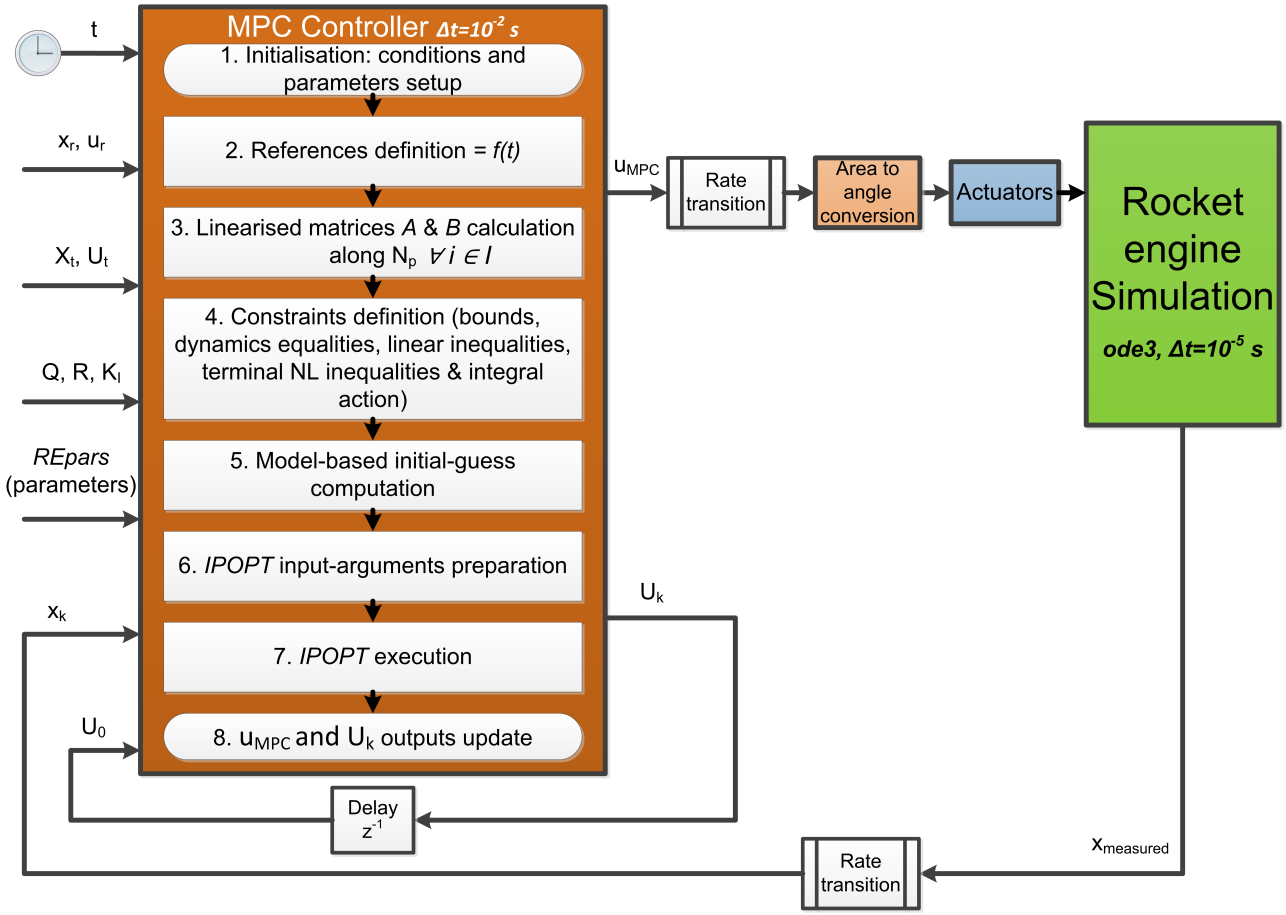


Figure 6.5.1: Overview of the MPC algorithm implemented *Simulink*® (non-introduced variables correspond to Chapter 7)

6.5.1 Continuous start-up transient

Regarding the start-up operation, simulations of the loop 6.1.1 with the MPC synthesised in this chapter (6.4.10)-(6.4.16) are executed from $1.5s$ until $3s$ after the start command. That interval corresponds to the time window in which continuous control is possible in engine start-up transient.

As introduced in Chapter 4, mixture ratios naturally start from values very far from the allowable area, due to the low initial mass flows that hinder the definition of quotients. Indeed, chambers are not physically fully ignited during the first instants (even if igniters are active); hence, MR are not representative. The corresponding temperature calculations, performed within the controller via polynomial functions of MR according to the assumptions in Section 5.1, are neither realistic during those initial phases. Figure 6.5.2 depicts the results of p_{CC} tracking for three main operating points in *Vulcain 1* design envelope: $p_{CC,r} = 1$ (nominal), $p_{CC,r} = 0.7$ (minimum for this engine) and $p_{CC,r} = 1.2$ (maximum). At all three points, the reference mixture ratios remain approximately the same $MR_{CC,r} = 6$, $MR_{GG,r} = [0.8, 1.1]$ and $MR_{PI,r} = 5.25$, as usual during start transients. MR tracking for the three cases is depicted in Figure 6.5.3.

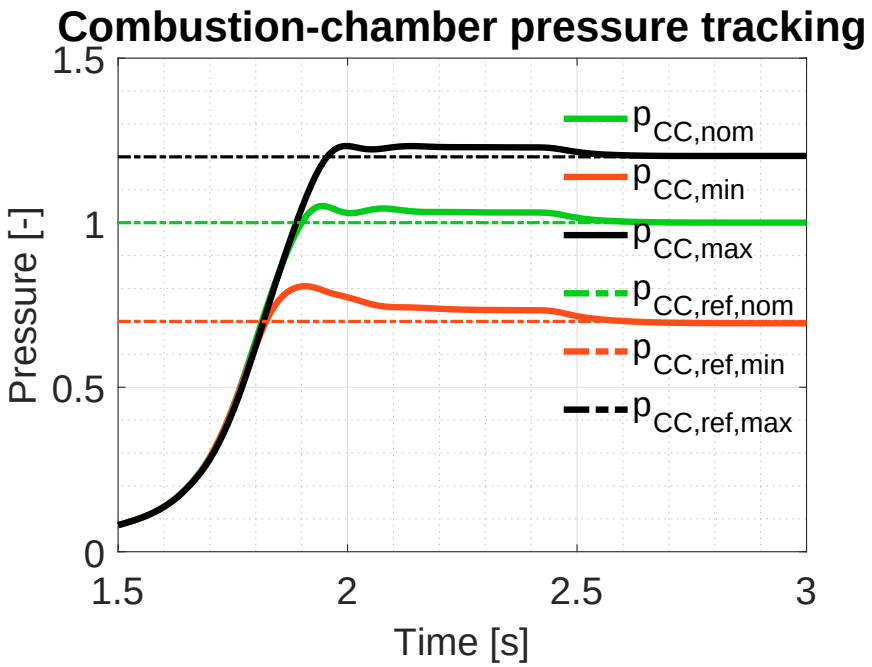
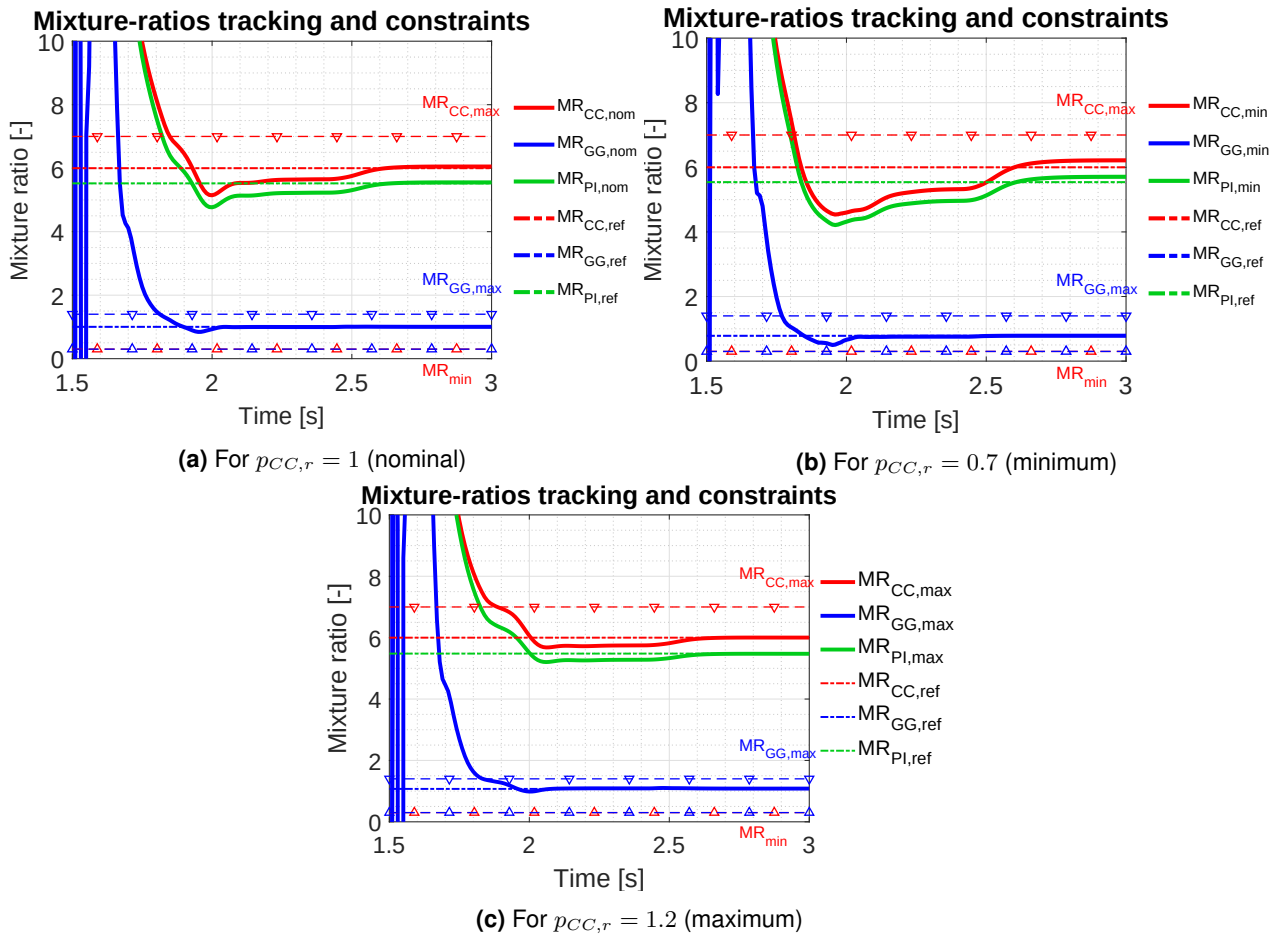


Figure 6.5.2: E.MPC Tracking results in p_{CC} for $p_{CC,r} = 1$ (nominal), $p_{CC,r} = 0.7$ (minimum) and $p_{CC,r} = 1.2$ (maximum)

Tracking is achieved with acceptable accuracy in p_{CC} for all cases (under 0.02% in the nominal case, under 0.9% in off-nominal) and with little error in MR (under 0.9% in nominal, under 3.6% in off-nominal) while respecting constraints up from the time when it is considered feasible and acceptable to respect them in practice ($t = 1.9s$). Indeed, maximum-MR constraints are relaxed during the first instants until $t = 1.6s$, due to the far infeasible initial values. Up from that instant, those constraints are fully active. This implies that the optimiser provides slightly infeasible solutions, albeit those constraints help to drive the system to the allowable zone before $t = 1.9s$. The rest of the constraints are verified along the entire simulations.

The overshoot and slight oscillations present until around $t = 2.5s$, before achieving the final tracking, are generated by the nonlinear exogenous influence of the GG-starter input mass flow, which is not taken into account in the linearised model, as mentioned before. Overshoot is more pronounced in the minimum case since the relative influence of the starter is more elevated. All these performance indicators are explicitly listed in Table 6.5.1.

The controller is able to achieve analogous tracking performance after altered initial conditions (independent, uniformly distributed random $\pm 50\%$) coming from the sequential transient.



(a) For $p_{CC,r} = 1$ (nominal)

(b) For $p_{CC,r} = 0.7$ (minimum)

(c) For $p_{CC,r} = 1.2$ (maximum)

Figure 6.5.3: E.MPC tracking results in MR

6.5.2 Throttling transient

With regard to the throttling scenario, a throttle-down operation from nominal to minimum thrust (100% to 70%) of thrust has been selected as a representative case. This is a scenario that reusable LPREs must face during their operation. Figure 6.5.4 depicts the complete results of the control of this operation via the MPC controller from this chapter. The engine is already active at its nominal state at $t = 3s$ and a step-wise order to track 70% of thrust is sent, with the corresponding pre-calculated $(\mathbf{x}_r, \mathbf{u}_r)$. It can be seen in Figure 6.5.4 that the operation shift is safely performed (no constraints violated) within $0.34s$. Precise performance indicators are summarised in Table 6.5.1. Other throttling scenarios such as an increase to 120% are also satisfactorily accomplished.

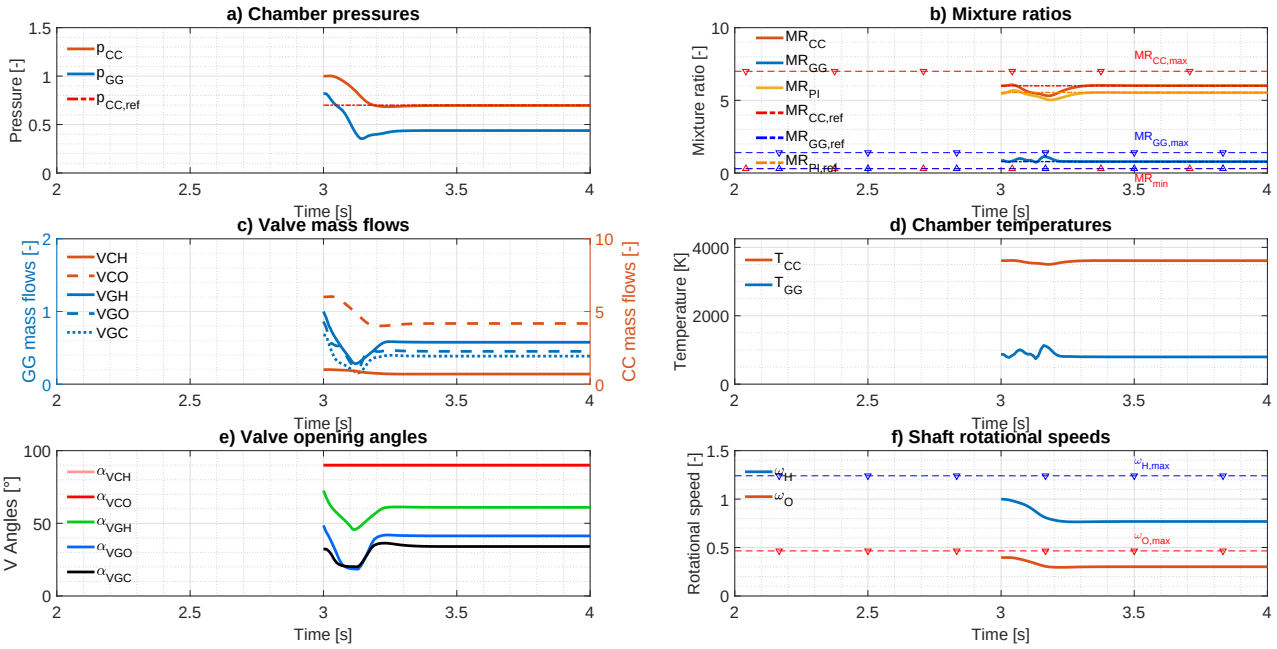


Figure 6.5.4: Complete results for E.MPC-controlled throttle down to $p_{CC,r} = 0.7$ from $p_{CC,r} = 1$ (100% to 70%)

6.5.3 Comparison with OL and other linear controllers

Table 6.5.1 summarises the comparison between this end-state tracking proposal and OL applied to the selected scenarios, in terms of some performance indicators: settling time to 99% of $p_{CC,r}$, overshoot in p_{CC}, the instant from which constraints are verified, static errors in p_{CC}, MR_{CC}, MR_{GG} and MR_{PI}, and the overall-performance criterion J_{perf} (5.4.5) defined in Section 5.4. The nominal OL

Table 6.5.1: Performance-indicators comparison between this E.MPC CL proposal and OL for start-up and throttling scenarios

Operating point	**Nominal start-up**		**Minimum start-up**		**Maximum start-up**		**Throttle down to** 70%	
Indicator	**OL**	**E.MPC**	**OL**	**E.MPC**	**OL**	**E.MPC**	**OL**	**E.MPC**
Settling time ($p_{CC,r} \pm 1\%$) $[s]$	2.76	**2.53**	-	**2.56**	2.7	**2.52**	-	**0.34**
Overshoot (% in p_{CC})	6.29	**5.11**	**15.18**	15.28	3.43	**2.76**	3.19	**2.29**
Constraints verification $[s]$	**1.81**	1.85	1.83	**1.81**	**1.77**	1.89	0	0
p_{CC} static error (%)	0.21	**0.019**	2.67	**0.82**	0.45	**0.23**	3.19	**0.498**
MR_{CC} static error (%)	**0.18**	0.87	**2.66**	3.6	3.25	**0.041**	2.72	**0.18**
MR_{GG} static error (%)	1.41	**0.0036**	1.35	**0.21**	1.28	**0.94**	1.37	**0.05**
MR_{PI} static error (%)	1.41	**0.55**	**2.92**	2.99	3.48	**0.11**	2.98	**0.29**
J_{perf} overall performance	4.16	**2.43**	11.45	**9.26**	9.54	**2.36**	10.29	**1.04**

is engine's original command, which is precisely tuned for the standard case, as traditionally done in flight-ready engines. Conversely, the minimum and maximum OL commands have been computed by

means of the preprocessor explained in Section 6.2, maintaining the same control as in the reference $\mathbf{u}_r$ along the transient. As mentioned in that section, these OL commands already represent relevant solutions for the off-nominal steady-state multivariable control. The improvement of CL with respect to OL can be observed in the reduction in J_{perf} in all cases, even though some individual indicators deteriorate during the start-up transient. Indeed, the real gain of this CL MPC control appears for operating points different from the nominal, where multivariable tracking was difficult to achieve with high performance while respecting constraints during the transient. Furthermore, valves openings are not saturated with this predictive approach, even for the maximum thrust case. The enhancement in the down-throttling operation is noticeable in all indicators.

When comparing with the other linear control methods described earlier in this chapter (Section 6.3), concerning PID (Figure 6.3.1) and LQR (Figure 6.3.2) correctors, the following conclusions can be drawn. Their respective performance indicators are included in Table 6.5.2. Tracking results of PID controllers are good in some of the reference variables (under 0.005% in nominal p_{CC}), but not for all of them simultaneously. Moreover, there are no guarantees of complying with all the constraints in this problem. Hence, when aiming at tracking off-nominal points, constraints are indeed highly violated. For instance, while throttling up until $p_{CC} = 1.2$, the system controlled by PID or LQR has the tendency to surpass rotational speeds bounds, as depicted in Figure 6.5.5, whereas the MPC controller designed in this chapter respects them. In fact, PID control presents large overshoot at all cases. The improvement in global performance with respect to OL is only given at 70% thrust by PID and at 120% by LQR; and E.MPC presents better overall indicators.

Table 6.5.2: Performance-indicators for PID and LQR control of the nominal start-up

Operating point	**Nominal start-up**			**Minimum start-up**			**Maximum start-up**		
Indicator	**OL**	**PID**	**LQR**	**OL**	**PID**	**LQR**	**OL**	**PID**	**LQR**
Settling time ($p_{CC,r} \pm 1\%$) $[s]$	2.76	3.1	**2.54**	-	2.72	**2.57**	**2.7**	3.87	-
Overshoot (% in p_{CC})	**6.29**	50.64	9.98	**15.18**	70.77	36.42	3.43	22.09	**2.98**
Constraints verification $[s]$	**1.81**	2.45	2.25	1.83	**1.78**	1.92	**1.77**	3.69	1.98
p_{CC} static error (%)	0.21	**0.005**	0.044	2.67	**0.003**	0.03	**0.45**	0.55	1.14
MR_{CC} static error (%)	**0.18**	3.15	1.2	**2.66**	2.7	4.63	3.25	3.3	**2.38**
MR_{GG} static error (%)	1.41	1.41	**0.65**	1.35	**0.075**	1.09	1.28	0.66	**0.46**
MR_{PI} static error (%)	1.41	3.41	**0.81**	**2.92**	3.06	3.99	3.48	3.45	**2.63**
J_{perf} overall performance	**4.16**	11.74	5.9	11.45	**9.63**	12	9.54	11.86	**7.92**

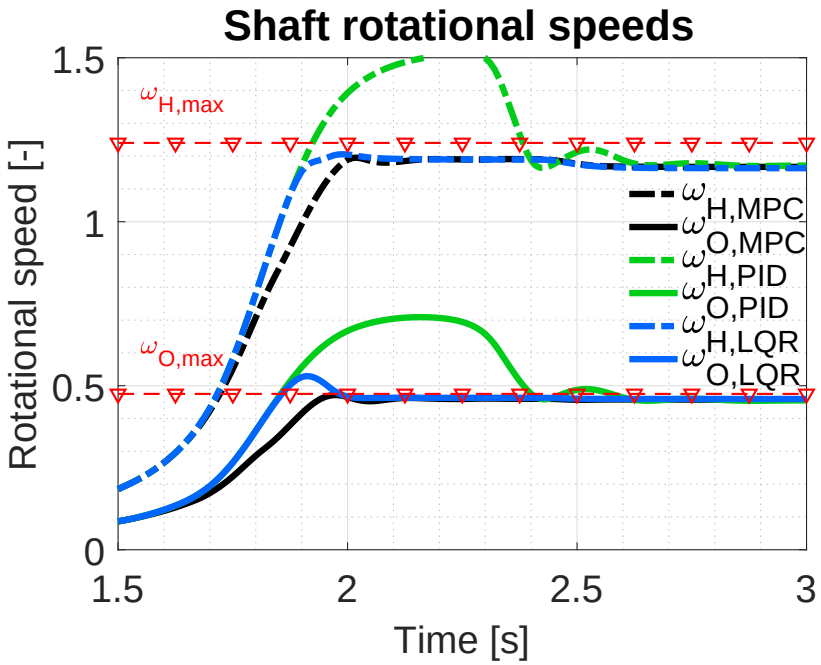


Figure 6.5.5: Rotational speeds ω_H and ω_O for a start-up to $p_{CC,r} = 1.2$ with end-state MPC, PID and LQR controllers

6.5.4 Robustness analysis

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 $\mathbf{x}_r$. Robustness is also an advantage of this controller. In Section 6.4.2, 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 or are uncertain. These scenarios have been simulated by intentionally altering those parameters in the simulator. The performance indicators obtained with this end-state-tracking MPC controller (6.4.10)-(6.4.16) for attaining a nominal-thrust start-up are shown in Table 6.5.3.

Figure 6.5.6 depicts the CL simulation corresponding to the worst-case-scenario start-up (to 100% thrust). It is observable that robustness is demonstrated in all scenarios. Constraints are normally respected and tracking errors generally diminish with respect to OL control, especially in mixture ratios. Drawbacks appear in constraint-verification times, where the controller needs more time to arrive to the feasible MR zone. The robustness analysis concerning throttling transients, is explicitly stated in Table 6.5.4, where the comparison between nominal and worst-cases is made. All indicators are enhanced in that case.

All in all, results point to considerably greater robustness to parameters variations with respect to OL in both transient phases simulated. The performance criterion J_{perf} diminishes in all scenarios. Hence it can be concluded that this synthesised end-state-tracking MPC controller fulfils the control goals defined in Section 6.4.2. However, there is still room for improvement. Tracking errors and overshoot can still be reduced, as it will be shown in Chapter 7.

Table 6.5.3: Performance-indicators comparison between E.MPC and OL in perturbed scenarios for start-up transients

Perturbed cases	**Case 1: nominal**		**Case 2: p_{PI} increased by 10%**		**Case 3: p_{PI} decreased by 10%**		**Case 4: worst case**	
Indicator	**OL**	**E.MPC**	**OL**	**E.MPC**	**OL**	**E.MPC**	**OL**	**E.MPC**
Settling time ($p_{CC,r} \pm 1\%$) $[s]$	2.76	**2.53**	2.8	**2.55**	2.74	**2.52**	-	**2.5**
Overshoot (% in p_{CC})	6.29	**5.11**	6.52	**4.85**	6.08	**4.96**	**4.28**	5.9
Constraints verification $[s]$	**1.81**	1.85	**1.8**	1.84	**1.81**	1.86	**1.8**	1.85
p_{CC} static error (%)	0.21	**0.019**	0.38	**0.11**	**0.062**	0.066	1.75	**0.46**
MR_{CC} static error (%)	**0.18**	0.87	**0.38**	1.05	**0.02**	0.68	1.25	**1.11**
MR_{GG} static error (%)	1.41	**0.0036**	1.39	**0.0016**	1.42	**0.0089**	1.62	**1.33**
MR_{PI} static error (%)	1.41	**0.55**	1.21	**0.72**	1.61	**0.36**	2.97	**0.57**
J_{perf} overall performance	4.16	**2.43**	4.25	**2.82**	4.17	**2.18**	8.57	**4.52**

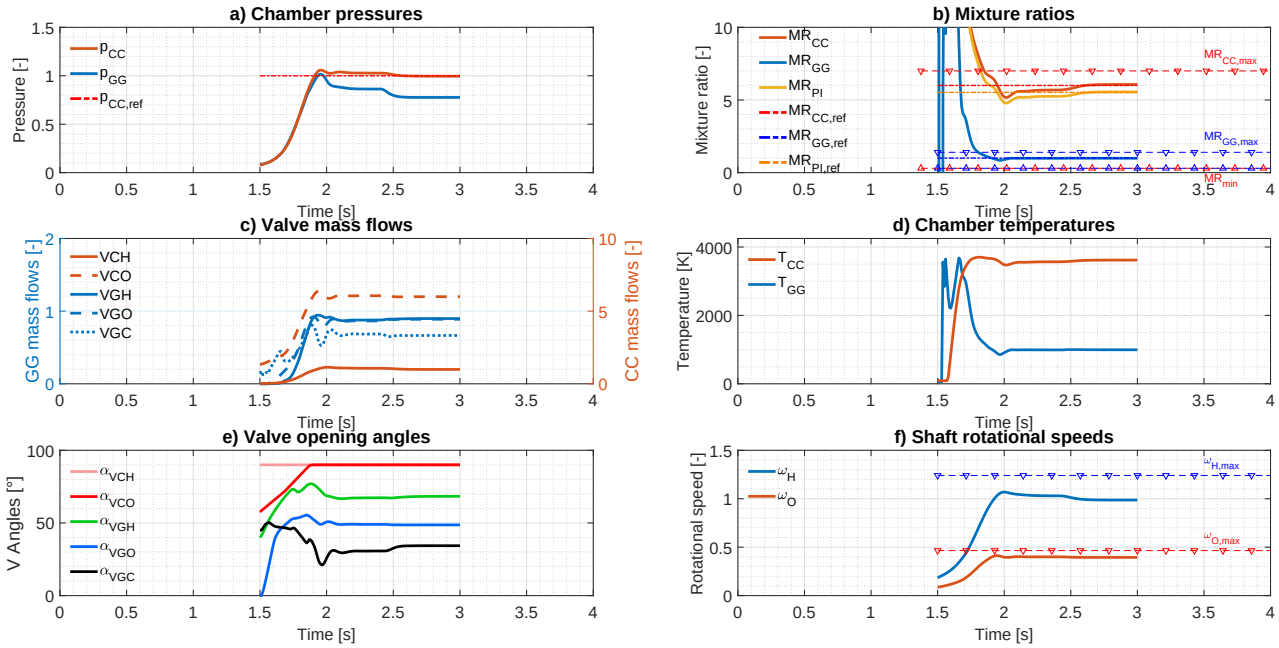


Figure 6.5.6: Complete E.MPC-controlled results in worst-case scenario 4 for a start-up to $p_{CC,r} = 1$ (nominal)

Table 6.5.4: Performance-indicators comparison between end-state-tracking MPC and OL in perturbed scenarios for throttling transients

Perturbed cases	**Case 1: nominal**		**Case 4: worst case**	
Indicator	**OL**	**E.MPC**	**OL**	**E.MPC**
Settling time ($p_{CC,r} \pm 1\%$) $[s]$	-	**0.34**	-	-
Overshoot (% in p_{CC})	3.19	**2.29**	5.24	**2.57**
p_{CC} static error (%)	3.19	**0.498**	5.24	**1.12**
MR_{CC} static error (%)	2.72	**0.18**	4.61	**0.82**
MR_{GG} static error (%)	1.37	**0.05**	2.68	**0.58**
MR_{PI} static error (%)	2.98	**0.29**	4.9	**1.38**
J_{perf} overall performance	10.29	**1.04**	17.48	**3.92**

6.6 Summary

As reviewed in Chapter 2, the control of the transient phases of liquid-propellant rocket engines has traditionally been performed either in open loop or with conventional closed-loop techniques. In this chapter it has been sought to improve in terms of performance and robustness the control of fully continuous transient phases of gas-generator-cycle LPREs, in which discrete actuators are already active. The considered phases are the fully continuous part of the start-up as well as throttling or operation shift, whose thermodynamic behaviour has been captured with the state-space models derived in Chapter 5. Valves sections are continuously adjusted for controlling pressure in the main chamber and mass-flow mixture ratios. Control goals concern not only multivariable tracking, but also hard-constraints verification and robustness to parametric variations at a fair computational cost. In a first step, reference linear controllers have been elaborated with PID and LQR techniques. However, in order to comply with all the control goals, an MPC controller has been synthesised for end-state tracking while respecting the set of hard operational constraints. This controller is accompanied by a preprocessor that serves to provide a full-state and control reference built from launcher needs, equivalent to an OL command, by making use of the nonlinear state-space model of the engine derived in Chapter 5. The linear MPC controller with integral action and QIH is able to track that end-state reference with acceptable accuracy within the design envelope and constraints are respected when necessary in simulations. The simulator from Chapter 4 is considered. Robustness, vital in this application, is taken into account and demonstrated in simulations for a given set of parameter-variation scenarios. The costly nested *minimax* optimisation of typical robust MPC approaches has been rewritten as the minimisation of a scalar cost within a scenario-based algorithm. In this chapter, no trajectories have been imposed. The next Chapter 7 seeks to evaluate the tracking of pre-computed reference trajectories in state and control so as to meet the same goals and improve performance via a similar MPC formulation.

