POLITECNICO DI MILANO
AEROSPACE ENGINEERING DEPARTMENT
DOCTORAL PROGRAM IN AEROSPACE ENGINEERING

MULTIDISCIPLINARY DESIGN OPTIMIZATION FOR EXPENDABLE LAUNCH VEHICLES

Doctoral Dissertation of:
Francesco Castellini
Matr. 738531

Supervisor:
Prof. Michèle Lavagna

Tutor:
Prof. Luciano Galfetti

The Chair of the Doctoral Program:
Prof. Sergio Ricci

Year 2012 – Cycle XXIV

Without question, the greatest invention in the history of mankind is beer.
Oh, I grant you that the wheel was also a fine invention, but the wheel does not go nearly as well with pizza.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank all the people who contributed to the work described in this dissertation in many different ways: fractions of code, financial and technical support, advices, discussions, and sharing of expertise. In particular, I would like to thank Michèle who has always backed my line of research at Politecnico di Milano, also providing useful advices in the short breaks of her super-busy schedule. And of course Annalisa, we've been working together so long that we're starting to read each other's mind about our research issues. From all those I got to know in the months I spent at ESA, I would especially like to mention the people at the former TEC-ECM section (now TEC-ECN): Antonio Rinalducci (whose knowledge on launchers is source of great admiration) and Alvaro Martinez Barrio, who have been technical officers of the PRESTIGE activity, always providing extremely valuable inputs, Guillermo Ortega, who supported me and Annalisa throughout almost four years of work, and Sven Erb for its technical support and interest in the MDO for launchers. I would also like to thank Javier Ventura-Traveset and Carlos Lopez from the Education Office at ESAC in Madrid, who started the PRESTIGE program and sponsored the research activity on MDO. Besides, a large number of engineers within and outside ESA provided technical support for the engineering modelling in their respective areas of expertise, in particular: Jens Kauffmann (ESA HQ, launchers directorate) and Carlos Corral Van Damme (ESA CDF) for the system design of launchers, Gianluigi Baldesi (ESA TEC-MSS) for structures, Marco De Rosa and Francesco Di Matteo (ESA TEC-MPC)) for propulsion, Marco Chiappone (ESA TEC-QQD) for risk assessment, Michel Van Pelt (ESA TEC-SYC) for cost analysis, Cem Ozam and Patrick Rambaud from the Von Karman Institute of Fluid Dynamics for aerothermodynamics, Massimiliano Bottacini (ESA HSF-ETS) and Fabio Caramelli (ESA HSF-PE) for system design of re-entry vehicles. Special thanks go to Paolo Martino, a MSc student from Politecnico di Milano who was responsible for the development of cost and reliability models of the MDO environment, as well as to Chris Laurel from the US, for his great help in all software-related issues. Finally, I would like to thank the WORHP and NOMAD teams for the use of their optimization libraries, especially Tim Nikolayzik and Dennis Wassel for their advices in the set-up of WORHP, as well as Prof. Kalyanmoy Deb and Abel Garcia Najera for providing the source code of the NSGA-II and MOACOr algorithms.

Passando all'italiano, due parole di ringraziamenti personali scritte di fretta prima della consegna non possono neanche lontanamente rendere l'idea dell'importanza che hanno avuto alcune persone in questi anni. Quindi, a parte ringraziare i miei genitori che hanno sopportato la mia assenza da casa fin dall'inizio dell'università (e la sopporteranno ancora), non aggiungo altro... per chi è stato ed è presente nella mia vita, un unico grande grazie.

You do not inherit the Earth from your ancestors: you borrow it from your children.

SUMMARY

Since the dawn of the space age, development, production and operations of launch vehicles and other Space Transportation Systems (STS) have required huge investments. After the Moon landings, drastically reduced budgets forced engineers to start including in the design cost and programmatic aspects. In the current international context, which shows several attempts to reduce the cost for the access to space, the quality of the design process in general and of the initial design decisions in particular assume a key role, being the major drivers of the total life cycle cost (LCC) of launch systems. Specifically, it has been shown that about 80% of the LCC is determined in conceptual design, while optimization efforts in later design phases only result in minor improvements. For these reasons, Multidisciplinary Design Optimization (MDO) was chosen as the main topic of this doctoral research, in light of its potential benefits on both the efficiency of the design process (i.e. required time and effort) and the quality of the design solutions. In the European scenario, ESA's Future Launchers Preparatory Programme (FLPP) and other national projects are paving the way - through technology demonstrators and system studies - for the transition in the 2025 timeframe from the current fleet of launchers (Ariane 5, Soyuz and Vega) to a more flexible and cost effective, modular family of Expendable Launch Vehicles (ELV). Hence, through collaboration with ESA, the primary application of the present research was identified in ELVs, although the developed environment was kept as flexible as possible to allow for easy extendibility to other STSs, such as re-entry vehicles, crewed and reusable systems.

The main objective of the research was therefore identified in the development and quantitative assessment of a MDO environment capable of tackling the early design phases of ELVs. The focus was primarily on the engineering modelling aspects, whereas the necessary global and local optimization infrastructure was only considered as a mathematical tool enabling MDO, in light of the criticality of the models definition for the applicability of MDO methodology to real-world design. In fact, in spite of the rather impressive list of achievements promised by MDO and of a growing interest in the research community, successful industrial applications are still extremely rare. This is in part due to a certain resistance of design offices, motivated by the large required initial investments, to the introduction of MDO against the traditional fixed-point iterations process. But the largest obstacle is certainly constituted by the fact that MDO's industrial applicability is subject to the assessment of the confidence which can be placed in the achieved design solutions. In particular, the definition and calibration of the multidisciplinary models are particularly challenging, the main obstacle being the identification of a reasonable compromise between simplicity and accuracy.

With this problem in mind, a quantitative assessment of the engineering models was therefore carried out, including a detailed analysis of the accuracy of all developed analysis codes, both in terms of disciplinary errors and of system level sensitivities and results. Although a rather significant number of developments documented in recent years have been focused on the application of MDO to the design of different types of STS, the evaluation of accuracy and reliability of such models to the extent described in the present research represents an innovative and necessary endeavour, to the author's knowledge.

The main question proposed for the research - whether it is possible to develop relatively simple models permitting fast design cycles while still ensuring sufficient accuracy to place confidence in the achieved design solutions – does not hold a straightforward answer. The computational effort required with the implemented models surely matches the original target (i.e. <2 s for a complete multidiscipinary analysis), but it is not nearly as easy to measure the accuracy requirement. It can however be said that the decided two-steps development process proved to be of key importance for the incremental improvement of the models, which ensured a sensible enhancement in accuracy at a manageable price in computational effort from Version 1 to Version 2. As a result, the final models are characterized by accuracies on the global performance which should be sufficient in most of the cases to fairly compare two significantly different design solutions through MDO.

Throughout lengthy details on the modelling and validation as well as on few relevant applicative cases, this PhD research highlights how the development and tuning of a reliable MDO environments is a very complex task, requiring large efforts in most engineering areas, besides computer science and mathematics. However, reasonable accuracies and physically sound design modifications can be obtained through the MDO approach, even when exploiting only fast engineering level models to avoid resorting to high performance computing. With today's computational resources, which even allow conceiving the introduction of high fidelity information in the design cycle, MDO guided by human expertise is therefore a powerful approach for the initial design phases of launchers and other STS. Although the huge initial investment in terms of development and personnel training is a major obstacle to widespread industrial application, it is the opinion of the author that the resulting benefit in terms of design quality is well worth the effort, with the potential of contributing to the long term goal of achieving low cost access to space.

CONTENTS

INDEX OF FIGURES

INDEX OF TABLES

ACRONYMS

AAO	All-At-Once
ACO	Ant Colony Optimization
ADD	Architectural Design Document
AoA	Angle of Attack
ARC	Ames Research Centre
ASTOS	Aerospace Trajectory Optimization Software
BB	Building Block
BBO	Black-Box Optimization
BLISS	Bi-Level Integrated System Synthesis
CM	Cavitation Margin
CO	Collaborative Optimization
CSSO	Concurrent SubSpace Optimization
BS	Boosters Set
BTL	Bilinear Tangent Law
BWB	Blended Wing Body
CB	Circularization Burn
CBS	Costs Breakdown Structure
CCB	Common Core Boosters
CEA	Chemical Equilibrium and Applications
CER	Cost Estimation Relationship
CDF	Concurrent Design Facility
CFD	Computational Fluid Dynamics
CFRP	Carbon Fibre Reinforced Plastic
CoG	Centre of Gravity
CoP	Centre of Pressure
CpL	Cost per Launch
CPU	Central Processing Unit
DAS	Debris Assessment Software
DG-MOPSO	Double Grid Multi-Objective Particle Swarm Optimization
DOE	Design of Experiments
DSF	Design Safety Factor
DSM	Design Structure Matrix
EA	Engines Assembly
EC	Expander Cycle
ELV	Expendable Launch Vehicle
EoL	End of Life
EoM	Equations of Motion
EPS	Electrical and Power System
ESA	European Space Agency
ESOC	European Space Operations Centre
ESTEC	European Space Research and Technology Centre
FAA-AST	Federal Aviation Administration's Office of Commercial Space Transportation
FAC	Finite Area Combustor
FEM	Finite Elements Modelling
FF	Filling Factor
FPI	Fixed Point Iterations
FoS	Factor of Safety
FY	Fiscal Year
GA	Genetic Algorithm
GG	Gas Generator
GEO	Geostationary Earth Orbit
GO	Global Optimization
GPWv3	Gridded Population of the World, version 3

GRC	Glenn Research Centre
GTO	Geostationary Transfer Orbit
GTOW	Gross Take-Off Weight
GNC	Guidance, Navigation and Control
GUI	Graphical User Interface
HD	Hierarchic Decomposition
HEDM	High Energy Density Material
HLLV	Heavy Lift Launch Vehicle
HPC	High Performance Computing
HSRP	High Speed Research Program
HTHL	Horizontal Take-Off Horizontal Landing
HTVL	Horizontal Take-Off Vertical Landing
HW	Hardware
IAC	Infinite Area Combustor
IADC	Inter-Agency Space Debris Coordination Committee
IDF	Individual Disciplines Feasible
IS	Inter Stage
IT	Inter Tank
LaRC	Langley Research Centre
LaWGS	Langley Wireframe Geometry Standard
LB	Lower Bound
LCC	Life Cycle Cost
LCE	Low Cost Engine
LEO	Low Earth Orbit
LH2	Liquid Hydrogen
LO	Local Optimization
LOx	Liquid Oxygen
LP	Liquid Propellant
LPM	Launcher Processing Mode
LRE	Liquid Rocket Engines
LS	Lower Stage
LSC	Lower Stage Coast
LTO	Lunar Transfer Orbit
MDA	Multidisciplinary Design Analysis
MDAO	Multidisciplinary Design Analysis and Optimization
MDF	Multi-Disciplinary Feasible
MDO	Multidisciplinary Design Optimization
MEO	Medium Earth Orbit
MEOP	Maximum Expected Operating Pressure
MMH	Mono-Metil Hydrazine
MOACOr	Multi-Objective Ant Colony Optimization for continuous problems
MOLD	Multilevel Optimization by Linear Decomposition
MOPSO	Multi-Objective Particle Swarm Optimization
MSFC	Marshall Space Flight Centre
MSP	Mission Success Probability
NASA SP	NASA Scientific Publication
NASA TM	NASA Technical Memorandum
NGL	Next Generation Launcher
NHD	Non Hierarchic Decomposition
NLP	Non Linear Programming
NOL	Nested Optimization Loop
NPSH	Net Pump Suction Head
NSGA-II	Non-Dominated Sort Genetic Algorithm II
OGT	Optimized Gravity Turn
OTI	Optimized Target Inclination
OTS	Off-The-Shelf
PAES	Pareto-Archived Evolutionary Strategy
PF	Pressure Fed
PHR	Pump Head Rise
PL	PayLoad
PLA	PayLoad Adapter
PLF	PayLoad Fairing
PLSF	PayLoad Scaling Factor
PO	Pitch Over

PSO	Particle Swarm Optimization
RAAN	Right Ascension of the Ascending Node
RBCC	Rocket Based Combined Cycle
RBS	Risks Breakdown Structure
REV	Re-Entry Vehicle
RID	Review Item Discrepancy
RK	Runge-Kutta
RKF	Runge-Kutta-Fehlberg
RSC	Re Start Capability
RV	Redundancy Level
RLV	Reusable Launch Vehicle
RP1	Rocket Propellant 1
RPSH	Required Pump Suction Head
RSM	Response Surface Methodology
RSRM	Reusable Solid Rocket Motor
SA	Simulated Annealing / Sensitivity Analysis
SBT	Single Boosters Type
SC	Staged Combustion
SET	Single Engine Type
SF	Sliver Fraction
SLO	System Level Optimization
SLP	Sequential Linear Programming
SLV	Space Launch Vehicle
SM	Structural Material
SSM	Structural Safety Margin
SSO	Sun Syncronous Orbit
SP	Solid Propellant
SQP	Sequential Quadratic Programming
SRD	Software Requirements Document
SRM	Solid Rocket Motor
SS	SubSystem
SSDL	Space Systems Design Laboratory
SSME	Space Shuttle Main Engine
SSTO	Single-Stage-To-Orbit
STS	Space Transportation Systems
SW	Software
TA	Tanks Arrangement
TCS	Thermal Control System
ThC	Throttle Capability
TIL	Target Inclination Law
TOE	Thrust Oversized Engine
TPA	Turbo-Pump Assembly
TPS	Thermal Protection System
TRF	Technology Reduction Factors
TRL	Technology Readiness Level
TSTO	Two-Stage-To-Orbit
TT	Tanks Type
TVC	Thrust Vector Control
UB	Upper Bound
UDMH	Unsymmetrical Di-Methyl Hydrazine
US	Upper Stage
USC	Upper Stage Coast
VEB	Vehicle Equipment Bay
VTHL	Vertical Take-Off Horizontal Landing
VTVL	Vertical Take-Off Vertical Landing
VVD	Verification and Validation Document
WAATS	Weight Analysis of Advanced Transportation Systems
WBS	Weights Breakdown Structure
WER	Weights Estimation Relationship
WF	Web Fraction
WORHP	We Optimize Really Huge Problems

LAUNCHER COMPONENTS

The names and acronyms reported hereafter are commonly used throughout the whole dissertation, coming from European industry's names of specific components of Ariane 5 and VEGA launch vehicles, as well as very common NASA terminology for Space Shuttle and Saturn V.

EAP / P241	Ariane 5 solid propellant boosters
EPC	Ariane 5 cryogenic main stage
EPS	Ariane 5 storable upper stage
ESC-A	Ariane 5 cryogenic upper stage for ECA version, HM-7B engine
ESC-B	Ariane 5 cryogenic upper stage for ECB version (or Ariane 5 ME), Vinci engine
VEB	Ariane 5 Vehicle Equipment Bay
SYLDA-5	Ariane 5 dual launch adapter structures
Cone 3936	Ariane 5 interface between VEB upper section and the standard payload adapter systems
JAVE	Interstage and boosters support structure between EPC and upper stage
BME	EPC's thrust structure for Vulcain engine
SSHeL	EPC's He pressurization system
JAR	EAP's aft aerodynamic and pad interface skirt
JAV	EAP's front aerodynamic skirt
DAAR	EAP's aft attachment device
DAAV	EAP's front attachment device
SCA	VEB's attitude control system
P80	VEGA P80 first stage motor
Z23	VEGA Zefiro-23 second stage motor
Z9	VEGA Zefiro-9 third stage motor
AVUM	VEGA liquid upper stage
IS01	P80 aft aerodynamic and pad interface skirt
IS12	P80-Z23 interstage
IS23	Z23-Z9 interstage
IS3A	Z9-AVUM interstage
LPS	AVUM's main liquid propulsion system
RACS	AVUM's roll and attitude control system
STS	Space Transportation System (Space Shuttle)
ET	External Tank
RSRM	Reusable Solid Rocket Motor
SSO	Space Shuttle Orbiter
S-IC	Saturn V LO_x-Kerosene first stage
S-II	Saturn V LO_x-LH_2 second stage
S-IVB	Saturn V LO_x-LH_2 third stage (Earth Departure Stage - EDS)

SYMBOLS

The symbols reported hereafter are commonly used throughout the whole dissertation, often in combination with subscripts and/or superscripts. Additional symbols may be defined in specific sections.

Greek alphabet:

α_P	Propellants mixture ratio
α_{tot}	Total angle of attack
α	Angle of attack
β	Angle of sideslip
γ	Flight-path angle
χ	Heading angle
λ	Longitude
δ	Declination
δ_{div}	Engine's divergent section half-cone angle
δ_{th}	Engine's throttle level
δ_{tvc}	Engine's maximum TVC angle
$\delta_{tvc,\theta}$	Engine's TVC angle in pitch plane
$\delta_{tvc,\psi}$	Engine's TVC angle in yaw plane
ε	Nozzle's expansion ratio
μ	Mean value
σ	Standard deviation
θ	Pitch
ψ	Yaw
φ	Roll
η	Efficiency
ζ	Bilinear tangent law parameter
τ	Torque
ω	Longitude of pericentre
Ω	Right Ascension of the Ascending Node (RAAN)

English alphabet:

A	Area
C_A	Axial force coefficient
C_L	Lift coefficient
C_D	Drag coefficient
C_N	Lateral force coefficient
C_M	Lateral moment coefficient
C_T	Nozzle's thrust coefficient
C_*	Combustion chamber's characteristic velocity
D	Diameter
H	Altitude
H_{clear}	Altitude of clearance of the launch tower
H_{atmo}	Altitude of atmospheric interface for the ascent models
H_a	Altitude of apocentre
H_p	Altitude of pericentre
I	Moment of inertia
I_{sp}	Specific impulse
L	Length
M	Mach number, Mass, Bending moment external load (depending on the context)
N_x	Longitudinal (axial) running load
$N_{x,c}$	Longitudinal (axial) running load in compression
$N_{x,t}$	Longitudinal (axial) running load in tension
N_{xy}	Transverse (shear) running load
N_y	Circumferential (hoop) running load

N_{bsets}	Number of booster sets
$N_{bpset,j}$	Number of boosters per each set
$N_{boosters}$	Total number of boosters
N_{stages}	Number of stages
N_{CDV}	Number of component (i.e. stage or boosters set) level design variables
N_{CDC}	Number of component (i.e. stage or boosters set) level design constraints
N_{SDV}	Number of system level design variables
N_{SDC}	Number of system level design constraints
N_{TDV}	Number of trajectory variables
N_{TDC}	Number of trajectory constraints
n_{ax}	Axial acceleration
n_{lat}	Lateral acceleration
P	Axial external load
p	Pressure
p_{cc}	Combustion chamber pressure
p_{tanks}	Liquid propellant tanks pressure
p_{press}	Pressurizer tanks pressure
q_{dyn}	Dynamic pressure
q_{heat}	Heat flux per unit surface
$q_{heat,max}$	Maximum heat flux per unit surface with payload fairing attached
$q_{heat,max,PL}$	Maximum heat flux per unit surface on the payload, after payload fairing jettison
q_{load}	Heat load (heat flux integrated over time) per unit surface
r_b	SP burning rate
t	Time
S	Surface
T	Thrust, Lateral external load (depending on the context)
V	Velocity, Volume (depending on the context)
X_{CoG}	Centre of Gravity
X_{CoP}	Centre of Pressure
X_{CDV}	Vector of component (i.e. stage or boosters set) level design variables
Y_{CDC}	Vector of component (i.e. stage or boosters set) level design constraints
X_{SDV}	Vector of system level design variables
Y_{SDC}	Vector of system level design constraints
X_{TDV}	Vector of trajectory variables
Y_{TDC}	Vector of trajectory constraints
W	Weight

Superscripts:

*	Conditions at the optimal nozzle altitude (i.e. external pressure equal to exhaust pressure)

Subscripts:

a	Ambient conditions
cc	Conditions in the combustion chamber
e	Conditions at the nozzle's exhaust
t	Conditions at the nozzle's throat
BS_j	Parameter referred to j-th set of boosters
LS_j	Parameter referred to j-th lower stage
PLF	Parameter referred to the payload fairing
PLA	Parameter referred to Payload Adapter
US	Parameter referred to the upper stage
core	Parameter referred to the whole core of the launcher (i.e. all stages + payload fairing)
eng	Parameter referred to one engine in a given engines assembly (i.e. corresponding the whole assembly for single engines configurations)
sea	Sea-level conditions
vac	Vacuum conditions
ABA	Parameter referred to the aft boosters attachment
AES	Parameter referred to the avionics and electric systems
ASk	Parameter referred to the aft skirt
BN	Parameter referred to the booster's nose ogive
EA	Parameter referred to the whole engines assembly

FBA	Parameter referred to the forward boosters attachment
FSk	Parameter referred to the forward skirt
F	Parameter referred to the fuel
FT	Parameter referred to the fuel's tank
IS	Parameter referred to the interstage
IT	Parameter referred to the intertank
Ox	Parameter referred to the oxidizer
OxT	Parameter referred to the oxidizer's tank
N	Parameter referred to a stage's nose section (conical or cylindrical)
PI	Parameter referred to the pad interface structure
PS	Parameter referred to the pressurization system
Sp	Parameter referred to the solid propellant
SpC	Parameter referred to the solid propellant case
TF	Parameter referred to the thrust frame structure
TI	Parameter referred to the thermal insulation
UP	Parameter referred to the unused propellants

So many people live within unhappy circumstances and yet will not take the initiative to change their situation because they are conditioned to a life of security, conformity, and conservatism, all of which may appear to give one peace of mind, but in reality nothing is more dangerous to the adventurous spirit within a man than a secure future. The very basic core of a man's living spirit is his passion for adventure. The joy of life comes from our encounters with new experiences, and hence there is no greater joy than to have an endlessly changing horizon, for each day to have a new and different sun.

CHAPTER 1

INTRODUCTION

1.1. Research motivation

Since the dawn of the space era, the development, production and operations of Space Launch Vehicles (SLV) and Space Transportation Systems (STS) in general have been a very costly business. During the Cold War, both the United States and the Soviet Union diverted a significant fraction of their entire industrial resources to the respective space programs, resulting in basically no financial constraint being imposed on the engineers' decisions. To give an idea of this "engineering freedom", the peak year funding for the Apollo program reached the 0.4% of the US Gross Domestic Product (GDP) according to a recent Congressional Research (**[1]**). This situation rapidly changed after the Moon landings, with all national space agencies having to cope with budgets of a different order of magnitude. As a consequence, engineers started to include cost and programmatic aspects in the design process. During the past decades, monetary concerns in worldwide space programs have continued to rise - with the only exception of the emerging Chinese effort - only partially mitigated by the new climate of cooperation between nations which allows sharing of the investments. For the foreseeable future, the financial situation is not anticipated to change. In fact, no remarkable break-through in the exploration of space is to be expected worldwide unless a drastic reduction of the cost for the access to space can be achieved.

Different approaches to SLV design were explored in recent years for this purpose, such as complexity reduction strategies (e.g. use of solid propellants for VEGA) or the exploitation of flight heritages and modular architectures (e.g. Delta and Atlas families of Evolved Expendable Launch Vehicles). These however led to very limited improvements in terms of launch costs. On the contrary, NASA's commercial cargo and crew initiatives, started in 2006 and now at a crucial moment, seem to be able to change the conventional paradigm for doing business in the space industry. For example, SpaceX is cutting costs[1] through a flat management structure and a clean sheet design driven by cost and reliability, with simplicity, modularity and reliability as the core design philosophies.

On a smaller scale, Europe is starting to prepare for an important transition in STS. The current fleet of launch vehicles has just reached its full capacity, with the recent successes in the maiden flights of VEGA and Soyuz from Kourou. These new launch systems complement the workhorse Ariane 5 in its two presently operated versions, ES and ECA, covering the whole market range from heavy GTO commercial payloads and ISS cargo supply spacecrafts, to navigation and scientific satellites. In the meantime, the Future Launchers Preparatory Program (FLPP) is aimed at paving the way, through both technology developments and system studies, for a successor to the present European systems in the 2025 timeframe. The Next Generation Launcher (NGL) family should be capable of reducing specific launch costs by half with respect to Ariane 5 and of largely increasing its flexibility through a modular design approach. In the shorter period, upgrades of both Ariane 5 and VEGA are being considered. Detailed design and advanced testing are available for Ariane 5 ME (Midlife Evolution) featuring a new ESC-B upper stage with the restartable expander-cycle Vinci engine, pending political decisions to move on to the implementation phase, whereas different VEGA enhancements have been investigated for quite some time now, though still on paper.

In this variegated context, it is apparent that the design process in general, and early design decisions in particular, are critical in the road towards low cost access to space. Hence, Multidisciplinary Design Optimization (MDO) was chosen as

1 Falcon 9 Heavy, scheduled for a maiden flight in 2013, is advertised as capable of beating the 1000 $/kg to LEO threshold.

the main subject of this PhD research in light of the potential benefits that this new approach could lead to the design process. In light of the European background and the collaboration with the European Space Agency (ESA)[2], classical unmanned Expendable Launch Vehicles (ELV) were defined as the applicative scenario to be most thoroughly investigated. Other classes of STS, such as Re-Entry Vehicles (REV), Reusable Launch Vehicles (RLV) and manned systems were also studied and represent natural extensions of the work presented in this dissertation. Engineering models were nevertheless not developed, due to the limited relevance in the European scenario and the desire to avoid excessive complications.

Multidisciplinary Design Optimization (MDO) is intended as the coupling together of two or more analysis disciplines with numerical optimization methods. It was defined as *"a methodology for the design of complex engineering systems and subsystems that coherently exploits the synergy of mutually interacting phenomena"* by NASA Langley Research Centre's Multidisciplinary Design Optimization Branch, or in more simple words by AIAA's MDO Technical Committee as *"how to decide what to change, and to what extent to change it, when everything influences everything else"*.

MDO has been increasingly studied since the 80's in various fields of engineering, with the main purpose of improving the design process for complex systems and thus ultimately achieving enhancements in performance, costs, reliability, or any other design objective. As already mentioned, the motivation at the basis of the present research is represented by the desire to improve the design of ELVs. The next two paragraphs attempt to justify the predicated advantages of MDO in this area, from both mathematical and financial standpoints.

1.1.1. Optimality advantage of Multidisciplinary Design Optimization

Traditional engineering design is performed sequentially and cyclically by experts of the various disciplines, who optimize the design of each subsystem to satisfy domain specific constraints and move towards domain specific objectives. Several Fixed Point Iterations (FPI) of this process are usually required to reach a design point where all disciplines agree on the common variables and satisfactory values of the global objectives are reached, thus requiring a large amount of time.

Moreover, the resulting design is likely to be sub-optimal, because the sequential approach may lead to the loss of good solutions, which would be identified in a concurrent optimization process. A trivial but enlightening example of how this enhancement in the design objective (performance, cost, or any other driving criterion) is achieved is presented in reference **[2]**. Consider a hypothetical aircraft with simplified design model reduced to the contours plot of Figure 1**a**: the generic performance measure $\boldsymbol{P}$ and two constraints $\boldsymbol{C_1}$ and $\boldsymbol{C_2}$ are represented as a function of the wing aspect ratio $\boldsymbol{AR}$ and the wing structural minimum weight $\boldsymbol{W_{min}}$, condensing the aerodynamic and structural aspects of the design. Clearly, $\boldsymbol{O_1}$ is the constrained maximum of $\boldsymbol{P}$, but if an additional constraint $\boldsymbol{C_3}$ is found to be critical in a later stage of the project as in Figure 1**b**, a new optimum is established at $\boldsymbol{O_3}$. However, in a sequential design process, one of the variables may have already been frozen so that only one design freedom is left to deal with the new constraint. If for example the $\boldsymbol{AR}$ is frozen, only $\boldsymbol{W_{min}}$ can be adjusted, resulting in a sub-optimal design solution at $\boldsymbol{O_2}$.

MDO approach does not present this limitation, being able to concurrently vary all modelled design variables at the same time to achieve the global optimum. In general, a physical behaviour resulting in a loss such as that between $\boldsymbol{O_2}$ and $\boldsymbol{O_3}$ is typical for most engineering applications. Although for some applications this may not be an issue, for complex aerospace systems it may represent a critical difference, especially if the model include considerations about costs, reliability, programmatics and other non performance related aspect.

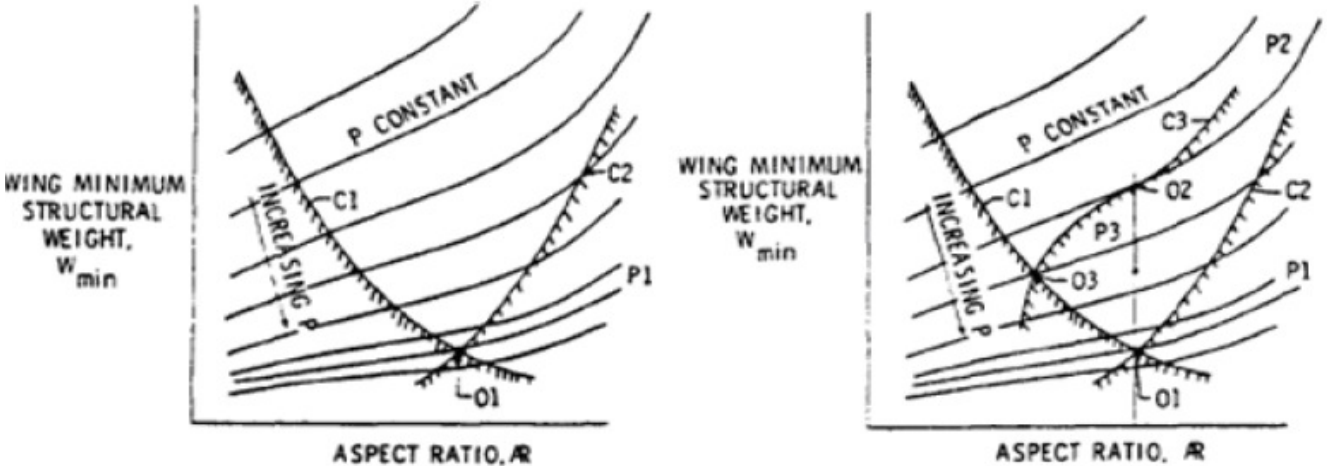


Figure 1: Sub-optimality of solutions obtained with sequential optimization approach in aircraft design example, as taken from [2]. Figure 1a (left): 2-constraints scenario. Figure 1b (right): 3-constraints scenario.

[2] See **Section 1.3** for an overview of the research framework.

1.1.2. Financial advantages of Multidisciplinary Design Optimization

If successfully applied in the design process of complex engineering systems, MDO allows for significant monetary advantages, both for governmental institutions and private industries. This gain comes in two ways: 1) from the improvements in the design results shown in the previous paragraph; and 2) from the increase of the design process efficiency, which determines both schedule savings and lower man-hours required from disciplinary experts.

As regards to the first point, Figure 2 shows the percentage of the Life Cycle Cost (LCC) determination of a SLV as a function of the design progress in the classical phases of conceptual, preliminary and detailed design, up to manufacturing and integration/verification process, as defined by NASA in **[3]**. Similarly, Figure 3 reports the cost overruns as a function of the system engineering effort for past NASA vehicle development programs. It is clear that most of the LCC (80%) is determined early in the conceptual design phase, whereas design decisions performed later only marginally affect it. Moreover, a larger effort in the initial system engineering phase usually turns out in a great cost overrun risk reduction.

In this perspective, MDO techniques drastically reducing the time required for conceptual studies and hence increasing the number of analyzed design alternatives, have the potential to identify new and more cost-effective concepts that would not be envisioned with a classical sequential approach. The human intellect cannot of course be taken out of the loop, and is in fact fundamental in the set-up of both the models and the integration/optimization environment, in guiding the algorithms towards better solutions and in the critical analysis of the results. Nevertheless, MDO tools can be extremely useful in assisting the designers, especially at the conceptual and early preliminary stages.

As regards to the second point instead, the financial advantages of the exploitation of MDO in an industrial context were analyzed in **[4]** by Phoenix Integration[3], following the main effects on the financial health of an aerospace corporation:

- MDO is shown to bring an efficiency enhancement in the execution of a design iteration, leading to time savings in the order of the "hours-to-minutes", hence reducing the human effort (and related costs) in the process.
- Even when compared with concurrent design laboratories, such as ESA's Concurrent Design Facility (CDF), the MDO approach is expected to lead significant improvements in the time, cost and outcome of conceptual studies. This increases the chances of developing winning proposals. Through an economical analysis of 25 aerospace organizations, this aspect is shown to have a direct correlation with the stock price trend, which is positively influenced by key defence or government program wins.
- MDO finally may benefit existing programs as well, not being limited to conceptual design, since it can contribute to the improvement of the resources allocation, limiting the total costs.

In conclusion, the technical and economic considerations presented in this section allow justifying the increasing worldwide interest in MDO, and represent the motivational basis for the research work presented in this dissertation.

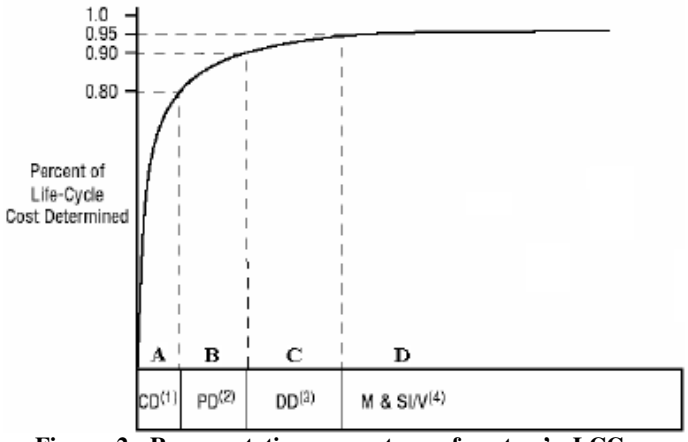


Figure 2: Representative percentage of system's LCC as a function of the design phase, as taken from [3]

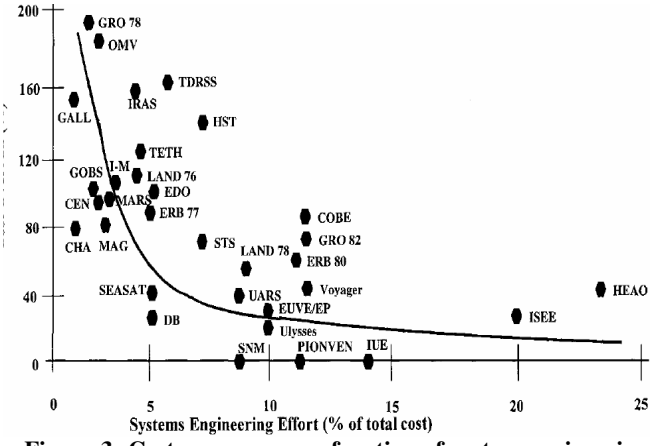


Figure 3: Cost overruns as a function of system engineering effort for past NASA programs, as taken from [3]

[3] Phoenix Integration is a company developing process integration and design optimization software, especially Phoenix ModelCenter is probably the first example of commercial MDO software (www.phoenix-int.com).

1.2. Research objectives, challenges and original contributions

The research work presented here is centred on the development and validation of an MDO environment for the early design phases of expendable launch vehicles, with the main objective of quantitatively assessing its suitability for real-world applicability in terms of overall accuracy and computational requirements. The focus is primarily on the engineering modelling aspects, whereas the necessary global and local optimization infrastructure is only considered as a mathematical tool enabling MDO, in light of the criticality of the multidisciplinary models definition for the applicability of MDO methodology to industrial design.

This objective was justified in the previous section, in terms of relevance of ELVs to the present European scenario as well as of benefits that the MDO approach could bring to the cost reduction for the access to space. MDO models incorporate all relevant disciplines simultaneously, achieving optimal solutions superior to those found by sequential design if the same fidelity level is employed. The availability of a reliable MDO environment as a support to the designers' experience ensures a more rapid exploration of the design space, leading to Pareto optimal fronts under different aspects, such as mass, cost, reliability, or mission flexibility. Designers can then select the most promising concepts to be used as starting points for refinements with more traditional design methodologies. This should in theory ensure the selection of the design option granting for example the lowest possible LCC, if this is the overall project's driver.

To date, in spite of the rather impressive list of achievements promised by MDO and of a growing interest in the research community, successful industrial applications are still extremely rare. This is in part due to a certain resistance of design offices to the introduction of MDO versus the use of traditional FPI process. This has in fact always represented the standard aerospace industry practice, most of the tools and procedures having been developed for it. The introduction of MDO would therefore require a large initial investment in software development or purchase and in personnel training, which must be offset by sensible global objective improvements or project time savings to make it worthwhile.

In addition, several undeniable obstacles contribute to make the development and integration of interdisciplinary design environments a particularly challenging task. In particular, the main issues can be grouped in the following categories:

- ***Modelling***:
  - compromise between fidelity and computational effort in the traditional disciplinary analyses (e.g. CFD/FEM versus simplified engineering models);
  - development of well-posed and clear interfaces among disciplines;
  - capture (at least partial) of human expertise in the models;
  - development of common parametric geometric models for the different disciplines (configuration, aerodynamics, structural analysis, thermal analysis, …);
  - mathematical modelling of non-technical disciplines (cost estimation, reliability assessment, schedule and programmatic aspects evaluation, …);
  - efficient and consistent multi-disciplinary problem formulation definition;
  - validation of the developed models with respect to real-world systems.
- ***Hardware***: need for large computational resources, most likely requiring High Performance Computing (HPC) techniques such as distributed parallel processing or GPU computing.
- ***Software***:
  - procurement and integration of proprietary, legacy, open source and commercial tools, often developed in different languages and epochs, as well as with completely different coding philosophies;
  - configuration control as well as management, storing and communication of large amounts of data;
  - need for two-ways, intuitive and interactive machine-human interface, to ease the introduction in the optimization loop of the designers' expertise
- ***Mathematical analysis and optimization***:
  - selection or implementation of efficient global and local optimization strategies, both required for search space pruning and successive solutions refinement;
  - selection of a suitable optimization architecture, either single or multi level depending on the problem characteristics;
  - definition of surrogate models if computational times for the involved disciplines are too large for allowing the full model to be included in the optimization loop.

Among the above issues, one particular aspect can be identified as the most critical: the trade-off between the engineering models' simplicity and accuracy. In fact, it is still not clear in the MDO community how to obtain reliable design information in the early design phases when accepting the compromises in design fidelity that are necessary to limit the computational efforts. This is an extremely challenging hurdle to be tackled: computational times can quickly become unmanageable when complete design freedom needs to be preserved to allow for large scale search space exploration, but

the physics of the problem must be sufficiently represented in order to validate the obtained design solutions and place confidence in the performed trade-offs.

With this respect, both CFD and FEM approaches to aerodynamic and structural analyses are not well suited to MDO deployment, unless the resources of modern large computer clusters are invested. Examples of recent applications are NASA/Boeing's Blended-Wing-Body (BWB) (**[5]**) research aircraft and the European Union's HISAC project (**[6]**) for an environmentally friendly supersonic transport. These made extensive use of the MDO approach in early design, integrating parallel CFD and FEM codes into complete aircraft design loops. The final configurations selected in both projects were strongly affected by the MDO results. Although HISAC is still only on paper, the BWB reached the level of flying prototype, validating the design process employed in the conceptual and preliminary stages.

Due to limitations in the available computational resources and to the excessive technical complexities, the exploitation of high fidelity analyses was not considered for the present research, focusing instead on "traditional" engineering level models. The intention was therefore to study the complex relations among disciplines and identify the main drivers for the global accuracy in Multi-disciplinary Design Analysis (MDA) processes. A MDA can be defined as a set of analysis modules for several disciplines, to be executed in a given order and possibly requiring iterations of two or more modules, apt to exhaustively determine the design of a complex engineering system. *Independent variables*, also called *design variables*, need to be provided as inputs to the MDA, either through human expertise or an automated optimization algorithm, whereas the outputs of each discipline are typically used as *optimization constraints* or *objectives*. Finally, *coupling variables* are the parameters which need to be exchanged among the different disciplines, hence determining the sequence of execution and the necessity of iterative cycles.

In this context, the main question which will be attempted to be answered can be formulated as:

> *"Is it possible to reduce the complexity of the design models for ELVs to the point that a full MDA can be executed within seconds, without losing the accuracy necessary to place confidence on the achieved design solutions and trade-offs (i.e. around 10%)?"*

The availability of a MDA process requiring only seconds to execute, in contrast with the minutes or hours typical of CFD/FEM, would enable the exploration of ELVs arbitrary configurations in MDO loops within reasonable computational times without the need for a HPC infrastructure. In order to answer the above question, the engineering models were developed in two steps. First, a *conceptual level* modelling environment was developed, implemented and tested. A thorough validation procedure and critical analysis of the results, together with an independent review from ESA, highlighted the key weaknesses of these models. A wide range of upgrades spanning all disciplines was identified, which allows for higher fidelity and larger functionality at a reasonable price in terms of computational effort. The enhancements were implemented in a second modelling step, targeted to the *early preliminary* design, with a further validation campaign assessing the improvements in accuracy.

This incremental strategy founded on the critical analysis of the validation results obtained at each step was at the basis of the engineering modelling effort, and resulted in the achievement of good global performance accuracy with limited computational times, as will be presented in the continuation of the dissertation. It has to be remarked how such rigorous procedure for the development and assessment of the MDA actually constitutes the most innovative aspect of the present research work. In fact, a significant number of MDO developments are documented in recent literature, covering the design of different types of aerospace vehicles, but none includes a quantitative evaluation of the engineering methods to the extent described here. In a research field which seems mature for more important applications, the lack of accuracy and reliability evaluations represents a "showstopper" for the industrial applicability, since large initial investments in MDO technology can only be justified in front of a certain confidence in the results which can be achieved. Hence, the information obtained within this research represents a relevant original contribution to the field of MDO, providing useful hints for the adoption of this design approach and for the evaluation of its suitability to different scenarios. In particular, results related to the disciplinary models errors, their system level sensitivities and criticalities, as well as the effects of optimization when applied to different typologies of analysis models hold general validity. Single and multi-objective optimal design solutions presented throughout the validation and application chapters (**5**, **6** and 8) are also filled with significant and rather original lessons learned about the capabilities - or in some cases the difficulties - of MDO in reproducing realistic design considerations.

Finally, the specific analysis models implemented for the design of ELVs are mainly derived from well-known methods, whose development sometimes dates back to the origins of space flight. However, their integration in a unified environment tackling arbitrary ELV configurations and encompassing all the core disciplines represents a rare undertaking. Comprehensive details are provided in **Chapters 4** and **6** for the multi-disciplinary models of the two steps mentioned above, constituting an exhaustive synthesis of launcher engineering "know-how". Among the described disciplinary methods, the following aspects are worth lingering over, in light of the originality of the selected approach:

- Liquid rocket engines' inert mass estimation through Weight Estimation Relationships (WERs), derived from the statistical analysis of a database collected from various sources (**Sections 4.2** and 6.2).

- Time-based reliability estimation, leading to an assessment of the overall mission success probability (**Section 4.7**).
- Analytical methods for solid grain geometry definition and pumps cavitation assessment (**Section** 6.2).
- Safety related analyses, including assessment of the impact ellipses for suborbital stages/boosters and calculation of the related density of population (**Section 6.4**).
- Analytical structural sizing model, including multiple-beams approximations for core and boosters (**Section 6.5**).

1.3. Research framework and SVAGO software

The research work described in this dissertation was carried out by the author in the 2009-2011 timeframe at Politecnico di Milano, in collaboration with the Optimal Control Group of the Centre for Industrial Mathematics of the University of Bremen. Annalisa Riccardi from the University of Bremen participated in the research project working on several mathematical aspects - especially local optimization - and on the graphical user interface. The combination of engineering and mathematical skills proved to be a winning scenario, allowing for different points of view and capabilities to be merged.

The framework that allowed such a fruitful relationship was ESA's PRESTIGE PhD program. The PRrogram in Education for Space, Technology, Innovation and knowledGE (PRESTIGE) was set up by ESA's Education Office as a formal collaboration between the Agency and one or more universities on topics of great importance for future space missions. As taken from PRESTIGE announcement website[4] *"… ESA has identified the need to reinforce education and research in particular technical domains, seen as the critical areas of knowledge for future space programmes. In this context, the ESA Education PRESTIGE programme intends to foster that knowledge by granting research and training opportunities to European universities linked to these technical domains. The objective of this initiative is twofold. To increase the competence of European universities in areas identified as critical for the space domain, through a close research collaboration with ESA. And to provide opportunities to European students in their final year of a university Master or PhD course for a research internship at ESA in the identified fields, thus facilitating the transfer of knowledge and interaction between ESA and the universities."*

For the research topic on Multidisciplinary Design Optimization, PRESTIGE grants were given to Politecnico di Milano and Universität Bremen, foreseeing research activities respectively on the engineering and mathematical aspects of the launch vehicles design optimization problem. This dissertation deals almost exclusively with the engineering aspects, which were directly investigated by the author. However, a divagation on mathematical optimization is included in the state-of-the-art review (**Section 2.3**) and a short chapter is dedicated to the optimization architecture and algorithms implemented or integrated for the MDO environment (**Chapter 3**).

Several advantages of the close relationship with ESA, especially TEC-ECM section, arose in the course of the PhD. In particular, the possibility of interacting with ESA's disciplinary experts during the modelling and validation phase proved to be extremely useful, as well as the two formal ESA reviews conducted at the end of each step. Moreover, validation data in excess of those freely available on the internet were provided, even though the delicate nature of launch vehicle data still prevented from obtaining reliable information in several critical areas.

The final deliverables for PRESTIGE research consist in a large amount of technical documentation, as well as in the software tool SVAGO, which will be used by ESA in the frame of concurrent design and industrial design evaluation in the early phases of launch systems development (see **Paragraph 1.3**). Due to length constraints, this dissertation does not report the full content of the PRESTIGE technical documentation presented to ESA. The attached cd-rom however contains all relevant documents, which will be referenced as **[A1]**, **[A2]**, ..., for "Applicable Document" and are listed at the end of the dissertation besides the bibliography.

A software program written in C++, named SVAGO for Space Vehicles Analysis and Global Optimization, was the result of the practical implementation carried out within PRESTIGE research, including both engineering models and optimization algorithms in a generic MDO environment with specialized disciplinary modules for ELVs design. The basic philosophy at its core can be summarized in four points:

- ***Extreme user interactivity***: based on the idea that no computer code will ever capture all of the human design experience, large user control capabilities were introduced. The user can select different disciplinary analyses modes, modify number and range of optimization variables and constraints, define aggregated or separate objectives, and execute successive global and local optimization processes, each time inspecting results and changing the parameters accordingly. The user should in this way support the MDO process, by introducing human experience in the loop and actively steering the design process towards the most promising regions of the search space.
- ***Optimization framework generality***: several types of global and local optimization methods were implemented or integrated, in order to provide full optimization capabilities for MDO. A global approach is in fact necessary for tackling architectural and technological discrete trade-offs, design from scratches and multiple objectives ("design-

[4] PRESTIGE announcement is available on ESA's website at: http://www.esa.int/esaMI/Education/SEMBTFO4KKF_0.html

to-performance", "design-to-cost", "design-to-reliability", ...). Local algorithms are instead extremely useful for efficient subproblem optimization (e.g. trajectory optimization) and solutions refinement.

- ***Object-Oriented architecture modularity***: flexible data storage structures as well as modular disciplinary classes with clear interfaces were defined, easing the maintainability of the MDO environment and its expandability to other design problems (e.g. re-entry vehicles, re-usable launchers, ...).
- ***Computational efficiency:*** particular attention was paid to computational efficiency, keeping in mind the criticality of computational resources in MDO, even though parallel computing with OpenMP for shared memory machines was partially exploited (see **Annex 2**).

In the continuation of this dissertation, the terminology ***SVAGO Version 1 (V1)*** and ***Version 2 (V2)*** will be used to identify the two modelling steps outlined in the previous paragraph for the ***conceptual*** and ***early preliminary*** detail levels. Note however that most of the programming and implementation details are not reported here due to length constraints, focusing instead on the engineering analysis aspects.

1.4. Research industrial applicability

As justified in the previous paragraphs, the application of MDO to the conceptual design phases has the potential to sensibly reduce the LCC of STS, or other complex engineering projects. For this reason, the present research has an immediate industrial applicability in the European framework, in particular with respect to the FLPP as well as VEGA upgrade studies mentioned in **Paragraph 1.1**.

To exemplify the possibilities enabled by MDO, a case study taken from the FLPP and carried out with the developed MDO environment is summarized in Figure 4. Here the objective is the definition of baseline concepts for a family of launchers, ranging from 3 tons to 8 tons of payload capability to GTO, as a successor to Ariane 5. Without focusing on the numerical results, the purpose is to show how a complex launch vehicle family study, requiring the evaluation of a large number of possible configurations, can be performed with a very limited effort. In this case, ~2 man-hour a day and a single personal computer were necessary for a time span of one month to implement the full study logic on the left of the figure. This made available a large set of design solutions for each of the three configurations, enabling the designer to select the preferred concepts among those in multi-objective Pareto fronts for maximum payload mass vs. minimum launch mass or minimum launch cost vs. minimum launch mass. Although this achievement looks particularly impressive, the confidence which can be entrusted in the final solutions must be carefully weighted by quantitatively evaluating the expectable errors, in order to make the MDO approach viable in an industrial context.

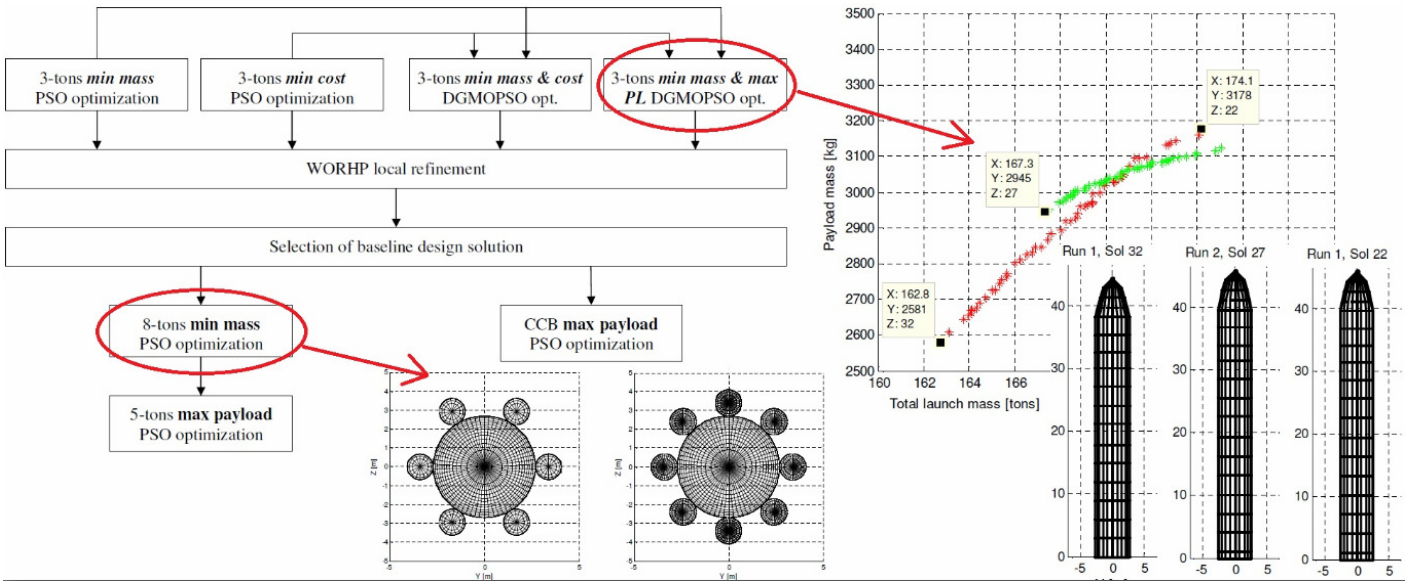


Figure 4: Example of study logic and type of results for a launch vehicles family under investigation within FLPP.

If MDO viability is verified, several possible applications of the ELV MDO environment are already available within the European industrial context, such as the following:

- *FLPP conceptual studies*, as the one described above.
- *CDF pre-studies*: the concurrent engineering approach was proven to be very successful in improving the conceptual design process of space systems by several similar organizations around the world, such as ESA ESTEC's CDF. However, an initial guess solution needs to be available in order to start the iterative design cycle, and its quality is of

fundamental importance for the final outcome. In this context, semi-automatic MDO environments would be extremely useful in search space pruning, possibly allowing to drastically improve the final selected concept.

- *Launcher upgrade*: in case of payload capability enhancements or other upgrades, it is possible to analyze several configurations with MDO, in order to identify which component ensures the most cost effective upgrade. For example, different VEGA configurations can be investigated to determine which stage (or stages) would be more suited for modification to satisfy specific requirements.
- *Component upgrade*: conceptual studies may be performed also for specific launcher components, such as Ariane 5's ESC-B upper stage or a European version of VEGA's AVUM upper stage, to evaluate the system level impact of modifications to a specific stage or booster.

1.5. Work outline

In the continuation of the PhD dissertation, a brief overview of the state-of-art in topics related to MDO is first given in **Chapter 2**, including a short history of MDO, multidisciplinary problem formulations and architectures, global and local optimization methods and relevant MDO tools and applications. Although the focus of the research is on the engineering modelling, **Chapter 3** outlines the optimization architecture and main algorithms which were integrated in the MDO environment, in order to provide the reader with an operational understanding of the optimization aspects.

Then, the most consistent section of the work describes the two engineering modelling steps. In particular, details are given on the development of the models and on the validation results, first for the conceptual step in **Chapters 4** and **5** and successively in **Chapters 6** and **7** for the early preliminary step. Finally, applicative results for test cases relevant to the European scenario highlighted in the previous paragraph are presented in **Chapter 8**, whereas lessons learned from the MDO effort and related conclusions are drawn in **Chapter 9**. Several research lines are also suggested in this last chapter as possible future developments, directly stemming from the main open points identified within the work described in the previous chapters.

In addition to the main body of the dissertation, annexes are included to schematically document relevant side aspects of the research. Specifically, **Annex 1** reports a summary of the CPU times required for MDA and MDO processes, **Annex 2** presents the most important aspects of the parallelization of global optimization algorithms with OpenMP for shared memory machines, and **Annex 3** reviews the system-level software architecture and the related GUI.

Questo mondo è una meraviglia. Non c'è niente da fare, è una meraviglia. E se riesci a sentirti parte di questa meraviglia – ma non tu, con i tuoi due occhi e i tuoi due piedi; se Tu, questa essenza di te, sente d'essere parte di questa meraviglia – ma che vuoi di più, che vuoi di più? Una macchina nuova?

CHAPTER 2

STATE-OF-ART

MDO is a very wide research field, encompassing problem decomposition and problem formulation techniques, MDO architectures and other connected areas such as engineering modelling, global and local optimization, sensitivity analyses, meta-modelling techniques, etc. For this reason, a comprehensive overview of the current state-of-art is not practical for length limitations nor is within the scope of this dissertation. However, an overview is given in this Chapter regarding the topics which are more strictly related to the present work. In particular, the three main sections below provide an historical background of MDO and of its recent applications to STS design, followed by an analysis of typical MDO problem formulations, and finally summarize the main concepts of local and global optimization.

For more details, Technical Notes D1a (**[A1]**) and D1b (**[A2]**) include a much broader review of the state-of-art in all topics mentioned above. In particular, decomposition methods and the related single and multi-level MDO architectures were subjects of a thorough investigation. Although these were not successively applied, due to insufficient time and limited applicability to the developed models, a broad bibliographic overview is documented in **Chapter 4** of **[A1]**. Moreover, an extensive collection of the available engineering level analysis methods for all the disciplinary areas involved in the design of launch and re-entry vehicles is the subject of **Chapter 4** of **[A2]**.

2.1. Multi-disciplinary Design Optimization

2.1.1. Historical origins and development of MDO

The first idea of collaboration between disciplines has originated in the aeronautical field from the inherent coupling of aerodynamics and structures, causing important issues such as flutter and loss of control surface effectiveness. The study of these effects resulted after World War II in the birth of the discipline of aeroelasticity, aimed at the equation level integration of the two disciplines. Later, the introduction of two major technological advancements - composite materials and active control systems - led to more complex interdisciplinary couplings, involving the tailoring of the material properties to meet specific requirements and the introduction of controls discipline in the integrated design process (e.g. see the design of the forward-swept-wing fighter Grumman X-29A, dating back to the early 80's). It was however with the first conceptual studies of advanced aerospace vehicles (supersonic/hypersonic aircrafts or RLV), that MDO gained a relevant role in the engineering design: for instance, whereas in a conventional transport aircraft the exploitation of multidisciplinary global optimality could make a difference between very good and merely good vehicle performances, in hypersonic aircrafts it may very well make the difference between flying and staying on the ground.

With the increasing complexity, the representation of the interdisciplinary couplings moved from the equation-level typical of aeroelastic approach to the integration by decomposition, for which each discipline remains a self-contained task and the challenge lays in the definition of the "information channels" among these blocks. Optimization algorithms were also added to the design loop, initially based on the gradient local approach such as Sequential Linear Programming (SLP) or Sequential Quadratic Programming (SQP), and more recently expanded to global techniques such as Genetic Algorithms, Particle Swarm Optimization, and others.

The first large efforts towards multi-disciplinary design were carried out in the 80's and 90's mainly in the US by researchers at the Multidisciplinary Optimization Branch of NASA's Langley Research Centre (LaRC) and its collaborating universities, such as Sobieski, Alexandrov, Haftka, Lewis, Braun, Kroo, Olds and others. These investigations focused on the development of *decomposition techniques*, to deal with the task of efficiently dividing a large multi-disciplinary problem

in smaller blocks. The most straightforward MDO architecture is represented by the single level "Black-Box" Optimization (BBO), for which a MDA model built from disciplinary analyses constitutes a black-box function, taking as inputs the design variables and returning as outputs objectives and constraints. In BBO, the MDA is directly coupled to a System Level Optimization (SLO) algorithm, handling the full optimization problem. Although very simple in implementation, BBO may result inadequate for applications with large number of design variables/constraints and requiring numerous, computationally intensive iterations to converge to a consistent design. With the limited computational resources of the 80's and 90's, multi-level decomposition techniques arose naturally, in an attempt to reduce problem's complexity, size and expense by means of disciplinary autonomy in both implementation and execution.

Figure 5 summarizes the possible forms of disciplinary integration, including the equation-level approach typical of aero-elasticity, No Decomposition (ND) approaches such as BBO, Hierarchic Decomposition (HD) and Non Hierarchic Decomposition (NHD). The choice between the latter two is in general not left to the designer, but is determined by the coupling characteristics of the specific system under study. HD applies if the system can be divided into a set of black boxes (i.e. input-output functions) representing the physical subsystems or analysis disciplines, forming a multi-level hierarchy as shown in **Figure 6**. The flow of input/output information is only in the vertical direction and no data are transmitted between pairs of boxes located at the same decomposition level, which is instead the case of NHD systems. The possibility to exploit HD simplifies the process, since each block only needs information from higher levels, hence analyses and optimizations at the same level can be executed concurrently.

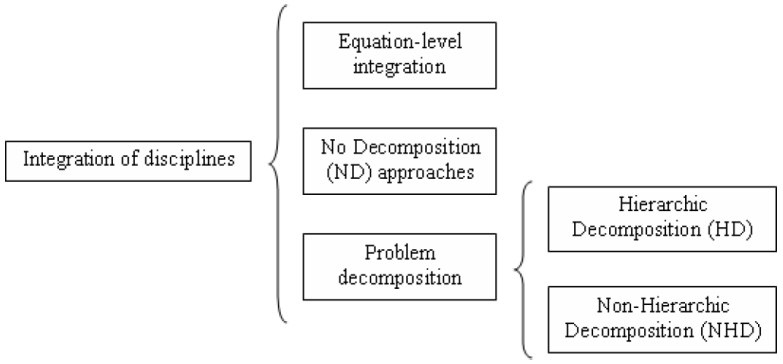


Figure 5: Different forms of disciplinary integration in MDO

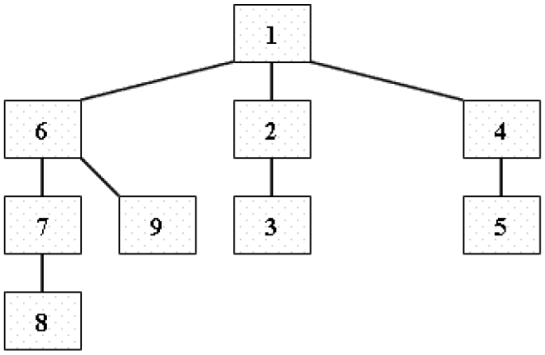


Figure 6: Example of hierarchical decomposition

In the relatively short time span between 1982 and 1998, a large number of *problem decomposition* methods and resulting MDO architectures were developed, starting from HD and then proceeding to NHD approaches, with the goal of improving one or more of the desirable features for MDO: a) *quality of the optimum*, intended as the capability of converging to the "best" optimum (or at least a very good one) over a large search domain; b) *effectiveness*, sometimes referred to as *robustness*, defining the probability of success of the optimization process; c) *efficiency* in terms of computational resources required to achieve convergence; d) *reusability* of existing codes; e) *maintainability* of the software environment; and f) *parallelization potential*. A chronological list of the most interesting proposed decomposition techniques is given here, together with bibliographic references for more details:

- Multilevel Optimization by Linear Decomposition (MOLD): proposed by Sobieszczanski-Sobieski (NASA LaRC) in 1982 (**[7]**, **[8]**) and based on multi-level HD, it's probably the most deeply investigated multi-level approach, with interesting applications for example in multidisciplinary wing design problem, as in references **[9]** and **[10]**.
- Concurrent SubSpace Optimization (CSSO): proposed again by Sobieszczanski-Sobieski in 1989 (**[11]**, **[12]**) as the first attempt to tackle NHD problems of large dimensions through the division in subspaces with separate optimization processes running concurrently and exchanging sensitivity information.
- Collaborative Optimization (CO): NHD technique proposed in 1994 by Kroo and Braun at Stanford University (**[13]**, **[14]**, **[15]**, **[16]**) and constituted of a bi-level decomposition based on the IDF formulation (see **Paragraph 2.2**), where separate subsystem optimizers satisfy domain specific constraints and a system level coordination is in charge of the overall objective(s) and of the compatibility constraints among coupling variables. CO has had a large success in the research community, with plenty of derivatives and applications.
- Bi-Level Integrated System Synthesis (BLISS): alternative NHD architecture proposed at NASA LaRC in 1998 (**[17]**, **[18]**, **[19]**), especially meant for those problems that include many system level design variables which are shared by more disciplines and for which the assignment to a single block is not physically meaningful. As CO, several derivatives of BLISS were introduced, most notably BLISS-2000 (**[20]**) which applies the bi-level scheme directly to a generated RSM instead of the original disciplinary tool.

In addition to the above problem decomposition methods, several researchers also focused since the beginning of the 90's on single level architectures with different *problem formulations*, intended as the way of handling the disciplinary and

interdisciplinary couplings in MDA. The most commonly followed classification is that advanced in 1992 by Cramer and others (**[21]**), including the so-called All-At-Once (AAO), Multi-Disciplinary Feasible (MDF) and Individual Disciplines Feasible (IDF) formulations. Since MDF and IDF were directly investigated in this research work, **Section 2.2** is dedicated to problem formulations.

In this apparent jungle of methodologies, several attempts were made to quantitatively compare different formulations and decomposition techniques on both mathematical benchmarks and reference applicative problems (see for example LaRC's MDO test suite in **[22]**), mainly using effectiveness and efficiency metrics. Worth mentioning in this area are the works of Balling and Wilkinson (**[23]**, 1996), Alexandrov and Lewis (**[24]**, 2000), Brown and Olds (**[25]**, 2005), Tedford and Martins (**[26]**, 2006), De Wit and Van Keulen (**[27]**, 2007), and Yi (**[28]**, 2008).

From the analysis of these efforts, a clear hierarchy cannot be outlined, and such a hierarchy is actually unlikely of ever being defined. In fact, effectiveness and efficiency of each architecture are strictly related to the characteristics of the problem being solved, not to mention to the practical implementation details varying from case to case. Nevertheless, few considerations can be considered generally valid. In particular, the level of detail implemented in the disciplinary models and the available computational power are fundamental aspects in the choice of the MDO technique. For reasonably "small" (in number of variables and constraints) and "fast" (evaluation times up to seconds or tens of seconds) models, BBO architecture is the most adequate due to its simplicity and effectiveness. However, BBO does not scale well with the size and complexity of the problem, hence is not well suited when aiming much beyond the conceptual-level. In fact, larger design spaces complicate the search for global optimum, possibly making it an impossibly complex task for a single system level optimizer, therefore requiring problem decomposition. For this purpose, MOLD is definitely to be taken into consideration whenever lateral communications are limited, in light of its remarkable computational savings. Among multi-level NHD approaches instead, CO is promising when the disciplines are not excessively coupled, though significant convergence errors with respect to the coupling constraints have been reported. Finally BLISS and its derivative BLISS-2000 have shown very good effectiveness and efficiency in several studies, but can only be used when derivative information is available (i.e. smooth problems). The authors of reference **[25]** provided the most convincing evidence of BLISS-2000's superiority on a real-world test problem (MDO of NASA's ACRE-92 RLV configuration), but this is only one experimental data point and should be complemented by similar studies on other applications. As reported by the authors, *"... while it is tempting to declare BLISS the most promising multi-level MDO technique available, this study is only one data point. It is statistically impossible to declare BLISS "most promising" unless more studies employing the blocking effect substantiate the results found here."*

Since the beginning of the 2000's, work on the development of new MDO architectures drastically diminished, in favour of a significant increase of applications. In fact, the availability of larger computational resources resulted in the first attempts to introduce high fidelity methods in multidisciplinary models through massive parallelization. Moreover, classical engineering level disciplinary models have been often integrated in recent years in MDO tools for diverse applications, exploiting only low cost computing infrastructures (i.e. personal computers or at most workstations). An overview is given in the next paragraph of those applications which are most relevant to the design of STS.

2.1.2. Recent applications of MDO to STS design

The initial steps in the development of multi-disciplinary models for Space Transportation Systems (STS) were undertaken already in the 90's, in parallel to the studies on problem decomposition and MDO architectures. In particular, Olds at North Carolina University first tackled the MDA of a Rocket Based Combined Cycle (RBCC) and a dual fuel Single-Stage-To-Orbit (SSTO) systems, through manual iteration between disciplines and the exploitation of sensitivity information and response surfaces (**[29]**, **[30]**). Researchers at NASA LaRC then expanded on that approach, handling the design of SSTO RLVs through MDF and IDF single level architectures as well as CO (**[31]**, **[32]**, **[33]**). However in these first attempts, the lack of computational power restrained the application to the study of a specific launcher configuration and imposed the use of very simple disciplinary models. Additionally, no global optimization approach was included in these studies due to its limited maturity at the time.

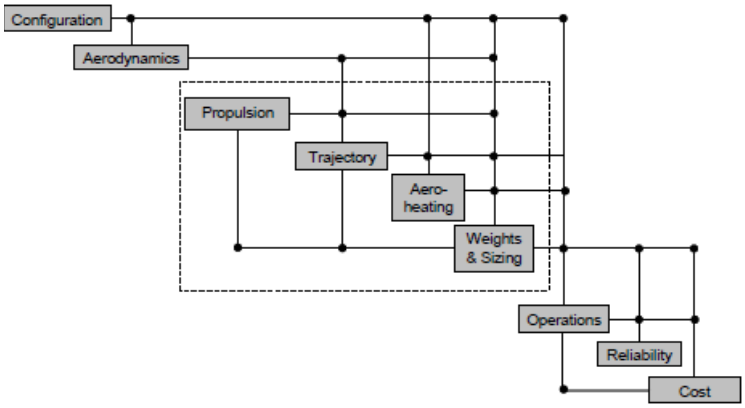
A continuative research activity spanning the years 1998-2006 was carried out at the Space Systems Design Laboratory (SSDL)[5] of Georgia Tech under the supervision of Dr. Olds, in an effort to improve the accuracy and completeness of the engineering models as well as to automate the MDA process. A generic framework for MDO was presented in **[34]** (1998), exploiting a commercial tool named *ModelCenter* (see next paragraph on general purpose MDO frameworks) to allow for easy replacement of analysis blocks. This ensured the flexibility to plug in or substitute analysis codes according to the specific application, so that different types of vehicles were investigate through MDAO in the following years.

In the area of ELVs, a Heavy Lift Launch Vehicle (HLLV) family called *Centurion* was designed in **[35]** for NASA's exploration beyond Earth orbit, reporting an interesting trade study on the core stage's engine as well as one of the first

[5] SSDL's website at www.ssdl.gatech.edu includes an interesting page on MDO including a large number of related publications.

attempts to integrate cost, reliability and operations disciplines in an automated design cycle for ELVs. However, formal optimization was not exploited, except for trajectory design which was performed with the well known NASA's POST[6].