7 Synthesis of MPC controllers for planned-trajectory tracking during GG-LPRE start-up transients

In the previous chapter, the synthesis of an MPC controller for multivariable terminal-state tracking for continuous GG-LPRE transients has been presented. No trajectories during those transients were imposed, the tracking goal only concerning the desired equilibrium values. In this chapter, a further step is made. The same control methodology based on MPC is used again, with a relevant difference in terms of tracking goals. Now, tracking of pre-generated reference trajectories is demanded. These trajectories, in states and control inputs, are generated off-line via model-based optimisation too. This control approach is particularly suitable for start-up transients (described in Section 3.1.1), whose desired steady states are known a priori. In this sense, trajectories can be computed prior to operation, since this process is costly and would not be appropriate for real-time control. This approach might not be useful for throttling transients of reusable LPREs, where the tracking reference might continuously vary during operation. For those cases, the end-state controller of Chapter 6 might be more adequate, since the reference computation could be performed on-line at each time step.
This trajectory generation is again split in this chapter into the continuous and discrete sub-phases of the start-up transient. Indeed, the planning goals and model structure are different in each sub-phase. For the former, all actuators are active and the continuous ones are freely adjustable. For the latter, the computation has to deal with timing constraints, assuming that the sequence of events is to be carried out in the same safe order. The time differences between events, concerning chamber ignitions and valve openings, are the main variables to optimise in order to reach a nominal state at the end of that sub-phase. Hence, not all actuators are available at all times depending on those Δt or τ, which entails a highly nonlinear calculation. The continuous generation is presented in Section 7.1.1 and a proposal for the discrete one is made in Section 7.3. The differences in terms of the MPC algorithm used to track the continuous-phase trajectories (T.MPC) are explained in Section 7.1.2. In this case, it can be regarded as a linearised MPC since systems dynamics is considered as a linearisation along trajectories. The outcome of this modified CL is presented in Section 7.2. The developments in this chapter are included in [122] and [120].

7.1 Continuous-phase controller design

A similar loop structure as in Chapter 6 (explained in Section 6.1) is taken in this chapter, with modified control algorithm and preprocessor block, including trajectory generation. Figure 7.1.1 depicts this loop.

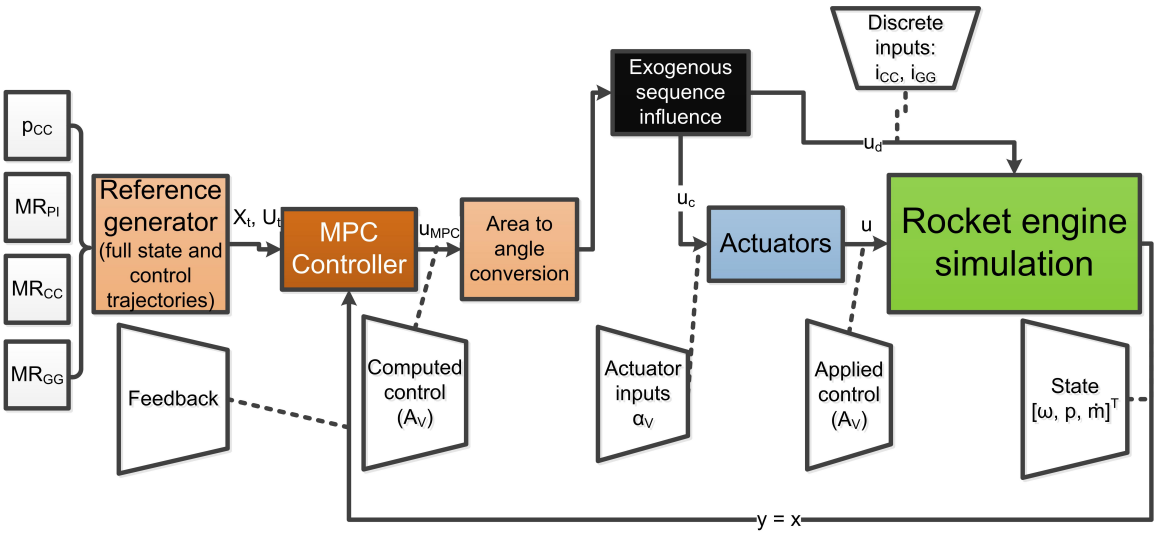


Figure 7.1.1: Control-loop diagram for trajectory tracking

7.1.1 Preprocessor block: final-reference and trajectory generation

As in Chapter 6, the preprocessing block serves as an off-line reference generator for the MPC controller. First of all, the end reference has to be constructed from the set of reference commands derived from launcher needs ($p_{CC,r}$, $MR_{CC,r}$, $MR_{GG,r}$ and $MR_{PI,r}$) by following the procedure explained in Section 6.2. 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.

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 build-up transient, considered between $1.5s$, the end of the discrete sequential phase, and $2.5s$, the desired reference-crossing time in the studied engine *Vulcain 1*. This is a common approach in the literature of trajectory planning [68, 148]. Other trajectory-planning methods have been proposed, especially in the field of robotic motion, but they are also mostly based on optimisation [15]. Here, it is of interest to take advantage of the nonlinear models developed in Chapter 5.

This algorithm is based on the minimisation of a classical discretised quadratic cost function J_{OL}, defined as:

$$J_{OL}(\mathbf{X}_t, \mathbf{U}_t) = \left(\sum_{k=0}^{N_{p,OL}} \mathbf{\Delta x}_{t,k}^T Q_{OL} \mathbf{\Delta x}_{t,k} + \sum_{k=0}^{N_{u,OL}} \mathbf{\Delta u}_{t,k}^T R_{OL} \mathbf{\Delta u}_{t,k} \right) \Delta t, \tag{7.1.1}$$

where $\mathbf{\Delta x}_{t,k} = \mathbf{x}_{t,k} - \mathbf{x}_r$ and $\mathbf{\Delta u}_{t,k} = \mathbf{u}_{t,k} - \mathbf{u}_r$ are the variables to cancel, that is to say, the distances with respect to the reference equilibrium point. Only the step index k is used in this subsection (no j) since there is no receding horizon, only one OL computation is performed. $Q_{OL} = I_n$ 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 same discretisation time $\Delta t = 10ms$ of Chapter 6.

Concerning the dynamics considered to predict the behaviour at each time step k, as shown in Chapter 5, the system is highly nonlinear. Neglecting nonlinear dynamics at points far from the equilibrium can lead to non-negligible prediction errors (Section 5.4). However, the main repercussion of imposing nonlinear constraints on optimisation problems is generally the loss of convexity of the optimised function and hence the increase in resolution complexity. The compromise chosen here, 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% (step $k = N_{p,OL_{90\%}}$). 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 simplified NLSS f_s (5.2.19)-(5.2.30) is discretised via an EULER implicit scheme:

$$\mathbf{x}_{t,k+1} = \mathbf{x}_{t,k} + f_s(\mathbf{x}_{t,k+1}, \mathbf{u}_{t,k+1}, \mathbf{w}_{t,k+1})\Delta t, \quad k \in [0, N_{p,OL_{90\%}} - 1]. \tag{7.1.2}$$

The exogenous input $\mathbf{w}_{t,k}$ is considered in these nonlinear models, having a relevant influence on dynamics. 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 (6.4.1) (the most relevant for tracking) are forced to be equal to the end-reference values. A few time steps before the start of those end-state constraints (5), a smooth transition between nonlinear and linear dynamics is set via maximum slope constraints ($\dot{p}_{CC,max} = p_{CC,r}/200$, according with a proportioned nominal build-up). Linear dynamics (A_c, B_c) stems from the end-state linearisation of f_s about ($\mathbf{x}_r$, $\mathbf{u}_r$), which is then discretised via zero-order hold at Δt (A_d, B_d):

$$\mathbf{\Delta x}_{t,k+1} = A_d(\mathbf{x}_r, \mathbf{u}_r)\mathbf{\Delta x}_{t,k} + B_d(\mathbf{x}_r, \mathbf{u}_r)\mathbf{\Delta u}_{t,k}, \quad k \in [N_{p,OL_{90\%}}, N_{p,OL} - 1]. \tag{7.1.3}$$

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

$$\min_{\mathbf{X}_t, \mathbf{U}_t} \quad J_{OL}(\mathbf{X}_t, \mathbf{U}_t) \tag{7.1.4}$$

$$\text{s.t.} \quad \mathbf{X}_t \in X_{OL}, \quad \mathbf{U}_t \in U_{OL} \tag{7.1.5}$$

$$A_{ineq,OL}[\mathbf{X}_t \quad \mathbf{U}_t]^T \leq \mathbf{b}_{ineq,OL} \tag{7.1.6}$$

$$\mathbf{x}_{t,k+1} \leq \mathbf{x}_{t,k} + f_s(\mathbf{x}_{t,k+1}, \mathbf{u}_{t,k+1}, \mathbf{w}_{t,k+1})\Delta t + \boldsymbol{\varepsilon}, \qquad \forall k \in [0, N_{p,OL_{90\%}} - 1] \tag{7.1.7}$$

$$\mathbf{x}_{t,k+1} \geq \mathbf{x}_{t,k} + f_s(\mathbf{x}_{t,k+1}, \mathbf{u}_{t,k+1}, \mathbf{w}_{t,k+1})\Delta t - \boldsymbol{\varepsilon}, \qquad \forall k \in [0, N_{p,OL_{90\%}} - 1] \tag{7.1.8}$$

$$A_{eq,OL}[\mathbf{\Delta X}_t \quad \mathbf{\Delta U}_t]^T = \mathbf{b}_{eq,OL} \qquad (including \quad \mathbf{x}_{t,N_{p,OL}} = \mathbf{x}_r). \tag{7.1.9}$$

X_{OL} and U_{OL} from (7.1.5) 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). As in (6.4.17) the set U_{OL} for the first control $\mathbf{u}_{t,1}$ is specially constrained to comply with actuators capacity.

Regarding the rest of constraints, (7.1.4) contains linear inequality constraints (7.1.6) (defined by $A_{ineq,OL}$ and $\mathbf{b}_{ineq,OL}$), for satisfying MR and actuators sectional-velocity bounds, as well as for defining a monotonically increasing pressure build-up. Maximum-MR constraints are not initially relaxed in this planning so as to enforce a feasible solution during the whole horizon. Nonlinear dynamic constraints (7.1.7) and (7.1.8) are not defined as strict equality constraints, but are treated as

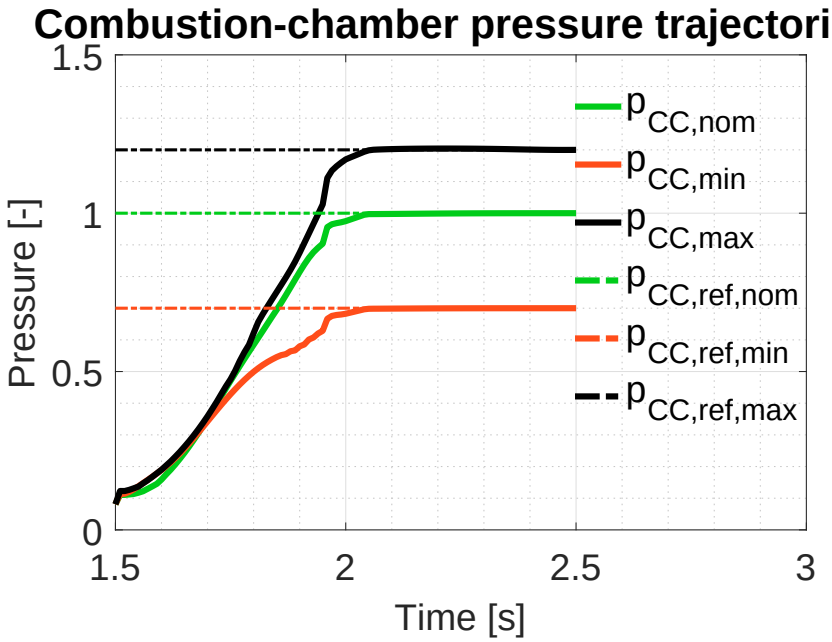


Figure 7.1.2: Trajectory planning results in p_{CC} for $p_{CC,r} = 1$ (nominal), $p_{CC,r} = 0.7$ (minimum) and $p_{CC,r} = 1.2$ (maximum)

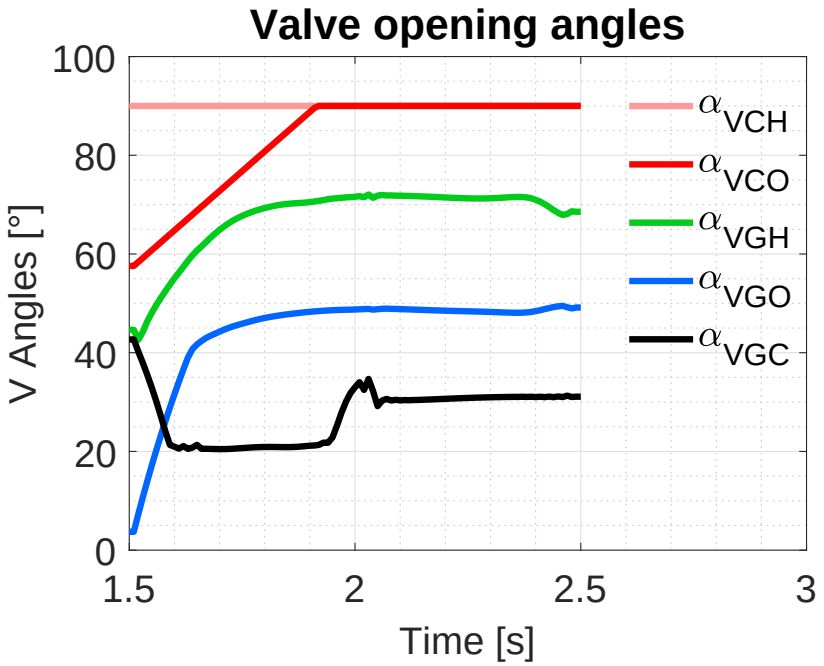


Figure 7.1.3: Trajectory planning results in $\mathbf{u}$ for $p_{CC,r} = 1$ (nominal). Maximum opening angles are $90°$.

inequalities with a small margin $\varepsilon = 10^{-2} \times \mathbf{1}_{n\times 1}$ (non-dimensional) so as to simplify the computation of a feasible solution. Linear dynamics (7.1.3), initial conditions, and end-state reaching are considered in the equality constraints (7.1.9) (defined by $A_{eq,OL}$ and $\mathbf{b}_{eq,OL}$). The end-state hard constraint forces the trajectory to precisely finish at the desired point. Figure 7.1.2 depicts the set of generated trajectories for the different operating points (70%, 100%, 120% of thrust) in terms of combustion-chamber pressures. The same optimisation tool as in Chapter 6, the interior-point optimisation software *IPOPT* [164] with equivalent settings (Section 6.5), is used to solve (7.1.4)-(7.1.9). Due to the inclusion of nonlinear constraints in this quadratic-cost optimisation problem, the solution might not be global. Even so, reference values are attained while respecting constraints.

Figure 7.1.3 presents the control trajectories for the nominal case. Initial conditions are fixed according to the end of the discrete part of the transient (until $t = 1.5s$). As mentioned in the previous chapters, the tuning of the first control inputs (VCF and VCO), related to the main chamber, is not convenient for engine control. Therefore, straight trajectories are imposed. Computation times of around 1 minute are required in *MATLAB*® .

7.1.2 MPC algorithm for trajectory tracking (T.MPC)

The main MPC structure explained in Section 6.4.2 is reused for the controller synthesis in this chapter, with certain differences so as to track the predefined trajectories ($\mathbf{X}_t$, $\mathbf{U}_t$). The main control goals, including the verification of constraints, are identical in this chapter. The aim here is to evaluate if tracking trajectories enables a more precise control of start-up transients. Thus, the target of the controller is to drive $\mathbf{X}$ and $\mathbf{U}$ to $\mathbf{X}_t$ and $\mathbf{U}_t$ respectively. Dynamics in this controller is predicted in a linearised way. Indeed, 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 in (6.4.8):

$$\mathbf{\Delta x}_{i,k+1} = A_d(\mathbf{x}_{t,k}, \mathbf{u}_{t,k}, \mathbf{w}_{t,k}, \Delta_{i,k})\mathbf{\Delta x}_{i,k} + B_d(\mathbf{x}_{t,k}, \mathbf{u}_{t,k}, \mathbf{w}_{t,k}, \Delta_{i,k})\mathbf{\Delta u}_k, \quad i \in I. \tag{7.1.10}$$

The influence of the known exogenous input $\mathbf{w}_t$ on the linearised model via B_w (defined in (5.2.31)) has been removed in these controllers since its inclusion globally led to poorer prediction performance, it mainly being a nonlinear effect on p_{GG} (x_4). However, its influence on A_d and B_d during the transient has been kept. The $\Delta_{i,k}$ correspond to the same perturbed cases defined in Section 6.4.2. The matrix A_d is always stable along the trajectory. The integral action based on [135] is also included with the same weight matrix S. Therefore, the discretised cost J_{traj} to be minimised in nominal MPC is defined as:

$$J_{traj}(\mathbf{X}, \mathbf{U}, \mathbf{Z}) = \left(\sum_{j=0}^{N_p-1} (\mathbf{x}_{k+j} - \mathbf{x}_{t,k+j})^T Q_{traj} (\mathbf{x}_{k+j} - \mathbf{x}_{t,k+j}) + \sum_{j=0}^{N_u} (\mathbf{u}_{k+j} - \mathbf{u}_{t,k+j})^T R_{traj} (\mathbf{u}_{k+j} - \mathbf{u}_{t,k+j}) \right.$$
$$\left. + \sum_{j=0}^{N_p-1} \mathbf{z}_{k+j}^T S \mathbf{z}_{k+j} \right) \Delta t + (\mathbf{x}_{k+N_p} - \mathbf{x}_{t,k+N_p})^T P (\mathbf{x}_{k+N_p} - \mathbf{x}_{t,k+N_p}). \tag{7.1.11}$$

The same prediction horizon $N_p = 10$ and control horizon $N_u = 5$ are used. Since these horizons are shorter than those used in trajectory generation (Section 7.1.1), the last step does not necessarily correspond to the end reference point. Longer horizons (more costly) did not enhance tracking or constraints satisfaction. Q_{traj} and R_{traj} are positive-definite symmetric weighting matrices $Q_{traj} \in \mathbb{R}^{n\times n}$, $R_{traj} \in \mathbb{R}^{m\times m}$, whose diagonals have also been tuned off-line via Kriging-based black-box optimisation as in [98]. The same criterion for that weight definition has been considered, which leads to different values to the E.MPC controllers from the previous chapter (listed in Section 7.2). This controller also incorporates the QIH notions from [30], with some differences concerning the linearised dynamics. The LYAPUNOV equation (6.4.3) is modified and solved every MPC step for the continuous-time matrices evaluated at the end of the horizons (N_p, N_u) and for every perturbed case $i \in I$:

$$\begin{aligned}&(A_{K,i}(\mathbf{x}_{t,k+N_p}, \mathbf{u}_{t,k+N_u}, \mathbf{w}_{t,k+N_p}, \Delta_i) + \kappa I)^T P_i + P_i(A_{K,i}(\mathbf{x}_{t,k+N_p}, \mathbf{u}_{t,k+N_u}, \mathbf{w}_{t,k+N_p}, \Delta_i) + \kappa I) \\ &\quad = -Q_K - K_i^T R_K K_i,\end{aligned} \tag{7.1.12}$$

where $A_{K,i} = A_c(\mathbf{x}_{t,k+N_p}, \mathbf{u}_{t,k+N_u}, \mathbf{w}_{t,k+N_p}, \Delta_i) + B_c(\mathbf{x}_{t,k+N_p}, \mathbf{u}_{t,k+N_u}, \mathbf{w}_{t,k+N_p}, \Delta_i)K_i$ and $\kappa \in \mathbb{R}^+$ (satisfying $\kappa < -\lambda_{max}(A_{K,i})$). The same α_P can be obtained for all cases. This promotes CL stability and feasibility in all scenarios.

Robustness is treated likewise to Section 6.4.2. Thus, the discretised minimisation problem for trajectory tracking accounting for all $\mathbf{X}_i$ is:

$$\min_{\mathbf{X}_i, \mathbf{U}, \mathbf{Z}_i, \gamma} \quad \gamma \tag{7.1.13}$$

$$\text{s.t.} \quad J_{traj}(\mathbf{X}_i, \mathbf{U}, \mathbf{Z}_i) \leq \gamma \qquad \forall i \in I \tag{7.1.14}$$

$$\mathbf{X}_i \in X, \quad \mathbf{U} \in U \qquad \forall i \in I \tag{7.1.15}$$

$$A_{ineq}[\mathbf{X}_i \quad \mathbf{U}]^T \leq \mathbf{b}_{ineq} \qquad \forall i \in I \tag{7.1.16}$$

$$A_{i,eq,traj}[\mathbf{X}_i \quad \mathbf{U}]^T = \mathbf{b}_{i,eq,traj} \qquad \forall i \in I \tag{7.1.17}$$

$$(\mathbf{x}_{i,k+N_p} - \mathbf{x}_{t,k+N_p})^T P_i (\mathbf{x}_{i,k+N_p} - \mathbf{x}_{t,k+N_p}) \leq \alpha_P \qquad \forall i \in I \tag{7.1.18}$$

$$\mathbf{z}_{i,k+1+j} = \mathbf{z}_{i,k+j} + \Delta t K_{I,traj}(\mathbf{x}_{z,i,k+j} - \mathbf{x}_{z,t,k+j}) \qquad \forall i \in I, j \in [0, N_p - 1]. \tag{7.1.19}$$

Apart from the already defined bounds X and U, in (7.1.13) there are also equality constraints (7.1.17) (defined by $A_{i,eq,traj}$ and $\mathbf{b}_{i,eq,traj}$) for initial conditions (measured $\mathbf{x}_k$) and the different linear dynamics along the whole horizon (7.1.10). Besides, the same linear inequality constraints (7.1.16) as in the E.MPC algorithm (6.4.13) are applied to all $\mathbf{X}_i$. Concerning the terminal-region inequalities (7.1.18), as defined in (7.1.12), a specific P_i is used in each perturbed case and MPC step since matrices are varying. Indeed, matrices are evaluated at the end of the horizons, which slide with time, and at each set of perturbed parameters. Integrator dynamics (7.1.19) is also analogous to Section 6.4.2, but different values for $K_{I,traj}$ have been computed off-line in the same manner as Q_{traj} and R_{traj}.

As mentioned in Chapter 6, the resulting $\mathbf{U}$ computed in (7.1.13)-(7.1.19) is forced to comply with all these uncertainty scenarios and all propagated perturbed states must verify all constraints, thereby increasing the robustness of the controller.

7.2 CL simulations and analysis of results

The main results concerning the control of the continuous sub-phase of the start-up transient via trajectory-tracking MPC (T.MPC) are presented and analysed in this section. Analogous loop and optimiser settings to Section 6.5 are selected for CL simulations of the diagram in Figure 7.1.1, implemented as in Figure 6.5.1. The MPC-parameters tuning results for these controllers via the aforementioned Kriging-based optimisation are:

- $Q_{traj} = diag(2736.63, 5082.30, 5123.46, 61.73, 349.79, 5000, 3312.76, 5000, 3518.52, 6604.94, 6728.4, 6728.4)$.
- $R_{traj} = diag(0, 0, 1419.75, 3271.60, 5000)$.
- $K_{I,traj} = diag(90.74, 32.72, 32.72, 48.77, 99.38)$.

IPOPT [164] is again chosen for solving the on-line optimisation problem (7.1.13)-(7.1.19) during the time frame between $t_{cont} = 1.5s$ and $3s$, leading to computation times similar to the previous chapter and to the same amount of variables and constraints.

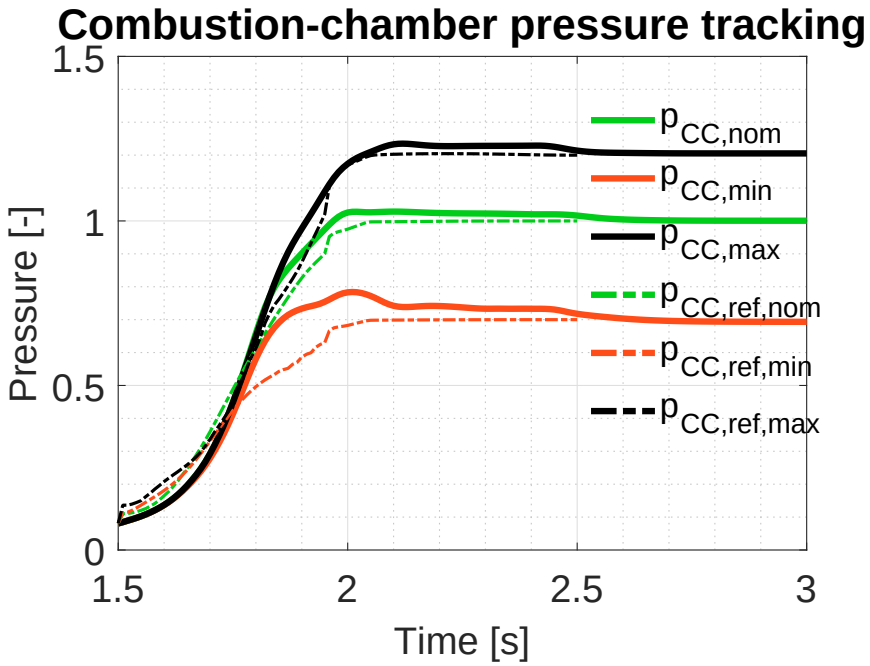
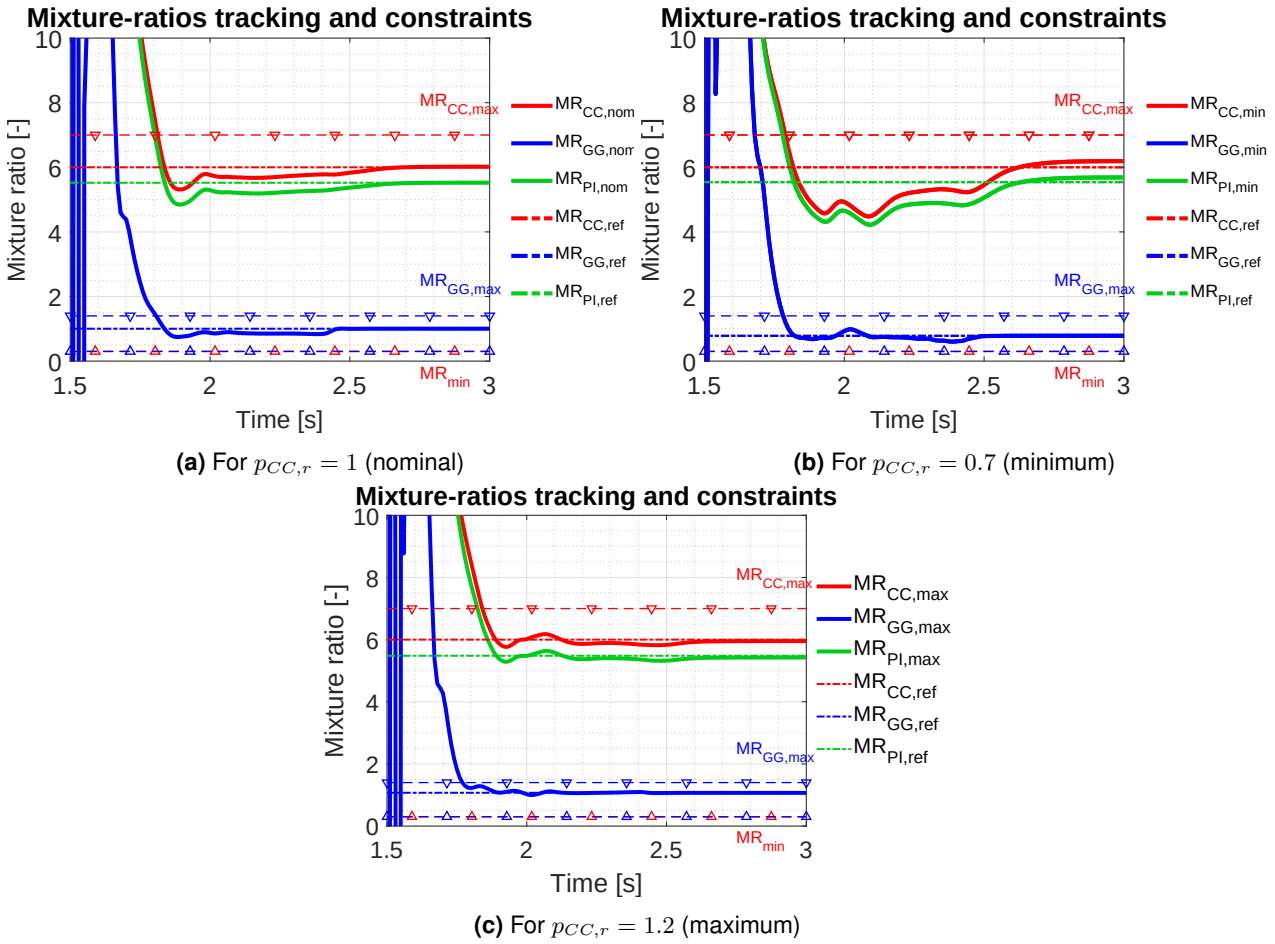


Figure 7.2.1: T.MPC tracking results in p_{CC} for $p_{CC,r} = 1$ (nominal), $p_{CC,r} = 0.7$ (minimum) and $p_{CC,r} = 1.2$ (maximum)

Figure 7.2.1 illustrates the results of p_{CC} tracking for three main operating points in *Vulcain 1* design envelope. As mentioned in the last chapter, the reference mixture ratios remain the same $MR_{CC,r} = 6$, $MR_{GG,r} = [0.8, 1.1]$ and $MR_{PI,r} = 5.25$. MR tracking for the three cases is shown in Figure 7.2.2. Tracking is accomplished with sufficient accuracy in p_{CC} for all cases (under 0.064% in nominal thrust and under 0.94% in off-nominal) and with little error in MR (under 0.32% in nominal, under 3.2% in off-nominal). Simultaneously, constraints are respected. Analogous considerations to Chapter 6 in terms of MR constraints are made here.



(a) For $p_{CC,r} = 1$ (nominal)

(b) For $p_{CC,r} = 0.7$ (minimum)

(c) For $p_{CC,r} = 1.2$ (maximum)

Figure 7.2.2: T.MPC tracking results in MR

Overshoot is again more pronounced in the minimum case due to the effects of the starter. Planned trajectories from Figure 7.1.2 do not present overshoot in order to counteract this aspect of real engine behaviour. All these performance indicators are explicitly listed in Table 7.2.1. Figure 7.2.3 shows the computed control as well as the planned trajectories. Some differences are present, especially during the first $0.5s$ in α_{VGC} (u_5), 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 present due to simplifications (Section 5.3).

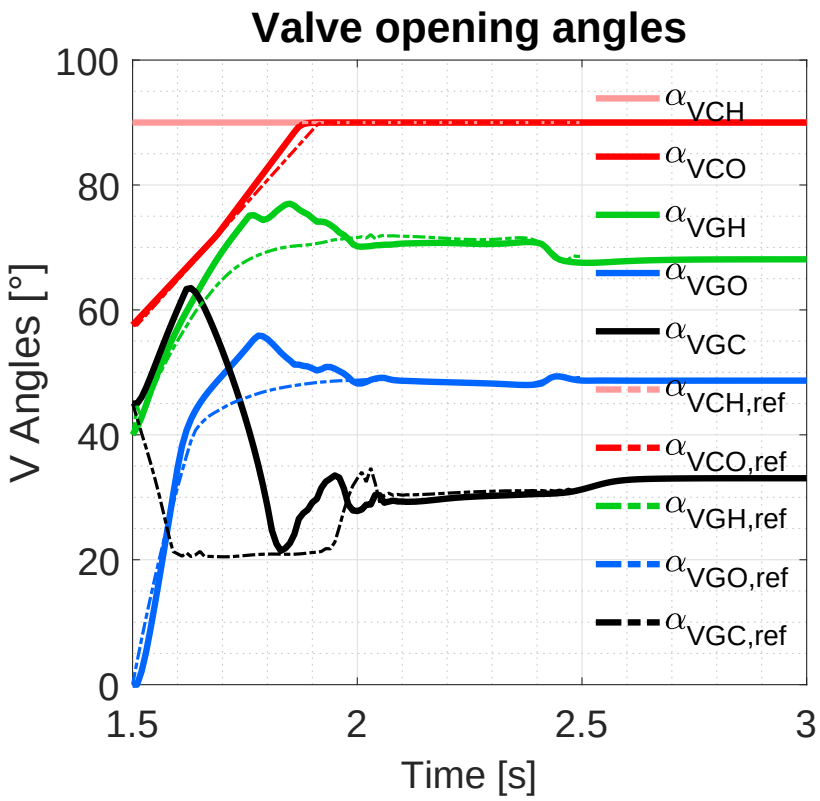


Figure 7.2.3: T.MPC control results for $p_{CC,r} = 1$ (nominal)

7.2.1 Comparison with OL and E.MPC

Table 7.2.1 summarises the comparison between this trajectory-tracking proposal (T.MPC), the previous end-state tracking E.MPC and OL simulations, in terms of the already defined performance indicators.

It seems clear that the trajectory-tracking strategy in this chapter results in a higher level of performance than OL and that it reduces overshoot with respect to the alternative terminal-tracking approach of Chapter 6. Tracking errors are similar or even greater, but the overshoot reduction leads to better overall-performance indicators for the nominal and 70% thrust cases in comparison with E.MPC. The remaining errors in MR can be due to the increase in modelling error at the off-nominal levels. One could attempt to reduce these errors with a modified performance criterion for the optimal tuning of Q_{traj}, R_{traj} and $K_{I,traj}$. Nonetheless, putting more weight on MR results in increased errors in p_{CC} and vice versa. Therefore, an equal weight has been set on both tracking goals in (5.4.5).

Table 7.2.1: Performance-indicators comparison between this T.MPC proposal, E.MPC and OL for start-up scenarios

Operating point	**Nominal start-up**			**Minimum start-up**			**Maximum start-up**		
Indicator	**OL**	**E.MPC**	**T.MPC**	**OL**	**E.MPC**	**T.MPC**	**OL**	**E.MPC**	**T.MPC**
Settling time ($p_{CC,r} \pm 1\%$) $[s]$	2.76	**2.53**	2.54	-	**2.56**	2.57	2.7	2.52	**2.51**
Overshoot (% in p_{CC})	6.29	5.11	**2.84**	15.18	15.28	**12.01**	3.43	**2.76**	2.9
Constraints verification $[s]$	**1.81**	1.85	**1.81**	1.83	1.81	**1.79**	**1.77**	1.89	1.84
p_{CC} static error (%)	0.21	**0.019**	0.064	2.67	**0.82**	0.94	0.45	**0.23**	0.43
MR_{CC} static error (%)	**0.18**	0.87	0.32	**2.66**	3.6	3.2	3.25	**0.041**	0.8
MR_{GG} static error (%)	1.41	**0.0036**	0.069	1.35	**0.21**	0.23	1.28	0.94	**0.31**
MR_{PI} static error (%)	1.41	0.55	**0.022**	2.92	2.99	**2.62**	3.48	**0.11**	1.04
J_{perf} overall performance	4.16	2.43	**1.27**	11.45	9.26	**8.43**	9.54	**2.36**	3.49

7.2.2 Robustness analysis

The same perturbed cases from Section 6.4.2 are considered in this analysis. The performance indicators obtained with this T.MPC controller (7.1.13)-(7.1.19) for attaining a nominal-thrust start-up are shown in Table 7.2.2. Comparisons with OL and E.MPC are also included.

It is observable that robustness is again demonstrated in all scenarios by means of the tracking approach in this chapter. Figure 7.2.4 depicts the T.MPC-controlled simulation corresponding to the worst-case-scenario start-up (to 100% thrust).

Results point to the moderately greater robustness to parameters variations of the approach presented in this chapter in comparison to the E.MPC from the previous chapter, and considerably greater with respect to OL. Comparing both CL approaches, the enhancements of T.MPC in terms of static errors, settling time and constraints verification are not elevated. The noticeable reduction of overshoot is the major advantage, which improves the overall-performance indicators in all cases.

Further robustness tests can be performed by introducing other types of perturbations, not explicitly considered in the developed control algorithm. For instance, the effects of external, additive perturbations in the state during the nominal start-up are depicted in Figure 7.2.5. In that case, a band-limited white-noise block has been added to the simulator differential equations before integration (as in (6.4.9)), with a power spectral density corresponding to perturbation spans of $\pm 10\%$ of the nominal

Table 7.2.2: Performance-indicators comparison between T.MPC, E.MPC and OL in perturbed scenarios for start-up transients

Perturbed cases	**Case 1: nominal**			**Case 2: p_{PI} increased by 10%**			**Case 3: p_{PI} decreased by 10%**			**Case 4: worst case**		
Indicator	**OL**	**E.MPC**	**T.MPC**	**OL**	**E.MPC**	**T.MPC**	**OL**	**E.MPC**	**T.MPC**	**OL**	**E.MPC**	**T.MPC**
Settling time ($p_{CC,r} \pm 1\%$) $[s]$	2.76	**2.53**	2.54	2.8	2.55	**2.54**	2.74	**2.52**	2.54	-	**2.5**	2.51
Overshoot (% in p_{CC})	6.29	5.11	**2.84**	6.52	4.85	**2.96**	6.08	4.96	**2.79**	4.28	5.9	**2.67**
Constraints verification $[s]$	**1.81**	1.85	**1.81**	**1.8**	1.84	1.81	**1.81**	1.86	1.82	**1.8**	1.85	1.82
p_{CC} static error (%)	0.21	**0.019**	0.064	0.38	**0.11**	0.16	0.062	0.066	**0.02**	1.75	**0.46**	0.48
MR_{CC} static error (%)	**0.18**	0.87	0.32	**0.38**	1.05	0.49	**0.02**	0.68	0.14	1.25	1.11	**0.44**
MR_{GG} static error (%)	1.41	**0.004**	0.069	1.39	**0.002**	0.09	1.42	**0.009**	0.04	1.62	1.33	**1.21**
MR_{PI} static error (%)	1.41	0.55	**0.022**	1.21	0.72	**0.19**	1.61	0.36	**0.15**	2.97	0.57	**0.056**
J_{perf} overall performance	4.16	2.43	**1.27**	4.25	2.82	**1.69**	4.17	2.18	**1.22**	8.57	4.52	**3.03**

state, and a correlation time of $10^{-5}s$ (as in the simulator). The resulting performance criterion is $J_{perf} = 2.33$, not particularly degraded.

Another relevant robustness analysis, related to the future estimation block (commented in Perspectives), is the consideration of the possible effects that a state estimator would introduce. Namely, a unitary delay (of $10ms$) and an estimation error source have been inserted in the measurement fed back to the controller. That error source is considered as a set of independent, uniformly distributed random numbers that multiply the feedback. This is aimed at simulating the estimation error, which is assumed to remain within $\pm 1\%$ of the true values. Figure 7.2.6 shows the respective results for the nominal start-up, with a performance criterion of $J_{perf} = 2.16$.

Finally, a robustness test combining all the considered sources of disturbance simultaneously, namely internal parametric variations (worst-case), external state perturbations and estimator influence, has been simulated, obtaining the results in Figure 7.2.7. A $J_{perf} = 3.45$ is achieved. Thus, it can be claimed that the proposed loop is robust to all these scenarios.

Remark:
A smaller integration time step ($10^{-6}s$) is required in these latter complex simulations.

More degraded scenarios, involving for example actuator faults or other radical dynamic modifications related to component malfunction, would require the use of reconfiguration techniques such as [140]. Further discussion on this topic is provided in Perspectives.

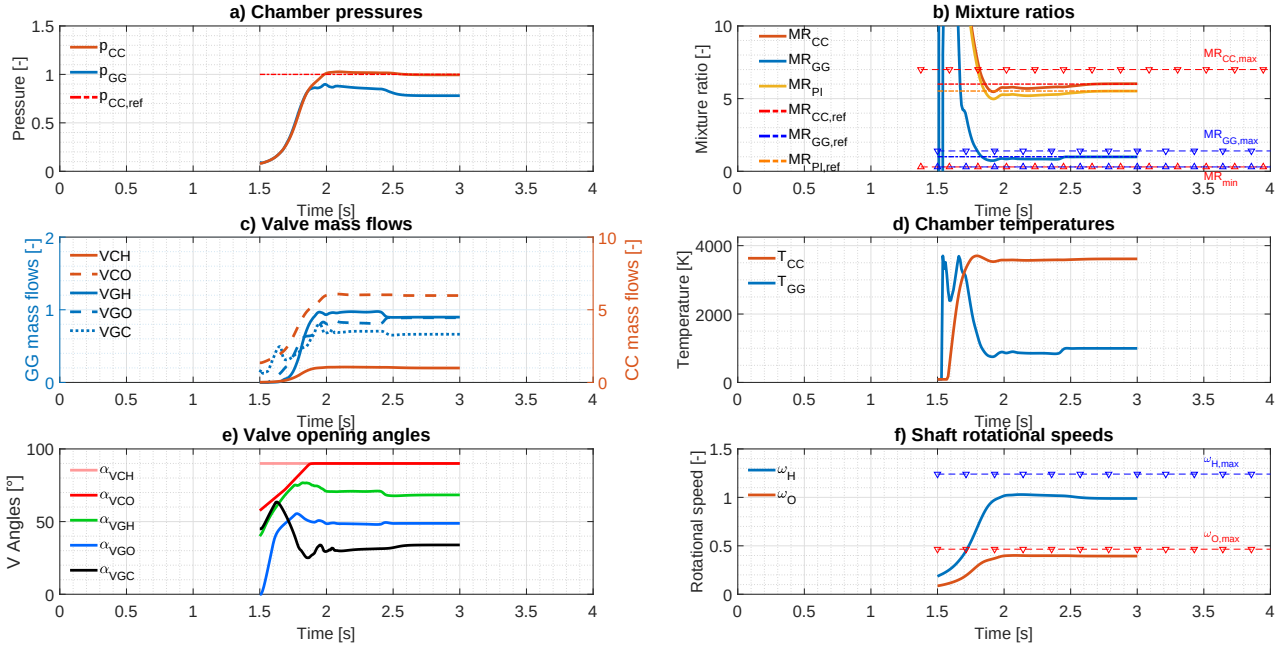


Figure 7.2.4: Complete T.MPC-controlled results in worst-case scenario 4 for a start-up to $p_{CC,r} = 1$ (nominal)

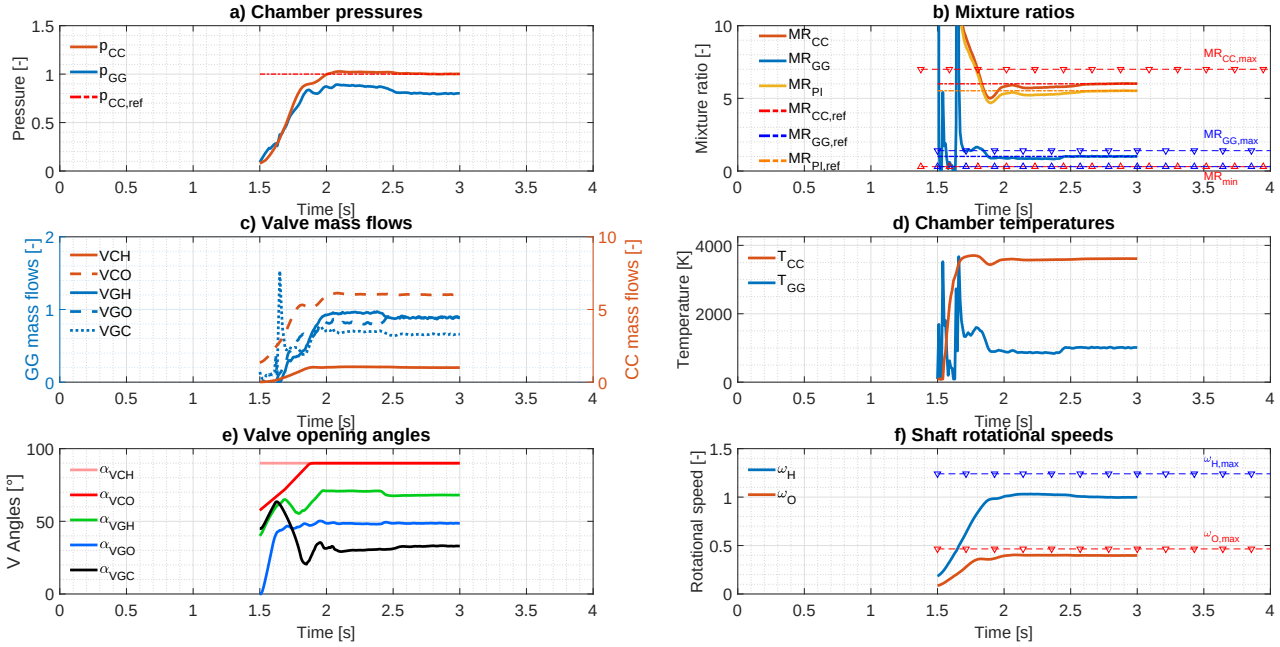


Figure 7.2.5: Complete T.MPC-controlled results with external state perturbations for a start-up to $p_{CC,r} = 1$ (nominal)

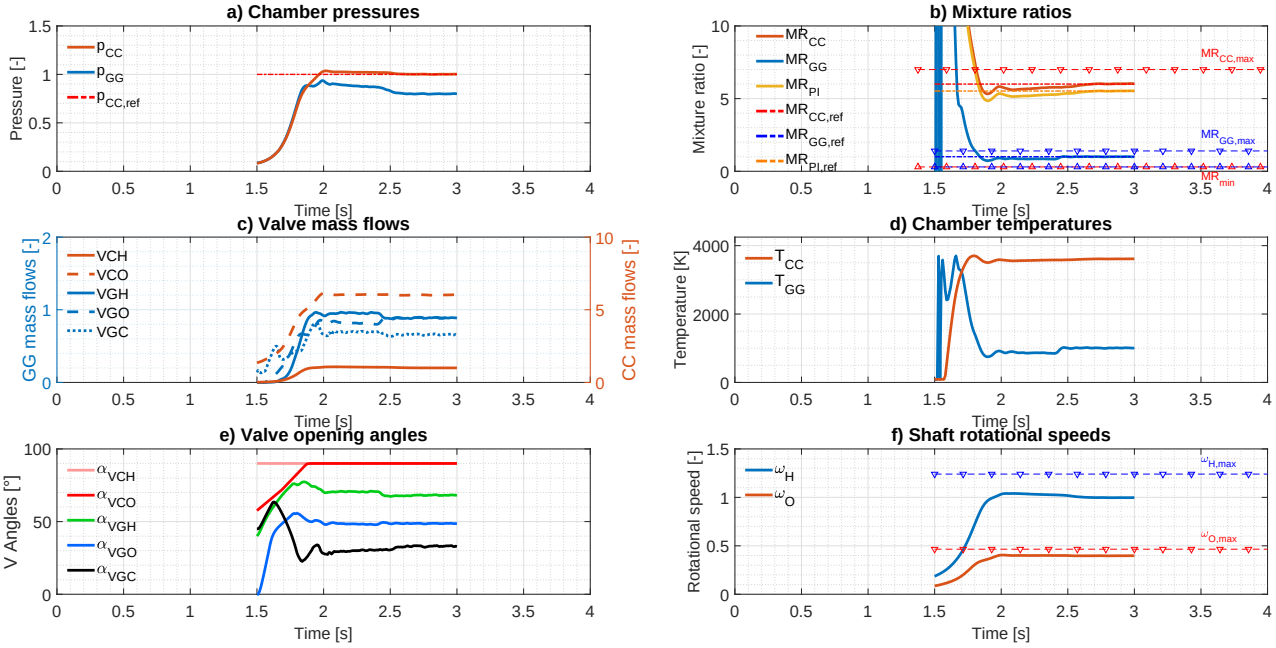


Figure 7.2.6: Complete T.MPC-controlled results with estimator influence for a start-up to $p_{CC,r} = 1$ (nominal)

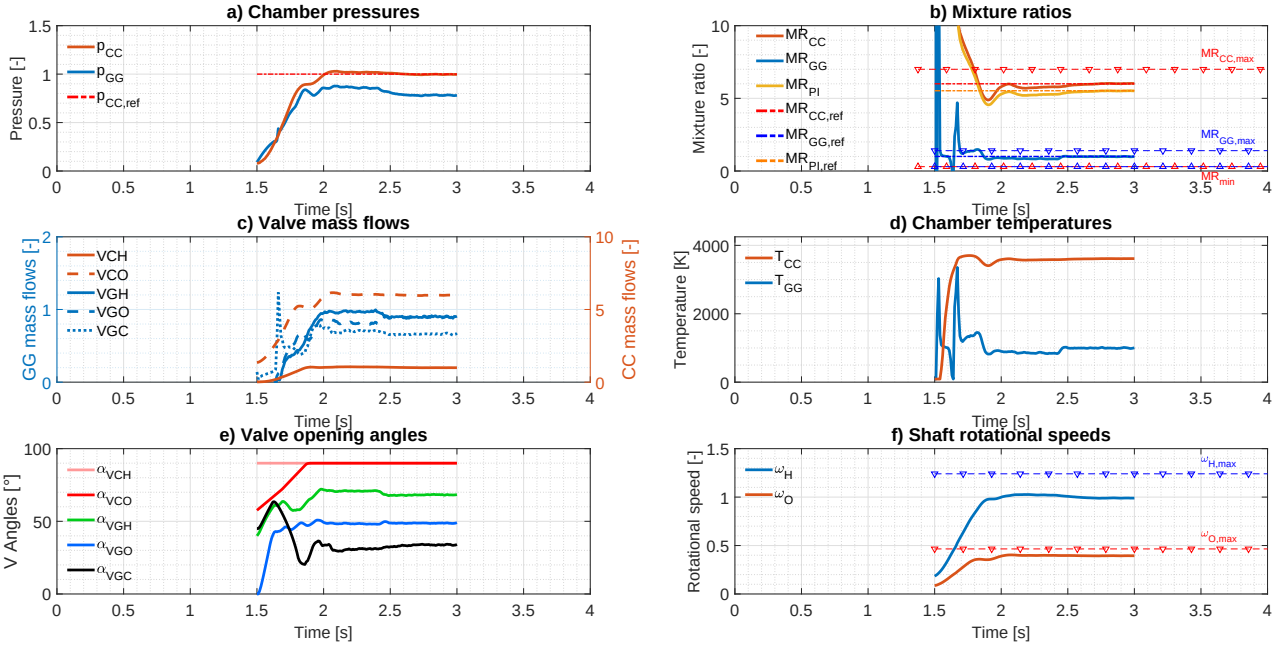


Figure 7.2.7: Complete T.MPC-controlled results with all considered sources of disturbance for a start-up to $p_{CC,r} = 1$ (nominal)

7.3 Discrete-phase trajectory-planning proposal

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 $\mathbf{u}$. In practice, all these events are normally executed at certain instants which are precisely tuned for nominal conditions. As a reminder, ignition and starter activations are modelled here as discrete inputs which activate combustion-related terms in differential equations (5.2.19)-(5.2.30).
Building upon the tools developed in this thesis, a strategy for optimising the time differences between those 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 exploit the controllability of valves, which otherwise are simply opened to pre-defined degrees. The nominal order of events (from Section 4.3) is maintained, since it consists in a safe succession of actuations according to fuel and oxidiser properties [55, 126].
An optimisation problem based on MPC principles is again envisaged. An OL finite-horizon scheme (with $N_{p,d}$ and $N_{u,d}$ as horizons), 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 5.4. 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 1*, they are:

$$\boldsymbol{\tau} = [\tau_{VCF} \quad \tau_{VCO} \quad \tau_{i_{CC}} \quad \tau_{i_{GG}} \quad \tau_{VGF}]^T. \tag{7.3.1}$$

The definition of these intervals is graphically shown in Figure 7.3.1.

The opening of the last valve (VGO) is considered at a fixed time ($t_{continuous} = 1.5s$), 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 (the same $\Delta t = 10ms$ is considered):

Initialise $\mathbf{X}_{NL}$ *and* $\mathbf{U}_{off}$ *as void matrices*;
for $k = 0, 1, \ldots, N_{p,d} - 1$ **do**
  $t_k = t_0 + (k+1)\Delta t$;
  $[\mathbf{u}_{c,k,off}, \mathbf{u}_{d,k+1}] =$ *discrete_ tree* $(\boldsymbol{\tau}, t_k)$;
  $\mathbf{x}_{d,k+1} = f_s\left(\mathbf{x}_{d,k+1}, \mathbf{u}_{c,k+1}, \mathbf{u}_{d,k+1}, \mathbf{w}_{t,k+1}\right)\Delta t + \mathbf{x}_{d,k}$;
  Append $\mathbf{x}_{d,k+1}$ *to* $\mathbf{X}_{NL}$;
  Append $\mathbf{u}_{c,k,off}$ *to* $\mathbf{U}_{off}$;
end

Algorithm 1: Definition of hybrid nonlinear dynamic constraints

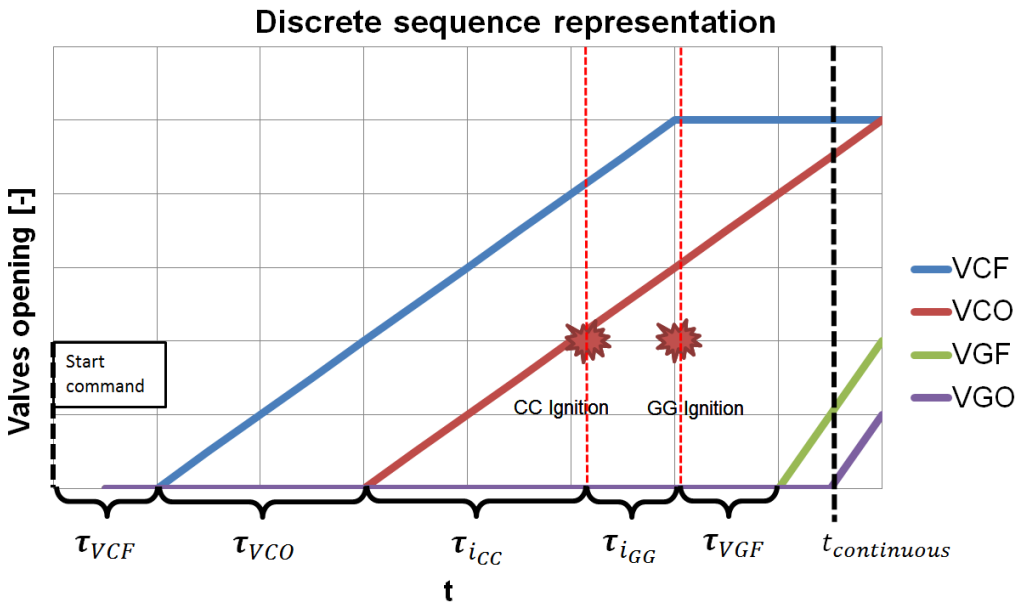


Figure 7.3.1: Representation of considered time intervals during *Vulcain 1* discrete sequence

The function *discrete_tree*, consisting of a tree of logical statements, provides the value of discrete inputs $\mathbf{u}_d$ (i_{CC}, i_{GG} and i_{sta}) according to the current time intervals and instant. If the current instant t_k surpasses the sum of the respective intervals $\boldsymbol{\tau}$, 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 (simplified NLSS (5.2.19)-(5.2.30)) adapts according to the discrete inputs, and an implicit backward EULER scheme is proposed so as to define dynamic constraints $\mathbf{X}_{NL}$. In this manner, the changing dynamics are expressed via these implicit nonlinear constraints.