In the 90's and 2000's, NASA's X-programs were aiming at the development of a feasible and viable RLV, as well as other agency-funded studies worldwide. In parallel to these efforts, Georgia Tech investigated four advanced SSTO and TSTO concepts, with the same MDA methodology with manual trade studies employed for *Centurion*, but slightly different DSMs to account for the specific application. See references **[36]** for *Hyperion* (VTHL RBCC SSTO), **[37]** for *Magel* (TSTO, magnetic 1st stage propulsion), **[38]** for *Aztec* (VTHL RBCC TSTO), and **[39]** for *Lazarus* (sled-assisted HTHL RBCC SSTO). In spite of the attractiveness of these novel concepts, the economical evaluation included in SSDL's studies was disappointing, showing no reduction of launch costs with respect to current ELVs. This conclusion confirms the technical evaluation on RLVs stemming from NASA X-33 program made by Art Stephenson, Director of NASA's Marshall Space Flight Center (MSFC) on the 1st of March 2001: *"We have gained a tremendous amount of knowledge from these X-programs, but one of the things we have learnt is that our technology has not yet advanced to the point that we can successfully develop a new reusable launch vehicle that substantially improves safety, reliability and affordability"*.



Discipline	*Analysis Tool*
Configuration	Pro/E
Aerodynamics	APAS (HABP)
Propulsion Design	REDTOP
Trajectory	POST 3-D
Weights & Sizing	MS Excel
Structures	GT-STRESS
Aeroheating	Miniver
Reliability	Relex
Operations	Historical Estimates
Cost	TRANSCOST

Figure 7: DSM and related disciplinary codes for the MDA process of the Centurion HLLV family, as taken from [35].

In the last 15 years, MDO found applications to STS and other space systems also outside the United States, through both agency-sponsored and industrial efforts within the main space fairing nations[7]. Even though work in other countries is not documented here, European developments in ELVs MDO are briefly outlined, in order to provide a better understanding of the immediate background of the present PhD research. Specifically, recent industrial (EADS, **[40]** and **[41]**) and academic (ONERA, **[42]**) researches have tackled the design of ELVs for future European needs. Automatic trade-offs among different configurations were introduced exploiting global optimization (Genetic Algorithms), obtaining interesting design results and proposing an ingenious new decomposition method for ELV design (**[42]**). Nevertheless, these solutions are still limited to an early conceptual level, employing classical disciplinary models of fidelity similar or lower than those in **Figure 7**. A similar modelling approach but exploiting gradient-based algorithms was followed at ASTOS (Aerospace Trajectory Optimization Software), aiming at design optimization starting from an initial guess in the desired search space region (**[43]**). Widening the horizon to non ELV applications, worth mentioning in Europe is the work at DLR's SART[8] in Germany, ONERA[9] in France, and Politecnico di Milano[10] in Italy.

From an analysis of the research presented above for the current European scenario, two main weaknesses were identified: 1) the lack of a complete optimization architecture efficiently handling both global search with respect to multiple objectives and local refinement, exploiting at least to some extent HPC; and 2) the inadequate characterization of the accuracy of the disciplinary models employed to generate the MDA. Work for the PRESTIGE research focuses exactly in these areas, with the development of a large optimization infrastructure, as described in **Chapter 3**, and the two-steps engineering modelling which constitutes the main topic of **Chapters 4** to 7.

The last subject of large relevance in the historical review of MDO applications is the introduction of high fidelity. A fascinating example of how engineering knowledge is sometimes passed on to the next generations, legacy computer

[6] https://post2.larc.nasa.gov/index.html
[7] Mainly in Europe, Russia, Japan, China and India. Emerging MDO centres are the Multidisciplinary Aerospace Design Optimization Research Centre (MADORC) and the Centre for Aerospace Systems Design and Engineering (CASDE) in India. International organizations are AIAA's MDO Technical Committee and International Society of Structural and Multidisciplinary Optimization (ISSMO).
[8] http://www.la.dlr.de/ra/sart/index.php.en
[9] http://www.onera.fr/prf-en/index.php
[10] http://www.aero.polimi.it/space/?Major_Research_Areas:Optimization:Multidisciplinary_Optimization

programs originally developed in the 60's and 70's for various types of disciplinary analyses have remarkably come back into fashion in past and present MDO applications. In fact, they have been considered for a long time the only methods capable of providing engineering assessments with a reasonable accuracy and at the very limited computational expense necessary for MDO. But as said before, the drastic increase in computational power of modern computer clusters allowed to consider the introduction of high fidelity analyses, mainly CFD and FEM, in the MDA of aerospace vehicles.

Most of the examples of high fidelity information's introduction in MDO are in the field of advanced aircrafts. The first large scale attempt was documented by LaRC in 2000 (**[44]**, **[45]**, **[46]**) after almost 10 years of efforts in the development of a MDO tool supporting upcoming configuration decisions for NASA's High Speed Research Program (HSRP). This program targeted an operational supersonic cruise transport able to carry 300 passengers for 5000 nautical miles at Mach 2.4, but was cancelled due to unattainable cost objectives (over-cost with respect to subsonic transport <20%). During HSRP MDO studies, three increasingly complex integrated design models were developed, up to including CAD geometry, non-linear Euler CDF applied to a linear aerodynamics code and detailed FE analysis with loads convergence process under linear aero-elasticity assumption. Very large MDA problems, to be solved with state-of-art workstations, were originated from this model. Standard gradient-based optimization was supposed to be applied to evaluate the optimal design point of different configurations, but the program was cancelled before actual optimization results could be achieved. Nevertheless, work at LaRC was reused for successive efforts in high-fidelity MDO at NASA, most remarkably the already mentioned BWB experimental aircraft designed and tested at NASA GRC's MDAO branch[11].

With the exploitation of parallel computing, other examples have been reported in very recent years of high fidelity multidisciplinary models, as EU's HISAC project for a silent supersonic aircraft and NASA's Advanced Engineering Environment for launchers (**[47]**, **[48]**), suggesting that this approach may be affordable for industrial trade-off analysis of complex systems. Although high fidelity was not tackled in this PhD research due to both time and computational resources limitations, the bibliographic review highlighted a definite need for further research in this area.

As a final note, particularly interesting are also MDO applications to the conceptual design of REV, both classical capsules (**[49]**, **[50]**, **[51]**) and advanced waveriders for aero-gravity assists (**[52]**). Due to the limited relevance of these studies to the present research, refer to **Section 5.2** of **[A1]** for more details.

2.1.3. Commercial general purpose software environments for MDO

When dealing with the task of the MDA of an engineering product, one of the most difficult challenges lays in the integration of inhomogeneous disciplinary analyses in a uniform tool capable of efficiently exchanging data and ensuring the correct execution of all its parts. Three main approaches to this issue can be identified:

1. *Monolithic design code*: integration of the contributing analyses as subroutines of a larger program, hence intended for single user execution with simplified self-developed models for pre-conceptual or conceptual design.
2. *Loosely integrated analyses*: at the opposite extreme of the monolithic code, foresees use of standard legacy codes of any fidelity with manual exchange of data and very little electronic integration of contributing analyses. This is the classical design approach in large organizations but does not provide an environment for automatic MDO.
3. *Tightly integrated analyses*: intended as the "wrapping" together of available codes, from conceptual to preliminary and even detailed level, with the purpose of reducing manual cycle times and enabling integrated optimization.

In the last 20 years, the research community has focused the efforts on the latter approach, so that the first commercial engineering frameworks for MDO started to appear on the market. General purpose MDO frameworks are intended as software tools designed to couple available disciplinary codes and expedite the input, output and execution phases of those analyses in a single MDA process. Some of them also include mathematical methods for global and local optimization, Design of Experiments (DoE) and Response Surface Methodology (RSM), sensitivity analysis, etc. A very interesting recent effort from NASA LaRC (**[53]**) retraced all papers submitted at AIAA's Multidisciplinary Analysis and Optimization conferences from 1984 to 2004 which concern with engineering frameworks, summarizing the work of several companies and stressing on four key ideas in framework research: *modularity*, *data handling*, *parallel processing* and *user interface*.

Note that the largest difference between these software and SVAGO lays in the general purpose characteristics of the former, which usually do not include specific disciplinary codes but rather provide GUI environments for the automatic linking of disciplinary tools which are either provided by the user or available through commercial agreements. Among those mentioned in **[53]** and other highlighted by further investigations, the following commercial, industrial and research frameworks are worth mentioning here together with bibliographic references for further details:

- Phoenix Integration's *ModelCenter* (**[54]**): probably the most known engineering framework for MDO, it includes several legacy codes and a user friendly GUI where the user can select the available tools and/or add its own, coupling the different disciplines in the desired Design Structure Matrix (DSM). Among others, it has been used in the past decade by NASA for its AEE, by Georgia Tech for its IMAGE framework and by Spaceworks Engineering for several industrial trade studies.

[11] http://mdao.grc.nasa.gov/index.html

- Engineous Software's *iSight* (**[55]**): commercial MDO framework especially focused on optimization, with an extension for Parallel Computing named Fliper.
- Technosoft's *Adaptive Modeling Language* (**[56]**): object oriented modelling framework for collaborative knowledge-based engineering.
- Lockheed Martin's *FASTPASS* (Flexible Analysis for Synthesis, Trajectory, and Performance for Advanced Space Systems) tool (**[57]**), developed in the 90's and flowing into the more recent *RCD* (Rapid Conceptual Design) tool used by SkunkWorks team for several projects (**[58]**).
- Georgia Tech SSDL's *IMAGE* (Intelligent Multidisciplinary Aircraft Generation Environment) environment (**[34]**), the already mentioned academic engineering framework used for the design of several ELV and RLV concepts.
- *OpenMDAO*[12]: Open-source MDO framework, proposed in 2008 by NASA GRC (**[59]**) and based on Python programming language. In spite of the challenges of managing the development of a complex tool in an open source community, the inherent advantages of this approach are potentially very interesting in MDO, for which each contributor may provide specific disciplinary codes or optimization functionalities.

2.2. MDO problem formulation

Problem formulation in MDO can be viewed as a translation step between the multidisciplinary problem as stated in the engineering models and the standard optimization problem to be solved by the selected optimization method. The choice of problem formulation defines how the disciplinary analyses are executed and how both interdisciplinary couplings and feasibility constraints are handled. Differently from the choice of problem decomposition (HD or NHD), this is not strictly imposed by the system characteristics, although they strongly influence it. Besides, problem formulation and decomposition are strictly intertwined, with different possible combinations defining a wide variety of architectures. In this section, a single-level BBO approach is assumed (no decomposition), in order to focus on the formulation aspects.

The classification advanced by Cramer in **[21]** (1992), includes the MDF, IDF and AAO formulations. A similar classification was proposed in **[60]**, representing a more general and versatile taxonomy, but is not reported here to avoid confusion on the terminology. MDF, IDF and AAO are instead described below, after few necessary definitions:

- *Individual discipline feasibility*: satisfaction of all state equations of an analysis code for the given discipline (i.e. the inputs and outputs of the disciplinary analysis are coherent with respect to the model).
- *Interdisciplinary mappings*: relations to convert coupling variables from one discipline to another; a typical example are the interpolation laws between aerodynamic and structural grids in an aeroelastic problems.
- *Multidisciplinary feasibility*: condition for which, in addition to the individual disciplines feasibility, all outputs of each discipline exactly match the inputs of the others through the interdisciplinary mappings, and vice versa.

Multi-Disciplinary Feasible, MDF:

The optimization algorithm controls only the "real" design variables (e.g. number of stages, chamber pressure, ...) and multidisciplinary feasibility is maintained at each optimization iteration[13]. The MDF formulation avoids the introduction of additional optimization variables and constraints, besides those defined by the MDO problem, and allows the exploitation of specialized software developed for the individual disciplines. However, the optimizer "wastes" time in ensuring full feasibility even when far from the optimum of the problem. This can be particularly time consuming in case of computationally expensive MDA processes which require several iterations to converge.

Individual Disciplines Feasible, IDF:

The need for full multidisciplinary feasibility is eliminated in the IDF formulation, for which only individual disciplines feasibility is maintained at each iteration while the optimizer is in charge of driving the individual disciplines toward multidisciplinary feasibility and optimality. The interdisciplinary coupling variables are duplicated for this purpose, imposing compatibility constraints to be satisfied only at the end of the optimization process. The main advantage with respect to MDF lays in the elimination of MDA iterations, possibly determining large savings in computational time per evaluation for those models that include feed-back information flows. However, one variable and one constraint need to be added to the optimization problem for each coupling variable, complicating the algorithm's search. In general, IDF progressively loses efficiency as the number of interdisciplinary couplings increases and as the ratio of time required for each iterative cycle and the total MDA time decreases. Note that if no iterations are necessary, in absence of feed-back information, IDF formulation loses significance.

[12] http://openmdao.org/

[13] Note that this does not obviously include that physical constraints of the engineering model are satisfied at each iteration.

All-At-Once, AAO:

Some confusion can be found in literature regarding AAO: several papers talk about a BBO approach using the term AAO, which does indeed recall the terminology of BBO (see for example **[43]**). However, the original classification of Cramer is used here, for which in AAO both individual disciplines and multidisciplinary feasibility are not assured but have to be satisfied by the optimization algorithm. This is allowed to optimize the design of the system and to solve the governing equations simultaneously, being in direct control of the values of all design, state, and coupling variables. No full disciplinary analysis is performed and the governing equations are only used to evaluate the residuals of the governing equations, whereas the SLO assumes the difficult task of satisfying such equations through additional disciplinary compatibility constraints. In this way, no kind of feasibility is ensured until convergence, but very time consuming analyses can be skipped in favour of much shorter residual evaluations. Even though in some cases this seems extremely attractive in terms of CPU efficiency, there are two relevant disadvantages: 1) the optimization problem's size often explodes, with the addition of one variable and one constraint for each disciplinary state variable and each interdisciplinary coupling, and 2) AAO requires a high degree of software integration, with only limited amount of previously available disciplinary code which can be directly exploited. In particular, this latter point greatly increases the required work load, so that very few applications of this method are reported in literature.

Under the same reasoning, AAO formulation was not considered for implementation in the present research, hence a wider overview is given only of MDF and IDF. When first facing MDO problems, MDF formulation is usually employed without even realizing it has been categorized with such term. In MDF in fact, there is basically no translation from the engineering model to the mathematical optimization problem. **Figure 8** shows how the MDA process is handled in MDF, represented on the left, and in two types of IDF, named here partial and full IDF but also referred to as Partial and Full Optimization-Based Decomposition in literature. When moving from MDF to IDF, information chains between disciplinary blocks need to be broken. First, only feed-back loops are eliminated in partial IDF, resulting in a system analysis which can be executed without iterations but still sequentially (middle). Then, feed-forward flows are removed as well, originating a fully decoupled system (right). With full IDF, this advantage has to be traded against the increase in the number of coupling variables/constraints. For this research, no parallelization of the MDA is foreseen, since parallelization of the SLO algorithm is much easier to obtain and much more efficient in terms of speed-up (see **Annex 2**). Thus, full IDF was also not considered for implementation, reducing the problem formulations under investigation to MDF and partial IDF.

As it was true for MDO architectures described in the previous section, it is hardly possible to define which approach is generally the best, with the specific problem characteristics being discriminating. Nevertheless, IDF approach usually leads to smoother functions for gradient-based optimization and was in several cases reported to ensure better efficiency. For example, it was shown in reference **[13]** that CPU times can be reduced by ~30% using partial IDF instead of MDF for a simplified aircraft design problem. This was confirmed in **[61]** for TSTO branching trajectories, where better payload performances are obtained in shorter computational times with partial IDF. Here, full IDF is also confirmed to be much slower than both other approaches, due to the sensibly larger problem's size.

As a final note, IDF was the logical precursor to more complex MDO architectures, most notably CO. IDF still involves in fact decomposition of the analysis only, whereas the actual design work is entirely done by the SLO process. Hence, subsystems are not permitted to apply discipline-specific design expertise and to use specialized tools for disciplinary optimization. Especially for problems with weak interdisciplinary coupling and large numbers of domain-specific constraints, this centralized design approach is not appropriate, hence leading to the concept of CO, allowing the distribution of both the analysis and the optimization among separate independent modules.

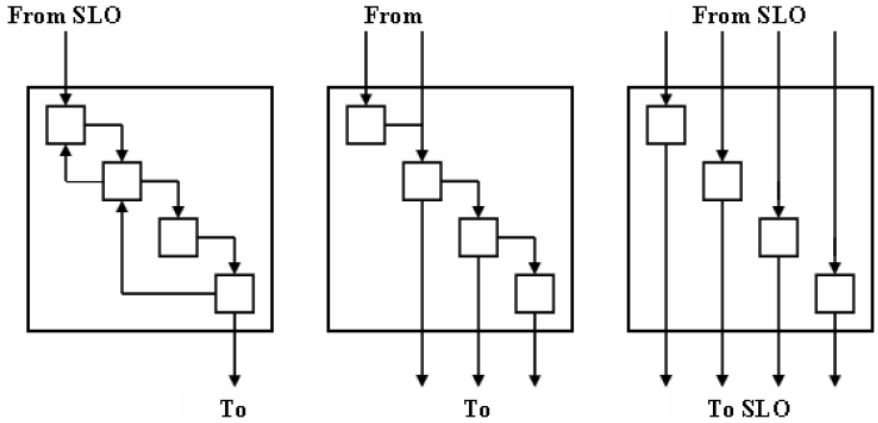


Figure 8: MDA processes for MDF and IDF formulation. Squares represent disciplinary analyses, arrows information flow among disciplines and from/to the SLO algorithm. Left: MDF, sequential execution with iterations. Centre: Partial IDF, sequential execution without iterations (no feed-backs). Right: Full IDF, parallel execution (no feed-back nor feed-forward).

2.3. Optimization methods

Optimization is the second substantial part of MDO efforts, besides multidisciplinary modelling. Since the focus of this research is however on the engineering aspects of MDO, only a brief introduction is given here on the wide topic of optimization. In particular, a justification is given of the ever increasing number of applications of Global Optimization (GO), which has started to complement classical Local Optimization (LO) since the end of the 80's. Most of the recent MDO researches for launch, re-entry or aero gravity assist systems, such as those described in **Paragraph 2.1.2**, are examples of GO space applications. Moreover, global interplanetary trajectory optimization has been one the most researched fields of astrodynamics in the past ten years (**[62]**). The widespread participation of research institutes, space agencies and private companies around the world in recent editions of the Global Trajectory Optimization Competition (GTOC)[14] testifies to the important role that GO currently plays in the design of space systems.

To understand this interest, consider a generic single-objective constrained optimization problem, described by $\boldsymbol{N_{VAR}}$ optimization variables $\boldsymbol{x}$, an objective function $y = f(x)$ to be minimized and $\boldsymbol{N_{CSTR}}$ equality and/or inequality constraints $g(x) \geq 0$ to be satisfied. A local minimum is defined as a solution $\boldsymbol{x}$ for which the function $\boldsymbol{f}$ is smaller than at all other feasible nearby points. A global minimum is instead a point with the lowest function value among all feasible points in the design space. For convex programming problems, local solutions are also global solutions. Unfortunately, most practical applications result in non-convex problems with numerous local optima (i.e. multi-modal problems), complicating the search for the global solution.

Local, or gradient-based, optimization algorithms constitute a wide class of methods with the common feature of exploiting the information from the gradient of constraints and objective to determine a search direction towards feasibility and optimality. In case of multi-modal problems, different local solutions may be obtained depending on the arbitrarily provided starting point, and the global optimum cannot be reached unless the initial guess is within its region of attraction. Moreover, gradient information is not available in case of discontinuities, either in the decision variables space or in the objective/constraints functions space. For these reasons, many engineering optimization problems cannot be solved with a purely local approach, being characterized by large and multi-modal search spaces as well as by the presence of discrete variables and non-smooth models. MDO problems with architectural and/or technological trade-offs and interplanetary trajectories surely fall within this category.

Additionally, gradient-based techniques are not well suited to multi-objective optimization. Consider extending the problem formulation by allowing for $\boldsymbol{N_{OBJ}}$ contrasting criteria, to be concurrently optimized. The usual optimality definition based on the objective function value no longer applies, and has to be substituted with the concept of Pareto-optimality. A solution for which a given objective cannot be improved without degradation of any of the others is in this case said to be non-dominated or Pareto-optimal. An entire set of non-dominated optimal solutions in general exists for multi-objective problems, called Pareto front. The goal of the optimization is therefore the attainment of a set of solutions not only as close as possible to the real Pareto front of the problem, but also with uniform spread of solutions over its entire extension.

Several methods exist to obtain multi-objective non-dominated solutions with local algorithms, the most common being synthesizing the objectives in a single aggregate function and repeating the optimization with different weight parameters. However, this technique requires a separate run per each desired Pareto solution, is sensitive to the front's shape since non-convex Pareto fronts cannot be identified (**[63]**), and demands some knowledge of the problem to efficiently tune the weight parameters. Besides, several locally optimal fronts exist for multi-modal problems, where gradient-based algorithms may get stuck exactly as in single-objective optimization.

GO techniques were gradually introduced starting from the 80's to overcome the above limitations. *Stochastic, population-based* approaches were included in the optimization architecture of SVAGO MDO environment, since they are particularly well suited for highly multi-modal and multi-objective problems. This class of methods mimics natural phenomena and physical processes, as for example natural selection and survival of the fittest, social behaviour of bird flocks, metals annealing, and others. An initial set of solutions, the so-called population, is usually randomly provided and evaluated. Constraint violations and objective function values are then used to rank the solutions, and a combination of random search and nature-based heuristics allows updating the population for the next iteration. Therefore, no derivative information is required, and issues related to discontinuities and local optimality are avoided. Stochastic jumps ensure the algorithms' exploration capability, which is necessary to locate globally optimal regions in large search spaces. Moreover, population-based algorithms are particularly well suited for handling multiple objectives, since the simultaneous evaluation of a set of solutions allows identifying several members of the Pareto front in a single run. Finally, easiness of implementation and of parallelization represents a further important advantage of GO techniques.

On the other hand, global algorithms usually require a larger numbers of function evaluations, and due to a rather coarse control over the decision variables cannot provide the accurate location of the optimum. In fact, there's substantial evidence (see for example **[21]**) that local calculus-based optimization is, when applicable, much more efficient with respect to global

[14] GTOC was conceived by ESA in 2005 to foster the advancement of optimization techniques for space trajectories design. GTOC homepage is available at http://www.esa.int/gsp/ACT/mad/op/GTOC/indexII.htm

approaches. For this reason, GO and LO often need to be used complementarily, by first exploring the search space through global methodologies and then refining a few selected solutions with a local algorithm. This was also the choice taken in the present research, as will be described in **Chapter 3**.

Another drawback of GO is the presence of several tuning parameters, whose values sensibly affect the convergence properties. Some of these parameters are common to all population-based techniques. In particular, population size and number of iterations determine the search space coverage capability and the number of function evaluations, whereas the size of the archive containing the best non dominated solutions defines the maximum number of final Pareto-solutions. Others are instead peculiar to each method, and allow controlling the balance between stochastic and deterministic search, the convergence rate versus the global exploration capability, and the relative importance of non domination against diversity of solutions.

Since a comprehensive description of local and global optimization methodologies is out of the scope of the present work, refer to classical texts on optimization such as **[64]** or **[65]** for LO and **[66]** or **[67]** for GO. Moreover, the main subject of the author's MSc thesis (**[68]**)[15] was the development and quantitative comparison of several stochastic population-based multi-objective global algorithms, and constitutes a shorter overview of GO approach with respect to full texts.

[15] Available at http://publishing.yudu.com/Freedom/Anpj3/FrancescoCastelliniS

La montagna è fatta per tutti, non solo per gli alpinisti: per coloro che desiderano il riposo nella quiete come per coloro che cercano nella fatica un riposo ancora più forte.

CHAPTER 3

OPTIMIZATION ARCHITECTURE

3.1. Global and local architecture

As justified within the state-of-art chapter, BBO is considered the most adequate MDO architecture when the problem's size and computational requirements are not excessive, due to its simplicity and effectiveness. Since the main objective of the present research is to quantitatively evaluate the accuracy of engineering-level multidisciplinary analysis methods, the developed problem seemed tractable with such architecture. Moreover, it was evident that since the multidisciplinary design of launchers involves discrete trade-offs, large search spaces and multiple objectives, a combination of both GO and LO algorithms is necessary. As a result, the MDO architecture was kept as simple as possible, with a single system level optimization process capable of global exploration and handling all design variables, constraints and objectives.

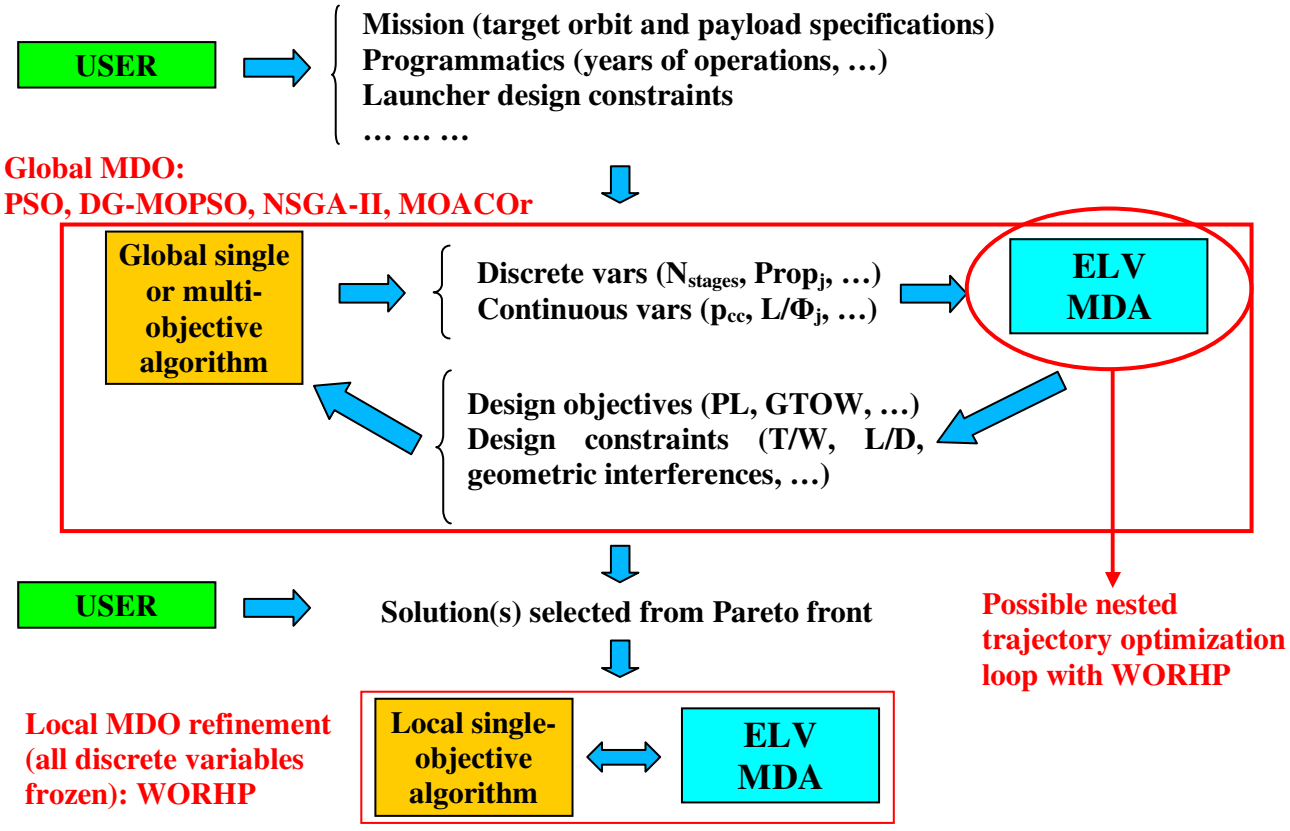


Figure 9: Overview of the single-level, multi-process BBO architecture defined for SVAGO MDO environment. Note that the parameters qualitatively described here are only examples, for a comprehensive definition of all data flows see Chapters 4 and 6.

This process can be sequentially repeated several times, varying each time the number and bounds of the design variables, the objectives and/or the constraints, in order to allow the user to steer the optimization according to the human

experience. More than one global runs can be executed in series, possibly with different GO algorithms. Successively, one or more of the obtained solutions can be locally refined by means of a LO method, freezing all discrete design variables and selecting a single objective function from an aggregate of the possible objectives.

This *single-level*, *multi-process*, *BBO* architecture is schematically represented in **Figure 9**, where only one global optimization and one local refinement processes are shown, highlighting three main entities:

- ***User***: in charge of the initial definition of the problem and of the selection of solutions for local refinement. In case of multiple GO runs, the user is also in charge of modifying the parameters of the processes at each repetition.
- ***Optimization algorithm***: either GO or LO algorithm, which is in charge of the single-level optimization process.
- ***Launcher MDA***: multidisciplinary analysis cycle for the design of ELVs, which returns the values of objectives and constraints starting from the launcher's architecture and design parameters, as given by user and optimizer.

In addition to the BBO approach, a Nested Optimization Loop (NOL) in the Trajectory subsystem was considered. In fact, trajectory optimization variables are not shared with other subsystems, so that variables defining the design can be frozen by the SLO and a NOL can be executed at each MDA. The advantages of this approach lay in a sensible decrease of the number of optimization variables of the system-level problem and in the possibility to exploit local optimization for the trajectory subproblem, which is generally recognized as a more efficient approach. However, the NOL architecture was discarded after preliminary testing due to the very long computational times required for each trajectory optimization. In fact, even after relevant improvements in the local trajectory optimization process which were mostly spurred by the NOL concept (see the separate Trajectory Optimization Study in **[A11]**), the time required for a full ascent trajectory optimization is in the order of 30 seconds to 1 minute. This is one order of magnitude larger than the total MDA time with a simple trajectory simulation (0.5-2 seconds), largely offsetting the advantage in terms of system level problem's size. For this reason, the NOL architecture results much less efficient than the original BBO, especially since the full ascent optimization needs to be performed for each technically feasible launcher design solution, even far from the optimum region.

Although the NOL architecture was not further investigated, the availability of a trajectory optimization loop nested within the MDA process proved several times to be useful when the MDA had to be executed stand-alone, with no SLO. In particular, a nested trajectory optimization is necessary to estimate the payload performance of a given launcher with fixed design, which was repeatedly the case during the validation procedure for both engineering modelling steps.

In the next two sections, the global and local optimization methods which were integrated within the developed MDO environment are briefly described. Then, **Chapters** from **4** to **7** will be focused on the "ELV MDA" blue box of **Figure 9**, with particular attention on the MDA process and information flows in **Sections 4.1** and **6.2** for the two modelling steps.

3.2. Global optimization algorithms

The availability of global optimization algorithms for integration in the MDO environment directly stems from the previous MSc research of the author. In particular, one of the goals of that research ([68], **[69]**) was the quantitative comparison of different stochastic, population-based, multi-objective, global techniques. Seven algorithms from four classes of these methods were considered: Genetic Algorithms (GA), Simulated Annealing (SA), Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO). For generic reviews of these techniques, the interested reader should refer to Deb (**[63]**) and Coello Coello (**[70]**) for GA, Ingber for SA (**[71]**), again Coello Coello (**[72]**) for PSO, and Dorigo (**[73]**) for ACO. The *"No Free Lunch"* theorem by Wolpert and Macready (**[74]**) mathematically states that no optimization method outperforms all others in the totality of the problems. To confirm this statement, several papers in recent literature present comparisons among GO techniques (e.g. **[75]** and **[76]**), but none was able to identify a consistent hierarchy, being mainly based on qualitative considerations and limited to few test cases and one or two classes of methods. Through the quantitative comparison carried out within the MSc research on a wide set of benchmark cases, a hierarchy was instead proposed for each class of mathematical problems, to be used in support of the algorithm's selection for any applicative problem. Moreover, the well-known NSGA-II and the newly developed DG-MOPSO were identified as the most promising methods for the optimization of ascent trajectories, which has direct applicability to the ELVs MDO problem.

As a consequence, both NSGA-II and DG-MOPSO algorithms were integrated within the MDO environment, together with MOACOr which also showed interesting performances on other classes of mathematical problems. Finally, a single-objective PSO was added to the set of global optimizers, because of its better convergence properties when one objective is sufficient. The following sections briefly introduce these techniques but do not provide details due to length limitations. The original bibliographic references are also given for further study, in addition to the MSc research in **[68]**.

3.2.1. PSO-1D: Single-objective Particle Swarm Optimization

Eberhart and Kennedy introduced the idea of Particle Swarm Optimization in 1995 (**[77]**), inspired by the social behaviour of bird flocks. PSO follows the generic scheme of all stochastic population-based global algorithms, which is represented in **Figure 10**. First, an initial set of solutions to the optimization problem to be solved is randomly generated, which are evaluated in terms of objective function and constraints. In PSO, each solution $\boldsymbol{X}$ is a particle and the set is the

swarm. Then, the initial set is updated through nature-based heuristics which allow moving towards feasibility and optimality, and the procedure is repeated until some convergence criterion is met. In PSO, the update of the set of solutions at each iteration is described as the swarm's movement, through the velocity formula:

$$\begin{cases} V_{jk}(t+1) = \chi \cdot \left[w \cdot V_{jk}(t) + c_1 r_1 \cdot (P_{jk}(t) - X_{jk}(t)) + c_2 r_2 \cdot (P_{gk}(t) - X_{jk}(t)) \right] \\ X_{jk}(t+1) = X_{jk}(t) + V_{jk}(t+1) \end{cases}$$

where ($\boldsymbol{X_{jk}}$, $\boldsymbol{V_{jk}}$) are position and velocity of particle j with respect to the k-th variable of the vector $\boldsymbol{X}$ of optimization variables. The PSO's velocity law comprehends an inertia term partly maintaining the velocity of the previous iteration and an attraction towards the personal best position (*pbest*, $\boldsymbol{P_{jk}}$) and the globally best position (*gbest*, $\boldsymbol{P_{jk}}$) found so far by the swarm. Stochastic aspects are demanded to the random numbers $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$, whereas the main parameters affecting the exploration vs. exploitation capabilities of the algorithm are the *inertia* $\boldsymbol{w}$, *self-confidence* $\boldsymbol{c_1}$ and *swarm-confidence* $\boldsymbol{c_2}$, and a constriction parameter $\boldsymbol{\chi}$ introduced later in **[78]**. A maximum velocity $\boldsymbol{V_{max}}$ is also employed in several versions of the algorithm. No more details are given here, but a large bibliography exists, testifying the enormous success of this method, including theoretical analyses, empirical studies for the definition of the parameters settings and several comparisons with other evolutionary techniques, which convey the feeling that PSO may be considered superior in most applications.

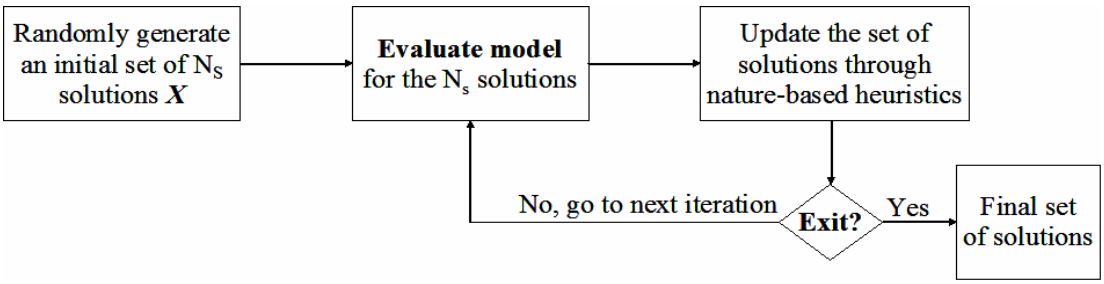


Figure 10: Generic scheme for stochastic, population-based, global optimization algorithms such as PSO.

3.2.2. DG-MOPSO: Double-Grid Multi-Objective Particle Swarm Optimization

The first Multiple Objectives Particle Swarm Optimization (MOPSO) algorithm was proposed in 2002 by Coello Coello and others (**[79]**), the main modification with respect to the single-objective version consisting in the definition of several swarm's "leaders" according to Pareto-dominance principles instead of a single gbest as for PSO-1D. The Double-Grid MOPSO (DG-MOPSO), developed within the above mentioned MSc research effort, improved MOPSO through the idea of a geometrical grid similar to the one introduced in PAES evolutionary algorithm (**[80]**). In particular, the original MOPSO was complemented with new features related to the selection of the swarm's leaders and to the pruning of the archive of non dominated solutions. The idea of a geometric grid dividing the n-dimensional objectives space is exploited, defining two nested structures, a coarse outer grid and a fine inner grid obtained by further bisection of the former. The purpose of the double grid is to move the particles towards leaders in the less crowded inner grid of the same outer grid where they are located. This allows both to improve the convergence properties of the algorithm and to obtain a more uniform distribution of solutions.

3.2.3. NSGA-II: Non-Dominated Sorting Genetic Algorithm II

Genetic Algorithms are the oldest family of stochastic global techniques, inspired by the evolution of individuals through chromosomes' mating and natural selection, and therefore can count on a long research record. Reference **[68]** reports a summarized history of the evolution of GAs for both single and multiple objectives, together with an introduction on the subject. Among all multi-objective GAs, the Non-Dominated Sorting Genetic Algorithm-II (NSGA-II) is by far the most successful in terms of number of applications and bibliographic references. NSGA-II was developed in 2000 (**[81]**) by Prof. Deb and others improving the original NSGA. In NSGA-II, several drawbacks of NSGA are corrected, through the introduction of more efficient operators (e.g. fast non-dominated sorting technique for the creation of successive fronts of non-dominated solutions, new mutation operator) and of the concept of elitism, which ensures the survival of the fittest individuals from one generation to the following.

3.2.4. MOACOr: Multi-Objective Ant Colony Optimization for real problems

Ant Colony Optimization, introduced by Dorigo in 1999 (**[82]**) for discrete optimization problems and extended to continuous problems in 2006 (**[83]**), is based on the behaviour of the ants which leave a pheromone behind them returning to the nest after having found food, in order to lead other ants to the food's location. The version integrated within the MDO environment is the Multiple Objectives Ant Colony Optimization for real problems (MOACOr), proposed by Najera in 2007

([**84**]) and based on the idea of incorporating in the original ACOr the multi-objective structure of NSGA-II. After short experimentation with the code provided by Najera, NSGA-II mutation operator was also added to MOACOr, in order to avoid premature convergence towards a single solution when this solution is dominant with respect to all others at any given point of the optimization.

3.3. Local optimization algorithms

3.3.1. WORHP: "We Optimize Really Huge Problems" NLP solver

In light of the research collaboration with the University of Bremen described in **Section 1.3**, the gradient-based algorithm WORHP (We Optimize Really Huge Problems)[16] was available for use within the MDO environment. WORHP is a SQP method, designed to solve sparse large-scale NLP problems with more than 1.000.000 variables and constraints. Although this size is out of the scope of the MDO problem at hand, WORHP is also capable of handling smaller dense problems, and was therefore considered more than valid for the purpose of local refinement of the global solutions found by the global algorithms as well as for fast trajectory optimizations.

WORHP was recently developed by the joint work of teams from the University of Bremen and University of Wurzburg ([**85**]), elaborating on the general idea of SQP methods which was introduced by Han back in 1977. Since then, SQP has probably been the most employed technique for the solution of practical optimization problems, due to its robustness and convergence properties. Among the advantages of WORHP are the efficient calculation of derivatives through different methodologies (e.g. group strategies for finite differences and a new sparse BFGS technique), a number of recovery strategies and a useful reverse communication architecture, which ensures flexibility and full control over the optimization process. The robustness of WORHP was proved by the CUTEr test set, which consists of 920 sparse large-scale and small dense problems, for which WORHP was able to efficiently solve 99.5% of all test cases, a significantly better result with respect to other state-of-art SQP libraries such as SNOPT or IPOPT.

[16] www.worhp.de

I would have written a shorter letter, but I did not have the time.

CHAPTER 4

CONCEPTUAL DESIGN MODELS

This chapter presents in details the engineering models for the conceptual design of ELVs, developed from scratches as first modelling step, which will be referred to from now on as Version 1 (V1). Three hierarchy levels can be identified within these models: system level, subsystem level and unit level, where the latter represents the smaller functional blocks that build up each subsystem's disciplinary analysis. The complete description of all the disciplinary models, Input/Output (I/O) parameters, information flow among subsystems and with the system, optimization variables, constraints and objectives, constitutes a large amount of information. For this reason, the graphical tool of the Design Structure Matrix (DSM) is used in **Section 4.1** to synthesize the system level Multidisciplinary Design Analysis (MDA) process. The DSM is complemented by tables reporting the system level variables as well as the I/O parameters for each subsystem. All disciplinary analyses are then presented in **Sections 4.2 to 4.7**, which give an overview down to the unit-level, although several minor details are not given here due to length limitations (for a more comprehensive description, see **[A3]**). A similar descriptive approach will be followed in **Chapter 6** for the early preliminary level models.

*Note that the software infrastructure associated to the MDO environment is extremely complex. In order to focus on the engineering aspects rather than the software issues, this chapter only presents the engineering models and variables flow. Details are however given in **Annex 3** regarding SVAGO software architecture, C++ classes and interfaces.*

*Moreover, a complete MDA cycle and disciplinary models were designed for the conceptual level analysis of classical, capsule shaped REV, though implementation was not carried out in order to focus on the more relevant ELVs application. Due to length limitations, REVs' MDA model is not reported here, but a thorough description is the subject of **Chapter 4** of Technical Note D6 (**[A6]**) at the system level and of **Chapter 3** of Technical Note D2 (**[A3]**) at the subsystem level.*

4.1. Multidisciplinary design cycle and optimization problem

The engineering modelling of launch systems is a particularly complex task, even when restricting the target to classical (i.e. simple cylindrical stages and boosters with no wings), unmanned, and expendable launchers. Moreover, with the rationale explained within the research motivation (**Section 1.1**), the MDO approach imposes the constraint of limiting the single processor CPU time for a full MDA to less than one second[17].

When this constraint is combined with the need to exploit only freely available software, the choice of the main engineering tools is actually rather limited. For this reason, several researchers in the past (**[40]**, **[41]**, **[42]**, **[43]**) have independently converged toward common tools, such as Chemical Equilibrium with Applications (CEA) for propulsion performance analysis and Missile DATCOM for aerodynamics, or analogous in-house developed algorithms.

The disciplinary models for SVAGO were developed following this common approach, with the additional decision of allowing different modelling fidelities according to the impact of each discipline on the vehicle's performance. Assumptions were taken regarding such impacts, which were then a posteriori verified with a thorough sensitivity analysis (see **Chapter 5**). These assumptions include largely favouring the propulsion system analysis, which inarguably constitutes the most relevant subsystem of any launch vehicle, with respect to other disciplines with lower system level impact, such as aerodynamics. Stemming from these considerations, analysis processes for each discipline were defined and the related models were implemented in C++ classes. As described above, open source Fortran legacy tools were also integrated,

[17] All computational times mentioned in this dissertation are referred to a single ***Intel Core2 Duo T6500 processor (2.10 GHz, 2 MB L2 Cache, 800 MHz FSB)*** with ***4096 MB DDR2 (800 MHz) RAM***. Use of a single core is assumed, unless explicitly stated otherwise.

employing calls to external executables and defining scripts for writing/reading input/output files. These are the already mentioned CEA for propulsion and Missile DATCOM for aerodynamics, as well as two geometry visualization tools called 3-View and Silhouette, used in combination with the GNU plot library.

The next paragraph outlines the DSM for the multidisciplinary design cycle. Based on the DSM, the optimization problem is stated in the following paragraphs, in terms of optimization variables, possible objectives and constraints.

4.1.1. Design Structure Matrix

The DSM is a graphical tool commonly used to represent complex MDA processes. It is mainly constituted by several components:

- Boxes on the diagonal of the DSM represent the subsystem analysis modules, which are executed sequentially from the left to the right. External tools integrated in each discipline are reported within brackets. The trajectory block is highlighted in blue indicating a discipline for which a NOL may be taken under consideration.
- Lines connecting the blocks represent data flows among disciplines. These are above the diagonal for feed-forward information and below the diagonal for feed-back information. The latter require iterations of the involved blocks in order to achieve convergence of the design.
- Each dot corresponds to a data structure containing one of the following vectors:
  - Input parameters $\mathbf{P_j}$: vector of input parameters fixed by the user for subsystem j.
  - Design variables $\mathbf{X_j}$: vector of optimization variables for subsystem j. Several optimization variables belong to more than one X_j, being shared by more disciplines.
  - Coupling variables $\mathbf{X_{jk}}$, outputs of discipline j which are needed as inputs by discipline k.
  - Output data $\mathbf{Y_j}$: analysis outputs for discipline j, some of which constitute the constraints and objectives of the optimization problem.

Figure 11 shows the DSM synthesizing the models developed for the conceptual design of ELVs (V1), with all data structures contained in the DSM detailed in Table 1. As stated in the introduction, user interactivity is one of the most important features in a successful MDO environment. For this reason, in addition to P_j, the user has full control on the boundaries of all optimization variables and on the thresholds of all constraints. In particular, upper and lower bound of the continuous design variables as well as allowed values of the integer variables can be specified, overwriting the defaults. In this way, any desired design constraint can be enforced (e.g. "all solid" launch vehicle, a given number of stages or engines in each stage, etc.), even throughout several successive optimization processes, directing the search towards regions of the search space that appear more appealing in light of the designer's personal experience.

However, note that the description of the models given in this chapter assumes that the user does not restrain any of the design features, hence describing the most generic optimization problem that can be set up with V1 models.

As shown in Figure 11, the MDA cycle is constituted by the sequential execution of Propulsion, Geometry, Aerodynamics, Weights, Trajectory, Cost and Reliability analyses. The Propulsion module defines, for each stage and boosters set, the propulsion system's dimensions (for Geometry), inert masses (for Weights) and performances (for Trajectory). It includes the external public domain code CEA from NASA as well as several other developed models. The Geometry module exploits then information from the system design variables and propulsion outputs to generate the complete external geometry of the launch vehicle. This information is stored during the MDA in a data structure with the format required by Missile DATCOM aerodynamic code. However, for visualization purposes, a file in Langley Wireframe Geometry Standard (LaWGS) format can be generated, which is particularly convenient due to its simplicity and compatibility with a number of public domain codes for both visualization and aerodynamic analysis. Two of these LaWGS compatible tools for visualization are included in the geometry module: 3-View and Silhouette, from the Public Domain Aeronautical Software (PDAS[18]). The LaWGS file generation script and the visualization codes are not executed within MDO processes in order not to slow down the process, but only at the end of the optimization so that the final solution(s) can be easily visualized by the user. The Aerodynamics and Weights modules follow Geometry, and could be executed in parallel if necessary[19]. The former runs the external Missile DATCOM software from USAF, obtaining estimates of the drag, lift and pitching moment coefficients as a function of Mach number and total angle of attack. The latter implements instead several WERs for each of the launcher's structural and non structural components, building up a rather detailed Weights Breakdown Structure (WBS) of the entire vehicle. With all launcher data now available, the Trajectory module can be executed, integrating the 3 Degrees of Freedom (DoF) Equations of Motion (EoM) from take-off to orbit insertion (or crash on the planet) and thus evaluating the error on the final orbital parameters. Along the whole trajectory, the path

[18] Available for a small fee at www.pdas.com .

[19] Although in both V1 and V2 MDA cycles several modules could be executed in parallel, this is not recommended due to the inhomogeneous distribution of the work load among the different processors. Moreover, population-based global optimization algorithms are extremely well suited for parallelization and it is therefore much simpler to define the entire execution of the MDA process as a single-processor job. Annex **2** describes the developed OpenMP parallelization scheme for the employed PSO algorithms.

constraints (q_{dyn}, n_{ax}, q_{heat}, α_{tot}) are computed and a static controllability verification is performed, by comparing aerodynamic and thrust torques. As a last step, Cost and Reliability models are called, assessing the overall Life Cycle Cost (LCC) of the launch vehicle and its Mission Success Probability (MSP).

For the concurrent optimization of launchers trajectory and design, one may expect the outputs from the trajectory to be fed back to the weights estimation in terms of heat and structural loads. Due to the simplicity of the weight estimation models however, these loads can be synthesized by only three parameters, namely the maximum axial acceleration, dynamic pressure and heat flux encountered from take-off to orbit. $n_{ax,max}$, $q_{dyn,max}$, and $q_{heat,max}$ constitute the feedback information flow $\mathbf{X_{54}}$, which would result in the Weights and Trajectory disciplines needing to be iterated to achieve convergence on the inert masses. However, it was decided to eliminate this iterative cycle, as indicated in the DSM, through the IDF technique already theoretically described in **Paragraph 2.2**. The three load parameters can in fact be added as optimization variables of the problem, hence making them available to both the Weights and Trajectory disciplines. Within the former, $n_{ax,max}$, $q_{dyn,max}$, and $q_{heat,max}$ are used as inputs to the WERs to define the inert masses, whereas in the latter they are employed as thresholds for the related path constraints.

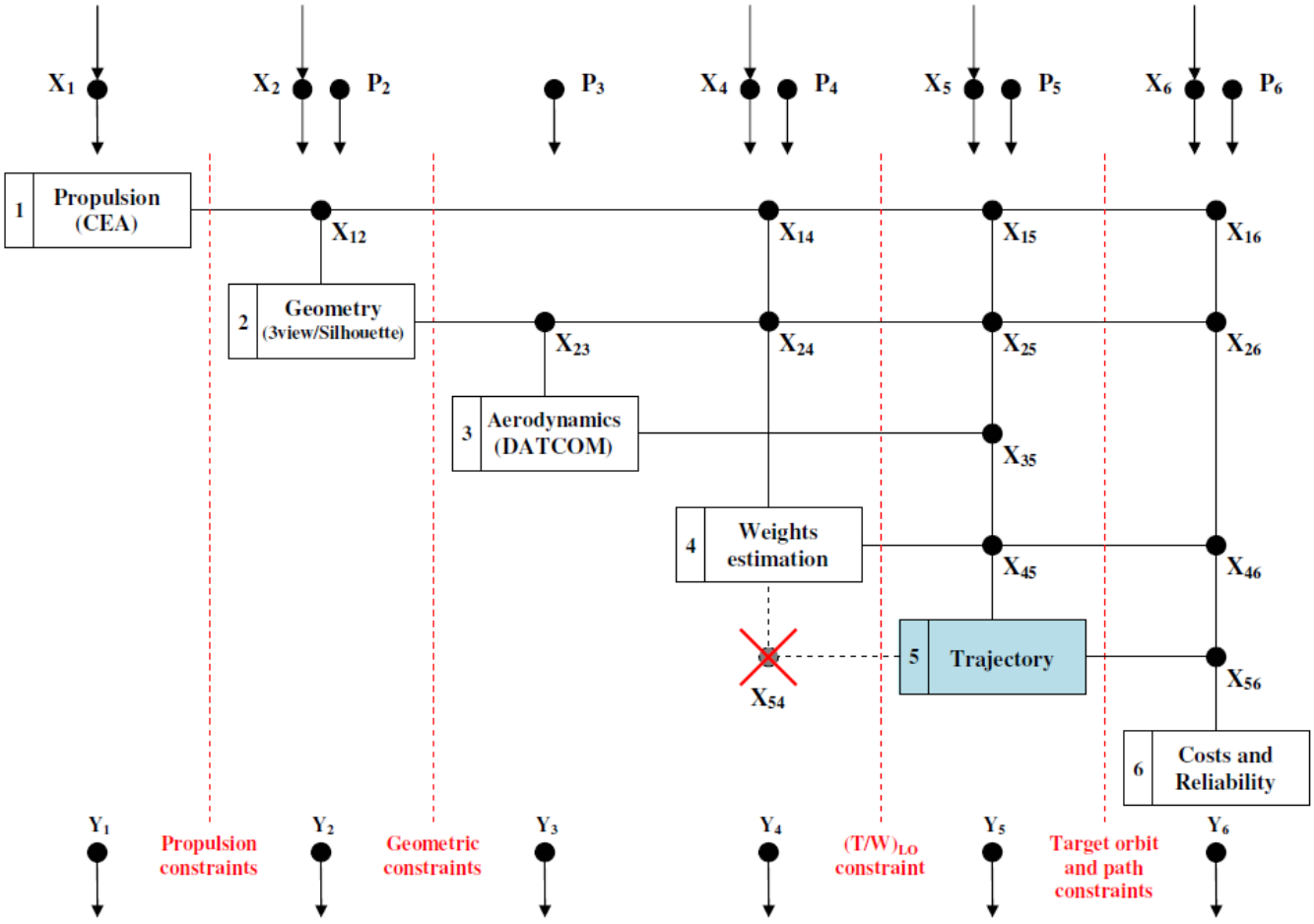


Figure 11: Design Structure Matrix for the conceptual level Multidisciplinary Design Analysis of Expendable Launch Vehicles

Dotted red vertical lines represent gates of the MDA execution where one or more constraints have just been evaluated; here, the process is interrupted if one or more of the constraints are violated, with some threshold which may be properly tuned on a case-by-case basis, in order to avoid unnecessary computations when far from the feasible region. Note that this stopping procedure is employed only in case of global optimization whereas for local refinements the full design loop is anyway executed, to avoid introducing discontinuities which would complicate the search of the gradient-based algorithm.

As a final note on the multidisciplinary design process, all design variables are normalized in the range [0; 1] and all constraints/objectives are kept as close as possible to the unity. When a MDA cycle is started, all optimization variables are mapped from the [0;1] range to the actual design space for each variable, which may be boolean, integer or double. This has the twofold advantage of easing the work of local optimization algorithms, which are extremely sensitive to bad scaling of the problem, and of enabling a simpler visualization of the obtained design solutions (i.e. all optimization variables can be plotted in a graph with variable ID in x and optimal value in y in the [0; 1] range).

Data	Variables set	Detailed list of variables
$\mathbf{X_1}$	Launcher's architecture Stage/bset's propulsion system architecture Stage/bset's propulsion system design Stage/bset's dimensions	Conf, N_{stages}, $N_{boosters}$, CCB, SET, SBT $N_{eng,j}$, OTS_j, $ID_{OTS,j}$, $Prop_j$, $Feed_j$, LCE_j, ThC_j, RC_{US} T^*_j, H^*_j, $\alpha_{p,j}$, $p_{cc,j}$, $(A_{cc}/A_t)_j$, $Nozzle_j$, $\delta_{div,j}$, TVC_j, $\delta_{TVC,j}$ $M_{prop,j}$, $D_{const,j}$, $(L/D)_j$
$\mathbf{X_2}$	Launcher's architecture Stage/bset's propulsion system architecture	Conf, CCB, N_{stages}, $N_{boosters}$ $N_{eng,j}$
$\mathbf{X_4}$	Launcher's architecture Stage/bset's prop.sys. architecture & design Stage/bset's dimensions Stage/bset's structural layout Trajectory loads and system level variables	Conf, CCB, N_{stages}, $N_{boosters}$, SBT $Prop_j$, $\alpha_{p,j}$ $M_{prop,j}$, TA_j, TT_j, SM_j $n_{ax,max}$, $q_{dyn,max}$, $q_{heat,max}$, SSM, RL, LPM
$\mathbf{X_5}$	Launcher's architecture Stage/bset's prop.sys. dimensions Stage/bset's prop.sys. architecture & design Stage/bset's structural layout Trajectory loads Trajectory control variables	Conf, CCB, N_{stages}, $N_{boosters}$ $M_{prop,j}$ $\alpha_{p,j}$, T^*_j, H^*_j, ThC_j, RC_{US}, $N_{eng,j}$, $N_{eng,out,j}$, LCE_j, TOE_j TA_j $n_{ax,max}$, $q_{dyn,max}$, $q_{heat,max}$ PLSF, $\Delta\theta_{PO}$, t_{PO}, t_{decay}, $\Delta\psi_{PO}$, $\theta_{LS,jk}$, $\psi_{LS,jk}$, $\psi_{US,k}$, ζ_{BTL}, $\Delta\theta_{BTL,i}$, $\theta_{BTL,f}$, $\delta_{th,jk}$, Δ_{BI}, Δt_{BI}, Δ_{CI}, Δt_{CI}, Δt_{CB}, $t_{LSC,j}$
$\mathbf{X_6}$	Launcher's architecture Stage/bset's propulsion system architecture Stage/bset's propulsion system design Stage/bset's structural layout System level variables	Conf, CCB, N_{stages}, $N_{boosters}$, SET, SBT $N_{eng,j}$, $N_{eng,out,j}$, OTS_j, $ID_{OTS,j}$, $Prop_j$,$Feed_j$, LCE_j, ThC_j, RC_{US}, LCE_j, TOE_j $\alpha_{p,j}$, T^*_j, $Nozzle_j$, TVC_j, $N_{test,eng,j}$ TT_j, SM_j FoS, RL, PPM
$\mathbf{P_2}$	Payload	L_{PL}, D_{PL}
$\mathbf{P_3}$	Aerodynamic database settings	$N_{data,Mach}$, $N_{data,\alpha tot}$
$\mathbf{P_4}$	Payload	$n_{ax,max,PL}$, $q_{heat,max,PL}$
$\mathbf{P_5}$	Environment and initial conditions Payload and target orbit Trajectory control variables settings	Planet, Atmosphere, λ_{launch}, δ_{launch}, H_{launch} M_{PL}, $X_{CoG,PL}$, $n_{ax,max,PL}$, $q_{heat,max,PL}$, a_T, e_T, i_T, a_{tol}, e_{tol}, i_{tol}, a_{norm}, e_{norm}, i_{norm}, H_{clear}, H_{atmo}, $\alpha_{tot,max}$, $\Delta\psi_{max,atmo}$, $\Delta\theta_{max,atmo}$, $\Delta\psi_{max,exo}$, $\Delta\theta_{max,exo}$ $N_{nodes,\psi,j}$, $N_{nodes,\theta,j}$, $N_{nodes,\delta th,j}$, $\delta_{th,min}$, CB, LSC_j
$\mathbf{P_6}$	Programmatic aspects	$N_{launches}$, $N_{years,ops}$, f_0, f_1, f_3, f_7, f_8
$\mathbf{X_{12}}$	Stage/bset's propulsion system dimensions	D_j, $L_{Ox,j}$, $L_{Fuel,j}$, $L_{grain,j}$, $L_{eng,j}$, $D_{eng,j}$
$\mathbf{X_{14}}$	Stage/bset's propulsion sys. weights & dim.	$M_{EA,j}$, $M_{unused,j}$, $L_{Ox,j}$, $L_{Fuel,j}$, $L_{grain,j}$, D_j
$\mathbf{X_{15}}$	Stage/bset's propulsion system performance	$I^*_{sp\_j}$, $A_{e,j}$, $p^*_{e,j}$, $H_{min,j}$
$\mathbf{X_{16}}$	Stage/bset's propulsion system weights	$M_{EA,j}$, $M_{unused,j}$
$\mathbf{X_{23}}$	Launch vehicle external geometry	$D_{BS,j}$, $L_{BS,j}$, $L_{N,BSj}$, $(D_{core}, L_{core}, L_{N,core})$ or $(X_k, R_k)_{core}$, A_{ref}, L_{ref}
$\mathbf{X_{24}}$	Launch vehicle external geometry	$L_{IS,j}$, $L_{N,j}$, L_{PLF}, L_{VEB}
$\mathbf{X_{25}}$	Launch vehicle external geometry	A_{ref}, L_{ref}
$\mathbf{X_{26}}$	Launch vehicle external geometry	L_{PLF}, V_{PLF}
$\mathbf{X_{35}}$	Launch vehicle aerodynamic database	$C_L(M, \alpha_{tot})$, $C_D(M, \alpha_{tot})$, $C_M(M, \alpha_{tot})$
$\mathbf{X_{45}}$	Launch vehicle weights	M_{PLF}, M_{PLA}, $M_{inert,j}$, $X_{CoG,PLF}$, $X_{CoG,PLA}$, $X_{CoG,inert,j}$
$\mathbf{X_{46}}$	Launch vehicle weights	M_{PLF}, M_{PLA}, $M_{inert,j}$, GTOW
$\mathbf{X_{54}}$	Trajectory loads	$n_{ax,max}$, $q_{dyn,max}$, $q_{heat,max}$
$\mathbf{X_{56}}$	Trajectory phases	N_{phases}, $t_{phase,j}$
$\mathbf{Y_1}$	Stage/bset's propulsion system	OTS availability constraint, complete engine specifications
$\mathbf{Y_2}$	Launch vehicle's external geometry Launch vehicle's internal geometry	A_{ref}, L_{ref}, $L_{tot,core}$, $L_{tot,bset,j}$, LaWGS geometry file, geometric constraints $L_{EA,j}$, $D_{EA,j}$, $L_{IT,j}$, $L_{IS,j}$
$\mathbf{Y_3}$	Launch vehicle aerodynamic database	$C_L(M, \alpha_{tot})$, $C_D(M, \alpha_{tot})$, $C_M(M, \alpha_{tot})$
$\mathbf{Y_4}$	Launch vehicle weights	Detailed WBS, GTOW, $(T/W)_{takeoff}$
$\mathbf{Y_5}$	Trajectory data	PLSF, trajectory data at regular intervals, final orbit and path constraints
$\mathbf{Y_6}$	Cost and reliability data	Detailed CBS and RBS, LCC, CpL, MSP

Table 1: Summary of optimization variables, user parameters, interdisciplinary couplings and disciplinary outputs for V1 models. In blue: optimization variables NOT shared with other subsystems. In red: feed-back information eliminated through IDF approach. Subscript j indicates a parameter repeated over several elements of the vehicle (i.e. once per stage and/or boosters set).

In the next paragraphs, the ELV design optimization problem with V1 models is outlined through the detailed definition of all variables defined in Table 1: user parameters, optimization variables, inter-disciplinary couplings, analysis outputs, optimization objectives and constraints. Then, in the continuation of the chapter, each of the disciplines included in the DSM is described.

4.1.2. User parameters

A rather limited number of fixed input parameters is sufficient to define the problem of designing an ELV. These can be divided in few main categories:

Environment:

- Planet: defines radius R_P, rotation velocity Ω_P, planetary constant K_P, and additional gravitational harmonics J_2, J_3, J_4. Data for all solar system planets and the Moon are included in the model, and custom planetary data can be defined by the user, allowing design of launchers for any arbitrary celestial body.
- Atmosphere: defines temperature, pressure, density and sound velocity as a function of altitude. In light of the potential interest for future sample return missions, an exponential atmosphere for Mars was added to the standard Earth's US '76 atmosphere, and the user can provide a custom atmospheric data file.

Payload:

- Nominal payload mass M_{PL} and Centre of Gravity (CoG) position $X_{CoG,PL}$, given in meters from its bottom plane. If the trajectory is being optimized within a MDA or one of the objectives of the MDO process is the payload mass (see below the section on the optimization objectives), M_{PL} is multiplied by the Payload Scaling Factor (PLSF), decided by the optimizer within the user defined bounds.
- Length L_{PL} and diameter D_{PL}, used in the geometry module for the definition of the Payload Fairing (PLF) dimensions.
- Maximum axial acceleration ($n_{ax,max,PL}$) and heat flux without PLF ($q_{heat,max,PL}$), both used in the Weights module for the vehicle sizing and in the Trajectory module as path constraints threshold. $q_{heat,max,PL}$ is also used as triggering event for the payload fairing jettison.

Mission:

- Launch site, given in terms of longitude (λ_{launch}), declination (δ_{launch}) and altitude above the average surface level (H_{launch}). Data for predefined launch sites can be used, or the user can specify custom launch coordinates.
- Target orbit, given in terms of semiaxis, eccentricity and inclination. Longitude of pericentre (ω) and Right Ascension of the Ascending Node (RAAN or Ω) are not considered, since ω and/or Ω targeting requirements have very limited effect on the system design. Besides the target orbital parameters (a_T,e_T,i_T), tolerances (a_{tol},e_{tol},i_{tol})and normalization factors (a_{norm},e_{norm},i_{norm}) have to be provided.

Programmatics:

- Foreseen number of launches ($N_{launches}$) and number of years of operations ($N_{years,ops}$), important parameters affecting the production costs (learning curve) and operations costs (launch rate).
- Cost factors f_0 (system engineering complexity), f_1 (design complexity), f_3 (team expertise), f_7 (industrial structure), and f_8 (development country), which all affect in different proportions the LCC.

Aerodynamics settings:

- Number of Mach and AoA points ($N_{data,Mach}$, $N_{data,\alpha tot}$) in the aerodynamic database. Increasing the number of points allows for more accurate evaluation of the coefficients, but also increases the computational times for both database generation and interpolation. Due to the low accuracy of the aerodynamic analysis (see **Section 4.4**), a rather coarse grid is in general sufficient, typically with 10 to 20 Mach nodes and 3 to 5 AoA nodes.

Trajectory settings (see **Paragraph 4.6** for more details)**:**

- Integration settings, such as integrator type, tolerances and time step.
- Launch pad clearing altitude (H_{clear}), defining the instant of pitch-over manoeuvre start.
- Atmosphere interface altitude H_{atmo}, which separates the atmospheric and exoatmospheric parts of the flight. The transition can also be given in terms of jettison events (e.g. atmospheric until first stage or payload fairing jettison).
- Maximum total AoA in the atmospheric phase ($\alpha_{tot,max}$), which can be used to arbitrarily limit the aerodynamic loads below a certain threshold.
- Maximum deviation of pitch and yaw angles from the flight path and flight heading angles during atmospheric and exoatmospheric flights ($\Delta\psi_{atmo}$, $\Delta\psi_{exo}$, $\Delta\theta_{atmo}$, $\Delta\theta_{exo}$), defining the pitch and yaw optimization boundaries. Gravity turn conditions throughout the whole atmospheric phase can be imposed by setting $\Delta\psi_{atmo}=\Delta\theta_{atmo}=0$.
- Number of controls discretization nodes per each flight phase (i.e. stage or boosters set's flight), in particular yaw ($N_{nodes,\psi,j}$), pitch ($N_{nodes,\theta,j}$) and thrust throttle in case of liquid throttleable engines ($N_{nodes,\delta th,j}$). For this latter case, the minimum throttle level $\delta_{th,min}$ also needs to be defined, or is assumed to be 50% by default.

- Activation of the coast phases between one stage and the following (***LSC$_j$***) and of the circularization burn option (***CB***) for upper stages with liquid restartable engines.

4.1.3. Optimization variables

Even though only classical, unmanned, expendable launchers were taken into account, the number of optimization variables is considerable if the most generic optimization problem is considered. In fact, the design environment was developed trying to allow modelling of all ELVs currently operating or in development, resulting in a wide set of trade-offs at both system and subsystem level. The outcome is a large optimization variables vector, which can be divided in three main parts:

1) $\mathbf{N_{SDV}}$ system level design variables, with a maximum $\mathbf{N_{SDV}=14}$.
2) $\mathbf{N_{CDV}}\cdot(N_{stages}+N_{bsets})$ component level design variables which are repeated for each component (stage or set of boosters) included in the current architecture. The maximum number of variables per each component is $\mathbf{N_{CDV}=26}$.
3) $\mathbf{N_{TDV}}$ trajectory design variables, where $\mathbf{N_{TDV}}$ varies according to the number of flight phases (i.e. number of considered stages and booster sets) and number of control discretization nodes. As an example, typical trajectory optimization problems for Ariane 5 and VEGA consist respectively in $\mathbf{N_{TDV}=11}$ and $\mathbf{N_{TDV}=15}$ variables.

Since the component level design variables are repeated for each stage and for each boosters set included in the launcher's architecture, the maximum theoretical number of design variables (hence excluding trajectory) is of ***170 variables*** for a 4 stage launcher with two sets of different boosters. However, note that as the size of the MDO problem increases, the global search becomes more and more difficult. Only by constraining the design of the launch vehicle it is therefore possible to obtain good optimized results, as will be shown in **Chapter 8**, with typical problem sizes in the order of less than ten to a few tens of variables.

The most generic optimization variables vector for V1 models can be written as follows:

$$
\begin{cases}
X = \{X_{SDV}, X_{CDV,1}, ..., X_{CDV,N}, X_{TDV}\} \\
X_{SDV} = \{N_{stages}, Conf, CCB, N_{boosters}, SET, SBT, n_{ax,\max}, q_{dyn,\max}, q_{heat,\max}, RSC, SSM, SM_{PLF}, RL, LPM\} \\
X_{CDV,j} = \{OTS, OTS_{ID}, N_{eng}, Prop, Feed, M_P, T^*, \alpha_P, H^*, p_{cc}, A_{cc}/A_t, LCE, N_{eng,tests}, N_{eng,out}, TOE, \text{Nozzle}, \delta_{\text{div}}, \\
\qquad \text{TVC}, \delta_{\text{TVC}}, ThC, D_{const}, L/D, TT, TA, SM\} \\
X_{TDV} = \{PLSF, \Delta\theta_{PO}, t_{PO}, t_{PO,dec}, \Delta\psi_{PO}, \theta_j, \psi_j, \xi_{BTL}, \Delta\theta_{BTL,i}, \theta_{BTL,f}, \delta_{th,j}, \delta_{SP,j}, T^*_{\min,j}, t_{switch,j}, \Delta_{CI}, \Delta t_{CI}, \Delta_{BI}, \Delta t_{BI}, \Delta t_{LSCj}, \Delta t_{CB}\}
\end{cases}
$$

X contains continuous variables as well as integer and boolean variables, as specified below. For local refinement processes however, only the continuous design variables can be optimized, whereas all other must be frozen either a priori by the user or from previous global optimization runs.