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

$$\min_{\mathbf{X}_d, \mathbf{U}_d, \boldsymbol{\tau}} \quad \boldsymbol{\Delta}\mathbf{x}_{d,N_{p,d}}^T Q_d \boldsymbol{\Delta}\mathbf{x}_{d,N_{p,d}} \tag{7.3.2}$$

$$\text{s.t.} \quad \mathbf{X}_d \in X_d, \quad \mathbf{U}_d \in U_d \tag{7.3.3}$$

$$A_{ineq,d}\left[\mathbf{U}_d\right] \leq \mathbf{b}_{ineq,d} \tag{7.3.4}$$

$$A_{eq,d}\left[\mathbf{X}_d \quad \mathbf{U}_d\right]^T = \mathbf{b}_{eq,d} \tag{7.3.5}$$

$$\sum \boldsymbol{\tau} \leq t_{continuous} \tag{7.3.6}$$

$$|\mathbf{X}_d - \mathbf{X}_{NL}| = \mathbf{0} \tag{7.3.7}$$

$$\mathbf{U}_{off} = \mathbf{0}. \tag{7.3.8}$$

Bounds, given by $X_d \in \mathbb{R}^{n(N_{p,d}+1)}$ and $U_d \in \mathbb{R}^{mN_{u,d}}$, are again considered in (7.3.3). In inequality constraints (7.3.4) (defined by $A_{ineq,d}$ and $\mathbf{b}_{ineq,d}$), in contrast to the previous (7.1.16), only actuators limits are constrained, since mixture ratios are not meaningful if some valves are closed. Equation

(7.3.5) (defined by $A_{eq,d}$ and $\mathbf{b}_{eq,d}$) fixes the initial conditions of the problem (engine off, all inputs off), but not dynamics, which is constrained in (7.3.7) by running Algorithm 1. The sum of all time differences τ must not exceed the beginning of the continuous phase $t_{continuous}$ (7.3.6). In (7.3.8) 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 5.4), 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.

Due to the numerous nonlinearities, high coupling and implicitness of constraints, the optimisation problem (7.3.2)-(7.3.8) becomes highly non-convex. The software *IPOPT* [164] is not well suited for these computations, which might lose their twice-continuously-differentiable properties due to the non-smooth discrete events (explained in Section 3.3.3). Thus, this approach remains to be numerically evaluated with other numerical tools so as to carry out extensive simulations. Some perspectives of this study can be found in Chapter 8.

7.4 Summary

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. Building upon the end-state-tracking MPC developed in Chapter 6, another tracking strategy has been evaluated in this chapter for the control of start-up transients. Concretely, the on-line tracking of pre-computed reference trajectories in states and control inputs has been tackled. A major initial distinction has been made according to the nature of each sub-phase of these start-up transients. Indeed, there is an initial phase which comprises a series of discrete events, consisting in valves openings and chamber ignitions. Once this phase is over, the system behaves in a completely continuous manner. Control goals and system structure are therefore different at each sub-phase. Concerning the continuous sub-phase, reference trajectories are computed in a preprocessor which takes into account the full-state terminal reference constructed according to launcher commands, explained in Chapter 6. For this sake, a nonlinear model-based optimisation scheme, which can be regarded as OL finite-horizon MPC, has been used. All the main constraints which the on-line controller has to face are already set in that scheme. The on-line linearised MPC controller with integral action and QIH achieves trajectory tracking with sufficient accuracy within the design envelope and constraints are satisfied during the continuous start-up transient. The on-line consideration of nonlinear dynamics has been explored in the development of these MPC controllers. However, linearisation about trajectories has proved to be more adequate for the on-line resolution of the optimal control problem. The same robustness considerations from Chapter 6 are included in the controller so as to face a set of internal-parameter variation scenarios, plausible in real engine operation. The controller demonstrates robustness against the predefined worst-case scenarios, since performance is not degraded. In addition, enhanced performance indicators are obtained with respect to Chapter 6. Robustness against external state

perturbations and possible estimator influence, introduced simultaneously with worst-case parametric variations, is also shown in simulations.
With regard to the discrete sub-phase, an algorithm for optimising time differences between events and valve opening sections has been proposed, also based on a constrained nonlinear programme. The approach here mimics in a relatively precise way the hybrid physics of that complex problem, where the model structure changes after each event. Concretely, the timing determines which control inputs are active. The proposed cost concerns the reaching of the nominal initial state of the continuous sub-phase. Due to the numerous nonlinearities, coupling and implicitness of constraints, the optimisation problem becomes highly non-convex and might lose second-order differentiability. General non-convex optimisation solvers should be employed in order to find local optimal solutions of the proposed problem. However, other ways of expressing the problem in a more simplified manner could be envisaged. Thus, this approach remains to be numerically evaluated with extensive simulations. Some future-work paths are proposed in Chapter 8.

Part III

Concluding part

Conclusions and Perspectives

8

8.1 Conclusions

This thesis tackled the field of convergence between the liquid-propellant rocket engines (LPREs) and automatic-control disciplines. It consisted in developing a control loop which is adapted to the whole set of operating phases of gas-generator-cycle LPREs, transient and steady-state, and which is robust to internal parametric variations. The state of the art of LPREs control was reviewed in **Chapter 2** and in the publication [121]. Namely, closed-loop control of LPREs has been achieved around their steady states via linearised models. Besides, robust approaches confronting uncertainties and certain faulty scenarios are present in the literature. Damage mitigation in certain components has also been dealt with. Sequential transient phases of engine operation (start-up and shutdown) are generally executed in open loop with narrow correction margins due to the controllability and observability issues at very low mass flows. Indeed, 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. There is also a lack of method comparisons on a common benchmark, even simulated. In addition, only narrow throttling domains are feasible.

Several blocks have been developed to constitute the control loop in this thesis: engine simulation, reference generation and controllers. In **Chapter 4**, simulators representative of the thermo-fluid-dynamic behaviour of gas-generator-cycle (GG) engines were built. The reason for the choice of this type of cycle is that the European reusable engine currently under development, *PROMETHEUS*, will present it. The main case study for modelling and control in this thesis has been the *Vulcain 1* engine, while *PROMETHEUS* has only been modelled. Zero-dimensional models of each main engine component were selected from the literature or developed in order to construct a *Simulink*® library adapted to LPREs. These sub-models were then assembled to build the transient simulator based on ODEs, concerning the thermo-fluid-dynamic and mechanical conservation equations. No high-frequency transient phenomena, which could not be counteracted by the main actuators of the cycle (valves), are explicitly considered in models. The capacity of this simulator to capture transient phases was then validated. The whole start-up transient is satisfactorily predicted, with the exception of the less accurate temperature behaviour.

The purely thermodynamic modelling of the cycle was subsequently adapted to the control framework in **Chapter 5** with the goal of applying model-based methods. Nonlinear state-space models of the same engines were derived. In order to do so, the component models defined were symbolically joined. That is to say, all differential and algebraic thermodynamic equations were causally combined according to engine's flow plan. The manner in which the hybrid features of the system were treated, involving continuous and discrete actuators, was chosen according to the hybrid-system modelling literature. Discrete elements, concerning the igniter and starter actuators, are considered as part of the control inputs and manage certain terms in differential equations (based on the Mixed Logic

Dynamical paradigm [10]). The rest of control inputs, which are continuous, consist in valve opening sections. State variables are shaft rotational speeds, cavity pressures and mass flow rates through valves and pipes. Measurements of the entire state vector are assumed to be available. These GG-LPRE state-space system representations for control purposes with time-varying coefficients and discrete inputs are one of the main contributions of this thesis. It was then assessed to what extent these transient models were suitable for the synthesis of control laws, they being nonlinear and non-affine with respect to control inputs. Physical and mathematical simplifications had to be carried out in order to obtain a tractable model, which can also be linearised. A compromise between simplicity and system representativeness was reached. A comprehensive analysis of models response and characteristics was then conducted, leading to the verification of natural stability, controllability and observability. Sensitivity analyses allowed to identify the most influential parameters and the worst-case scenario of plausible parametric variations, relevant for the controller synthesis. The main contributions from this chapter and from Chapter 4 are included in [118]. The conference paper [120] contains a summary of models analysis.

These derived models were subsequently employed for synthesising controllers. In **Chapter 6**, their complex nonlinear form is used to compute off-line a complete steady-state reference as a function of the selected final combustion-chamber pressure (related to thrust) and mixture ratios. This reference, in terms of states and control inputs, already serves as an OL control valid for off-nominal thrust levels. On-line control goals have not only concerned multivariable tracking, but also hard-constraints verification and robustness to parametric variations at a fair computational cost. Owing to the complexity of the models and in order to satisfy all these goals, Model Predictive Control (MPC) was selected as the most appropriate method. In that first chapter devoted to controllers synthesis, end-state-tracking linear MPC (E.MPC) with integral action and a quasi-infinite horizon (QIH) was developed, aiming at the control of two types of transient phases. These phases are throttling operations and the continuous sub-phase of the start-up, which begins once all the discrete events that define the sequential sub-phase (ignitions and valve openings) have taken place. Valves sections are continuously tuned for tracking the steady-state reference in terms of states and control inputs, especially in terms of the pressure in the main chamber and the injected mass flows, while respecting operational constraints. Robustness to variations in the parameters, which are checked to be predominant according to sensitivity analyses from Chapter 5, is accomplished via a scenario-based robust-MPC algorithm. Good tracking performance in simulations (of Chapter 4) is attained in both types of transient phases with respect to OL and other linear controllers (PID, LQR), but overshoot and static errors along the start-up transient can still be improved with the control approach from Chapter 7. These contributions are presented in [119, 120].

In **Chapter 7**, another more precise tracking strategy for start-up transients was assessed. Specifically, the on-line tracking of pre-computed reference trajectories (in states and control inputs) with an analogous MPC scheme was evaluated (T.MPC). Regarding the generation of those trajectories, different approaches were pursued according to the nature of each sub-phase of the transient. In fact, control goals and system structure are different at each sub-phase. Concerning the continuous sub-phase, trajectories are computed in a preprocessor which accounts for the terminal reference built according to launcher commands from Chapter 6. To do so, a nonlinear model-based optimisation scheme, which can be regarded as OL finite-horizon MPC, was employed. All the main constraints which the on-line controller has to satisfy are already enforced within that computation, which results

in a safe OL trajectory. The on-line linearised MPC controller, designed in a manner similar to Chapter 6, accomplishes trajectory tracking in simulations (Chapter 4) with sufficient accuracy within the design envelope and constraints are satisfied during the continuous start-up transient. The controller, presenting the same robustness considerations as in Chapter 6, demonstrates robustness not only against the predefined worst-case scenarios, but also against simultaneous external state perturbations and a possible estimator impact. In addition, performance indicators are enhanced with respect to E.MPC. An automatic tuning of MPC parameters, aimed at optimising an overall-performance criterion, was considered in all controllers. This enables an objective comparison of approaches.

From the application point of view, the tracking of a previously calculated terminal state (E.MPC) would be more appropriate for throttling transients, in which the reference may vary quickly. Trajectory tracking T.MPC is more adequate for the start-up transient, since the generation of reference trajectories is more computationally costly, and these phases would generally allow sufficient computation time prior to operation.

A trajectory-generation strategy for the discrete sub-phase of the start-up has also been proposed in Chapter 7. This would robustify this phase with an a priori safe trajectory, seeing that on-line controllability and observability are limited. The proposal consists in optimising time differences between events and also valve opening sections by means of another constrained nonlinear programme. In this case, the evolving nonlinear dynamics is taken into account, since the model structure changes after each event. Concretely, the timing determines which control inputs are active. Further discussion on this on-going work area is included in the following Perspectives Section.

8.2 Perspectives

The continuation of this work is possible in the different disciplines covered. Some perspectives have already been mentioned in Section 2.6 as a conclusion to the literature review. It has also been hinted that the control loop for the *PROMETHEUS* engine, only modelled, could be completed once more consolidated data are available by following the same methodology applied to *Vulcain 1*.

The immediate question that might come to the mind of the reader is how these developed control algorithms could be applied to test benches or even in real engines. So far, they have been implemented within the framework of *MATLAB-Simulink®*. Computational times of the developed MPC schemes, run within that software and the *IPOPT* optimiser [164], are about ten times the real ones. Hence, a translation of the code into an embeddable language (such as *C++*) would probably allow the real-time control of the engine. However, other necessary steps prior to that implementation would consist in testing the control algorithms on more precise LPRE simulators, built with multidisciplinary simulation tools such as *EcosimPro®* or *CARINS* [116]. In the subsequent step, representativeness could be augmented by conducting hardware-in-the-loop simulations in the ISFM bench [127], for instance. In this manner, control laws could be tested on realistic actuators. If controllers are validated in those simulations, their implementation in complete test benches could be performed. Regarding their implementation in flying engines, a longer validation procedure would obviously be necessary, provided that it is decided to augment the on-board computing power. In current engines, it may not

be possible to solve nonlinear programmes in real time. In all cases, a specific integration analysis will have to be realised, concretely on the numerical computation requirements and processing-modules organisation.

With regard to the modelling of LPREs, several improvement areas can be discussed. State-space models of other cycles such as expander and staged-combustion ones could be conceivable with this modelling approach. The main weakness of the nonlinear models developed in this thesis is the transient temperature-behaviour prediction. A more complex modelling of temperature dynamics could be conducted within the simulator for increasing representativeness, taking into account more accurate injection, ignition and combustion formulations [20, 96]. The ARRHENIUS equations did not appreciably increase accuracy with the considered differential equations: continuity for $\frac{d\rho}{dt}$ and energy conservation for $\frac{dp}{dt}$. Temperatures here are dependent on those densities and pressures. A possible alternative would be to consider the continuity equation for pressures and the energy one for temperatures, together with a more accurate combustion model. Beyond combustion chambers, further thermal dynamics to account for could be the cooling-circuit transient behaviour, which might be more relevant in other cycles such as the expander one. However, the impact of including more states in the subsequently derived state-space models could be penalising for control purposes. The modelling enhancement might have to remain within the simulator. Hence, it may be more appropriate to directly connect the controllers developed via the methodology in this thesis to multidisciplinary precise simulators, as mentioned before.

Other modelling aspects to be improved, which are simplified in this thesis due to the absence of more precise knowledge on their characteristics, might be turbopump and valve dynamics. Their more accurate off-design modelling, especially of turbopumps [9], in which operating points very far from the nominal would be more precisely considered, could turn out to be highly relevant for control. Indeed, the control of those off-nominal points would become more robust with better predictions. In the same sense, tracking errors at those extreme points could be reduced with more specific parametric estimations at those levels. Many varying parameters are generally estimated for the nominal behaviour. Besides, if damage on the main components is also modelled and taken into account within the controllers [37, 42, 89], robustness to ageing would increase.

Regarding the measurability of mass flows, assumed to be complete in the derived state-space models and control approaches, an adaptation to real engines should be performed. The lack of measurements of these quantities in flying LPREs would have to be mitigated by estimation, which is conceivable due to the observable nature of the corresponding state-space models (Section 5.4). Some advanced estimation solutions specific to LPRE systems could be envisaged by looking at the work of SAROTTE [137–140], in which fault detection, isolation and reconfiguration control approaches are also derived. That research trend, aiming at developing more sophisticated HMS strategies for LPREs, could also be coupled with the control studies of this thesis.

Some perspectives particularly related to control approaches are the following. Concerning MPC, the on-line consideration of nonlinear dynamics has been explored in the development of these MPC controllers. However, linearisation about trajectories has proved to be more adequate for the on-line resolution of the optimal control problem. Perhaps, if other optimisation solvers were used, nonlinear dynamics could effectively be considered. Promising efficient MPC frameworks, making the bridge between *MATLAB®* and *C/C++*, are *MATMPC* [31], *ACADO* [54] and *CVXGEN* [99].

Robustness in these algorithms might also be extended to larger perturbation sets via LMI-based optimisation [67, 83] if the computation is efficient enough for real-time control, which has proved to be problematic with the control problems in this thesis. Another approach which could robustify the loop is the interval algebra for parameter variations [3]. The consideration of the allowable variation ranges for parameters in the integration of system dynamics allows to have a validated simulation, in which all possible propagations are covered in regions. This approach can also be blended with the nonlinear MPC so as to robustify the dynamic constraints considered. For this sake, Branch&Bound and Branch&Prune algorithms have to be annexed. For any choice of solver, an analysis of its computational requirements, in terms of required memory and FLOPS, will be necessary for studying its integration in real machines.

Beyond MPC, robust variable-structure control methods, such as the ones in [64], may be applicable for simplified models of the engine. In that paper, sliding-mode schemes with chatter reduction are described, in which the variable structure guarantees robustness via disturbance estimation and switching functions.

As for the hybrid optimisation proposal made for the first sub-phase of the start-up transient, the successful resolution of the problem is still to be obtained. Due to the numerous nonlinearities, coupling and implicitness of constraints, the optimisation problem becomes highly non-convex and might lose second-order differentiability. General non-convex optimisation solvers alternative to *IPOPT* [164], such as *BARON* [157] or *KNITRO* [24], could be assessed in order to find local optimal solutions of the proposed problem. The automatic convexification of the programme would also be an option for coping with its resolution. For example, the *CVXGEN* utility [99] could be an appropriate tool for that sake. That tool could also be used for fast computation of on-line MPC. However, other ways of expressing the problem in a more simplified manner could also be envisaged. For instance, system dynamics could be modelled via time-delay models, which might help to simplify dynamic constraints.

The resolution of this hybrid problem would not only be beneficial for the main focus here, start-up control, but also for the other main sequential transient, the shutdown. This operation is greatly affected by the events timing too, consisting in valve closings in this case. There is also a short continuous sub-phase after the shutdown command and before the execution of the first event. Nonetheless, it is less clear if a low thrust level should be tracked before events take place, or directly a hybrid consideration of the problem driving the engine safely to zero thrust should be made. Controllability and observability issues may still be present at low mass flows and with some closed valves. Therefore, the whole multidisciplinary conception of the shutdown operation, and also of the start-up one, might have to be rethought if a robust on-line controlled transient, possibly via HMPC (Section 3.3.4), is desired. Possible design solutions could be the priming of lines with secondary flow [115], or the inclusion of further sensors and actuators. Otherwise, only an off-line planning of timing and control-trajectories could be conceivable. Computational times will in any case remain the most constraining limit to be faced.

Appendix A: *PROMETHEUS* complex nonlinear state-space model

$$
\begin{aligned}
\dot{\omega} = i_{sta} \Bigg[& \left(a_{1T_0} + RadT_0\omega \left(a_{2\,T_0} + \frac{RadT_0\, a_{3\,T_0}\omega}{\sqrt{\gamma_{outGG_{fp,x}} R_{outGG_{fp,x}} T_{GG_{fx}}}} \right) \left(\sqrt{\gamma_{outGG_{fp,x}} R_{outGG_{fp,x}} T_{GG_{fx}}} \right)^{-1} \right. \\
& + AthLE_p \left(a_{4\,T_0} + \frac{AthLE_p a_{5\,T_0}}{AthT_p} \right) {AthT_p}^{-1} + \frac{a_{6\,T_0} AthLE_p RadT_0\omega}{AthT_p \sqrt{\gamma_{outGG_{fp,x}} R_{outGG_{fp,x}} T_{GG_{fx}}}} + a_{7\,T_0} \ln\left(\frac{AthLE_p}{AthT_p} \right) \\
& \left. + a_{8\,T_0} \ln\left(\frac{RadT_0\omega}{\sqrt{\gamma_{outGG_{fp,x}} R_{outGG_{fp,x}} T_{GG_{fx}}}} \right) \right) AthT_p \gamma_{outGG_{fp,x}} RadT_0 \left(1 \right. \\
& \left. - \left(\frac{AthLE_p}{AthT_p} \right)^{\frac{1-\gamma_{outGG_{fp,x}}}{\gamma_{outGG_{fp,x}}}} \right) \left(\sqrt{\left(2 \left(\gamma_{outGG_{fp,x}} + 1 \right)^{-1} \right)^{-\frac{\gamma_{outGG_{fp,x}}+1}{\gamma_{outGG_{fp,x}}-1}}} \right)^{-1} \left(\gamma_{outGG_{fp,x}} - 1 \right)^{-1} p_{LT} \\
& - \frac{aCplPM_0 \dot{m}_{VCM}^2}{RhoPM_p} - 2\frac{aCplPM_0 \dot{m}_{VCM}\dot{m}_{VGM}}{RhoPM_p} - \frac{aCplPM_0 \dot{m}_{VGM}^2}{RhoPM_p} - bCplPM_0\omega\, \dot{m}_{VCM} - bCplPM_0\omega\, \dot{m}_{VGM} \\
& - cCplPM_0 RhoPM_p\omega^2 - \frac{aCplPO_0 \dot{m}_{VCO}^2}{RhoPO_0} - 2\frac{aCplPO_0 \dot{m}_{VCO}\dot{m}_{VGO}}{RhoPO_0} - \frac{aCplPO_0 \dot{m}_{VGO}^2}{RhoPO_0} - bCplPO_0\omega\, \dot{m}_{VCO} \\
& \left. - bCplPO_0\omega\, \dot{m}_{VGO} - cCplPO_0 RhoPO_0\omega^2 \right] {JinTP_0}^{-1}
\end{aligned}
\tag{8.2.1}
$$

$$\dot{p}_{CC} = \left(\gamma_{outCC_{fp,x}} - 1\right)(\dot{m}_{VCM} \tag{8.2.2}$$
$$+ \dot{m}_{VCO})\, {V_{CC_0}}^{-1} \left(\frac{\left(T_{PIM_p}\dot{m}_{VCM} + T_{PIO_p}\dot{m}_{VCO}\right)\left(R_{fu_0}\dot{m}_{VCM} + R_{ox_0}\dot{m}_{VCO}\right)\left(\gamma_{fu_0}\dot{m}_{VCM} + \gamma_{ox_0}\dot{m}_{VCO}\right)}{\left(\gamma_{fu_0}\dot{m}_{VCM} + \gamma_{ox_0}\dot{m}_{VCO} - \dot{m}_{VCM} - \dot{m}_{VCO}\right)(\dot{m}_{VCM} + \dot{m}_{VCO})^2} \right.$$
$$\left. + \frac{T_{CC_{fx}} R_{outCC_{fp,x}} \left(\gamma_{fu_0}\dot{m}_{VCM} + \gamma_{ox_0}\dot{m}_{VCO} - \gamma_{outCC_{fp,x}}\dot{m}_{VCM} - \gamma_{outCC_{fp,x}}\dot{m}_{VCO}\right)\left(R_{fu_0}\dot{m}_{VCM} + R_{ox_0}\dot{m}_{VCO}\right)}{\left(\gamma_{outCC_{fp,x}} - 1\right)^2 \eta_{CC_p}\left(Cp_{outCC_{fp,x}} - R_{outCC_{fp,x}}\right)(\dot{m}_{VCM} + \dot{m}_{VCO})\left(\gamma_{fu_0}\dot{m}_{VCM} + \gamma_{ox_0}\dot{m}_{VCO} - \dot{m}_{VCM} - \dot{m}_{VCO}\right)} \right)$$
$$- {\gamma_{outCC_{fp,x}}}^{3/2} AthCH_p \sqrt{R_{outCC_{fp,x}}} \sqrt{T_{CC_{fx}}}\, 2^{\frac{\gamma_{outCC_{fp,x}}+1}{2\gamma_{outCC_{fp,x}}-2}} \left(\gamma_{outCC_{fp,x}} + 1\right)^{-\frac{\gamma_{outCC_{fp,x}}+1}{2\gamma_{outCC_{fp,x}}-2}} {V_{CC_0}}^{-1} p_{CC}$$
$$- i_{CC}\left[\left(-\frac{\left(\gamma_{outCC_{fp,x}} - 1\right)\eta_{CC_p}\mu_{oxinCC_{fx}}h_{fox_0}}{V_{CC_0}} - \frac{\left(\gamma_{outCC_{fp,x}} - 1\right)\eta_{CC_p}\mu_{fuinCC_{fx}}h_{ffu_0}}{V_{CC_0}}\right.\right.$$
$$\left. + \frac{\left(\gamma_{outCC_{fp,x}} - 1\right)\eta_{CC_p}\left(\mu_{fuinCC_{fx}} + \mu_{oxinCC_{fx}}\right)\mu_{p1_0}h_{fp1_0}}{V_{CC_0}} + \frac{\left(\gamma_{outCC_{fp,x}} - 1\right)\eta_{CC_p}\left(\mu_{fuinCC_{fx}} + \mu_{oxinCC_{fx}}\right)\mu_{p2_0}h_{fp2_0}}{V_{CC_0}}\right.$$
$$+ \gamma_{outCC_{fp,x}} R_{outCC_{fp,x}} T_{CC_{fx}} \left(\frac{Cp_{oxCC_{fp,x}}}{Cp_{outCC_{fp,x}}} - \frac{Cp_{oxCC_{fp,x}} - R_{ox_0}}{Cp_{outCC_{fp,x}} - R_{outCC_{fp,x}}}\right) \mu_{oxinCC_{fx}} \left(\gamma_{outCC_{fp,x}} - 1\right)^{-1} {\eta_{CC_p}}^{-1} {V_{CC_0}}^{-1}$$
$$+ \gamma_{outCC_{fp,x}} R_{outCC_{fp,x}} T_{CC_{fx}} \left(\frac{Cp_{fuCC_{fp,x}}}{Cp_{outCC_{fp,x}}} - \frac{Cp_{fuCC_{fp,x}} - R_{fu_0}}{Cp_{outCC_{fp,x}} - R_{outCC_{fp,x}}}\right) \mu_{fuinCC_{fx}} \left(\gamma_{outCC_{fp,x}} - 1\right)^{-1} {\eta_{CC_p}}^{-1} {V_{CC_0}}^{-1}$$
$$- \gamma_{outCC_{fp,x}} R_{outCC_{fp,x}} T_{CC_{fx}} \left(\frac{Cp_{p1CC_{fp,x}}}{Cp_{outCC_{fp,x}}} - \frac{Cp_{p1CC_{fp,x}} - R_{p1_0}}{Cp_{outCC_{fp,x}} - R_{outCC_{fp,x}}}\right) \frac{\left(\mu_{fuinCC_{fx}} + \mu_{oxinCC_{fx}}\right)\mu_{p1_0}}{\left(\gamma_{outCC_{fp,x}} - 1\right)\eta_{CC_p} V_{CC_0}}$$
$$- \gamma_{outCC_{fp,x}} R_{outCC_{fp,x}} T_{CC_{fx}} \left(\frac{Cp_{p2CC_{fp,x}}}{Cp_{outCC_{fp,x}}} - \frac{Cp_{p2CC_{fp,x}} - R_{p2_0}}{Cp_{outCC_{fp,x}} - R_{outCC_{fp,x}}}\right) \left.\frac{\left(\mu_{fuinCC_{fx}} + \mu_{oxinCC_{fx}}\right)\mu_{p2_0}}{\left(\gamma_{outCC_{fp,x}} - 1\right)\eta_{CC_p} V_{CC_0}}\right) (\dot{m}_{VCM} + \dot{m}_{VCO})$$
$$\left. - \frac{\left(\gamma_{outCC_{fp,x}} - 1\right)\left(Cp_{outCC_{fp,x}} T_{vap,ox_0} - Lv_{ox_0} - Cp_{oxCC_{fp,x}}\left(T_{vap,ox_0} - T_{PIO_p}\right)\right)\dot{m}_{VCO}}{V_{CC_0}}\right]$$