The trajectory optimization variables vector $\mathbf{X_{TDV}}$ will be described in **Paragraph 4.6.2**, whereas details are given here for the launcher's multidisciplinary design variables vectors $\mathbf{X_{SDV}}$ and $\mathbf{X_{CDV,j}}$. For better understanding, the variables in $\mathbf{X_{SDV}}$ are presented in three groups (launcher architecture, trajectory loads, system design features) and the variables in $\mathbf{X_{CDV,j}}$ in five groups (propulsion system architecture, propulsion system design, geometry, structural layout). Note how the number of optimization variables related to propulsion largely exceeds that of other subsystems, confirming the primary role of the propulsion decision on the system design of launch vehicles.

<u>Launcher architecture:</u>

- ***N_{stages}***: integer variable defining the number of stages in the core, which may take any value from 1 for Single-Stage-To-Orbit (SSTO) systems to 4 for VEGA-like inline launchers. Note that a distinction is made in the models between ***N_{stages}-1*** lower stages and the upper stage.
- ***CCB (Common Core Boosters)***: boolean variable activating the CCB configuration, for which the design of the core stage is exactly replicated in two laterally arranged boosters (e.g. Delta IV Heavy, Falcon 9 Heavy).
  If ***CCB=true***, ***Conf*** and ***$N_{boosters}$*** variables described below lose significance.
- ***Conf*** and ***$N_{boosters}$***: the combination of these two integer variables defines the booster configuration, if ***CCB=false***. Figure 12 shows all possible configuration considered within the implemented models:
  - ***Conf=0***: no boosters (in line configuration) → ***$N_{boosters}$*** has no meaning.
  - ***Conf=1***: one set of boosters, with circumferential arrangement and equal angular spacing → ***$N_{boosters}$*** may take any integer value allowed by the user, between 2 and 9 (configurations with more than 4 boosters are not shown in the figure). The maximum number of boosters for an existing launcher is 9 for the Delta II 7900 series.
  - ***Conf=2***: one set of boosters, side arrangement → ***$N_{boosters}$=4*** or ***$N_{boosters}$=8*** for 1/2 boosters on each of the 4 sides.
  - ***Conf=3***: two sets of boosters of different type → ***$N_{boosters}$=4*** or ***6*** or ***8*** for 2+2 (circumferential), 2+4 (circumferential + side) or 4+4 (circumferential) configurations.

- ***SET (Single Engine Type)***: boolean variable activating the use of the upper stage's Liquid Rocket Engine (LRE, this option is incompatible with solid upper stages) for all lower stages, as well as for the boosters if used in combination with CCB option, as for example in the Falcon 9 and Falcon 9 Heavy launchers. If SET=true, only the upper stage's LRE is designed or selected from an available database of Off-The-Shelf (OTS) engines. The same engine is then duplicated on all the lower stages, the only modification being represented by the nozzle optimal altitude, and hence expansion ratio and resulting parameters (length, diameter, mass, thrust, specific impulse). Besides, the number of engines in the lower stages is not decided by the related optimization variable N_{eng} (see below), but is derived from the required total thrust level for each lower stage. Note that both CCB and SET design solutions have the purpose of cutting development costs, by exploiting the modular approach already mentioned in the introduction.
- ***SBT (Single Boosters Type)***: boolean variable activating the use of the same boosters design for both the first and second set of boosters (i.e. can be active only for Conf=3), the only modification being again the nozzle optimal altitude. The first set of boosters is ignited on the ground, whereas the second set is air-lit, with the advantage of reducing the dynamic pressure peak that may prevent from using a large number of boosters. This is for example the approach taken with the Atlas V when 9 boosters are strapped around the core, of which only 6 are ignited at take-off.

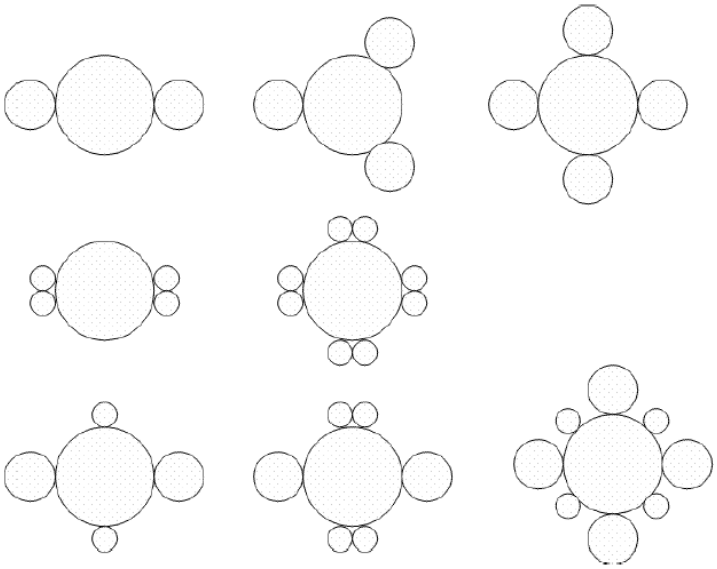


Figure 12: Implemented booster configurations. First, second and third line from the top represent Conf=1, Conf=2 and Conf=3

Note that the four variables N_{stages}, CCB, Conf and $N_{boosters}$ fully describe the launch vehicle's architecture in terms of number and collocation of stages and booster sets. A large number of design parameters will then be repeated for each of the stages and booster sets as well as for the payload fairing. Although models for these components are mostly identical, there are slight differences which will be described throughout this chapter. For this reason, the following subscripts will be used for components identification: 1) PLF for payload fairing, 2) LS_j for j-th lower stage, 3) US for the upper stage, and 4) BS_j for j-th boosters set.

Note also that, in case of CCB, SET and/or SBT, the regular system level MDA cycle needs to be adapted. In particular:

- **CCB**:
  - Propulsion: no analysis performed for boosters, the entire propulsion system is cloned from the core stage
  - Geometry, Weights, Costs, Reliability: regular analyses are performed, with all input data cloned from the core stage → identical outputs, except for the substitution of the front interstage with an ogival nose and common development costs.
- **SET**:
  - Propulsion: a regular analysis is performed for each of the lower stages, but all input data are cloned from the upper stage, except for the nozzle optimal altitude H^*, the nominal thrust level T^*, the number of engines N_{eng} and the propellant mass M_{prop}. Through this analysis cycle, the nominal thrust per each engine T^*_{eng} is determined and the required number of engines is then obtained as $N_{eng} = ceil\left(\frac{T^*}{T^*_{eng}}\right)$. Although the same engine is used, most of the output performance parameters are different due to the different expansion ratio. Similarly, geometrical data for a single engine are identical to the upper stage except for the nozzle's expansion ratio and length, but the propellant mass is different, hence the geometry of the tanks is different.

- Geometry, Weights, Costs, Reliability: regular analyses, except for the common engine development costs.

- **SBT**:
  - Propulsion: a regular analysis is performed for the second set of boosters, but all input data are cloned from the first set, except for H^*. As opposed to the SET option, the two sets of boosters are absolutely identical in terms of number of engines (determined by the related optimization variable), engines geometry and tanks/grain geometry, with the only exception of the nozzle's length and expansion ratio.
  - Geometry, Weights, Costs and Reliability: regular analyses are performed, with all input data except for H^* cloned from the first boosters set as in the Propulsion analysis.

    → Identical outputs, except for the engine's nozzle and the common development costs.

Trajectory loads:

- $n_{ax,max}$, $q_{dyn,max}$, $q_{heat,max}$: already introduced in **Paragraph 4.1.1** within the description of the IDF approach to avoid Weights-Trajectory iterations.

System design features:

- ***RSC (Re-Start Capability)***: boolean variable activating the restart capability for the upper stage engine, determining the possibility to perform an orbital circularization burn. Note that RSC applies only for new design Liquid Propellant (LP) upper stages, since Solid Propellant (SP) rockets cannot be re-ignited and each OTS engine has a predefined RSC.
- ***SSM (Structural Safety Margin)***: margin assumed for all launch vehicle structures, in theory affecting both the inert masses and the failure probability. Note however that the developed risk models (**Section 4.7**) are very much insensitive to the structural reliability, which is extremely high for any reasonable structural margin, hence preventing from obtaining significant performance vs. reliability trade-offs. For this reason, the SSM loses relevance as optimization variable, but may be varied by the user to define different structural margin philosophies.
- ***SM_{PLF} (Structural Material, PLF)***: integer variable defining the type of material used for the PLF. Due to the low fidelity of the weights estimation models, the actual properties of the different materials cannot be used as inputs in a physical analysis of the structure. However, a corrective factor of the WERs is defined to represent "advanced" materials, such as composite sandwiches or Al-Li tanks. SM_{PLF} can therefore take values 1 for classical materials (no corrective factor) or 2 for advanced materials. As for SSM on reliability, SM_{PLF} has a very limited impact on the LCC within the developed cost models, so that performance vs. cost trade-offs cannot be easily recognized by the optimizer.
- ***RL (Redundancy Level)***: integer variable defining the redundancy level for the avionics and power subsystems, located in the upper stage's Vehicle Equipment Bay (VEB). RL may take value 1 for critical redundancies only approach, with lower reliability, cost and mass, or 2 for full redundancy approach, with maximum reliability but larger cost and mass.
- ***LPM (Launcher Processing Mode)***: integer variable that defines the processing mode of the launcher, either vertical (e.g. European vehicles) or horizontal (e.g. Russian vehicles). This influences both the inert masses, with heavier structures required for horizontal integration and successive lifting on pad, and operations costs, which are clearly larger for vertically processed launchers. Although the CERs for operations included a correction for integration mode, it was not possible to include reliable factors in the WERs for the same purpose.

Four of the system design features described here (***SSM***, ***SM_{PLF}***, ***RL*** and ***SSM***) were introduced to allow for performance vs. cost vs. reliability trade-offs, which are typical of the design process of ELVs. Nevertheless, results clearly indicate that simple engineering level disciplinary analyses cannot ensure the required modelling fidelity, especially but not only with respect to costs and reliability, which would be necessary to ensure fair results in such trade-offs. Much more detailed methodologies would be required for this purpose, for example including high fidelity structural analysis and low level non structural masses breakdown, as well as bottom-up cost estimation processes and better statistical determination of the failure rates. These aspects are out of the scope of the present investigation, and are left as a future development, however the above mentioned variables are kept in the MDO environment, allowing for a partial representation of the related design decisions.

Propulsion system architecture:

- ***OTS (Off-The-Shelf)***: boolean variable determining the selection of an OTS engine from an available database or of a new design propulsion system. The first choice results in fixed performances which cannot be tailored to the vehicle's requirements, but also in large saving in terms of development costs.
- ***ID_{OTS}***: integer variable defining the ID of the OTS engine to be used. If a new design is selected, ID_{OTS} does not apply. In case of OTS instead, the user may select a given engine, allow choice between several allowed types, or define a thrust range triggering a search in the database for compatible engines.

- N_{eng}: integer variable defining the number of engines for a given stage or booster, which may take any value between 1 and 9[20]. This variable applies only for LP stages/boosters, whereas 1 engine is assumed for Solid Rocket Motors (SRM). As already mentioned above, if SET=true the number of engines is directly computed from the required thrust level and the thrust of the upper stage's engine.
- ***Prop***: integer variable defining the type of propellants used for new design rocket engines. The following alternatives are considered:
  - ***Prop=1***: liquid cryogenic engine, burning Liquid Hydrogen (LH_2) and Liquid Oxygen (LOx).
  - ***Prop=2***: liquid cryo-storable engine, burning Rocket Propellant One (RP1) and LOx.
  - ***Prop=3***: liquid storable engine, burning Mono-Metil Hydrazine (MMH) and Nitrogen Tetroxide (N_2O_4).
  - ***Prop=4***: solid propellant engine with a fixed Ammonium Perclorate (AP), Hydroxyl-Terminated Poli Butadiene (HTPB) and Aluminium powder formulation.

  Other combinations of oxidizer and fuel or different SP formulations are not considered within V1 models.
- ***Feed***: integer variable defining the type of feed system used for new design LP systems. The following alternatives are considered:
  - ***Feed=1***: Pressure-fed system.
  - ***Feed=2***: Turbopump system with gas generator cycle (open cycle).
  - ***Feed=3***: Turbopump system with expander cycle (close cycle).
  - ***Feed=4***: Turbopump system with staged combustion cycle (close cycle).

  Other engine cycles exist but were not included, due to their limited application and consequent lack of a database sufficiently large to obtain reliable historical fits of specific impulse losses and inert weights.

Propulsion system design:

- M_P: continuous variable defining the mass of propellant of a stage/booster, intended as the *usable* fraction only. The lower and upper bounds can either be specified by the user or derived from a first guess sizing procedure which allocates propellant mass M_P and thrust level T^* to the different stages and boosters on the basis of rocket equation, payload-to-launch mass ratio and mission ΔV[21].
- T^*: continuous variable defining the *nominal* thrust of new design LP or SP engines, intended as *the maximum thrust level (full throttle for LP, peak thrust profile for SP) assuming the external pressure to correspond to the nozzle exhaust pressure*. This condition corresponds to the nozzle design conditions, which are indicated with the superscript * and are achieved at an altitude ***H**** depending on the expansion ratio of the nozzle.
- α_P: continuous variable defining the mixture ratio of new design LP systems, assuming that the engine's and tank's mixture ratios coincide. The bounds can either be specified by the user or taken from default ranges which depend on the propellant combination, as reported in **Table 2**.
- H^*: continuous variable defining the nozzle design point for a new design LP engine, also referred to as optimal nozzle expansion altitude. This parameter is directly related to the expansion ratio of the nozzle ε, and its default bounds are reported in **Table 2**.
- p_{cc}: continuous variable defining the chamber pressure of new design LP or SP engines (default bounds in **Table 2**).
- A_{cc}/A_t: continuous variable defining the combustion chamber's contraction ratio. Note that although this parameters affects both the theoretical specific impulse and the engine's dimensions, this impact is rather limited and the relevance of this variable is thus definitely subordinate to the mixture ratio, expansion ratio or chamber pressure.
- ***Nozzle***: integer variable discriminating between *bell* and *cone* divergent section's shape, the former resulting in lower I_{sp} losses and shorter nozzle in front of higher development and manufacture costs.
- δ_{div}: continuous variable determining the cone divergent angle or the cone-equivalent divergent angle in case of bell nozzles.
- ***TVC (Thrust Vector Control)***: integer variable defining the type of TVC: if TVC is installed, either its actuation type can be either hydraulic or electro-mechanic. Hydraulic actuation is in general advantageous in terms of costs and is suitable also for very large nozzles, whereas electric systems are definitely lighter when applicable, up to medium to large size nozzles (e.g. VEGA's P-80 first stage is the largest current electro-mechanically actuated TVC system).
- δ_{TVC}: continuous variable determining the maximum deflection angle allowed by the TVC system, with a trade-off between control authority and overall mass of the system.
- ***ThC (Throttle Capability)***: boolean variable activating the throttle capability of a LP engine. A throttle capability down to 50% is allowed by default in LP engines with ThC=true, but may be modified by the user to different minimum throttle percentages.

[20] 9 corresponds to the largest number of engines in a single stage for any existing launcher (Falcon 9 first stage).

[21] For a complete description of the first guess staging definition procedure, see PRESTIGE Technical Note D6 (**[A6]**), Chapter 8.

Propulsion system costs and reliability:

- ***LCE (Low Cost Engine)***: boolean variable activating the option of designing a new LP engine with lower performance, cost and failure rate than traditional high performance LREs. This variable was introduced to account for design choices such as the Rocketdyne's RS-68 engine, which exploits a drastic reduction in complexity and number of parts to increase reliability and reduce development and production costs, in front of a non excessive reduction in shear performance.
- $N_{eng,tests}$: integer variable defining the number of hot-fire tests to be conducted for the development of a new LP engine. $N_{eng,tests}$ was added to the optimization variables vector because of its relevance in the cost models, as outlined in **Section 4.7** and detailed in **[A4]**. The number of engine tests is in fact the main driver for LREs development costs, but its reduction must be carefully weighted against the risk of increased failure rates.
- $N_{eng,out}$: integer variable defining the number of engines in a given stage/booster which can fail without hindering the mission's success. This is typically either 0 for stages with one or few engines, or 1 for stages with multiple engines. Although implementing engine-out capability requires an oversized overall thrust and more complex integration, this design choice drastically increases system reliability, contributes to ensuring high mission success rates - as the Saturn V experience proves - and is one of the core philosophies at the basis of the new SpaceX success.
- ***TOE (Thrust Oversized Engine)***: boolean variable activating the option to install a new or OTS engine with 5% higher thrust with respect to the actual mission requirement. This is actually the step before the engine-out capability, as implemented for stages with fewer engines, and ensures an increased reliability due to the non full throttle operations as well as a thrust margin for contingency situations.

Note that the same considerations apply for these four cost/reliability related variables as for the system design features mentioned above (***SSM***, SM_{PLF}, ***RL*** and ***SSM***).

Geometry:

Several geometric parameters of the launch vehicles cannot be optimized in the V1 models, such as for example the length-over-diameter of the nose ogive of the PLF and of the boosters, or the interstage/aft skirt cone angles. In fact, although they might have rather important local effects, especially on aerodynamic forces and heat flows, their impact on the system performance is limited. Hence, only two geometric parameters are used as design optimization variables:

- D_{const}: boolean variable defining a stage which has the same diameter as the stage, or PLF, above. If D_{const}=true, the interstage is therefore cylindrical, otherwise it is conical with a fixed cone angle. This variable only applies for stages, not for booster sets.
- ***L/D***: continuous variable which is active for boosters and for stages with D_{const}=false, defining the overall lengh-over-diameter of the stage. Default bounds are again shown in **Table 2**.

Variable	Case	Lower Bound	Upper Bound
α_P	LOx-LH_2	4.50	6.71
α_P	LOx-RP1	1.35	2.99
α_P	N_2O_4-MMH	1.49	2.94
p_{cc}	SP	50 bar	120 bar
p_{cc}	LP pressure fed	6 bar	15 bar
p_{cc}	LP gas generator	30 bar	120 bar
p_{cc}	LP expander cycle	30 bar	70 bar
A_{cc}/A_t	LP staged combustion	80 bar	270 bar
H^*	LP/SP systems	2.0	5.0
H^*	Boosters	1.0 km	5.0 km
H^*	First stages	2.0 km	15.0 km
L/D	Upper stages	10.0 km	50.0 km
L/D	Boosters	3.0	11.0
L/D	First stages	2.0	6.0
δ_{div}	Upper stages	1.0	4.0
δ_{TVC}	-	8 deg	20 deg

Table 2: Default upper and lower bounds of the continuous design optimization variables for the conceptual MDO of ELV.

Structural layout:

In light of the very low level of fidelity of the engineering models for weights estimation, a very limited number of design variables were introduced for the structural layout of stages and boosters. In general, 3 parameters at most can be optimized for each stage/booster:

- ***TT (Tanks Type)***: integer variable defining the type of propellant tanks in LP systems, either separate tanks for oxidizer and fuel or a single tank with common bulkhead. The latter choice determines a lower inert mass reflected in a corrective factor of the main structure's WER as well as in higher costs as shown in the cost models section.

- ***TA (Tanks Arrangement)***: integer variable defining whether the oxidizer or the fuel is places aft in LP systems, activating a weights vs. controllability trade-off, due to the lower mass but larger instability (CoG closer to the thrust pivot point) resulting from arranging the heavier oxidizer tank behind the lighter fuel tank.
- ***SM (Structural Material)***: same as SM_{PLF} for each stage/booster.

4.1.4. Optimization constraints

As previously mentioned, optimization constraints related to the physical consistency and engineering feasibility of the launch vehicle are computed at each discipline of the design cycle. When the design results to be unfeasible, the MDA is stopped[22] to avoid spending computational effort on unrealistic design, and very large values are assigned to the objectives. However, if the analysis of all design disciplines is carried out without violating any constraint but one or more of the trajectory constraints are not satisfied, the MDA process is completed with the cost and reliability analysis. In fact, such a solution represents a feasible design point, and all the design objectives should be evaluated in order to better guide the optimizer towards the best region of the search space.

As for optimization variables, the constraints can be divided in three main groups:

1) $\mathbf{N_{SDC}}$ system level design constraints, with $\mathbf{N_{SDC}=8}$.
2) $\mathbf{N_{CDC}\cdot(N_{stages}+N_{bsets})}$ component level design constraints, repeated for each component with $\mathbf{N_{CDC}=5}$.
3) $\mathbf{N_{TDC}}$ trajectory design constraints, where $\mathbf{N_{TDC}}$ varies according to the number of stages and booster sets as well as the number of launch site heading constraints. As an example, Ariane 5 and VEGA launches from Kourou consist respectively in $N_{TDC}=12$ and $N_{TDC}=13$ constraints.

Table 3 introduces all the component level and system design constraints, whereas those related to the trajectory are described within the trajectory optimization problem formulation in **Paragraph 4.6.2**. As for the optimization variables, all non continuous design constraints are deactivated during local refinements, in order to ensure a smooth problem for the gradient-based algorithm.

Discipline	Constraint	Type	Applicability	LB	UB
Propulsion	OTS thrust range	Boolean	Each stage/boosters set (OTS only)	-1	1
Propulsion	CEA execution failure	Boolean	Each stage/boosters set	-∞	0
Propulsion	Number of engines	Integer	Each stage/boosters set (SET only)	-∞	0
Geometry	Geometric interference	Double	First stage and each boosters set	-∞	0
Geometry	Engine clearance	Double	Each stage/boosters set	-∞	0
Geometry	Length-over-diameter	Double	Launch vehicle	0	1
Geometry	PLF geometric interference	Double	Payload Fairing	-∞	0
Geometry	Geometry execution failure	Boolean	System analysis	-∞	0
Aerodynamics	DATCOM execution failure	Boolean	System analysis	-∞	0
Weights	Weights execution failure	Boolean	System analysis	-∞	0
Weights	Lift-off thrust-to-weight	Double	Launch vehicle	0	1
Costs	Costs execution failure	Boolean	System analysis	-∞	0
Risks	Risks execution failure	Boolean	System analysis	-∞	0

Table 3: List of ELV design constraints, including the involved discipline, the type of constraint, its applicability to a single component or to the whole launch vehicle, and the feasibility lower and upper bounds.

System design constraints can be divided in two main branches:

- ***Engineering feasibility constraints***: lift-off thrust-to-weight ratio $\boldsymbol{(T/W)_{LO}}$, core's total length-over-diameter ratio $\boldsymbol{(L/D)_{Core}}$ and PLF's geometric interference with the upper stage, as follows:

$$\begin{cases} (T/W)_{LO,min} < (T/W)_{LO} < (T/W)_{LO,max} \\ (L/D)_{Core,min} < (L/D)_{Core} < (L/D)_{Core,max} \\ D_{PLF} > K_{D_{PLF}} \cdot D_{US} \end{cases}$$

  where all bounds can be provided by the user, in accordance with the general user interactivity philosophy, and are also used to normalize the constraints values. Default bounds employed for T/W and L/D are [1.1 – 2.0] and [5.0 – 15.0], whereas $K_{DPLF}=0.95$ is used.
- ***Analysis failure constraints***: in all disciplinary analysis, output checking routines were implemented to determine whether the analysis was successful. In case of analysis failure, which may be related to the external codes execution

[22] Unless the MDA is part of a local refinement process, for which discontinuities in the model pose significant hindrance to the employed gradient-based NLP algorithm.

or to very particular occurrences in the ad hoc developed models, the MDA process is stopped, setting the related constraint as violated and hence determining very large violations of objectives.

The following component level constraints are instead defined, related only to the Propulsion and Geometry disciplines:

- ***Thrust range***: if the user imposes an OTS selection within a given thrust range, and no OTS engine is available in the database in such thrust range, a constraint violation is returned.
- ***Number of engines***: in case of SET design, the number of upper stage engines which are required to be installed in a lower stage to match its thrust requirement cannot exceed the maximum number of 9 engines.
- ***Geometric interference***: in case of parallel configurations with at least one set of boosters, the length of the boosters cannot exceed the length of the first stage (first stage's geometry interference). Similarly, the boosters must not interfere with each other, for example if the diameter is too large for the given number of boosters in the circumferential arrangement.
- ***Engine clearance:*** for all stages and booster sets, a constraint is imposed on the total diameter of the engines assembly, which cannot be larger than the external diameter of the stage/booster.

4.1.5. Optimization objectives

The MDO problem for the conceptual design of ELV with V1 models is formulated as:

$$\min_{X_i, i=1,\ldots,N_{\text{var}}} (F_1,\ldots,F_{N_{obj}}), \quad with \quad F_j = \sum_{k=1}^{4} \frac{w_{jk} \cdot O_k}{s_{jk}}; \quad and \quad \begin{cases} O_1 = GTOW \\ O_2 = CpL \\ O_3 = MSP \\ O_4 = PLSF \end{cases}$$

$$subject\ to\ \ LB_l \le G_l \le UB_l, \quad l = 1,\ldots,N_{cstr}$$

where N_{obj} is the number of desired optimization objectives, $X=\{X_1,\ldots,X_{Nvar}\}$ is the vector of optimization variables described in **Paragraph 4.1.3**, and $G=\{G_1,\ldots,G_{Ncstr}\}$ is the vector of optimization constraints in **Paragraph 4.1.4**. Note that although the multi-objective algorithms presented in **Section 3.2** are theoretically capable of solving problems with any number of objective functions, N_{obj} is to be practically limited to at most 3 figures, since larger objective spaces complicate the search and prevent from an intuitive visualization of the Pareto results.

The equation defining j-th cost function F_j is completely general, because any of the available output figures O_1 to O_4 can be used as a stand-alone objective, or more than one can be aggregated through the weight and scaling parameters w_{jk} and s_{jk}. Although w_{jk} and s_{jk} could be merged in a single number, they are kept separate to provide a more clear understanding of the weighting process.

The following four figures can be used to build up the objectives, including performance, cost and reliability outputs in order to allow for "design-to-cost" and "design-to-reliability" approaches in the MDO process:

- O_1***=GTOW*** (Gross Take-Off Weight, *min*):
- O_2***=CpL*** (Cost per Launch, *min*):
- O_3***=MSP*** (Mission Success Probability, *max*):
- O_4***=PLSF*** (Payload Scaling Factor, *max*): this performance figure is included among the possible objectives in order to allow for launch vehicle scaling studies. For example, it is possible to simultaneously minimize the total launcher's mass (O_2) and maximize the payload mass, obtaining Pareto fronts of design solutions with payload performance distributed within given bounds around the nominal value.

4.2. Propulsion

Reflecting its complexity and importance, the propulsion model is composed of a large number of smaller modules, which allow sizing specific components or evaluating specific performance indexes. The disciplinary analysis is repeated for the total number of propulsion systems to be included in the design, which is determined through the architectural variables ***Conf***, ***CCB***, N_{stages} and $N_{boosters}$. Specific analysis requirements determined by the activation of the ***CCB***, ***SET*** and ***SBT*** variables are also taken into account as already described in **Paragraph 4.1.3**.

For each propulsion system, the analysis flow is then represented in **Figure 13**, with the main difference being determined by the choice between an OTS and a new design engine. In case of OTS engine, an existing LRE is selected from the available database and all its specifications are loaded. In case of new design engine instead, the analysis procedure at the bottom of **Figure 13** is followed, determining the theoretical and corrected performances as well as the dimensions and weights of the LP or SP system being designed. In both cases, an analysis of the overall geometry and weights is then performed, computing the total length and diameter of the tanks/grain case as well as the weight of auxiliary propulsion system items, such as pressurization system and unused propellants. The next paragraphs present the main aspects related to each of the blocks represented in **Figure 13**. Note that the main source used for the development of the propulsion models

are the classical *"Rocket Propulsion Elements"* by Sutton and Biblarz (**[86]**) and NASA's design manual "Design of Liquid Propellant Rocket Engines" (**[87]**) and "Solid Rocket Motors Performance Analysis and Prediction" (**[88]**).

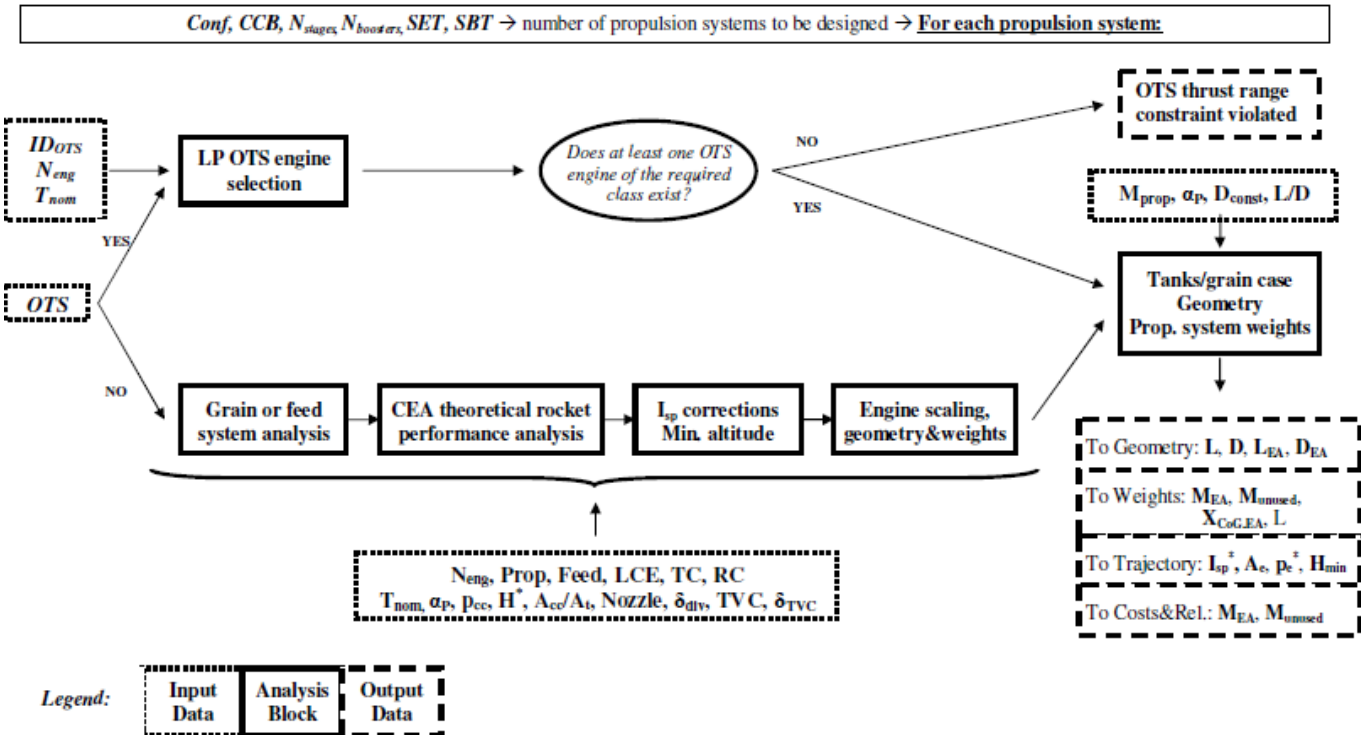


Figure 13: Analysis flow for the propulsion system of a given stage or boosters set.

4.2.1. Off-The-Shelf Liquid Rocket Engines database

In case of OTS option, one of the engines in the available database of LREs – described below - needs to be selected. In particular, the optimization variable ID_{OTS} determines this choice, through one of two selection methods which were implemented to account for different user requirements:

1. *Selection from a list of allowed engines*: this method allows to directly constrain the use of a preferred engine or to optimize the choice between two or more available engines.
2. *Selection on the basis of the thrust range*: this method allows requesting a desired thrust range, so that all engines in the database belonging to that interval can be chosen by the optimizer. If no such engines are available in the database, the OTS thrust range constraint is considered violated and the design solution is discarded.

The database, containing only LREs (see V2 models in **Paragraph 6.2.1** for the extension to SRMs), was collected including information from several US, Russian, European and Japanese companies. It has the dual purpose of providing engines for OTS design and of serving as a knowledge base to develop historical relations for several important parameters such as chamber pressure, specific impulse losses, dimensions and inert masses. This empirical methodology, though not extremely accurate especially with respect to dimensions and masses, was considered sufficiently reliable for conceptual level modelling. Note that only engines flight qualified in the last 25 years are included in the database, with the exception of the Space Shuttle Main Engines (SSME), so that both the resulting historical relations and the employed OTS are based on current state-of-art technologies and can therefore be considered realistic for future launcher applications.

The main sources used to compile the database are, in this order of priority[23]: the datasheets on each company's websites, Isakowitz's "International Reference Guide to Space Launch Systems" (**[89]**), and Astronautix[24] website. Moreover, cost and reliability data were taken from the TRANSCOST model (**[90]**), which will be further described in **Paragraph 4.7**. Additional web sources were analyzed but did not provide additional information with respect to the above sources, as well as a direct interrogation of the manufacturers conducted via email without success. The following 11 companies were considered, for a total of 34 LREs:

- United States:
  - Aerojet (AJ): www.aerojet.com/capabilities/spacelift.php

[23] Several contrasting numbers are often found in open literature and different web sources, hence a rigorous collection procedure was maintained, entrusting official company datasheets more than open literature and other web sources.
[24] www.astronautix.com

  - SpaceX (SX): www.spacex.com
  - Rocketdyne (RD): http://www.pw.utc.com
- Russia/Ukraine:
  - KB Khimavtomatiki (KBA): www.kbkha.ru
  - NPO Energomash (NPO): www.npoenergomash.ru/eng
  - Yuzhnoye (YZN): www.yuzhnoye.com
  - KB KhIMMASH (KBI)
  - Korolev (KRL): www.energia.ru/english/energia/launchers/engines.html
- Europe:
  - EADS: cs.astrium.eads.net/sp/launcher-propulsion/rocket-engines/index.html
  - Snecma (SN): www.snecma.com
- Japan: Mitsubishi Heavy Industries (MHI): www.mhi.co.jp/en/products/space index.html

All the collected information is reported in Table 4, including data from 34 liquid rocket engines belonging to 9 possible combinations of propellants and feed systems. Unfortunately, despite the large investigation effort, several data are still not available, for which "NA" is reported in the table. For this reason, the Propulsion analysis procedure (bottom branch in **Figure 13**) is executed also for OTS engine, using the input data specified in the database - required input data are fortunately available for all engines - instead of those which are defined by the optimization variables or user selections in case of new design engines. All the outputs required for the continuation of the MDA are therefore obtained. Then, if a given output is available in the database for the given engine, the computed value is substituted with the literature value, which is supposed to be more reliable. Similarly, the rarely available cost and reliability information are substituted to those computed in the Cost and Reliability models at the end of the MDA cycle.

Engine	Application	Man.	Propellants	α []	Feed	p_{cc} [bar]	ε []	$T_{nom,vac}$ [kN]	Restart	Throttle	$I_{sp,vac}$ [s]	D_e [m]	L_{engine} [m]	M [kg]
SSME	Shuttle orbiter	RD	LO_X-LH_2	6.03	SC	206.4	77.0	2278.8	no	yes	452.3	2.44	4.52	3526.2
RD-0120	Buran orbiter	KBA	LO_X-LH_2	6.00	SC	218.7	85.7	1813.1	no	yes	455.0	2.42	4.55	3450.0
LE-7A	H-II 1st stage	MHI	LO_X-LH_2	5.90	SC	121.0	51.9	1098.3	no	yes	440.0	NA	3.67	1980.0
RL10A-4	Atlas V US	RD	LO_X-LH_2	5.50	EC	42.1	85.0	99.2	yes	no	451.0	1.17	2.29	181.1
RL10B-2	Delta-IV US	RD	LO_X-LH_2	5.88	EC	32.1	285.0	110.0	yes	no	465.5	2.15	2.20	301.2
RD-0146	Angara US	KBA	LO_X-LH_2	6.00	EC	79.0	210.0	98.0	yes	no	463.2	1.25	2.20	502.0
Vinci	Ariane5 ESC-B	SN	LO_X-LH_2	5.80	EC	60.0	240.0	180.0	yes	no	465.0	2.20	4.20	336.0
LE-5B	H-II US	MHI	LO_X-LH_2	5.00	EC	36.3	110.0	137.3	yes	yes	449.0	NA	2.77	285.0
RS-68	Delta IV 1st stage	RD	LO_X-LH_2	5.97	GG	102.6	21.5	3371.8	no	yes	409.0	2.44	5.18	6747.6
J2-X	SLS US	RD	LO_X-LH_2	5.50	GG	92.2	92.0	1307.8	yes	no	448.0	3.05	4.70	2472.1
Vulcain-2	Ariane5 EPC	SN	LO_X-LH_2	6.10	GG	115.0	59.5	1340.0	no	no	431.0	2.10	3.45	2100.0
HM-7B	Ariane5 ESC-A	SN	LO_X-LH_2	5.00	GG	37.0	83.1	64.8	no	no	446.0	0.99	2.01	165.0
AJ26-59	K-1 1st stage	AJ	LO_X-RP1	2.59	SC	145.4	27.0	1683.0	yes	yes	331.3	1.50	3.71	1222.0
RD-180	Atlas 1st stage	NPO	LO_X-Kerosene	2.72	SC	266.7	36.9	4158.0	no	yes	337.8	NA	3.60	5400.0
RD-171M	Zenit 1st stage	NPO	LO_X-Kerosene	2.63	SC	245.2	37.0	7904.1	no	yes	337.0	NA	4.15	9300.0
RD-191	Angara 1st stage	NPO	LO_X-Kerosene	2.60	SC	263.4	37.0	2089.0	no	yes	337.0	1.45	4.00	2200.0
RD-120	Zenit 2nd stage	NPO	LO_X-Kerosene	2.58	SC	178.1	106.0	912.0	no	yes	350.0	1.95	3.87	1125.0
11D58MF	Zenit 3SL US	KRL	LO_X-Kerosene	2.48	SC	77.4	280.0	83.4	yes	no	356.0	1.17	2.27	230.0
RS-27A	Delta II 1st stage	RD	LO_X-RP1	2.25	GG	48.3	12.0	1054.2	no	no	302.0	1.70	3.78	1146.7
Merlin-1C	Falcon 9 1st stage	SX	LO_X-RP1	2.17	GG	47.0	14.5	617.0	no	yes	304.0	1.25	NA	760.0
Merlin-1V	Falcon 9 US	SX	LO_X-RP1	2.17	GG	47.0	117.0	411.5	yes	yes	342.0	3.55	NA	NA
RD-107A	Sojuz Boosters	NPO	LO_X-Kerosene	2.47	GG	60.0	18.9	1020.0	no	yes	320.2	NA	2.58	1090.0
RD-108A	Sojuz Core	NPO	LO_X-Kerosene	2.47	GG	54.4	18.9	886.5	no	yes	320.6	NA	2.87	1075.0
Kestrel	Falcon 1 US	SX	LO_X-RP1	2.35	PF	10.3	60.0	27.8	yes	yes	325.0	NA	NA	52.0
RD-275M	Proton 1st stage	NPO	N_2O_4-UDMH	2.67	SC	165.2	26.0	1831.9	no	no	315.8	1.50	3.05	1070.0
S5.98	US Breeze	KBI	N_2O_4-UDMH	2.00	SC	97.0	153.8	19.6	yes	no	325.5	NA	NA	95.0
S5.92	US Fregat	KBI	N_2O_4-UDMH	2.00	GG	98.0	153.8	196	yes	yes	327.0	0.84	1.03	75.0
LR-91-5	Titan II 2nd stage	AJ	N_2O_4-UDMH	1.80	GG	57.0	49.0	440.0	no	no	316.0	1.68	2.80	500.0
RD-861K	3rd Tsyklon-4	YZN	N_2O_4-UDMH	2.41	GG	88.8	115.8	77.6	yes	yes	330.0	1.53	1.56	194.0
RD-869	VEGA AVUM	YZN	N_2O_4-UDMH	2.00	PF	20.0	81.5	2.5	yes	no	314.7	0.30	NA	NA
RS-72	Ariane 5 EPS evol.	EADS	N_2O_4-MMH	1.90	GG	60.0	300.0	55.4	yes	no	340.0	1.30	2.29	138.0
Aestus	Ariane 5 EPS	EADS	N_2O_4-MMH	2.05	PF	11.0	84.0	29.4	yes	no	324.0	1.31	2.18	111.0
AJ10-118-K	Delta-II, 2nd stage	AJ	N_2O_4-MMH	1.80	PF	9.0	65.0	43.7	yes	no	319.2	1.70	NA	127.4
AJ10-190	Shuttle OMS	AJ	N_2O_4-MMH	1.65	PF	8.6	55.0	26.7	yes	no	316.0	NA	NA	118.0

Table 4: Database of LRE collected both as OTS table look-up and knowledge base for historical based relationships development

4.2.2. Grain and feed system analyses

In order to assess the theoretical performance of a solid or liquid rocket engine with simplified chemical equilibrium methods such as CEA, only few design parameters need to be defined. In particular, these are the chemical formulation of

the liquid or solid propellants, the chamber pressure p_{cc}, the expansion ratio ε or the exhaust pressure p_e, and the mixture ratio α_P in case of LP. The propellants and exhaust pressure p_e are directly obtained from the mapping of the related optimization variables, whereas p_{cc} and α_P depend on the propellants and - for LP - on the feed system. Moreover, for SP, the grain shape selection determines the thrust profile as well as the motor's performances. For this reason, SP grain and LP feed system analyses were developed, but were kept extremely simple in V1, as described in the next two subparagraphs.

4.2.2.1. SP grain analysis

In case of SRMs, grain design plays an important role in the definition of the burning area over time, hence of the chamber pressure and net thrust profiles, as well as on the volumetric efficiency and geometric dimensions of the motor. However, grain shape design is rather complex and its introduction in the design cycle requires using several additional design variables. For example, a generic star grain section can be described by 7 parameters (**[91]**, **[92]**), and similar relations are available for other shapes. In order to avoid these complications in the first modelling step, three simple thrust profiles were defined, which can be either selected by the user of optimized, reflecting the most typical configurations in SP motors for launch vehicles:

- ***Constant thrust***: representative of neutral star grains, for which pressure variations are usually below 15% of the nominal value (**[86]**), with the nominal thrust T^* as the only optimizable parameter. This profile is typically used for motors ignited after max Q_{dyn} event and for which the acceleration at the end of the burn is not a dimensioning case.
- ***Two-level thrust***: constituted of a first phase with maximum thrust and a second with lower thrust level. Optimization variables are thus the two thrust levels (T^* and T^*_{min}) and the switch time (t_{switch}). This profile, which can be approximately obtained with multiple segments of different shapes, is mostly used in atmospheric boosters or stages (e.g. Ariane 5's P241 or VEGA's P80), for which an in-flight thrust reduction allows to more efficiently satisfy the max Q_{dyn} constraint.
- ***Regressing thrust***: linear thrust profile, decreasing from ignition to burnout and hence with the initial and final thrust levels (T^* and T^*_{min}) as optimization variables. This profile is mostly used in second or third stages (e.g. VEGA's Z23), for which the decreasing thrust allows to avoid peaks in axial acceleration at the end of burn.

Examples from the 3 stages of VEGA are reported in **Figure 14**. The values of the optimization parameters were obtained by minimizing the sum of square errors on the experimental data subject to the constraint of matching the total impulse of the motor. Data points after the actual jettison of the stage were removed, since they do not contribute to the thrust of the launcher and introduce errors in the fitting due to the very low thrust values. Although the approximations look rather rough, the introduction of two level and linear thrust profiles allow reproducing important design features, such as the reduction in thrust to match both max Q_{dyn} and max n_{ax} constraints.

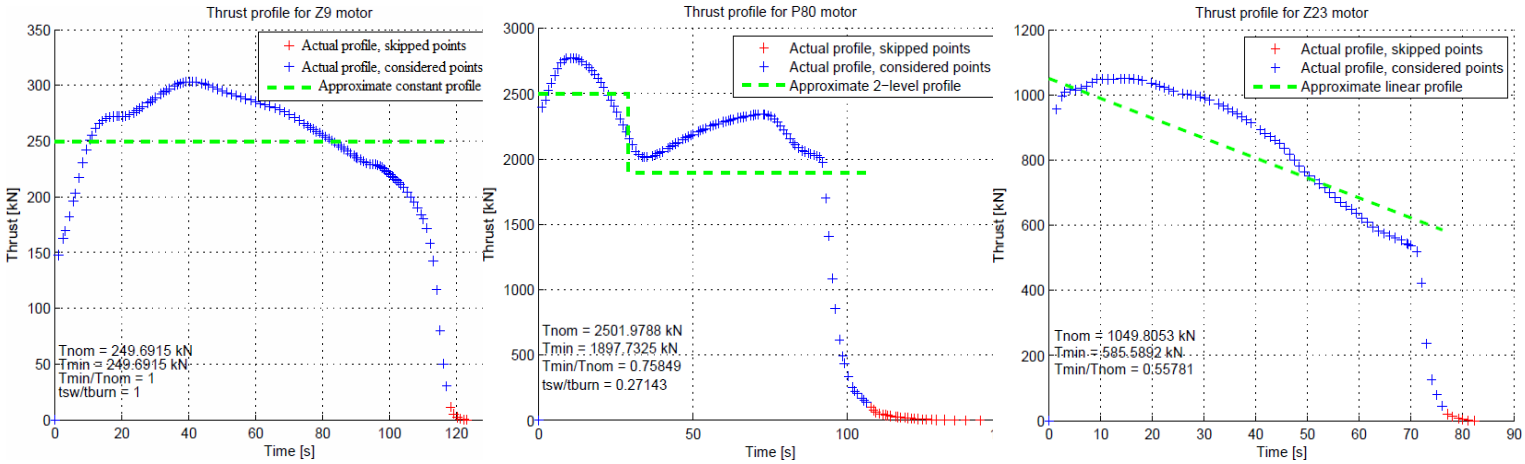


Figure 14: VEGA SRMs: examples of approximate thrust profiles developed for V1 models: constant thrust for Z9, 2-level thrust for P80 and regressing thrust for Z23.

In order to keep the analysis as simple as possible, motor's performance analysis and nozzle's sizing are performed only on the basis of the maximum thrust level T^*, so that T^*_{min} and t_{switch} remain as simple trajectory optimization variables with no influence on the design. A single chamber pressure is therefore assumed and performances are computed with CEA code for a single design point. Grain geometry is not included in the model, assuming a fixed Filling Factor (FF) to compute the motor's dimensions. This avoids both the introduction of shape design variables and the instant-by-instant computation of the burning area with complex geometric relations. Although largely simplifying the analysis and saving considerable CPU times (single CEA execution), this approach presents the important drawback that the thrust profile's actual feasibility cannot be verified, and its impact on the motor's design (in particular on its dimensions) is neglected. Constraints need thus

to be a priori imposed by the user to avoid thrust profiles resulting in unfeasible grain or large differences in the motor's dimensions. Three types of constraints can be defined: the motor's ***FF*** (80% by default), the ranges for the three thrust optimization variables (T^*, T^*_{min}, t_{switch}) and the maximum achievable ratio of burning area to throat area A_b/A_t.

The burn-to-throat area ratio is computed within V1 models starting from the chamber pressure p_{cc}, which is obtained by simple mapping of the optimization variable to a predefined pressure range of ***[40;120] bars***, and the fixed SP properties resulting from the assumed formulation. From the mass conservation law under steady-state approximation, p_{cc} can be written as:

$$\frac{A_b}{A_t} = \frac{p_{cc}^{(1-n)}}{K_{SP}} \qquad with \;\; K_{SP} = \frac{a \cdot \rho_{SP} \cdot \sqrt{R_{gas} \cdot T_{cc}}}{\sqrt{\gamma \cdot \left(2/\gamma+1\right)^{\gamma+1/\gamma-1}}}$$

where K_{SP} depends on the chosen SP formulation and chamber conditions. Since p_{cc} is assumed to be constant, the chamber temperature T_{cc} as well as the burn rate coefficients n and a are also considered constant. The default SP formulation is taken from the P241 boosters of Ariane 5 rocket, with main constituents being Ammonium Perclorate (AP, ***68%***), Hydroxyl-Terminated PoliButadiene (HTPB, ***14%***) and Aluminium powder (***18%***). The required physical and ballistic properties are the density of the solid ρ_{SP}***=1770 kg/m^3***, the chamber temperature T_{cc}***=3328.9 K***, the ratio of specific heats γ***=1.142***, molar mass ***M=27.4kg/kmol*** and the gas constant R_{gas}***=303.4479 J/(kg/K)*** of the exhaust gases, as well as the burn rate coefficients ***a=3.5958·10^{-5}*** and ***n=0.35***.

4.2.2.2. LP feed system analysis

As for SP, any type of physics-based analysis of the feed system is skipped to simplify V1 models. The value of chamber pressure p_{cc} is therefore obtained by mapping the related optimization variable to ranges which depend on the type of feed system. Specifically, turbopump systems with closed (Staged Combustion, SC, or Expander Cycle, EC) and open (Gas Generator, GG) cycles were considered, as well as simpler Pressure Fed (PF) systems. Other parameters derived from the propellant and feed type are the tanks pressure p_{tanks}, chamber inlet temperature $T_{cc,in}$ and the mass flow percentage which is "wasted"' to run the turbopumps in case of open-cycle pump-fed design. The discrete optimization variables ***Prop*** and ***Feed*** (or their stored values for OTS LREs) respectively define the propellant and feed system type. Figure 15 shows the 9 different combinations which were implemented for LP systems, reflecting the following engineering considerations:

- EC engines require a cryogenic fuel, which easily reaches the boiling point when circling around the thrust chamber walls, hence is only available in case of LOx-LH_2 engines.
- EC engines are subject to the "square-cube rule": as the size of the nozzle increases with increasing thrust, the nozzle surface area and fuel volume increase respectively as the square and the cube of the radius. For this reason, there exists a maximum engine size (i.e. thrust) beyond which there is no longer enough nozzle area to heat enough fuel to drive the turbopumps. As a consequence, engines with up to only 300 kN can be obtained with EC design.
- PF systems cannot be efficiently applied with cryogenic propellants, hence these are only available if at least the fuel is storable.

Other variations of the four main feed systems included in Figure 15 are possible, such as expander bleed cycle, full flow staged combustion, or bleed pressure-fed systems. However, examples of these architectures are rare and their modelling was not deemed necessary.

Table 5 reports the assumed ranges of chamber pressure p_{cc}, tanks pressure p_{tanks}, and percentage mass flow loss $\dot{m}_{loss}$, which vary depending on the feed type. p_{tanks} and $\dot{m}_{loss}$ are assumed to be linear functions of p_{cc}, and hence linearly depend on the optimization variable which defines the chamber pressure. Table 6 shows instead the chamber inlet temperature $T_{cc,in}$ for the different fuels and oxidizers, including also assumptions on whether passage through nozzle cooling jackets is exploited to heat propellants before injection in the combustion chamber.

Feed type	p_{cc}, min-max	p_{tank}, min-max	$\dot{m}$ losses, min-max
Pressure-fed	6.8-15.6 bars	13-30 bars	0%
Gas generator	30-120 bars	2.0-3.3 bars	1-3%
Expander cycle	30-70 bars	2.0-3.4 bars	0%
Staged combustion	80-270 bars	2.0-3.4 bars	0%

Table 5: Assumed ranges for the main feed system parameters, with linear variation of p_{tanks} and mass losses as a function of p_{cc}.

Propellant	LO_x	LH_2	CH_4	$RP1$	MMH	N_2O_4
Cooling jacket	No	Yes	Yes	No	No	No
$T_{cc,in}$	90 K	200 K	360 K	250 K	291 K	279 K

Table 6: Assumed values of chamber inlet temperature and passage through cooling jacket for each oxidizer or fuel.

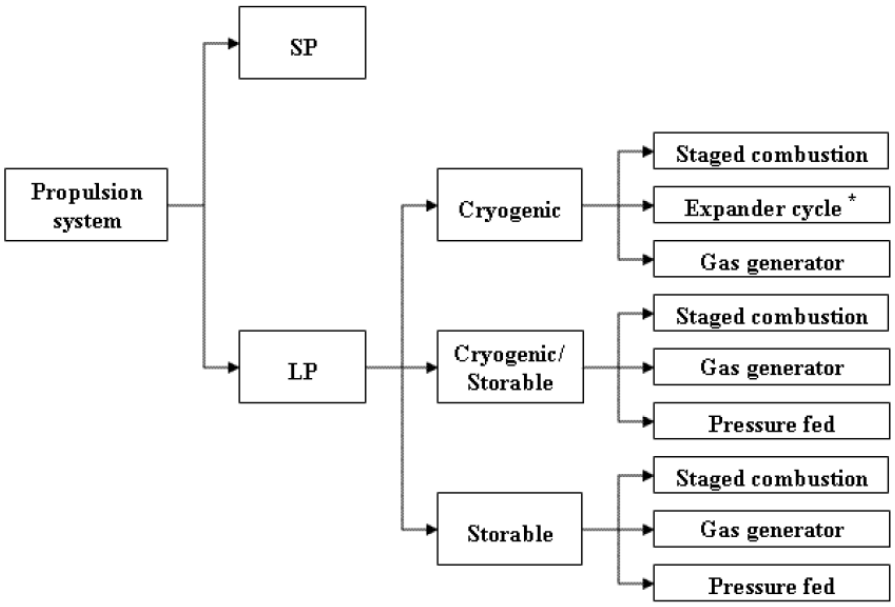


Figure 15: Possible propellant and feed systems tree defined for the propulsion system design in V1 models

All the assumptions regarding pressure ranges are based on an analysis of the collected LREs database, whereas tank pressures in the order of 2.0-3.4 bars (a positive pressure is needed to suppress pump cavitation) and mass losses for GG engines in the order of 3-7% are reported by Sutton (**[86]**). Although GG's pre-burner mass flow partially contributes to the thrust (with specific impulses in the order of 70-80 s), this was neglected in front of lower values of mass flow loss with respect to Sutton's suggestions (up to 1-3% instead of 3-7%), which also accounts for the better present technology. Finally, regarding nozzle cooling, it was assumed that only cryogenic fuels are passed through a thrust chamber's cooling jacket (i.e. liquid hydrogen and methane at 200/450 K instead of the 21/110 K for storage), whereas liquid oxygen and other storable propellants are directly fed into the combustion chamber at storage temperature.

4.2.3. Theoretical performance analysis with NASA's CEA

The core of the propulsion system analysis is constituted by the calculation of the theoretical performance of rocket engines, which is usually performed in conceptual and preliminary phases through simplified chemical equilibrium analyses. As reviewed in Technical Note D1b (**[A2]**), several computer codes for chemical equilibrium calculations were developed in the past in several institutions, but the recognized standard is NASA's CEA software, which is freely distributed on Glenn Research Centre (GRC)'s website[25] and is well documented with a theoretical reference publication (**[93]**, 1994) and a Users Manual (**[94]**, 1996). CEA was selected for V1 models due to its open source distribution as well as its good accuracy, computational efficiency and reliability which were proven by countless applications.

Although NASA publications **[93]** and **[94]** should be used as reference for any detail on CEA, a brief overview is given here for better clarity. The Chemical Equilibrium and Applications (CEA), in active development since the late 50's at NASA Lewis Flight Propulsion Laboratory (now GRC), is a generic program for solving chemical equilibrium problems which can be used in four main modes with different purposes:

1. To obtain chemical equilibrium compositions for assigned thermodynamic states
2. To calculate theoretical rocket performance for a finite- or infinite-area combustion chamber
3. To evaluate Chapman-Jouguet detonations
4. To calculate shock tube parameters for both incident and reflected shocks.

Obviously, the second functionality is the one of interest for the theoretical performance assessment of solid and liquid rocket engines. Chemical equilibrium is usually described by either of two equivalent formulations, equilibrium constants or minimization of free energy. In CEA, the minimization-of-free-energy method is employed, since each species can be treated independently without specifying a set of reactions a priori, as is required with equilibrium constants. A Newton-Raphson iterative procedure allows to solve the minimization problem, supported by a number of routines aimed at avoiding numerical difficulties, such as initial estimates, tests for condensed phases, phase transitions and triple points, convergence, accidental singularities, special handling of ions and consideration of trace species. Typically, 8 to 20 iterations are required for convergence (**[93]**), resulting in a very fast execution of the code (i.e. less than ~0.05 s).

[25] http://www.grc.nasa.gov/WWW/CEAWeb/ceaHome.htm

Assumptions in the chemical equilibrium formulation and rocket engines analysis used in CEA include the following:

- Ideal gas and homogeneous mixing without interactions among species, which holds rather well up to few percent of condensed species contained in the mixture.
- Mono-dimensional form of the continuity, energy and momentum equations.
- Zero velocity at the combustion chamber inlet.
- Constant cross-sectional area of the combustion chamber, with non isentropic, irreversible combustion process.
- Complete and adiabatic combustion.
- Isentropic expansion in the nozzle.
- Performances can be estimated with both Finite Area Combustor (FAC) and Infinite Area Combustor (IAC) geometric assumption, as shown in Figure 16. For the IAC model, only one combustion point is calculated at infinite area (inf) whereas for the FAC model two combustion points are calculated at the combustion chamber injectors (inj) and at the combustor end (c), modelling a pressure drop from these two points due to non null flow velocity along the chamber. In addition to these two combustion points, a combustion calculation for an infinite-area combustor indicated by the dashed line is also made, as an aid in an iteration procedure to obtain combustor end conditions. According to **[93]**, FAC model ensures slightly more realistic results at the price of a negligible computational cost.
- Both shifting chemical equilibrium and frozen composition expansion can be computed throughout the nozzle, the first overestimating and the second underestimating the engine's performances. A mixed approach is also common in literature, assuming frozen mixture starting from the nozzle's throat.

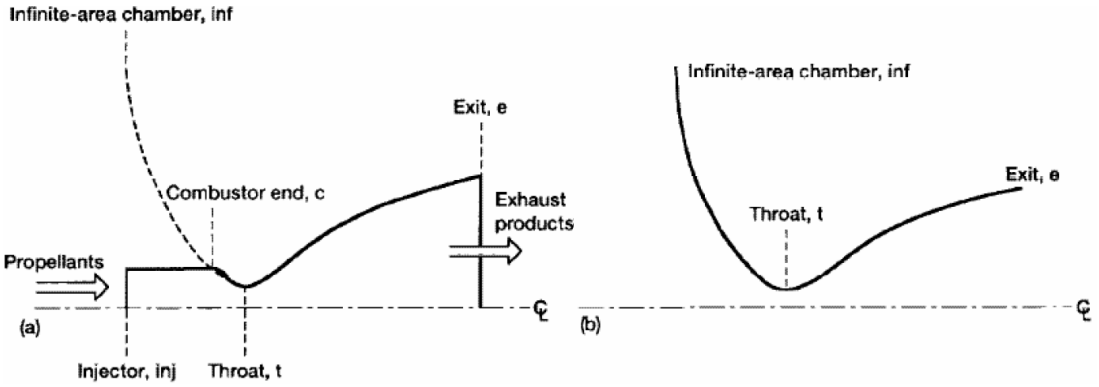


Figure 16: Schematics of rocket combustion chamber cross sections for a) FAC and b) IAC models, as taken from [93].

CEA is used within SVAGO to compute the theoretical specific impulse and the nozzle geometry for both liquid and solid propulsion systems, therefore using the ***ro*** (rocket) problem type. CEA allows running FAC calculations only for full shifting equilibrium, hence a choice was presented between the following two models:

- FAC with shifting equilibrium throughout the entire nozzle (in CEA format: ***ro eq fac***)
- IAC with shifting equilibrium until the nozzle throat and frozen expansion thereafter (in CEA format: ***ro nfz=2 iac***, where nfz=2 option indicates the mixed nozzle expansion approach).

As already stated, the former allows higher accuracy related to the combustion chamber geometry, whereas the second provides a good compromise between overestimating shifting equilibrium and underestimating frozen expansion. After several test runs of the code, the FAC vs. IAC model was proven to cause only very slight variations of the specific impulse[26], as also confirmed in a former empirical study by NASA (**[95]**). On the contrary, the shifting equilibrium throughout the entire nozzle results in much higher specific impulses, up to 10%, with respect to the mixed approach. Considering the shifting equilibrium approach too optimistic, it was decided to implement the second option, neglecting the contraction ratio A_{cc}/A_t in the calculation of the theoretical performance. Note however that empirical corrections to derive the real I_{sp} depend on A_{cc}/A_t, as shown in the next paragraph.

On the implementation side, CEA was linked through input/output files parsing. In fact, execution of the compiled CEA is almost instantaneous and the Fortran code is particularly complicated, so that direct linking through dll seemed an unworthy task. Input files have very limited dimensions and a clear syntax, which made the development of a parsing script very easy. Modification of the Fortran code to only output the few parameters needed for the prosecution of the MDA also allowed to limit the dimensions of the output files to 8 double numbers. As a result, slow-down caused by the hard drive access within CEA call is very limited.

As previously mentioned, all calculations related to the performance of the propulsion system are referred to the operational conditions at the optimal altitude for the nozzle (i.e. nominal conditions, indicated with superscript *), whereas

[26] Maximum variation of I_{sp} from $A_{cc}/A_t = 1.5$ to $A_{cc}/At = \infty$ is less than 0.1% for 6 different LREs and up to 0.7% for 3 SP formulations.

the specific impulse at each given altitude is computed from I_{sp}^*, p_e^* and A_e directly within the EoM in the Trajectory module.

Stemming from all of the above assumptions, the C++ parser translates to CEA's input files syntax the following data:

- Reactants chemical composition, defined by the discrete optimization variable ***Prop***
- Mixture ratio for LP, optimized within given boundaries which are defined by default as ***[4.5; 6.1]***, ***[2.1; 2.8]*** and ***[1.6; 2.1]*** respectively for $\boldsymbol{LO_x}$***- LH***$_2$, $\boldsymbol{LO_x}$***-RP1*** and $\boldsymbol{N_2O_4}$***- MMH***.
- Percentages of reactants in the SP formulation, which is assumed to be constant (***68% AP***, ***14% HTPB***, ***18% Al***).
- Chamber pressure $\boldsymbol{p_{cc}}$ and propellants initial temperature $\boldsymbol{T_{cc}}$, as obtained from the grain/feed analysis.
- Nozzle pressure ratio $\boldsymbol{p_{cc}/p_e}$, where the exhaust pressure $\boldsymbol{p_e}$ is obtained directly from the nozzle optimal altitude ***H*** through typical atmosphere definitions (Earth US '76 or exponential, see **Paragraph 4.6.1**).

Throughout the call to CEA, there's no reference to the actual engine's size. Output areas are given as ratios of the throat section and output performances are given as adimensional parameters, hence engine's scaling is performed after the execution of CEA (see **Paragraph 4.2.5**). As mentioned above, only 8 output parameters are printed by CEA to an output file:

- Characteristic velocity $C_* = \frac{p_c \cdot A_t}{\dot{m}}$ and thrust coefficient $C_T = \frac{T_{nom}}{p_c \cdot A_t}$, which are used to define the corrected specific impulse through different efficiency factors as shown in the next paragraph.
- Specific impulse $I_{sp} = C_* \cdot C_T \Big/ g_0$, which is only used as a cross-check.
- Nozzle expansion ratio $\varepsilon = A_e / A_t$, used to define the engine scaling and nozzle dimensions.
- Nozzle pressure ratio $\boldsymbol{p_{cc}/p_e}$, exhaust gases specific heats ratio $\boldsymbol{\gamma_e}$, molar mass $\boldsymbol{M_e}$ and temperature $\boldsymbol{T_e}$: although $\boldsymbol{p_{cc}/p_e}$ is an input of CEA, its value at the end of the calculations is checked. A common issue with CEA is in fact the inability to converge calculations in the divergent section of the nozzle for frozen composition of some types of reactants generating condensed species among the chemical products, typically solid propellants[27]. When this occurs, conditions at the throat are returned instead of those at the exhaust, hence with a modified pressure ratio $\boldsymbol{p_{cc}/p_t}$ instead of $\boldsymbol{p_{cc}/p_e}$. To avoid losing potentially good design solutions, this condition is detected by checking the pressure ratio value. If this is different from the input value, calculations for the nozzle expansion are performed analytically through isentropic relations, therefore obtaining the exhaust conditions (and overall theoretical performances) starting from the throat conditions $\boldsymbol{p_{cc}/p_t}$, $\boldsymbol{\gamma_t}$, $\boldsymbol{M_t}$ and $\boldsymbol{T_t}$.

Complete nozzle geometry in terms of area distribution A_j/A_t at any number of stations from combustion chamber to nozzle exhaust, which is not included among the outputs within the optimization process for efficiency reasons, may be requested by the user for visualization of the final solutions.

As a final note, the expansion ratio $\boldsymbol{\varepsilon}$ of LREs is usually available from companies' datasheets or other sources, whereas no information can be found regarding the pressure ratio used as input in the procedure described above. For this reason, a different CEA execution mode needs to be exploited when analyzing the theoretical performances of OTS engines, by parsing $\boldsymbol{\varepsilon}$ as input and reading $\boldsymbol{p_{cc}/p_e}$ as output of the analysis. There's basically no difference among these two approaches, since - if data are coherent - exactly the same performances are returned.

Four examples of CEA input files created by the C++ parser are reported below, the first three for LP and the last for a SP formulation. Note how detailed reactants composition needs to be provided in examples 2 and 4 for Kerosene and HTPB. Although the problem's set-up allows optimizing the propellant selection among only four options (cryogenic, cryo-storable, storable, solid), the implemented CEA parser allows the user to select any type of reactant for each of these classes. Several reactants are stored within CEA in the file *thermo.inp* provided with the source code, or the detailed chemical composition can be given in alternative. Russian Kerosene and Unsymmetrical Di-Methyl Hydrazine (UDMH) are for example used instead of the default RP1 and MMH for several OTS LREs, as reported in **Table 4**. For more information about CEA format see reference **[94]**.

CEA input file example 1: Prop=1 (LO_x-LH_2), α_P=6.03, p=206.4 bar, A_e/A_t=77

```
prob ro nfz=2
reac fu=H2 w=100 t,k=200
ox=O2(L) w=100 t,k=90
o/f=6.03 p,bar=206.4 sup=77
outp massf short
end
```

[27] The error message for this case, which may be recognized by those familiar with CEA, is the following: "Warning!! Calculations were stopped because next point is more than 50 K below the temperature range of a condensed species (rocket)"

CEA input file example 2: Prop=2 (LO_x-Kerosene[28]), α_P=2.4, p=150 bar, p_{cc}/p_e=700

prob ro nfz=2
reac fu=kerosene C 1 H 1.9532 w=100 t,k=250 h,BTU/#=-22923.0
ox=O2(L) w=100 t,k=90
o/f=2.4 p,bar=150 pip=700
outp massf short
end

CEA input file example 3: Prop=3 (N_2O_4-MMH), α_P=2.05, p=15 bar, p_{cc}/p_e=600

prob ro nfz=2
reac fu=CH6N2(L) w=100 t,k=298
ox=N2O4 w=100 t,k=298
o/f=2.05 p,bar=15 pip=600.0
outp massf short
end

CEA input file example 4: Prop=4 (SP), 68% AP, 14% HTPB[29], 18% Al, p=63.3 bar, p_{cc}/p_e=118.8

prob ro nfz=2
reac na=AL(cr) w=18 t,k=298.15
na=NH4CLO4(I) w=68 t,k=298.15
na=HTPB C 7.075 H 10.65 O 0.223 N 0.063 w=14 t,k=298.15 h,kj/mol=-58
p,bar=63.3 pip=118.8
outp massf short
end

4.2.4. Specific impulse losses and minimum altitude estimation

The values of C_* and C_T obtained from CEA analysis represent theoretical upper limits of the performances achievable respectively in the combustion chamber and in the nozzle, hence they need to be corrected with analytical or empirical factors for a large number of physical effects which contribute to degrading the performance with respect to the theoretical values. Following a common approach, all degrading effects were condensed in two efficiency factors η_{C*} and η_{CT} for the chamber and nozzle losses, plus an additional term η_{mloss} due to the mass flow losses in turbopump open cycles, so that the real specific can be obtained as:

$$I_{sp} = \frac{C_* \cdot C_T \cdot \eta_{C_*} \cdot \eta_{C_T} \cdot \eta_{\dot{m}_{loss}}}{g_0}$$

The efficiency factors were derived in part from Sutton (**[86]**) and in part from an interesting Rocketdyne technical paper (**[97]**). Note that the numerical values of all parameters building up η_{C*}, η_{CT} and η_{mloss} are extremely important for the accuracy of the performance assessment. These were therefore determined in three steps:

1. Reference figures suggested in **[86]** and in **[97]** were taken as initial values.
2. Additional hints from ESA technical support were incorporated.
3. All parameters were finely tuned through a calibration procedure based on a set of LP and SP engines. Calibration against existing engines is particularly critical, since information on rocket engine losses are some of the most protected information in the launchers industry and it is thus not possible to obtain reliable data directly in the form of I_{sp} loss factor. Conceptual models validation of **Chapter 5** includes a detailed outlook on the calibration process.

Mass flow loss efficiency η_{mloss} is the most simple of the three corrective terms, linearly varying for GG engines depending on the chamber pressure as specified in Table 5 and being equal to 1 for EC, SC and PF engines. Although Sutton suggests values between 0.93 and 0.97 for GG cycle, the calibration procedure determined that higher efficiencies better reflect the current state-of-art, hence the range 0.97-0.99 was implemented.

Combustion chamber losses contributing to η_{C*} are instead of various origins; the following terms were implemented:

- Non-perfect combustion efficiency and unsteady conditions, usually resulting in very small effects. This is translated in a constant factor $\eta_{combustion}=0.995$ (**[86]**).
- Real gas properties, also resulting in small losses and conservatively taken from (**[86]**) as a constant $\eta_{realgas}=0.993$.
- Non infinite cross-sectional area, resulting in a pressure drop at the end of the combustor due to non-zero flow velocity. Since this aspect is not accounted for in the employed IAC model, an empirical correction factor depending on the chemical reactants as well as on the contraction ratio A_{cc}/A_t was defined. To assess the numerical values of this factor, FAC - shifting equilibrium CEA analyses were executed for 3 SRMs (Ariane-5's P241 boosters, Space

[28] Heat of combustion and other data were taken from a very interesting paper (**[96]**) comparing Russian Kerosene with American RP1.
[29] HTPB formulation and heat of combustion are those commonly used at the Solid Propulsion Laboratory (SPLab), Politecnico di Milano

Shuttle's Reusable Solid Rocket Motor, RSRM, and VEGA's P80 first stage) and 6 LREs representative of all propellant combinations (SSME, RL-10-B-2, RS-27A, RD-120, RD-275M and Aestus). Results are consistent throughout the different SRMs/LREs, and examples of I_{sp} vs. A_{cc}/A_t profiles are shown in **Figure 17** for SSME and P241. In particular, it was highlighted how the effect of the chamber's contraction ratio is negligible for LP and slightly more influential for SP engines. Two different functions of the contraction ratio up to 10 were therefore implemented as corrective factor for SP and LP, with two-piece linear interpolations of the percentage loss at $A_{cc}/A_t=4$ and $A_{cc}/A_t=10$. Percentage losses were averaged over the 3 tested SRMs and 6 LREs, resulting in minimum efficiencies for $A_{cc}/A_t=1.5$ (lower contraction ratios are considered unrealistic) of $\eta_{contraction}=0.9928$ and 0.9996 respectively for SP and LP systems.

Other secondary effects are neglected in the implemented models, building up an overall combustion chamber efficiency which can be written as: $\eta_{C^*} = \eta_{combustion} \cdot \eta_{realgas} \cdot \eta_{contraction}$

This results in quite high chamber efficiencies, ranging from $\eta_{C^*}=0.981$ to $\eta_{C^*}=0.988$, which is in line with typical rocket engines values.

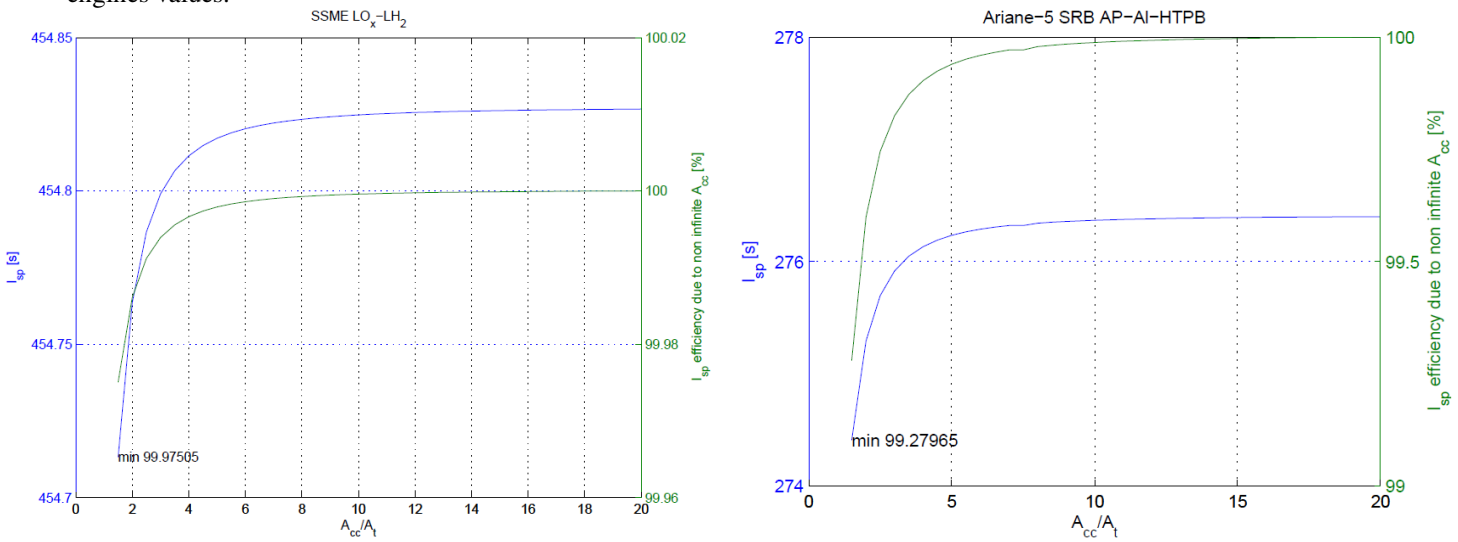


Figure 17: Absolute value and percentage variation of theoretical I_{sp} with respect to the chamber-to-throat area for SSME cryogenic engine (left) and P241 solid motor (right). Data are referred to shifting equilibrium FAC analyses at the nozzle's optimal altitude.

Nozzle losses contributing to η_{CT} are higher than combustion chamber losses, and are mainly due to the following effects:

- Geometrical losses due to radial component of the exhaust momentum, which depend on the nozzle type (bell or cone) defined by the optimization variable ***Nozzle*** and on the divergent angle δ_{div}, which may by default take values from 8 to 20 deg. The following corrective factor is implemented as taken from **[97]**:

$$\eta_{div} = \begin{cases} 0.992 & if \quad Nozzle = 1 \ (bell) \\ \dfrac{1+\cos(\delta_{div})}{2} & if \quad Nozzle = 2 \ (cone) \end{cases}$$

- Presence of condensed phases, which may have a rather large effect on the specific impulse for SP engines, typically from 1% to 5% according to **[86]**. The percentage of Al powder in the SP is the main variable affecting such loss, is taken as 18% in the default formulation but can be manually provided by the user in the range 10-22%. A linear variation of $\eta_{condensed}$ in the range 1-2.5% is therefore assumed following the calibration procedure, again less conservatively with respect to Sutton ([86]).
- Boundary layer losses, due to viscous effects at the wall, taken as a constant $\eta_{bl}=0.986$ from **[97]**.
- Finite rate chemistry contributes to reducing the theoretical specific impulse computed with chemical equilibrium expansion in the nozzle. However, this effect is already conservatively taken into account with the frozen mixture hypothesis after the throat, hence $\eta_{chemical}=1.0$ is assumed.
- Nozzle erosion, especially in the throat region, may cause shape variations towards the end of the burn period for ablatively cooled nozzles, resulting in an additional loss of performance. This effect is relatively significant and is taken as $\eta_{erosion}=0.993$ (**[86]**) for storable LP and SP and $\eta_{erosion}=1$ otherwise (under the assumption of regenerative cooling).

From the above constituents, the overall nozzle efficiency is written as: $\eta_{C^*} = \eta_{combustion} \cdot \eta_{realgas} \cdot \eta_{contraction}$, resulting in a largely variable factor ranging from a worst-case of $\eta_{CT}=0.914$ for a 22% Al-powder SP with 20 deg conical nozzle to a best

case of η_{CT}=*0.978* for a cryogenic LP with bell-shaped nozzle. This matches very well with the approximate range from 92 % to 98% indicated by ESA propulsion expert as typical overestimation values for CEA. Note also that the overall engine efficiency may be further corrected by a factor considering the class of engine, i.e. high performance or low cost engine as defined by the optimization variable ***LCE***, but this was not modelled within V1 propulsion analysis.

A final note on empirical corrections of the theoretical performance is related to the altitude effects. Over/under expansion of the nozzle is already translated within the Trajectory models in a lower/higher theoretical performance (thrust and specific impulse). Hence, it is assumed that all efficiency factors described above do not depend on the external pressure conditions, so that the same efficiency corrections are applied at all altitudes. This way, both the calculation of the instantaneous performance at each trajectory time step and the engines' performance data structure are simplified.

However, an assessment of the minimum operational altitude for a given engine must be introduced in the propulsion system analysis even at the conceptual level of V1 models. In fact, if no information on the minimum altitude is available, optimizers will most likely push for the highest possible expansion ratio in new designs, without considering low altitude operability: if over-expansion is excessive (too low altitude for the given nozzle's expansion ratio and shape), flow separation occurs and a normal shock wave is generated within the nozzle, resulting in an unacceptable performance loss and possibly in a catastrophic failure of the engine due to induced stresses.

In order to identify such inadequate conditions, a simple model was developed to estimate the maximum operational ambient pressure $p_{a,max}$, which is then used within the trajectory integration to constraint the ignition altitude of each stage/booster. Different approximate methods, both analytical and empirical, were proposed in the past to determine $p_{a,max}$ of rocket nozzles (see **[98]** for a good review). However, a simple criterion taken from Sutton was implemented and shown to provide reliable information with limited input data, making it adequate for use within the MDA. This estimate is based on the empirical value of p_{cc}/p_a resulting in flow separation and/or normal shock wave in the nozzle, which is expressed in **Figure 18** - taken directly from **[86]** - as a function of the expansion ratio. This function is represented by the dotted lower curve, assuming a ratio of specific heats γ=*1.3*. Similar graphs are reported in **[86]** for other values of γ, hence simple bi-dimensional linear interpolation can be used to determine $p_{cc}/p_{a,max} = f_{LinInterp2D}(\varepsilon, \gamma)$, which is then used to compute $p_{a,max}$ from the known chamber pressure p_{cc}. The maximum ambient pressure is finally used to derive the nozzle's minimum altitude H_{min} as: $H_{min} = \begin{cases} H(p_{a,max}) - 3\,km & if \quad Nozzle = 1\,(bell) \\ H(p_{a,max}) & if \quad Nozzle = 2\,(cone) \end{cases}$

where the correction of 3 km to account for anti-separation bell shapes was tuned within the calibration procedure mentioned above for specific impulse losses.

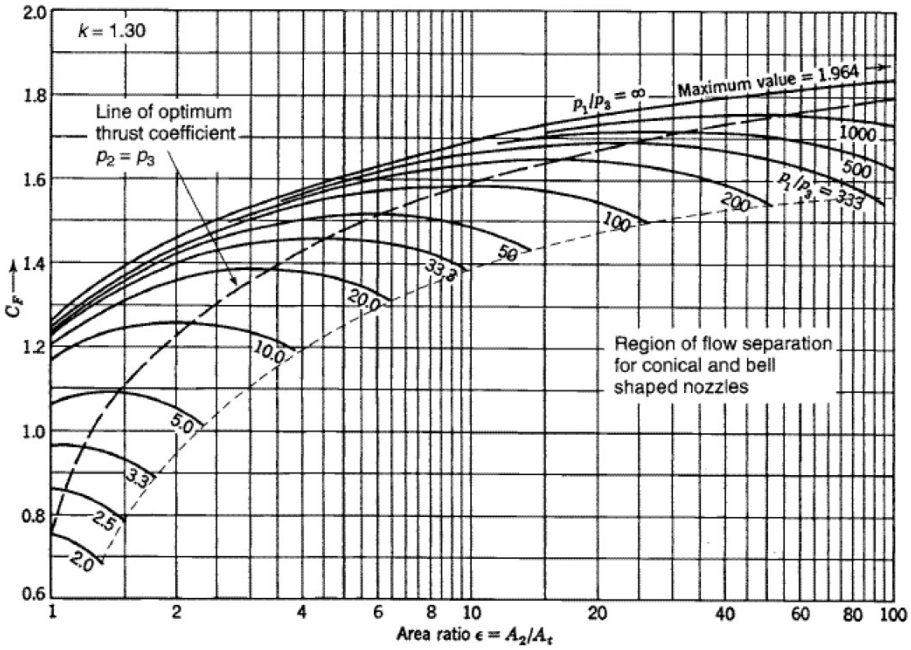


Figure 18: Theoretical thrust coefficient CT versus nozzle area ratio for γ=1.3, including flow separation/shock wave region, as directly taken from [86]. According to Sutton's convention, subscripts 1, 2 and 3 indicate respectively the combustion chamber, nozzle's exhaust section and ambient conditions.

4.2.5. Engine's scaling and propulsion system dimensions

At this point in the propulsion system analysis, the empirically corrected nominal values of C_* and C_T (hence I_{sp}), as well as the area ratios A_e/A_t, A_{cc}/A_t and A_b/A_t (SP only) and the pressures p_{cc}, p_t, p_e and $p_{a,max}$ are available. However, the engine still needs to be scaled to its nominal thrust level, which is obtained from the total required stage/booster's thrust T^* as $T^*_{eng}=T^*/N_{eng}$ (always referred to nominal – i.e. optimal altitude – conditions). Engine's scaling determines the areas of the characteristic nozzle sections (chamber, throat, exhaust), which are then used to estimate the length of the combustion chamber and nozzle. Finally, the overall propulsion system dimensions including tanks, solid grain and additional subsystems (e.g. SP igniter, LP pressurization, ...) can be derived from the propellant mass and geometric characteristics of the stage/booster. These three steps are presented in details in the next subparagraphs.

4.2.5.1. Engine's scaling

Engine scaling is extremely simple, only involving the definitions of I_{sp}, C_* and C_T. First, the engine's specific impulse and required mass flow per each engine in the nominal conditions are computed as:

$$I_{sp} = \frac{C_* \cdot C_T}{g_0} \text{ and } \dot{m} = \frac{T_{eng}}{(C^* \cdot C_T)}$$

Then, the throat area can be determined as:

$$A_t = \frac{C_* \cdot \dot{m}}{p_{cc}}$$

and all other sections' areas are obtained from the area ratios. A_b, A_{cc} and A_t are only used for the propulsion system's dimensions estimation, whereas A_e is also passed to the Trajectory module to define the losses due to non-nominal external pressure conditions.

4.2.5.2. Engine's dimensions estimation

The engine's dimensions need to be estimated in terms of maximum length and diameter (L_{eng} and D_{eng}), to be then used in the Geometry module to build-up the overall stage/boosters dimensions and verify interference constraints. The maximum diameter D_{eng} is simply computed from the exhaust area A_e, whereas its length is computed summing up the feed system and combustion chamber (for LP only) to the convergent and divergent sections of the nozzle.

A common geometry model is used for LP and SP nozzles, with total length given by:

$$L_{nozzle} = L_{conv} + L_{div} = \frac{(D_{cc} - D_t)}{2 \cdot tan(\delta_{conv})} + K_{shape} \cdot \frac{(D_e - D_t)}{2 \cdot tan(\delta_{div})}$$

Here, the convergent section is assumed to be a cone with a constant angle $\delta_{conv}=45$ **deg** and initial/final diameters defined by A_{cc} and A_t. In case of SP, although the whole solid grain actually constitutes the combustion chamber, the area A_{cc} still represents the nozzle inlet section area and is therefore used for the evaluation of the convergent section length. The divergent section's length is instead computed considering the angle defined by the optimization variable δ_{div}. Moreover, for bell nozzles (i.e. ***Nozzle=1***), a corrective factor of $K_{shape}=0.8$ is applied to take into account the reduced length, whereas for cone nozzles $K_{shape}=1.0$.

As for the combustion chamber of LP systems, a simple and rather effective method to estimate the length is based on the definition of the characteristic length $L_{cc}^{char} = \frac{V_{cc}}{A_t}$, where the chamber's volume V_{cc} includes also the convergent section of the nozzle. L^{char}_{cc} represents the length that a chamber of the same volume would have if it were a straight tube of area A_t without convergent nozzle section, and is a characteristic value which remains approximately constant for the same propellants combination. In fact, values from 0.5 and 2.0 m are reported in **[86]**, with variations mostly due to propellants reactivity. Hence, values shown in **Table 7** are assumed within V1 propulsion models and are used to compute L_{cc} from A_t and V_{cc}.

Propellants combination	L^{char}_{cc}
LP cryogenic (LO_x-LH_2)	0.89
LP cryogenic-storable (LO_x-RP1)	1.15
LP storable (N_2O_4-MMH)	0.74

Table 7: Assumed values of combustion chamber's characteristic length for different propellant combinations, adapted from [86].

Finally, LP systems also include a feed system (i.e. turbopumps, tubing and valves) and an assembly of injectors. Although these items are arranges as much as possible on the sides of the combustion chamber, some additional length needs to be allocated. A corrective factor is thus applied, such that $L_{feed\,\&inj} = K_{ft} \cdot (L_{cc} + L_{conv})$, where K_{ft} depends on the engine's type, with the numerical values reported in , which were derived from visual inspection of existing LREs.

Feed system	K_{ft}
Pressure-Fed	0.6
Gas Generator turbopump	1.0
Expander Cycle turbopump	0.8
Staged Combustion turbopump	1.0

Table 8: Assumed values of feed and injector systems length as a percentage of $L_{cc}+L_{conv}$

From the above definitions, the total engine's length can be written as: $\begin{cases} L_{eng} = L_{feed\,\&inj} + L_{cc} + L_{nozzle} & for\ LP\ systems \\ L_{eng} = L_{nozzle} & for\ SP\ systems \end{cases}$

4.2.5.3. LP and SP system dimensions estimation

In addition to the maximum length and diameter of each engine, the dimensions of the other components of the propulsion system need to be assessed. For LP systems, these are length and diameter of the fuel and oxidizer tanks, whereas the volume occupied by the pressurization system is neglected in V1 models. For SP systems instead, the only additional component is the propellant grain, since the igniter is assumed to be placed within the grain's perforation. Note that by summing the maximum length of the engine to the length of the liquid tanks or solid grain, the stage/booster's length is almost completely determined. Only few geometrical parameters remain to be defined within the Geometry model, such as the interstage and intertank length (see **Section 4.3**). Moreover, although there may be diameter discontinuities between two consecutive stages, each stage's or booster's diameter is constant throughout its entire length and is equal to the LP tanks/intertank or SP grain diameter.

For LP stages/booster, the overall propulsion system's dimensions can be described with the following geometric parameters: the tanks' lengths L_{Ox} and L_{Fuel}, the intertank length L_{IT} and the overall stage/booster diameter $D=D_{Ox}=D_{Fuel}$. The procedure to compute these parameters is simple and involves two steps:

- Compute the required volume of oxidizer and fuel tanks from the values of propellants mass, mixture ratio and the fuel/oxidizer's densities, which are reported for common chemicals in **Table 9**. A ***10%*** margin on the required volume is conservatively assumed to account for both propellant ullage and unused propellants.
- Compute the stage/booster's diameter and length, either by directly imposing diameter continuity with respect to the stage or payload fairing above (i.e. if it is a stage and the boolean optimization variable D_{const}***=true***) or starting from the optimizable length-over-diameter ratio ***L/D*** and the required volumes. In both cases, oxidizer and fuel tanks are assumed to be cylinders with ellipsoidal domes of length equal to ***30%*** of the diameter. Note that in case of common bulkhead tanks (i.e. if the integer optimization variable ***TT=1***), a single cylinder is considered enclosing the sum of both volumes.

Propellant	ρ [kg/m^3]
LO_x	1140.0
LH_2	71.0
RP1	807.0
N_2O_4	1447.0
MMH	878.8
AP-HTPB-Al SP	1770.0

Table 9: Assumed storage densities for different liquid fuels/oxidizers and solid propellant formulations, as taken from [86].

For SP stages/boosters, solid grain's length L_{grain} and diameter $D=D_{grain}$ are sufficient - in combination with L_{eng} and D_{eng} - to describe the overall dimensions of the propulsion system. The procedure to obtain L_{grain} and D_{grain} results extremely simplified by the choice of neglecting the actual grain shape design. A constant ***FF=0.8*** is in fact assumed, which is a typical value for star grain configurations of existing SRMs ([86]). Besides, the internal burning area A_b computed from the SP grain analysis and engine scaling is considered constant and equal to that of the final burn instant, hence corresponding to the external stage/booster surface. From these considerations, two geometrical equations can be written:

$$\begin{cases} A_b = \pi \cdot D_{grain} \cdot L_{grain} \\ M_{prop} = \dfrac{FF \cdot \pi \cdot D_{grain}^2}{4 \cdot L_{grain} \cdot \rho} \end{cases}$$

from which L_{grain} and D_{grain} can be computed. With this model the ***L/D*** ratio defined by the related optimization variable cannot be satisfied for SRMs. This is in general not an issue, except for stages with diameter constrained to be the same as the stage (or fairing) above. In this latter case, the FF is adjusted to achieve the desired burn area with the given diameter, with resulting FF higher than 80% considered unfeasible. As a final note, in case of feasible grain designs the volumetric

filling factor can be used to compute the approximate volume of the grain's internal cavity (which will be used within the Weights model) as: $Vol_{cavity} = \pi / 4 \cdot (1 - FF) \cdot D_{grain}^2$.

4.2.6. Inert mass Weight Estimation Relationships

Besides performances and dimensions, the propulsion system largely affects the inert weights of each stage/booster. The total inert mass of the propulsion system $\boldsymbol{M_{PropSys}}$ is written according to V1 models as follows:

$$M_{PropSys} = M_{EA} + M_{ununsed}$$

$$with\ M_{EA} = \begin{cases} N_{eng} \cdot M_{eng} + M_{PressSys} & for\ LP\ pressure\text{-}fed \\ N_{eng} \cdot M_{eng} & for\ LP\ pump\text{-}fed \\ M_{eng} + M_{igniter} & for\ SP \end{cases} \quad and \quad M_{eng} = \begin{cases} M_{WER} + M_{TVC} & for\ LP \\ M_{nozzle} + M_{TVC} & for\ SP \end{cases}$$

where $\boldsymbol{M_{EA}}$ is the mass of the engines assembly, intended as the sum of one ore more engines and of the secondary propulsion subsystems (LP pressurization and SP igniter), and $\boldsymbol{M_{eng}}$ is the mass of a single engine. This includes the TVC system and the nozzle of SRMs or the thrust chamber and feed system of LREs. Note that the former two elements are synthesized in a common term for the main engine's mass, designated $\boldsymbol{M_{WER}}$. Typically in fact, both thrust chamber and feed system (turbopumps, gas generators, valves and tubing) are included in the mass figures available in literature for existing engines. Since these figures were used to develop historical based WERs, no additional terms should be considered when building up the main engine's mass.

The next subparagraphs briefly present the models implemented to estimate the weight of each of the terms included in the above weight build-up, in particular: 1) LREs' main engine (i.e. thrust chamber and feed system), 2) SRMs' nozzle, 3) unused LP and SP propellants, 4) LP pressurization system, 5) SP igniter, and 6) TVC system.

4.2.6.1. LREs' main engine mass

In order to estimate the main engine's mass of LREs, several WERs are available in literature, for example from a recent CDF study (**[99]**) on a HLLV and from a large collection of WERs compiled at Georgia Tech (**[100]**), which will also constitute the main WERs base for the other launcher subsystems (see **Section 4.5**). However, none of these available regressions allows representing the trade-off between all 9 different LP architectures considered in the generic Propulsion models developed for V1. For this reason, the database of LREs shown in **Paragraph 4.2.1** was used to develop from scratches 9 WERs which appear well suited in terms of accuracy for the architectural trade-off.

All of the developed historical WERs are based on the vacuum thrust level $\boldsymbol{T_{vac}}$, which is typically used for this purpose. Confidence thrust ranges were defined for each technology, on the basis of the class of existing engines (e.g. no more than 300 kN of thrust for expander engines). Note that in case the required thrust range is out of the confidence range, the thrust range constraint is considered violated and hence the design solution unfeasible.

Mass vs. thrust data were added for few engines which were not included in **Table 4** due to insufficient information, so that in total 51 LREs were employed as knowledge base for the WERs development. Linear, quadratic, power law and logarithmic curves were analyzed, with either a single fit over the whole confidence range or with two- piece fits. The best resulting regressions in terms of quadratic fit error for each technology were implemented within the propulsion models.

The developed WERs are shown from **Figure 19** to **Figure 27**, whereas the selected equations and numerical values of the coefficients are reported in **Table 10**. Finally, **Figure 28** condenses all regressions in a single figure, in order to give a broader overview of the implemented mass models, also in comparison with linear CDF-developed regressions (**[99]**).

Prop	Feed	T_{vac} [kN]	Components	N_{LRE}	WER equations (M_{WER} in [kg], T_{vac} in [N])
Cryogenic	SC	[500; 5000]	Pre-combustor(s) & turbopump(s) Valves and piping Injector and igniter Thrust chamber	5	$M_{WER} = a \cdot (T_{vac})^b + c$ $\begin{cases} a = -1.17899e+008 \\ b = -7.380845e-001 \\ c = +6.09805e+003 \end{cases}$
Cryogenic	EC	[50; 300]	Turbopump(s) Valves and piping Injector and igniter Thrust chamber	5	$M_{WER} = a \cdot (T_{vac})^b + c$ $\begin{cases} a = -9.76421e+004 \\ b = -4.27622e-001 \\ c = +8.97980e+002 \end{cases}$
Cryogenic	GG	[0; 8000]	Gas-generator(s) & turbopump(s) Valves and piping Injector and igniter Thrust chamber	6	$M_{WER} = a \cdot (T_{vac})^b + c$ $\begin{cases} a = 7.54354e-003 \\ b = 8.85635e-001 \\ c = 2.02881e+001 \end{cases}$
Cryo-Storable	SC	[0; 10000]	Pre-combustor(s) Turbopump(s) Valves and piping Injector and igniter Thrust chamber	8	$M_{WER} = a \cdot (T_{vac})^b + c$ (a_1, b_1, c_1) for $T_{vac} < 2050\,kN$ $\begin{cases} a_1 = 8.51852e-003 \\ b_1 = 8.52826e-001 \\ c_1 = 1.06632e+002 \end{cases}$ $\begin{cases} a_2 = +1.65368e+000 \\ b_2 = +5.69842e-001 \\ c_2 = -4.37849e+003 \end{cases}$
Cryo-Storable	GG	[200; 2000]	Gas-generator(s) & turbopump(s) Valves and piping Injector and igniter Thrust chamber	5	$M_{WER} = a \cdot (T_{vac})^b + c$ $\begin{cases} a = +3.75407e+003 \\ b = +7.05627e-002 \\ c = -8.84790e+003 \end{cases}$
Cryo-Storable	PF	[0; 400]	Valves and piping Injector and igniter Thrust chamber	5	$M_{WER} = a \cdot (T_{vac})^2 + b \cdot T_{vac} + c$ $\begin{cases} a = -2.13325e-009 \\ b = +1.70870e-003 \\ c = +6.38629e+000 \end{cases}$
Storable	SC	[0; 5000]	Pre-combustor(s) & turbopump(s) Valves and piping Injector and igniter Thrust chamber	5	$M_{WER} = a \cdot (T_{vac})^b + c$ $\begin{cases} a = +4.74445e-001 \\ b = +5.35755e-001 \\ c = -7.73681e+000 \end{cases}$
Storable	GG	[0; 3000]	Gas-generator(s) & turbopump(s) Valves and piping Injector and igniter Thrust chamber	6	$M_{WER} = a \cdot (T_{vac})^b + c$ $\begin{cases} a = +6.37913e+000 \\ b = +3.53665e-001 \\ c = -1.48832e+002 \end{cases}$
Storable	PF	[0; 150]	Valves and piping Injector and igniter Thrust chamber	6	$M_{WER} = a \cdot (T_{vac})^2 + b \cdot T_{vac} + c$ $\begin{cases} a = -3.36532e-008 \\ b = +4.74402e-003 \\ c = -1.93920e+001 \end{cases}$

Table 10: WERs for the main engine's mass of different types of LRE, including also the thrust confidence range, components included in the mass figure, and number of data points used for the fit.

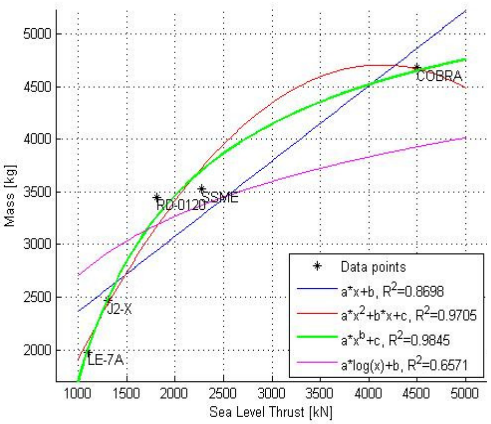


Figure 19: WERs for cryogenic SC LREs

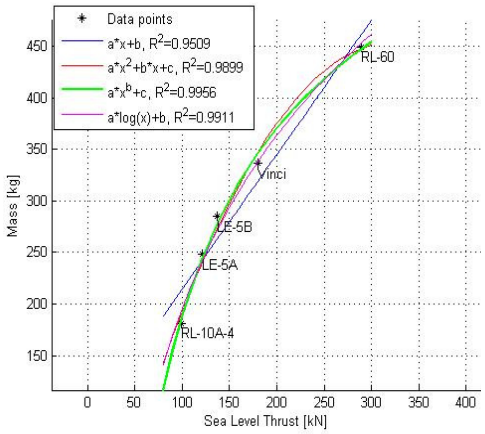


Figure 20: WERs for cryogenic EC LREs

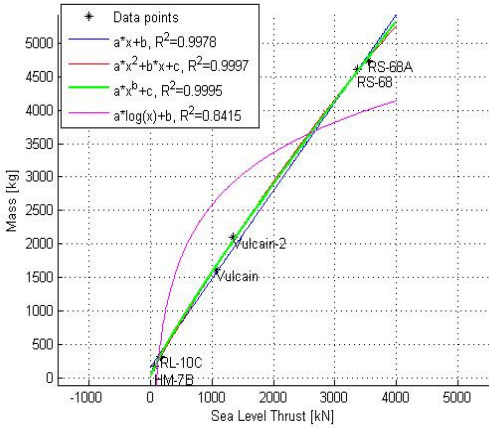


Figure 21: WERs for cryogenic GG LREs

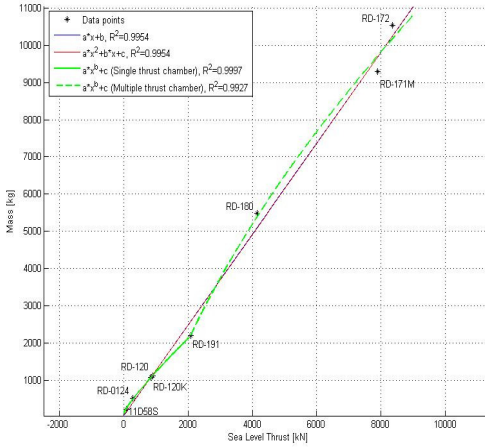


Figure 22: WERs for cryo-storable SC LREs

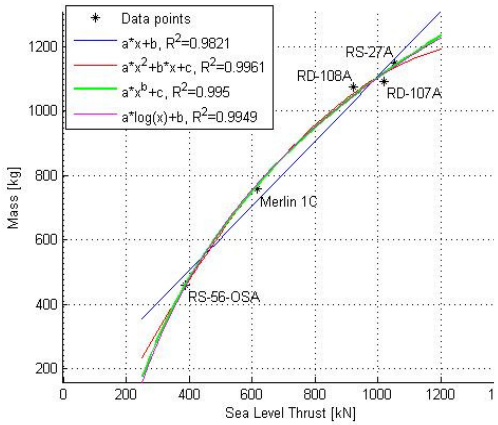


Figure 23: WERs for cryo-storable GG LREs

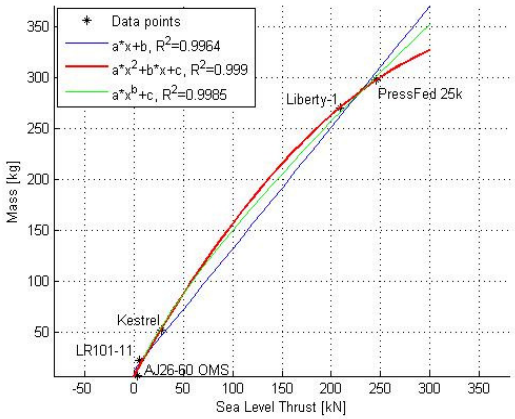


Figure 24: WERs for cryo-storable PF LREs

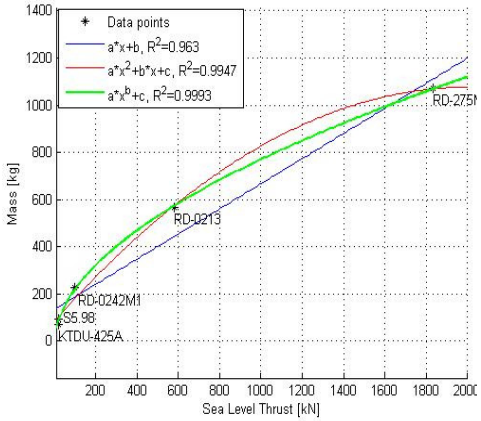


Figure 25: WERs for storable SC LREs

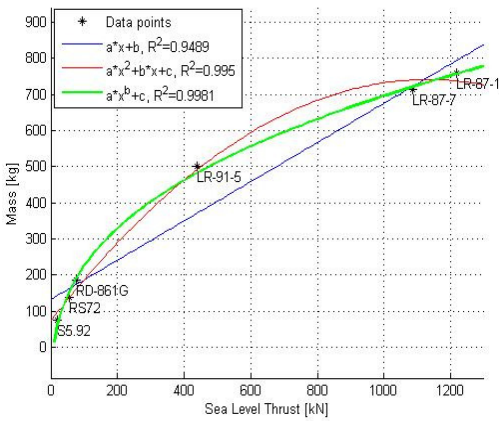


Figure 26: WERs for storable GG LREs

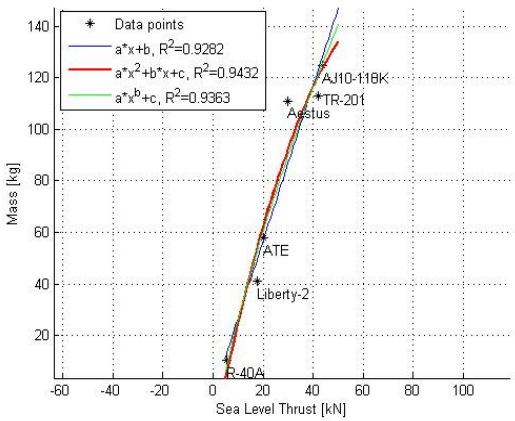


Figure 27: WERs for storable PF LREs

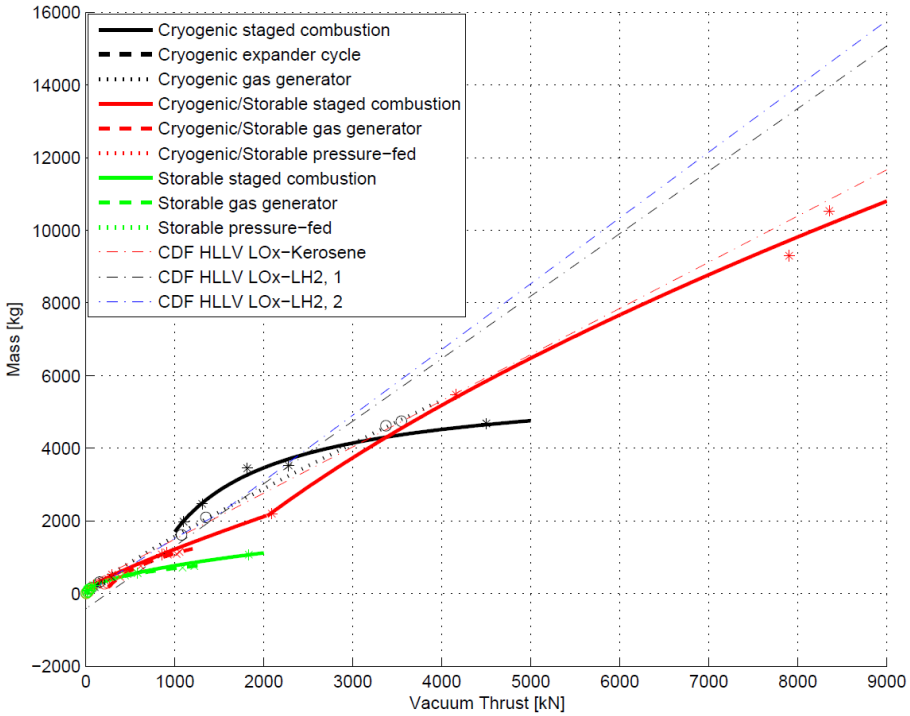


Figure 28: Summarizing view of all WERs developed for different types of LREs. Simplified linear regressions from HLLV's CDF study [99] are also reported for visual comparison

4.2.6.2. SRMs' nozzle mass

The main engine's mass of SRMs is intended as the mass of the nozzle alone, since TVC mass is given separately and no other propulsive components of relevant weight constitute SP systems. As an initial attempt, a database of SP engines from web sources was exploited to fit regression curves similar to those of LREs. However, most of the available data represent the total dry mass of the whole SRM, including case, TVC, igniter, TPS, etc, whereas limited information could be found for the mass of the nozzles alone. Moreover, data scattering and obtained regression error are rather large (R^2~0.95), as shown in **Figure 29**.

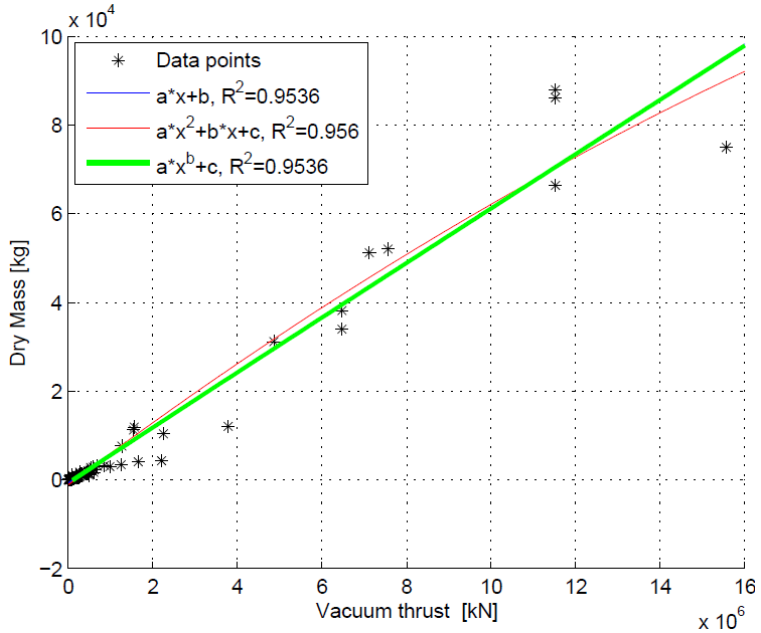


Figure 29: WERs developed to fit the total SRM's inert mass as a function of the vacuum thrust. Data are taken from a large database of existing SP systems collected on www.astronautix.com**.**

For these reasons, a very simple alternative model was developed and implemented for V1, computing the nozzle's external surface area and assuming a surface density value which is a linear function of the nozzle's expansion ratio. Calibrating the coefficients of the linear function with available data for Ariane 5 and VEGA SRMs, the nozzle's density and resulting nozzle's mass are:

$$M_{nozzle} = A_{nozzle} \cdot \rho_{nozzle}$$

$$\begin{cases} \rho_{nozzle} = 200 & for\ \varepsilon < 11 \\ \rho_{nozzle} = 200 - 2.889 \cdot \varepsilon & for\ 11 < \varepsilon < 56 \\ \rho_{nozzle} = 70 & for\ \varepsilon > 56 \end{cases}$$

4.2.6.3. Unused propellants

In any rocket engine system, some of the propellant loaded in the tanks/case cannot be used to produce effective thrust, but is "wasted" due to several causes. Although the quantity of unused propellants is usually small in terms of percentage of the total propellant mass, this cannot be neglected because it often constitutes a relevant fraction of the stages/boosters total inert weights. To keep the model as simple as possible, the unused propellants mass $\boldsymbol{M_{unused}}$ is taken as a constant percentage of the usable propellant mass $\boldsymbol{M_{prop}}$:

$$M_{unused} = K_{unused} \cdot M_{prop}$$

where $\boldsymbol{K_{unused}}$ is function only of the propulsion system type as reported in **Table 11**. Numerical data were adapted from references **[86]** and **[101]**, taking into account the following considerations:

- For LP systems, unused propellants include as most influential terms: a) propellant trapped in pipes, valves, or wetting tank walls and b) propellant flight reserves. According to the survey of launch vehicles presented in reference **[101]**, these are is in the range 0.16-0.49 % of the total propellant mass. In case of pump-fed systems, an additional fraction is to be typically allocated for the pressurization gas, which is in the range 0.23-0.35%.
- For SP systems, unused propellant is constituted by small portions of unburned propellant, knows as slivers, which remain at the periphery of the grain after the pressure drops below the deflagration limit. Although percentages of

sliver up to 5-7% were common in old SRMs, more recent designs are extremely efficient for this aspect, with less than 1% of sliver.

Propulsion system type	K_{unused}
LP pump-fed	0.0061
LP pressure-fed	0.0032
SP	0.0050

Table 11: Assumed values of unused propellants (trapped/reserve LP or sliver SP), adapted from [86] and [101].

4.2.6.4. Pressurization system

In all LP rockets, fuel and oxidizer tanks need to be pressurized to a predefined pressure limit in order to allow feeding propellants to the engine's combustion chamber. In case of pressure-fed engines, the mass of the pressurization system is particularly relevant, since much larger tank pressures are necessary in absence of a turbopump system to increase the chamber pressure. For this reason, in V1 models the pressurization system mass is added to the other components only in case of pressure-fed LP systems. Simple analytical relations can be used to estimate the required mass of pressurization gas (always considering Helium) and to size its tank, so that the total mass of the system is $M_{PressSys} = M_{press} + M_{press,tank}$

In particular, assuming an adiabatic process with ideal gas and a negligibly small initial mass of gas in the piping and propellant tank, the required pressurization gas mass can be estimated on the basis of the conservation of energy, with a 10% margin, as:

$$M_{press} = 1.1 \cdot \frac{p_{tank} \cdot Vol_{tank}}{R \cdot T_{press}} \cdot \frac{\gamma}{1 - p_{tank} / p_{press}}$$

where Vol_{tank} is the sum of the volumes of the oxidizer and fuel tanks, $R=2077\ J/(Kg \cdot K)$ and $\gamma=1.667$ are He's gas constant and ratio of specific heats, and $p_{press}=286\ bars$ and $T_{press}=293\ K$ are typical values of initial pressure and temperature in the pressurization gas tank as taken from **[86]**.

Once the mass of pressurization mass is available, the tank required to contain it can be sized according to spherical pressure vessel relations. Specifically, the stress in a spherical pressure vessel can be approximated with membrane theory, in absence of external accelerations. Under these assumptions and with a structural margin of 2.0 to conservatively account for external accelerations, the required thickness of the tank becomes:

$$t_{press,tank} = 2.0 \cdot \frac{p_{press} \cdot R_{press,tank}}{2 \cdot \sigma_{yield,Ti}}$$

and thus its mass can be computed as:

$$M_{press,tank} = 4 \cdot \pi \cdot R_{press,tank}^2 \cdot t_{press,tank} \cdot \rho_{Ti}$$

where only Titanium alloy 6Al4V is considered as possible tank material within V1 models ($\sigma_{yield}=830\ MPa$, $\rho=4510\ kg/m^3$) and the radius of the spherical tank is directly obtained from the pressurization gas mass and density.

4.2.6.5. Igniter

Whereas for LREs the ignition system's mass is negligible, SP igniters can reach up to several hundreds of kg in large motors, covering up to 1-2% of the motor's total inert mass. Though this does not constitute a relevant percentage, simple historical based regressions were developed to account for the igniter's mass based on the available data for the SRMs of Ariane 5 and VEGA, shown in **Table 12**. The free volume of the internal perforation of the solid grain was selected as the independent variable of the regression, as suggested by Sutton (**[86]**). Data on in the internal volume were computed on the basis of available geometric information, whereas igniter's masses were taken from freely available web sources.

SRM	Application	Internal volume [m^3]	Igniter's mass [kg]
P241	Ariane 5 boosters	40.93	315.0
P80	VEGA 1st stage	10.23	130.0
Z23	VEGA 2nd stage	3.38	34.3
Z9	VEGA 3rd stage	0.98	14.7

Table 12: Available data for internal volume and igniter's mass of the SRMs of Ariane 5 and VEGA launchers.

Linear, quadratic and power law regressions of the available data are presented in **Figure 30**. The following power law curve best fits Ariane 5 and VEGA's data, and was therefore implemented in V1 Propulsion models:

$$M_{igniter} = 20.62 \cdot Vol_{cavity}^{0.7368}$$

An alternative formulation is reported in **[86]**, but this is referred to small igniters for orbital kick motors, hence results in a largely underestimated mass for booster applications.

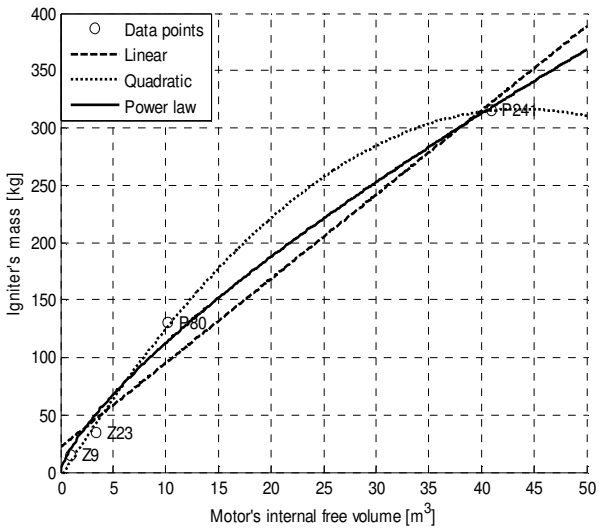


Figure 30: Linear, quadratic and power law fits of the SP igniter's mass as function of the motor's internal free volume.

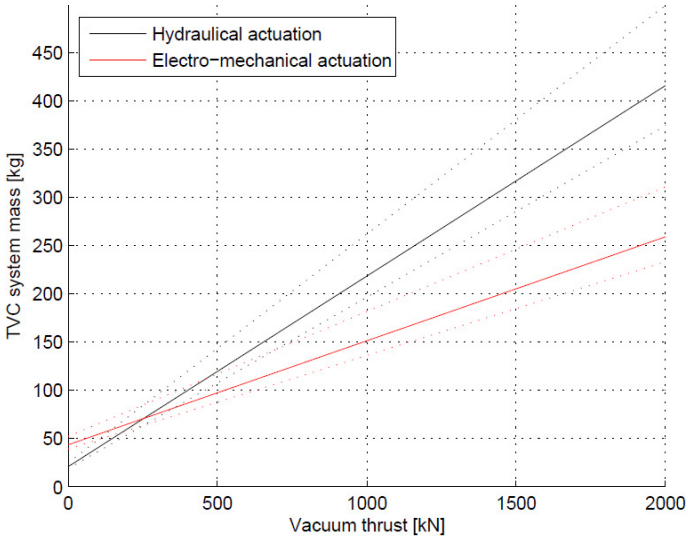


Figure 31: linear regressions of $M_{TVC}(T_{vac})$ for hydraulic and electro-mechanical TVC, as taken from [99]. Plain: $\delta_{TVC,max}$=6 deg. Dotted lines: upper/lower bound for $\delta_{TVC,max}$=3 and 12 deg.

4.2.6.6. TVC system

The mass of the TVC system $\boldsymbol{M_{TVC}}$ is typically not included in the mass figures provided by the manufacturers or other web sources, hence has to be estimated separately from the main engine's mass. $\boldsymbol{M_{TVC}}$ depends on the type of actuation, which is determined by the integer optimization variable ***TVC*** as follows:

- ***TVC=0***: no TVC system, which is possible only for boosters which can be strapped on the core without need of thrust vectoring. Although this is in theory always possible for parallel launcher configurations, it is often not advantageous in terms of controllability.
- ***TVC=1***: TVC system with hydraulic actuation, which was the common choice in most small and large rocket engines in the past.
- ***TVC=2***: TVC system with electro-mechanic actuation, which is currently preferred over hydraulic actuation in several new engine applications. For example, all four stages of VEGA employ newly developed electro-mechanic systems, which result in significant mass saving.

In both cases of TVC=1 and TVC=2, $\boldsymbol{M_{TVC}}$ may be approximately considered to be dependant only on the vacuum thrust of the engine and on the maximum actuation angle. Hence, the following linear WERs for the two different technologies were directly taken from:

$$\begin{cases} M_{TVC} = 0.1976 \cdot T_{nom} + 20.922 & if \;\; TVC = 1 \\ M_{TVC} = 0.1078 \cdot T_{nom} + 43.702 & if \;\; TVC = 2 \end{cases}$$

These are applicable to both LP and SP engines. As clear from the graphical representation in Figure 31, electrical actuation is always favourable except for very small thrust levels, although the trade-off also involves cost issues as shown in **Paragraph 4.7.1**. The maximum actuation angle $\boldsymbol{\delta_{TVC}}$ defines the controllability margin and has to be reflected in the TVC mass to ensure a fair trade-off. This is achieved by shifting the values obtained with the HLLV reference regressions by a percentage that depends on $\boldsymbol{\delta_{TVC}}$. A linear variation is assumed, considering 6 deg as the reference value and varying the TVC mass from 90% to 120% from 3 to 12 deg, with resulting lower and upper bounds also shown in Figure 31.

4.3. Geometry

Even at the conceptual level, an analysis of the external geometry of the launch vehicle is necessary, as it allows generating the geometric model for the aerodynamic analysis and is useful for user visualization of the design solutions. Internal geometry issues however mostly do not need to be addressed, except for the verification of basic constraints such as engines fitting within the external diameter. The Geometry discipline therefore has three main purposes within V1 models:

1. Complementing the geometric information coming from user (payload), optimization variables (architecture) and propulsion analysis (engine and tanks/grain) with the remaining components to build-up the overall external geometry of the launch vehicle, to be used as input for the Aerodynamics discipline.
2. Verifying the launcher's geometric constraints, both related to external and internal geometry.
3. Visualizing the launcher's external geometry for the user to inspect the output design solutions.

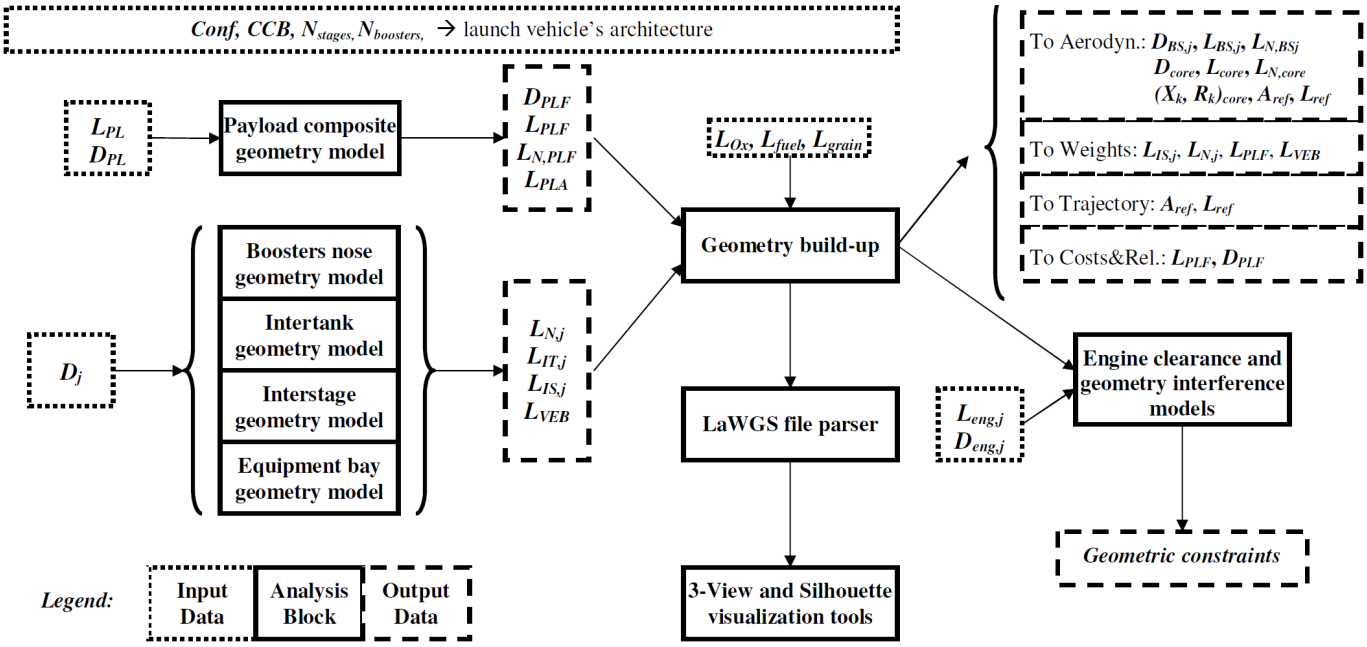


Figure 32: Geometry analysis flow for a generic launch vehicle, including geometry visualization.

The analysis flow is schematically represented in **Figure 32**, showing a first layer of geometry models estimating the dimensions of the payload and boosters nose fairing, of the interstages and intertanks per each stage/booster, and of the upper stage's equipment bay for avionics and EPS. Then, the overall launcher's external geometry can be generated and passed to the Aerodynamics module. Finally, engines' clearance as well as other geometric interferences can be evaluated, computing the values of the constraints violations.

When the Geometry analysis is executed within a MDO process, there is no need of visualizing the geometry, since the user cannot inspect all the investigated design solutions. On the contrary, at the end of an optimization run or after a given number of iterations, geometry visualization is particularly useful to analyze the currently optimal design. Hence, a geometry description file in the LaWGS format is generated from the external geometry information, which is then used by the general purpose 3-View and Silhouette tools for three-dimensional visualization.

4.3.1. Components geometry models

Payload fairing:

The payload fairing is modelled with a power law nose section, plus a cylindrical section containing payload and adapter, plus a conical interface if the upper stage has a different diameter.

The power law employed for the nose section has the following equation: $r/R_{ogive} = \left(x/L_{ogive} \right)^n$, where r is the radius at a distance x from the nose tip, R_{ogive} is the radius at the base and L_{ogive} is the ogive's length. Constant values of the power law exponent n and of the L_{ogive}/R_{ogive} ratio are assumed, with numerical values obtained from an analysis of existing PLFs: Ariane-5 short and long, VEGA, Delta IV Type 1, 2, 3 and 4, Soyuz Type-ST and Type-5, Falcon 1 and Falcon 9. Excluding VEGA's PLF which is significantly more slender, the L_{ogive}/R_{ogive} ratio is always in the range [2.23; 2.37]. The average $L_{ogive}/R_{ogive}=2.313$ and $n=1/2.3$ were therefore selected, resulting in the adimensional geometry of **Figure 33**.

It is assumed that the payload envelope consists of a cylinder of length equal to the cylindrical section of the fairing and of diameter reduced to allow for tolerances and dynamic oscillations. A ratio of external diameter to payload envelope diameter equal to $k_{DPL}=1.12$ is taken, as the average of the values of Ariane 5, VEGA, and other 5 Russian and 6 American launchers, all with ratios ranging from 0.85 and 0.92.

Finally, an additional length $L_{PLA}=K_{PLA} \cdot D_{PL}$ needs to be added to the cylindrical section for the PayLoad Adapter (PLA), with $K_{PLA}=0.15$ from an average of Ariane-5 and VEGA adapters. If upper stage and fairing have different diameters, the PLA is assumed to start at the bottom of the cylindrical section and continue downwards inside the conical connection with the upper stage. The remaining length of the conical connection is assumed to be part of the upper stage. In summary, given the payload diameter and length as mission requirements, D_{PLF} and L_{PLF} are computed as:

$$\begin{cases} D_{PLF} = k_{DPL} \cdot D_{PL} \\ L_{PLF} = L_{N,PLF} + L_{PL} + L_{PLA} = 0.5 \cdot \left(L_{ogive} / R_{ogive} \right) \cdot D_{PL} + L_{PL} + k_{PLA} \cdot D_{PL} \end{cases}$$

Boosters nose ogive:

The boosters' nose ogive is represented with the same power law geometry of the fairing but a lower $L_{ogive}/R_{ogive}=1.0$, as represented in **Figure 33**. This results in a boosters nose length $L_{N,BSj}=D_{BSj}$. Power law nose geometry, employed for example in Delta IV Heavy or Falcon 9 Heavy, was chosen instead of an Ariane-5 like truncated cone configuration because the aerodynamic code Missile DATCOM only allows axi-symmetric shapes.

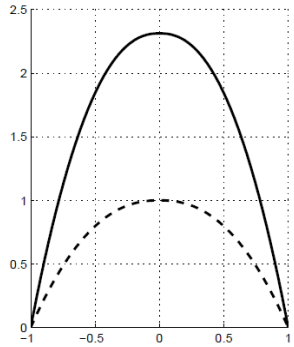


Figure 33: Adimensional geometry of the nose ogive of payload fairing and boosters

Stages/Boosters intertank:

An intertank section needs to be added to the length of oxidizer and fuel tanks only in case of LP systems with separate bulkhead tanks, which is identified by the integer variable $TT=1$. In this case, the intertank's length is taken equal to the length of the ellipsoidal domes of the Ox/Fuel tanks ($K_{IT}=0.3$), in order to provide sufficient margin. For any j-th stage or boosters, total tanks length is thus written as: $L_{tanks,j} = L_{Ox,j} + L_{fuel,j} + L_{IT,j}$ with $\begin{cases} L_{IT,j} = K_{IT} \cdot D_j & if\ (Prop_j \neq 4\ and\ TT_j = 1) \\ L_{IT,j} = 0 & otherwise \end{cases}$

Interstages:

Interstages between each stage and the following are modelled as cylindrical or conical segments depending on the value of the boolean variable $D_{const,j}$. Note that j-th interstage is assumed to connect j-th and (j+1)-th stages (the interstage between the upper stage and the payload fairing is the VEB, whose geometry is described below) and to be entirely jettisoned together with j-th stage. The interstage starts at the top of the upper tank or solid grain and extends up to the exhaust section of the nozzle of the stage above. It has the functions of containing separation mechanisms and other auxiliary subsystems as well as of providing sufficient margin for interferences, for a total length assumed to be $L_{IS,j}=K_{IS} \cdot D_j$, with $K_{IS}=0.20$.

If $D_j \neq D_{j+1}$, a constant cone angle is assumed, equal respectively to $\delta_{IS1}=16.4\ deg$ and $\delta_{IS2}=12.4\ deg$ for $D_{j+1}<D_j$ and $D_{j+1}>D_j$, as directly taken from VEGA. Within this model, the length of the conical section does not in general correspond to the length of the interstage. The conical section length is thus referred to as *nose length* for j-th stage, and is determined by the difference between D_j and D_{j+1}. The part of conical section exceeding the interstage length is considered to be part of (j+1)-th stage, not being jettisoned with the interstage.

Equipment bay:

For conceptual design purposes, avionics, EPS and other equipments included in lower stages and boosters can be neglected both in terms of mass and volume. However, this is not realistic for upper stages, since a relevant percentage of their volume and mass is usually allocated for the subsystems compartment, or VEB. Since accurate breakdown of each subsystem is not within the scope of a conceptual analysis, a constant length-to-diameter ratio is assumed for the VEB equal to $K_{VEB}=0.287$ from Ariane 5. As for lower stages, the upper stage's nose length is computed with a constant cone angle $\delta_{VEB}=30\ deg$ if the upper stage has a different diameter with respect to the PLF, and is instead taken equal to L_{VEB} otherwise. In summary, interstage and nose length for j-th lower stage as well as for the VEB are written as:

$$L_{IS,j} = K_{IS} \cdot D_j \qquad L_{VEB} = K_{VEB} \cdot D_{US}$$

$$L_{N,j} = \begin{cases} L_{IS,j} & if\ \ D_j = D_{j+1} \\ 0.5 \cdot \left(D_j - D_{j+1} \right) / tan(\delta_{IS1}) & if\ \ D_j > D_{j+1} \\ 0.5 \cdot \left(D_{j+1} - D_j \right) / tan(\delta_{IS2}) & if\ \ D_j < D_{j+1} \end{cases} \qquad L_{N,VEB} = \begin{cases} L_{VEB} & if\ \ D_{US} = D_{PLF} \\ 0.5 \cdot \left(D_{PLF} - D_{US} \right) / tan(\delta_{VEB}) & if\ \ D_{US} < D_{PLF} \\ 0.5 \cdot \left(D_{US} - D_{PLF} \right) / tan(\delta_{VEB}) & if\ \ D_{US} > D_{PLF} \end{cases}$$

4.3.2. Launcher's geometry build-up

All the assumptions defined in the previous paragraph are graphically shown in **Figure 34**, which reports a sketch of a generic launcher including 3 LP stages (the first with separate tanks, the other two with common bulkhead tanks) and PL fairing with different diameters as well as one set of SP boosters.

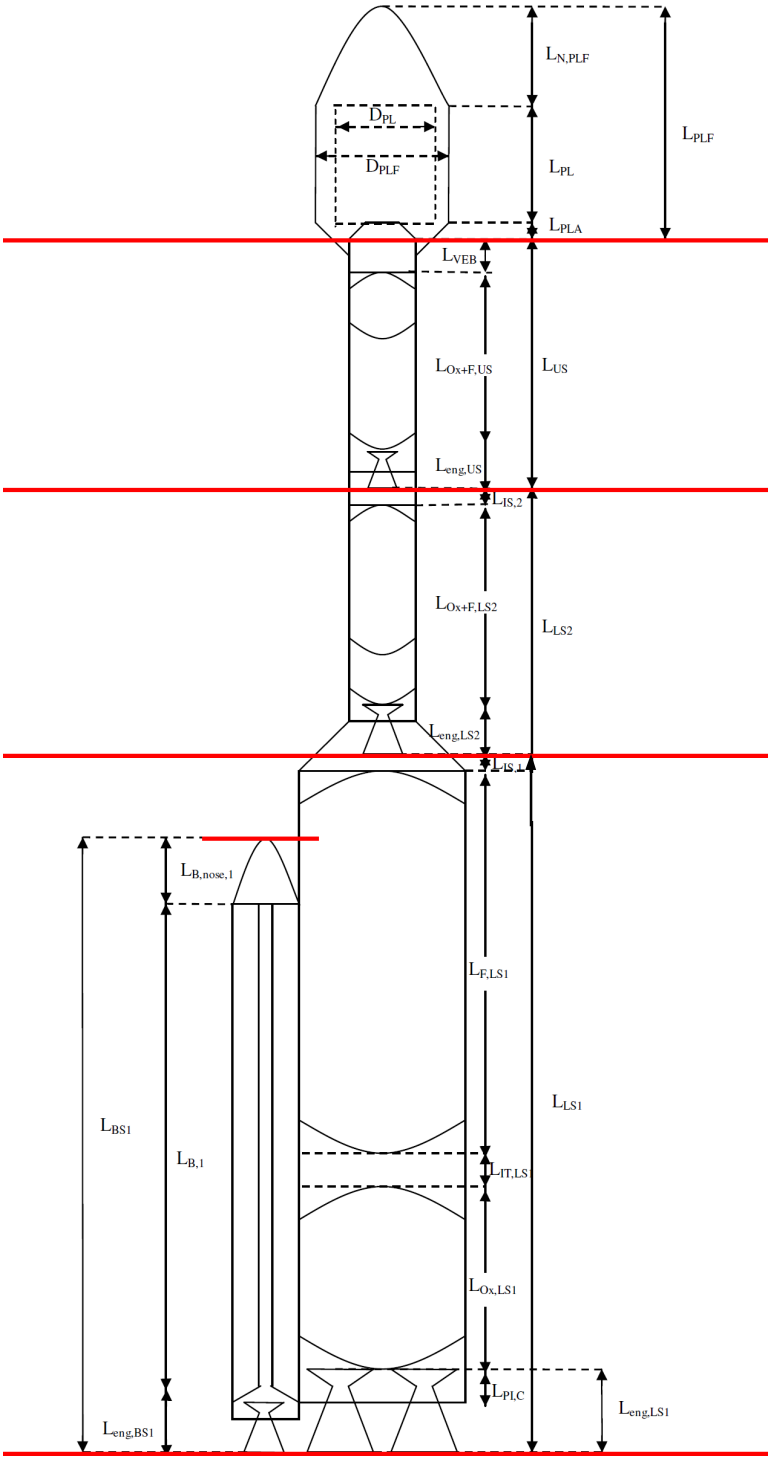


Figure 34: Sketch of a generic launch vehicle's geometry, showing the main geometrical parameters and related assumptions.

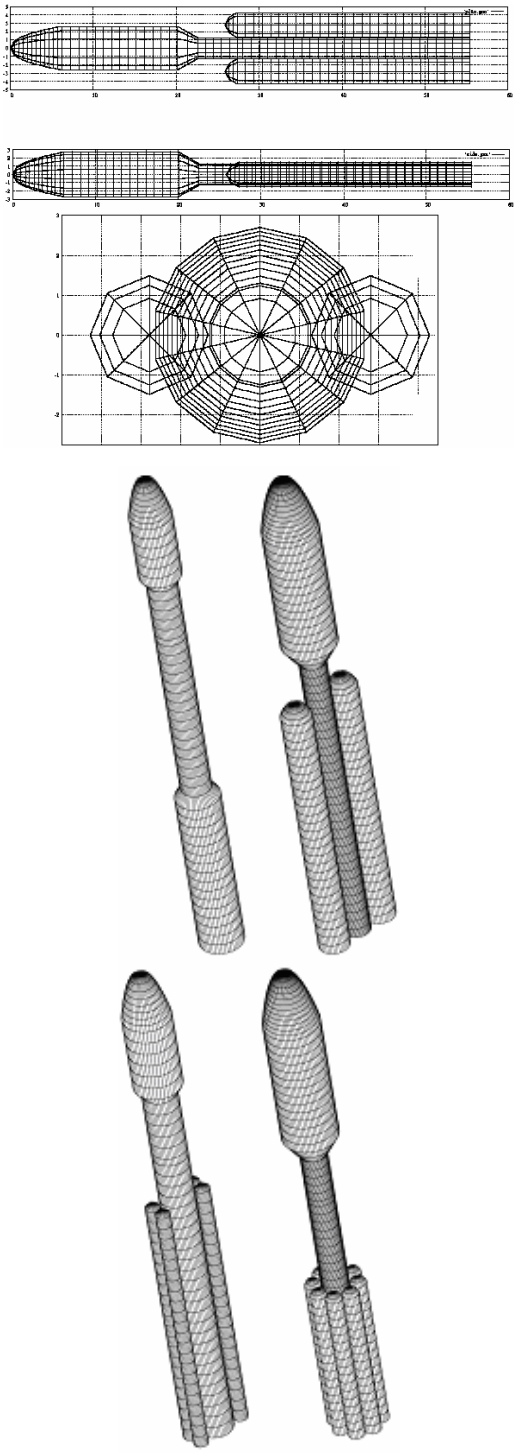


Figure 35: Examples of 2D/3D geometry for different launchers visualized by 3-View/Silhouette.

As already mentioned, the objective of the Geometry discipline within a MDA is to provide a complete description of the external geometry to be passed to the Aerodynamics discipline. For this purpose, boosters and cores with continuous diameters only require three parameters: the diameter and total length of the cylindrical section, and the length of the nose ogive (assuming that the power law exponent ***n*** is kept constant). These are all already available from the previous paragraph, except for the total length of the core and boosters, which can be derived for generic launcher configurations as:

$$\begin{cases} L_{core} = \sum_{j=1}^{N_{stages-1}} \left(L_{LS_j}\right) + L_{US} + L_{PLF} \quad with \quad \begin{cases} L_{LS_j} = L_{eng,LS_j} + \left(L_{tanks,LS_j} \ or \ L_{grain,LS_j}\right) + L_{IS,LS_j} \\ L_{US} = L_{eng,US} + \left(L_{tanks,US} \ or \ L_{grain,US}\right) + L_{VEB} \end{cases} \\ L_{BS_j} = L_{eng,BS_j} + \left(L_{tanks,BS_j} \ or \ L_{grain,BS_j}\right) + L_{N,BS_j} \end{cases}$$

In case the diameter of the core is instead not constant, the diameter variation along the longitudinal axis needs to be provided in terms of radius $\boldsymbol{R_k}$ and coordinate $\boldsymbol{X_k}$ of N stations distributed along the total length of the core. Note that the $\boldsymbol{X}$ coordinate is intended as the distance computed downward from the PLF's nose tip. This holds throughout the whole MDA cycle, so that all gravity, aerodynamic and thrust moments will be referred to the same point in the weights, aerodynamic and controllability analyses. The vector of geometric data $\boldsymbol{(X_k,R_k)}$ for variable diameter cores is computed within the geometry build-up block, starting from diameter $\boldsymbol{D}$, total length $\boldsymbol{L}$, interstage length $\boldsymbol{L_{IS}}$ and nose length $\boldsymbol{L_N}$ of each of the LS, of the US and of the PLF.