$$\dot{p}_{GG} = \left(\gamma_{outGG_{fp,x}} - 1\right)(\dot{m}_{VGM} + \dot{m}_{VGO} \tag{8.2.3}$$
$$+ \dot{m}_{stat})\, {V_{GG_0}}^{-1} \left(\frac{\left(T_{PIM_p}\dot{m}_{VGM} + T_{PIO_p}\dot{m}_{VGO}\right)\left(R_{fu_0}\dot{m}_{VGM} + R_{ox_0}\dot{m}_{VGO}\right)\left(\gamma_{fu_0}\dot{m}_{VGM} + \gamma_{ox_0}\dot{m}_{VGO}\right)}{\left(\gamma_{fu_0}\dot{m}_{VGM} + \gamma_{ox_0}\dot{m}_{VGO} - \dot{m}_{VGM} - \dot{m}_{VGO}\right)(\dot{m}_{VGM} + \dot{m}_{VGO})^2} \right.$$
$$\left. + \frac{T_{GG_{fx}} R_{cGG_{fp,x,t}} \left(\gamma_{fu_0}\dot{m}_{VGM} + \gamma_{ox_0}\dot{m}_{VGO} - \gamma_{outGG_{fp,x}}\dot{m}_{VGM} - \gamma_{outGG_{fp,x}}\dot{m}_{VGO}\right)\left(R_{fu_0}\dot{m}_{VGM} + R_{ox_0}\dot{m}_{VGO}\right)}{\left(\gamma_{outGG_{fp,x}} - 1\right)^2 \eta_{GG_p}\left(Cp_{outGG_{fp,x}} - R_{outGG_{fp,x}}\right)(\dot{m}_{VGM} + \dot{m}_{VGO})\left(\gamma_{fu_0}\dot{m}_{VGM} + \gamma_{ox_0}\dot{m}_{VGO} - \dot{m}_{VGM} - \dot{m}_{VGO}\right)} \right)$$
$$- \frac{\gamma_{outGG_{fp,x}} R_{cGG_{fp,x,t}} T_{GG_{fx}}}{V_{GG_0}} \dot{m}_{LT}$$
$$- i_{GG}\left[\left(-\frac{\left(\gamma_{outGG_{fp,x}} - 1\right)\eta_{GG_p}\mu_{oxinGG_{fx}}h_{fox_0}}{V_{GG_0}} - \frac{\left(\gamma_{outGG_{fp,x}} - 1\right)\eta_{GG_p}\mu_{fuinGG_{fx}}h_{ffu_0}}{V_{GG_0}}\right.\right.$$
$$+ \frac{\left(\gamma_{outGG_{fp,x}} - 1\right)\eta_{GG_p}\left(\mu_{fuinGG_{fx}} + \mu_{oxinGG_{fx}}\right)\mu_{p1_0}h_{fp1_0}}{V_{GG_0}}$$
$$\left. + \frac{\left(\gamma_{outGG_{fp,x}} - 1\right)\eta_{GG_p}\left(\mu_{fuinGG_{fx}} + \mu_{oxinGG_{fx}}\right)\mu_{p2_0}h_{fp2_0}}{V_{GG_0}}\right.$$
$$+ \gamma_{outGG_{fp,x}} R_{cGG_{fp,x,t}} T_{GG_{fx}} \left(\frac{Cp_{ox_{fp,x}}}{Cp_{outGG_{fp,x}}} - \frac{Cp_{ox_{fp,x}} - R_{ox_0}}{Cp_{outGG_{fp,x}} - R_{outGG_{fp,x}}}\right) \mu_{oxinGG_{fx}} \left(\gamma_{outGG_{fp,x}} - 1\right)^{-1} {\eta_{GG_p}}^{-1} {V_{GG_0}}^{-1}$$
$$+ \gamma_{outGG_{fp,x}} R_{cGG_{fp,x,t}} T_{GG_{fx}} \left(\frac{Cp_{fu_{fp,x}}}{Cp_{outGG_{fp,x}}} - \frac{Cp_{fu_{fp,x}} - R_{fu_0}}{Cp_{outGG_{fp,x}} - R_{outGG_{fp,x}}}\right) \mu_{fuinGG_{fx}} \left(\gamma_{outGG_{fp,x}} - 1\right)^{-1} {\eta_{GG_p}}^{-1} {V_{GG_0}}^{-1}$$
$$- \gamma_{outGG_{fp,x}} R_{cGG_{fp,x,t}} T_{GG_{fx}} \left(\frac{Cp_{p1_{fp,x}}}{Cp_{outGG_{fp,x}}} - \frac{Cp_{p1_{fp,x}} - R_{p1_0}}{Cp_{outGG_{fp,x}} - R_{outGG_{fp,x}}}\right) \frac{\left(\mu_{fuinGG_{fx}} + \mu_{oxinGG_{fx}}\right)\mu_{p1_0}}{\left(\gamma_{outGG_{fp,x}} - 1\right)\eta_{GG_p} V_{GG_0}}$$
$$- \gamma_{outGG_{fp,x}} R_{cGG_{fp,x}} T_{GG_{fx}} \left(\frac{Cp_{p2_{fp,x}}}{Cp_{outGG_{fp,x}}} - \frac{Cp_{p2_{fp,x}} - R_{p2_0}}{Cp_{outGG_{fp,x}} - R_{outGG_{fp,x}}}\right) \left.\frac{\left(\mu_{fuinGG_{fx}} + \mu_{oxinGG_{fx}}\right)\mu_{p2_0}}{\left(\gamma_{outGG_{fp,x}} - 1\right)\eta_{GG_p} V_{GG_0}}\right) (\dot{m}_{VGM}$$
$$\left. + \dot{m}_{VGO}) - \frac{\left(\gamma_{outGG_{fp,x}} - 1\right)\left(Cp_{outGG_{fp,x}} T_{vap,ox_0} - Lv_{ox_0} - Cp_{ox_{fp,x}}\left(T_{vap,ox_0} - T_{PIO_p}\right)\right)\dot{m}_{VGO}}{V_{GG_0}}\right]$$

$$\dot{p}_{LT} = \frac{\gamma_{outGG_{fp,x}} R_{outGG_{fp,x}} T_{GG_{fx}}}{V_{cav_p}} \dot{m}_{LT} - {\gamma_{outGG_{fp,x}}}^{3/2} AthT_p \sqrt{R_{outGG_{fp,x}}} \sqrt{T_{GG_{fx}}} 2^{\frac{\gamma_{outGG_{fp,x}}+1}{2\gamma_{outGG_{fp,x}}-2}} \left(\gamma_{outGG_{fp,x}} + 1\right)^{-\frac{\gamma_{outGG_{fp,x}}+1}{2\gamma_{outGG_{fp,x}}-2}} {V_{cav_p}}^{-1} p_{LT} \tag{8.2.4}$$

$$\ddot{m}_{LT} = \frac{p_{GG} - p_{LT} - ResLT_p \dot{m}_{LT}^2}{InePipe_{LT_p}} \tag{8.2.5}$$

$$\ddot{m}_{VCM} = \left(RhoPM_p cPrsPM_0 \omega^2 + \left(bPrsPM_0 \dot{m}_{VCM} + bPrsPM_0 \dot{m}_{VGM}\right)\omega + \left(-ResMIC_p - ResLCM_p + \frac{aPrsPM_0}{RhoPM_p} - ResCR_p\right)\dot{m}_{VCM}^2 + \left(2\frac{aPrsPM_0}{RhoPM_p} - 2\,ResLCM_p\right)\dot{m}_{VGM}\dot{m}_{VCM} + \left(-ResLCM_p + \frac{aPrsPM_0}{RhoPM_p}\right)\dot{m}_{VGM}^2 + p_{PIM_p} - p_{CC} - \frac{\dot{m}_{VCM}^2}{2RhoVCM_0 {A_{VCM}}^2}\right)\left(\frac{LngVCM_0}{A_{VCM}} + IneCR_p\right)^{-1} \tag{8.2.6}$$

$$\ddot{m}_{VCO} = \left(\frac{RhoPO_0 cPrsPO_0 \omega^2}{LngVCO_0} + \left(\frac{bPrsPO_0 \dot{m}_{VCO}}{LngVCO_0} + \frac{bPrsPO_0 \dot{m}_{VGO}}{LngVCO_0}\right)\omega + \left(-ResOIC_p - ResLCO_p + \frac{aPrsPO_0}{RhoPO_0}\right) {LngVCO_0}^{-1} \dot{m}_{VCO}^2 + \left(2\frac{aPrsPO_0}{RhoPO_0} - 2\,ResLCO_p\right) {LngVCO_0}^{-1} \dot{m}_{VGO}\dot{m}_{VCO} + \left(-ResLCO_p + \frac{aPrsPO_0}{RhoPO_0}\right) {LngVCO_0}^{-1} \dot{m}_{VGO}^2 + \frac{p_{PIO_p} - p_{CC}}{LngVCO_0}\right) A_{VCO} - \frac{\dot{m}_{VCO}^2}{2RhoVCO_0 LngVCO_0 A_{VCO}} \tag{8.2.7}$$

$$\ddot{m}_{VGM} = \left(\frac{RhoPM_p cPrsPM_0 \omega^2}{LngVGM_0} + \left(\frac{bPrsPM_0 \dot{m}_{VCM}}{LngVGM_0} + \frac{bPrsPM_0 \dot{m}_{VGM}}{LngVGM_0}\right)\omega + \left(-ResLGM_p + \frac{aPrsPM_0}{RhoPM_p} - ResLCM_p\right) {LngVGM_0}^{-1} \dot{m}_{VCM}^2 + \left(-2\,ResLCM_p - 2\,ResLGM_p + 2\frac{aPrsPM_0}{RhoPM_p}\right) {LngVGM_0}^{-1} \dot{m}_{VGM}\dot{m}_{VCM} + \left(-ResLCM_p + \frac{aPrsPM_0}{RhoPM_p} - ResLGM_p - ResMIG_p\right) {LngVGM_0}^{-1} \dot{m}_{VGM}^2 + \frac{p_{PIM_p} - p_{GG}}{LngVGM_0}\right) A_{VGM} - \frac{\dot{m}_{VGM}^2}{2RhoVGM_0 LngVGM_0 A_{VGM}} \tag{8.2.8}$$

$$\ddot{m}_{VGO} = \left(\frac{RhoPO_0 cPrsPO_0 \omega^2}{LngVGO_0} + \left(\frac{bPrsPO_0 \dot{m}_{VCO}}{LngVGO_0} + \frac{bPrsPO_0 \dot{m}_{VGO}}{LngVGO_0}\right)\omega + \left(-ResLGO_p + \frac{aPrsPO_0}{RhoPO_0} - ResLCO_p\right) {LngVGO_0}^{-1} \dot{m}_{VCO}^2 + \left(-2\,ResLCO_p - 2\,ResLGO_p + 2\frac{aPrsPO_0}{RhoPO_0}\right) {LngVGO_0}^{-1} \dot{m}_{VGO}\dot{m}_{VCO} + \left(-ResLCO_p + \frac{aPrsPO_0}{RhoPO_0} - ResLGO_p - ResOIG_p\right) {LngVGO_0}^{-1} \dot{m}_{VGO}^2 + \frac{p_{PIO_p} - p_{GG}}{LngVGO_0}\right) A_{VGO} - \frac{\dot{m}_{VGO}^2}{2RhoVGO_0 LngVGO_0 A_{VGO}} \tag{8.2.9}$$

Appendix B: Simplified NL state-space model terms, including *PROMETHEUS*

Constant terms in *Vulcain 1* simplified nonlinear state-space model

$$
\begin{aligned}
kp_{GG1} = &-\frac{-\gamma_{outGG_{f0}} R_{inGG_{fp,t}} \left(-Cp_{ox_{f0}} R_{outGG_{fp,t}} + Cp_{outGG_{fp,t}} R_{ox_0}\right) \mu_{oxinGG_{fp}}}{\eta_{GG_p} \left(\gamma_{outGG_{f0}} - 1\right) Cp_{outGG_{fp,t}} \left(Cp_{outGG_{fp,t}} - R_{outGG_{fp,t}}\right) V_{GG_0}} \\
&+ \frac{\gamma_{outGG_{f0}} R_{inGG_{fp,t}} \left(Cp_{fu_{f0}} R_{outGG_{fp,t}} - Cp_{outGG_{fp,t}} R_{fu_0}\right) \mu_{fuinGG_{fp}}}{\eta_{GG_p} \left(\gamma_{outGG_{f0}} - 1\right) Cp_{outGG_{fp,t}} \left(Cp_{outGG_{fp,t}} - R_{outGG_{fp,t}}\right) V_{GG_0}} \\
&+ \frac{\gamma_{outGG_{f0}} R_{inGG_{fp,t}} \left(-Cp_{p1_{f0}} R_{outGG_{fp,t}} + Cp_{outGG_{fp,t}} R_{p1_0}\right) \left(\mu_{fuinGG_{fp}} + \mu_{oxinGG_{fp}}\right) \mu_{p1_0}}{\eta_{GG_p} \left(\gamma_{outGG_{f0}} - 1\right) Cp_{outGG_{fp,t}} \left(Cp_{outGG_{fp,t}} - R_{outGG_{fp,t}}\right) V_{GG_0}}
\end{aligned}
\tag{8.2.10}
$$

$$
kp_{GG2} = \frac{R_{inGG_{fp,t}} \left(\gamma_{inGG_{fp,t}} - \gamma_{outGG_{f0}}\right) \left(Cp_{inGG_{fp,t}} - R_{inGG_{fp,t}}\right)}{\left(\gamma_{outGG_{f0}} - 1\right) \left(Cp_{outGG_{fp,t}} - R_{outGG_{fp,t}}\right) \eta_{GG_p} V_{GG_0}} \tag{8.2.11}
$$

$$
kp_{GG4} = \frac{R_{inGG_{fp,t}} \gamma_{outGG_{f0}}}{V_{GG_0}} \tag{8.2.12}
$$

$$
\begin{aligned}
kp_{GG5} = \Big(&\left(\gamma_{outGG_{f0}} - 1\right) \eta_{GG_p} \mu_{oxinGG_{fp}} h_{fox_0} + \left(\gamma_{outGG_{f0}} - 1\right) \eta_{GG_p} \mu_{fuinGG_{fp}} h_{ffu_0} \\
&- \left(\gamma_{outGG_{f0}} - 1\right) \eta_{GG_p} \left(\mu_{fuinGG_{fp}} + \mu_{oxinGG_{fp}}\right) \mu_{p1_0} h_{fp1_0}\Big) V_{GG_0}^{-1}
\end{aligned}
\tag{8.2.13}
$$

$$
kp_{GG6} = \frac{\left(\gamma_{outGG_{f0}} - 1\right) \gamma_{inGG_{fp,t}} R_{inGG_{fp,t}} T_{inGG_{fp,t}}}{\left(\gamma_{inGG_{fp,t}} - 1\right) V_{GG_0}} \tag{8.2.14}
$$

$$
kp_{LTH1} = \frac{\gamma_{outGG_{f0}} R_{outGG_{fp,t}}}{V_{cav_p}} \tag{8.2.15}
$$

$$
kp_{LTH1} = \frac{\gamma_{outGG_{f0}} AthTH_p \sqrt{R_{outGG_{fp,t}}} \left(m_\gamma \gamma_{outGG_{f0}} + n_\gamma\right)}{V_{cav_p}} \tag{8.2.16}
$$

$$
kp_{VGC1} = \frac{\gamma_{outGG_{f0}} R_{outGG_{fp,t}}}{V_{cav_p}} \tag{8.2.17}
$$

$$kpVGC2 = \frac{\gamma_{outGG_{f0}} AthTO_p \sqrt{R_{outGG_{fp,t}}} \left(m_\gamma \gamma_{outGG_{f0}} + n_\gamma\right)}{V_{cav_p}} \tag{8.2.18}$$

$$kom_{H1} = \frac{cCplPH_0 RhoPH_0}{JinTPH_0} \tag{8.2.19}$$

$$kom_{H2} = -a_{1TH_0} RadTH_0{}^2 AthTH_p \left(m_\gamma \gamma_{outGG_{f0}} + n_\gamma\right) \sqrt{\gamma_{outGG_{f0}}} \left(1 - \left(\frac{AthLEH_p}{AthTH_p}\right)^{-\frac{\gamma_{outGG_{f0}}{}^{-1}}{\gamma_{outGG_{f0}}}}\right) \left(\sqrt{\gamma_{outGG_{f0}} R_{outGG_{fp,t}}}\right)^{-1} \left(\gamma_{outGG_{f0}} - 1\right)^{-1} JinTPH_0{}^{-1} \tag{8.2.20}$$

$$kom_{H3} = \frac{bCplPH_0}{JinTPH_0} \tag{8.2.21}$$

$$kom_{H4} = a_{2\,TH_0} AthLEH_p \left(m_\gamma \gamma_{outGG_{f0}} + n_\gamma\right) \sqrt{\gamma_{outGG_{f0}}} RadTH_0 \left(1 - \left(\frac{AthLEH_p}{AthTH_p}\right)^{-\frac{\gamma_{outGG_{f0}}{}^{-1}}{\gamma_{outGG_{f0}}}}\right) \left(\gamma_{outGG_{f0}} - 1\right)^{-1} JinTPH_0{}^{-1} \tag{8.2.22}$$

$$kom_{H5} = -2\frac{aCplPH_0}{RhoPH_0 JinTPH_0} \tag{8.2.23}$$

$$kom_{O1} = \frac{cCplPO_0 RhoPO_0}{JinTPO_0} \tag{8.2.24}$$

$$kom_{O2} = -a_{1TO_0} RadTO_0{}^2 AthTO_p \left(m_\gamma \gamma_{outGG_{f0}} + n_\gamma\right) \sqrt{\gamma_{outGG_{f0}}} \left(1 - \left(\frac{AthLEO_p}{AthTO_p}\right)^{-\frac{\gamma_{outGG_{f0}}{}^{-1}}{\gamma_{outGG_{f0}}}}\right) \left(\sqrt{\gamma_{outGG_{f0}} R_{outGG_{fp,t}}}\right)^{-1} \left(\gamma_{outGG_{f0}} - 1\right)^{-1} JinTPO_0{}^{-1} \tag{8.2.25}$$

$$kom_{O3} = \frac{bCplPO_0}{JinTPO_0} \tag{8.2.26}$$

$$kom_{O4} = a_{2\,TO_0} AthLEO_p \left(m_\gamma \gamma_{outGG_{f0}} + n_\gamma\right) \sqrt{\gamma_{outGG_{f0}}} RadTO_0 \left(1 - \left(\frac{AthLEO_p}{AthTO_p}\right)^{-\frac{\gamma_{outGG_{f0}}{}^{-1}}{\gamma_{outGG_{f0}}}}\right) \left(\gamma_{outGG_{f0}} - 1\right)^{-1} JinTPO_0{}^{-1} \tag{8.2.27}$$

$$kom_{O5} = -2\frac{aCplPO_0}{RhoPO_0 JinTPO_0} \tag{8.2.28}$$

$$kp_{CC1} = -\frac{-\gamma_{outCC_{f0}} R_{inCC_{fp}} \left(-Cp_{oxCC_{f0}} R_{outCC_{fp}} + Cp_{outCC_{fp}} R_{ox_0}\right) \mu_{oxinCC_{fp}}}{\eta_{CC_p} \left(\gamma_{outCC_{f0}} - 1\right) Cp_{outCC_{fp}} \left(Cp_{outCC_{fp}} - R_{outCC_{fp}}\right) V_{CC_0}} + \frac{\gamma_{outCC_{f0}} R_{inCC_{fp}} \left(Cp_{fuCC_{f0}} R_{outCC_{fp}} - Cp_{outCC_{fp}} R_{fu_0}\right) \mu_{fuinCC_{fp}}}{\eta_{CC_p} \left(\gamma_{outCC_{f0}} - 1\right) Cp_{outCC_{fp}} \left(Cp_{outCC_{fp}} - R_{outCC_{fp}}\right) V_{CC_0}} + \frac{\gamma_{outCC_{f0}} R_{inCC_{fp}} \left(-Cp_{p1CC_{f0}} R_{outCC_{fp}} + Cp_{outCC_{fp}} R_{p1_0}\right) \left(\mu_{fuinCC_{fp}} + \mu_{oxinCC_{fp}}\right) \mu_{p1_0}}{\eta_{CC_p} \left(\gamma_{outCC_{f0}} - 1\right) Cp_{outCC_{fp}} \left(Cp_{outCC_{fp}} - R_{outCC_{fp}}\right) V_{CC_0}} \tag{8.2.29}$$

$$kp_{CC2} = \frac{R_{inCC_{fp}} \left(\gamma_{inCC_{fp}} - \gamma_{outCC_{f0}}\right) \left(Cp_{inCC_{fp}} - R_{inCC_{fp}}\right)}{\left(\gamma_{outCC_{f0}} - 1\right) \left(Cp_{outCC_{fp}} - R_{outCC_{fp}}\right) \eta_{CC_p} \ V_{CC_0}} \tag{8.2.30}$$

$$kp_{CC3} = \frac{\gamma_{outCC_{f0}} AthCH_p \left(m_\gamma \, \gamma_{outCC_{f0}} + n_\gamma\right) R_{inCC_{fp}}}{\sqrt{R_{outCC_{fp}}} V_{CC_0}} \tag{8.2.31}$$

$$kp_{CC4} = \left(\left(\gamma_{outCC_{f0}} - 1\right) \eta_{CC_p} \mu_{oxinCC_{fp}} h_{fox_0} + \left(\gamma_{outCC_{f0}} - 1\right) \eta_{CC_p} \mu_{fuinCC_{fp}} h_{ffu_0} - \left(\gamma_{outCC_{f0}} - 1\right) \eta_{CC_p} \left(\mu_{fuinCC_{fp}} + \mu_{oxinCC_{fp}}\right) \mu_{p1_0} h_{fp1_0}\right) V_{CC_0}^{-1} \tag{8.2.32}$$

$$kp_{CC5} = \frac{\left(\gamma_{outCC_{f0}} - 1\right) \gamma_{inCC_{fp}} R_{inCC_{fp}} T_{inCC_{fp}}}{\left(\gamma_{inCC_{fp}} - 1\right) V_{CC_0}} \tag{8.2.33}$$

$$km_{LTH} = \frac{KhyLTH_p R_{outGG_{fp,t}}}{InePipe_{LTH_p}} \tag{8.2.34}$$

$$km_{VCH1} = 2 \, RhoPH_0 RhoVCH_0 IneCR_p \tag{8.2.35}$$

$$km_{VCH2} = 2 \, RhoPH_0 RhoVCH_0 LngVCH_0 \tag{8.2.36}$$

$$km_{VCH3} = 2 \, RhoVCH_0 RhoPH_0{}^2 cPrsPH_0 \tag{8.2.37}$$

$$km_{VCH4} = \left(2 \, \left(ResLCH_p + ResCR_p + ResHIC_p\right) RhoPH_0 - 2 \, aPrsPH_0\right) RhoVCH_0 \tag{8.2.38}$$

$$km_{VCH5} = \left(4 \, ResLCH_p RhoPH_0 - 4 \, aPrsPH_0\right) RhoVCH_0 \tag{8.2.39}$$

$$km_{VCH6} = 2 \, RhoPH_0 RhoVCH_0 \tag{8.2.40}$$

$$km_{VCH7} = 2 \, RhoPH_0 RhoVCH_0 p_{PIH_p} \tag{8.2.41}$$

$$km_{VCH8} = RhoPH_0 \tag{8.2.42}$$

$$km_{VCO1} = \frac{RhoPO_0 cPrsPO_0}{LngVCO_0} \tag{8.2.43}$$

$$km_{VCO2} = -\frac{bPrsPO_0}{LngVCO_0} \tag{8.2.44}$$

$$km_{VCO3} = \frac{(ResOIC_p + ResLCO_p)\, RhoPO_0 - aPrsPO_0}{RhoPO_0 LngVCO_0} \tag{8.2.45}$$

$$km_{VCO4} = {LngVCO_0}^{-1} \tag{8.2.46}$$

$$km_{VCO5} = \frac{p_{PIO_p}}{LngVCO_0} \tag{8.2.47}$$

$$km_{VCO6} = 1/2\, \frac{1}{RhoVCO_0 LngVCO_0} \tag{8.2.48}$$

$$km_{VGH1} = \frac{RhoPH_0 cPrsPH_0}{LngVGH_0} \tag{8.2.49}$$

$$km_{VGH2} = \frac{-aPrsPH_0 + ResLCH_p RhoPH_0}{RhoPH_0 LngVGH_0} \tag{8.2.50}$$

$$km_{VGH3} = \frac{2\, ResLCH_p RhoPH_0 - 2\, aPrsPH_0}{RhoPH_0 LngVGH_0} \tag{8.2.51}$$

$$km_{VGH4} = \frac{(ResLCH_p + ResHIG_p)\, RhoPH_0 - aPrsPH_0}{RhoPH_0 LngVGH_0} \tag{8.2.52}$$

$$km_{VGH5} = {LngVGH_0}^{-1} \tag{8.2.53}$$

$$km_{VGH6} = \frac{p_{PIH_p}}{LngVGH_0} \tag{8.2.54}$$

$$km_{VGH7} = 1/2\, \frac{1}{RhoVGH_0 LngVGH_0} \tag{8.2.55}$$

$$km_{VGO1} = \frac{RhoPO_0 cPrsPO_0}{LngVGO_0} \tag{8.2.56}$$

$$km_{VGO2} = -\frac{bPrsPO_0}{LngVGO_0} \tag{8.2.57}$$

$$km_{VGO3} = \frac{-aPrsPO_0 + ResLCO_p RhoPO_0}{RhoPO_0 LngVGO_0} \tag{8.2.58}$$

$$km_{VGO4} = \frac{(ResLCO_p + ResOIG_p)\, RhoPO_0 - aPrsPO_0}{RhoPO_0 LngVGO_0} \tag{8.2.59}$$

$$km_{VGO5} = {LngVGO_0}^{-1} \tag{8.2.60}$$

$$km_{VGO6} = \frac{p_{PIO_p}}{LngVGO_0} \tag{8.2.61}$$

$$km_{VGO7} = 1/2\,\frac{1}{RhoVGO_0 LngVGO_0} \tag{8.2.62}$$

PROMETHEUS simplified nonlinear state-space model

$$\begin{aligned}\dot{x}_1 = \Bigg(& kom1_{fp}{x_1}^2 - \frac{kom2_{fp}x_4x_1}{\sqrt{T_{GG_{fx}}}} + kom4_{fp}x_4 + \left(-kom3_{M_{fp}}x_6 - kom3_{M_{fp}}x_8 - kom3_{O_{fp}}x_7 - kom3_{O_{fp}}x_9\right)x_1 \\ & - kom5_{M_{fp}}{x_6}^2 - kom5_{O_{fp}}{x_7}^2 - 2\,kom5_{M_{fp}}x_6x_8 \\ & - 2\,kom5_{O_{fp}}x_7x_9\Bigg)\,i_{sta}\end{aligned} \tag{8.2.63}$$

$$\begin{aligned}\dot{x}_2 = & \left(\left(kp_{CC4_{fp}} - kp_{CC1_{fp}}T_{CC_{fx}}\right)x_6 + \left(kp_{CC4_{fp}} - kp_{CC1_{fp}}T_{CC_{fx}}\right)x_7\right)i_{CC} \\ & + \left(kp_{CC5_{fp}} + kp_{CC2_{fp}}T_{CC_{fx}}\right)x_6 + \left(kp_{CC5_{fp}} + kp_{CC2_{fp}}T_{CC_{fx}}\right)x_7 - kp_{CC3_{fp}}x_2\sqrt{T_{CC_{fx}}}\end{aligned} \tag{8.2.64}$$

$$\begin{aligned}\dot{x}_3 = & \left(\left(kp_{GG5_{fp}} - kp_{GG1_{fp}}T_{GG_{fx}}\right)x_8 + \left(kp_{GG5_{fp}} - kp_{GG1_{fp}}T_{GG_{fx}}\right)x_9\right)i_{GG} + \left(kp_{GG6_{fp}} + kp_{GG2_{fp}}T_{GG_{fx}}\right)x_8 \\ & + \left(kp_{GG6_{fp}} + kp_{GG2_{fp}}T_{GG_{fx}}\right)x_9 - kp_{GG4_{fp}}T_{GG_{fx}}x_5 + \left(kp_{GG6_{fp}} + kp_{GG2_{fp}}T_{GG_{fx}}\right)w_t\end{aligned} \tag{8.2.65}$$