Finally, the aerodynamic reference area and length are defined by the diameters of the core and boosters as:

$$\begin{cases} L_{ref} = D_{core} = max(D_{LS_j}, D_{US}, D_{PLF}) \\ A_{ref} = A_{ref,core} + A_{ref,boosters} = \frac{\pi \cdot D_{core}^2}{4} + N_{bpset,1}\frac{\pi \cdot D_{BS_1}^2}{4} + N_{bpset,2}\frac{\pi \cdot D_{BS_2}^2}{4} \end{cases}$$

4.3.3. Launcher's geometric constraints

With all geometric information available, several interference constraints can be checked:

- *Engines clearance*: the diameter of the engines assembly $\boldsymbol{D_{EA}}$ must be lower than the stage/booster diameter. In case of multiple engines, the total diameter $\boldsymbol{D_{EA}}$ defined by $\boldsymbol{N_{eng}}$ engines each with diameter $\boldsymbol{D_{eng}}$ needs to be computed. The configurations shown in **Figure 36** were assumed, deriving the maximum engine radius for a stage unit radius and the surface filling efficiency as reported in **Table 13**. Note that the gimbal requirement ($\boldsymbol{\delta_{TVC}}$) is not considered in the engine clearance constraint, because when the engines need to be oriented during the firing they are allowed to protrude out of the stage length.
- *Boosters' length*: for each set of boosters, the total length must be lower than the first stage length.
- *Boosters' interference*: there must be no interference between boosters in circumferential configuration. For laterally arranged boosters, attachment points must be located within 40 deg of angular distance from each other
- *Diameter discontinuities*: there must not be more than two diameter discontinuities along the core, with a maximum difference in diameter between consecutive stages not exceeding 50% of the larger diameter.

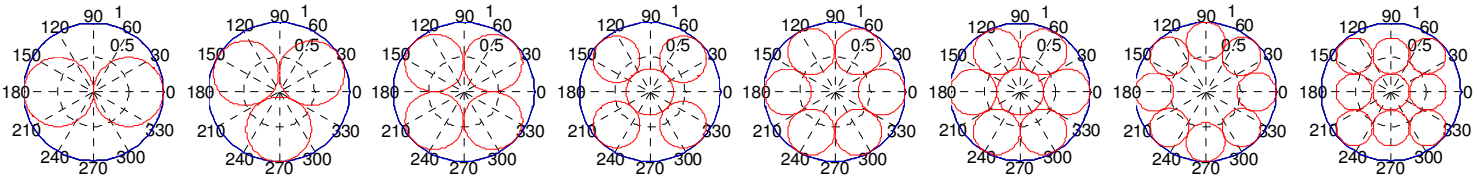


Figure 36: Geometrical configuration for engine assemblies with $N_{eng}>1$

N_{eng}	1	2	3	4	5	6	7	8	9
Max R_{eng}	1.00	0.50	0.47	0.41	0.33	0.33	0.33	0.28	0.26
$\eta_{surface}$	1.00	0.50	0.65	0.69	0.56	0.67	0.78	0.61	0.61

Table 13: Maximum engine radius for unitary external radius and surface filling efficiency for engine configurations in Figure 36.

4.3.4. LaWGS file generation and geometry visualization

For generic MDA processes, it is useful to define a common geometry description standard which can be used by as many disciplinary analyses as possible. Langley Wireframe Geometry Standard (LaWGS) was selected for this purpose, due to its simplicity and generality of the shapes it is capable of describing. Moreover, several open source legacy codes for both geometry visualization and aerodynamic analysis require a LaWGS compliant input file, such as the tools distributed within the Public Domain Aeronautical Software (PDAS)[30] collection.

A C++ script was therefore developed, which translates the geometric information to the LaWGS format, writing a ***.wgs*** file afterwards read by two visualization tools included in PDAS and called by SVAGO: ***3-View*** and ***Silhouette***. Although LaWGS file generation and geometry plotting with 3-View/Silhouette is not required for the MDA execution, it useful for the user to visualize the design solutions to better interact with the optimization process. Although Missile DATCOM

[30] Available at www.pdas.com

aerodynamic code has its own input file format (see **Paragraph 4.4**), .wgs files may be reused for future developments of the MDA, for example following the introduction of panel codes such as PDAS' PANAIR, which does indeed require a LaWGS input file.

NASA TM 85767 (**[102]**) defined in 1985 the LaWGS format as *"... a format for describing configuration geometry with discrete points. These points are coordinates of the locus of points for contour lines over the configuration. In the LaWGS context, a contour line can be thought of as a set of points that when connected by straight lines will follow the contour of the object. Additionally, when respective points on all adjoining contour lines of the object are similarly connected, the mesh or wireframe object is created. Thus a LaWGS file consists of coordinates of the sets of contour points that are the nodes for this wireframe structure."*

All details of LaWGS standard are available in reference **[102]**, hence only few considerations relevant to the geometry description developed for ELVs are reported here:

- Any number of objects can be created within the same file, allowing the definition of arbitrary configurations of both linear and parallel ELVs, covering all architectures defined by the set of variables (***Conf***, ***CCB***, N_{stages}, $N_{boosters}$).
- The number of contours and points per contour may be chosen by the user, but must be consistent throughout each object, so that all contours have the same number of points. Additionally, consistency must be maintained also in such a way that all contours are defined in the same order, e.g. clockwise.
- In order to allow use of panelling methods for future developments of the Aerodynamics discipline, all panel normals must be consistently pointing either inwards or outwards.
- Recommendations reported in **[102]** for "fusiform" objects must be followed due to the axi-symmetric nature of the cylindrical stages and boosters: objects are defined as to contain contour lines starting at the nose and ending at the rear of each component, with the first point on each line the one with lowest z-location and successive points given in a clockwise manner when viewed from the front.
- Translation, local and global symmetry conventions included in the standard are exploited to more compactly define the boosters geometries in all configurations.
- Introduction in the LaWGS file of the internal configuration (e.g. engines, tanks, grain) was also conceived, but it was not deemed a priority to enhance user-software interactions and was thus left as a future development.

PDAS geometry visualization tools, taking as input the ***.wgs*** file, generate both ***.ps*** files for offline visualization and ***.gnu*** files for online GNUPLOT[31] plotting. Specifically, GNUPLOT is called after 3-View and Silhouette within the analysis cycle when required by the user, allowing the following views (see **Figure 35** for some examples):

- 3-View[32]: plan view, side view and rear or front view. This is particularly useful for objects with many hidden lines, for which three orthogonal views are generally less confusing than isometric views.
- Silhouette[33]: static perspective views. No real time rotation of the object can be performed, because Silhouette only writes a 2D representation of a 3D object given the view angles and view point distance.

4.4. Aerodynamics

The aerodynamics disciplinary analysis, represented in **Figure 37**, is extremely simple. It is constituted by the call to the external code Missile DATCOM, two scripts for DATCOM I/O files writing/reading and a synthesis operator to define the interference between stages and boosters. In alternative, user-defined aerodynamic coefficients can be specified using the same format as that generated by DATCOM, so that a launcher with fixed aerodynamic properties can be designed. The three subparagraphs below give an overview of the assumptions behind the aerodynamic analysis, of Missile DATCOM in general and as applied for ELVs' aerodynamic coefficients assessment, and of the boosters-core interference model.

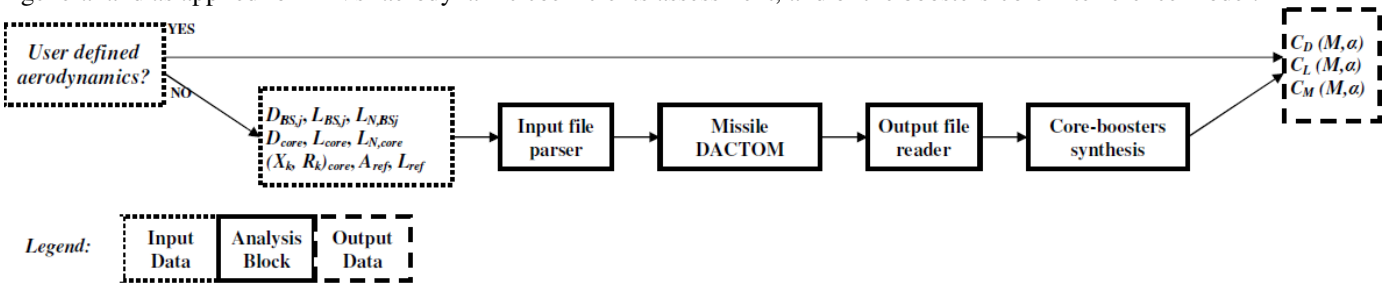


Figure 37: Aerodynamic analysis flow, including selection between user-defined and DATCOM aerodynamics.

[31] www.gnuplot.info
[32] www.pdas.com/3view.html
[33] www.pdas.com/hlp.html

4.4.1. Aerodynamic assumptions

Several assumptions lay at the basis of the aerodynamic models developed for V1, some of which may seem rather rough but were considered acceptable in light of the "limited" influence of the launcher's aerodynamics on the overall performance within conceptual design. The goal is to obtain approximate values of C_D, C_L and C_M coefficients as a function of Mach and total angle of attack, to be used in the Trajectory discipline for the calculation of the aerodynamic forces (drag and lift, for EoM integration) and moments (lateral only, for static controllability verification). As a posteriori verified (see **Chapter 5**), very limited errors on the launcher's performance result from poor aerodynamic coefficients accuracy (average errors in the order of 20 %, with instantaneous values off by as much as 100% for certain flight conditions).

Two main assumptions are worth mentioning:

- Aerodynamic forces are included in the EoMs only up to the first jettison event, either of the boosters or of the first stage, so that only one aerodynamic configuration has to be considered. This simplifies the analysis and largely reduces the computational times. This assumption was justified through a quantitative analysis of the drag-over-thrust (***D/T***) ratio following the first jettison event for existing launchers, assuming thrust, diameter, staging altitude and velocity reported in the respective user manuals and a conservative value C_D=0.5. Calculations for Ariane 5 (GTO), VEGA (LEO), Taurus II (LEO), Soyuz (LEO), Zenit 3SL (GTO), Proton M (LEO), Delta IV (LEO), Delta IV Heavy (LEO) and Atlas V 401 (GTO) showed a maximum ***D/T~0.017*** for both Taurus II and Delta IV. Since this value decreases rapidly as the launcher increases its altitude, neglecting the aerodynamic forces after the first jettison event does not sensibly affect the global performances.
- A null[34] roll angle is always assumed in case of non-axisymmetric booster configurations. This largely simplifies the model for four different reasons: 1) the vehicle's aerodynamics can be approximated as a function only of Mach number ***M*** and total angle of attack α_{tot}; 2) the side force can be neglected in the EoM; 3) the static controllability can be verified considering only a single lateral moment (i.e. pitch=yaw); and 4) the control variables to be optimized are reduced from three to two (pitch and yaw). Classical non-axisymmetric launchers such as Ariane 5 usually fly null roll trajectories, hence this assumption is fully justified for ELVs. Note however that for lifting configurations this is in general not true for various reasons (e.g. Space Shuttle roll program), hence roll should be introduced in the model in order to tackle more complex configurations.

4.4.2. Missile DATCOM

Missile DATCOM is a Fortran computer program initially developed between September 1981 and December 1985 for the USAF by McDonnell Douglas. The goal was the implementation of a large collection of analytical and semi-empirical methods encompassing a wide range of configurations, geometries and flight conditions, and allowing fast estimation of missiles aerodynamics. The final report of the activity (**[103]**, 1988) contains the detailed description and references for all the methods implemented within the first version of the code, whereas the version available for integration within SVAGO is Version 7/97 (distributed with **[104]**). Converted from Fortran 1977 to Fortran 1990 and detailed in a through User's Manual (**[105]**, 1998), Version 7/97 includes additional algorithms to cover aspects not previously modelled. Note that Version 8/08 was also released in 2008, but is not available in the public domain.

The methods implemented within Missile DATCOM were chosen according to their range of applicability, ease of use and automation, accuracy and computational time, with the main applicability requirements shown in Figure 38. Geometries which can be tackled with DATCOM include much more complex configurations with respect to what is needed for the conceptual design of classical ELVs, including for example up to 4 sets of fins/wings and nacelles for air-breathing engines. The applicability in terms of flight conditions also appears well suited, covering with the highest priority the range ***M=[0; 6]***[35] and α_{tot}***=[0; 20] deg***, which include the initial ascent phase of axi-symmetric shapes (boosters are analyzed separately and contributions added to that of the core).

Components build-up approach with analytical and semi-empirical methods was preferred for Missile DATCOM over panel methods, mainly due to the excessive computational times required for such codes to be run (in particular in subsonic conditions). **Figure 39** shows an overview of the main advantages and weaknesses of both approaches.

The only functionalities of Missile DATCOM used for the assessment of the aerodynamic characteristics of ELVs are the body-alone methods for circular cross sections, summarized in Table 14. Specifically, methods used within SVAGO cover the following physical phenomena: 1) skin friction drag; 2) subsonic pressure drag; 3) transonic pressure/wave drag; 4) supersonic wave drag; 5) boat tail pressure/wave drag; 6) nose shape pressure/wave drag; 7) base drag; 8) jet plume interaction effects on drag; 9) angle-of-attack effects on drag; 10) potential normal force and pitching moment; 11) viscous normal force and pitching moment.

[34] For axi-symmetric launchers, the roll angle loses significance. For other configurations (e.g. Ariane 5), null roll is referred to an attitude and velocity such that the total angle of attack plane coincides with the vehicle's symmetry plane.

[35] Taurus II shows the highest Mach at the 1st jettison (M=6.7) among the existing launchers, with all others well below Mach 6.

PARAMETER	SYMBOL	PRIORITY 1	PRIORITY 2	SOURCE
ANGLE OF ATTACK, DEG.	α	$-20\leq\alpha\leq30$	$-180\leq\alpha\leq180$	AEROMECHANICS
ANGLE OF YAW, DEG.	β	$-20\leq\beta\leq20$	$-180\leq\beta\leq180$	SURVEY AND
AERODYNAMIC ROLL, DEG.	ϕ	$0\leq\lvert\phi\rvert\leq45$	$0\leq\lvert\phi\rvert\leq180$	WORLD'S
MACH NUMBER	M	$0\leq M\leq6$	$0\leq M\leq10$	MISSILE
BODY FINENESS RATIO	$(l/d)_B$	$6\leq(l/d)_B\leq20$	$1\leq(l/d)_B\leq30$	SYSTEMS
NOSE FINENESS RATIO	$(l/d)_N$	$.5\leq(l/d)_N\leq5$	$0\leq(l/d)_N\leq7$	
FIN EXPOSED SPAN TO DIAMETER	b/d	$1\leq b/d\leq6$	$0\leq b/d\leq10$	
FIN ASPECT RATIO	AR	$0.6\leq AR\leq4$	$0.1\leq AR\leq10$	
FIN PLANFORM		TRIANGULAR TRAPEZOIDAL	ALL	
WING/TAIL ORIENTATION		IN-LINE	ALL	
CONTROL METHOD		ALL MOVEABLE FIN	ALL	
REYNOLDS NUMBER/FT	R_N	$3\times10^5\leq R_N\leq2\times10^7$	$10^3\leq R_N\leq3\times10^7$	MACH-ALTITUDE
FIN DEFLECTION/INCIDENCE, DEG.	δ	$0\leq\delta\leq30$	$0\leq\delta\leq60$	BOUNDARY
ROLL RATE, RAD/SEC.	p	$0\leq\lvert p\rvert\leq1$	$0\leq\lvert p\rvert\leq6$	MISSILE SYSTEM
PITCH RATE, RAD/SEC.	q	$0\leq\lvert q\rvert\leq1.5$	$0\leq\lvert q\rvert\leq3$	ANALYSIS
YAW RATE, RAD/SEC.	r	$0\leq\lvert r\rvert\leq1.5$	$0\leq\lvert r\rvert\leq3$	
FIN DEFLECTION RATE, RAD/SEC.	$\dot{\delta}$	$0\leq\lvert\dot{\delta}\rvert\leq10$	$0\leq\lvert\dot{\delta}\rvert\leq28$	

Figure 38: Missile DATCOM applicability to different geometric configurations and flight conditions ([103])

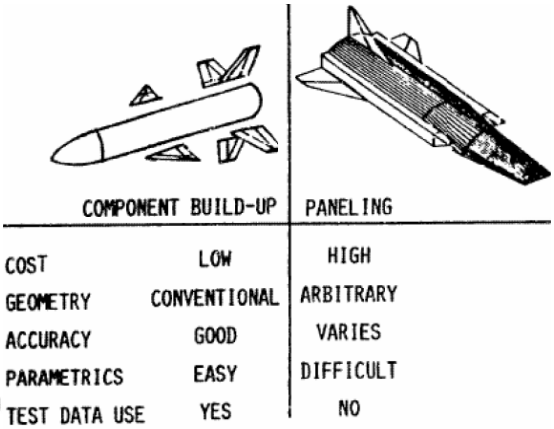


Figure 39: Comparison of components build-up approach (DATCOM) and panels codes ([103]).

Parameter	Subsonic/Transonic (M<1.2)	Supersonic (M>1.2)
Potential Normal Force	Option 1: Nose-cylinder: MBB charts, MBB TN-WE-2-9769 and Boattail: NSWC charts, NSWC-TR-81-156 and Flare: Army charts AMCP 706-280, or Option 2: Slender Body Theory	Option 1 and Option 2: Second Order Shock Expansion, NSWC-TR-81-156, or Van Dyke Hybrid theory, NSWC-TR-81-156 , or Modified Newtonian theory, NASA-TND-176
Viscous Normal Force	Jorgensen viscous crossflow, NASA-TR-R-474 and AEDC-TR-75-124	Jorgensen viscous crossflow, NASA-TR-R-474 and AEDC-TR-75-124
Potential Pitching Moment	Option 1: Nose-cylinder: MBB charts, MBB TN-WE-2-9769 and Boattail: NSWC charts, NSWC-TR-81-156 and Flare: Army charts AMCP 706-280, or Option 2: Slender Body Theory	Option 1 and Option 2: Second Order Shock Expansion, NSWC-TR-81-156, or Van Dyke Hybrid theory, NSWC-TR-81-156 , or Modified Newtonian theory, NASA-TND-176
Viscous Pitching Moment	Jorgensen viscous crossflow, NASA-TR-R-474 and AEDC-TR-75-124	Jorgensen viscous crossflow, NASA-TR-R-474 and AEDC-TR-75-124
Skin Friction Drag	Turbulent: Van Driest II, MDAC West Handbook Laminar: Blasius, Hoerner Fluid Dynamic Drag Roughness: USAF Datcom section 4.1.5.1	Turbulent: Van Driest II, MDAC West Handbook Laminar: Blasius, Hoerner Fluid Dynamic Drag Roughness: USAF Datcom section 4.1.5.1
Pressure/Wave Drag	M<Mcrit: USAF Datcom section 4.2.3.1 M>Mcrit: Transonic area rule, AIAA-90-0280	Second Order Shock Expansion, NSWC-TR-81-156, or Van Dyke Hybrid theory, NSWC-TR-81-156 , or Modified Newtonian theory, NASA-TND-176
Base Drag	Cylinder: NSWC charts, NSWC-TR-92/509 Boattail: NASA method, NASA-TR-R-100 Flare: NSWC charts, NSWC-TR-81-358	Cylinder: NSWC charts, NSWC-TR-92/509 Boattail: NASA method, NASA-TR-R-100 Flare: NSWC charts, NSWC-TR-81-156
Protuberance Drag	M<0.6: Hoerner Fluid Dynamic Drag M>0.6: cubic fairing, AIAA-94-0027	M<5.0 Modified Newtonian theory wit, AIAA-94-0027 M>5.0: Modified Newtonian theory
Axial force at angle of attack	Allen and Perkins Crossflow, NASA TR-1048	Second Order Shock Expansion, NSWC-TR-81-156 Assumed zero for Van Dyke Hybrid and Modified Newtonian theory
Dynamic derivatives	LMSC code, LMSC-D646354 and D646354A Slender Body Theory, AIAA 97-2280	LMSC code, LMSC-D646354 andD646354A Slender Body Theory, AIAA 97-2280
Magnus derivatives	SPIN 73 code, FRL-TR-4588	SPIN 73 code, FRL-TR-4588
Plume effects	not calculated	Chapman Korst model, AIAA 90-0618

Table 14: Missile DATCOM methods for axi-symmetric body-alone aerodynamics, as taken directly from [103]

For all details, geometry/flight regime applicability limitations and typical accuracies of these methods, see **Section 3.1** of reference **[103]**, whereas refer to **[105]** for their practical implementation. Finally, for representative examples of independent validations of Missile DATCOM see **[106]** and **[107]**, where average errors lower than 20% (10% for axial force) are reported. As for CEA, integration of Missile DATCOM in the MDA was realized through input/output files. Although calling DATCOM binary with an interface based on pipe files leads to higher computational times due to hard

drive access, the simplicity of this approach outweighs the efficiency disadvantage, also in light of the small dimensions of both input and output file. DATCOM requires a single input file containing the flight conditions and geometry description for an arbitrary number of cases, and prints up to 7 output files depending on the user's requests. The Fortran 90 code was however modified to write to file the minimum required output - i.e. the database $C_D(M,\alpha_{tot})$, $C_L(M,\alpha_{tot})$, $C_M(M,\alpha_{tot})$ on a grid by default composed of $N_{data,Mach}$=21 Mach points and $N_{data,\alpha tot}$=4 AoA points.

A parsing routine was implemented to write the input file ***for005.dat*** according to DATCOM syntax. For parallel configurations, one DATCOM case is defined for each of the core and booster sets separately, since DATCOM geometry options do not encompass laterally arranged boosters. For each case, the following namelists are written to for005.dat (see DATCOM's manual **[105]** for details on the format):

- *FLTCON*, containing the Mach and AoA values defining the analysis grid. Besides, default approximate altitude values are provided for each Mach number to allow internal computation of the Reynolds number of each flight condition. Although the altitude vs. Mach profile should be fed back from the trajectory, the effect of changes with respect to the assumed profile is negligible.
- *REFQ*, including boundary layer information (conservatively assumed to be turbulent with a Roughness Height Rating of 250.0, common for aerospace surfaces) and CoG position, which is used only as the reference point for the determination of the pitching moment coefficient. Hence, in order to compute the Centre of Pressure (CoP) position directly as distance from the nose tip, a null CoG position is given to DATCOM.
- *AXIBOD*, defining the geometry of axi-symmetric bodies, with two possibilities allowed within DATCOM:
  - *Option 1*, for boosters or core with constant diameter: only the nose shape type (power law) and related parameters (***L/D***, ***n***), length and diameter of the cylindrical section are provided.
  - *Option 2*, for core with discontinuous diameter (e.g.: VEGA, Falcon-9 with large fairing): the set of stations coordinates (X_k, R_k) is provided, together with a vector defining slope discontinuities of such coordinates.

In addition to the above namelists, a number of control cards are set up to provide more control on the aerodynamics analysis (a quantitative justification of these assumption following preliminary testing is reported in SVAGO V1 Verification and Validation Document **[A9]**):

- *DIM M*, to obtain results in SI units.
- *NO LAT*, to avoid computation of lateral-directional derivatives for significant computational time savings.
- *SOSE* and *BNOSE=0*, to select the Second Order Shock Exapansion method to be used for the supersonic (M > 1.2) wave drag estimation, assuming a sharp nose front.
- SAVE and NEXT CASE, to save FLTCON and REFQ input data and start the definition of the boosters cases.

As all these options affect the behaviour of Missile DATCOM, a complete sensitivity analysis with respect to three validation test cases (DNEPR, VEGA and Ariane-5) was executed and is reported in **[A9]**. Two examples of typical DATCOM input files, in particular for VEGA and Ariane 5 ECA, are reported in **Figure 40** for more information on the type of analysis being requested.

```
$FLTCON NALPHA=5.0, NMACH=20.0,
   ALPHA=0.0,2.0,5.0,8.0,12.0,
   MACH=0.20,0.50,0.80,0.85,0.90,0.95,0.99,1.03,1.05,1.07,1.10,
     1.20,1.35,1.55,1.80,2.10,2.50,3.00,4.50,6.00,
   ALT=1000.0,3000.0,5000.0,5500.0,6000.0,6500.0,
     7000.0,7500.0,8000.0,8500.0,9000.0,
     1000.0,12000.0,15000.0,18000.0,22000.0,
     26000.0,31000.0,50000.0,70000.0$
$REFQ XCG=0.0,BLAYER=TURB,RHR=250.0$
$AXIBOD NX=11.0,
   X=0.000,0.760,1.520,2.280,3.040,3.800,7.880,9.920,22.430,
     22.980,34.180,
   R=0.000,0.650,0.870,1.040,1.180,1.300,1.300,0.950,0.950,
     1.500,1.500,
   DISCON=7.0,8.0,9.0,10.0,11.0,
   BNOSE=0.,DEXIT=0.0,BASE=.FALSE.$
 DIM M
 NO LAT
 SOSE
 NEXT CASE
```

```
$FLTCON NALPHA=5.0, NMACH=20.0,
   ALPHA=0.0,2.0,5.0,8.0,12.0,
   MACH=0.20,0.50,0.80,0.85,0.90,0.95,0.99,1.03,1.05,1.07,1.10,
     1.20,1.35,1.55,1.80,2.10,2.50,3.00,4.50,6.00,
   ALT=1000.0,3000.0,5000.0,5500.0,6000.0,6500.0,
     7000.0,7500.0,8000.0,8500.0,9000.0,
     1000.0,12000.0,15000.0,18000.0,22000.0,
     26000.0,31000.0,50000.0,70000.0$
$REFQ XCG=0.0,BLAYER=TURB,RHR=250.0$
$AXIBOD TNOSE=POWER,POWER=0.434783,BNOSE=0.,LNOSE=6.030,DNOSE=5.400,
  LCENTR=46.400,DCENTR=5.400,
  DEXIT=0.000,BASE=.FALSE.$
  DIM M
  NO LAT
  SOSE
  SAVE
  NEXT CASE
  DELETE AXIBOD
 $AXIBOD TNOSE=POWER,POWER=0.434783,BNOSE=0.,LNOSE=3.050,DNOSE=3.050,
   LCENTR=28.110,DCENTR=3.050,
   DEXIT=0.000,BASE=.FALSE.$
  NEXT CASE
```

Figure 40: Examples of DATCOM input file for VEGA (left) and Ariane 5 ECA (right). Two analysis cases are defined for Ariane (core and boosters) both with geometry option 1, whereas a single case with geometry option 2 is defined for VEGA.

4.4.3. Boosters-core interference model

Having been conceived for the analysis of missile configurations, Missile DATCOM is not capable of analyzing launcher geometries with laterally arranged boosters. Two work-around solutions would exist within Missile DATCOM to simulate the presence of boosters, but present drawbacks which led to discard them:

- Use of *air-breathing propulsion inlets*: the implemented methods consider a given air mass flow to be passing through the inlets, hence not providing realistic results for the boosters case. Besides, only up to 4 inlets can be added whereas up to 9 boosters are foreseen, and the methods cover only the supersonic range.
- Use of *protuberances*: this functionality covers small items such as sensors, electrical connections, launch shoes or similar, and is therefore not suited to large boosters. Besides, the only contribution added is that of axial force at zero-lift, neglecting the effect of lift and non zero angle of attack flight.

In light of these weaknesses, none of these two methodologies was implemented. Alternatively, separate cases are run for the core and for each boosters set, obtaining 1/2/3 separate aerodynamic databases depending on the architecture. Accounting for the interference effects between the core and the boosters remains however a complex task, performed through the introduction of arbitrary interference factors that multiply each aerodynamic coefficient before their synthesis in a single database. The interference equation, outlined for C_L but formally identical for C_D and C_M, is the following:

$$C_L(M,\alpha) = \frac{C_{L,c}(M,\alpha)\cdot A_{ref,C} + K_{C_L}\cdot C_{L,B1}(M,\alpha)\cdot A_{ref,B1} + K_{C_L}\cdot C_{L,B2}(M,\alpha)\cdot A_{ref,B2}}{A_{ref,C} + A_{ref,B1} + A_{ref,B2}}$$

where the subscripts C, B1 and B2 refer to the core (stages + fairing), first and second set of boosters. The values of the interference coefficients were calibrated on Ariane 5 ECA aerodynamics, taking the following values: $\boldsymbol{K_{CL}=1.2}$, $\boldsymbol{K_{CD}=0.9}$ and $\boldsymbol{K_{CM}=1.1}$. Note that obtaining the interference factors from a single data point may be questionable, as well as the method of the interference coefficients itself, only being able to provide very rough approximations of the actual launch vehicle's aerodynamic properties. However, this is still in line with the expectable accuracy of the methods within Missile DATCOM and no reasonable alternative methods exist without largely increasing the fidelity of the models, hence this was considered acceptable.

4.5. Weights estimation

Weights analysis has the main purpose of assessing the total inert mass of all launch vehicle's components (stages, fairing, boosters) through the WBS shown in Figure 41. Besides, the longitudinal position of the CoG of each component's inert masses is derived from the detailed WBS. Note that the propellant's CoG will be computed and added at each time instant during the trajectory simulation, in order to obtain the instantaneous CoG position for static controllability verification. The WBS for generic ELVs is rather complex, including propulsion system, structural components and non structural components. WERs were implemented for each of the mass items in Figure 41, excluding those coming directly from the user (PL mass) or from the Propulsion discipline. The resulting analysis flow is straightforward, with all estimation blocks sequentially executed and a final assembly of both the dry weights and the CoG positions, as shown in **Figure 42**. Note how three main types of inputs are required for each WER or sizing relation:

- Propulsion and geometry disciplinary outputs: diameters, lengths, lateral surfaces and volumes of the components
- Trajectory loads, in terms of encountered dynamic pressure, axial acceleration, and heat flux.
- Technological trade-offs discrete optimization variables, such as tanks type and arrangement, structural material, etc.

The implemented WERs were mainly taken from a comprehensive set collected from various sources by Georgia Tech in reference **[100]**. Other WERs developed in the 70's by NASA for the Weight Analysis of Advanced Transportation Systems (WAATS) program (**[108]**) are also applied, with corrective factors occasionally included to account for specific requirements of the developed multidisciplinary model. Analytical sizing relations based on pressure vessels and buckling theory were also investigated for the structural components, especially from the classical Bruhn's text for aircrafts (**[109]**) and its extension to rocket vehicles (**[110]**). Although this approach ensures physics based mass estimation and should provide more accurate mass estimates, it has few important disadvantages with respect to WERs:

- Modelling complexity: to take into account all possible launcher architectures and structural concepts, a complex model with several different analytical sizing relations need to be implemented.
- Load cases definition: a realistic evaluation of the most relevant load cases in flight and on the ground has to be performed to obtain reliable weight information, further complicating the model.
- Introduction of iterative cycle: since loads for the structural sizing directly come from the trajectory, an iterative cycle needs to be set up of the Trajectory and Weights disciplines, requiring much larger computational efforts.

In light of these considerations, WERs were selected for the conceptual level models of V1, with no mass margins since a constant scaling would not affect the optimization process. Technology Reduction Factors (TRF) are nevertheless applied as suggested in **[100]**, for those subsystems for which recent technological advances have sensibly reduced weights with

respect to those used for the WERs development. The next paragraphs detail all WERs and sizing relations implemented for V1 models, as well as the hypotheses at the basis of the dry CoG definition.

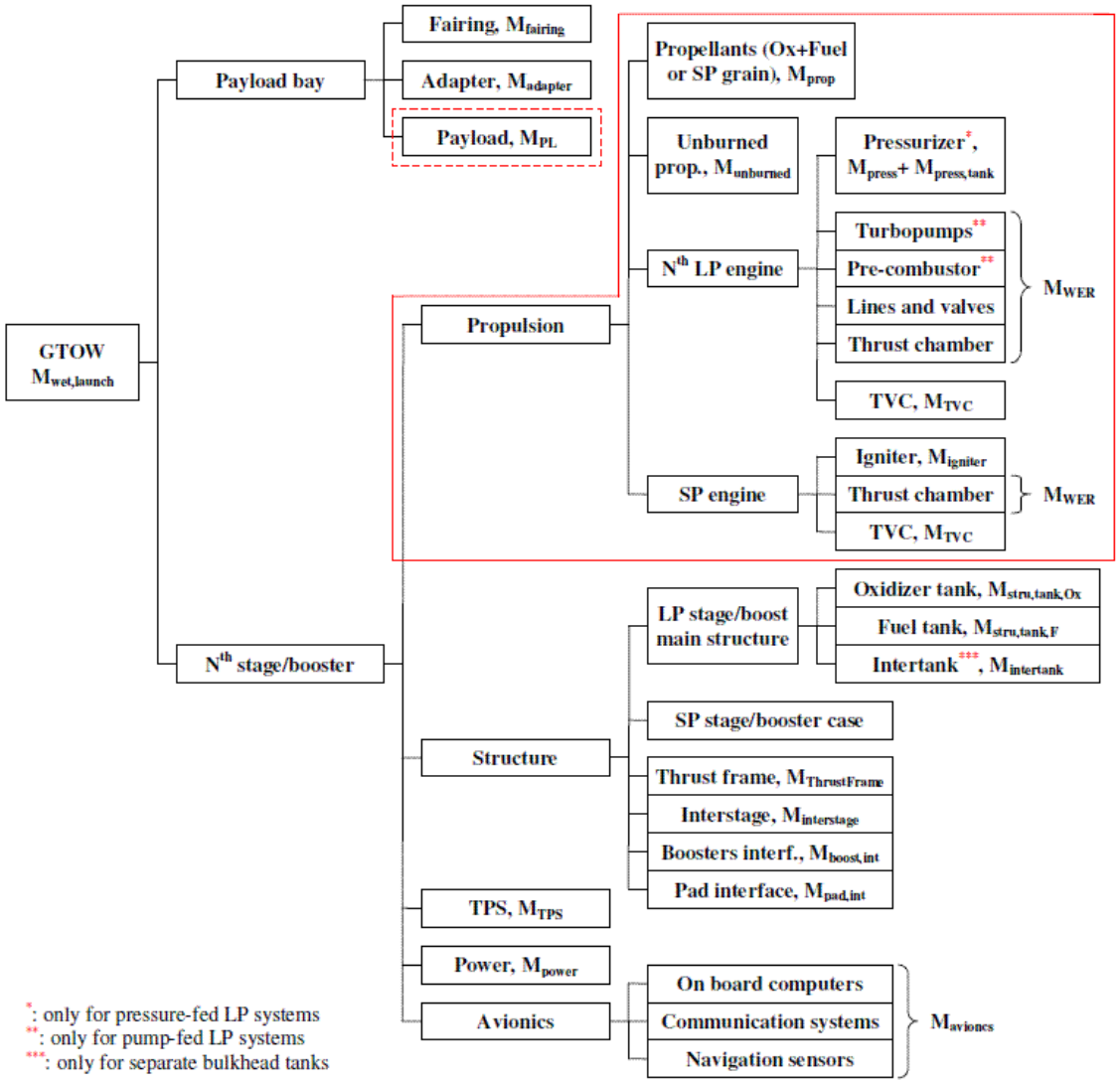


Figure 41: Developed WBS for a generic ELV. Weights provided by the propulsion analysis and by the user are circled.

4.5.1. Payload fairing and payload adapter

The PL fairing mass is obtained through a linear WER (WAATS, **[108]**) of its total external surface S_{PLF}, which was selected against other 3 available alternatives due its better accuracy demonstrated on a set of 30 existing payload fairings:

$$M_{PLF} = a \cdot K_M \cdot K_Q \cdot K_T \cdot S_{PLF} + b$$

where M_{PLF} is the total mass of the PLF in kg, including structure and TPS. To represent the external loads and structural concept, corrective factors are added according to the following variables:

- K_Q, non-linear function of the maximum dynamic pressure encountered along the trajectory $q_{dyn,max}$ (see **Figure 43**).
- K_T, non-linear function of the maximum external wall temperature (see **Figure 43**). The temperature is computed with the Boltzmann law from the maximum heat flux encountered along the trajectory $q_{heat,max}$
- K_M, depending on the structural material: K_M**=1** for state-of-art composite fairings (***SM=2***) and K_M**=1.591** for older Al-alloy based structures (***SM=1***).

As regards to the PL adapter instead, its influence on the global performance is very limited. Hence, simple regressions of the PLA's mass as a function of the supported payload mass were developed, obtaining that a power law curve best fits the experimental data, constituted of 10 PLAs from Ariane 5, VEGA, Soyuz and Atlas families, as shown in **Figure 44**:

$$M_{PLA} = a * (M_{PL})^b \quad with \begin{cases} a = 4.77536e-002 \\ b = 1.01317e+000 \end{cases}$$

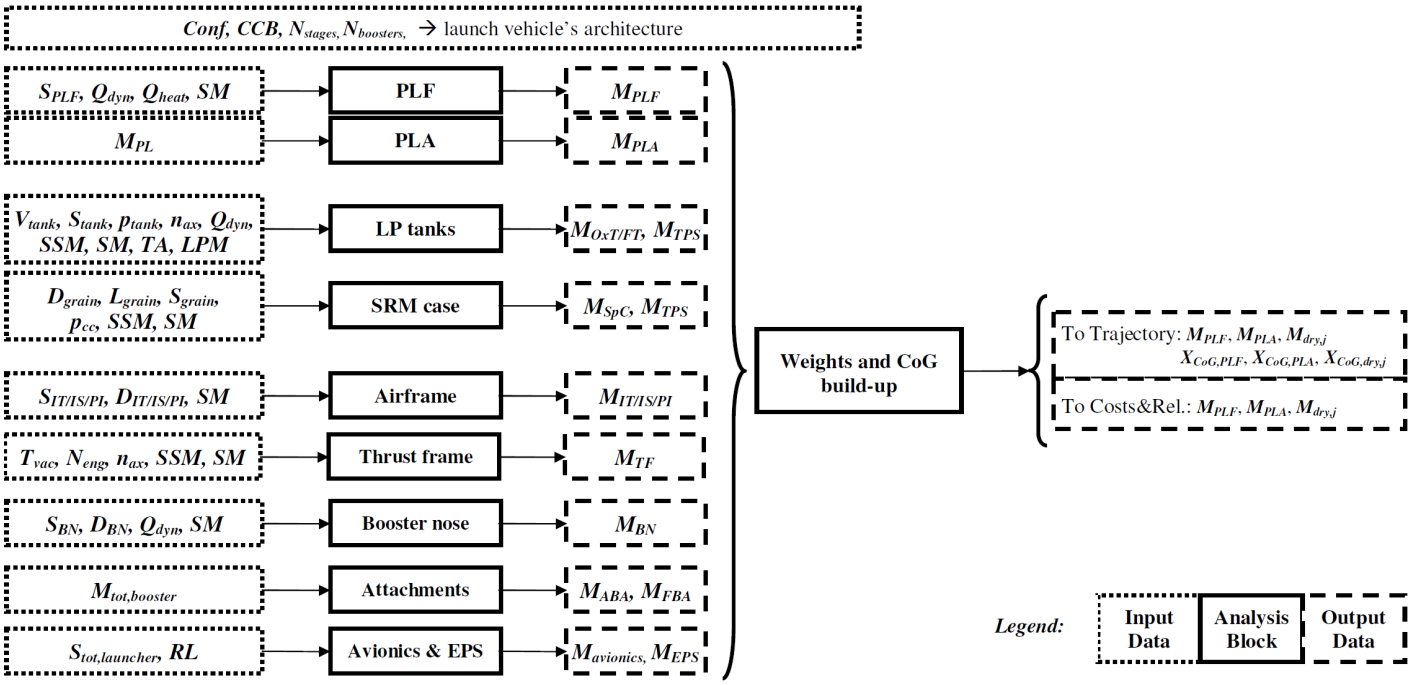


Figure 42: Weights analysis flow, including component WERs and WBS build-up.

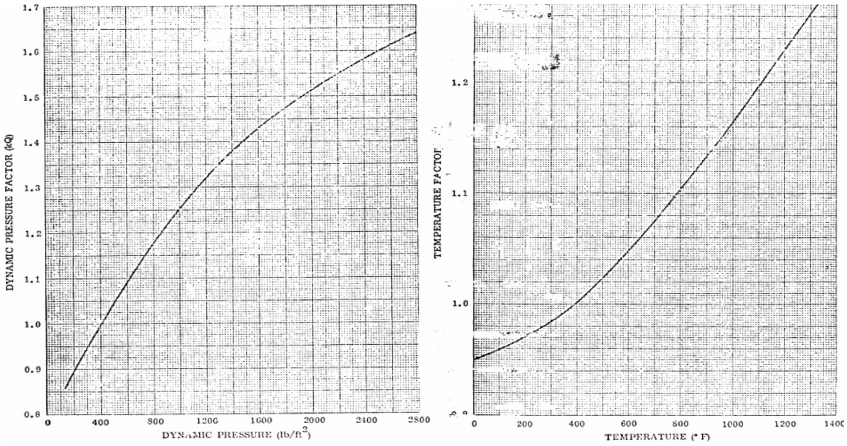


Figure 43: Effect of the dynamic pressure (left) and external wall temperature at stagnation point (right) on the fairing mass, as taken from [108]

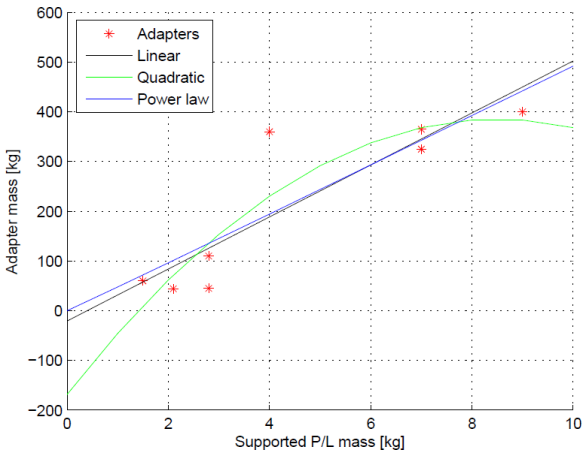


Figure 44: Developed payload adapter mass regression as a function of the supported PL mass.

4.5.2. LP tanks

In case of LP systems, fuel and oxidizer tanks usually constitute the largest part of the structural mass. A rather complex WER was synthesized from **Ref.6** of **[100]** and from **[108]**, which allows for sensibly better accuracy with respect to other available regressions. Its basic form is linear regression of the tank volume for fuels and a slightly non-linear power law regression of the tank volume for oxidizers:

$$\begin{cases} M_{FT} = \prod_{j=1}^{6} k_j \cdot (V_F \cdot 35.315) \cdot 0.4856 + 800) \cdot 0.4536 \\ M_{OxT} = \prod_{j=1}^{6} k_j \cdot (V_{Ox} \cdot 35.315)^{1.04} \cdot 1.0850 + 700) \cdot 0.4536 \end{cases} \quad with \begin{cases} k_1 = f_1(SM) \\ k_2 = f_2(TT) \\ k_3 = f_3(LPM) \\ k_4 = f_4(Q_{dyn}) \\ k_5 = f_5(SSM, n_{ax,max}) \\ k_6 = f_6(p_{tanks}) \end{cases}$$

where k_1 to k_6 are coefficients used to account for relevant load parameters and discrete optimization variables:

- *Dynamic pressure*: the max Q_{dyn} encountered along the trajectory is reflected in a factor $k_4 = q_{dyn,max}^{0.16} / k_{4,ref}$ taken from **[108]** and represented in **Figure 45**, with $k_{4,ref}$**=5.7669** arbitrarily tuned on Ariane- 5's max Q_{dyn} of 57000 Pa.

- *Axial acceleration*: the maximum axial g load is also reflected in a factor $k_5 = (SSM \cdot \min(n_{ax,max}, n_{ax,max,PL}))^{0.15} / k_{5,ref}$ taken from **[108]**, with $\boldsymbol{k_{5,ref}=1.3724}$ tuned for Ariane-5's 5.5 g of axial acceleration (**Figure 45**).
- *Structural material*: if ***SM=2***, use of advanced Al-Li alloy is assumed, involving a TRF weight correction $\boldsymbol{k_1=0.9}$.
- *Tanks type*: if ***TT=2*** (common bulkhead), the masses of fuel and oxidizer tanks are still computed separately, but a coefficient $k_2 = S_{tot} - 1.5 \cdot S_{dome} / S_{tot}$ is introduced to consider the mass saving due to reduction in domes mass.
- *Horizontal/vertical processing*: if ***LPM=1*** (horizontal integration), a coefficient $\boldsymbol{k_3}$ which linearly varies in the range ***[1-1.3]*** for length-over-diameter ratios in the range ***[1-10]*** is applied.
- *Tank's pressure*: a higher tanks pressure $\boldsymbol{p_{tanks}}$ is reflected in heavier tanks through a corrective factor $\boldsymbol{k_6}$ which is a linear function of the tanks pressure: $k_6 = 1.3012 + 1.4359e-6 \cdot p_{tanks} / k_{6,ref}$ with $\boldsymbol{k_{6,ref}=2.7862}$.

Besides the structural mass obtained with the above WER, a thermal insulation layer must be added in case of cryogenic systems, for which its mass may reach considerable percentages of the total tank's mass (typically up to 5%). The TPS mass is therefore computed through a WER taken from **Ref.6** of **[100]**, as a linear function of the tank external surface:

$$M_{TPS,OxT/FT} = k_{ins} \cdot S_{OxT/FT}$$

with $\boldsymbol{k_{ins}=0.9765}$ for LOx tanks and $\boldsymbol{k_{ins}=1.2695}$ for LH_2 tanks. A null thermal insulation mass is assumed for storable propellants.

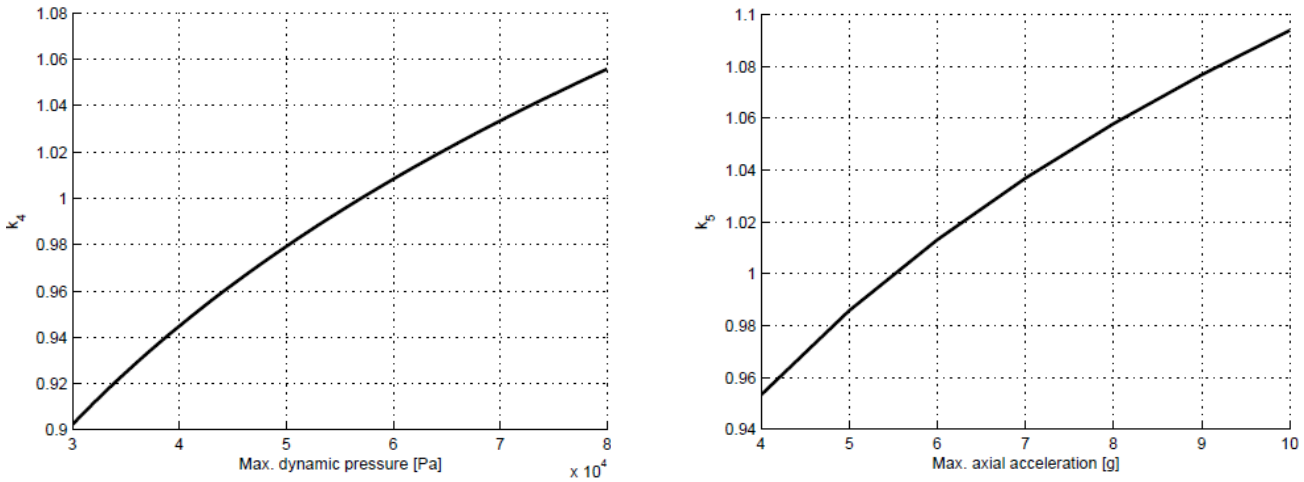


Figure 45: LP tanks corrective factors k_4 for dynamic pressure (left) and k_5 for axial acceleration (right), as taken from [108]

4.5.3. SP case

For SP cases, it was found that the classical cylindrical pressure vessels relation allows for a better accuracy with respect to WERs. This is equivalent to the assumption that the main structural mass is entirely determined by the chamber pressure. All other flight loads are therefore neglected, implying that compression stresses due to flight loads can be sustained by the case sized for internal pressure. Conservatively assuming spherical domes for the solid grain, simple analysis of cylindrical pressure vessels leads to the definition of the SP grain case's mass as:

$$M_{SpC} = 2 \cdot \pi \cdot R_{grain}^2 \cdot (R_{grain} + L_{grain}) \cdot SSM \cdot p_{cc} \cdot (\rho / \sigma(SM))$$

where $\boldsymbol{\rho/\sigma}$ is determined by the material's properties, considering an Al-7075 alloy for ***SM=1*** ($\boldsymbol{\rho=2730\ kg/m^3}$, $\boldsymbol{\sigma=505\ MPa}$) and a high strength Carbon Fibre Reinforced Plastic (CFRP) for ***SM=2***. ($\boldsymbol{\rho=1600\ kg/m^3}$, $\boldsymbol{\sigma=800\ MPa}$ assuming a 54.7 degrees winding angle which is optimal for composite pressure vessels according to **[111]**).

Due to the limited impact of the thermal insulation on the total SP stage/booster mass, an insulation layer with constant thickness is assumed. Quantitative data are taken from Ariane 5 P-241 boosters, which use a 3 mm thick layer of silicon rubber.

4.5.4. Intertanks, interstages and pad interface structures

Other flight structures which need to withstand the primary flight loads - such as axial and lateral accelerations - are the intertank, interstage and pad interface structures. Although their location and function in the launch vehicle are different, these structures are similar in construction, being subjected to the same type of loads and usually failing for compression loads. Hence, the same type of WER was used for all these components, modifying the parameters to account for different conditions, as for example depending on the stage number.

The WER used for intertank, interstages and pad interface structures is a bidimensional linear function of the lateral surface and the diameter, as taken from **Ref.6** of **[100]**, in the form:

$$M_{IT/IS/PI} = k_{SM} \cdot k_1 \cdot S_{IT/IS/PI} \cdot (D_{IT/IS/PI} \cdot 3.2808)^{k_2}$$

where k_{SM} is a corrective factor for the structural material, again taken from **Ref.6** of **[100]** as $k_{SM}=1$ for classical Al-alloy based structures (***SM=1***) and $k_{SM}=0.7$ for advanced composite-based structure (***SM=2***).

Note that the surface of the intertank is computed from a simple cylinder of length equal to lengths of the ellipsoidal domes plus the length L_{IT} defined in **Paragraph 4.3.1**, assuming that the whole intertank section covers the two tanks domes plus some tolerance for systems. Moreover, interstages are jettisoned together with the lower of the two stages up to the exhaust section of the above stage's engine. Finally, the interface between the launch pad and the launcher is assumed to be only with the first stage, for a total length equal to 25% of the engine's length.

The parameters k_1 and k_2, initially taken from reference **[100]**, were further tuned on available mass data for Ariane-5 and VEGA, so that the final values implemented are those reported in **Table 15**. As an example of the bi-dimensional mass surface originating from this WER, **Figure 46** and **Figure 47** show the estimated interstage mass for lower and upper stages as a function of the body maximum diameter and total lateral surface, highlighting larger masses for lower stages due to the higher loads they have to withstand.

Structure	Component	k_1	k_2
Intertank	Lower stages	5.4015	0.5169
Intertank	Upper stages and boosters	3.8664	0.6025
Interstage	Lower stages	7.7165	0.4856
Interstage	Upper stages	5.5234	0.5210
Pad interface	First stage only	25.763	0.5498

Table 15: WER parameters for the assessment of the mass of intertank, interstage and pad interface structures

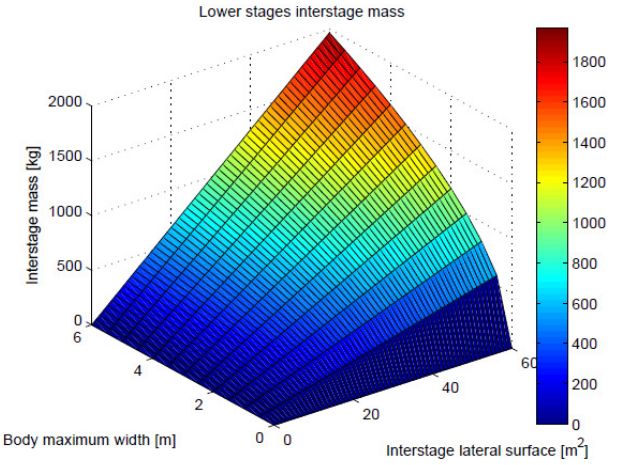


Figure 46: Interstage WER for lower stages with kSM=1, adapted from Ref.6 of [100]

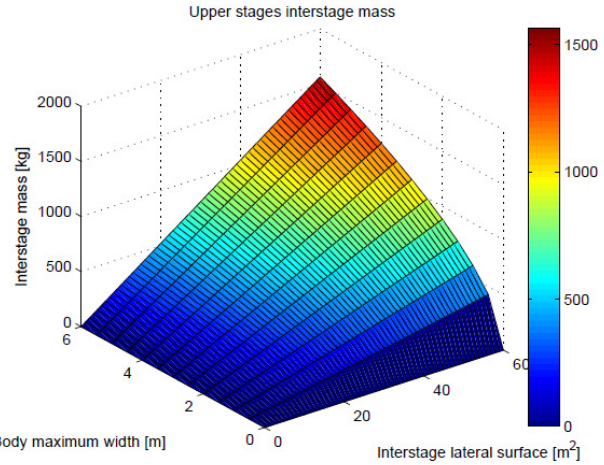


Figure 47: Interstage WER for upper stages with kSM=1, adapted from Ref.6 of [100]

4.5.5. Thrust frame, boosters attachment systems, boosters nose, avionics and EPS

Besides the primary structures described in the previous subparagraphs, few more structural and non structural components need to be considered to determine the total inert mass of stages and boosters, as described here.

Thrust frame:

Several WERs are available from **[100]** allowing to estimate the mass of a thrust frame structure, which is assumed to be necessary for all LP stages and boosters. Figure 48 shows a comparison of these WERs, with the total stage/booster thrust in vacuum as independent variable. The thrust structure of Ariane 5's Vulcain engine is also reported, being very accurately matched by the WER of **Ref. 5**. Although a single data point is hardly sufficient to validate a weight estimation model, no other data were found in public literature. Therefore, the following WER from **Ref. 5** was implemented:

$$M_{TF} = (0.013 \cdot N_{eng}^{0.795} \cdot (224.81 \cdot T_{vac})^{0.579} + 0.01 \cdot N_{eng} \cdot (M_{eng} / 0.45)^{0.717}) \cdot 0.45 \cdot \left(1.5 \cdot SSM \cdot n_{ax,max} \cdot g_0 \right) \cdot k_{SM}$$

where the thrust frame's mass is written as a function of the number, mass and vacuum thrust of engines, as well as of the structural material ($k_{SM}=0.62$ for composite structures from **[100]**), safety margin and maximum axial acceleration.

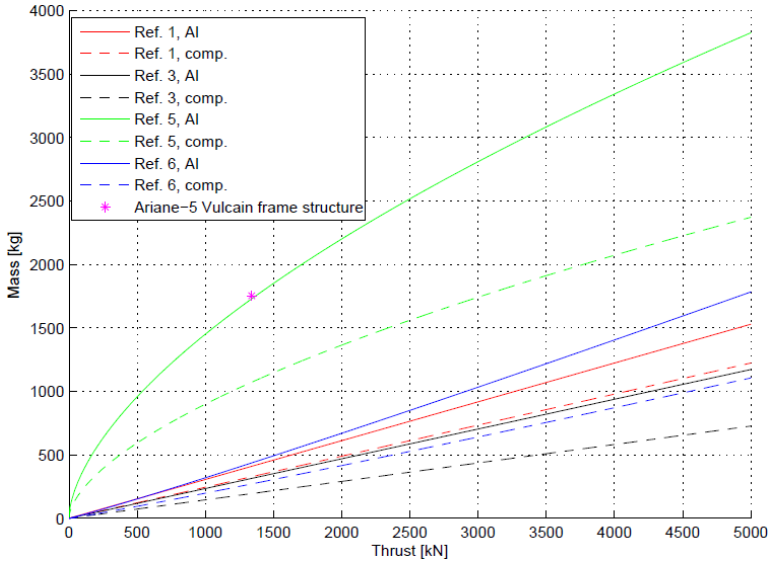


Figure 48: Thrust frame structure WERs from various references of [100], with and without TRF for advanced material (composites based). The two WERs from Ref.5 were selected for implementation in the V1 weights models.

Boosters attachment system:

The boosters attachment system is assumed to be split in two main parts: $M_{BA,booster}$ remaining on the booster after separation and $M_{BA,core}$ remaining instead on the core. These are estimated with a term proportional to the total mass of the booster calibrated on the only available data point (Ariane 5's P-241 attachment, from internal ESA data), as follows:

$$\begin{cases} M_{BA,booster} = 0.00055421 \cdot M_{tot,booster} \\ M_{BA,core} = 0.00145 \cdot M_{tot,booster} \end{cases}$$

Both masses are further split in two parts, one forward and one aft, for CoG computation only (see next subparagraph).

Boosters nose ogive:

The nose ogive of boosters is assumed to be composed of an aerodynamic skirt and a reinforced structure designed to transfer the structural loads to the core. For this reason the WER, taken from **[100]**, is divided in two components which add to each other,: an aerodynamic skirt function of the surface, max dynamic pressure and diameter, and a structural element.

$$\begin{cases} M_{BN,aero} = 4.8825 * S_{nose} \cdot (3.6245e-8 \cdot q_{dyn,max} + 1.7008 + (5.3591e-9 \cdot q_{dyn,max} - 3.252e-3) \cdot D_{BN} \cdot 3.2808) \\ M_{BN,struct} = 0.0111 \cdot k_{SM} \cdot M_{tot,booster} \end{cases}$$

Avionics and EPS:

Avionics and power systems are assumed to be located only within the VEB of the upper stage, and their mass is estimated through two linear WERs taken from **[100]**. Avionics includes communication subsystem, navigation sensors and on-board data handling electronics. Besides, batteries and electrical power conversion and distribution systems constitute the EPS required to run the above systems (in case of electrical TVC, the power required by each unit is included in the TVC mass obtained in the propulsion SSM). Even summed together, these subsystems do not considerably contribute to the total dry mass for ELV, but their influence on the upper stage's mass and therefore on the payload's mass is not negligible.

For avionics, the linear WER from **Ref. 6** of **[100]** assumes as independent variable the total launcher's external surface, considered an index of the dimensions of the vehicle and hence of the requirements in terms of "smart" systems:

$$M_{avionics} = K_{RL} * (246.76 + 1.3183 * S_{tot,launcher}) * (1 - TRF_{avionics})$$

where $TRF_{avionics}=0.75$ is a large reduction factor accounting for the advances in on board electronics since the Shuttle's technology **Ref. 6** is based on, and $K_{RL}=\{0.7;1.0;1.3\}$ for $RL=\{1;2;3\}$ accounts for a different redundancy approach: no redundancies ($RL=1$), critical components only ($RL=2$) or full redundancy ($RL=3$).

For EPS, a linear regression of the estimated avionics mass is instead implemented from **Ref. 1** of **[100]**:

$$M_{eps} = K_{RL} * 0.405 * M_{avionics} * (1 - TRF_{eps})$$

with a much lower technology reduction factor $TRF_{eps}=0.18$.

4.5.6. Centre of Gravity calculation

CoG computation procedure is rather simple, only involving the definition of the weights distribution on the longitudinal axis of the launcher. Coherently with Geometry and Aerodynamics disciplines, all distances are measured from the nose tip

downwards. $X_{CoG,PLF}$, $X_{CoG,PLA}$ and $X_{CoG,j}$ need to be estimated, with j including all stages and boosters, to derive the overall $X_{CoG,\ inert,launch}$, which will be then combined with the propellants CoGs within the Trajectory discipline for verification of the static controllability during atmospheric flight. The following assumptions were taken to define the weights distribution:

- M_{WER} or M_{nozzle} are uniformly distributed on the entire length of the engine for both LP and SP systems.
- M_{TVC} is concentrated at the beginning of the engine's length.
- M_{TF} is concentrated at the engine's upper section.
- $M_{PressSys}$ is concentrated above the oxidizer/fuel tanks, at a distance equal to the radius of the pressurizer tank.
- $M_{igniter}$ is concentrated at the top section of the SP grain.
- M_{unused} is uniformly distributed along the oxidizer and fuel tanks, or along the SP grain.
- M_{PL} is located at a user defined distance from the adapter's section.
- M_{PLA} is concentrated at the lowest section of the PLF.
- $M_{avionics}$ and M_{eps} are concentrated at half the length of the upper stage's VEB.
- The mass of PLF, oxidizer and fuel tanks, SP grain case, interstages and intertanks are assumed to be uniformly distributed on their entire *lateral surface*.
- TPS mass is also considered to be uniformly distributed on the lateral surface of tanks or grain.
- 20% of the boosters attachment mass is assumed located at the coordinate of the core's fore interstage section and the remaining 80% at the coordinate of the core's aft skirt.
- For SP systems, the propellant mass is considered a point mass constantly positioned at 50% of the solid grain length (assuming circumferential burning).
- For LP systems, the fuel and oxidizer masses are considered as two separate point masses whose position is evaluated instant-by-instant in the Trajectory discipline, assuming they are always located at the bottom of the tank.

4.6. Trajectory, guidance and control

The trajectory, or performance, analysis is often the core of any MDA process, with other disciplines being constructed around it. The multi-disciplinary models described in this PhD dissertation are no exception, having been developed starting from a trajectory simulation and optimization model which was already available from a previous MSc research. For this reason, although a wide overview of the trajectory models is given in this section, the reader should refer to **[68]**, **[112]** and **[113]** for more details.

The main tasks of the Trajectory discipline are related to both the simulation of the ascent trajectory and the guidance and control of the launch vehicle throughout such ascent. Navigation is not included in the developed models, since full knowledge of both position and attitude is assumed. The navigation subsystem has in fact a very limited impact on the system level design of the launch vehicle in terms of performance, with only a fraction of the avionics mass described in the previous section being due to position and attitude sensors. The following tasks are therefore defined for the trajectory, guidance and control models:

- Simulation of the ascent trajectory of a launch vehicle from lift-off to burn-out, assessing the final orbital parameters.
- Evaluation of a series of path constraints related to structural loads, heat loads and controllability.
- Optimization of the trajectory to assess the maximum payload which can be achieved while satisfying all final orbit and path constraints. Predefined guidance laws with a small number of optimizable control parameters.

The analysis flow for the Trajectory discipline is very straightforward. In order to compute all forces acting on the launcher at each time instant during the ascent, inputs from all the previous disciplinary analyses are collected before starting the EoMs integration. Specifically, propulsion system performance indexes (I_{sp}^{*}, p_e, A_e), aerodynamic coefficients (C_L,C_D,$C_M(M,\alpha_{tot})$) and pitch, yaw and thrust parameters are necessary, as well as targets and thresholds for path and final constraints. One of the available integrators is then called for the trajectory simulation, requiring computation of the derivatives of the flight-path state variables at each time instant through the repeated evaluation of environmental and force models, guidance laws, and path constraints equations.

The next paragraphs present the main models constituting the Trajectory analysis:

- Dynamic, environmental and force models, including all assumptions, EoMs definition and integration methods.
- Guidance reference profiles and trajectory optimization problem.
- Auxiliary models for non perfect nozzle expansion effects due to external pressure conditions, static controllability verification, structural and heat loads assessment.

4.6.1. Dynamic, environmental and force models

The objective of the trajectory simulation model is to achieve a good accuracy on the final orbital parameters with limited computational times. The following assumptions were introduced to simplify the model as much as possible without sacrificing too much accuracy:

- The launcher is considered a point mass in a rotating reference frame subject to gravitational, aerodynamic and thrust forces. Rotational dynamics is not considered, assuming that the launcher's attitude (yaw, pitch and roll) can be ensured by the attitude control system, which is not modelled. Further simplifications such as bi-dimensional motion or non rotating Earth were however further discarded in light of the large associated errors.
- The gravitational model includes as default only the effects of the main gravitational field harmonic (i.e. spherical planet), but it is given the possibility to increase the accuracy up to the J_4 term, in order to allow more precise computations for celestial bodies with irregular gravity fields (Moon, asteroids).
- Aerodynamic forces are subject to the assumptions described in **Paragraph 4.4.1**: drag and lift are considered functions of Mach and of total AoA, acting on the launcher only until the first jettison event.
- Earth US '76 atmosphere with no wind is assumed by default, with the possibility to switch to exponential (already implemented for Earth and Mars) or user defined atmospheres. Note that wind does not have a considerable effect on the overall trajectory, but may be critical for controllability as well as for structural considerations. For these reason, wind is not modelled for V1 but should be introduced if physics-based structural analysis is performed instead of an historical based weight assessment (see V2 models in **Chapter 6**).
- Propellant burnt on the launch pad before the lift-off is neglected.
- Jettison events are assumed to have constant duration which does not depend on the launcher's design, with a default delay of 5 s (modifiable by the user) between cut-off and ignition of the engines of successive stages.

Within these model assumptions, the EoMs can be solved at each time step with the knowledge of the instantaneous values of mass, thrust, pitch and yaw angles, drag and lift, and gravitational acceleration. Defining conventions and coordinate systems according to the Aerospace Standard ISO 1511 (**[114]**) and the ANSI Recommended Practice (**[115]**), the following complete 3D EoMs for a rigid body's motion with respect to a rotating planet can be written:

$$\begin{cases} \dot{R} = V\sin\gamma \\ \dot{\lambda} = \dfrac{V\cos\gamma\sin\chi}{R\cos\delta} \\ \dot{\delta} = \dfrac{V\cos\gamma\cos\chi}{R} \\ \dot{V} = X + \Omega_E^2 R\cos\delta\left(\cos\delta\sin\gamma - \sin\delta\cos\gamma\cos\chi\right) \\ \dot{\gamma} = -\dfrac{Z}{V} + \dfrac{V\cos\gamma}{R} + 2\Omega_E\cos\delta\sin\chi + \Omega_E^2 R\cos\delta\,\dfrac{\left(\cos\delta\cos\gamma - \sin\delta\sin\gamma\cos\chi\right)}{V} \\ \dot{\chi} = \dfrac{Y}{V\cos\gamma} + \dfrac{V\cos\gamma}{R}\tan\delta\sin\chi + 2\Omega_E\left(\sin\delta - \cos\delta\tan\gamma\cos\chi\right) + \Omega_E^2 R\cos\delta\,\dfrac{\sin\delta\sin\chi}{V\cos\gamma} \end{cases}$$

where the flight-path velocity state variables in **Table 16** are used, Ω_E is the planet's rotational velocity and X, Y, Z are the external forces components along the trajectory axes (i.e. X along velocity, Y starboard in the horizontal plane, Z downward in the vertical plane). Integration of the EoMs is executed starting from the lift-off instant and terminating at the burn-out of the last stage, at the impact on the ground in case the trajectory leads to a crash, or when the required orbital energy is reached. A variable step size 4^{th}-5^{th} order Runge-Kutta-Fehlberg (RKF) method was selected for this purpose, since it ensures the highest computational efficiency among several integrators from ESA's Ordinary Differential Equations (EODE) library. In particular, preliminary testing showed how the RKF45 largely outperforms fixed step size integrators (Adams-Bashforth, RK4 and RK5) in terms of simulation time, and provides slightly better performance even with respect to a higher order RKF78. The same testing phase also allowed identifying a tolerance of 10^{-8} as a good compromise between accuracy and computational effort.

Variable	Definition
$R \in [0;\infty]$	*Radius*: distance from the planet's centre
$\lambda \in [-\pi;\pi]$	*Longitude*: angle between reference meridian (Greenwich) and the meridian of the current position, positive east of the reference meridian.
$\delta \in [-0.5\pi;0.5\pi]$	*Declination*. angle between equatorial plane and current position, positive on the northern hemisphere.
$V \in [0;\infty]$	*Flight-path velocity*: magnitude of the vehicle's velocity relative to an Earth-fixed observer.
$\chi \in [0;2\pi]$	*Flight-path heading*: Angle between North direction and the flight-path velocity component projected onto the horizontal plane, measured clockwise from the North.
$\gamma \in [-0.5\pi;0.5\pi]$	*Flight-path angle*: Angle between flight-path velocity and local horizontal plane, positive for ascent.

Table 16: Definition of flight-path velocity state variables.