$$\dot{x}_4 = kp_{LT1_{fp}}T_{GG_{fx}}x_5 - kp_{LT2_{fp}}\sqrt{T_{GG_{fx}}}x_4 \tag{8.2.66}$$

$$\dot{x}_5 = \frac{x_3 - x_4}{InePipe_{LT_p}} - km_{LT_{fp}}{x_5}^2 \tag{8.2.67}$$

$$\dot{x}_6 = \frac{\left(km_{VCM3_{fp}}{x_1}^2 - km_{VCM4_{fp}}{x_6}^2 - km_{VCM5_{fp}}x_8x_6 - km_{VCM6_{fp}}x_2 + km_{VCM7_{fp}}\right){u_1}^2 - km_{VCM8_{fp}}{x_6}^2}{\left(km_{VCM1_{fp}}u_1 + km_{VCM2_{fp}}\right)u_1} \tag{8.2.68}$$

$$\dot{x}_7 = \left(km_{VCO1_{fp}}{x_1}^2 - km_{VCO2_{fp}}x_7x_1 - km_{VCO3_{fp}}{x_7}^2 - km_{VCO4_{fp}}x_2 + km_{VCO5_{fp}}\right)u_2 - \frac{km_{VCO6_{fp}}{x_7}^2}{u_2} \tag{8.2.69}$$

$$\dot{x}_8 = \left(km_{VGM1_{fp}}{x_1}^2 - km_{VGM2_{fp}}{x_6}^2 - km_{VGM3_{fp}}x_8x_6 - km_{VGM4_{fp}}{x_8}^2 - km_{VGM5_{fp}}x_3 + km_{VGM6_{fp}}\right)u_3 - \frac{km_{VGM7_{fp}}{x_8}^2}{u_3} \tag{8.2.70}$$

$$\dot{x}_9 = \left(km_{VGO1_{fp}}{x_1}^2 - km_{VGO2_{fp}}x_7x_1 - km_{VGO3_{fp}}{x_7}^2 - km_{VGO4_{fp}}{x_9}^2 - km_{VGO5_{fp}}x_3 + km_{VGO6_{fp}}\right)u_4 - \frac{km_{VGO7_{fp}}{x_9}^2}{u_4} \tag{8.2.71}$$

$$
\begin{aligned}
kp_{GG1} = &-\frac{-\gamma_{outGG_{f0}} R_{inGG_{fp,t}} \left(-Cp_{ox_{f0}} R_{outGG_{fp,t}} + Cp_{outGG_{fp,t}} R_{ox_0}\right) \mu_{oxinGG_{fp}}}{\eta_{GG_p} \left(\gamma_{outGG_{f0}} - 1\right) Cp_{outGG_{fp,t}} \left(Cp_{outGG_{fp,t}} - R_{outGG_{fp,t}}\right) V_{GG_0}} \\
&+ \frac{\gamma_{outGG_{f0}} R_{inGG_{fp,t}} \left(Cp_{fu_{f0}} R_{outGG_{fp,t}} - Cp_{outGG_{fp,t}} R_{fu_0}\right) \mu_{fuinGG_{fp}}}{\eta_{GG_p} \left(\gamma_{outGG_{f0}} - 1\right) Cp_{outGG_{fp,t}} \left(Cp_{outGG_{fp,t}} - R_{outGG_{fp,t}}\right) V_{GG_0}} \\
&+ \frac{\gamma_{outGG_{f0}} R_{inGG_{fp,t}} \left(-Cp_{p1_{f0}} R_{outGG_{fp,t}} + Cp_{outGG_{fp,t}} R_{p1_0}\right) \left(\mu_{fuinGG_{fp}} + \mu_{oxinGG_{fp}}\right) \mu_{p1_0}}{\eta_{GG_p} \left(\gamma_{outGG_{f0}} - 1\right) Cp_{outGG_{fp,t}} \left(Cp_{outGG_{fp,t}} - R_{outGG_{fp,t}}\right) V_{GG_0}} \\
&+ \frac{\gamma_{outGG_{f0}} R_{inGG_{fp,t}} \left(-Cp_{p2_{f0}} R_{outGG_{fp,t}} + Cp_{outGG_{fp,t}} R_{p2_0}\right) \left(\mu_{fuinGG_{fp}} + \mu_{oxinGG_{fp}}\right) \mu_{p2_0}}{\eta_{GG_p} \left(\gamma_{outGG_{f0}} - 1\right) Cp_{outGG_{fp,t}} \left(Cp_{outGG_{fp,t}} - R_{outGG_{fp,t}}\right) V_{GG_0}}
\end{aligned}
\tag{8.2.72}
$$

$$
kp_{GG2} = \frac{\left(Cp_{inGG_{fp,t}} - R_{inGG_{fp,t}}\right) \left(\gamma_{inGG_{fp,t}} - \gamma_{outGG_{f0}}\right) R_{inGG_{fp,t}}}{\left(\gamma_{outGG_{f0}} - 1\right) \left(Cp_{outGG_{fp,t}} - R_{outGG_{fp,t}}\right) \eta_{GG_p} V_{GG_0}} \tag{8.2.73}
$$

$$
kp_{GG4} = \frac{\gamma_{outGG_{f0}} R_{inGG_{fp,t}}}{V_{GG_0}} \tag{8.2.74}
$$

$$
\begin{aligned}
kp_{GG5} = \Big(&\left(\gamma_{outGG_{f0}} - 1\right) \eta_{GG_p} \mu_{oxinGG_{fp}} h_{fox_0} + \left(\gamma_{outGG_{f0}} - 1\right) \eta_{GG_p} \mu_{fuinGG_{fp}} h_{ffu_0} \\
&- \left(\gamma_{outGG_{f0}} - 1\right) \eta_{GG_p} \left(\mu_{fuinGG_{fp}} + \mu_{oxinGG_{fp}}\right) \mu_{p1_0} h_{fp1_0} \\
&- \left(\gamma_{outGG_{f0}} - 1\right) \eta_{GG_p} \left(\mu_{fuinGG_{fp}} + \mu_{oxinGG_{fp}}\right) \mu_{p2_0} h_{fp2_0}\Big) V_{GG_0}^{-1}
\end{aligned}
\tag{8.2.75}
$$

$$
kp_{GG6} = \frac{\left(\gamma_{outGG_{f0}} - 1\right) \gamma_{inGG_{fp,t}} R_{inGG_{fp,t}} T_{inGG_{fp,t}}}{\left(\gamma_{inGG_{fp,t}} - 1\right) V_{GG_0}} \tag{8.2.76}
$$

$$
kp_{LT1} = \frac{\gamma_{outGG_{f0}} R_{outGG_{fp,t}}}{V_{cav_p}} \tag{8.2.77}
$$

$$
kp_{LT1} = \frac{\gamma_{outGG_{f0}} AthT_p \sqrt{R_{outGG_{fp,t}}} \left(m_\gamma \gamma_{outGG_{f0}} + n_\gamma\right)}{V_{cav_p}} \tag{8.2.78}
$$

$$
kom_1 = -\frac{-cCplPM_0 RhoPM_p - cCplPO_0 RhoPO_0}{JinTP_0} \tag{8.2.79}
$$

$$
\begin{aligned}
kom_2 = -a_{1T_0} RadT_0{}^2 AthT_p \left(m_\gamma \gamma_{outGG_{f0}} + n_\gamma\right) \sqrt{\gamma_{outGG_{f0}}} \Bigg(&1 \\
&- \left(\frac{AthLE_p}{AthT_p}\right)^{-\frac{\gamma_{outGG_{f0}} - 1}{\gamma_{outGG_{f0}}}} \Bigg) \left(\sqrt{\gamma_{outGG_{f0}} R_{outGG_{fp,t}}}\right)^{-1} \left(\gamma_{outGG_{f0}} - 1\right)^{-1} JinTP_0{}^{-1}
\end{aligned}
\tag{8.2.80}
$$

$$
kom_{3M} = \frac{bCplPM_0}{JinTP_0} \tag{8.2.81}
$$

$$kom_{3O} = \frac{bCplPO_0}{JinTP_0} \tag{8.2.82}$$

$$kom_4 = a_{2\,T_0} AthLE_p \left(m_\gamma\, \gamma_{outGG_{f0}} + n_\gamma\right) \sqrt{\gamma_{outGG_{f0}}} RadT_0 \left(1 - \left(\frac{AthLE_p}{AthT_p}\right)^{-\frac{\gamma_{outGG_{f0}}{}^{-1}}{\gamma_{outGG_{f0}}}}\right) \left(\gamma_{outGG_{f0}} - 1\right)^{-1} JinTP_0{}^{-1} \tag{8.2.83}$$

$$kom_{5M} = -\frac{aCplPM_0}{RhoPM_p JinTP_0} \tag{8.2.84}$$

$$kom_{5O} = -\frac{aCplPO_0}{RhoPO_0 JinTP_0} \tag{8.2.85}$$

$$\begin{aligned} kp_{CC1} = & -\frac{-\gamma_{outCC_{f0}} R_{inCC_{fp}} \left(-Cp_{oxCC_{f0}} R_{outCC_{fp}} + Cp_{outCC_{fp}} R_{ox_0}\right) \mu_{oxinCC_{fp}}}{\eta_{CC_p} \left(\gamma_{outCC_{f0}} - 1\right) Cp_{outCC_{fp}} \left(Cp_{outCC_{fp}} - R_{outCC_{fp}}\right) V_{CC_0}} \\ & + \frac{\gamma_{outCC_{f0}} R_{inCC_{fp}} \left(Cp_{fuCC_{f0}} R_{outCC_{fp}} - Cp_{outCC_{fp}} R_{fu_0}\right) \mu_{fuinCC_{fp}}}{\eta_{CC_p} \left(\gamma_{outCC_{f0}} - 1\right) Cp_{outCC_{fp}} \left(Cp_{outCC_{fp}} - R_{outCC_{fp}}\right) V_{CC_0}} \\ & + \frac{\gamma_{outCC_{f0}} R_{inCC_{fp}} \left(-Cp_{p1CC_{f0}} R_{outCC_{fp}} + Cp_{outCC_{fp}} R_{p1_0}\right) \left(\mu_{fuinCC_{fp}} + \mu_{oxinCC_{fp}}\right) \mu_{p1_0}}{\eta_{CC_p} \left(\gamma_{outCC_{f0}} - 1\right) Cp_{outCC_{fp}} \left(Cp_{outCC_{fp}} - R_{outCC_{fp}}\right) V_{CC_0}} \\ & + \frac{\gamma_{outCC_{f0}} R_{inCC_{fp}} \left(-Cp_{p2CC_{f0}} R_{outCC_{fp}} + Cp_{outCC_{fp}} R_{p2_0}\right) \left(\mu_{fuinCC_{fp}} + \mu_{oxinCC_{fp}}\right) \mu_{p2_0}}{\eta_{CC_p} \left(\gamma_{outCC_{f0}} - 1\right) Cp_{outCC_{fp}} \left(Cp_{outCC_{fp}} - R_{outCC_{fp}}\right) V_{CC_0}} \end{aligned} \tag{8.2.86}$$

$$kp_{CC2} = \frac{\left(Cp_{inCC_{fp}} - R_{inCC_{fp}}\right) \left(\gamma_{inCC_{fp}} - \gamma_{outCC_{f0}}\right) R_{inCC_{fp}}}{\left(\gamma_{outCC_{f0}} - 1\right) \left(Cp_{outCC_{fp}} - R_{outCC_{fp}}\right) \eta_{CC_p}\ V_{CC_0}} \tag{8.2.87}$$

$$kp_{CC3} = \frac{\gamma_{outCC_{f0}} AthCH_p \left(m_\gamma\, \gamma_{outCC_{f0}} + n_\gamma\right) R_{inCC_{fp}}}{\sqrt{R_{outCC_{fp}}} V_{CC_0}} \tag{8.2.88}$$

$$\begin{aligned} kp_{CC4} = & \left(\left(\gamma_{outCC_{f0}} - 1\right) \eta_{CC_p} \mu_{oxinCC_{fp}} h_{fox_0} + \left(\gamma_{outCC_{f0}} - 1\right) \eta_{CC_p} \mu_{fuinCC_{fp}} h_{ffu_0} \right. \\ & - \left(\gamma_{outCC_{f0}} - 1\right) \eta_{CC_p} \left(\mu_{fuinCC_{fp}} + \mu_{oxinCC_{fp}}\right) \mu_{p1_0} h_{fp1_0} \\ & \left. - \left(\gamma_{outCC_{f0}} - 1\right) \eta_{CC_p} \left(\mu_{fuinCC_{fp}} + \mu_{oxinCC_{fp}}\right) \mu_{p2_0} h_{fp2_0}\right) V_{CC_0}^{-1} \end{aligned} \tag{8.2.89}$$

$$kp_{CC5} = \frac{\left(\gamma_{outCC_{f0}} - 1\right) \gamma_{inCC_{fp}} R_{inCC_{fp}} T_{inCC_{fp}}}{\left(\gamma_{inCC_{fp}} - 1\right) V_{CC_0}} \tag{8.2.90}$$

$$km_{LT} = \frac{ResLT_p}{InePipe_{LT_p}} \tag{8.2.91}$$

$$km_{VCM1} = 2\, RhoPM_p RhoVCM_0 IneCR_p \tag{8.2.92}$$

$$km_{VCM2} = 2\,RhoPM_p RhoVCM_0 LngVCM_0 \tag{8.2.93}$$

$$km_{VCM3} = 2\,RhoPM_p{}^2 cPrsPM_0 RhoVCM_0 \tag{8.2.94}$$

$$km_{VCM4} = \left(2\,\left(ResLCM_p + ResCR_p + ResMIC_p\right) RhoPM_p - 2\,aPrsPM_0\right) RhoVCM_0 \tag{8.2.95}$$

$$km_{VCM5} = \left(4\,ResLCM_p RhoPM_p - 4\,aPrsPM_0\right) RhoVCM_0 \tag{8.2.96}$$

$$km_{VCM6} = 2\,RhoPM_p RhoVCM_0 \tag{8.2.97}$$

$$km_{VCM7} = 2\,RhoPM_p RhoVCM_0 p_{PIM_p} \tag{8.2.98}$$

$$km_{VCM8} = RhoPM_p \tag{8.2.99}$$

$$km_{VCO1} = \frac{RhoPO_0 cPrsPO_0}{LngVCO_0} \tag{8.2.100}$$

$$km_{VCO2} = -\frac{bPrsPO_0}{LngVCO_0} \tag{8.2.101}$$

$$km_{VCO3} = \frac{\left(ResOIC_p + ResLCO_p\right) RhoPO_0 - aPrsPO_0}{RhoPO_0 LngVCO_0} \tag{8.2.102}$$

$$km_{VCO4} = LngVCO_0{}^{-1} \tag{8.2.103}$$

$$km_{VCO5} = \frac{p_{PIO_p}}{LngVCO_0} \tag{8.2.104}$$

$$km_{VCO6} = 1/2\,\frac{1}{RhoVCO_0 LngVCO_0} \tag{8.2.105}$$

$$km_{VGM1} = \frac{RhoPM_p cPrsPM_0}{LngVGM_0} \tag{8.2.106}$$

$$km_{VGM2} = \frac{-aPrsPM_0 + \left(ResLGM_p + ResLCM_p\right) RhoPM_p}{RhoPM_p LngVGM_0} \tag{8.2.107}$$

$$km_{VGM3} = \frac{\left(2\,ResLCM_p + 2\,ResLGM_p\right) RhoPM_p - 2\,aPrsPM_0}{RhoPM_p LngVGM_0} \tag{8.2.108}$$

$$km_{VGM4} = \frac{\left(ResLCM_p + ResLGM_p + ResMIG_p\right) RhoPM_p - aPrsPM_0}{RhoPM_p LngVGM_0} \tag{8.2.109}$$

$$km_{VGM5} = LngVGM_0{}^{-1} \tag{8.2.110}$$

$$km_{VGM6} = \frac{p_{PIM_p}}{LngVGM_0} \tag{8.2.111}$$

$$km_{VGM7} = 1/2 \frac{1}{RhoVGM_0 LngVGM_0} \tag{8.2.112}$$

$$km_{VGO1} = \frac{RhoPO_0 cPrsPO_0}{LngVGO_0} \tag{8.2.113}$$

$$km_{VGO2} = -\frac{bPrsPO_0}{LngVGO_0} \tag{8.2.114}$$

$$km_{VGO3} = \frac{-aPrsPO_0 + (ResLGO_p + ResLCO_p)\, RhoPO_0}{RhoPO_0 LngVGO_0} \tag{8.2.115}$$

$$km_{VGO4} = \frac{(ResLCO_p + ResLGO_p + ResOIG_p)\, RhoPO_0 - aPrsPO_0}{RhoPO_0 LngVGO_0} \tag{8.2.116}$$

$$km_{VGO5} = {LngVGO_0}^{-1} \tag{8.2.117}$$

$$km_{VGO6} = \frac{p_{PIO_p}}{LngVGO_0} \tag{8.2.118}$$

$$km_{VGO7} = 1/2 \frac{1}{RhoVGO_0 LngVGO_0} \tag{8.2.119}$$

Appendix C: Linearised state-space model of *Vulcain 1*

This appendix presents the symbolic structure of the linearised model of the *Vulcain 1* engine (A_c, B_c and B_w matrices defined in (5.2.31)), as well as their non-dimensional evaluation about its nominal steady state. Furthermore, the most relevant corresponding transfer functions are also included.

A_c

$$= \begin{bmatrix}
A_{1,1} & 0 & 0 & 0 & A_{1,5} & 0 & 0 & A_{1,8} & 0 & A_{1,10} & 0 & 0 \\
0 & A_{2,2} & 0 & 0 & 0 & A_{2,6} & 0 & 0 & A_{2,9} & 0 & A_{2,11} & 0 \\
0 & 0 & A_{3,3} & 0 & 0 & 0 & 0 & A_{3,8-9} & A_{3,8-9} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & A_{4,7-12} & 0 & 0 & A_{4,10-11} & A_{4,10-11} & A_{4,7-12} \\
0 & 0 & 0 & 0 & A_{5,5} & 0 & A_{5,7} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & A_{6,6} & 0 & 0 & 0 & 0 & 0 & A_{6,12} \\
0 & 0 & 0 & A_{7,4} & A_{7,5} & 0 & A_{7,7} & 0 & 0 & 0 & 0 & 0 \\
A_{8,1} & 0 & A_{8,3} & 0 & 0 & 0 & 0 & A_{8,8} & 0 & A_{8,10} & 0 & 0 \\
0 & A_{9,2} & A_{9,3} & 0 & 0 & 0 & 0 & 0 & A_{9,8} & 0 & 0 & 0 \\
A_{10,1} & 0 & 0 & A_{10,4} & 0 & 0 & 0 & A_{10,8} & 0 & A_{10,10} & 0 & 0 \\
0 & A_{11,2} & 0 & A_{11,4} & 0 & 0 & 0 & 0 & A_{11,9} & 0 & A_{11,11} & 0 \\
0 & 0 & 0 & A_{12,4} & 0 & A_{12,6} & 0 & 0 & 0 & 0 & 0 & A_{12,12}
\end{bmatrix}.$$

$A_c(\mathbf{x}_{r,nom}, \mathbf{u}_{r,nom}, \mathbf{w}_{t,nom})$

$$= \begin{bmatrix}
-14.92 & 0 & 0 & 0 & 9.57 & 0 & 0 & -1.92 & 0 & -0.35 & 0 & 0 \\
0 & -17.18 & 0 & 0 & 0 & 6.29 & 0 & 0 & -0.14 & 0 & -0.019 & 0 \\
0 & 0 & -1666.56 & 0 & 0 & 0 & 0 & 225.45 & 225.45 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -131 & 0 & 0 & 144.26 & 144.26 & -131 \\
0 & 0 & 0 & 0 & -29.62 & 0 & 19.06 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -23.56 & 0 & 0 & 0 & 0 & 0 & 19.06 \\
0 & 0 & 0 & 24932.24 & -24251.71 & 0 & -903.83 & 0 & 0 & 0 & 0 & 0 \\
6130.08 & 0 & -1301.4 & 0 & 0 & 0 & 0 & -3814.35 & 0 & -293.91 & 0 & 0 \\
0 & 68848.77 & -7510.35 & 0 & 0 & 0 & 0 & 0 & -2167.64 & 0 & 0 & 0 \\
806.75 & 0 & 0 & -171.27 & 0 & 0 & 0 & -346.04 & 0 & -281.59 & 0 & 0 \\
0 & 600.85 & 0 & -65.54 & 0 & 0 & 0 & 0 & -12.51 & 0 & -66.37 & 0 \\
0 & 0 & 0 & 163668.62 & 0 & -123844.32 & 0 & 0 & 0 & 0 & 0 & -93809.02
\end{bmatrix}.$$

$$A_{1,1} = \left(-kom_{H3_{fp}} x_{r8} - kom_{H3_{fp}} x_{r10} - \frac{kom_{H2_{fp,t}} x_{r5}}{\sqrt{T_{GG_{fx}}}} \right) i_{GG}. \tag{8.2.120}$$

$$A_{1,5} = \left(-\frac{kom_{H2_{fp,t}} x_{r1}}{\sqrt{T_{GG_{fx}}}} + kom_{H4_{fp,t}} \right) i_{GG}. \tag{8.2.121}$$

$$A_{1,8} = \left(-2\, kom_{H5_{fp}} x_{r8} - kom_{H3_{fp}} x_{r1} - 2\, kom_{H5_{fp}} x_{r10} \right) i_{GG}. \tag{8.2.122}$$

$$A_{1,10} = \left(-kom_{H3_{fp}} x_{r1} - 2\, kom_{H5_{fp}} x_{r8} \right) i_{GG}. \tag{8.2.123}$$

$$A_{2,2} = \left(-kom_{O3_{fp}} x_{r9} - kom_{O3_{fp}} x_{r11} + 2\, kom_{O1_{fp}} x_{r2} - \frac{kom_{O2_{fp,t}} x_{r6}}{\sqrt{T_{GG_{fx}}}} \right) i_{GG}. \tag{8.2.124}$$

$$A_{2,6} = \left(-\frac{kom_{O2_{fp,t}} x_{r2}}{\sqrt{T_{GG_{fx}}}} + kom_{O4_{fp,t}} \right) i_{GG}. \tag{8.2.125}$$

$$A_{2,9} = \left(-2\, kom_{O5_{fp}} x_{r9} - kom_{O3_{fp}} x_{r2} - 2\, kom_{O5_{fp}} x_{r11} \right) i_{GG}. \tag{8.2.126}$$

$$A_{2,11} = \left(-kom_{O3_{fp}} x_{r2} - 2\, kom_{O5_{fp}} x_{r9} \right) i_{GG}. \tag{8.2.127}$$

$$A_{3,3} = -kp_{CC3_{fp}} \sqrt{T_{CC_{fx}}}. \tag{8.2.128}$$

$$A_{3,8-9} = \left(kp_{CC4_{fp}} - kp_{CC1_{fp}} T_{CC_{fx}} \right) i_{CC} + kp_{CC5_{fp}} + kp_{CC2_{fp}} T_{CC_{fx}}. \tag{8.2.129}$$

$$A_{4,7-12} = -kp_{GG4_{fp,t}} T_{GG_{fx}}. \tag{8.2.130}$$

$$A_{4,10-11} = \left(kp_{GG5_{fp}} - kp_{GG1_{fp,t}} T_{GG_{fx}} \right) i_{GG} + kp_{GG6_{fp,t}} + kp_{GG2_{fp,t}} T_{GG_{fx}}. \tag{8.2.131}$$

$$A_{5,5} = -kp_{LTH2_{fp,t}} \sqrt{T_{GG_{fx}}}. \tag{8.2.132}$$

$$A_{5,7} = kp_{LTH1_{fp,t}} T_{GG_{fx}}. \tag{8.2.133}$$

$$A_{6,6} = -kp_{VGC2_{fp,t}} \sqrt{T_{GG_{fx}}}. \tag{8.2.134}$$

$$A_{6,12} = kp_{VGC1_{fp,t}} T_{GG_{fx}}. \tag{8.2.135}$$

$$A_{7,4} = InePipe_{LTH_p}{}^{-1} + \frac{km_{LTH_{fp,t}} T_{GG_{fx}} x_{r7}{}^2}{x_{r4}{}^2}. \tag{8.2.136}$$

$$A_{7,5} = -InePipe_{LTH_p}{}^{-1}. \tag{8.2.137}$$

$$A_{7,7} = -2\, \frac{km_{LTH_{fp,t}} T_{GG_{fx}} x_{r7}}{x_{r4}}. \tag{8.2.138}$$

$$A_{8,1} = 2\, \frac{u_{r1} km_{VCH3_{fp}} x_{r1}}{km_{VCH1_{fp}} u_{r1} + km_{VCH2_{fp}}}. \tag{8.2.139}$$

$$A_{8,3} = -\frac{u_{r1} km_{VCH6_{fp}}}{km_{VCH1_{fp}} u_{r1} + km_{VCH2_{fp}}}. \tag{8.2.140}$$

$$A_{8,8} = \frac{\left(-2\, km_{VCH4_{fp}} x_{r8} - km_{VCH5_{fp}} x_{r10} \right) u_{r1}{}^2 - 2\, km_{VCH8_{fp}} x_{r8}}{\left(km_{VCH1_{fp}} u_{r1} + km_{VCH2_{fp}} \right) u_{r1}}. \tag{8.2.141}$$

$$A_{8,10} = -\frac{u_{r1} km_{VCH5_{fp}} x_{r8}}{km_{VCH1_{fp}} u_{r1} + km_{VCH2_{fp}}}. \tag{8.2.142}$$

$$A_{9,2} = \left(-km_{VCO2_{fp}} x_{r9} + 2\, km_{VCO1_{fp}} x_{r2}\right) u_{r2}. \tag{8.2.143}$$

$$A_{9,3} = -km_{VCO4_{fp}} u_{r2}. \tag{8.2.144}$$

$$A_{9,8} = \left(-km_{VCO2_{fp}} x_{r2} - 2\, km_{VCO3_{fp}} x_{r9}\right) u_{r2} - 2\frac{km_{VCO6_{fp}} x_{r9}}{u_{r2}}. \tag{8.2.145}$$

$$A_{10,1} = 2km_{VGH1_{fp}} x_{r1} u_{r3}. \tag{8.2.146}$$

$$A_{10,4} = -km_{VGH5_{fp}} u_{r3}. \tag{8.2.147}$$

$$A_{10,8} = \left(-2\, km_{VGH2_{fp}} x_{r8} - km_{VGH3_{fp}} x_{r10}\right) u_{r3}. \tag{8.2.148}$$

$$A_{10,10} = \left(-km_{VGH3_{fp}} x_{r8} - 2\, km_{VGH4_{fp}} x_{r10}\right) u_{r3} - 2\,\frac{km_{VGH7_{fp}} x_{r10}}{u_{r3}}. \tag{8.2.149}$$

$$A_{11,2} = \left(2\, km_{VGO1_{fp}} x_{r2} - km_{VGO2_{fp}} x_{r9}\right) u_{r4}. \tag{8.2.150}$$

$$A_{11,4} = -km_{VGO5_{fp}} u_{r4}. \tag{8.2.151}$$

$$A_{11,9} = \left(-km_{VGO2_{fp}} x_{r2} - 2\, km_{VGO3_{fp}} x_{r9}\right) u_{r4}. \tag{8.2.152}$$

$$A_{11,11} = -2\, km_{VGO4_{fp}} x_{r11} u_{r4} - 2\,\frac{km_{VGO7_{fp}} x_{r11}}{u_{r4}}. \tag{8.2.153}$$

$$A_{12,4} = \left(1 + 1/2\,\frac{\zeta VGC_{f0} R_{outGG_{fp,t}} T_{GG_{fx}} {x_{r12}}^2}{{x_{r4}}^2 {u_{r5}}^2}\right) u_{r5}. \tag{8.2.154}$$

$$A_{12,6} = -u_{r5}. \tag{8.2.155}$$

$$A_{12,12} = -\frac{\zeta VGC_{f0} R_{outGG_{fp,t}} T_{GG_{fx}} x_{r12}}{x_{r4} u_{r5}}. \tag{8.2.156}$$

$$B_c = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ B_{8,1} & 0 & 0 & 0 & 0 \\ 0 & B_{9,2} & 0 & 0 & 0 \\ 0 & 0 & B_{10,3} & 0 & 0 \\ 0 & 0 & 0 & B_{11,4} & 0 \\ 0 & 0 & 0 & 0 & B_{12,5} \end{bmatrix}. \tag{8.2.157}$$

$$B_c(\mathbf{x}_{r,nom}, \mathbf{u}_{r,nom}, \mathbf{w}_{t,nom}) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1.39 & 0 & 0 & 0 & 0 \\ 0 & 2092.08 & 0 & 0 & 0 \\ 0 & 0 & 226.54 & 0 & 0 \\ 0 & 0 & 0 & 130.75 & 0 \\ 0 & 0 & 0 & 0 & 562206.30 \end{bmatrix}. \quad (8.2.158)$$

$$B_{8,1} = \frac{2\,km_{VCH3_{fp}} {x_{r1}}^2 - 2\,km_{VCH6_{fp}} x_{r3} - 2\,km_{VCH5_{fp}} x_{r10} x_{r8} - 2\,km_{VCH4_{fp}} {x_{r8}}^2 + 2\,km_{VCH7_{fp}}}{km_{VCH1_{fp}} u_{r1} + km_{VCH2_{fp}}} + \left(-\frac{km_{VCH1_{fp}}}{\left(km_{VCH1_{fp}} u_{r1} + km_{VCH2_{fp}}\right)^2} - \frac{1}{\left(km_{VCH1_{fp}} u_{r1} + km_{VCH2_{fp}}\right) u_{r1}} \right) \left(\left(km_{VCH3_{fp}} {x_{r1}}^2 - km_{VCH6_{fp}} x_{r3} - km_{VCH5_{fp}} x_{r10} x_{r8} - km_{VCH4_{fp}} {x_{r8}}^2 + km_{VCH7_{fp}} \right) {u_{r1}}^2 - km_{VCH8_{fp}} {x_{r8}}^2 \right) {u_{r1}}^{-1}. \quad (8.2.159)$$

$$B_{9,2} = km_{VCO1_{fp}} {x_{r2}}^2 - km_{VCO4_{fp}} x_{r3} - km_{VCO3_{fp}} {x_{r9}}^2 - km_{VCO2_{fp}} x_{r9} x_{r2} + km_{VCO5_{fp}} + \frac{km_{VCO6_{fp}} {x_{r9}}^2}{{u_{r2}}^2}. \quad (8.2.160)$$

$$B_{10,3} = km_{VGH1_{fp}} {x_{r1}}^2 + km_{VGH6_{fp}} - km_{VGH3_{fp}} x_{r10} x_{r8} - km_{VGH2_{fp}} {x_{r8}}^2 - km_{VGH5_{fp}} x_{r4} - km_{VGH4_{fp}} {x_{r10}}^2 + \frac{km_{VGH7_{fp}} {x_{r10}}^2}{{u_{r3}}^2}. \quad (8.2.161)$$

$$B_{11,4} = km_{VGO1_{fp}} {x_{r2}}^2 + km_{VGO6_{fp}} - km_{VGO3_{fp}} {x_{r9}}^2 - km_{VGO2_{fp}} x_{r9} x_{r2} - km_{VGO5_{fp}} x_{r4} - km_{VGO4_{fp}} {x_{r11}}^2 + \frac{km_{VGO7_{fp}} {x_{r11}}^2}{{u_{r4}}^2}. \quad (8.2.162)$$

$$B_{12,5} = \frac{\zeta VGC_{f0} R_{outGG_{fp,t}} T_{GG_{fx}} {x_{r12}}^2}{2 x_{r4} {u_{r5}}^2} + x_{r4} - x_{r6}. \quad (8.2.163)$$

$$B_w = \begin{bmatrix} 0 \\ 0 \\ 0 \\ kp_{GG6_{fp,t}} + kp_{GG2_{fp,t}} T_{GGfx} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \rightarrow B_w(\mathbf{x}_{r,nom}, \mathbf{u}_{r,nom}, \mathbf{w}_{t,nom}) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 36.25 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \quad (8.2.164)$$