4.6.2. Guidance and trajectory optimization problem formulation

With the above assumptions, three continuous control variables are available to steer a 3-DoF trajectory towards orbit: the pitch and yaw attitude angles $\boldsymbol{\theta(t)}$ and $\boldsymbol{\psi(t)}$, defining the thrust direction, as well as the thrust level $\boldsymbol{T(t)}$. Besides, the ignition instant of the different stages and boosters can also be varied to gain payload advantages or satisfy specific constraints. To define the time history of the control variables, standard guidance laws are implemented, which provide reference profiles to be adjusted through the application of a limited number of optimizable parameters. Specifically, a generic ascent trajectory is divided in the following flight phases and guidance profiles for pitch and yaw angles, as summarized in **Table 17**:

1. *Vertical take-off*: up to an altitude given by the user or assumed by default to be equal to $\boldsymbol{H_{clear}=2.0\cdot L_{core}}$.
2. *Pitch-Over (PO)*: constituted of a pitch-down phase (**Figure 49**) of duration $\boldsymbol{t_{PO}}$, with the angle $\boldsymbol{(\gamma-\theta)}$ linearly increasing up to a maximum $\boldsymbol{\Delta\theta_{PO}=max(\gamma-\theta)}$, followed by an exponential decay of $\boldsymbol{(\gamma-\theta)}$ to match gravity turn conditions, which are assumed to be reached after three time constants $\boldsymbol{t_{decay}}$. A constant yaw $\boldsymbol{\psi_{PO}=\psi_{TIL}(t_0)+\Delta\psi_{PO}}$ is assumed through the entire phase, where $\Delta\boldsymbol{\psi_{PO}}$ is an optimizable deviation with respect to the azimuth $\boldsymbol{\psi_{TIL}}$ computed with the target inclination law: $\Psi_{TIL}(t)=sin^{-1}\left(cos\left(i(t)\right)/cos\left(\delta(t)\right)\right)$.
3. *Optimizable Gravity Turn (OGT)* for pitch and *Optimized Target Inclination (OTI)* for yaw: for the whole flight from the end of phase 2 to the ignition of the upper stage, gravity turn pitch conditions are maintained, with a yaw angle constrained to follow the yaw profile defined by the target inclination law. A deviation in both pitch and yaw with respect to these conditions is allowed through a piece-wise linear interpolation of optimizable nodal values $\boldsymbol{\Delta\theta_{jk}}$ and $\boldsymbol{\Delta\psi_{jk}}$, which are referred to the $\boldsymbol{k}$-th interpolation node within the $\boldsymbol{j}$-th stage's flight. Phase 3 is further divided by the atmospheric interface altitude $\boldsymbol{H_{atmo}}$ (or correspondent jettison event) in atmospheric and exoatmospheric sub-phases, for which different boundaries for $\boldsymbol{\Delta\psi}$ and $\boldsymbol{\Delta\theta}$ can be defined. Moreover, Lower Stage Coast (LSC) phases can be activated through the boolean user parameter $\boldsymbol{LSC_j}$, adding a coast period of optimizable duration $\boldsymbol{t_{LSCj}}$ in excess of the default 5 seconds required for the jettison. This may be useful in particular cases, such as for VEGA's Z23/Z9 separation, for which Z9 is ignited after more than one minute, allowing for safer payload fairing separation.
4. *Bilinear Tangent Law* (BTL) for pitch and *OTI* for yaw: for the upper stage's flight, a BTL is assumed for pitch: $tan(\theta)=\dfrac{a^{\xi}\,tan(\theta_{BLT_i})+(tan(\theta_{BLT_f})-a^{\xi}\,tan(\theta_{BLT_i}))\cdot t'}{a^{\xi}+(1-a^{\xi})\cdot t'}$ where $\boldsymbol{t'}$ is the normalized time within the phase, $\boldsymbol{\theta_{BTLi}}$ and $\boldsymbol{\theta_{BTLf}}$ are the pitch angles at the beginning and end of the phase, $\boldsymbol{a}$ is an arbitrary constant and $\boldsymbol{\zeta}$ is a parameter defining the shape of the BTL (**Figure 49**). The BTL was chosen instead of the OGT for the upper stage since it ensures much finer control over the pitch profile with only 3 optimizable parameters: $\boldsymbol{\Delta\theta_{BTLi}=\theta_{BTLi}-\theta(t_{F3})}$, $\boldsymbol{\theta_{BTLf}}$ and $\boldsymbol{\zeta}$.
5. *Upper Stage's Coast (USC)* and *Circularization Burn (CB)*: if the upper stage engine is restartable (***RSC=true***) and the CB option is active (***CB=true***, mandatory for high LEO or MEO/GEO targets), the engine is deactivated when the orbit's apocentre matches the target apocentre. The propellant mass required to circularize the orbit is then computed assuming a 2% gravity loss, and a reference ignition instant is derived. The real ignition instant is then set by an additional optimization variable $\boldsymbol{\Delta t_{CB}}$, corresponding to the difference with respect to the reference ignition instant given in percentage of the CB duration. The final burn is finally simulated assuming tangential thrust.

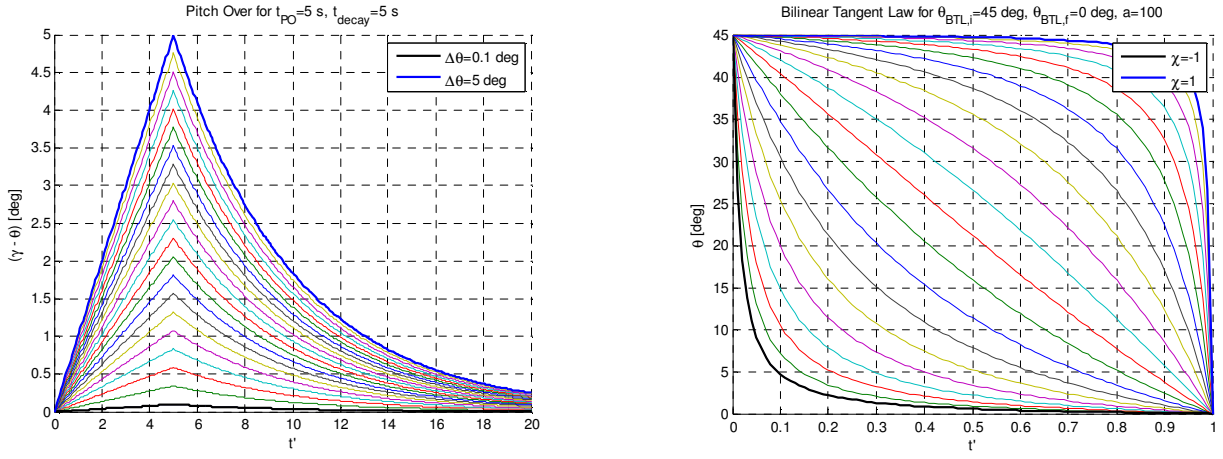


Figure 49: Pitch profiles for the initial pitch-over manoeuvre (left) and the upper stage's bilinear tangent law (right)

Phase	Pitch	Yaw	Fixed parameters	Optimization variables
1	Vertical	None	H_{clear}, Θ=90°	None
2	PO	Constant	None	$\Delta\theta_{PO} \in [1;5]$ deg, $t_{PO} \in [1;10]$ s, $t_{decay} \in [1;5]$ s $\Delta\psi_0 \in [-10;10]$ deg
3	OGT	OTI	H_{atmo}, $\alpha_{tot,max}$, $\Delta\psi_{max,atmo}$, $\Delta\theta_{max,atmo}$, $\Delta\psi_{max,exo}$, $\Delta\theta_{max,exo}$ $N_{nodes,\psi,j}$, $N_{nodes,\theta,j}$, LSC_j	$\Delta\theta_{jk} \in \pm\Delta\theta_{max,atom}$ for $H<H_{atmo}$, $\Delta\theta_{jk} \in \pm\Delta\theta_{max,exo}$ for $H>H_{atmo}$ $\Delta\psi_{jk} \in \pm\Delta\psi_{max,atom}$ for $H<H_{atmo}$, $\Delta\psi_{jk} \in \pm\Delta\psi_{max,exo}$ for $H>H_{atmo}$ $t_{LSCj} \in [0;1]$
4	BTL	OTI	$\Delta\psi_{max,exo}$	$\Delta\theta_{BTLi} \in [-60;60]$ deg, $\theta_{BTLf} \in [-10;90]$ deg, $\xi \in [-1;1]$ $\Delta\psi_{jk} \in \pm\Delta\psi_{max,exo}$
5	Tangential thrust		CB	$\Delta t_{CB} \in [-0.1;0.1]$

Table 17: Pitch/yaw/coast guidance profiles for the different flight phases, including fixed parameters and optimization variables.

In addition to the optimization variables in **Table 17**, the following are related to the j-th stage/booster thrust definition:

- For SP systems, the thrust profile can be tuned with the optimization variables $\boldsymbol{\delta_{SP,j}}$, $\boldsymbol{T_{min,j}}$ and $\boldsymbol{t_{switch,j}}$ according to one of the three thrust profiles defined in **Paragraph 4.2.2** ($\boldsymbol{\delta_{SP,j}=1}$: constant, $\boldsymbol{\delta_{SP,j}=2}$: two-level, $\boldsymbol{\delta_{SP,j}=3}$: linear).
- For LP systems, the thrust level is instead either constant or, for LREs with throttle capability (***ThC=true***), can be adjusted through an optimizable piece-wise linear throttle profile $\boldsymbol{\delta_{th,jk}}$ (with $\boldsymbol{N_{nodes,\delta th,j}}$ interpolation nodes).
- For complex booster configurations, it is also possible to selectively ignite part of the available engines. In particular, Δ_{CI} and Δ_{BI} boolean optimization variables respectively activate in-air ignition of the core and of the second set of boosters, in both cases assuming that the first set is ignited on the ground. The continuous variables $\boldsymbol{\Delta t_{CI}}$ and $\boldsymbol{\Delta t_{BI}}$ are then used to define the ignition delay time, normalized with the first boosters set burn time.

All of the above variables are added to the PLSF, which needs to be varied to maximize the payload performance, to build up the optimization variables vector $\boldsymbol{X_{TDV}}$ of the trajectory optimization problem. The MDO problem defined in **Section 4.1** includes $\boldsymbol{X_{TDV}}$, which can be treated as described in **Paragraph 3.1** either at the same level of the launcher's design by a single optimizer (BBO architecture) or in a nested optimization process (NOL architecture). For better clarity, the trajectory optimization problem is described here as a stand-alone process, mathematically defined as follows:

$$\max_{X_{TDV}} \; PLSF$$

$$with\ X_{TDV} = \{PLSF, \Delta\theta_{PO}, t_{PO}, t_{PO,dec}, \Delta\psi_{PO}, \theta_j, \psi_j, \xi_{BTL}, \Delta\theta_{BTL,i}, \theta_{BTL,f}, \delta_{th,j}, \delta_{SP,j}, T^*_{\min,j}, t_{switch,j}, \Delta_{CI}, \Delta t_{CI}, \Delta_{BI}, \Delta t_{BI}, \Delta t_{LSCj}, \Delta t_{CB}\}$$

$$subject\ to\ \ LB_l \le G_l \le UB_l, \quad l = 1,\ldots,N_{TDC}$$

In case of stand-alone trajectory optimization, for example for the evaluation of the performance of an existing launcher, the PLSF is to be maximized. However, in case of MDO processes for which the payload needs to be fixed (i.e. for which PLSF is not an objective), the scaling factor is frozen to ***1.0*** and the trajectory optimization problem is reduced to the search for a feasible solution. With this respect, the following trajectory constraints are defined for V1, as summarized in **Table 18**:

- Final orbit constraints: the final semiaxis, eccentricity and inclination must match the target orbital parameters within predefined tolerances, which are also used to normalize the values, so that the feasibility region is [-1;1]. Note that although the trajectory models are fully general and any kind of orbit could in theory be reached, RAAN and argument of pericentre targeting as well as orbital plane changes are not addressed.
- Path constraints: structural (n_{ax}, q_{dyn}), thermal (q_{heat}) and controllability (α_{tot}, $M_{TVC\text{-}aero}$) requirements must be satisfied at each time instant during the trajectory simulation.
- Nozzle minimum altitude H_{min} for each stage/booster, resulting in a minimum ignition altitude.
- Geographic heading, for which launch azimuth must be within predefined corridors.

Constraint	Type	Applicability	LB	UB
Target semiaxis	Double	Final conditions	-1	1
Target eccentricity	Double	Final conditions	-1	1
Target inclination	Double	Final conditions	-1	1
Maximum axial acceleration	Double	Each time instant	0	$n_{ax,max}$
Maximum heat flux	Double	Each time instant (atmospheric flight only)	0	$q_{heat,max}$
Maximum dynamic pressure	Double	Each time instant (atmospheric flight only)	0	$q_{dyn,max}$
Maximum angle of attack	Double	Each time instant (atmospheric flight only)	0	$\alpha_{tot,max}$
Static controllability	Double	Each time instant (atmospheric flight only)	$-\infty$	0
Nozzle minimum altitude	Double	Each stage/booster's ignition	0	∞
Geographic heading	Double	Allowed launch site azimuth corridors	$\psi_{min,j}$	$\psi_{max,j}$

Table 18: List of trajectory optimization constraints, including type, applicability and feasibility's lower and upper bounds.

4.6.3. Auxiliary models

4.6.3.1. Altitude effects on thrust and Isp

Since the propulsion system is designed at the nominal nozzle conditions, a corrective term needs to be included in the EoMs to take into account the variation of the thrust (and hence I_{sp}) due to different exhaust and ambient pressures. The standard equation $T(t)=T^*(t)+(p_e-p_a(t))\cdot A_e$ is used for this purpose, where $T^*=\dot{m}\cdot u_e$ corresponds to the nominal thrust value defined as optimization variable and used to scale the engine in the propulsion analysis. The instantaneous propellant consumption is therefore computed as: $\dot{m}=T^*(t)\big/ g_0\cdot I_{sp}^*$ and the resulting specific impulse as: $I_{sp}(h)=I_{sp}^*+\frac{(p_e-p_a)\cdot A_e}{\dot{m}\cdot g_0}$

4.6.3.2. Structural and heat loads

The structural and heat loads defined as path constraints are computed during the trajectory simulation with standard equations. The dynamic pressure is obtained as $\boldsymbol{q_{dyn}=0.5\cdot\rho\cdot V^2}$, the axial acceleration $\boldsymbol{n_{ax}}$ from the time derivative of the velocities, and the convective stagnation point heat flux from Detra-Kamp-Riddel formula (**[116]**), whereas radiative heat flux is neglected due to its negligible influence in rocket launch scenarios.

4.6.3.3. Static controllability

A classical approach to the verification of the capability of the launch vehicle to follow a nominal trajectory is to include the rotational dynamics in a 6-DoF model, implementing a control strategy to define the TVC angles $\boldsymbol{\delta_{TVC\psi}}$ and $\boldsymbol{\delta_{TVC\theta}}$ at each time instant. Further increasing fidelity, multiple bodies can be modelled, for example a movable nozzle and the main launcher, possibly including the flexible dynamics which holds large relevance in case of thin configurations such as VEGA. However, these approaches require both a complex modelling and computational times incompatible with the MDA loop CPU time requirement, and were thus discarded. A static controllability verification was instead implemented, consisting in the comparison of the aerodynamic torque resulting from the instantaneous flight conditions and of the control torque achievable by steering the nozzles at the maximum deflection angle $\boldsymbol{\delta_{TVC}}$.

The CoG/CoP computation and static controllability verification, not requiring large resources, are performed at each integration time step during the atmospheric flight, with the aerodynamic and TVC torque computed as:

$$\begin{cases}\tau_{TVC}=d_C\cdot sin(\delta_{TVC_C})\cdot T_c(t)+d_{B1}\cdot sin(\delta_{TVC_{B1}})\cdot T_{B1}(t)+d_{B2}\cdot sin(\delta_{TVC_{B2}})\cdot T_{B2}(t)\\ \tau_{aero}=N\cdot(X_{CoG}-X_{CoP}), \quad with\ X_{CoP}=-(C_m\cdot L_{ref})/C_N\ and\ C_N=C_L\cdot cos(\alpha_{tot})+C_D\cdot sin(\alpha_{tot})\end{cases}$$

In case the vehicle is designed to be statically stable (which is usually not feasible in multistage launchers), the aerodynamic torque becomes negative and is thus summed to the TVC torque, automatically assuring controllability.

4.7. Cost and reliability assessment

This section aims at providing a brief outlook on the developed cost and reliability analysis models, which have the purpose of estimating the overall CpL and MSP of the launch vehicle, to be used as MDO objectives. A detailed overview of these models, including all Cost Estimation Relationships (CER) and assumed failure rates as well as a thorough verification on existing European and American launchers, can be found in the separate "Cost and Reliability Model" Technical Note D2b (**[A4]**) prepared by MSc student Paolo Martino[36], and is therefore not reported here.

4.7.1. Enhanced TRANSCOST model for Life Cycle Cost determination

The cost model implemented for the estimation of the LCC of a launch vehicle program (and hence the spread CpL) is mostly based on the TRANSCOST model (**[90]**), resulting in the Cost Breakdown Structure of **Figure 50**. TRANSCOST model was selected for several reasons:

- It allows estimation of the total LCC of a launch vehicle program only on the basis of system level information, without requiring specific design data for any subsystem except for propulsion, which is by far the most relevant cost driver. TRANSCOST defines in fact the LCC as sum of the development, production and operation costs, as in Figure 50. Both development and production costs are split, for each stage/booster, in propulsion and system cost. CERs in the generic form $C_j=f(technology_j)\cdot a\cdot M_j^b$ are then defined for each cost element j, originating a cost model which is mostly mass based, though influenced by the technological trade-offs of the discrete optimization variables.
- It is the only publicly available cost model for launch vehicles with complete transparency of all CER data.

[36] The work on cost and reliability was carried out under supervision of the author by a MSc student, Paolo Martino, who prepared the applicable document [A4], including all details on the modelling and verification of cost and reliability estimation disciplines.

- If used coherently with its basic assumptions, it was shown to provide accuracies in the order of 20% on the LCC, which is exceptionally good for a conceptual cost estimate.
- It defines all costs in Man-Hours, hence neglecting any information related to the year or country of development. a table of FY and in country conversion then allows to derive the cost in current M€ for better interpretation.

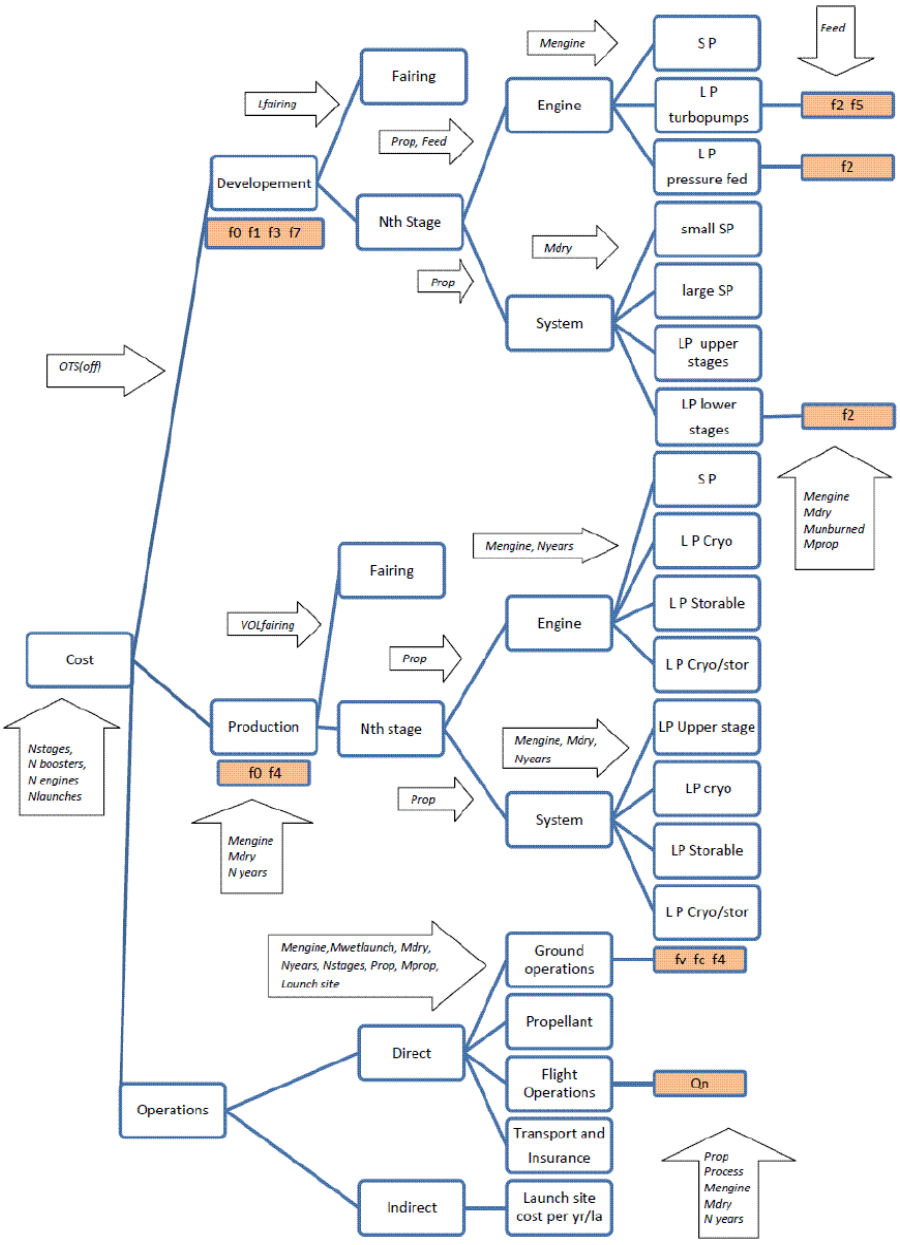


Figure 50: CBS for LCC estimation, including development, production and operations branches. Program level corrective factors fj (defined in 0) are applied where indicated, as well as the design variables influencing each branch (within arrows).

Several modifications to the TRANSCOST models were introduced with technical support by ESA cost engineering section, with the twofold goal of overcoming certain limitations of TRANSCOST, such as the inability to accurately predict the cost of small SP stages/boosters or the lack of PLF/PLA CERs, and of representing the effects of additional technological trade-offs included in the developed multi-disciplinary model. Specifically, the effect on LCC of any design change which results in a performance advantage (or disadvantage) was modelled, attempting to ensure fair multi-objective trade-offs. Note however that, as will be shown in **Chapter 8**, this is a particularly critical point in multi-objective multi-disciplinary, since the optimal solutions returned by MDO largely depend on the tuning of the introduced CER parameters.

4.7.2. Time-based reliability estimation

The purpose of the reliability assessment is to compute a single mission success probability index on the basis of the launcher architecture, technological trade-offs and design parameters. Following a common approach in conceptual and preliminary design, the reliability of a launcher is computed as a function of mission time, so that successive events contribute to increase the failure probability, until the final MSP is obtained as cumulative reliability at the end of flight. This generally results in a lower overall reliability for higher number of stages.

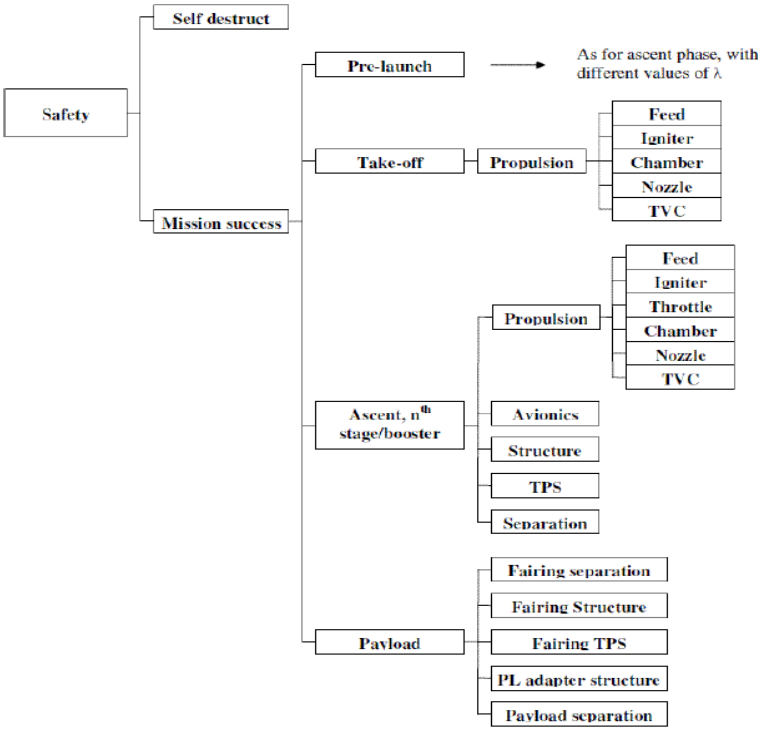


Figure 51: RBS for MSP estimation, including all possible failure chains for both safety (not included in MDO objectives) and mission success

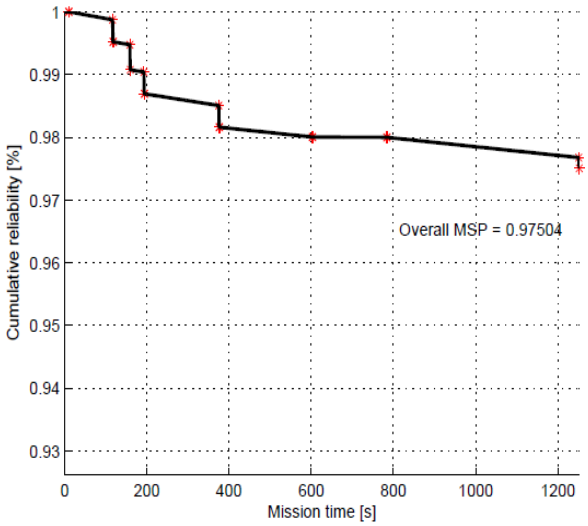


Figure 52: Example of cumulative reliability profile from lift-off to orbit insertion for VEGA's ascent

The cumulative reliability profile over time in each mission phase is computed from the multiplication of several exponential functions for each component contributing to the success of that phase, with the failure rates λ defining their decay rate. A typical reliability curve resulting from this type of analysis is shown in **Figure 52**, for VEGA 4-stage inline launcher. Each of the exponential branches corresponds to a different mission phase, including ground storage, take-off, boosters and stages flights with separation events, PLF jettison and finally PL release. In each phase, different elements contribute to the overall failure rate λ, as summarized in the Risks Breakdown Structure (RBS) shown in **Figure 51**. Note that also a mission safety index is computed on the basis of the mission success and the self-destruct device success probabilities, but does not represent an additional optimization objective since the self-destruct device is simply modelled with a constant failure rate value. Again, see **[A4]** for all details on the reliability assessment process.

It seldom works the way we think it works… in fact, it never does.

CHAPTER 5

CONCEPTUAL MODELS VALIDATION

5.1. Validation philosophy

One of the many challenges presented by MDO is the verification and validation of the developed multi-disciplinary models. Verification requires the execution of all branches of the engineering analysis, first stand-alone and then joined together, to identify bugs and provide evidence that the behaviour of the models corresponds to the one originally intended. Then, optimization needs to be linked and the coupled behaviour also has to be verified for possible errors. Though this is a particularly time consuming process when dealing with a complex MDA such as that described in the previous Chapter, it is not critical. In fact, the real challenge lays in the validation, intended as the quantitative assessment of the suitability of the MDA to represent the design of actual ELVs, and of the capability of the MDO to bring an improvement in the design in terms of the desired objective(s). Moreover in this specific case, the validation procedure also had the additional goals of identifying the most relevant weaknesses of the developed model, in order to improve the analysis cycle in the next step.

In light of this critical role, particular attention was paid to the development of a suitable validation procedure, which was defined in four successive steps (without considering validation of the optimization algorithms, assumed to be reliable):

- *Disciplinary models stand-alone validation* against existing subsystems with known data (e.g. engines, structures, ...), which allows to determine the errors that can be expected on each of the main outputs from each discipline.
- *Global performance Sensitivity Analysis (SA)*, using disciplinary errors information to both identify the most critical disciplines (i.e. causing the largest global error) and to statistically estimate the expectable error. Maximized payload mass is used as performance measure, with existing European launchers (Ariane-5 ECA and VEGA) as test cases.
- *MDA processes* for Ariane 5 ECA and VEGA, aimed at verifying the capability of the multi-disciplinary models to correctly assess performance, cost and reliability on real-world scenarios.
- *Simple MDO processes* for Ariane 5 ECA and VEGA, targeted at understanding the optimization capabilities on a simplified (i.e. kept small) MDO problem.

The key points of these four steps are shown in the next sections, and a full report of all validation results can be found in the Verification and Validation Document (VVD), **[A9]**. Only the last step involves the full MDO, but trajectory optimization is used in sensitivity analyses and MDA with the purpose of assessing the payload performance of a given launcher. In this case, the NOL architecture described in **Section 3.1** is employed without any SLO algorithm (no MDO).

5.2. Stand-alone disciplines validation

To assess the accuracy of the engineering models, each disciplinary analysis was executed stand-alone on a set of existing components (e.g. engines for propulsion, whole launcher for aerodynamics, ...) whose input/output data are available in a collected database, in order to determine the mean, maximum and standard deviation of the error on each relevant output parameter. During this phase, several tuning parameters specific to each disciplinary model were also calibrated, as already hinted in the modelling chapter. For example, parameters as the specific impulse loss coefficients due to specific physical effects were varied with respect to the original values taken from literature, in order to better match the available experimental data points.

The complete validation procedure and results for each discipline cannot be shown here. However, the calibration of the parameters and the assessment of the accuracy for the specific impulse in vacuum are reported to exemplify the general methodology. For the validation of Propulsion disciplinary analysis, the database of OTS LREs described in **Section 4.2** was reused, with the addition of few other LREs of older origin and of several SRMs, for a total of 38 LP and 9 SP systems. This is the largest available experimental base which can be collected from free sources, whereas more limited information

exists on other disciplines. Specifically, only Ariane-5's and VEGA's approximate C_L and C_D databases were used for the validation of Aerodynamics discipline, whereas the total inert masses of 30 PLFs, 7 SP boosters, 11 SP stages, 14 LP lower stages and 9 LP upper stages were collected for the Weights, with very limited information on specific components. Moreover, all these experimental data are subject to the possibility of error, hence caution was used in the statistical analyses, for example by eliminating few clearly inconsistent data points.

In addition to the vacuum specific impulse, the nozzle's exhaust area A_e was considered as the second propulsive performance parameter, in order to evaluate the errors on the specific impulse in non-vacuum conditions. However, errors on A_e have an impact on global performance which is one order of magnitude lower than $I_{sp,vac}$ for first stages and boosters and becomes negligible in thinner atmospheres. Hence, results for A_e as well as for the other propulsion output parameters (minimum nozzle's operational altitude, engine's inert mass and dimensions) are not presented here. **Figure 53** shows the actual vs. computed values of vacuum specific impulse, including both theoretical (CEA) and corrected performance. Two sets of engines are represented:

- *Calibration set*: 16 LP engines, two per each of the 9 modelled architectures and 5 SP engines, used to calibrate the model tuning parameters, especially (but not only) those related to the specific impulse losses evaluation.
- *Validation set*: 22 LP and 4 SP engines, used to further validate the results by evaluating the output parameters for a set of engines the models have not been calibrated for. This form of cross-validation increases the confidence on the models accuracy's assessment and supports their extendibility to other and new liquid and solid rocket engines.

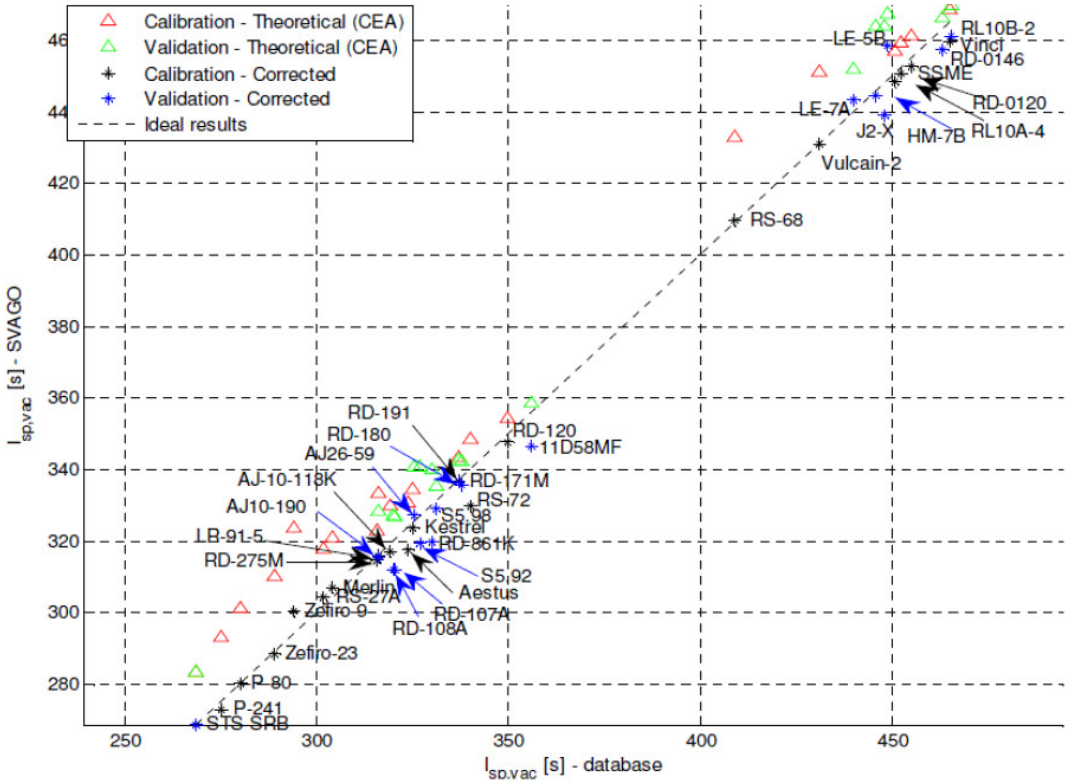


Figure 53: Propulsion performance model results: actual and computed $I_{sp,vac}$ for both calibration and validation sets of engines.

Table 19 reports the same results in tabular form, showing errors lower than 1% for most calibration set's engines and a small degradation of the accuracy from the calibration to the validation set. Results obtained with the commercial software REDTOP ([**117**]), developed by Spaceworks Engineering[37] and whose academic version employs engineering methods of comparable fidelity, are also shown in the table. A bias toward overestimating $I_{sp,vac}$ with REDTOP can be highlighted, especially for cryogenic engines. After the calibration procedure, the values of vacuum specific impulse obtained with the developed models are instead generally adequate for both conceptual and early preliminary level design, with a maximum error $M=3.1\%$, a mean error of $\mu=-0.6\%$ (slightly conservative calibration) and standard deviation of $\sigma=1.3\%$ over the entire database of LP and SP engines.

These statistical figures are summarized for the most relevant outputs of each discipline in **Table 20**. Several important design variables are not included (e.g. propellant mass, nominal thrust, ...) because they are treated as input parameters given by the user or by the optimization algorithm, and are therefore not obtained as outputs of the disciplinary models. It is

[37] http://www.sei.aero

clear that the vacuum specific impulse is the only parameter which can be accurately assessed with the developed analyses, while masses and dimensions show maximum errors up to ~40% and aerodynamic coefficients up to ~100% (only for some specific flight conditions).

Engine	$I_{sp,vac}$ [s]						
	Database	*SVAGO theoretical*	*Error % ($svago_{theo}$)*	*SVAGO corrected*	*Error % ($svago_{corr}$)*	*Redtop theoretical*	*Redtop corrected*
SSME	452.3	459.1	1.50	450.5	-0.39	475.0	461.9
RD-0120	455.0	461.3	1.39	452.7	-0.50	477.0	464.6
RL10A-4	451.0	458.6	1.68	448.4	-0.57	477.6	459.6
Vinci	465.0	468.3	0.71	459.5	-1.19	490.7	471.1
Vulcain-2	431.0	451.0	4.64	431.1	0.01	469.4	443.1
RS-68	409.0	433.1	5.89	409.6	0.13	447.7	418.6
RD-120	350.0	354.4	1.26	347.8	-0.64	374.3	351.2
RD-191	337.0	343.4	1.90	337.0	0.01	375.1	351.6
Merlin-1C	304.0	321.0	5.60	307.3	1.07	330.1	304.2
RS-27A	302.0	318.0	5.31	304.4	0.79	328.4	301.6
Kestrel	325.0	334.7	2.98	323.7	-0.39	358.2	326.1
RD-275M	315.8	322.8	2.22	314.7	-0.34	329.9	319.3
RS-72	340.0	348.7	2.56	329.9	**-2.98**	362.2	338.3
LR-91-5	316.0	333.6	5.56	316.1	0.03	341.0	318.5
AJ10-118K	319.2	330.0	3.39	317.0	-0.69	350.6	326.1
Aestus	324.0	330.6	2.04	317.5	**-1.99**	343.7	319.7
P-241	275.3	293.1	6.47	272.9	-0.86	NA	NA
P-80	280.0	301.5	7.67	280.5	0.17	NA	NA
Zefiro-23	289.0	310.2	7.34	288.4	-0.22	NA	NA
Zefiro-9	294.0	323.7	10.1	300.6	**2.24**	NA	NA
STS SRB	268.6	283.3	5.49	268.7	0.04	NA	NA

Engine	$I_{sp,vac}$ [s]				
	Database	*SVAGO theoretical*	*Error % ($svago_{theo}$)*	*SVAGO corrected*	*Error % ($svago_{corr}$)*
J2-X	448.0	469.6	4.81	444.4	-0.80
RL10B-2	465.5	464.5	-0.22	455.7	**-2.10**
LE-7A	440.0	451.8	2.68	443.4	0.78
LE-5B	449.0	467.4	4.10	458.6	**2.14**
HM-7B	446.0	464.0	4.04	444.6	-0.33
RD-0146	463.2	466.1	0.62	457.3	-1.27
AJ26-59	331.3	335.5	1.27	329.3	-0.60
RD-107A	320.2	327.1	2.16	312.2	**-2.51**
RD-108A	320.6	326.7	1.89	312.1	**-2.65**
RD-180	337.8	342.3	1.32	335.9	-0.56
RD-171M	337.0	342.8	1.73	336.5	-0.16
11D58MF	356.0	358.7	0.75	346.8	**-2.59**
AJ10-190	316.0	328.4	3.92	315.4	-0.18
S5.92	327.0	340.6	4.17	319.6	**-2.26**
S5.98	325.5	340.6	4.64	327.1	0.48
RD-861K	330.0	340.0	3.04	319.8	**-3.09**

Table 19: Propulsion performance models results: actual and computed $I_{sp,vac}$ for the calibration (left) and validation (right) sets of engines. For the calibration set, a comparison with the commercial tool of similar fidelity REDTOP is also reported.

Discipline	Parameter	Description	E [%]	M [%]	μ [%]	σ [%]
Propulsion	$I_{sp,vac}$ [s]	Vacuum specific impulse	1.02	3.09	-0.59	1.27
Propulsion	A_e [m^2]	Nozzle exhaust area (for I_{sp}(h))	14.03	31.19	-0.85	15.03
Aerodynamics	C_D	Drag coefficient	9.35	81.80	4.28	9.27
Aerodynamics	C_L	Lift coefficient	10.40	98.47	9.10	14.27
Weights	$M_{inert,SP,BS}$ [kg]	SP boosters total inert mass	10.64	21.09	-0.04	13.50
Weights	$M_{inert,SP,LS}$ [kg]	SP lower stages total inert mass	22.46	36.06	8.31	16.07
Weights	$M_{inert,LP,LS}$ [kg]	LP lower stages total inert mass	10.06	37.60	5.63	13.47
Weights	$M_{inert,LP,US}$ [kg]	LP upper stages total inert mass	10.02	21.24	-3.30	14.18
Weights	M_{PLF} [kg]	Payload fairing mass	15.04	33.62	-8.68	16.40

Table 20: Summary of disciplinary level validation procedure: statistical figures of the errors in the estimation of the most relevant output parameters for each discipline: $E = mean(|e|)$, $M = max(|e|)$, $\mu = mean(e)$, $\sigma = stdev(e)$

5.3. Sensitivity analyses

In order to evaluate the suitability of the engineering models for the conceptual design of ELV, the discipline-level accuracy results shown in the previous section are not sufficient. The different errors combine in fact in the MDA process to determine the overall error on the global performance of the vehicle. For this reason, a detailed analysis of the sensitivity of the global performance to the disciplinary errors is the next logical validation step. Only global performance, which can be measured in terms of payload performance for a given design, is considered in the sensitivity analyses, since no realistic terms of comparison for cost and reliability of launchers exist. Local trajectory optimizations with WORHP were therefore executed for Ariane 5 ECA and VEGA[38], perturbing the main disciplinary output parameters with respect to the actual launcher design by percentages reflecting the disciplinary errors in **Table 20**. Note that, in order for all sensitivity analyses to be fair, the robustness of the payload performance assessment must be proven. This was the subject of a large investigation, aimed at improving efficiency and robustness of local trajectory optimization with WORHP, which is described in a separate document (**[A11]**). Through enhancements in the model's smoothness and constraints handling, very

[38] All data from this point are referred to the reference European missions from Kourou: a standard GTO (a=24383.6 km, e=0.7292, i=7 deg) for Ariane 5 ECA and a polar LEO at 700 km for VEGA (a=7078 km, e=0, i=90 deg).

good robustness properties were obtained (see **Annex 1** of **[A11]**). To further ensure fairness to the sensitivity analyses presented here, each trajectory optimization was composed of three WORHP processes, each starting from a different initial guess computed with a short (100 function evaluations) PSO-1D run. Two types of SA were then considered:

- *One-variable-at-a-time* analysis: only one parameter is perturbed, while keeping all others equal to the actual design values, by a percentage equal to plus and minus the average of the absolute errors $\pm E$ (worst-case approach). This is repeated for each parameter, assessing the payload performance. The goal of this analysis is to identify the critical discipline(s) which is (are) most likely to determine the largest errors in global performance, on the basis of both the parameters' relevance and modelling accuracy.
- *Monte-Carlo* analysis: all parameters are randomly varyied at the same time according to a Gaussian distribution, obtained with the mean and standard deviation of the errors μ and σ. Since the distribution of perturbations is based on realistic error probabilities, such as for $I_{sp,vac}$ in **Figure 54**, repeated Montecarlo runs should allow determining realistic global performance distributions. From these, the possible bias toward higher or lower payloads can be determined (μ_{PL}). And, even more important in a MDO context, $1\sigma_{PL}$ or $3\sigma_{PL}$ payload ranges can be obtained, representing the deviations in performance which can be expected to be due to modelling errors rather than to actual design changes.

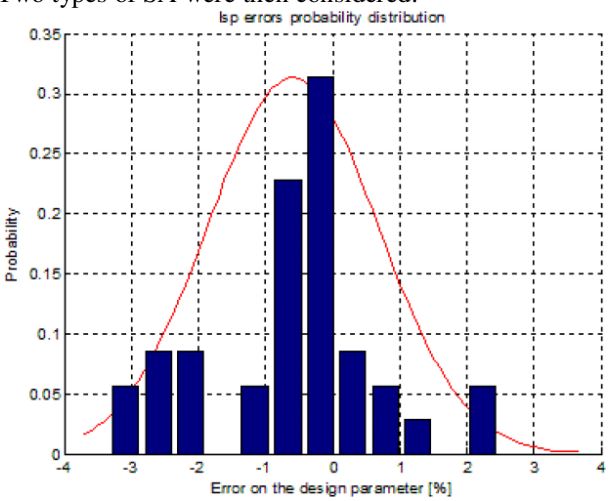


Figure 54: Example of input parameter distribution ($I_{sp,vac}$ in this case) for Montecarlo sensitivity analyses.

Component	Parameter	ΔParameter		ΔM_{PL}		$\partial M_{PL} / \partial Param$
Ariane 5 ECA:						
ESC-A	$M_{inert,LP,US}$	+652 kg	+10.0%	-649 kg	**-6.4 %**	-0.995 kg/kg
Ariane PLF	$M_{inert,PLF}$	+402 kg	+15.0%	-29 kg	-0.3 %	-0.071 kg/kg
EPC	$M_{inert,LP,LS}$	+1740 kg	+10.1%	-679 kg	**-6.7 %**	-0.390 kg/kg
EAP	$M_{inert,SP,BS}$	+3415 kg	+10.6%	-409 kg	-4.0 %	-0.120 kg/kg
ESC-A	$I_{sp,vac}$	+4.4 s	+1.0%	+130 kg	+1.3 %	+29.5 kg/s
EPC	$I_{sp,vac}$	+4.3 s	+1.0%	+283 kg	+2.8 %	+66.5 kg/s
EAP	$I_{sp,vac}$	+2.6 s	+1.0%	+93 kg	+0.9 %	+35.8 kg/s
ESC-A	A_e	+0.11 m^2	+14.0%	+51 kg	+0.5 %	+470.4 kg/m^2
EPC	A_e	+0.49 m^2	+14.0%	+124 kg	+1.2 %	+255.5 kg/m^2
EAP	A_e	+0.98 m^2	+14.0%	+291 kg	+2.9 %	+296.4 kg/m^2
Ariane	C_D	+0.09	+9.4%	-70 kg	**-0.7 %**	-746.5 kg
Ariane	C_L	+0.10	+10.6%	+5 kg	**+0.0 %**	+46.0 kg
VEGA:						
AVUM	$M_{inert,LP,US}$	+72 kg	+10.0%	-73 kg	**-4.6 %**	-1.016 kg/kg
VEGA PLF	$M_{inert,PLF}$	+80 kg	+15.0%	-11 kg	-0.7 %	-0.143 kg/kg
P80	$M_{inert,SP,LS}$	+1908 kg	+22.5%	-77 kg	**-4.8 %**	-0.040 kg/kg
Z23	$M_{inert,SP,LS}$	+612 kg	+22.5%	-87 kg	**-5.4 %**	-0.142 kg/kg
Z9	$M_{inert,SP,LS}$	+318 kg	+22.5%	-92 kg	**-6.1 %**	-0.289 kg/kg
AVUM	$I_{sp,vac}$	+3.1 s	+1.0%	+6 kg	+0.4 %	+1.9 kg/s
P80	$I_{sp,vac}$	+2.6 s	+1.0%	+24 kg	+1.5 %	+9.0 kg/s
Z23	$I_{sp,vac}$	+2.7 s	+1.0%	+32 kg	+2.0 %	+11.7 kg/s
Z9	$I_{sp,vac}$	+2.8 s	+1.0%	+42 kg	+2.6 %	+14.7 kg/s
AVUM	A_e	+0.01 m^2	+14.0%	+2 kg	+0.1 %	+183.3 kg/m^2
P80	A_e	+0.43 m^2	+14.0%	+41 kg	+2.6 %	+94.1 kg/m^2
Z23	A_e	+0.24 m^2	+14.0%	+42 kg	+2.6 %	+174.5 kg/m^2
Z9	A_e	+0.17 m^2	+14.0%	+42 kg	+2.6 %	+250.4 kg/m^2
Ariane	C_D	+0.09	+9.4%	-15 kg	**-0.9 %**	-157.2 kg
Ariane	C_L	+0.10	+10.6%	+1 kg	**+0.0 %**	+11.4 kg

Table 21: Results of one-variable-at-a-time sensitivity analysis for +*E* parameter perturbation.

From the results of the *one-variable-at-a-time* analyses, shown in **Table 21** for a variation of the parameters equal to ***+E***, significant considerations can be drawn. First, the most critical discipline is by far the Weights analysis, resulting in errors up to ***±10%*** on the payload due to varying the mass of the upper stage of an equal ***±E***. The large payload sensitivity to the upper stage's mass, in fact equal to ***-1***, suggests an increase in the modelling effort for upper stage weights, which are also the most difficult to evaluate with statistical historic-based methods since very different structural ratios can be found in existing vehicles[39]. Generic improvements in structural and non structural masses estimation of all launcher components are anyway necessary for the second modelling step, even though sensitivity to other stages/boosters is more limited. For instance, Ariane 5's boosters and core were found to have sensitivities of approximately ***8.3 to 1*** and ***2.6 to 1***, that is 830 kg saved on each booster or 260 kg saved on the core correspond to a ~100 kg payload upgrade. Similar figures were obtained also for VEGA SP stages, with respectively ***25.0 to 1***, ***7.0 to 1*** and ***3.5 to 1*** for P80, Z23 and Z9.

On the contrary, Propulsion discipline seems accurate enough for the purpose of conceptual and possibly early preliminary design, at least as regards to the pure performance. The vacuum specific impulse of stages and boosters, although having a large sensitivity on the PL performance (in particular Ariane's EPC and VEGA's Z23 and Z9), is modelled with so much higher accuracy with respect to the weights that the resulting payload error is not critical (***<3%***). Moreover, errors in the order of ***15%*** on the exhaust area seem to have a similar percentage impact on the PL mass (***<3%***) with respect to errors in the order of ***1%*** on the specific impulse. The inaccurate evaluation of the propulsion inert masses instead contributes to the large global error resulting from the stages/boosters weights, and improvements should therefore be taken under consideration. Finally, a possibly surprising result is that Aerodynamics, even though very badly estimated by DATCOM, does not sensibly affect the design solution's quality, with a ***~1%*** influence limited to C_D and basically no influence of C_L.

Back to Montecarlo analyses, results are visually presented in **Figure 55**, which shows the payload mass distribution for Ariane and VEGA after 100 Montecarlo runs. The numerical values μ_{PL} and σ_{PL} are compared in **Table 22** with the reference payload mass from manual and with the estimated performance with all launcher data (i.e. propulsion performances, inert masses, aerodynamics database, which would normally be computed within the MDA) fixed to the actual design. These show a sensible bias of the developed trajectory models towards overestimating the performance of ELVs, with ***+9%*** for Ariane and ***+14%*** for VEGA which were confirmed by further testing on an alternative configuration (Russian launcher DNEPR). Although the stand-alone disciplinary validation proved the trajectory integration models to be correct (negligible errors were highlighted in a comparison with the commercial tool ASTOS), the coupling of guidance and optimization algorithms results in optimistic estimates of the payload. Several reasons can be identified for this behaviour, mainly ascribable to lacking modelling features (realistic SP thrust profiles and related p_{cc} and I_{sp} variations, wind and manoeuvres steering losses, specific impulse degradation over time due to throat erosion) and to possible inaccuracies in the launchers' design data.

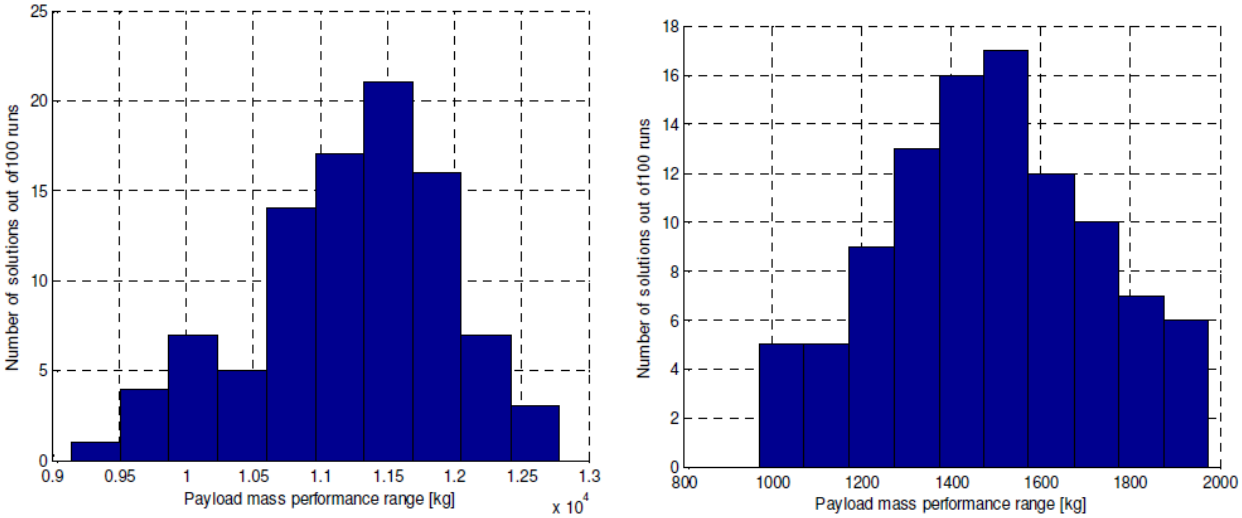


Figure 55: Montecarlo analysis results: Ariane 5 ECA's (left) and VEGA's (right) payload performance distributions.

Taking into account the performance assessment errors, Montecarlo results provide important clues on the engineering models' representation of existing launchers and on the expectable payload errors. Specifically, a tendency towards overestimating/underestimating the performances of Ariane 5 ECA/VEGA is evident, with ***11217.0 kg*** and ***1487.6 kg*** of mean payload over 100 Montecarlo runs. This different behaviour can be again traced back mainly to the weight models, and in particular to the different mean errors for the dry masses of stages and boosters: Ariane 5's liquid upper stage shows

[39] e.g. $SR=M_{inert}/M_{tot}$ ranges from 0.085 of Falcon 1's upper stage to 0.457 of VEGA's AVUM, even though most figures are in [0.1-0.2].

in fact a negative mean error, therefore resulting in higher mean payload mass, whereas all VEGA solid stages masses have a positive mean error, that more than offsets the lower mass of the small storable upper stage. As regards to the standard deviations, σ_{PL}=8% for Ariane 5 and σ_{PL}=16% for VEGA are reasonable figures for expected 1σ launcher performance error in a conceptual-level design environment employing simplified engineering models. This is an extremely important parameter in a MDO context, since it defines the confidence level that a given optimized design solution is actually better with respect to the discarded options. Finally, μ_{PL} and σ_{PL} combine in the payload distribution ranges for Ariane 5 and VEGA, with a maximum expectable (1σ) error of **+19%** in the case of Ariane 5.

	Ariane 5 ECA		VEGA	
Payload mass from manual	10050 kg	-	1500 kg	-
Payload mass for fixed actual design	10944 kg	+8.9%	1715 kg	+14.3%
Payload mass distribution mean value μ_{PL}	11217 kg	+11.6%	1488 kg	-0.2%
Payload mass distribution stdev σ_{PL}	762 kg	7.6%	239 kg	+15.9%
Payload mass expectable ($1\sigma_{PL}$) range	[10455; 11979] kg	[+4.0; +19.2]%	[1249; 1727] kg	[-15.7; +16.1]%

Table 22: Ariane 5 ECA's and VEGA's payload performances from manual, fixed design and Montecarlo distributions.

5.4. Multi-Disciplinary Analysis for existing launchers

This section presents results for the stand-alone MDA of European launchers, for which the payload performance was maximized through nested trajectory optimization for the reference mission and for which all discrete and continuous design variables were frozen to the actual design. Note the difference with respect to the fixed design approach of the previous section: here, the *input* variables of the MDA are frozen (e.g. propellant masses, chamber pressures, ...), whereas all *outputs* required for the trajectory optimization are computed regularly within the technical analyses (Propulsion, Geometry, Aerodynamics, Weights). By employing 3 optimization runs started with PSO-1D and continued with WORHP, payload performance is then estimated with good confidence, with the aim of understanding the errors committed with respect to that and other global performance metrics such as the total wet mass. Finally, execution of the non-technical disciplines closes the MDA loop, providing cost per launch and mission success probability to be additionally compared with real world figures.

5.4.1. Ariane 5 ECA

Ariane-5 ECA's reference mission is to put a maximum of 10050 kg of payload to a standard, 7 deg, GTO from Kourou, assuming the following path constraints: n_{ax} = 4.55 g; q_{dyn} = 57000 Pa; $q_{heat,max}$ = 40 MW/m^2; $q_{heat,max,PL}$ = 1035 W/m^2.

The overall results of the MDA are presented in several tables and figures: **Table 23** summarizes all numerical outputs of the technical disciplines, including mass properties, geometric parameters and propulsion performances, and also reports an overview of the global ELV's design figures, namely the GTOW, CpL, MSP and the optimized payload mass. All data are compared with the actual values, whose reference source is mentioned for fairness. Then, **Figure 56** and **Figure 57** show the drag coefficient, CoG and CoP profiles, which are also used for the Trajectory discipline, whose most relevant outputs are presented in **Figure 58** to **Figure 63**. Finally, **Table 24** and **Figure 64** deal with the non technical disciplines, laying out the CBS and reliability over time.

Parameter	MDA	Actual	Actual data source
Mass properties			
Payload fairing mass [kg]	2718	2675	**M: Ariane 5 User's Manual, [118]**
Payload adapter mass [kg]	542	400	**M**: Cone 3936: 200 kg + Payload Adapter System 1666S: 200 kg
Upper stage inert mass [kg]	3261	6504	ESC-A + VEB: 6000 kg (**M**) + M_{unused}=504 kg (assumed 3.5%)
Lower stage inert mass [kg]	17413	17300	EPC + JAVE: 14700 kg (**M**) + M_{unused}=2600 kg (assumed 1.5%)
Boosters inert mass [kg] (each)	31446	32098	From sum of available internal component data and M_{unused}
Geometry			
Payload fairing ogive length [m]	6.25	5.55	**M**
Payload fairing total length [m]	17.00	17.00	**M**
VEB length [m]	1.55	1.56	**Geometry Inspection** (through graph digitizer software[40]) – **GI**
Upper stage length [m]	5.22	4.71	**M**
Lower stage length [m]	25.42	30.52	Lower stage plus JAVE (**M**) + protruding nozzle from **GI**
Boosters length [m]	32.76	33.59[41]	Top to lower skirt: 31.6 m (**M**) + protruding nozzle from **GI**

[40] www.getdata-graph-digitizer.com
[41] Including nozze extending from the case (31.61 m case only)

Total launcher length [m]	**49.19**	**53.79**	
Reference aerodynamic area [m^2]	**37.51**	**37.51**	
Propulsion performance			
Upper stage nominal thrust [kN]	62.9	-	Actual data **Not Available** (**NA**)
Upper stage exhaust pressure [Pa]	2358.9	-	**NA**
Upper stage nozzle exit diameter [m]	0.77	0.99	Snecma web[42]
Upper stage nominal I_{sp} [s]	433.5	-	**NA**
Upper stage vacuum I_{sp} [s]	**444.7**	**446.0**	Snecma web
Upper stage minimum altitude [km]	9.8	-	**NA**
Lower stage nominal thrust [kN]	1291.9	-	**NA**
Lower stage exhaust pressure [Pa]	12692.5	-	**NA**
Lower stage nozzle exit diameter [m]	2.19	2.10	Snecma web
Lower stage nominal I_{sp} [s]	416.8	-	**NA**
Lower stage vacuum I_{sp} [s]	**430.2**	**431**	Snecma web
Lower stage minimum altitude [km]	0	0	-
Boosters nominal thrust (each) [kN]	5796.0	-	**NA**
Boosters exhaust pressure [Pa]	85088.3	-	**NA**
Boosters nozzle exit diameter [m]	3.05	3.06	Fiat-Avio web[43]
Boosters nominal I_{sp} [s]	248.3	-	**NA**
Boosters minimum altitude [km]	0	-	**NA**
Boosters vacuum I_{sp} [s]	**275.0**	**275.3**	Fiat-Avio web
Global design objectives			
Gross Take-Off Weight [kg]	**764835**	**766383**	Sum of all masses, including **10050 kg** nominal payload
Payload mass to reference GTO [kg]	**12472**	**10050**	**M**
Total Cost per Launch [M€]	**171.6**	**173.0**	FAA-AST's Semi-Annual Launch Report (2nd 2011): 220 M$
Mission Success Probability	**0.927**	**0.970**	32 successes in 33 ECA launches, updated to end 2011

Table 23: Ariane-5 ECA long fairing *output* design parameters, results from the MDA compared with the actual values. All *input* design variables (e.g. propellant mass, ...) and user parameters (e.g. fairing diameter) are not reported here.

Several considerations can be drawn from the outcome of Ariane 5 ECA's MDA, in particular:

- The external geometry calculations are rather approximate: the diameters coincide with the actual values (***D_{core}=5.40 m***, ***B_{bset1}=3.05 m***), but the lengths are sensibly underestimated. Although geometric parameters do not directly affect the vehicle's EoMs (except for ***A_{ref}***), lengths errors are propagated through the inert masses calculation.
- Although the total wet mass at launch matches very well the real value, the WBS shows significant inaccuracies, confirming the impression from the sensitivity analyses that weights models should be improved. In particular, the upper stage's inert mass is underestimated by almost ***50%***, which is indeed critical considering the high sensitivity of the global performance to this value. The main causes are the underestimation of the stage's length and the lack in the modelling of aft and forward skirts, which in case of ESC-A play a prominent role in the total structural mass.
- As already mentioned, propulsion performance indexes are very close to reality. Moreover, exploiting the OTS engines functionality, values reported in the MDA column (for LP only) are substituted at the end of the Propulsion analysis with the actual figures, so that the engines performance can be considered "exact", within data inaccuracies.
- The payload performance of Ariane 5 ECA assessed with the full MDA with nested trajectory optimization is of **12472 kg**, a sensible ***24.1%*** overestimation with respect to the user manual's value. This represents a further ***1.5 tons*** payload mass increase with respect to the ***10944 kg*** obtained with frozen launcher *output* parameters, as reported in the sensitivity analysis section. The main cause is surely the ***3.2 tons*** error in the upper stage's inert mass, which is only partially offset by the higher mass of the core and lower performance of the boosters.
- Actual CBS and reliability data are not available, hence an exhaustive validation of the non-performance disciplines is impossible. However, the total cost per launch of ***~172 FY2009 M€*** is comparable to the reported 220 M$ of current Ariane 5 ECA's launch price (FAA AST), and the CBS in **Table 24** appears reasonable, with approximately 3/5 of the LCC allocated to production, as typical in ELVs, and the remaining equally split in development and operations. The vehicle's assessed reliability instead does not do justice to the impressive launch record of Ariane 5 ECA. The final MSP is driven by the cryogenic propulsion's high failure rate, which comes from historical evaluations, determining a low ***92.7%*** probability of mission success. This is definitely pessimistic if compared to the proven 32 successes out of 33 ECA version launches, especially considering that the failure occurred on the maiden

[42] http://www.snecma.com/-space-engines-.html
[43] http://www.aviogroup.com/en/catalog/space/propulsione_spaziale/products

flight which is by definition very risk. Although a better match can easily be obtained by tuning the failure rates, this is in not in line with the goal of ensuring fair trade-offs of different architectures on the basis of the theoretical reliability of the design, which should indeed be lower with respect to Ariane 5's actual mission success record.

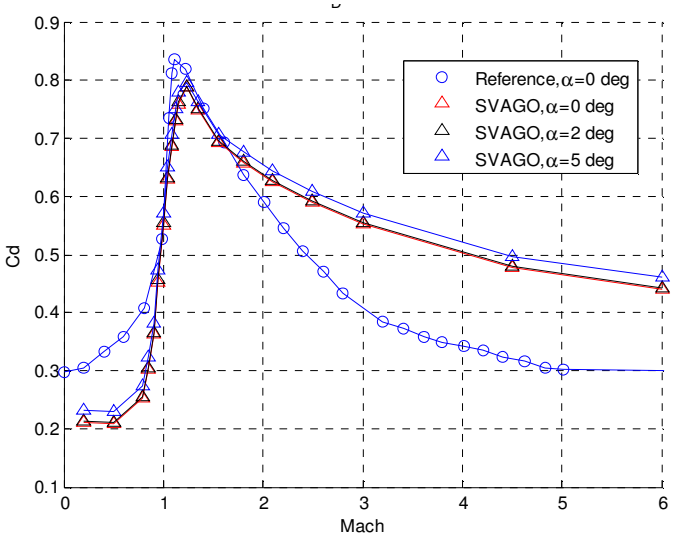


Figure 56: Ariane 5 ECA, drag coefficient.

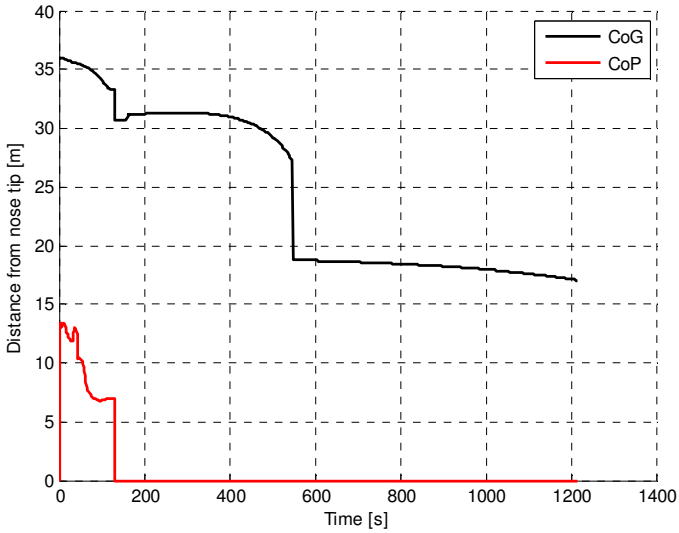


Figure 57: Ariane 5 ECA, CoG and CoP profiles over time.

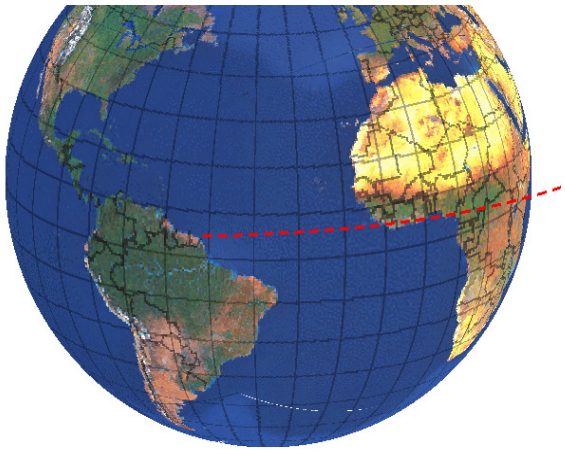


Figure 58: Ariane 5 ECA to GTO, 3D ECI trajectory.

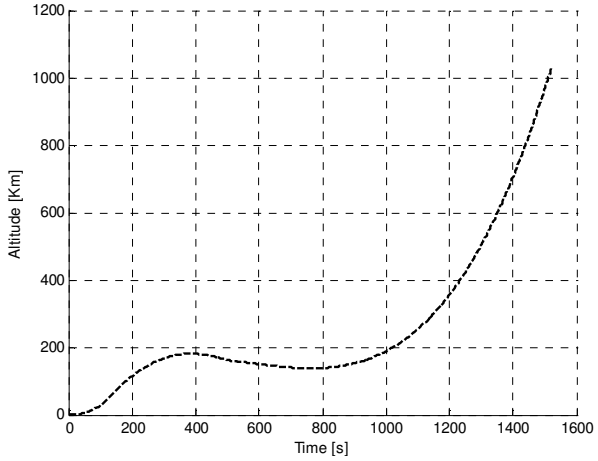


Figure 59: Ariane 5 ECA to GTO, altitude profile.

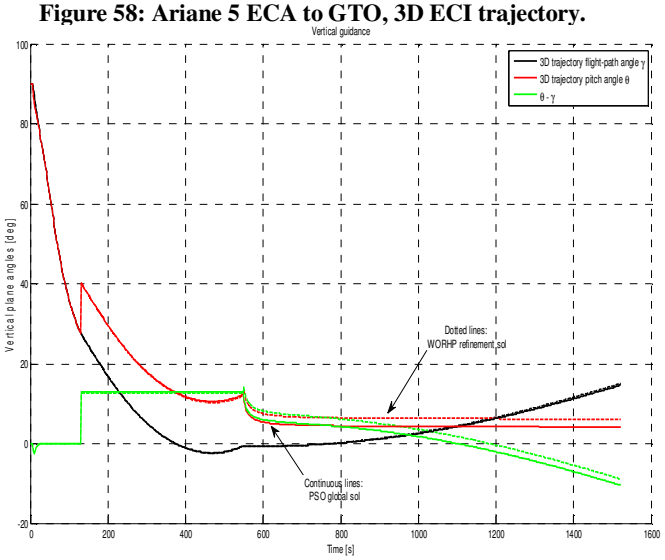


Figure 60: Ariane 5 ECA to GTO, vertical guidance profiles.

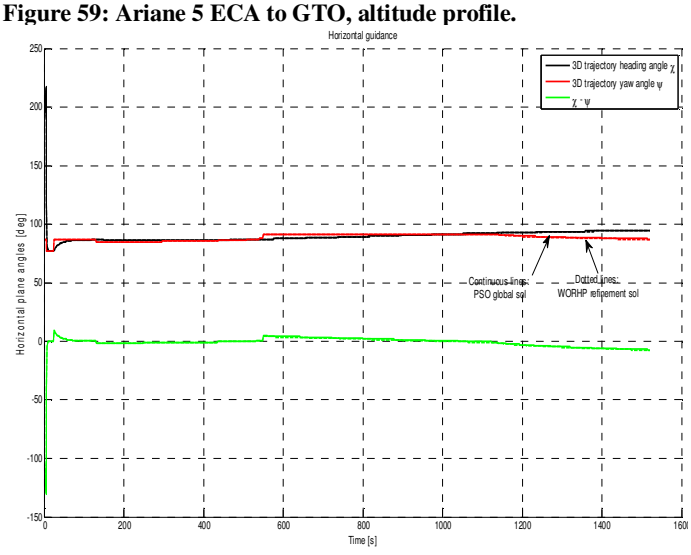


Figure 61: Ariane 5 ECA to GTO, horizontal guidance profiles.

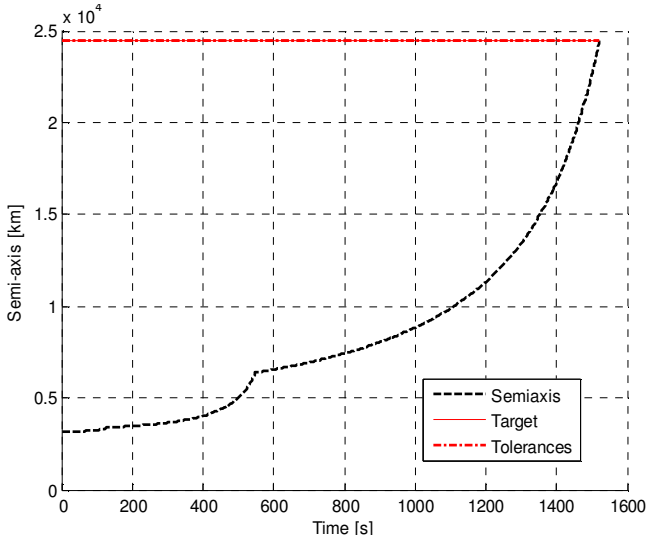


Figure 62: Ariane 5 ECA to GTO, semiaxis profile.

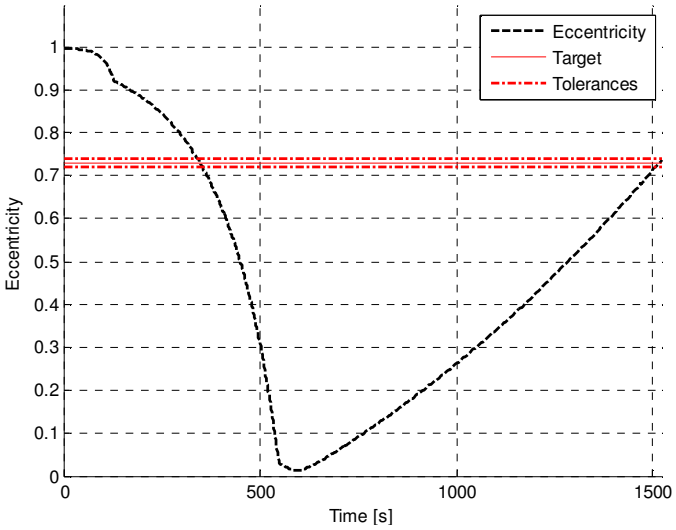


Figure 63: Ariane 5 ECA to GTO, eccentricity profile.

	Dev.Sys.	Dev.Eng.	Prod.Sys.[44]	Prod.Eng.
PLF	33.5	0	6.0	0
EPC	267.5	456.1	23.3	13
ESC-A	154.5	1522.7	38.1	8.3
EAP (each)	506.1	0	15.6	0
Operations	**Ground**	**Propellant**	**Flight**	**Other[45]**
(per flight)	30.5	0.2	1.9	3.7
LCC	**Develop.**	**Product.**	**Operations**	**Total**
total	**4428.3**	**11831.6**	**4331.8**	**20592.1**
per flight	**36.9**	**98.6**	**36.1**	**171.6**
k€/kg	**3.7**	**9.8**	**3.6**	**17.1**

Table 24: CBS from Ariane-5 ECA MDA (all costs in FY2009 M€ if not otherwise stated), assuming 120 launches in 20 years.

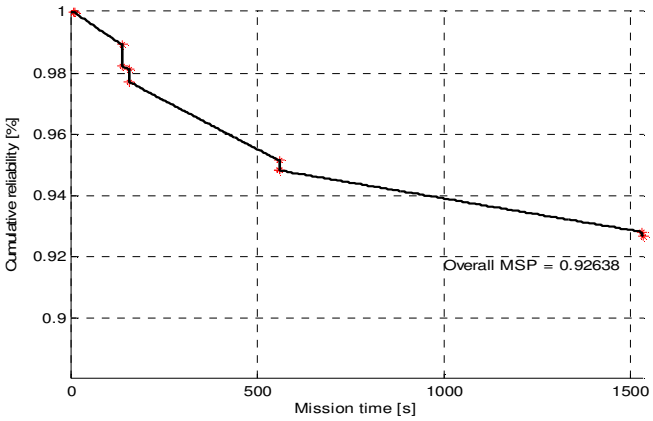


Figure 64: Ariane 5 ECA to GTO, cumulative reliability over elapsed mission time, originating the final MSP.

5.4.2. VEGA

VEGA's MDA results presentation retraces that of Ariane-5 ECA's, for a reference mission consisting in 1500 kg of payload to a 700 km polar LEO from Kourou, assuming similar path constraints: n_{ax} = 5.5 g; q_{dyn} = 57 kPa; $q_{heat,max}$ = 40 MW/m²; $q_{heat,max,PL}$ = 1035 W/m². **Table 25** summarizes mass, geometry, propulsion and global design figures, **Figure 65** and **Figure 66** show C_D, CoG and CoP, **Figure 67** to **Figure 72** describe the optimal trajectory, and finally **Table 26** and **Figure 73** outline CBS and reliability over time.

Parameter	MDA	Actual	Actual data source
Mass properties			
Payload fairing mass [kg]	606	529	**M: VEGA User's Manual, [119]**
Payload adapter mass [kg]	79	60	**M**
1st stage inert mass [kg]	10131	8497	**I: Internal ESA data**
2nd stage inert mass [kg]	3425	2725	**I**
3rd stage inert mass [kg]	1340	1416	**I**
Upper stage inert mass [kg]	846	715	**I**
Geometry			
Payload fairing total length [m]	7.88	7.88	**M**

[44] Production costs are intended as averaged over the total number of items produced for 120 launches (total cost is obtained through the application of learning curves to the Theoretical First Unit (TFU) cost, which is directly derived from the production CERs).
[45] Other operations costs include transport, launch site management, insurance and fees.

1st stage length [m]	12.32	10.80	**Geometry Inspection** (through graph digitizer software) – **GI**
2nd stage length [m]	8.74	7.55	**GI**
3rd stage length [m]	5.21	3.63	**GI**
Upper stage length [m], outside PLF	-	1.21	**GI**
Upper stage length [m], total	2.28	2.04	**GI**
Total launcher length [m]	**36.63**	**31.07**	
Reference aerodynamic area [m^2]	**7.21**	**7.21**	
Propulsion performance			
1st stage nominal thrust [kN]	2060.6	-	Actual data **Not Available** (**NA**)
1st stage exhaust pressure [Pa]	78961	-	**NA**
1st stage nozzle exit diameter [m]	1.95	1.98	
1st stage nominal I_{sp} [s]	255.9	-	**NA**
1st stage vacuum I_{sp} [s]	**280.1**	**280.1**	
1st stage minimum altitude [km]	0	0	
2nd stage nominal thrust [kN]	803.1	-	**NA**
2nd stage exhaust pressure [Pa]	50565	-	**NA**
2nd stage nozzle exit diameter [m]	1.47	1.47	
2nd stage nominal I_{sp} [s]	266.4	-	**NA**
2nd stage vacuum I_{sp} [s]	**288.1**	**287.5**	
2nd stage minimum altitude [km]	0	-	**NA**
3rd stage nominal thrust [kN]	279.8	-	**NA**
3rd stage exhaust pressure [Pa]	13288	-	**NA**
3rd stage nozzle exit diameter [m]	1.24	1.23	
3rd stage nominal I_{sp} [s]	282.1	-	**NA**
3rd stage vacuum I_{sp} [s]	**300.3**	**294.4**	
3rd stage minimum altitude [km]	5.3	-	**NA**
Upper stage nominal thrust [kN]	2.4	-	**NA**
Upper stage exhaust pressure [Pa]	1190	-	**NA**
Upper stage nozzle exit diameter [m]	0.27	0.30	
Upper stage nominal I_{sp} [s]	307.0	-	**NA**
Upper stage vacuum I_{sp} [s]	**314.7**	**318.4**	
Upper stage minimum altitude [km]	13.8	-	**NA**
Global design objectives			
Gross Take-Off Weight [kg]	**139402**	**138089**	Sum of all masses, including **1500 kg** nominal payload
Payload mass to reference GTO [kg]	**1403**	**1500**	**M**
Total Cost per Launch [M€]	**34.8**	**32**	Advertised VEGA launch price[46]
Mission Success Probability	**0.975**	**0.980**	Projected ESA MSP

Table 25: VEGA *output* design parameters, results from the MDA compared with the actual values. All *input* design variables (e.g. propellant mass, ...) and user parameters (e.g. fairing diameter) are not reported here.

From the analysis of the results of VEGA's MDA, the following considerations can be added:

- The error in lengths estimation is more relevant for VEGA, since all assessments contribute to an overall overestimation of ***5.6 m*** of the total launcher's length. This is mainly due to the computed lengths of the SP nozzles (no nozzle submersion in the grain is considered) and of the upper stage (lack of "under fairing" configuration modelling). The length excess results in a ***10-20%*** increase in drag coefficient over the flight envelop, which however should not cause large payload deviations.
- As for Ariane, the GTOW is close to the real value, but with sensible errors on the single elements. Specifically, inert mass evaluation for P80 and Z23 SRMs is largely pessimistic, in a minor part due to the cases and more relevantly due to the nozzles. In fact, VEGA SRMs' nozzles employ new materials, allowing for very low mass, which is not captured by the models. AVUM's mass is also overestimated by more than ***130 kg***, translating to ***~10%*** PL error.
- For VEGA, the full MDA with nested trajectory optimization sets the payload mass to ***1403 kg***, ***6.5%*** less than the reference by manual. This is a drastic reduction (***315 kg***) with respect to the fixed design payload, which again can be explained with the errors in the inert masses, partially offset by a sensible overestimation of Z9's I_{sp}.

[46] http://spacenews.com/launch/012312-vega-expected-price-competitive-with-russian-rockets.html

- The total cost per launch of ***~35 FY2009 M€*** is reasonably close to the assessed ***32 M€***, with operations assuming a major role in light of the lower development and production costs of SP systems. The MSP is also very well represented, with a ***97.5%*** launch success probability which matches ESA's 0.98 reliability design target and is almost entirely due to the ignition and jettison sequence.

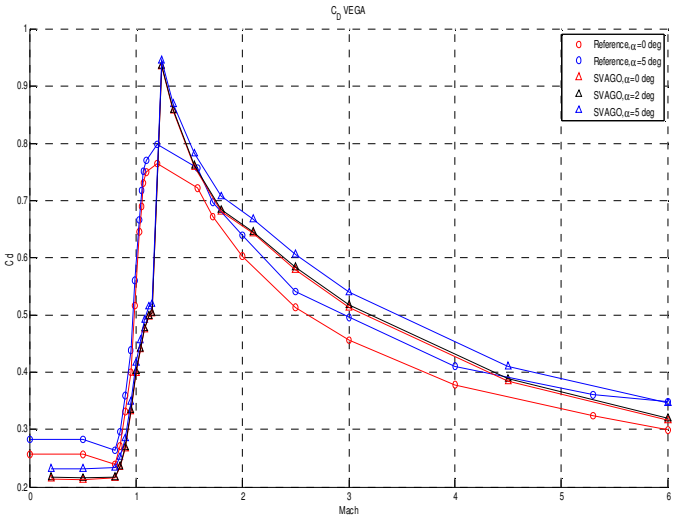


Figure 65: VEGA, drag coefficient.

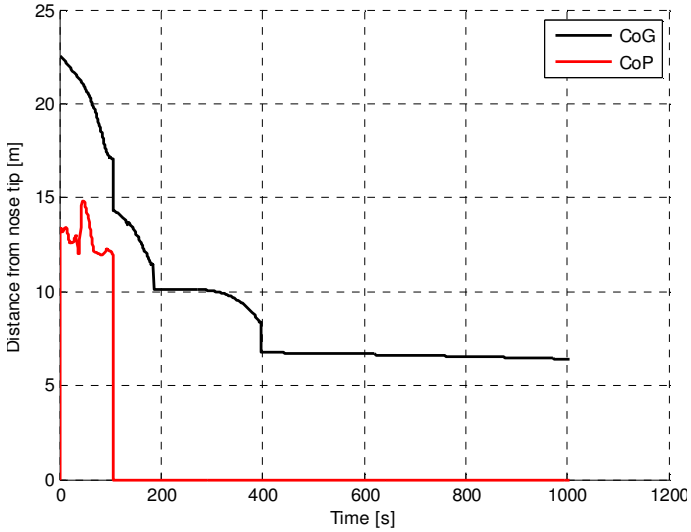


Figure 66: VEGA, CoG and CoP profiles over time.

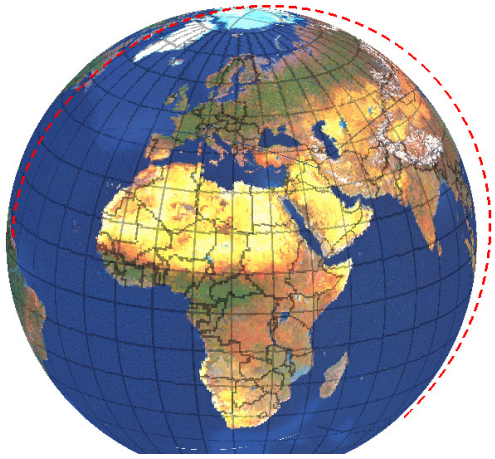


Figure 67: VEGA to polar LEO, 3D ECI trajectory.

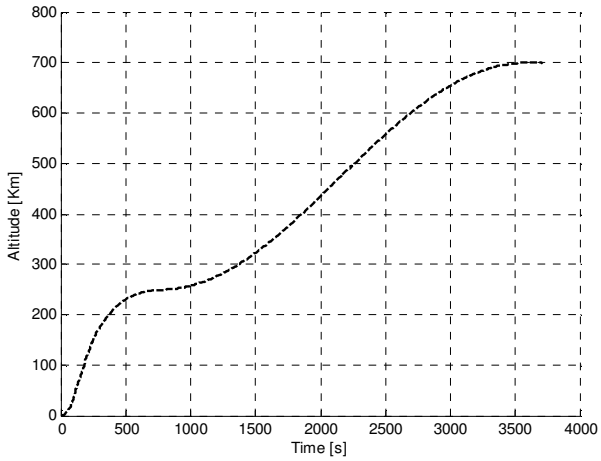


Figure 68: VEGA to polar LEO, altitude profile.

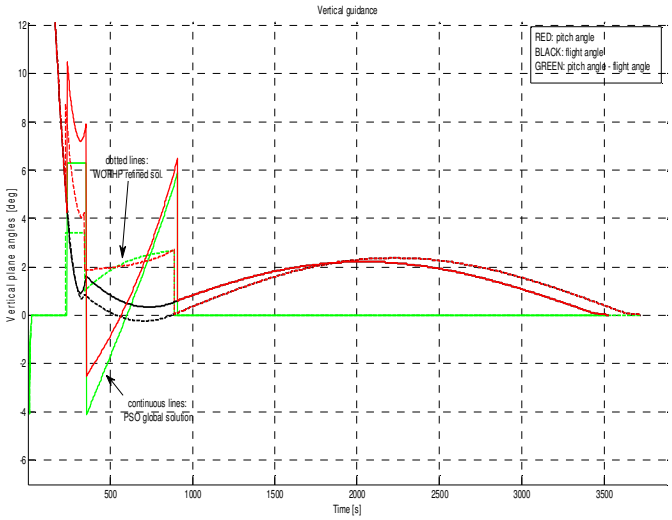


Figure 69: VEGA to polar LEO, vertical guidance profiles.

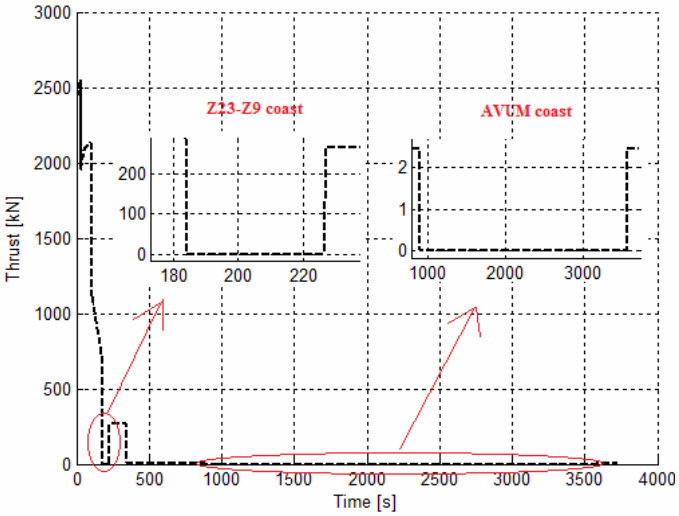


Figure 70: VEGA to polar LEO, thrust profile and coast phases.

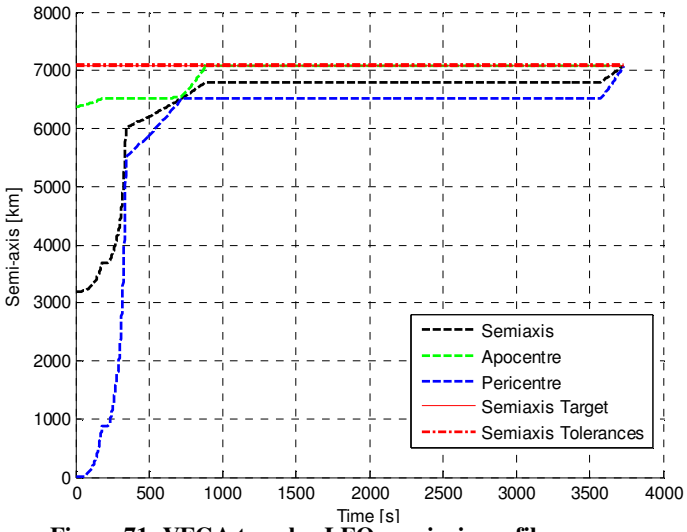


Figure 71: VEGA to polar LEO, semiaxis profile.

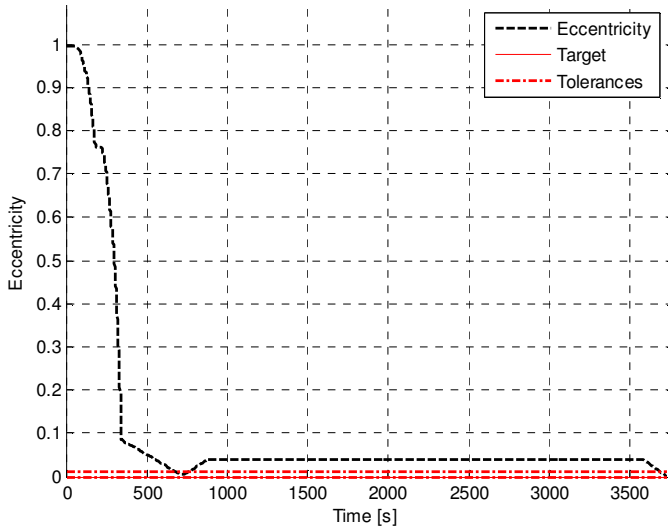


Figure 72: VEGA to polar LEO, eccentricity profile.

	Dev.Sys.	Dev.Eng.	Prod.Sys.	Prod.Eng.
PLF	25.8	0		0.7
P80	177.1	0		8.1
Z23	34.8	0		3.1
Z9	26.9	0		2.2
AVUM	113.9	114.8		3.6
Operations	**Ground**	**Propellant**	**Flight**	**Other**
per flight	8.1	0.1	1.6	1.1
LCC	**Develop.**	**Product.**	**Operations**	**Total**
total	740.0	2132.6	1300.9	4173.5
per flight	**6.2**	**17.7**	**10.8**	**34.8**
k€/kg	**4.1**	**11.8**	**7.2**	**23.2**

Table 26: CBS from VEGA MDA (all costs in FY2009 M€ if not otherwise stated), assuming 120 launches in 20 years.

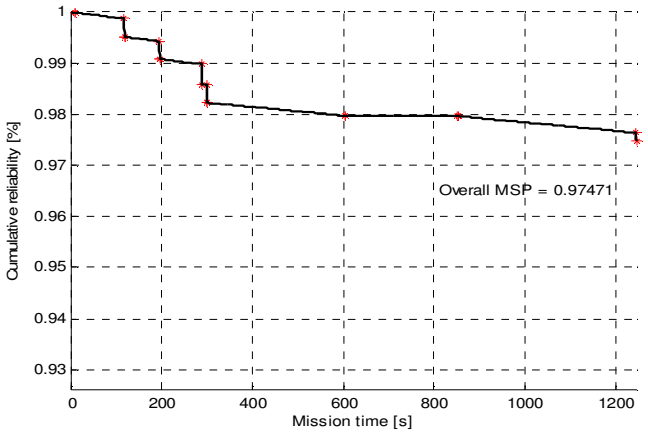


Figure 73: VEGA to polar LEO, cumulative reliability over elapsed mission time, originating the final MSP.

5.5. Single-Objective MDO for existing launchers

As last validation step, the full MDO process was tested, in order to verify the capability of the coupled optimization algorithms and engineering models to produce an improvement in the design. To both simplify the task of the optimizer and allow a more straightforward interpretation of the achieved design solutions, the investigated problems were kept small and the minimization of the GTOW was considered as single objective for PSO-1D. Complete MDO runs were executed for Ariane 5 ECA and VEGA, by freezing all technological and architectural trade-off variables as well as the propulsion systems design, and allowing a ±30% range for stages/boosters' propellant masses and length over diameters and for the trajectory load parameters. This results in the following MDO problems:

- Minimize the GTOW of an Ariane 5 ECA-like launcher for 10050 kg of fixed performance to 7 deg GTO from Kourou, varying $X=\{M_{prop,EPC}, M_{prop,ESC-A}, M_{prop,EAP}, (L/D)_{EAP}, q_{dyn,max}, q_{heat,max}, n_{ax,max}\}$.
- Minimize the GTOW of a VEGA-like launcher for 1500 kg of fixed performance to 700 km polar LEO from Kourou, varying $X=\{ M_{prop,P80}, M_{prop,Z23}, M_{prop,Z9}, M_{prop,AVUM}, (L/D)_{P80}, (L/D)_{Z9}, q_{dyn,max}, q_{heat,max}, n_{ax,max}\}$.

Respectively ***10*** and ***14*** trajectory optimization variables complement the ***7*** and ***9*** design optimization variables for Ariane and VEGA. With these settings, the final optimal solutions should prove the capability of the MDO to improve in terms of launch mass the real world designs, which have not been optimized for sheer performance. This represents a qualitative validation, with the objective of achieving comparable but improved solutions with respect to the actual launchers and of correctly reproducing physics-based trade-offs.

A preliminary step for the MDO validation was the definition of a standard setting for all PSO parameters. Several attempts were executed with various sets of parameters (w, c_1, c_2, χ, V_{max}, see **Section 3.2**) which had shown good behaviour in previous works both by the author (**[68]**) and other literature sources tackling the problem of PSO set-up (see

for example **[78]**, **[120]**, **[121]**, **[122]**). The chosen configuration, which was afterwards used consistently throughout all optimization runs of both the single-objective PSO-1D and the multi-objective DG-MOPSO, is the following:

w=1.2 → 0.4 (linearly decreasing from the first to the last iteration), c_1***=1.0***, c_2***=2.0***, χ***=1.0***, V_{max}***=1.0***.

Swarm size is also critical for the *exploration vs. exploitation* properties of the algorithm, but was selected on a case-by-case basis depending on the size of the problem being solved. For the single-objective MDO described in this section, 100 particles were found to be a good compromise (faster convergence was highlighted with respect to 50 and 200 particles). Results are presented in the next paragraphs, showing the minimum mass solutions obtained out of 3 PSO-1D runs of 1000 iterations each, and a subsequent WORHP local refinement process. Multiple PSO runs account for stochastic effects, with convergence histories for Ariane and VEGA shown in **Figure 74** and **Figure 75**. Full convergence is not reached in 1000 iterations, even though a maximum difference among the three runs of less than 3% is achieved for both MDO problems.

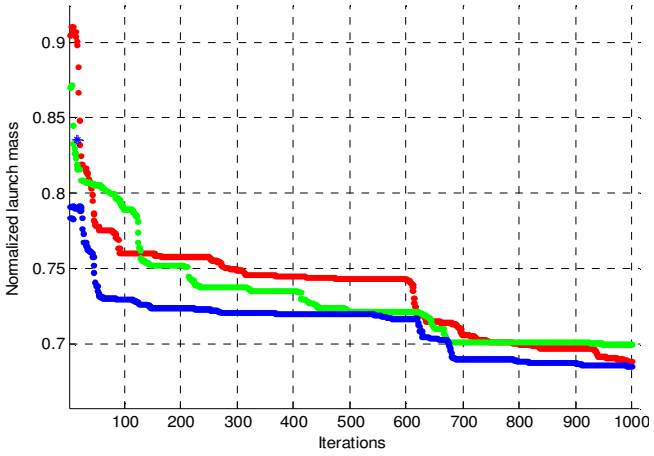


Figure 74: PSO-1D convergence history for Ariane 5 ECA

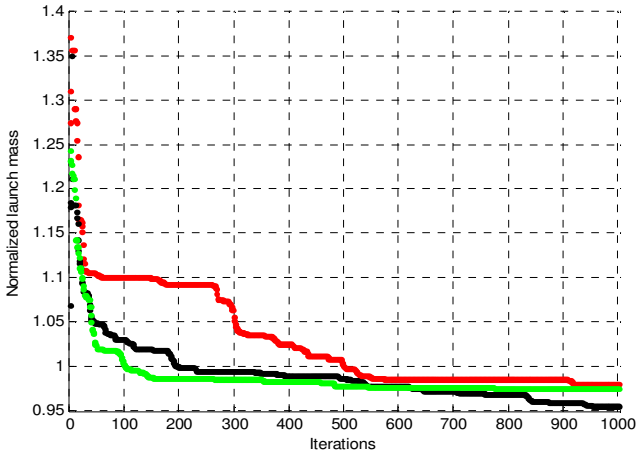


Figure 75: PSO-1D convergence history for VEGA

As shown above, MDO is capable of sensibly reducing the GTOW of existing launchers with respect to the real-world figures. Nonetheless, it is important to point out how this reduction is only partially corresponding to a real design improvement, due to the error in performance assessment highlighted by the MDAs, described in the previous section. In order to assess the real design improvement brought by MDO, the derivative $\partial PL/\partial GTOW$ needs to be evaluated. In fact, the 764.8 tons and 139.4 tons GTOWs obtained from the MDAs of Ariane and VEGA correspond to an estimated PL mass of 12472 kg and 1403 kg. In order to be compared with the GTOWs from the MDO processes, for which the payload is fixed to the real performances of 10050 kg and 1500 kg, the GTOWs from the MDAs need to be scaled to the approximate values that the launchers would have – according to the implemented design models –for these real performances. To derive the influence of PL on GTOW, multi-objective MDO for min GTOW and max PL can be exploited, with the purpose of obtaining Pareto fronts from which a linear regression can be considered an estimate of $\partial PL/\partial GTOW$.

Although more examples of multi-objective optimization will be given in **Chapter 8**, **Figure 76** reports the min GTOW vs. max PL Pareto fronts for Ariane 5 ECA and VEGA (*"small"* MDO problem), with the purpose of highlighting the effect on the total launch mass of varying payload requirements. The Pareto solutions show an almost linear correlation, with $\partial PL/\partial GTOW = \begin{cases} 17.0 \ \ kg/ton & for \ \ Ariane \\ 18.8 \ \ kg/ton & for \ \ VEGA \end{cases}$ obtained from linear regression, justifying the use of the derivative as a constant to assess the design improvements brought by MDO.

5.5.1. Ariane 5 ECA

MDO results are presented in **Table 27**, where the values of the optimization variables and objectives obtained with PSO and WORHP are compared to the actual design and MDA results. Moreover, **Figure 77** gives a visual impression of the optimal solutions achieved, since the external geometry is meaningful of the overall dimensions and hence staging characteristics. As for the MDAs, considerations evinced from the critical analysis of the results are schematically outlined:

- A minimum launch mass of ***522 tons*** is achieved with PSO-1D which is further lowered to ***506 tons*** by WORHP, a sensible ***2.1%*** reduction contributed by the local refinement. Note however that three different design solutions are obtained starting WORHP from the three global solutions, highlighting how the MDO model is very sensitive to the first guess, likely showing many local minima.
- The 506 tons design solution represents a ***34.0%*** reduction in GTOW with respect to the actual Ariane 5 ECA. Assuming the partial derivative $\partial PL/\partial GTOW = 17.0 \ kg/ton$ as derived above, this corresponds to an improvement in

GTOW of ***20.5%*** brought by the MDO process. This very relevant design enhancement is mainly obtained through the re-allocation of the propellant masses and the reduction of the trajectory loads, which affect the inert masses.

- From a mass-optimal perspective, the MDO shows how a drastic reduction in the size of the SP boosters allows improving the design, due to the higher specific impulse of the core's cryogenic systems. Moreover, high propellant mass is assigned to the upper stage with respect to the lower stage, since its structural ratio is underestimated by the weight models, hence favouring higher propellant loads.
- The maximum allowed trajectory loads $q_{dyn,max}$, $q_{heat,max}$ and $n_{ax,max}$ are all decreased in the optimization process, since this reduces the inert mass while still meeting the path constraints. As a consequence, the dynamic pressure constraint becomes active, although the optimal solution presents a steeper trajectory with respect to the nominal ascent from the MDA, flying a higher q_{dyn} profile. *Note how this is a rather synthetic result, because no actual structural analysis is performed to validate the mass reduction associated to the lower trajectory loads.*
- The visual comparison of the launcher external geometries confirms the decrease in volume of the boosters and a shortening of the core, with the ***(L/D)*** ratio of the EAPs not significantly affected by the optimization.
- Although the MDO's objective is GTOW minimization, the CpL is also reduced due to the mass-based nature of the CERs, with a ***20.6%*** lower total cost. The ELV's reliability is instead only marginally affected by the MDO process, since architecture, technologies and flight phases are unchanged and continuous variables have a limited impact.

	Actual	**MDA**		**PSO optimal**		**WORHP optimal**	
$M_{prop,EPC}$ [tons]	173.3	173.3	-	138.6	-20.0%	135.9	-21.2%
$M_{prop,ESC\text{-}A}$ [tons]	14.4	14.4	-	16.8	+16.7%	17.2	+19.4%
$M_{prop,P241}$ [tons] (each)	240.1	240.1	-	147.9	-38.4%	144.1	-40.0%
$(L/D)_{P241}$ []	11.01	10.74	-2.4%	10.44	-5.2%	9.67	-12.2%
$q_{dyn,max}$ [kPa]	57.0	57.0	-	50.6	-11.2%	50.4	-11.6%
$q_{heat,max}$ [MW/m^2]	40.0	40.0	-	28.0	-30.0%	28.0	-30.0%
$n_{ax,max}$ [g]	4.55	4.55	-	3.79	-16.7%	3.77	-17.1%
PL to GTO [kg]	**10050**	**12472**	+24.1%	**10050**	**-**	**10050**	-
GTOW [tons]	**766.4**	**764.8**	-0.3%	**522.4**	-31.8%	**505.8**	-34.0%
CpL [M€]	**173.0**	**171.6**	-0.8%	**142.0**	-17.9%	**137.4**	-20.6%
MSP []	**0.970**	**0.927**	-4.4%	**0.926**	-4.5%	**0.926**	-4.5%

Table 27: Optimization variables and objectives resulting from the global and local, minimum mass, MDO of Ariane 5 ECA problem, in comparison with the actual launcher's design and the outputs resulting from the MDA.

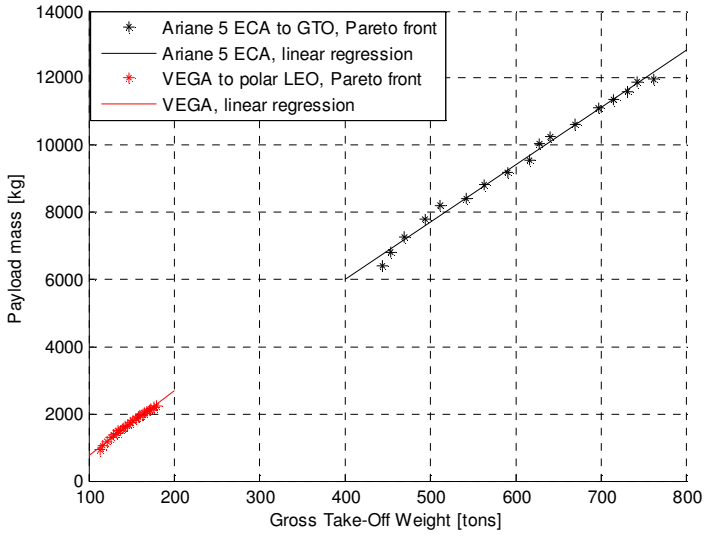


Figure 76: Min GTOW vs max PL Pareto fronts for Ariane 5 ECA and VEGA, including linear regression defining $\partial PL/\partial GTOW$.

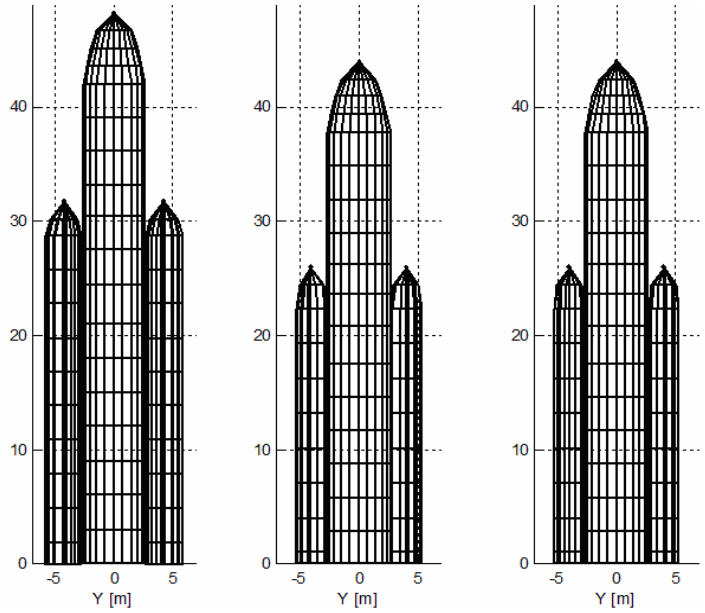


Figure 77: Geometry of Ariane 5 ECA, comparison of MDA (left), best solution from PSO-1D (centre) and WORHP refinement (right).

5.5.2. VEGA

As for Ariane, MDO results are presented in **Table 28** and **Figure 78** for the numerical values and external geometry. The following remarks add to those of the previous paragraph:

- The minimum launch mass for the mass-optimized VEGA is of ***132 tons***, with a very limited ***0.3%*** improvement obtained through local refinement of PSO-1D design. This 4.6% reduction in GTOW with respect to the real VEGA actually corresponds to an ***8.9%*** improvement in GTOW, assuming a $\partial PL/\partial GTOW = 18.8\ kg\ /\ ton$ as derived above. As for Ariane, both re-allocation of the propellant masses and reduction of trajectory loads contribute to this result.
- As expectable, the optimized AVUM contains much more propellant with respect to the original 550 kg, with the upper bound set to 30% always being hit. This testifies how the LP upper stage should be definitely enlarged to improve VEGA's performance, which is of course reasonable considering that the main purpose of the current AVUM is only to provide flexibility on the range of target orbits. With respect to the SRMs, the optimization pushes for smaller first and third stages, in favour of a larger Z23.
- The maximum allowed trajectory loads $q_{dyn,max}$, $q_{heat,max}$ and $n_{ax,max}$ are again decreased, but in case of VEGA the axial acceleration results to be the active constraint. Although the dynamic pressure constraint is not active, its maximum value is only reduced by ~1% in the final solution, proving the MDO is not fully converged.
- Regarding the external geometry, the optimized launcher is thinner than the actual VEGA, with an increase in the (L/D) ratio of both the first and third stage, both pushed to the upper limit for aerodynamic reasons. The static controllability constraint in a gravity turn atmospheric ascent without wind is in fact not sufficient to counteract the aerodynamic push towards thinner launchers, since the design models or V1 lack of any consideration about launcher's flexibility and structural sizing of thin fuselages.
- Considerations regarding CpL and MSP are the same as for Ariane.

	Actual	**MDA**		**PSO optimal**		**WORHP optimal**	
$M_{prop,P80}$ [tons]	87.8	87.8	-	83.0	-5.5%	81.8	-6.8%
$M_{prop,Z23}$ [tons]	23.8	23.8	-	23.6	-0.8%	26.9	+13.0%
$M_{prop,Z9}$ [tons]	10.6	10.6	-	10.0	-5.7%	8.9	-16.0%
$M_{prop,AVUM}$ [kg]	550	550	-	715	+30.0%	715	+30.0%
$(L/D)_{P80}$ []	3.56	4.07	+14.3%	4.88	+37.1%	4.88	+37.1%
$(L/D)_{Z9}$ []	1.91	2.74	+43.5%	3.29	+72.3%	3.29	+72.3%
$q_{dyn,max}$ [kPa]	57.0	57.0	-	55.9	-2.0%	56.4	-1.1%
$q_{heat,max}$ [MW/m^2]	40.0	40.0	-	28.0	-30.0%	28.0	-30.0%
$n_{ax,max}$ [g]	5.50	5.50	-	4.44	-19.3%	4.38	-20.4%
PL to GTO [kg]	**1500**	**1403**	-6.5%	**1500**	**-**	**1500**	-
GTOW [tons]	**138.1**	**139.4**	+0.9%	**132.0**	-4.4%	**131.7**	-4.6%
CpL [M€]	**32.0**	**34.8**	+8.8%	**33.3**	+4.1%	**33.2**	+3.8%
MSP []	**0.980**	**0.975**	-0.5%	**0.975**	-0.5%	**0.975**	-0.5%

Table 28: Optimization variables and objectives resulting from the global and local, minimum mass, MDO of VEGA problem, in comparison with the actual launcher's design and the outputs resulting from the MDA.

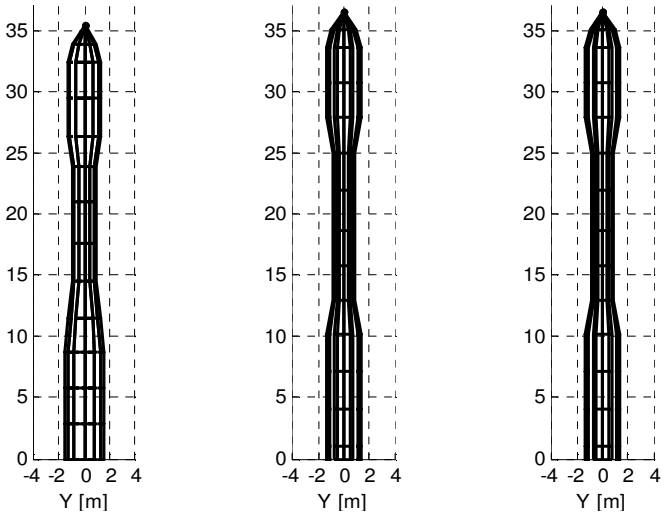


Figure 78: Geometry of VEGA, comparison of MDA (left), best solution from PSO-1D (centre) and WORHP refinement (right).

5.6. Summary of conceptual level modelling

The first modelling step for ELVs, presented throughout **Chapters 4** and **5**, allowed to collect a large number of lessons learned on several significant aspects of ELVs design. These were all incorporated in the second engineering step, giving birth to the multidisciplinary models for V2 (see **Chapters 6** and **7**). The key remarks originated from the modelling and validation process are summarized here for better clarity:

- The assessment of LP and SP rocket engines' performance with CEA and empirical corrections is particularly good. Specifically, the accuracy on the specific impulse is suitable to both conceptual and early preliminary design levels.
- Other important features are lacking in the Propulsion discipline. In particular, an analysis of the correlation between pressurization system, tanks pressure, chamber pressure and turbopumps system for LP as well as realistic representation of thrust profiles for SP are missing. Moreover, an improvement in the estimation of the overall inert masses of the propulsion system is necessary, in combination with a generic improvement of all weight models.
- External geometry definition and visualization is sufficient for the purpose of conceptual/early preliminary MDO of ELVs. However, additional features should be modelled to improve the lengths assessment of specific launcher configurations, such as "under fairing" upper stages, multiple and enclosed tanks, submerged SP nozzles and extendable LP nozzles, better definition of interstage separation planes and forward/aft skirts.
- In spite of the low fidelity of the DATCOM-based analysis, Aerodynamics discipline seems adequate in light of a very low sensitivity of the global performance to changes in drag and lift. Besides, accuracy improvements enabled by linear aerodynamics panel methods are not necessarily very large, at the price of a consistent integration and computation effort, and CFD is for the time being excessive for conceptual MDO of launchers.
- Weights analysis through WERs is the most critical point of the developed multidisciplinary model, as evinced from SA, MDAs and MDOs. A more accurate representation of the physics behind the evaluation of the structural masses is mandatory in order to entrust the MDO design solutions. Most of the errors in SA, MDA and MDO testing were in fact ascribed to low fidelity of the weight models, and the dependence of the structural masses on the trajectory load parameters determines a synthetic reduction in inert masses and therefore overall GTOW of the system.
- Although environmental and 3 DoF trajectory simulation models are accurate, the assessment of the payload mass of a given launch vehicle through trajectory optimization is optimistic (***+9%*** for Ariane, ***+14%*** for VEGA). Two main causes of the performance overestimation were identified: the lack of several modelling aspects playing an important role in the performance of ELVs (realistic thrust profiles, wind and manoeuvre steering losses, specific impulse degradation over time) and the inaccuracy of the launchers' input data, which is intrinsic to the validation approach.
- In terms of overall accuracy, Montecarlo SA runs show ***8%*** and ***16%*** 1σ variability for Ariane 5 and VEGA performances, with a worst case 1σ expectable error due to (μ,σ) ***~19%***, which is reasonable for this level of fidelity.
- MDA of existing launchers provide performance errors mainly consistent with the SAs, with a ***+24%*** and ***-6%*** over/underestimation of the payload mass for Ariane 5 ECA and VEGA. The main cause for these errors can be identified in poor modelling of the inert masses, in particular for ESC-A (***-3.2 tons*** due to length errors and lack of skirts modelling), VEGA's SRMs (***[+1.6, +0.7, -0.1] tons*** due to inadequate representation of advanced technologies in casing and mostly SP nozzle), and AVUM (***+130 kg*** due to lack of under fairing configuration).
- Small MDO problems for minimum launch mass can be successfully solved through PSO-1D global runs and successive WORHP local refinements, producing consistent improvements in the mass performance of Ariane and VEGA, respectively ***~20%*** and ***~9%*** with respect to the GTOW from MDA corrected for PL performance errors. Although part of this advantage is synthetically introduced through the reduction of the trajectory loads, the propellant masses are also rearranged with well represented physical trade-offs (e.g. push towards lower bound for EAP's low I_{sp} SP mass and towards the upper bound for the higher I_{sp} LP mass of AVUM). Moreover, according to the developed models, VEGA's actual design seems more GTOW-optimized with respect to Ariane 5's.
- Costs estimation within the MDA is sufficiently accurate in terms of overall CpL figure as well as of approximate breakdown of production, development and operation costs (***3/5, 1/5, 1/5*** for Ariane and ***3/6, 1/6, 2/6*** for VEGA whose SP technology allows for lower development/production costs with respect to Ariane 5's cryogenic systems). Moreover, the LCC of the system is reduced in min-GTOW MDO processes, due to the mass-based nature of the CERs, which does also however complicate the identification of mass vs. cost Pareto fronts in multi-objective MDO.
- Finally, reliability assessment results in reasonable estimates of the MSP for VEGA but highly underestimates that of Ariane 5, due to the high theoretical failure rates associated with cryogenic systems, which have been disregarded by the exceptional launch success record of the ECA version. Besides, reliability is shown to be entirely determined by the architecture and technologies employed in the launch vehicle, with the size and other continuous characteristics of the system being basically uninfluential.

There are a thousand things that can happen when you go light a rocket engine,
and only one of them is good.

CHAPTER 6

EARLY PRELIMINARY DESIGN MODELS

The upgraded multidisciplinary models for the early preliminary design of ELVs directly stem from the critical analysis of the validation of the conceptual level models, whose key points are synthetically summarized in **Section 5.6**. In addition to these remarks, an internal ESA review of the documentation for V1 highlighted a number of weaknesses, classified in Review Item Discrepancies (RIDs). Several RIDs coincide with the inaccuracies evinced by the validation process, but others represent lacking modelling features, especially related to industrial requirements in ELVs design, which were also incorporated in the model enhancements for V2.

Following the descriptive approach of the conceptual level models in **Chapter 4**, the early preliminary MDA cycle and the related disciplinary analyses are thoroughly documented in this chapter (a more comprehensive definition is given in **[A3]**). Note that for several aspects, the description of the engineering models for V2 retraces that of V1, with large parts in common. Redundant information is therefore only recalled from **Chapter 4** - avoiding repetitions - so that most of the disciplinary analysis sections below basically consist in a collection of model enhancements over the conceptual level version.

6.1. Multidisciplinary design cycle and optimization problem

6.1.1. Design Structure Matrix

The DSM in **Figure 79**, describing the enhanced MDA cycle, is not widely different from that of the original models. In fact, despite the large set of improvements in all disciplines which is reflected in most of the vectors $\boldsymbol{P_j}$, $\boldsymbol{X_j}$, $\boldsymbol{X_{jk}}$ and $\boldsymbol{Y_j}$ including a larger number of parameters, the basic structure of the MDA remains unchanged. The only major difference lays in the introduction of the structural sizing discipline, which is iterated with the Trajectory block to achieve convergence on the inert mass of fairing, stages and boosters, closing the performance cycle. In fact, loads from the trajectory are used as inputs to size the structural components of the launcher, therefore changing its inert masses which are in turn used to simulate/optimize the trajectory.

This iterative procedure is well suited to the application of two different problem formulations, as recalled from the current state-of-art in MDO (**Section 2.2**):

- **MDF**: for each design solution, the Trajectory-Structures iterative cycle is solved, until convergence on the inert mass of all launcher components is achieved, with some predefined tolerance. This ensures physical consistency of the design at any point during the MDO process, but requires longer computational times per design solution.
- **IDF**: Trajectory-Structures iterations are avoided by defining the inert mass of the fairing/stages/boosters through optimization variables, in a given range around the actual value from Weights assessment. These inert masses, "guessed" by the optimizer, are first used for the integration of the EoMs and then compared with the actual masses obtained from the structural sizing, imposing compatibility constraints. This therefore results in the introduction of $\boldsymbol{N_{stages}+N_{bsets}+1}$ new optimization variables and constraints, enlarging the optimization problem, but with the benefit of avoiding the computational times associated with the iterative cycle. The optimizer is then in charge of adjusting these so-called *"slack"* optimization variables to match the compatibility constraints, so that physical consistency is ensured only at the optimization level.

Although the IDF formulation has been reported in literature to be superior to the MDF formulation in several applications, this cannot be considered a result of general validity, being strictly dependant on the characteristics of the problem at hand: number of iterations generally required for convergence, time spent in the iterative loop, difficulty of achieving interdisciplinary compatibility, and other features. In particular, experience with the MDA cycle developed for

V2 showed that the number of iterations required in MDF mode for convergence is very limited. For instance, 2 iterations are generally sufficient with tolerances in the order of 1% of the actual mass. An a priori evaluation of the convenience of MDF or IDF approaches cannot therefore be performed, but should be assessed on a case-by-case basis (see **Section 8.1** for results on this aspect).

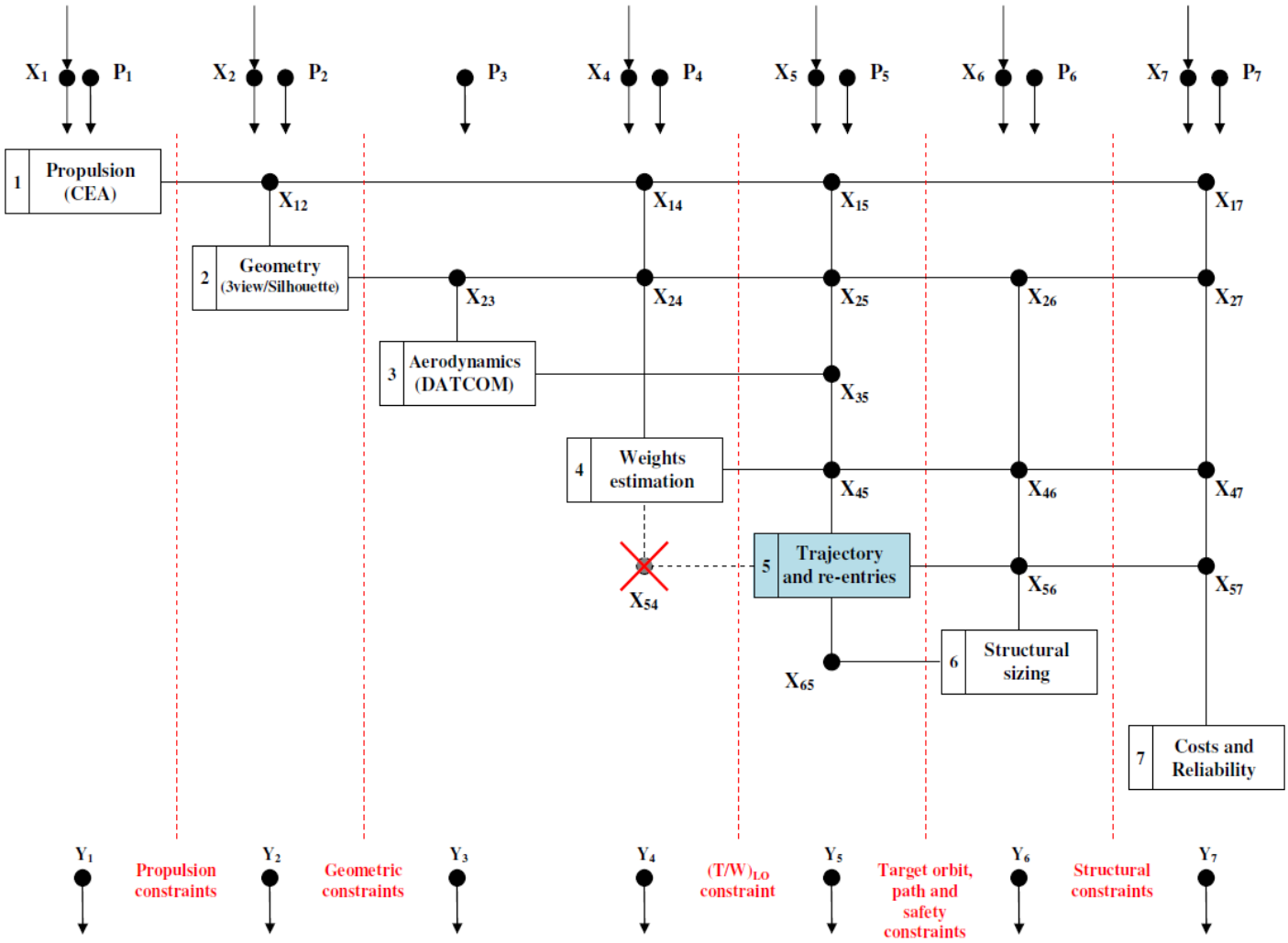


Figure 79: DSM for the early-preliminary design (V2 models) of classical unmanned ELVs.

Briefly going through the MDA, the bulk of Propulsion discipline with CEA and semi-empirical models is unchanged, but is complemented with several new analyses such as SP grain shape modelling, LP pressurization system and pumps cavitation analysis, and improved inert masses estimation. The design cycle continues with Geometry, which also introduces new modelling capabilities such as separation plane definition, under fairing upper stage configuration, multiple (e.g. AVUM) and enclosed tanks (e.g. ESC-A), and optimizable nose ogive's length. Aerodynamics analysis is instead identical to V1, with the only exception of allowing for more than one aerodynamic configuration, depending on the jettison status, which may only be useful in very specific conditions such as with extremely small strap-on boosters. Also Weights module is basically unchanged. However, it is only used for an initial guess definition of the inert masses, which are either used for the first iteration of the Trajectory simulation in MDF mode or constitute the reference value for the inert mass optimization in IDF mode.

Trajectory simulation (or optimization in case of NOL) is then based on the same 3-DoF modelling of V1, but includes a wide range of new functionalities. In particular, branching trajectories are simulated for the re-entry of all suborbital components, with the purpose of evaluating the splash-down safety constraints. Moreover, steering losses associated to both wind profiles and discrete manoeuvres are added for degraded effective ΔV as well as for improved static controllability verification. Following Trajectory, the Structural sizing is executed, being constituted of an entirely new module with analytical engineering-level methods for the estimation of the mass of the main structural elements. Structural material and layout for each element are introduced here as optimization variables, whereas shell and frame thicknesses are sized to withstand the flight and pressure loads. Finally, after one (IDF) or more (MDF) executions of the Trajectory-Structures

technical disciplines, cost and reliability models conclude the MDA, without any particular improvement over the original models. As a final note, several extensions were considered, the most relevant being a multi-body, 6-DoF dynamic controllability verification of the nominal trajectory flyability, which was prototyped in an external Matlab/Simulink environment but then not integrated for both time and computational efforts constraints.

6.1.2. User parameters, optimization variables, constraints and objectives

Since most of the elements of the vectors $\boldsymbol{P_j}$, $\boldsymbol{X_j}$, $\boldsymbol{X_{jk}}$, $\boldsymbol{Y_j}$ shown in the DSM for the early preliminary design of ELVs (**Figure 79**) are unchanged with respect to the conceptual level models, no table detailing each element for each vector is reported here. However, user parameters, optimization variables, constraints and objectives modified or added for V2 are outlined here.

In particular, the user parameters described in **Paragraph 4.1.2** were complemented with additional settings in Propulsion, Trajectory, and Structures disciplines. Note that in order to minimize the effort required to convert input files from V1 to V2 models, all parameters described below are implemented with built-in defaults, so that the user is not forced to provide numerical figures but can straightforwardly reuse V1's input files.

P_1, Propulsion models settings (see Section 6.2 for details):

- Custom SRM profiles: optional detailed definition of the thrust and specific impulse profiles in new design SRMs.
- Custom propellant properties: optional user definition of the LP or SP physical properties (e.g. μ, p_{vapour}, ρ, T_{flame}, ...)
- End of Life strategy: optional user selection of the strategy to be employed for the end of life of orbital stages.

P_5, Trajectory models settings (see Section 6.4 for details):

- Wind model: reference altitude and scaling factor for the synthetic wind profile or, if necessary, full wind profile.
- Re-entry model: selection of re-entry analyses mode (no safety constraints vs. numerical integration vs. Keplerian propagation), impact ellipse settings, population density map resolution and threshold, user-defined population map.

P_6, Structures models settings (see Section 6.5 for details):

- Load cases definition: time discretization for load cases events, aerodynamic forces split coefficients.
- Launcher beams approximation: discretization settings for calculation of both external and internal running loads.

As regards to the trajectory and design variables, the most generic optimization problem was enlarged quite significantly from the conceptual to the early preliminary model. Although the number of trajectory variables is not changed, the number of system and component level design variables is increased from $\boldsymbol{N_{SDV}=14}$ to $\boldsymbol{N_{SDV}=22}$ and from $\boldsymbol{N_{CDV}=26}$ to $\boldsymbol{N_{CDV}=73}$. The new variables, introduced for handling more detailed definition of ELVs' system and subsystems as described below, combine with those already available for conceptual design to constitute the full vector of optimization variables $\boldsymbol{X}$:

$$
\begin{cases}
X = \{X_{SDV}, X_{CDV,1}, ..., X_{CDV,N}, X_{TDV}\} \\
X_{SDV} = \{N_{stages}, Conf, CCB, N_{boosters}, SET, SBT, n_{ax,\max}, q_{dyn,\max}, q_{heat,\max}, RSC, SSM, RL, LPM, ... \\
\qquad ..., SM_{PLF}, SC_{PLF}, SM_{PLA}, SC_{PLA}, M_{slack,PLF}, (L/D)_{nose}, UFC, K_{UF}, PIC\} \\
X_{CDV,j} = \{OTS, OTS_{ID}, N_{eng}, Prop, Feed, Fuel, M_P, T^*, \alpha_P, \epsilon, p_{cc}, p_{tanks}, Nozzle_{ext}, N_{tc}, PressSys, p_{press}, ... \\
\qquad ...Nozzle_{sub}, r_b, SRM_{type}, N_{segm}, Shape, L_{rel}, w_g, N_g, R_{gi}, f_g, \epsilon_g, A_{cc}/A_t, LCE, N_{eng,tests}, N_{eng,out}, TOE, ... \\
\qquad ...\text{Nozzle}, \delta_{\text{div}}, \text{TVC}, \delta_{TVC}, ThC, D_{const}, D, TT, TA, SM_{main}, SC_{main}, SM_{sec}, SC_{sec}, M_{slack}\} \\
X_{TDV} = \{PLSF, \Delta\theta_{PO}, t_{PO}, t_{PO,dec}, \Delta\psi_{PO}, \theta_j, \psi_j, \xi_{BTL}, \Delta\theta_{BTL,i}, \theta_{BTL,f}, \delta_{th,j}, \delta_{SP,j}, T^*_{\min,j}, t_{switch,j}, \Delta_{CI}, \Delta t_{CI}, \Delta_{BI}, \Delta t_{BI}, \Delta t_{LSCj}, \Delta t_{CB}\}
\end{cases}
$$

Launcher architecture: No additional variables with respect to the conceptual models.

Trajectory loads:

Although the trajectory load variables $\boldsymbol{n_{ax,max}}$, $\boldsymbol{q_{dyn,max}}$ and $\boldsymbol{q_{heat,max}}$ can still be used as optimization variables, their relevance with the introduction of the structural sizing is very limited. This is a consequence of the inadequacy of these variables to correctly represent loads vs inert masses trade-offs, as highlighted within the validation of V1 models. In V2, their effect is therefore limited to the "first guess" value of the inert masses estimated in the Weights discipline, whereas the final values are only dependent on the actual trajectory loads through the structural analysis.

System design features:

The following additional system design optimization variables are included in the early preliminary models:

- $\boldsymbol{SM_{PLF}}$, $\boldsymbol{SC_{PLF}}$, $\boldsymbol{SM_{PLA}}$, $\boldsymbol{SC_{PLA}}$: integer variables for the structural material ***SM*** and structural configuration ***SC*** (i.e. stiffening lay-out) of the PLF and PLA components, as described in the Structures discipline. See **Section 6.5**.
- $\boldsymbol{M_{slack,PLF}}$: slack inert mass continuous optimization variable for the PLF (only for IDF problem formulation).
- ***UFC*** and $\boldsymbol{K_{UF}}$: upper stage's under-fairing configuration option activation (boolean ***UFC***) and percentage of upper stage's length contained in the PLF for ***UFC=true*** (continuous variable $\boldsymbol{K_{UF}}$). See **Paragraph 6.3.3**.

- $(L/D)_{nose}$: length-over-diameter ratio of the PLF's power law nose ogive, which can be defined as a continuous optimization variable in V2 models for structural vs. aerodynamic trade-offs.
- ***PIC***: integer variable defining the pad interface configuration for architectures including boosters, which can be optimized with the early preliminary models because of the higher accuracy and physics-based trade-offs which are enabled by the structural analysis. ***PIC=1/2/3*** indicates a pad interface with the core, first or second set of boosters.

Propulsion system architecture:

- Fuel: type of liquid fuel or of solid propellant formulation, see **Paragraph 6.2.4.4**.

Propulsion system design:

LP/SP systems detailed design is the area which has seen the largest number of added variables. In particular, for LP:

- p_{tanks}: the pressure in LP tanks is not fixed by default (or by the user) as in V1, but can be optimized through this continuous variable to identify trade-offs between higher performances and lower inert weights, through the new models for improved pressurization system mass assessment and pump cavitation analysis (see **Paragraph 6.2.3**).
- N_{tc}: integer variable defining the number of thrust chambers for each LRE, from 1 to 5 (see **6.2.4.1**)
- $Nozzle_{ext}$: boolean variable activating the extendable nozzle option for upper stage LREs (see **6.2.4.2**).
- ***PressSys***: integer variable defining the type of pressurization system, as defined in **Paragraph 6.2.3**.
- p_{press}: continuous variable defining the pressure in pressurization gas tank, for cold/heated He pressurization systems.

And for SP systems (see **Paragraph 6.2.2** for details on all these variables):

- $Nozzle_{sub}$: Boolean variable activating the submerged nozzle option for SRMs (see **6.2.4.3**).
- r_b: SP burn rate, adjustable through r_b modifiers to match web length and burn time.
- SRM_{type}: type of SRM: end-burning, internal-burning or custom.
- N_{segm}: number of SP grain segments with different section's geometric shape.
- $L_{rel,j}$: ratio of the length of j-th segment to the total grain's length.
- $Shape_{,j}$, $w_{g,j}$, $N_{g,j}$, $R_{gi,j}$, $f_{g,j}$, $\varepsilon_{g,j}$: shape of the j-th SP grain segment and related geometric adimensional parameters.

Note that the maximum number of design variables related to the grain detailed design is 35, assuming at most 5 segments each defined by 7 variables (length, shape and five geometric parameters).

Propulsion system costs and reliability, Geometry: No new design variables.

Structural layout:

- SM_{main}, SC_{main}, SM_{sec}, SC_{sec}: integer variables for the structural material ***SM*** and structural configuration ***SC*** of the main components containing the propellants - i.e. OxT/FT/SpC - and the other flight structures (see **Section 6.5**).
- M_{slack}: slack inert mass continuous optimization variable for the stage/booster (only for IDF problem formulation).

In addition to the introduction of new variables, two changes were performed in the definition of optimization variables: the diameter D was substituted to the length-over-diameter L/D for all stages with non constant diameter and for boosters, in order to provide a more physically representative optimization variable. For the same reason, the nozzle's expansion ratio $\varepsilon=A_e/A_t$ was substituted as input optimization variable to the nozzle optimal expansion altitude H^*, therefore becoming an output of the CEA analysis.

Finally, the optimization objectives were not affected by the upgrades to the models, whereas a few constraints were added in light of the new analyses. **Table 29** summarizes all ELV's design and trajectory constraints defined for V2, with those specifically added for the upgraded models in italics.

Discipline	Constraint	Type	Applicability	LB	UB
Propulsion	*SP filling factor*	*Double*	*Each stage/booster (SP only)*	-∞	*0*
Propulsion	Thrust range	Double	Each stage/booster (OTS only)	-1	1
Propulsion	CEA execution failure	Boolean	Each stage/booster	-∞	0
Propulsion	Number of engines	Integer	Each stage/booster (LP only)	-∞	0
Propulsion	*Grain design*	*Double*	*Each stage/booster (SP only)*	-∞	*0*
Propulsion	*Web length matching*	*Double*	*Each stage/booster (SP only)*	-∞	*0*
Geometry	Geometric interference	Double	Each stage/booster	-∞	0
Geometry	Gimbal clearance	Double	Each stage/booster	-∞	0
Weights	*Dry mass slack*	*Double*	*Each stage/booster and PLF (IDF only)*	*0*	∞
Geometry	Length-over-diameter	Double	Launch vehicle	0	1
Geometry	Fairing geom.. interference	Double	Payload fairing	-∞	0
Geometry	Geometry execution failure	Boolean	System analysis	-∞	0
Aerodynamics	DATCOM execution failure	Boolean	System analysis	-∞	0
Weights	Weights execution failure	Boolean	System analysis	-∞	0
Weights	Lift-off thrust-to-weight	Double	Launch vehicle	0	1

Weights	Fairing dry mass slack	Double	Payload fairing (IDF only)	0	∞
Costs	Costs execution failure	Boolean	System analysis	-∞	0
Risks	Risks execution failure	Boolean	System analysis	-∞	0
Trajectory	Target semiaxis	Double	Final conditions	-1	1
Trajectory	Target eccentricity	Double	Final conditions	-1	1
Trajectory	Target inclination	Double	Final conditions	-1	1
Trajectory	Path axial acceleration	Double	Maximum of all time instants	-∞	0
Trajectory	Path heat flux	Double	Maximum of all time instants	-∞	0
Trajectory	Path dynamic pressure	Double	Maximum of all time instants	-∞	0
Trajectory	Path static controllability	Double	Maximum of all time instants	-∞	0
Trajectory	Angle of attack	Double	Maximum of all time instants	-∞	0
Trajectory	*Total integrated heat load*	*Double*	*Integral over time*	-∞	*0*
Trajectory	Nozzle minimum altitude	Double	Each stage/booster's ignition	0	∞
Trajectory	Geographic heading	Double	Each allowed corridor at lift-off	$\chi_{min,j}$	$\chi_{max,j}$
Trajectory	*Impact population density*	*Double*	*Each stage/booster's impact*	-∞	*0*

Table 29: List of ELVs' design and trajectory constraints, including involved discipline, type of constraint, applicability to single components or to the whole launch vehicle, and feasibility's lower and upper bounds. *In italics: constraints added for V2.*

6.2. Propulsion

Although the core of the performance analysis with CEA and empirical I_{sp} losses is the same as for V1, the Propulsion discipline was widely expanded for V2, in particular including OTS SRMs definition, SP grain geometric design and internal ballistics analysis, LP pressurization system and pumps cavitation assessment, new modelling options for both SP and LP engines, and enhanced I_{sp} losses and inert masses evaluation. The analysis cycle for the propulsion system of each stage/booster is the same as described in **Section 4.2**, with the vector of input design variables $\boldsymbol{X_j}$ and of output parameters $\boldsymbol{X_{jk}}$ and $\boldsymbol{Y_j}$ only slightly enlarged.

6.2.1. Off-The-Shelf Solid Rocket Motors

In order to allow for a wider range of launcher architectures, the OTS engines database was extended to include SRMs of different types, suitable for both booster applications and upper stage kick motors. The same database structure was maintained for both SP and LP engines. In addition to the European P241, P80, Z23 and Z9, the following selection of motors from ATK's online catalogue (**[123]**) was added to the database: Shuttle's RSRM, Orion38, Orion50, Orion50S, Castor-IVA, Castor120, GEM-40, GEM- 40air, GEM-60, STAR-37FM, and STAR-48A. Some examples of the currently available SRMs are reported in **Table 30**, showing the most relevant design parameters. A functionality introduced along with the OTS SP engines is the possibility to define custom thrust and specific impulse profiles. An input file containing a list of normalized times, $\boldsymbol{T}$ and $\boldsymbol{I_{sp}}$ can be provided by the user, or one of the available files for the above mentioned motors can be alternatively loaded.

Engine	Application	Propellants	p_{cc} [bar]	ε []	$T_{nom,vac}$ [kN]	T profile	$I_{sp,vac}$ [s]	Φ_e [m]	L_{engine} [m]	M_{WER} [kg]
STS-SRB	Shuttle boosters	HTPB-AP-Al	46.0	11.3	11520.0	constant	268.6	3.80	-	-
P-241	Ariane 5 boosters	HTPB-AP-Al	64.0	11.0	6305.9	P-241.dat	275.3	2.98	3.78	6400.0
P-80	VEGA 1st stage	HTPB-AP-Al	95.0	16.0	2804.2	P-80.dat	280.1	1.98	-	2249.0
Z-23	VEGA 2nd stage	HTPB-AP-Al	106.0	25.0	1054.2	Z-23.dat	287.5	1.47	-	571.8
Z-9	VEGA 3rd stage	HTPB-AP-Al	74.0	56.0	304.9	Z-9.dat	294.4	1.23	-	227.7

Table 30: Examples of OTS SRMs implemented in the database for V2, including the most relevant design parameters

6.2.2. SP grain geometric design and internal ballistics analysis

The models developed for the grain analysis in V1 involve several modelling and design inaccuracies, highlighted during the conceptual models validation:

- Approximation of the thrust profiles (constant, 2-level, linear), with errors up to 20% over some parts of the burn.
- Assumption of constant burn area, chamber pressure and specific impulse over the entire burn phase. This is a small but still significant approximation, since $\boldsymbol{I_{sp,vac}}$ varies due to variations in the burning area and hence internal pressure, typically in the order of 1-3 seconds from maximum pressure to jettison instant (low residual pressure).
- Lack of matching constraint between Web Fraction (WF), internal pressure history and burn time (i.e. the burn time computed from regression rate and web length should be equal to the burn time computed from T and I_{sp} profiles).
- No evaluation of the Filling Factor (FF) and Sliver Fraction (SF) of the solid grain, assumed to be constant.

To solve these issues, a more detailed grain geometrical analysis and internal ballistics calculations were implemented, defining 3 possible options for new (i.e. non OTS) SRM design, selectable with an integer optimization variable $\boldsymbol{SRM_{type}}$:

1. *Custom SRM*: the user provides FF, SF, and normalized $T(t)$ and $I_{sp}(t)$ profiles (through data file) and the motor's thrust and dimensions (propellant mass) are allowed to be scaled in a range specified by the user. This option is useful for variations of existing SRMs, for which the grain characteristics are assumed to be maintained constant when varying the motor's scaling. For this reason, no geometric design nor internal ballistic analyses are performed, assuming the normalized $T(t)$ and $I_{sp}(t)$ profiles are always valid.
2. *End-burning motors*: to be used for upper stage or kick motors, it is the simplest possible grain design. Burn area, internal pressure, thrust, and specific impulse are assumed to be constant, with $FF=1$ and $SF=0$. Hence, no grain geometrical analysis is required, but the analysis of the matching of burn time and web length is executed.
3. *Internal-burning motors*: to be used for booster or lower stages applications, this option allows a rather detailed geometrical design of the solid propellant grain. The resulting thrust profiles are realistic representations of real world solid grains, even though the ignition and burn-out transitories are neglected (thrust build-up is assumed to be instantaneous and thrust varies according to the grain geometrical design until the sliver interface is reached). Up to 5 segments with different grain cross-sections and relative lengths can be specified, with three internal perforation types: a) tube grain, b) slot grain, or c) star grain. Geometric and internal ballistics analyses then allow computing the burn area, internal pressure, thrust, and specific impulse profiles.