The transfer function between p_{CC} (x_3) and A_{VGF} (u_3), one of the most relevant for system control, is (evaluated at nominal steady state):

$$\frac{-1.5 \times 10^{7} s^{9} - 1.5 \times 10^{12} s^{8} - 5 \times 10^{15} s^{7} - 1 \times 10^{19} s^{6} - 1. \times 10^{22} s^{5} - 2.6 \times 10^{24} s^{4} - 1.3 \times 10^{26} s^{3} + 2 \times 10^{28} s^{2} + 1.7 \times 10^{30} s + 2.4 \times 10^{31}}{s^{12} + 1 \times 10^{5} s^{11} + 9 \times 10^{8} s^{10} + 3.5 \times 10^{12} s^{9} + 8.5 \times 10^{15} s^{8} + 1.3 \times 10^{19} s^{7} + 1.3 \times 10^{22} s^{6} + 4.3 \times 10^{24} s^{5} + 6.2 \times 10^{26} s^{4} + 4.3 \times 10^{28} s^{3} + 1.4 \times 10^{30} s^{2} + 2 \times 10^{31} s + 7.7 \times 10^{31}}.$$

The other most relevant one relates p_{CC} (x_3) to A_{VGO} (u_4):

$$\frac{-3.8 \times 10^{7} s^{8} - 3.5 \times 10^{12} s^{7} + 1.5 \times 10^{15} s^{6} + 3.4 \times 10^{19} s^{5} + 3.2 \times 10^{22} s^{4} + 6.3 \times 10^{25} s^{3} + 2.5 \times 10^{28} s^{2} + 3.5 \times 10^{30} s + 5.6 \times 10^{31}}{s^{12} + 1 \times 10^{5} s^{11} + 9 \times 10^{8} s^{10} + 3.5 \times 10^{12} s^{9} + 8.5 \times 10^{15} s^{8} + 1.3 \times 10^{19} s^{7} + 1.3 \times 10^{22} s^{6} + 4.3 \times 10^{24} s^{5} + 6.2 \times 10^{26} s^{4} + 4.3 \times 10^{28} s^{3} + 1.4 \times 10^{30} s^{2} + 2 \times 10^{31} s + 7.7 \times 10^{31}}.$$

Appendix D: Frequency-response plots (BODE diagrams) of *Vulcain 1* at steady state

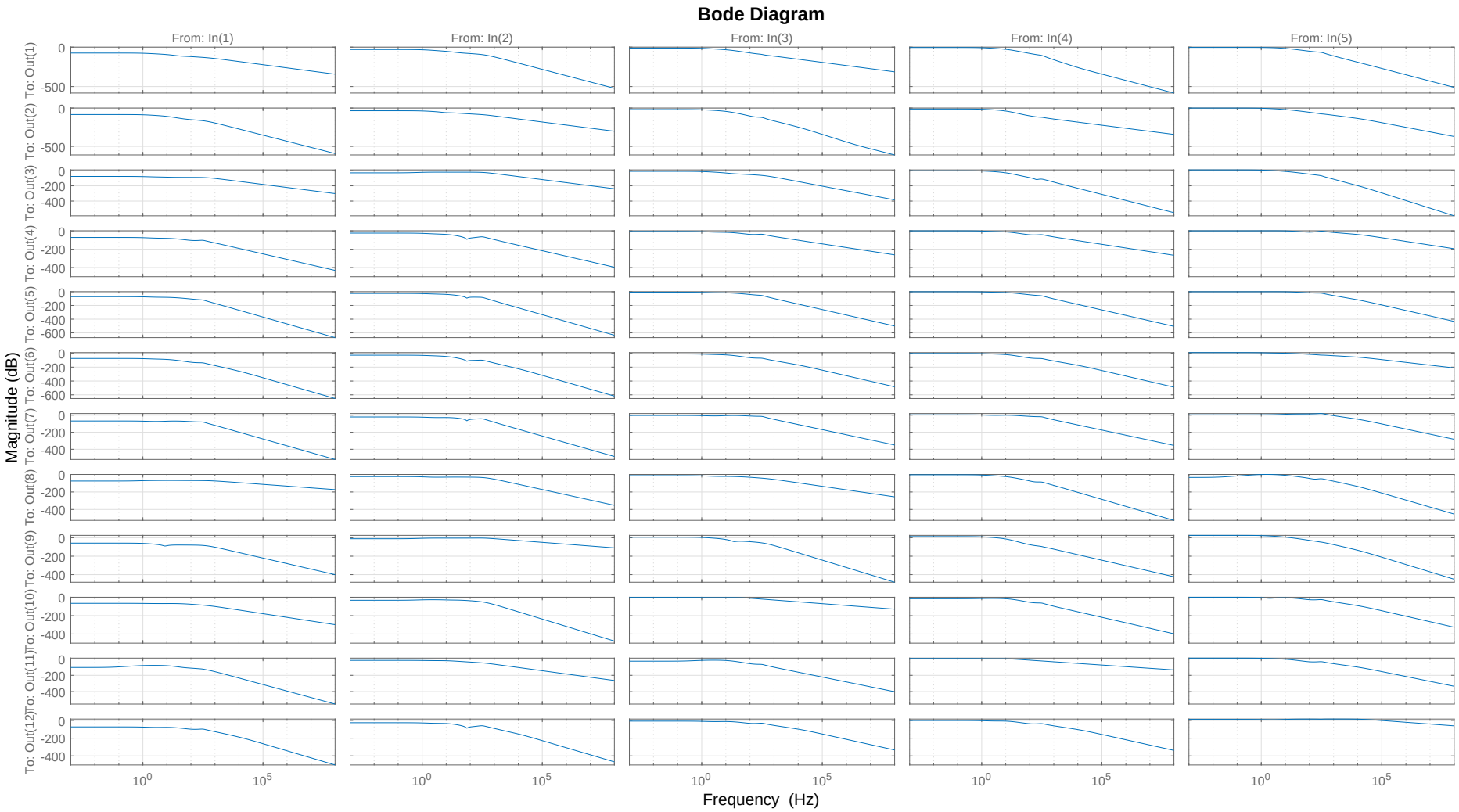


Figure 8.2.1: *Vulcain 1* multivariable BODE-plot magnitude at steady state

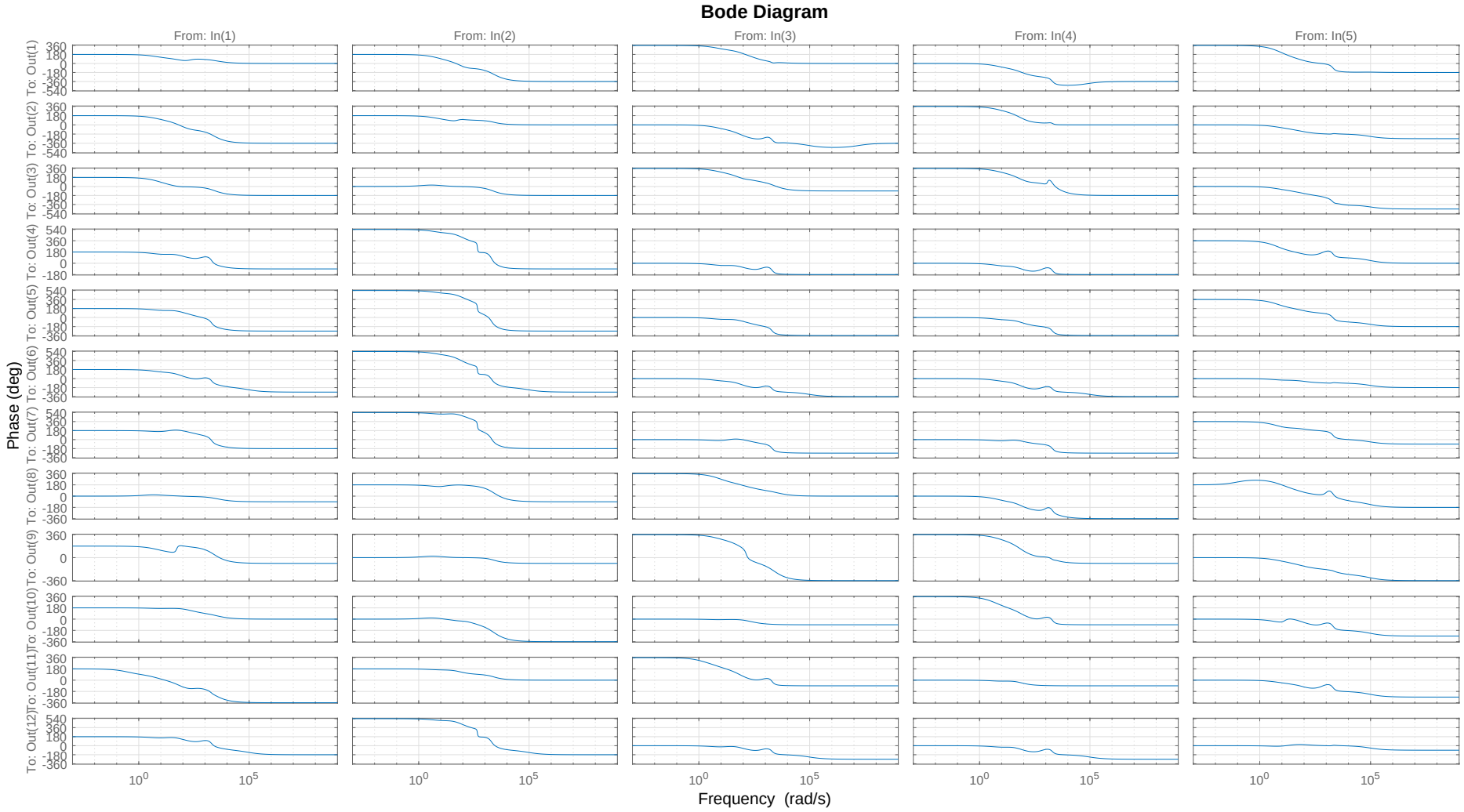


Figure 8.2.2: *Vulcain 1* multivariable BODE-plot phase at steady state

Bibliography

[1] ABDALLAH C., DAWSON D.M., DORATO P., AND JAMSHIDI M., *Survey of robust control for rigid robots*, IEEE Control Systems Magazine, 11 (1991), pp. 24–30.

[2] ABRAHAM E. AND KERRIGAN E.C., *Optimal Active Control and Optimization of a Wave Energy Converter*, IEEE Transactions on Sustainable Energy, 4 (2013), pp. 324–332.

[3] ALEXANDRE J. DIT SANDRETTO, *Reliable NonLinear Model-Predictive Control via Validated Simulation*, in 2018 Annual American Control Conference (ACC), Milwaukee, USA, June 2018, IEEE, pp. 609–614.

[4] ANTSAKLIS P.J., STIVER J.A., AND LEMMON M., *Hybrid system modeling and autonomous control systems*, in Hybrid systems, Springer, 1993, pp. 366–392.

[5] BAIOCCO P. AND BONNAL C., *Technology demonstration for reusable launchers*, Acta Astronautica, 120 (2016), pp. 43–58.

[6] BARS R., COLANERI P., DE SOUZA C.E., DUGARD L., ALLGOEWER F., KLEIMENOV A., AND SCHERER C., *Theory, algorithms and technology in the design of control systems*, Annual Reviews in Control, 30 (2006), pp. 19–30.

[7] BAUMRUCKER B.T., RENFRO J.G., AND BIEGLER L.T., *MPEC problem formulations and solution strategies with chemical engineering applications*, Computers & Chemical Engineering, 32 (2008), pp. 2903–2913.

[8] BELLOWS J.T., BREWSTER R.E., AND BEKIR E., *OTV liquid rocket engine control and health monitoring*, in 20th AIAA/SAE/ASME Joint Propulsion Conference, vol. AIAA-84-1286, Cincinnati, USA, 1984, AIAA. - Rockwell International/ Rocketdyne Division.

[9] BELYAEV E.N., CHVANOV V.K., AND CHERVAKOV V.V., *Matematicheskoe modelirovanie rabochego protsessa zhidkostnykh raketnykh dvigatelei (Mathematical Simulation of the Working Process of Liquid-Propellant Rocket Engines)*, tech. rep., Moscow Aviation Institute (MAI), 1999.

[10] BEMPORAD A. AND MORARI M., *Control of systems integrating logic, dynamics, and constraints*, Automatica, 35 (1999), pp. 407–427.

[11] ——, *Robust model predictive control: A survey*, in Robustness in identification and control, Garulli A. and Tesi A., eds., Lecture Notes in Control and Information Sciences, Springer London, 1999, pp. 207–226.

[12] BERA D., VAN HEE K., AND NIJMEIJER H., *Relationship between simulink and petri nets*, in International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH), IEEE, 2014, pp. 12–23.

[13] BERGMANS J.L. AND MYERS R.I., *Throttle valves for air turbo-rocket engine control*, in AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit, 33rd, vol. AIAA 97-3188, Seattle, USA, 1997, AIAA. - CFD Research Corporation.

[14] BERNARDINI D. AND BEMPORAD A., *Scenario-based model predictive control of stochastic constrained linear systems*, in Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference, Dec. 2009, pp. 6333–6338.

[15] BETTS J.T., *Survey of Numerical Methods for Trajectory Optimization*, Journal of Guidance, Control, and Dynamics, 21 (1998), pp. 193–207.

[16] BHATTACHARYYA S.P., CHAPELLAT H., AND KEEL L.H., *Robust Control: The Parametric Approach*, Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st ed., 1995.

[17] BLUE ORIGIN, *Blue Origin, Our Approach to Technology*, 2016.

[18] BONHOMME C., IANNETTI A., GIRARD N., TCHOU-KIEN D., RAVIER N., EDELINE E., AND DANOUS P., *Prometheus: European next generation liquid rocket engine*, in 68th International Astronautical Congress: Unlocking Imagination, Fostering Innovation and Strengthening Security, IAC 2017, vol. 13, Adelaide, Australia, 2017, IAF, pp. 8488–8499.

[19] BONNE F., ALAMIR M., AND BONNAY P., *Experimental investigation of control updating period monitoring in industrial PLC-based fast MPC: Application to the constrained control of a cryogenic refrigerator*, Control Theory and Technology, 15 (2017), pp. 92–108.

[20] BORONINE E. AND FREY M., *SMART code - A Versatile System Simulation Tool for Rocket Applications*, in Space Propulsion Conference SPC2014-2970773, Cologne, Germany, 2014.

[21] BRIAT C., *Robust Control and Observation of LPV Time-Delay Systems*, ph.D. Thesis, Institut National Polytechnique de Grenoble - INPG, Nov. 2008.

[22] BRUZELIUS F., *Linear parameter-varying systems: an approach to gain scheduling*, no. N.S., 2076 in Doktorsavhandlingar vid Chalmers Tekniska Högskola, Chalmers Univ. of Technology, Göteborg, 2004. OCLC: 249664937.

[23] BURESH J.F. AND SCHUDER C.B., *Development of a universal gas sizing equation for control valves*, 1964.

[24] BYRD R.H., NOCEDAL J., AND WALTZ R.A., *Knitro: An Integrated Package for Nonlinear Optimization*, in Large-Scale Nonlinear Optimization, Di Pillo G. and Roma M., eds., Nonconvex Optimization and Its Applications, Springer US, Boston, MA, 2006, pp. 35–59.

[25] CALAFIORE G.C. AND CAMPI M.C., *The scenario approach to robust control design*, IEEE Transactions on Automatic Control, 51 (2006), pp. 742–753.

[26] CALAFIORE G.C. AND FAGIANO L., *Robust Model Predictive Control via Scenario Optimization*, IEEE Transactions on Automatic Control, 58 (2013), pp. 219–224.

[27] CASIANO M.J., HULKA J.R., AND YANG V., *Liquid-Propellant Rocket Engine Throttling: A Comprehensive Review*, Journal of Propulsion and Power, 26 (2010), pp. 897–923.

[28] CHAMPAGNAT R., ESTEBAN P., PINGAUD H., AND VALETTE R., *Petri net based modeling of hybrid systems*, Computers in Industry, 36 (1998), pp. 139–146.

[29] CHAPMAN J.W., LAVELLE T.M., MAY R.D., LITT J.S., AND GUO T.H., *Propulsion System Simulation Using the Toolbox for the Modeling and Analysis of Thermodynamic Systems (T MATS)*, in Propulsion and energy forum in 50th AIAA/ASME/SAE/ASEE Joint Propulsion Conference, vol. 2014-3929, Cleveland, USA, July 2014, AIAA.

[30] CHEN H. AND ALLGOEWER F., *A Quasi-Infinite Horizon Nonlinear Model Predictive Control Scheme with Guaranteed Stability*, Automatica, 34 (1998), pp. 1205–1217.

[31] CHEN Y., BRUSCHETTA M., PICOTTI E., AND BEGHI A., *MATMPC - A MATLAB Based Toolbox for Real-time Nonlinear Model Predictive Control*, in 2019 18th European Control Conference (ECC), June 2019, pp. 3365–3370.

[32] CHISCI L., ROSSITER J.A., AND ZAPPA G., *Systems with persistent disturbances: predictive control with restricted constraints*, Automatica, 37 (2001), pp. 1019–1028.

[33] CHOPINET J-N., LASSOUDIERE F., ROZ G., FAYE O., LE GONIDEC S., ALLIOT P., GUEDRON S., AND PINHEDE H., *Progress of the development of an all-electric control system of a rocket engine*, in 48th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit, vol. AIAA 2012-3873, Atlanta, USA, 2012, American Institute of Aeronautics and Astronautics.

[34] CRASSIDIS J.L. AND MARKLEY F.L., *Sliding mode control using modified Rodrigues parameters*, Journal of Guidance, Control, and Dynamics, 19 (1996), pp. 1381–1383.

[35] CROCCO L., *Aspects of Combustion Stability in Liquid Propellant Rocket Motors Part I: Fundamentals. Low Frequency Instability With Monopropellants*, Journal of the American Rocket Society, 21 (1951), pp. 163–178.

[36] DAI X. AND RAY A., *Damage-Mitigating Control of a Reusable Rocket Engine: Part I-Life Prediction of the Main Thrust Chamber Wall*, Journal of Dynamic Systems, Measurement, and Control, 118 (1996), pp. 401–408. - NASA.

[37] ——, *Damage-Mitigating Control of a Reusable Rocket Engine: Part II-Formulation of an Optimal Policy*, Journal of Dynamic Systems, Measurement, and Control, 118 (1996), pp. 409–415. - NASA.

[38] DENEU F., MALASSIGNE M., LE-COULS O., AND BAIOCCO P., *Promising solutions for fully reusable two-stage-to-orbit configurations*, Acta Astronautica, 56 (2005), pp. 729–736.

[39] DI MATTEO F., *Modelling and Simulation of Liquid Rocket Engine Ignition Transients*, ph.D. Thesis, Sapienza Universita di Roma, 2011.

[40] DI MATTEO F., DE ROSA M., AND ONOFRI M., *Start-Up Transient Simulation of a Liquid Rocket Engine*, in 47th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, San Diego, USA, 2011, American Institute of Aeronautics and Astronautics.

[41] ——, *Transient Simulation of the RL-10a-3-3a Rocket Engine*, in Space Propulsion Conference, Bordeaux, France, 2012.

[42] DRESIA K., WAXENEGGER-WILFING G., RICCIUS J., DEEKEN J.C., AND OSCHWALD M., *Numerically Efficient Fatigue Life Prediction of Rocket Combustion Chambers using Artificial Neural Networks*, in 8TH EUROPEAN CONFERENCE FOR AERONAUTICS AND AEROSPACE SCIENCES (EUCASS), Madrid, Spain, 2019, p. 16.

[43] DURTESTE S., *A Transient Model of the VINCI Cryogenic Upper Stage Rocket Engine*, in 43rd AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit, Cincinnati, USA, 2007, American Institute of Aeronautics and Astronautics.

[44] DUYAR A., TEN-HUEI G., AND MERRILL W.C., *Space shuttle main engine model identification*, IEEE Control Systems Magazine, 10 (1990), pp. 59–65. - NASA.

[45] FERRARI-TRECATE G., GALLESTEY E., LETIZIA P., SPEDICATO M., MORARI M., AND ANTOINE M., *Modeling and Control of Co-Generation Power Plants: A Hybrid System Approach*, IEEE Transactions on Control Systems Technology, 12 (2004), pp. 694–705.

[46] FIERRO R., LEWIS F.L., AND LIU K., *Hybrid control system design using a fuzzy logic interface*, Circuits, systems, and signal processing, 17 (1998), pp. 401–419.

[47] FILATYEV A.S., BUZULUK V., YANOVA O., RYABUKHA N., AND PETROV A., *Advanced aviation technology for reusable launch vehicle improvement*, Acta Astronautica, 100 (2014), pp. 11–21.

[48] FINDEISEN R. AND ALLGOEWER F., *An Introduction to Nonlinear Model Predictive Control*, in 21st Benelux Meeting on Systems and Control, Veldhoven, The Netherlands, 2002.

[49] GADEWADIKAR J., LEWIS F.L., SUBBARAO K., PENG K., AND CHEN B.M., *H-Infinity Static Output-feedback Control for Rotorcraft*, Journal of Intelligent and Robotic Systems, 54 (2009), pp. 629–646.

[50] GANGULI S., MARCOS A., AND BALAS G., *Reconfigurable LPV control design for Boeing 747-100/200 longitudinal axis*, in Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301), vol. 5, Anchorage, USA, May 2002, pp. 3612–3617 vol.5.

[51] GILBERT W., HENRION D., BERNUSSOU J., AND BOYER D., *Polynomial LPV synthesis applied to turbofan engines*, Control Engineering Practice, 18 (2010), pp. 1077–1083.

[52] GOEBEL R., SANFELICE R.G., AND TEEL A., *Hybrid dynamical systems*, IEEE Control Systems Magazine, 29 (2009), pp. 28–93.

[53] GORDON S. AND MCBRIDE B.J., *Computer Program for Calculation of Complex Chemical Equilibrium Compositions and Applications*, Tech. Rep. NASA Reference Publication 1311, NASA Glenn Research Center, Cleveland, Ohio, USA, 1994.

[54] HOUSKA B., FERREAU H.J., AND DIEHL M., *ACADO toolkit - An open-source framework for automatic control and dynamic optimization*, Optimal Control Applications and Methods, 32 (2011), pp. 298–312.

[55] HUZEL D.K. AND HUANG D.H., *Chapter 7: Design of Rocket-Engine Control and Conditon-Monitoring Systems*, in Modern Engineering for Design of Liquid-Propellant Rocket Engines, AIAA Progress in Astronautics and Aeronautics Richard Seebass, 1992. Sponsored by Rocketdyne Division of Rockwell International.

[56] IANNETTI A., *Model-based fault diagnosis for rocket engines - Methodes de diagnostic pour les moteurs de fusee a ergols liquides*, ph.D. Thesis, Universite Paris-Saclay, 2016.

[57] IFFLY A. AND BRIXHE M., *Performance Model of the Vulcain Ariane 5 Main Engine*, in 35th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, vol. AIAA 99-2472, Los Angeles, USA, 1999, AIAA. - SNECMA (ArianeGroup).

[58] IOOSS B. AND LEMAÎTRE P., *A Review on Global Sensitivity Analysis Methods*, in Uncertainty Management in Simulation-Optimization of Complex Systems: Algorithms and Applications, Operations Research/Computer Science Interfaces Series, Springer US, Boston, MA, 2015, pp. 101–122.

[59] JÚLVEZ J., DI CAIRANO S., BEMPORAD A., AND MAHULEA C., *Event-driven model predictive control of timed hybrid Petri nets*, International Journal of Robust and Nonlinear Control, 24 (2012), pp. 1724–1742.

[60] KAI T., NIU K., OBASE K., SAKAI W., FUKUDA Y., HASHIMOTO T., SATO M., TAKADA S., KIMURA T., NARUO Y., OGAWA H., YAGISHITA T., AND ITO T., *Engine Control System for the Main Engine of the Reusable Sounding Rocket*, in Proceedings of the International Astronautical Congress, IAC, vol. 10 of IAC-15,C4,3,2,x28758, Jerusalem, Israel, 2015, IAF, pp. 7389–7394. - JAXA and MHI.

[61] KANMURI A., KANDA T., WAKAMATSU Y., TORII Y., AND KAGAWA E., *Transient analysis of LOX/LH2 rocket engine (LE-7)*, in 25th Joint Propulsion Conference, Monterey, USA, 1989, American Institute of Aeronautics and Astronautics.

[62] KHALIL H.K., *Nonlinear Systems*, Prentice Hall, Upper Saddle River, NJ, USA, 1996.

[63] KIFORENKO B.N. AND KHARITONOV A.M., *Control of Thrust of Liquid Rocket Engines: Simulation and Optimization*, Journal of Automation and Information Sciences, 32 (2000), pp. 47–63.

[64] KIM J.H., OH S.-H., CHO D.-I., AND HEDRICK J.K., *Robust Discrete-Time Variable Structure Control Methods*, Journal of Dynamic Systems, Measurement, and Control, 122 (2000), pp. 766–775.

[65] KLEIN M., HAYOUN D., LE GONIDEC S., AND REICHSTADT S., *Method and a circuit for regulating a rocket engine*, 2017. - SNECMA (ArianeGroup), Issy-les-Moulineaux, France, US 2017/0101963 A1, United States Patent Application Publication.

[66] KOLCIO K., HELMICKI A.J., AND JAWEED S., *Propulsion system modelling for condition monitoring and control : Part I, theoretical foundations*, in 30th AIAA/ASME/SAE/ASEE Joint Propulsion Conference, vol. AIAA 94-3227, Indianapolis, USA, 1994, AIAA.

[67] KOTHARE M.V., BALAKRISHNAN V., AND MORARI M., *Robust constrained model predictive control using linear matrix inequalities*, Automatica, 32 (1996), pp. 1361–1379.

[68] 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, Aug. 2005, American Institute of Aeronautics and Astronautics.

[69] KWON S., SHIMOMURA T., AND OKUBO H., *Pointing control of spacecraft using two SGCMGs via LPV control theory*, Acta Astronautica, 68 (2011), pp. 1168–1175.

[70] LAVELLE T.M., CHAPMAN J.W., MAY R.D., LITT J.S., AND GUO T.H., *Cantera Integration with the Toolbox for Modeling and Analysis of Thermodynamic Systems (T-MATS)*, in AIAA/ASME/SAE/ASEE Joint Propulsion Conference, Cleveland, USA, July 2014.

[71] LE FUR A., VOISIN C., GUILLARD M.H., AND VERGE M.M., *Study of control laws of an expander-cycle rocket engine - Etude de lois de commande pour moteur a cycle expander*, ArianeGroup internal report, ArianeGroup, ENSAM, LAAM, Vernon, France, 1997.

[72] LE GONIDEC S., *Device for controlling a regulated system, and an engine including such a device*, 2011. - Safran Aircraft Engines SAS, France, US8005554B2, United States Patent.

[73] ——, *Method and a device for calculating a starting or stop sequence for an engine*, 2013. - SNECMA (ArianeGroup), Vernon, France, US8364374B2, United States Patent.

[74] ——, *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.

[75] ——, *A method of controlling the pressure and a mixture ratio of a rocket engine, and corresponding device*, 2017. - Safran Aircraft Engines SAS (ArianeGroup), France, FR3039859B1, Institut National de la Propriete Industrielle (France).

[76] ——, *An overview of connections between scientific automatic topics and their applications in the propulsive systems*, in Journal of Physics: Conference Series: 13th European Workshop on Advanced Control and Diagnosis (ACD 2016), 783/011001, Lille, France, Jan. 2017, IOP Publishing.

[77] LE GONIDEC S. AND FAYE O., *Device for adjusting an operating variable of an engine*, 2015. - SNECMA (ArianeGroup), France, US9037380B2, United States Patent.

[78] LEFEBVRE J., ROUSSEL H., WALTER E., LECOINTE D., AND TABBARA W., *Prediction from wrong models: the Kriging approach*, IEEE Antennas and Propagation Magazine, 38 (1996), pp. 35–45.

[79] LENNARTSON B., EGARDT B., AND TITTUS M., *Hybrid systems in process control*, in 33rd Conference on Decision and Control, vol. 4, Lake Buena Vista, USA, 1994, IEEE, pp. 3587–3592.

[80] LEVINE W.S. AND REICHERT R.T., *An introduction to H/sub infinity / control system design*, in 29th IEEE Conference on Decision and Control, Dec. 1990, pp. 2966–2974 vol.6.

[81] LIU K. AND ZHANG Y.L., *A study on versatile simulation of liquid propellant rocket engine systems transients*, in 36th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Las Vegas, USA, 2000, American Institute of Aeronautics and Astronautics.

[82] LOEFBERG J., *Approximations of closed-loop minimax MPC*, in 42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475), vol. 2, Maui, Hawaii, USA, Dec. 2003, pp. 1438–1442 Vol.2.

[83] ——, *Minimax Approaches to Robust Model Predictive Control*, no. No. 812 in Linkoeping Studies in Science and Technology Dissertations, UniTryck, Linkoeping University Electronic Press, Apr. 2003. Google-Books-ID: 96VYBQAAQBAJ.

[84] ——, *YALMIP : a toolbox for modeling and optimization in MATLAB*, in 2004 IEEE International Conference on Robotics and Automation (IEEE Cat. No.04CH37508), Taipei, Taiwan, 2004, IEEE, pp. 284–289.

[85] ——, *Automatic robust convex programming*, Optimization Methods and Software, 27 (2012), pp. 115–129.

[86] LORENZO C.F., *Robust rocket engine concept*, in 31st Joint Propulsion Conference and Exhibit, San Diego, USA, 1995, American Institute of Aeronautics and Astronautics.

[87] LORENZO C.F., MERRILL W.C., MUSGRAVE J.L., AND RAY A., *Controls Concepts for Next Generation Reusable Rocket Engines*, in American Control Conference, vol. FP1-4:10, Seattle, USA, 1995, IEEE, pp. 3942–3950. - NASA Lewis Research Center.

[88] LORENZO C.F. AND MUSGRAVE J.L., *Overview of rocket engine control*, in AIP Conference Proceedings, vol. 246, AIP, 1992, pp. 446–455. - NASA.

[89] 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 (2001), pp. 998–1004. - NASA.

[90] LOZANO-TOVAR P.C., *Dynamic models for liquid rocket engines with health monitoring application*, Master's thesis, Massachusetts Institute of Technology, MIT Boston, 1998.

[91] LUNZE J. AND LAMNABHI-LAGARRIGUE F., *Handbook of Hybrid Systems Control: Theory, Tools, Applications*, Cambridge University Press, Oct. 2009. Google-Books-ID: pPLRV3ehMVIC.

[92] LUNZE J., NIXDORF B., AND RICHTER H., *Hybrid modelling of continuous-variable systems with application to supervisory control*, in European Control Conference (ECC), Brussels, Belgium, 1997, IEEE, pp. 1376–1381.

[93] 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, vol. 2, Boston, USA, 2004, pp. 1695–1700.

[94] MAGNI L. AND SCATTOLINI R., *Stabilizing model predictive control of nonlinear continuous time systems*, Annual Reviews in Control, 28 (2004), pp. 1–11.

[95] MAIWORM M., BAETHGE T., AND FINDEISEN R., *Scenario-based Model Predictive Control: Recursive Feasibility and Stability*, IFAC-PapersOnLine, 48 (2015), pp. 50–56.

[96] MANFLETTI C., *Transient behaviour modelling of liquid rocket engine components*, vol. PhD Thesis 2009 of Berichte aus der Luft- und Raumfahrttechnik, Shaker Verlag, Aachen, 2010. OCLC: 685183465.

[97] MARCOS A., PENIN L., LE GONIDEC S., AND LEMAITRE A., *HMS-Control-Interaction architecture for rocket engines*, in AIAA Guidance, Navigation, and Control Conference, vol. AIAA 2012-4679, Minneapolis, USA, 2012, American Institute of Aeronautics and Astronautics.

[98] 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), Nice, France, Oct. 2010, pp. 505–510.

[99] MATTINGLEY J. AND BOYD S., *CVXGEN: a code generator for embedded convex optimization*, Optimization and Engineering, 13 (2012), pp. 1–27.

[100] MAYNE D.Q., RAKOVIC S.V., FINDEISEN R., AND ALLGOEWER F., *Robust output feedback model predictive control of constrained linear systems*, Automatica, 42 (2006), pp. 1217–1222.

[101] MAYNE D.Q., RAWLINGS J.B., RAO C.V., AND SCOKAERT P.O.M., *Constrained model predictive control: Stability and optimality*, Automatica, 36 (2000), pp. 789–814.

[102] MCDERMOTT C.E., BRESHEARS R.R., AND MCCAFFERTY J., *Dynamic performance of surveyor throttleable rocket engine operating on propellants containing dissolved gas*, in AIAA Third Annual Meeting, vol. AIAA 66-949, Boston, USA, 1966, AIAA. - Jet Propulsion Laboratory.

[103] MENDEZ CUBILLOS X.C. AND DE SOUZA L.C.G., *Using of H-Infinity Control Method in Attitude Control System of Rigid-Flexible Satellite*, 2009.

[104] MIOSSEC S., YOKOI K., AND KHEDDAR A., *Development of a software for motion optimization of robots - Application to the kick motion of the HRP-2 robot*, in 2006 IEEE International Conference on Robotics and Biomimetics, Dec. 2006, pp. 299–304.

[105] MOSTERMAN P.J., *Hybrid Dynamic Systems: A hybrid bond graph modeling paradigm and its application in diagnosis*, ph.D. Thesis, Vanderbilt University, Nashville, USA, 1997.

[106] MUNOZ DE LA PENA D., BEMPORAD A., AND FILIPPI C., *Robust explicit MPC based on approximate multi-parametric convex programming*, in 2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601), vol. 3, Dec. 2004, pp. 2491–2496 Vol.3.

[107] MURILO A., ALAMIR M., AND ORTNER P., *A Real-Time Implementable NMPC Output Feedback for a Diesel Engine Air Path*, Oil & Gas Science and Technology – Revue d'IFP Energies nouvelles, 66 (2011), pp. 613–625.

[108] MUSGRAVE J.L., *Linear quadratic servo control of a reusable rocket engine*, Journal of Guidance, Control and Dynamics, 15 (1992), pp. 1149–1154. - NASA.

[109] 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 (1996), pp. 100–109. - NASA.

[110] MUSKE K.R. AND BADGWELL T.A., *Disturbance modeling for offset-free linear model predictive control*, Journal of Process Control, 12 (2002), pp. 617–632.

[111] NASSIRHARAND A. AND KARIMI H., *Mixture ratio control of liquid propellant engines*, Aircraft Engineering and Aerospace Technology, 77 (2005), pp. 236–242.

[112] NEGENBORN R.R., BECCUTI A.G., DEMIRAY T., LEIRENS S., DAMM G., DE SCHUTTER B., AND MORARI M., *Supervisory hybrid model predictive control for voltage stability of power networks*, in American Control Conference ACC'07, New York City, USA, 2007, IEEE, pp. 5444–5449.

[113] NEILL T., JUDD J., VEITH E., AND ROUSAR D., *Practical uses of liquid methane in rocket engine applications*, Acta Astronautica, 65 (2009), pp. 696–705.

[114] NEMETH E., ANDERSON R., MARAM J., NORMAN A., AND MERRILL W.C., *An advanced intelligent control system framework*, in 28th Joint Propulsion Conference and Exhibit, vol. AIAA92-3162, Nashville, USA, 1992, American Institute of Aeronautics and Astronautics.

[115] NEMETH E., ANDERSON R., OLS J., AND OLSASKY M., *Reusable rocket engine intelligent control system framework design, phase 2*, Tech. Rep. NASA Contractor Report 187213, Rockwell International, Canoga Park, California, Sept. 1991.

[116] ORDONNEAU G., MASSE J., AND ALBANO G., *CARINS: Modelling and simulation software for complex industrial processes based on open-source programmes// Un logiciel de modelisation et de simulation pour les procedes industriels complexes fonde sur des logiciels libres*, REE: Revue de l'Electricite et de l'Electronique, N. 4 (2006). - CNES-ONERA.

[117] OTTO E.W. AND FLAGE R.A., *Control of combustion-chamber pressure and oxidant-fuel ratio for a regeneratively cooled hydrogen-fluorine rocket engine*, Tech. Rep. Technical note D-82, NASA Lewis Research Center, Cleveland, USA, 1959.

[118] 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), Budapest, Hungary, July 2018, pp. 58–67.

[119] 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 IFAC-PapersOnLine, vol. 52-12 of 21st IFAC Symposium on Automatic Control in Aerospace ACA 2019, Cranfield, UK, Aug. 2019, pp. 268–273.

[120] PÉREZ-ROCA S., MARZAT J., PIET-LAHANIER H., LANGLOIS N., FARAGO F., GALEOTTA M., AND LE GONIDEC S., *Robust Transient Control of Reusable Liquid-Propellant Rocket Engines*, in 70th International Astronautical Congress (IAC) (to appear), Washington D.C., USA, Oct. 2019, International Astronautical Federation (IAF).

[121] ——, *A survey of automatic control methods for liquid-propellant rocket engines*, Progress in Aerospace Sciences, 107 (2019), pp. 63–84.

[122] ——, *Trajectory planning and tracking via MPC for transient control of liquid-propellant rocket engines*, in 15th European Workshop on Advanced Control and Diagnosis, ACD 2019 (to appear), Bologna, Italy, Nov. 2019, Springer.

[123] PLD SPACE, *News release for ESA's Future Launcher Preparatory Programme project award to PLD Space*, 2016.

[124] RAY A. AND DAI X., *Damage-Mitigating Control of a Reusable Rocket Engine for High Performance and Extended Life*, Tech. Rep. NASA Contractor Report 4640, Lewis Research Center, 1995.

[125] REBERGA L., HENRION D., BERNUSSOU J., AND VARY F., *LPV Modeling of a Turbofan Engine*, IFAC Proceedings Volumes, 38 (2005), pp. 526–531.

[126] ROCKETDYNE PROPULSION & POWER AND BOEING, *Space Shuttle Main Engine Orientation, in Space Transportation System Training Data BC98-04*, 1998. - Boeing, Rocketdyne Propulsion & Power.

[127] ROMET A., CLIQUET MORENO E., ESPINOSA RAMOS A., DREYER S., STILLACE T., GODI M., BRUNEAU Q., TATIOSSIAN P., JUES T., AND LE GONIDEC S., *A flexible real-time simulation platform dedicated to embedded rocket engine control systems development and testing*, in 7th European Conference for Aeronautics and Space Sciences (EUCASS), vol. 162, Milan, Italy, 2017, EUCASS Association.

[128] ROSIER L., *A survey of controllability and stabilization results for partial differential equations*, Journal Europeen des Systemes Automatises, 41 (2007), pp. 365–412.

[129] RTO/NATO, *Active control of engine dynamics, RTO Educational notes 20*, 2002. - OCLC: 249585275.

[130] RUTH E., AHN H., BAKER R., AND BROSMER M., *Advanced liquid rocket engine transient model*, in 26th Joint Propulsion Conference, Orlando, USA, 1990, American Institute of Aeronautics and Astronautics.

[131] SAINT-MARD M., HENDRICK P., AND PROMPER C., *Rocket engines performance evaluation and regulation properties*, in 35th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, vol. AIAA 99-2326, Los Angeles, USA, 1999, AIAA.

[132] SANTANA A., BARBOSA F.I., NIWA M., AND GOES L.C.S., *Modeling and Robust Analysis of a Liquid Rocket Engine*, in 36th Joint Propulsion Conference & Exhibit, vol. 2000-3160, Huntsville, USA, 2000, AIAA.

[133] SANTANA A. AND GOES L.C.S., *Design and dynamic characteristics of a liquid-propellant thrust chamber*, 1999.

[134] ——, *Dynamic modeling and stability analysis of a liquid rocket engine*, in 15th Brazilian Congress of Mechanical Engineering COBEM, Sao Paulo, Brazil, 1999, Centro Tecnico Aeroespacial, IAE/CTA.

[135] 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 (2001), pp. 847–857.

[136] SARABIA D., CAPRARO F., LARSEN L.F.S., AND DE PRADA C., *Hybrid NMPC of supermarket display cases*, Control Engineering Practice, 17 (2009), pp. 428–441.

[137] SAROTTE C., *Improvement of monitoring and reconfiguration processes for liquid-propellant rocket engines - Amelioration des processus de surveillance et de reconfiguration pour les moteurs fusee a ergols liquides*, ph.D. Thesis, Universite Paris-Saclay, Oct. 2019.

[138] SAROTTE C., MARZAT J., PIET-LAHANIER H., GALEOTTA M., AND ORDONNEAU G., *Fault Detection and Isolation with Fluid Mechanics Constraints For Cryogenic Combustion Bench Cooling Circuit*, in Proceedings of the Annual Conference of the PHM Society, vol. 10(1), Sept. 2018.

[139] ——, *Anti-windup Design for Linear Discrete-time Systems Subject to Actuator Additive Faults and Saturations*, in 2019 American Control Conference (ACC), Philadelphia, USA, July 2019, IEEE, pp. 3734–3739.

[140] 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, vol. 51, Warsaw, Poland, 2018, Elsevier, pp. 592–599. - ONERA/CNES.

[141] 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.

[142] SEITZ P.F. AND SEARLE R.F., *Space Shuttle Main Engine control system*, in National Aerospace Engineering and Manufacturing Meeting, Los Angeles, USA, 1973. - NASA NTRS.

[143] SEOK J., KOLMANOVSKY I., AND GIRARD A., *Integrated/coordinated control of aircraft gas turbine engine and electrical power system: Towards large electrical load handling*, in 2016 IEEE 55th Conference on Decision and Control (CDC), Las Vegas, USA, Dec. 2016, pp. 3183–3189.

[144] SHAFIEY DEHAJ M., EBRAHIMI R., AND KARIMI H., *Mathematical modeling and analysis of cutoff impulse in a liquid-propellant rocket engine*, Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, 229 (2015), pp. 2358–2374.

[145] SHIMA T., IDAN M., AND GOLAN O.M., *Sliding-Mode Control for Integrated Missile Autopilot Guidance*, Journal of Guidance, Control, and Dynamics, 29 (2006), pp. 250–260.

[146] SHIMOMURA T., *Hybrid control of gain-scheduling and switching: a design example of aircraft control*, in American Control Conference, vol. 6, Denver, USA, 2003, IEEE, pp. 4639–4644.

[147] SHURMAN J., *Calculus and Analysis in Euclidean Space*, Undergraduate Texts in Mathematics, Springer International Publishing, 2016.

[148] 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), vol. 3, Arlington, VA, USA, June 2001, pp. 2301–2308 vol.3.

[149] SOBOL I.M., *Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates*, Mathematics and Computers in Simulation, 55 (2001), pp. 271–280.

[150] SOLTANI M., IZADI-ZAMANABADI R., WISNIEWSKI R., BELAU W., AND LE GONIDEC S., *Robust Parametric Fault Estimation in A Hopper System*, IFAC Proceedings Volumes, 45 (2012), pp. 491–498.

[151] SONNTAG C., SU W., STURSBERG O., AND ENGELL S., *Optimized start-up control of an industrial-scale evaporation system with hybrid dynamics*, Control Engineering Practice, 16 (2008), pp. 976–990.

[152] SPACEX CORP., *Merlin Engines*, 2015.

[153] STEINFELD J.I., FRANCISCO J.S., AND HASE W.L., *Chemical Kinetics and Dynamics*, Prentice Hall, second edition ed., 1989.

[154] SUNAKAWA H., KUROSU A., OKITA K., SAKAI W., MAEDA S., AND OGAWARA A., *Automatic Thrust and Mixture Ratio Control of the LE-X*, in 44th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit, vol. 2008-4666, Hartford, USA, 2008, AIAA. - JAXA and MHI.

[155] SUTTON G.P. AND BIBLARZ O., *Rocket propulsion elements*, Wiley-Interscience publication, John Wiley and Sons, New York, 7. ed ed., 2001.

[156] TATIOSSIAN P., DESMARIAUX J., AND GARCIA M., *CALLISTO project - reusable first stage rocket demonstrator*, in 7TH EUROPEAN CONFERENCE FOR AERONAUTICS AND SPACE SCIENCES (EUCASS), vol. 680, Milan, Italy, 2017, EUCASS Association.

[157] TAWARMALANI M. AND SAHINIDIS N.V., *A polyhedral branch-and-cut approach to global optimization*, Mathematical Programming, 103 (2005), pp. 225–249.

[158] TIMNAT Y.M., *Advanced Chemical Rocket Propulsion*, Academic Press, London, Orlando, 1st edition ed., Nov. 1987.

[159] TURKI M., *Synthesis of self-adaptive predictive controllers for optimizing system performance - Synthese de controleurs predictifs auto-adaptatifs pour l'optimisation des performances des systemes*, ph.D. Thesis, Normandie, Oct. 2018.

[160] UMEDA Y., MARUYAMA T., ANAI H., EJIRI A., AND SHIMOTANI K., *A Fast Model Predictive Control Algorithm for Diesel Engines*, IFAC Proceedings Volumes, 45 (2012), pp. 466–471.

[161] VAN DER SCHAFT A.J. AND SCHUMACHER J.M., *An introduction to hybrid dynamical systems*, vol. 251, Springer London, 2000.

[162] VOINOT O., APKARIAN P., AND ALAZARD D., *Gain-Scheduling H-infinity Control of the Launcher in Atmospheric Flight via Linear-Parameter Varying Techniques*, in AIAA Guidance, Navigation, and Control Conference and Exhibit, Monterey, USA, 2002, American Institute of Aeronautics and Astronautics.

[163] VON GRAEVE C., DI MATTEO F., AND IERARDO N., *Transient Simulation Capabilities for the ORION-ESM Propulsion System Development*, in Space Propulsion Conference, Rome, Italy, 2016.

[164] 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 (2006), pp. 25–57.

[165] WAXENEGGER-WILFING G., DRESIA K., DEEKEN J.C., AND OSCHWALD M., *Heat Transfer Prediction for Methane in Regenerative Cooling Channels with Neural Networks*, arXiv:1907.11281 [physics, stat], (2019). arXiv: 1907.11281.

[166] WU M.-K. AND RAY A., *Damage-Mitigating Control of Power Systems for Structural Durability and High Performance*, Journal of Engineering for Gas Turbines and Power, 117 (1995), pp. 307–313. - NASA.

[167] WU Y., ZHANG H., HUI J., AND ZHOU X., *Non-fragile Hinf controller for combustion process in rocket motors*, International Journal of Automation and Control, 12 (2018), pp. 381–398.

[168] YANG E., XU Y., AND ZHANG Z., *Nonlinear dynamic neural network model for rocket propulsion systems*, Tuijin Jishu/Journal of Propulsion Technology, 22 (2001), pp. 50–53.

[169] YEDAVALLI R.K., *Improved measures of stability robustness for linear state space models*, IEEE Transactions on Automatic Control, 30 (1985), pp. 577–579.

[170] ZAMBRANO D., BORDONS C., GARCIA-GABIN W., AND CAMACHO E.F., *A solar cooling plant: a benchmark for hybrid systems control*, IFAC Proceedings Volumes, 39 (2006), pp. 199–204.

[171] ZAMBRANO D. AND GARCIA-GABIN W., *Hierarchical Control of a Hybrid Solar Air Conditioning Plant*, European Journal of Control, 14 (2008), pp. 464–483.

[172] ZHANG X.-Y., TRAME M.N., LESKO L.J., AND SCHMIDT S., *Sobol Sensitivity Analysis: A Tool to Guide the Development and Evaluation of Systems Pharmacology Models*, CPT: Pharmacometrics & Systems Pharmacology, 4 (2015), pp. 69–79.

[173] ZHANG Y. AND JIANG J., *Bibliographical review on reconfigurable fault-tolerant control systems*, Annual Reviews in Control, 32 (2008), pp. 229–252.

[174] ZHANG Y.L., *State-space analysis of the dynamic characteristics of a variable thrust liquid propellant rocket engine*, Acta Astronautica, 11 (1984), pp. 535–541.

[175] ZHENG F., CHENG M., AND GAO W.B., *Variable structure control of time-delay systems with a simulation study on stabilizing combustion in liquid propellant rocket motors*, Automatica, 31 (1995), pp. 1031–1037.

[176] ZHENG F. AND FRANK P.M., *Robust control of uncertain distributed delay systems with application to the stabilization of combustion in rocket motor chambers*, Automatica, 38 (2002), pp. 487–497.

[177] ZHENG Y. AND RUNDELL A., *Comparative study of parameter sensitivity analyses of the TCR-activated erk-MAPK signalling pathway*, IEE Proceedings - Systems Biology, 153 (2006), pp. 201–211.

[178] ZHOU X., *The analysis for regulation-performance of a variable thrust rocket engine control system*, Tech. Rep. FTD ID RS T 522-82 ; AD A 117495, USA Foreign Technology Division, China, 1982. Edited translation.

[179] ZINNECKER A., CHAPMAN J.W., LAVELLE T.M., AND LITT J.S., *Development of a twin-spool turbofan engine simulation using the Toolbox for Modeling and Analysis of Thermodynamic Systems (T-MATS)*, in 50th AIAA/ASME/SAE/ASEE Joint Propulsion Conference, Cleveland, USA, 2015, American Institute of Aeronautics and Astronautics.

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre: Commande robuste basée modèle des régimes transitoires des moteurs fusée à ergols liquides réutilisables

Mots clés: commande prédictive MPC, moteur fusée à ergols liquides, robustesse, transitoires, modélisation non-linéaire orientée commande, paramètres variants

Résumé: La tendance actuelle vers un accès plus abordable à l'espace se traduit par des lanceurs et moteurs réutilisables. Du point de vue du contrôle, ces moteurs fusée à propergol liquide (MFPL) réutilisables impliquent des spécifications de robustesse plus exigeantes que ceux à usage unique, principalement en raison de leurs capacités de redémarrage multiple et de modulation de poussée. Classiquement, le système de contrôle gère les opérations des MFPL autour d'un ensemble fini de points prédéfinis. Cette approche réduit leur domaine de modulation à un intervalle restreint dans lequel ils sont conçus pour être sûrs. De plus, les phases transitoires, qui ont un impact important sur la vie du moteur, ne sont pas exécutées de manière robuste. L'objectif de ce travail est donc de développer une boucle de régulation adaptée à l'ensemble des phases d'opération (transitoire et régime permanent) et robuste aux variations paramétriques internes. Plusieurs blocs ont été développés pour constituer la boucle de régulation : simulation de moteur, génération de référence et contrôleurs. Des simulateurs représentatifs des moteurs à cycle générateur de gaz ont tout d'abord été construits. La modélisation purement thermodynamique du cycle a ensuite été adaptée au contrôle, afin d'obtenir des modèles non-linéaires sous forme d'état. Dans ces modèles, l'influence des entrées de commande continues (ouvertures des vannes) et des entrées discrètes (activation des allumeurs et démarreur) est considérée dans un cadre hybride simplifié. La sous-phase continue du transitoire de démarrage est contrôlée en boucle fermée pour suivre des trajectoires de référence pré-calculées. Outre le démarrage, les scénarios de modulation présentent également un algorithme pour le suivi des états finaux. Une méthode de contrôle à base de modèles, la commande prédictive, a été appliquée de manière linéarisée avec des considérations de robustesse à tous ces scénarios, dans lesquels des contraintes dures doivent être respectées. Le suivi des points de fonctionnement en pression (poussée) et du rapport de mélange dans l'enveloppe de conception est atteint en simulation tout en respectant les contraintes. La robustesse aux variations des paramètres, qui sont identifiés comme prédominants par des analyses, est également démontrée. Ce travail ouvre la voie à la validation expérimentale par des simulations hardware-in-the-loop ou des tests sur banc d'essai.

Title: Model-based robust transient control of reusable liquid-propellant rocket engines

Keywords: MPC control, liquid-propellant rocket engine, robustness, transients, control-oriented nonlinear modelling, parameter-varying

Abstract: The current trend towards a more affordable access to space is 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 due to 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 narrow interval in which they are designed to be safe. Moreover, transient phases, which have a great impact on engine life, are not robustly operated. 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 developed to constitute the control loop: engine simulation, reference generation and controllers. First, simulators representative of the gas-generator-cycle engines were built. The purely thermo-fluid-dynamic modelling of the cycle was subsequently adapted to control, obtaining nonlinear state-space models. In these models, the influence of continuous control inputs (valve openings) and of discrete ones (igniters and starter activations) is considered within a simplified hybrid approach. The continuous sub-phase of the start-up transient is feedback controlled to track pre-computed reference trajectories. Beyond the start-up, throttling scenarios also present an end-state-tracking algorithm. A model-based control method, Model Predictive Control, has been applied in a linearised manner with robustness considerations to all these scenarios, in which a set of hard constraints must be respected. Tracking of pressure (thrust) and mixture-ratio operating points within the design envelope is achieved in simulation while respecting constraints. Robustness to variations in the parameters, which are checked to be predominant according to analyses, is also demonstrated. This framework paves the way to experimental validation via hardware-in-the-loop simulations or in test benches.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France