A description of the overall SP motors design process, as well as of the single geometric and internal ballistics models, is reported in the following subparagraphs.

6.2.2.1. SRMs design process

The analysis flow for SRMs of V2 differs in part from what shown in Figure 13, and requires as input the following design optimization variables:

- M_{prop}: usable propellant mass, not include the sliver fraction.
- D: stage/booster diameter, assumed to be the same as for the solid grain ($D_{grain}=D$).
- p_{cc} and ε: maximum internal pressure throughout the whole burn phase and nozzle's expansion ratio A_e/A_t.
- SRM_{type}: *custom*, *end-burning* or *internal-burning*.
- r_b: for end-burning and internal-burning motors only, the burn rate in mm/s is given as an optimization variable to allow for matching of the WF constraint. Considering a given burn rate exponent $n=0.3$, the constant a is computed to match the optimized $r_b=a\cdot p_{cc}^{\,n}$, assuming that such value can be obtained by proper modification of the SP formulation with suitable quantities of burn-rate modifiers (e.g. iron oxide/lithium fluoride to increase/decrease r_b).
- N_{segm}: for internal-burning motors only, number of grain segments with different section shapes.
- $Shape_j$: for each segment j of internal-burning motors, grain section shape: *tube*, *slot* or *star*.
- $L_{rel,j}$: for each segment j except the first, relative length $L_{rel,j}=L_j/L_{grain}$.
- $w_{g,j}$, $N_{g,j}$, $R_{gi,j}$, $f_{g,j}$, $\varepsilon_{g,j}$: Adimensional geometric parameters describing the grain section. N_j, $R_{i,j}$, f_j only apply to slot and star grain shapes, ε_j only applies to star shape.

With these input parameters available, the SRM's design process is composed of the following steps:

1. Only for internal-burning motors with $N_{seg}>1$: the parameter w_j for all segments except the first is adjusted so that the WF is constant throughout all grain segments.
2. Only for internal-burning motors: a geometric analysis of each grain segment is executed, depending on the grain section's shape. All parameters are adimensionalized with the external radius of the grain, so that the output of the analyses is the burning perimeter per unit external radius $P_b/R_{ext}(t)$ as a function of normalized time from ignition to burn-out in [0;1]. Additional outputs are the segment's filling factor FF_j and sliver fraction SF_j.
3. Only for internal-burning motors: build up of the overall adimensional burn area $A_b/(R_{ext}\cdot L_{grain})\,(t)$, FF and SF from the data of each segment ($P_b/R_{ext}(t)$, $L_{rel,j}$, FF_j, SF_j).
4. *Grain scaling*: for all motor types, the length of the grain can be scaled with respect to the propellant mass and thus overall dimensions as: $L_{grain} = M_{prop} \Big/ \pi\cdot R_{ext}^2\cdot\rho_{SP}\cdot FF\cdot(1-SF)$. The dimensional burn area vs. normalized time $A_b(t)$ can at this point be computed, as well as the cavity volume and sliver mass as: $Vol_{cavity} = (1-FF)\cdot\pi\cdot R_{ext}^2\cdot L_{grain}$ and $M_{sliver} = SF\cdot M_{prop}$
5. CEA nominal performance analysis, specific impulse losses and minimum altitude evaluation, as for V1.
6. Engine nozzle's scaling: the maximum burn area $A_{b,max}$ and chamber pressure p_{cc} are used to compute the maximum mass flow from the grain regression rate equation, from which the throat area and therefore all other nozzle dimensions are obtained.
7. Propulsion system masses and dimensions estimation, again as for V1 with the exception of the grain length computation which is already available from the grain scaling step above.

8. Only for internal-burning motors: evaluation of chamber pressure, specific impulse and thrust profiles over time, which are obtained as correction factors with respect to the maximum values p_{cc}, I_{sp} and T^*. Two different methods are possible, to be selected by the user, one involving a CEA iterative loop for each normalized time point to compute the chamber pressure from the instantaneous burn area and throat area, and one approximated but much less computational intensive assuming constant chamber properties and isentropic expansion in the nozzle.
9. Only for end-burning and internal-burning motors: burn time and web length matching constraint evaluation.

6.2.2.2. Tube grain geometry analysis

Tube grain sections simply consist of a cylindrical perforation with linearly increasing burn perimeter. Although this type of thrust profile is seldom useful in launcher applications, several motors combine tube and star grain segments to obtain complex profiles which could not be obtained otherwise (e.g. 2 of the 3 segments of P241 booster have tube shape). Burn perimeter analysis is simple, with only one input parameter $w=L_{web}/R_{ext}$ defining the geometry, from which the normalized port radius can be computed as $R_p=1-w$, so that: $P_b(y)=2\cdot\pi\cdot(R_p+y)$ where y is the burning coordinate that goes from 0 to the normalized web length, or web fraction WF, which is equal for tube grains to the parameter w. The filling factor is easily obtained as $FF=1-R_p^2$, but the burn perimeter analysis does not allow determining the sliver fraction from the assessment of the remaining propellant on the walls, since this would be null. A conservative constant value $SF=0.002$ is however assumed for tube grains to account for asymmetries in the burn and the pressure deflagration limit.

6.2.2.3. Slot grain geometry analysis

The most general type of star-like grain section can be described by 7 parameters, as detailed in reference **[91]**. However, this results in a very complex geometrical analysis, with 16 different burn configurations. To simplify the model, two subsets of this generic geometric were selected, following the approach in reference **[92]**, namely the *truncated star* or *slot* geometry described in this paragraph and the *star* geometry described in the next. The slot geometry is shown in **Figure 80**, where the region depicted with Phase 3 is that considered as sliver. By normalizing all geometrical parameters with the external grain radius, the slot grain geometry can be described by only 4 optimization variables:

- N: number of slots, by default assumed in the range [4,10].
- w: normalized web length, by default in the range [0.02,0.5].
- R_i: normalized internal perforation radius, by default in the range [0.05,0.5].
- f: normalized slot radius, by default in the range [0.01,0.1].

The normalized port radius can then be obtained as $R_p=1-f-w$. Note that constraints are imposed on the values of the above parameters in order to avoid inadequate geometries, such as those with very low filling factor or excessive sliver fraction.

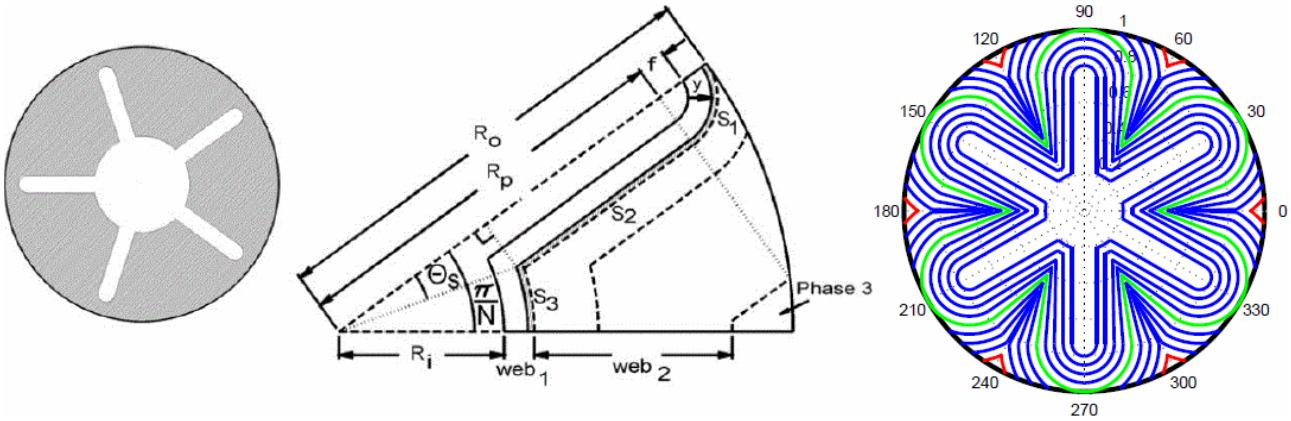


Figure 80: Slot grain geometry parameterization from [92] (left) and resulting burn back diagram from the geometric analysis (right). The green line represents the switch between Phase I and II burning, whereas the red line marks the sliver fraction.

The overall burn can be divided in two main burn phases, with different analytical expressions for the instantaneous burn perimeter and port area as a function of the burning coordinate y, directly taken from **[92]** and hence not reported here. The total normalized web length (i.e. web fraction) corresponds to the sum of webs 1 and 2, and has to be computed with different analytical formulas depending on the parameters as follows:

$$\begin{cases} WF=\sqrt{1-R_p^2}-f & if\ (1-f-w)\geq cos(\pi/N) \\ WF=\sqrt{(R_p\cdot sin(\pi/N))^2+(1-R_p\cdot cos(\pi/N))^2}-f & if\ (1-f-w)<cos(\pi/N) \end{cases}$$

The FF and SF are obtained instead from the initial and final port areas as $FF = 1 - Ap(0)/\pi$ and $FF = A_p(WF)/\pi$. Note that the second relation above corresponds to a grain geometry for which the burning segments S_1 and S_2 disappear before burn-out. In this case, the computed sliver fraction turns out to be null, with the burning perimeter reaching 0 at the final instant. To avoid this, the grain burn-out is assumed to occur when the burn perimeter reaches the 20% of the maximum value, so that the SF is recomputed at that point.

6.2.2.4. Star grain geometry analysis

The star geometry is shown in **Figure 81**, where the sliver region is the area close to the wall. As for the slot grain, all geometrical parameters are normalized with the external radius, so that 5 optimization variables are sufficient to univocally determine the adimensional geometry:

- ***N***: number of star points, with a default range in [3,15].
- ***w***: first web's normalized length, in the default range [0.3,0.7].
- R_i: star point normalized internal radius, in the range [0.01,0.5].
- ***f***: star's normalized curvature, in the range [0.01,0.15].
- ε: corresponding to the percentage of the star internal angle allocated for the tangential section, and may assume values in the range [0.5,0.9].

As for slot grains, the port radius is $R_p=1-f-w$, with constraints imposed to avoid inadequate configurations, in particular large sliver fraction and flat star point geometries. The web fraction in star grains corresponds to the parameter ***w***, and the analytical expressions for the instantaneous burn perimeter and port area as a function of y, again divided in two burn phases and allowing also to compute FF and SF, can be found in reference **[92]**.

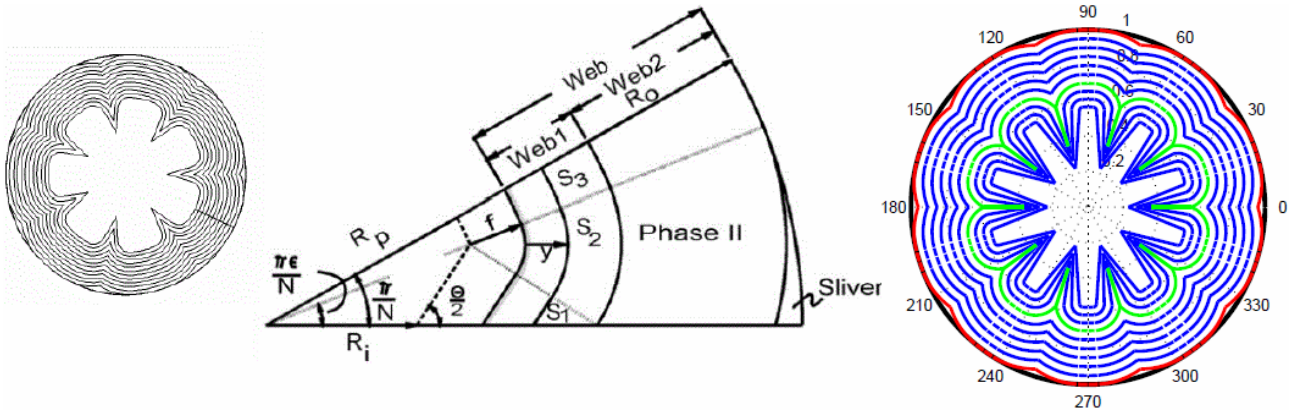


Figure 81: Star grain geometry parameterization from [92] (left) and resulting burn back diagram from the geometric analysis (right). The green line represents the switch between Phase I and II burning, whereas the red line marks the sliver fraction.

6.2.2.5. Grain segments build-up and grain scaling:

The burn properties of the three-dimensional SP grain are obtained from the sum of the properties of each segment, weighted with its relative length $L_{rel,j}$. The parameter ***w*** of all segments is instead rescaled so that WF_j for all segments equals that of the first segment. This ensures a consistent web fraction throughout the different segments, which avoids the flame to reach the external wall before the motor's burn-out. The adimensional burn area profile is also directly obtained from the normalized burn perimeters, though an interpolation scheme is necessary to account for different burn coordinate discretization in different segments. As soon as the solid grain outer dimensions are scaled with the values given by the optimizer of propellant mass and diameter, the dimensional burn area profile $A_b(t)$ can finally be computed.

Two examples of possible single-segment and dual-segment grain configurations are reported in **Figure 82** and **Figure 83**, showing the burn area profile as a function of burn time, both normalized in [0; 1]. Here, the grain geometries were optimized with a fast PSO-1D run to follow existing normalized thrust profiles from VEGA motors, in particular the P80 thrust profile through a single-segment slot grain and the Z23 thrust profile, through a dual-segment, tube and slot grain.

6.2.2.6. CEA analysis, nozzle scaling and (p_{cc}, T, I_{sp}) profiles:

CEA input file writing, execution and output file reading, as well as losses and minimum altitude evaluation, are executed exactly as for V1, except for the exploitation of the expansion ratio as input variable instead of the nozzle optimal expansion pressure. The CEA analysis is referred to the maximum thrust conditions, which corresponds to the maximum internal pressure. Engine nozzle's scaling instead differs from the classical approach for liquid engines, since the maximum

mass flow (and hence the nominal thrust) is no longer a free parameter but is determined from the solid grain regression rate relation. Hence, the following equations apply, allowing to obtain mass flow, nominal thrust and throat area:

$$\begin{cases} T^* = I_{sp} \cdot g \cdot \dot{m}_{max}, \quad with \quad \dot{m}_{max} = \rho_{SP} \cdot A_{b,max} \cdot a \cdot p_{cc}^n \\ A_t = C^* \cdot \eta_{C^*} \cdot \dot{m}_{max} \,/\, p_{cc} \end{cases}$$

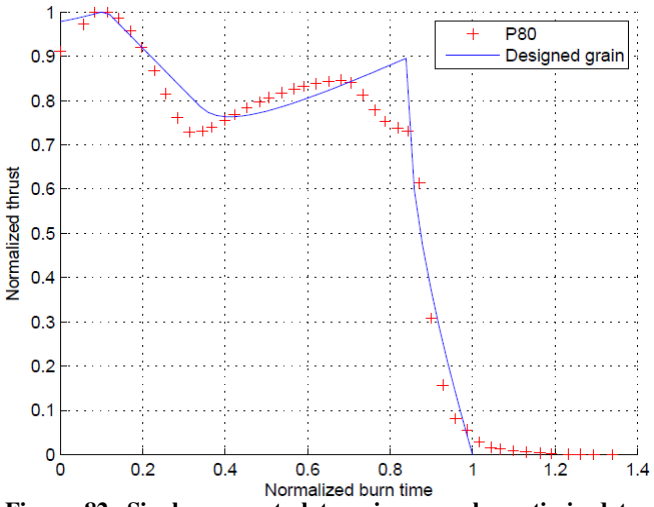


Figure 82: Single-segment slot grain example, optimized to match P80 thrust profile: normalized burn area profile

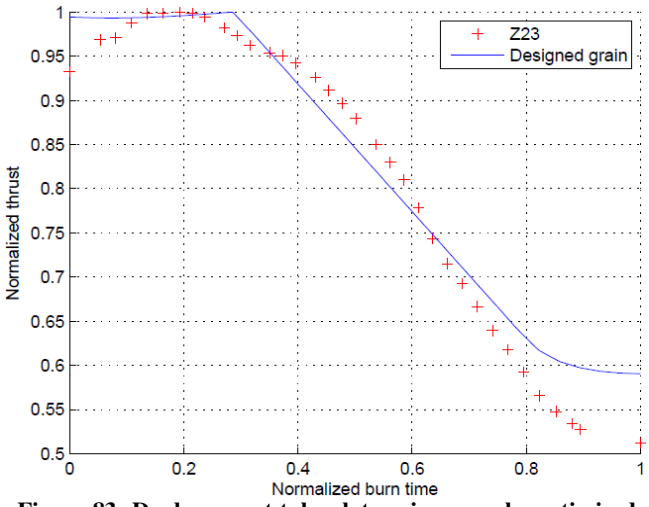


Figure 83: Dual-segment tube-slot grain example, optimized to match Z23 thrust profile: normalized burn area profile

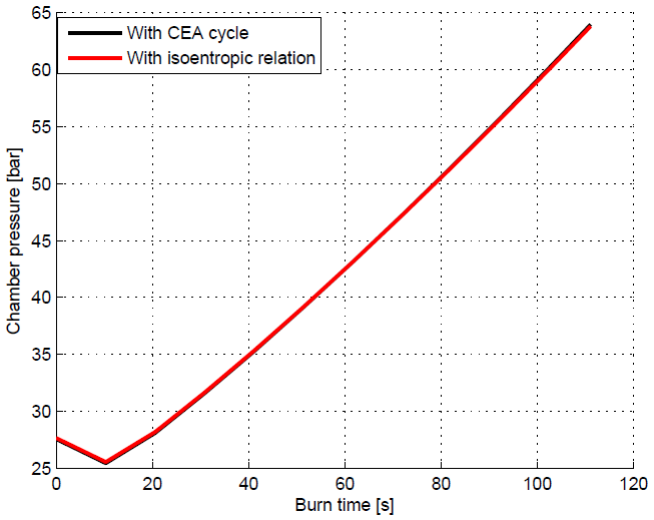


Figure 84: Chamber pressure profile over normalized time for the SP grain used for calibration (max pressure = 64 bar)

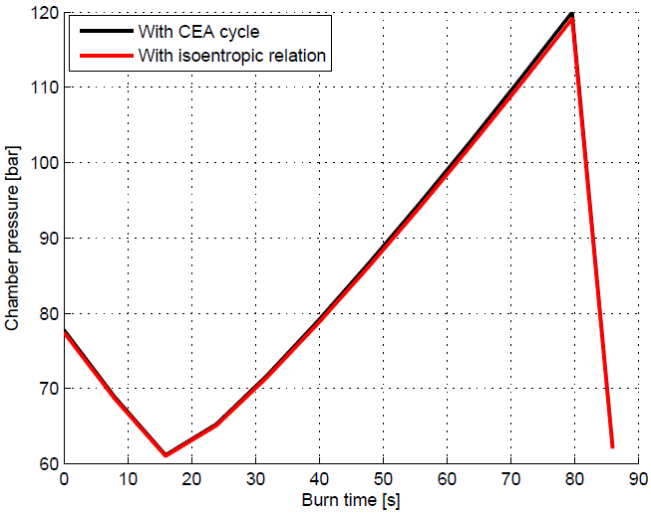


Figure 85: Chamber pressure profile over normalized time for the SP grain used for verification (max pressure = 120 bar)

Since the burn area of internal-burning SP motors is not considered constant, internal pressure, mass flow, thrust and specific impulse vary at each time instant. Two different procedures were implemented to determine these profiles: the first, more accurate, involves running CEA analysis loops for each data point to determine $p_{cc}(t)$ and $I_{sp}(t)$ for the given $A_b(t)/A_t$, coupled with mass flow balance equations to obtain $m(t)$ and $T(t)$. The second, much faster, substitutes the CEA loop with the isentropic analytical relations, assuming constant combustion properties (T_{cc}, γ_{gas}, R_{gas}) and frozen nozzle expansion. Although the CEA iterative loop is short (usually 2 or 3 iterations are necessary with a tolerance of 0.1 bar) and the number of burn points can be kept as low as 5, this process still requires repeated calls of CEA, resulting in 3 to 6 seconds of execution time. The approximate method was therefore chosen to be used by default within the MDA, since it requires negligible computational times at the expense of a very small accuracy error. The combustion properties for this method were calibrated on a star grain with chamber pressure p_{cc}=*64 bars*. Using R_{gas}=*303.4 J/(kg·K)*, γ=*1.142* and T_{cc}=*3250 K*, the error for $p_{cc}(t)$ profile with respect to the full CEA computation is very low, as shown in **Figure 84**. Also when using a completely different $A_{b(t)}$ profile and maximum p_{cc}=*120 bar* with respect to the calibration case, the matching is good enough for early preliminary design purposes, as highlighted by **Figure 85**, fully justifying skipping the CEA loop.

The approximate method does not include the analytical computation of the variation of specific impulse. Instead, a constant scaling for the specific impulse is assumed with the chamber pressure as follows:

$1 - I_{sp} / I_{sp,ref} = K \cdot \left(1 - p_{cc} / p_{cc,ref} \right)$ with ***K=0.0055*** again calibrated on the 64 bars case above.

6.2.2.7. Burn time and web length matching constraint:

As already mentioned, a constraint on the matching of burn time and web length needs to be imposed. In fact, the burn time can be computed from I_{sp} and T profiles as: $t_B = \frac{I_{sp,ref} \cdot g_0 \cdot M_{prop}}{T_{ref}} \cdot \left[1 \Big/ \int_0^1 \left(\frac{T(t)}{T_{ref}} \cdot \frac{I_{sp,ref}}{I_{sp}} \cdot dt_{norm} \right) \right]$.

At the same time, the chamber pressure profile $\boldsymbol{p_{cc}(t)}$ determines the regression rate $\boldsymbol{r_b(t)}$, which can be integrated over the burn time $\boldsymbol{t_b}$ to derive the total burned length. This must be equal to the total web length obtained from the geometrical analysis in order for the grain design to be consistent. Hence, iterations of all steps above to reach convergence on the web fraction would be required. However, in order to avoid such time consuming iterations, a compatibility constraint on the web length is imposed as follows: $w_{tot} = WF \cdot R_{ext} = t_B \cdot \int_0^1 (a \cdot [p_{cc}(t)]^n dt_{norm})$

Initial designs may therefore be unfeasible, but as the optimization proceeds the chamber pressure, propellant burn rate and grain parameters should be adjusted so that the final solution shows a consistent matching (within a given 5% tolerance to speed up convergence) of web length and burn time.

6.2.3. LP pressurization and feed system analyses

In the conceptual level models, the pressurization system had been modelled only for pressure-fed engines, but was not considered for turbopump-based propulsion architectures. However, the validation process highlighted how this assumption leads to neglect inert masses accounting for up to 5% of the total inert mass of the stage/booster, and fails to reflect the importance of the pressurization system on the overall stage/booster design. In particular, the pressurization system's selection and characteristics are strictly related to the tanks pressure and pumps specifications, aspects which strongly affect the total inert masses of the stage/booster. In order to add the tanks pressure $\boldsymbol{p_{tanks}}$ among the optimization variables of the early preliminary level MDO problem, the impact of the tanks pressure on the pressurization system's inert mass must be modelled, as well as its influence on the requirements of the turbopump feed system. For these reason, a more detailed *pressurization system analysis* was developed, valid for both pressure-fed and pump-fed engines, together with a *pump cavitation assessment method* linked with an historical WER for boost pumps necessary to suppress cavitation if occurring.

6.2.3.1. Pressurization systems sizing:

Three different pressurization methods are included in the model, which can be selected for both the oxidizer and the fuel tank(s) through the optimization variable ***PressSys***:

- *Evaporated propellants*: part of oxidizer and/or fuel are heated up through cooling jackets in the thrust chambers or heat exchangers at the turbines discharges, so that the resulting gases can be routed back to the tank to pressurize the LP. This system is best suited for LH_2 due to its low molecular weight and hence low mass, and has been historically used in most LH_2 applications (from Saturn V to the Shuttle's External Tank (ET), to Ariane 5's EPC and ESC-A stages). It can also be applied to LO_x and other propellants with some weight penalty (e.g. ET's LO_x tank).
- *Heated Helium*: as an alternative to using propellants, a separated source of high pressure gas can be stored and gradually provided to the LP tanks. Although nitrogen was often used in the past, helium is now the preferred choice due to its very low molecular weight. In this system, He is heated through heat exchangers at the turbines' discharge, allowing a reduction of the required He mass and enabling storage at very low cryogenic temperatures, which results in higher density and therefore lower tank's volume and mass. For instance, EPC's He gas tank is maintained through exceptional insulation at ***4 K*** throughout the whole flight, ensuring a very mass efficient system.
- *Stored Helium*: in case of smaller pressure-fed systems, it may be convenient to avoid complicating the design with the requirement of heat exchangers, and directly expand He from its tank to the LP tanks. This is the system allowing for maximum simplicity, but also requiring larger masses of both pressurization gas and tank.

Two alternative models were selected for the estimation of the required mass of pressurization gas (see **[86]** and **[87]** for derivations). The first consists in the ideal gas law applied to the final ullage conditions. This approach is reasonable for isothermal expansion of the pressurization gas, which is a sensible assumption when the burn time is rather long and the pressurization gas and liquid propellant temperatures are similar.

The resulting pressurization gas mass is:

$M_{press} = K_{mr} \cdot K_{me} \cdot \frac{p_{tank} \cdot V_{tank}}{R_{press,tank} \cdot T_{press,tank}}$ where K_{mr} and K_{me} are margins accounting for the residual pressurization gas remaining in its tank and for the evaporated propellant mass, p_{tank} and V_{tank} are the propellant tank's ullage pressure and volume, including ullage, and finally $R_{press,tank}$ and $T_{press,tank}$ are the gas constant and isothermal temperature of the pressurization gas in the propellant tank. The second method is based on the energy conservation law, considering adiabatic expansion of the pressurization gas and resulting in the equation already mentioned for the conceptual level models:

$M_{press} = K_{mr} \cdot K_{me} \cdot \frac{p_{tank} \cdot V_{tank}}{R_{press,tank} \cdot T_{press,tank}} \cdot \frac{k}{1 - p_{tank} / p_{press}}$ with the same meaning of the symbols and the addition of the pressurization gas pressure in its storage tank. Note that this method does not apply to evaporated propellants.

The two models were tested on the three different pressurization options, calibrating the margins with available data from Saturn, Shuttle, Ariane 5 and VEGA stages. As a result, the first method was selected for *stored He* and *evaporated propellants* pressurization, and the second for the *heated He* pressurization. Residual pressurization gas margin is set to $K_{mr}=1.05$ for stored He and $K_{mr}=1.2$ for heated He ($K_{mr}=1$ by definition for evaporated propellants, since there is no pressurization gas tank). The propellants evaporation margin is instead taken as $K_{me}=2.3$ for heated He and cryogenic propellants (LO_x or LH_2), $K_{me}=1.5$ for heated He and non-cryogenic propellants, $K_{me}=1.1$ for evaporated LO_x or LH_2, and $K_{me}=1$ for stored He. Finally, the pressurization gas temperature is set to the ambient temperature in all cases except for evaporated LO_x ($T_{press,tank}=260$ K) and LH_2 ($T_{press,tank}=150$ K).

The model for the estimation of the He tank's mass is reused from V1. However, different gas densities of He are used depending on the pressurization type and different materials characteristics are applied, in order to better match the available data for He tanks masses of Ariane 5's EPC, EPS, ESC-A and of Vega's AVUM. Specifically, the density of He is obtained from the ideal gas law, with the pressure p_{press} taken from the optimization variables vector and the temperature set to the ambient temperature (298.15 K). A limit value of 124.6 kg/m^3 for cryogenic liquid helium is defined, and employed in case of heated He pressurization. Different materials are assumed to be used in case of ambient temperature He (for stored gas systems) and cryogenic He (for heated gas systems). For ambient temperature, titanium liner with over-wrapped carbon composite is considered, assuming 1.5 mm of Ti for leak insurance and the composite alone supporting the pressure loads, as suggested in a paper by ATK (**[124]**). For cryogenic He instead, Al 7075 integral tanks are assumed, with a very large margin needed for insulation (k=2.8 calibrated on EPC's SSHeL and reasonably matching also Saturn's data for He bottles) and an additional 5 mm minimum gage.

6.2.3.2. Pump cavitation analysis and boost turbomachinery sizing:

As mentioned above, turbomachinery design in pump-fed engines is strictly connected with the pressurization system and the tanks and chamber pressure requirements. A compete engine cycle's analysis and turbomachinery design is still out of the scope of the early preliminary MDA. In fact, although it is possible to assess with relatively simple models all of the most relevant turbopump and gas generator parameters, it is then not straightforward to determine the related inert masses, which would be the main objective of such analysis. In fact, not enough data regarding the detailed WBS of propulsion systems are available to allow for reasonable correlations of the turbomachinery's weight with respect to their basic design parameters (mass flow, pressure head, dimensions). Turbopump systems information is available from a NASA publication dating back to 1974 (**[125]**), allowing to develop only a system level regression of the total turbomachinery's weight as a function of the engine's thrust, described below. Besides, this type of component-level engine's mass estimation would need to be complemented by models of similar accuracy for the mass of the thrust chamber, lines and valves, and TVC, which are not presently available. Such level of detail is for these reasons left for a future modelling step.

However, tanks pressure, chamber pressure and engine's cycle represent optimization variables in the MDO problem of the upgraded models, which strongly affect the inert mass of the engine, the propellant tanks and the pressurization system. Hence, it is necessary to introduce a model being able to verify the cavitation of the oxidizer and fuel pumps, otherwise optimization would simply push the propellants tanks pressure to the lower bound, in order to minimize the tanks' mass. A simple procedure was developed for this purpose, combining analytical and empirical relations from **[86]**, **[87]** and in particular **[125]**. In case pumps cavitation is detected, the implementation of a propulsion system architecture including boost turbopumps is assumed necessary, and an additional turbomachinery mass obtained through the above mentioned historical regression is summed to the main engine's mass. **Figure 86** shows the main TurboPump Assembly (TPA)'s mass (main oxidizer + fuel turbopumps, excluding boost turbopumps), correlating well with the engine's vacuum thrust according to the quadratic fit: $M_{TPA,main} = -8.734e-006 \cdot T_{vac}^2 + 0.2694 \cdot T_{vac} + 5.389$. From this regression, the boost TPA mass is obtained as: $M_{TPA,boost} = k_{TPA,ratio} \cdot M_{TPA,main}$ where a constant ratio $k_{TPA,ratio}=0.4127$ is taken from the SSME.

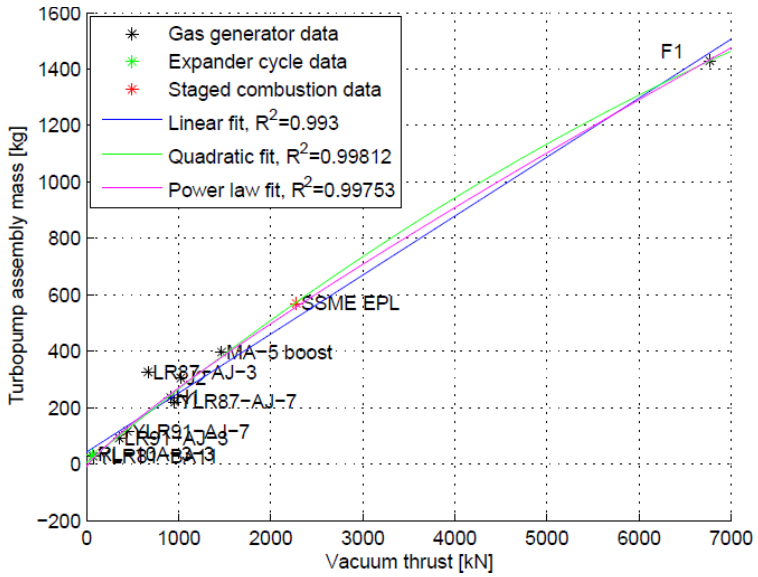


Figure 86: Historical regressions of the main turbopump assembly's mass (oxidizer + fuel turbopumps) versus engine's vacuum thrust from data extracted from [125].

The methodology for the pumps cavitation analysis can be summarized in 6 steps, repeated for both the oxidizer and fuel:

1. Compute friction losses from tank to pump as: $\Delta p_{friction} = f \cdot 0.5 \cdot \rho \cdot V^2 \cdot L_{lines} / D_{lines}$, where Darcy's turbulent friction factor $\boldsymbol{f}$ is obtained by solving Colebrook's equation under several assumptions related to the propellant lines.
2. Sum the head rise due to tank's elevation to obtain the suction pressure as: $p_{suction} = P_{tank} + g_0 \cdot \rho \cdot L_{lines} - \Delta p_{friction}$.
3. Compute the Net Pump Suction Head (NPSH), defined as $NPSH = \left(p_{suction} - p_{vapor} \right) / \left(\rho \cdot g_0 \right)$.
4. Compute the pump's discharge pressure, which is obtained from the chamber pressure as $p_{discharge} = k_{flow} \cdot p_{cc}$, where different loss coefficients $\boldsymbol{k_{flow}}$ are assumed depending on the engine's cycle and the regenerative vs. ablative cooling of the thrust chamber, which determine the path of oxidizer and fuel from pump outlet to chamber pressure. In particular:
   $\boldsymbol{k_{flow}=1.33}$ for the oxidizer in GG engines and the fuel in GG engines with ablative cooling (from Vulcain data)
   $\boldsymbol{k_{flow}=1.53}$ for the fuel in GG engines with regenerative cooling (from Vulcain engine data).
   $\boldsymbol{k_{flow}=1.49}$ for the oxidizer in EC engines (RL-10).
   $\boldsymbol{k_{flow}=2.47}$ for the fuel in EC engines (RL-10).
   $\boldsymbol{k_{flow}=1.59}$ for the oxidizer in SC engines (SSME).
   $\boldsymbol{k_{flow}=2.16}$ for the fuel in SC engines with regenerative cooling (SSME).
   $\boldsymbol{k_{flow}=1.88}$ for the fuel in SC engines with ablative cooling (adapted from SSME and Vulcain data).
5. Compute the Pump Head Rise (PHR), defined as: $PHR = \left(p_{discharge} - p_{suction} \right) / \left(\rho \cdot g_0 \right)$
6. Fixing the pump's specific speed $\boldsymbol{N_s=450\ rpm}$ for oxidizers and $\boldsymbol{N_s=400}$ for fuels (calibrated from data available on Space Shuttle, Saturn V and Ariane 5) and the pump's suction specific speed $\boldsymbol{S=20000}$ from [R56], cavitation can be verified using the following formulas (note that all values must be in British units [ft, ft^3/s, rpm]):

$$\begin{cases} N = \dfrac{N_s \cdot PHR^{0.75}}{21.2 * sqrt(\dot{V})} \\ RPSH = \left(\dfrac{21.2 \cdot N \cdot sqrt(\dot{V})}{S} \right)^{4/3} \end{cases}$$

where RPSH is the Required Pump Suction Head and $\dot{V}$ is the volumetric flow of oxidizer or fuel. The pump is assumed to cavitate when the RPSH exceeds the Cavitation Margin $\boldsymbol{CM=0.8}$ corresponding to ensuring a net suction head 20% higher than that causing cavitation.

6.2.4. New modelling functionalities

In addition to the improved grain and pressurization/feed analyses, several minor functionalities were added to the Propulsion disciplinary models, which are briefly described in this paragraph.

6.2.4.1. Multiple LP thrust chambers

Multiple thrust chambers design was introduced, with selection of the number of thrust chambers through the optimization variable N_{tc}, which is allowed to vary by default from 1 to 5. Multiple chambers design, typical of Russian engines, has the purpose of sensibly reducing the engine's length and thus the structural mass of the interstage, at the price of a higher engine mass for equal thrust. In the developed models, the thrust chamber's diameter and length are directly obtained from the required exhaust area divided by the number of chambers. A margin of 10% on the chamber's diameter is assumed to be required between the chambers to avoid interference during operations, and this allows deriving the overall engine's diameter. Note that all engine's parameters are still computed and saved per engine, except for the nozzle length and diameter which correspond to the single thrust chamber. Finally, the engine's mass obtained with the WERs is multiplied by a corrective factor K_{mc} which reflects the higher inert mass in case of multiple chamber engines, calibrated on available multiple chamber Russian engines data (RD-107A, RD-108A, RD-171M, RD-180).

6.2.4.2. Extendable LP nozzles

Again to take advantage of the mass savings related to the shortening of the interstage structures, extendable nozzles option was introduced, particularly useful for large expansion ratio vacuum LREs. This approach has recently been followed for upper stage engines in the US (Rocketdyne's RL-10-B), Russia (KBKhA's RD-0146) and Europe (Snecma's Vinci). As for the number of thrust chambers, the nozzle extension is selected through an additional boolean optimization variable, $Nozzle_{ext}$. For extendable nozzles, the following modifications apply for length, expansion ratio and dry mass:

$$\begin{cases} L_{eng,ext} = L_{eng,stowed} \cdot K_{ext,L} & with \quad K_{ext,L} = 1.828 \\ \epsilon_{ext} = \epsilon \cdot K_{ext,\epsilon} & with \quad K_{ext,\epsilon} = 3.294 \\ M_{eng,ext} = M_{eng} \cdot K_{ext,M} & with \quad K_{ext,M} = 1.797 \end{cases}$$

where $K_{ext,L}$ and $K_{ext,\varepsilon}$ are taken as average of the values of Vinci, RL-10-B and RD-0146, whereas $K_{ext,M}$ is the ratio of the masses of RL-10-A and RL-10-B models, which share the same design without and with the extendable nozzle section.

6.2.4.3. Submerged SP nozzles

As for LP systems, it is often advantageous to reduce the overall length of SRMs, in this case by submerging part of the nozzle in the solid grain, at the cost of a small performance loss. This design option can again be activated by a boolean optimization variable, $Nozzle_{sub}$, and results in length saving and I_{sp} losses equal to:

$$\begin{cases} L_{eng,sub} = L_{eng,total} \cdot K_{sub,L} & with \quad K_{sub,L} = 0.284 \\ I_{sp,sub} = I_{sp} \cdot K_{sub,I_{sp}} & with \quad K_{sub,I_{sp}} = 0.9975 \end{cases}$$

where $K_{sub,L}$ is taken as average of P241, P80, Z23 and Z9 motors and $K_{sub,Isp}$ from the Shuttle's RSRM (Sutton, **[86]**).

6.2.4.4. Detailed propellants definition

In order to allow for the additional analyses of the new models, several physical properties of the propellants need to be defined. In particular, temperature, density, vapour pressure and dynamic viscosity of LP in the tanks, as well as the minimum and maximum allowed mixture ratios and the chamber inlet temperatures are stored in V2 database, with all assumed values shown in **Table 31**. Similarly, default values for the grain density, flame temperature, gas constant, molar mass, burn rate exponent, burn rate constant, and Aluminium content are provided for SP systems. Moreover, an integer optimization variable ***Fuel*** was added for automatic selection of the propellant combination. For cryogenic systems, LO_x-LH_2 is still the only available option, but the choice between Kerosene and RP1, MMH and UDMH, or Avio and Thiokol SP formulations can be optimized. Kerosene/RP1 data are taken from a comparative study published in 1995 (**[96]**), which highlighted three main differences: Kerosene is ***3%*** denser than RP1, has lower heat of combustion (***-0.4%***) and higher heat capacity, which causes a lower temperature increase across cooling jackets. MMH shows a slightly higher density and theoretical performance with respect to UDMH, whereas the two SP formulations differ in Al percentage and type of binder.

Propellant	μ [10^{-4} kg/(ms)]	p_{vapour} [bar]	ρ_{tank} [kg/m^3]	T_{tank} [K]	$T_{cc,in}$ [K]
LO_x	1.906	1.013	90.04	1140	300
LH_2	0.143	1.013	20.43	71	91
RP1	24.341	0.002	298.15	806	300
Kerosene	33.363	0.002	298.15	832	250
N_2O_4	4.392	0.765	298.15	1447	300
MMH	10.273	0.011	298.15	878.8	300
UDMH	5.805	0.126	298.15	793	300

Table 31: Assumed values of the physical properties of LP. RP1/Kerosene data are from [96], all others from [86] and [87].

6.2.5. Enhanced specific impulse losses estimation

In light of the larger database of SRMs collected for the second modelling step, the models for the evaluation of the specific impulse losses for SP systems were re-evaluated, with the purpose of further increasing the accuracy. In particular, a significant effect of the nozzle's expansion ratio on the losses was identified, as clear from **Figure 87**. This effect is well represented by the linear fit: $I_{sp,\varepsilon} = I_{sp,CEA} \cdot (1.01275 - 0.001063 \cdot \epsilon)$. $\boldsymbol{I_{sp,\varepsilon}}$ is not the final corrected specific impulse $\boldsymbol{I_{sp}}$, since the other corrections related to the physical effects described in **Section 4.2** need to be applied. Nevertheless, the ratio $\boldsymbol{I_{sp,\varepsilon}/I_{sp}}$ is no longer dependent on the expansion ratio, hence the applications of such other corrections allows to obtain very good matching of the real specific impulse for all the tested SP engines.

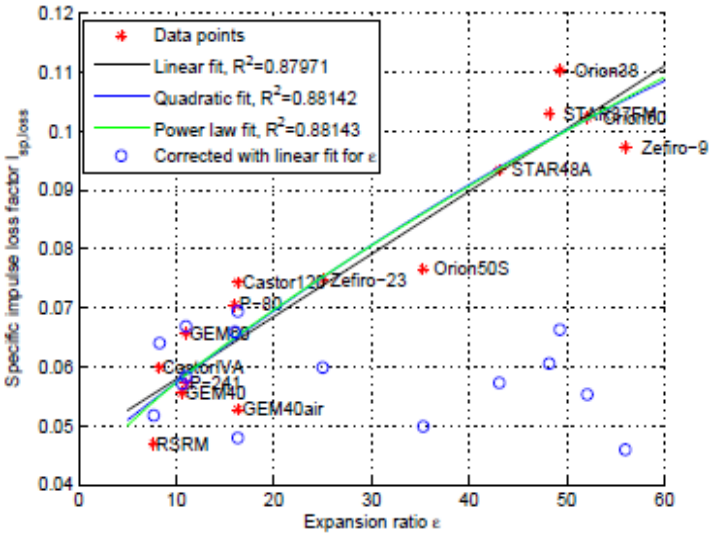


Figure 87: Linear correlation of specific impulse vs. expansion ratio for Avio and ATK SP motors.

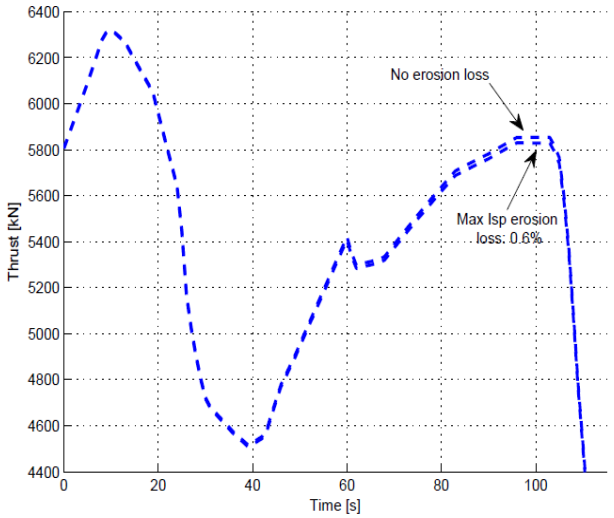


Figure 88: Thrust profile with and without nozzle erosion effect (max loss of 0.6%) for Ariane 5's P241.

Another phenomenon related to SP engines' specific impulse, which was modelled with the aim of partially correcting the overestimation of the payload performance, is the decrease in I_{sp} during the burn of SRMs due to progressive nozzle's throat erosion. From available data on VEGA motors, this effect can be approximated very well with a linear degradation from the beginning to the end of the burn. The value for the final loss can be provided by the user, or a default value of 0.6% is assumed, obtained from the Shuttle's RSRM: 1.07 inches of maximum erosion at the throat's wall is reported in **[86]**, resulting in an expansion ratio modification from 7.72 to 7.14 and therefore in a CEA theoretical I_{sp} reduction from 269.2 s to 267.5 s. As an example, **Figure 88** shows the loss of thrust due to throat erosion for P241 booster. Thrust variations include effects of varying mass flow, chamber pressure, external pressure and nozzle throat. This latter contribute is definitely less relevant than the others, but still slightly significant for an early preliminary design environment.

6.2.6. Enhanced inert masses estimation

The final area of Propulsion for which improvements were deemed necessary following the critical analysis of the validation results in the previous chapter is the estimation of the inert masses. Besides what already described above (SP sliver fraction estimation, pressurization and feed systems sizing, correction factors for multiple chambers and extendable nozzles), three more enhancements are worth mentioning: a more detailed definition of the unused LP, a more accurate estimation of the SP nozzles mass, and the updated calibration of the WERs introduced for V1.

6.2.6.1. Unused propellant masses

Validation of the conceptual models evinced an optimistic evaluation of the unused LP mass, especially for upper stages. A more refined model was thus developed, defining a more detailed propellant budget composed of the following elements:

- *Trapped LP*: intended as the propellant remaining attached to the walls tanks or trapped in the engine, lines and valves after burn-out. This is modelled as a linearly varying percentage of the total usable LP mass $\boldsymbol{M_{prop}}$ as follows: [0.8-0.4]% for $\boldsymbol{M_{prop}}$***=[0-500] tons*** in pump-fed systems, [0.7-0.3]% for $\boldsymbol{M_{prop}}$***=[0-10] tons*** for pressure-fed systems.
- *Mixture ratio unbalance*: fuel left in the tanks due to inaccuracies in mixture ratio control is modelled assuming a constant error on $\boldsymbol{\alpha_P}$ with respect to the reference value. This error is fixed to ***0.89%***, resulting in low fuel bias (e.g. 0.14% of the fuel mass for α_P=5.5), justified by assuming the presence of a propellant utilization system, which

continuously measures the remaining mass of oxidizer and fuel and adjusts α_P to minimize residuals. The numerical figure comes from the specifications of Centaur upper stage (**[126]** reports only 20 kg of fuel bias for M_{prop}=13.6 t).

- *Cryogenics boil-off*: only for restartable cryogenic upper stages, a realistic ***30 kg/hour*** evaporation of LH_2 is assumed. The upper stage coast time can be either provided by the user or is assumed to be half of the orbital period of the final target orbit. Note that boil-off on launch pad is not taken into account, since it is not considered part of the GTOW. This only results in a larger volume requirement, assumed to be covered by the tanks ullage margin.
- *Reserves and contingency*: further unused LP mass is allocated for uncertainties in loading or other contingencies, which is taken equal to ***0.5%*** of M_{prop} for both lower stages and upper stages. Moreover, an additional ***1.5%*** reserve propellant is added for upper stages, which may be used to ensure a correct payload orbital insertion.
- *End of Life (EoL) propellant*: only for restartable upper stages, propellant is allocated also for the implementation of an EoL strategy aimed at avoiding the pollution of useful orbital slots. A fixed upper stage structural ratio is considered for determining the remaining dry mass at the EoL burn, which is assumed to be impulsive. For non restartable upper stages, it is assumed that no EoL burn is required and other passivation measures are taken, having no effect on the propellant budget. The required EoL ΔV is instead computed according to the strategies defined in the Inter-Agency Space Debris Coordination Committee (IADC)'s Space Debris Mitigation Guidelines (**[127]**):

  a) For H_p***>2000 km***: assume an EoL ΔV sufficient for the transfer to a graveyard orbit with an altitude difference of $\Delta H = (235 + 1000 * C_R * A / m)$, as defined in **[127]** for GEO but extended here to other circular and elliptical MEOs.

  b) For H_p***>2000 km*** and $M_{inert,US}$***<1000 kg***: uncontrolled de-orbit is allowed by the guidelines, ensuring that the orbital lifetime is lower than ***25 years***. No detailed calculation of the orbital lifetime is performed, but it is assumed that if H_p<600 km, no EoL burn is necessary. If H_p>600 km, a small ΔV is introduced in the propellant budget, to reduce H_p down to [500-300] km for H_a=[600-2000] km.

  c) For H_p***>2000 km*** and $M_{inert,US}$***>1000 kg***: controlled de-orbit capability should be provided, since with uncontrolled re-entry possible impacts within inhabited regions cannot be excluded. For controlled de-orbit, the pericentre is assumed to be lowered down to ***60 km***, sometimes resulting in non negligible ΔVs.

  Figure 89 shows the EoL burn's ΔV for transfer to graveyard orbit, uncontrolled re-entry and controlled re-entry, with worst case ΔVs up to 400 m/s. Although these measures are currently suggested for disposal of orbital stages, they may be inapplicable in some specific cases, hence the user is allowed to enforce a different strategy if necessary.

Note that not much data is available for comparison of the total unused LP mass of actual vehicles, but the analysis with the above models shows reasonable values for Ariane 5 and VEGA liquid stages. For instance, ***2228 kg***, ***423 kg*** and ***26 kg*** of M_{unused} are obtained for EPC, ESC-A and AVUM, corresponding respectively to ***1.3%***, ***3.0%*** and ***4.7%*** of M_{prop}.

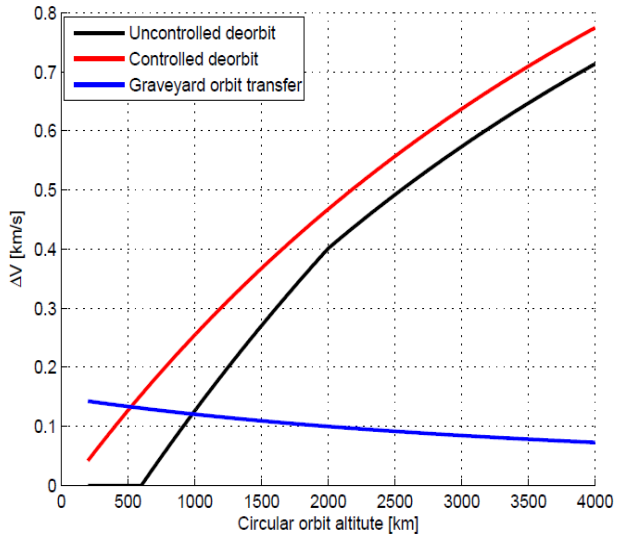


Figure 89: EoL ΔV for transfer to graveyard orbit, uncontrolled and controlled deorbit, with assumption of impulsive manoeuvres and null initial eccentricity.

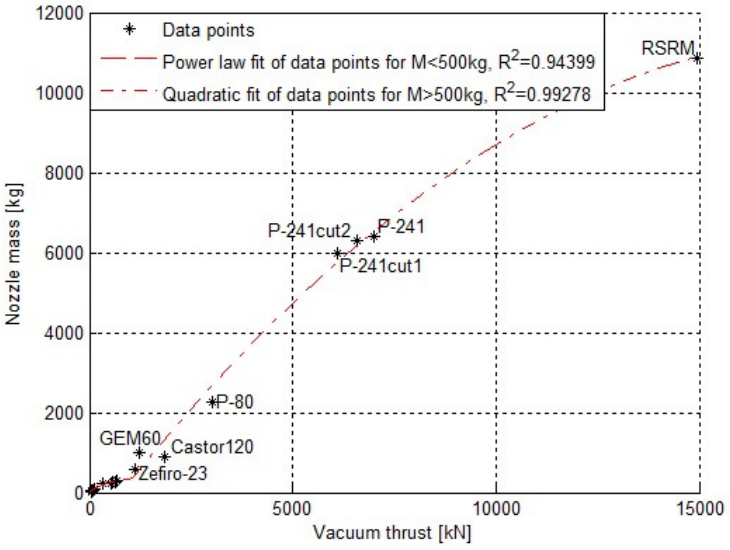


Figure 90: Correlation of SP nozzle mass vs. engine thrust for European and ATK SRMs, showing two different thrust ranges and related regressions for small motors and larger boost applications.

6.2.6.2. SRMs' nozzle mass estimation

In light of the availability of more data for both Avio and ATK engines, new regressions were developed for the inert mass of the SRM's nozzle, in order to improve the accuracy which resulted inadequate for the conceptual models, especially

with respect to the new technologies and materials used in VEGA stages. Three models were considered for the upgraded version:

- $M_{nozzle} = f$***(nozzle surface density, nozzle area)***, with nozzle surface density from a linear fit of the expansion ratio.
- $M_{nozzle} = f$***(expansion ratio, vacuum thrust)***, through a bi-dimensional linear fit.
- $M_{nozzle} = f$***(vacuum thrust)***, through the application of 1D linear, quadratic and power law fits.

In spite of its simplicity, the third model was found to be the most accurate, selecting a different fit for high thrust and low thrust engines as shown in **Figure 90**. A power law fit is therefore implemented for nozzle's inert mass if the vacuum thrust is below ***1066 kN***, whereas a quadratic fit is used otherwise. Upper and lower limits for T_{vac} are set to 30 kN and 15 MN, further pushing the bounds of the smallest and largest motors in the database (Orion 38 and Shuttle's RSRM).

6.2.6.3. LREs' updated WERs calibration

The LP engines inert mass WERs need to be adapted in light of the newly introduced modelling options. In fact, the engine's inert mass and total propulsion system mass are obtained in V2 models as:

$$M_{PropSys} = M_{EA} + M_{ununsed} + M_{PressSys} \qquad with\ M_{EA} = \begin{cases} N_{eng} \cdot M_{eng} & for\ LP\ systems \\ M_{nozzle} + M_{TVC} + M_{igniter} & for\ SP\ systems \end{cases}$$

$$and\ M_{eng} = M_{eng,WER} \cdot (1 + K_{restart} + K_{throttle}) \cdot K_{multipleChambers} \cdot K_{nozzleExtension} + M_{boostPumps} + M_{TVC}$$

where $M_{eng,WER}$ is obtained with the same WER approach described for V1. However, the introduction of the correction factors presented in the previous sections would result in large errors on the total engine inert mass if the same regression coefficients were used. For this reason, the WERs detailed in **Table 10** were updated, with the exception of those for pressure-fed engines, which are not affected by the model enhancements. The new regressions are not reported here, but generally determine an improvement in the overall accuracy of the inert engine's mass estimation, introducing information on the details of the engine's design.

6.3. Geometry

The basic approach to the geometric analysis of ELVs, including external geometry build-up, interference constraints verification and LaWGS file generation for 2D and 3D visualization is maintained unchanged. The analysis flow is the same as represented in Figure 32, with the addition of three new functionalities shown to be necessary to improve the accuracy in the estimation of the lengths, which are critical being inputs of the weights assessment. Specifically, these are two new tank lay-outs (multiple tanks as in AVUM and enclosed tanks as in ESC-A), separation plane position definition, and under-fairing upper stage configuration.

6.3.1. Multiple and enclosed tanks modelling

The common bulkhead and separate bulkhead tanks lay-outs already introduced in the conceptual level models are not suitable to small stages, for which the length of the cylindrical section may result to be very small or even negative, due to the small volume of propellants with respect to the stage's diameter. The design of several upper stages around the world confirms this consideration, showing multiple tanks configurations (e.g. AVUM and EPS in Europe, Fregat, Breeze and Zenit in Russia) or enclosed tanks such as in ESC-A and ESC-B cryogenic upper stages. Taking into account these lay-outs, the optimization variable ***TT*** is used to define the tanks configuration as follows:

- ***TT=1***: separate bulkhead tanks, as in V1 (e.g. EPC cryogenic tank).
- ***TT=2***: common bulkhead tanks, as in V1 (e.g. Space Shuttle's ET).
- ***TT=3***: enclosed oxidizer and fuel tanks reflecting ESC-A's design, with the smaller of the two tanks assumed to be a cylinder of fixed L/D ratio and a larger hollow cylinder tank enclosing it on top. **Figure 91** reports the geometry of ESC-A's tanks, which are used as reference to define all proportions for the enclosed tanks geometric model.
- ***TT=4/5***: multiple 2+2 or 3+3 spherical tanks. If the required volume is too large for 2+2/3+3 tanks, a second floor of tanks is added, resulting in 4+4/6+6 tanks if needed.

6.3.2. Separation plane definition

With the introduction of an analytical structural model (see **Section 6.5**), which aims at the physics-based sizing of all the structural components of ELVs, the definition of the separation plane of each stage becomes necessary, allowing to assess the mass staying with each of the stages after separation. Allocating a few hundred kg to one or the other corresponds in fact to non negligible variations in the launcher's payload performance. The separation plane $\beta_{sep,j}$ is therefore defined for all stages ***j*** except the first as the longitudinal section above which the structure remains attached to j-th stage after jettison. In case of boosters and first stages instead, β_{sep} is intended as the end of the lower aerodynamic skirt, which may also

constitute the structural interface with the launch pad if the booster/core is connected to the ground (for example, Ariane 5 is connected to the pad through the boosters' aft skirt, whereas the core is unsupported).

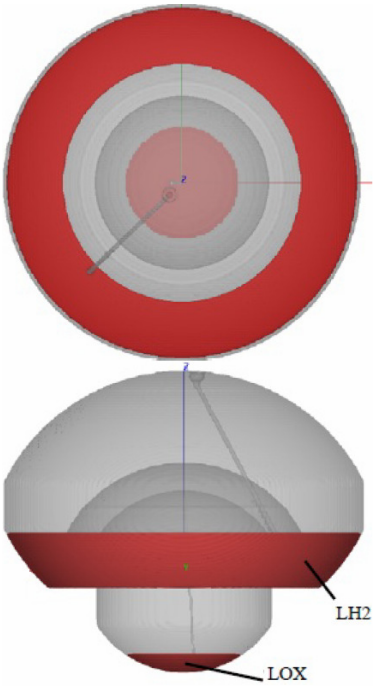


Figure 91: ESC-A cryogenic upper stage's enclosed tanks geometry, as taken from [128].

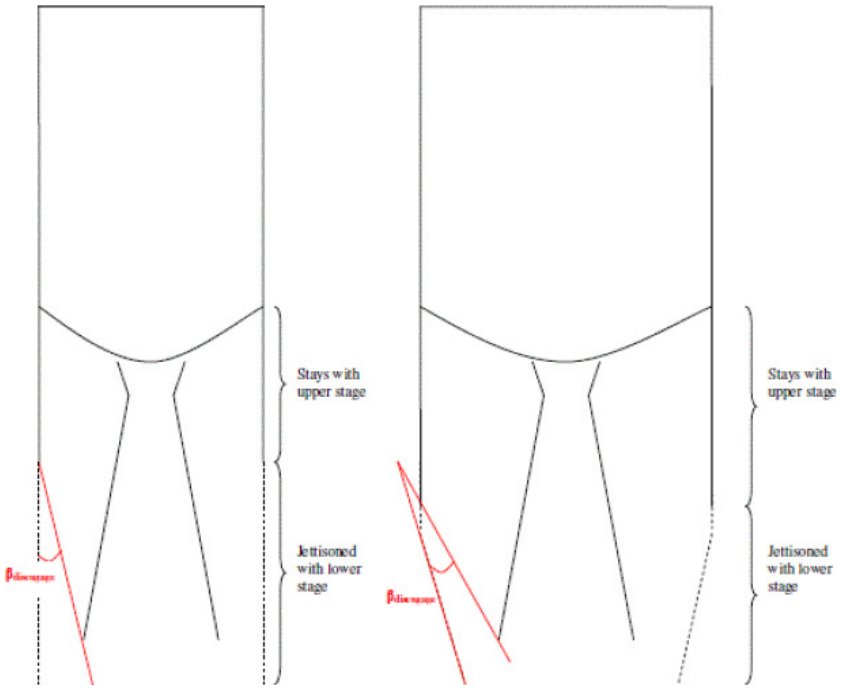


Figure 92: Skirt disengagement model: separation plane is defined by imposing $\beta_{sep}<15\ deg$ to ensure safe jettison while minimizing mass remaining on the stage above.

For all stages except the first, the separation plane is usually placed as high as possible, in order to jettison a larger mass and therefore reduce the inert mass carried along during the next stage's flight. However, safety concerns related to possible impacts between the jettisoned interstage and the engine's nozzle constrain the position of the separation plane. In particular, this is taken into account by defining β_{sep} as the highest plane which allows for a disengagement angle of at least ***15 deg***. The disengagement angle is shown in **Figure 92** for a configuration with constant and different diameter of the two stages. If this constraint results in a separation plane placed above the engine, the separation is assumed to occur exactly at the engine's top. For first stage and boosters instead, the separation plane is located at the top of the engine, assuming that the aerodynamic skirt is maintained as short as possible. If however the skirt also has structural support function (i.e. connection with the ground), this pad interface skirt is assumed to have a length equal to $L_{PI}=0.264 \cdot D$, where the numerical figure is an average of P-241 and P-80 aft support structures. Note that an optimization variable PI_{type} is also included in the system level design variables vector X_{SDV}, defining the location of the pad interface structure in case of parallel configurations, either on the boosters or on the core, having a relevant impact on the structural mass as well as launcher controllability.

6.3.3. Under-fairing upper stage configuration

The validation process highlighted significant errors on the stage's length in case of under-fairing configurations, such as VEGA's AVUM. This type of design generally reduces the inert mass of small upper stages by eliminating its external structure, substituted by the payload fairing. However, different geometry and weight models need to be defined in these cases to obtain reasonable inert mass estimates. Therefore, an under-fairing upper stage model was added to the upgraded Geometry discipline, introducing two optimization variables: a boolean variable ***UFC*** for the activation of the under-fairing option and a continuous variable K_{UF} which defines the percentage of the upper stage's length covered by the fairing (may be up to 100%).

The external geometry of the under-fairing configuration is similar to that of the classical configuration. There are however few differences, also depending on the diameter continuity between upper stage and fairing, generating four different cases as schematically shown in **Figure 93**. In case ***UFC=true***, the separation plane with the lower stage may result to be above or below the bottom of the fairing, depending on the value of K_{UF}. If it is below, the external structure above the separation plane up to the bottom of the fairing is assumed to be part of the upper stage (this is the case of the AVUM, which has a lateral cylindrical structure of 465 mm of height). Otherwise the upper stage is assumed to have no

external structure at all, and to be covered in the upper part by the fairing and in the lower part by the interstage structure with the lower stage. In this latter case, the separation planes fairing-upper stage and upper stage-lower stage coincide.

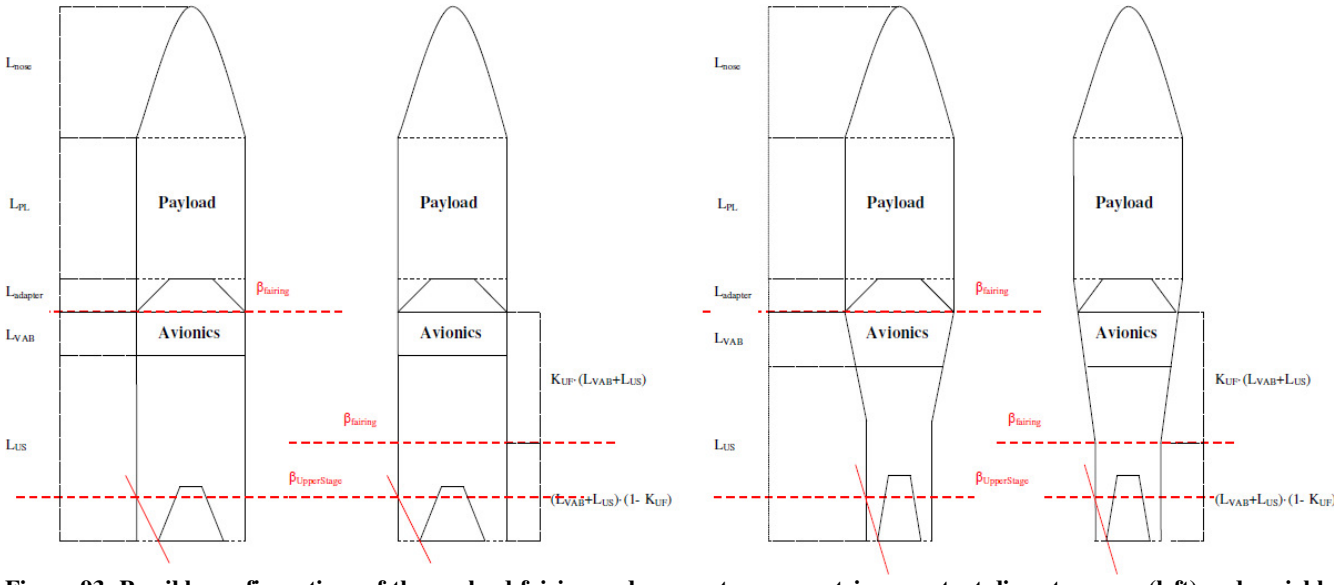


Figure 93: Possible configurations of the payload fairing and upper stage geometries: constant diameter cases (left) and variable diameter cases (right), both without and with under fairing option.

6.4. Trajectory, guidance and control

Although the EoMs integration, the related environmental models and guidance strategies are kept identical from the conceptual models, several enhancements were developed both to improve the accuracy of the performance assessment through trajectory optimization, which was shown to overestimate the payload mass by ~10%, and to introduce new functionalities. In particolar, these upgrades can be summarized in the following points:

- Calculation of the total heat load from payload fairing jettison to payload release, integrated from the convective stagnation point heat flux at each instant, to be used as either objective function or a constraint in the optimization.
- Wind modelling, accounting for both steady-state profiles and discrete gusts, for static controllability verification under worst case wind conditions and for evaluation of the steering losses due to a given synthetic wind profile.
- Estimation of steering losses due to the compensation of aerodynamic moments and the launch vehicle's manoeuvres (atmospheric pitch-over, exoatmospheric pitch and yaw profiles).
- Simplified re-entry simulation of all suborbital stages/boosters and verification of impact ellipse's interference with populated areas, which allows to determine whether a given trajectory complies with safety guidelines.

The latter three points, involving larger model upgrades, are described in details in the next paragraphs.

6.4.1. Wind model and steering losses

Wind influences the nominal trajectory, in terms of control profiles ensuring the achievement of the correct target orbit, as well as the structural loads and control requirements, with most critical effects in the ***[6-15] km*** altitude range. Horizontal wind is often significant also for preliminary design purposes, whereas vertical wind is less important. To account for the effect of wind in the trajectory simulation, the most common approach is to define a synthetic wind profile as a function of altitude, and include it in the EoM when the aerodynamic forces need to be computed. By varying the reference altitude and the direction of the synthetic wind profile, as well as by introducing uncertainties on other launch vehicle and trajectory parameters, it is possible to analyze off-nominal trajectories and verify the capability of the control system to compensate. The dispersion of the final orbital parameters can in this way be analyzed, assessing the 3σ orbital insertion's accuracy. Such an analysis is left as a future development, considering that repeated Montecarlo runs would need to be executed – hence greatly increasing the computational effort - and that design aspects related to the navigation sensors would also need to be modelled in order to obtain reasonable figures.

Nevertheless, a synthetic wind model based on tables from NASA's Handbook on Terrestrial Environment (**[129]**) was implemented, for including the effect of wind on both the static controllability verification of the launcher and the steering losses evaluation. In particular, look-up tables were taken from **[129]** for the steady-state wind speed, build-up and back-off wind shears. These are the 95-percentile envelopes of winds over the entire year for four different launch sites in the United States (Kennedy, Vandenberg, Edwards, White Sands) and therefore represent a worst-case for most launch scenarios.

From these tables the procedure to obtain the synthetic wind profile, documented in **Section 2.3.9** of **[129]**, is based on the definition of a reference altitude – often the maximum steady state wind altitude - where the wind is assumed to match the stead-state envelope. Wind build-up and back-off profiles are then constructed from this point downwards and upwards, and finally a discrete wind gust is superimposed at the reference altitude, and continuously connected to the build-up and back-off curves. The discrete gust is assumed to be divided in three regions: build-up and back-off regions with 1-cosine or power law profile, of 30 meters of thickness each, and a 240-meters thick plateau with a constant 9 m/s wind. This process is graphically represented in **Figure 94**.

The static controllability analysis proposed for V1 is maintained unchanged, with the difference that the angle of attack provided by the guidance laws is modified due to the presence of a worst-case wind speed, as shown in **Figure 95**. This is conservatively taken as the steady-state envelope, compared with the synthetic profile in **Figure 96**, in the direction that results in the largest increase of total AoA. Moreover, the required static controllability margin was modified and set to 50%, as specified by NASA practices reported in **[130]**.

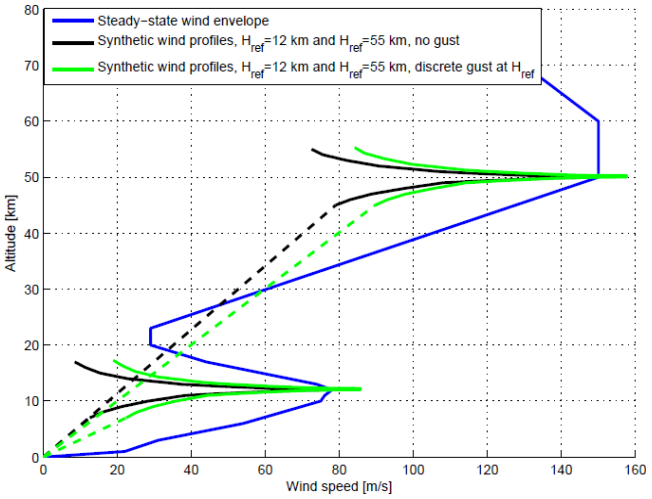


Figure 94: Synthetic wind profiles for different reference altitudes and with or without discrete gust superimposition.

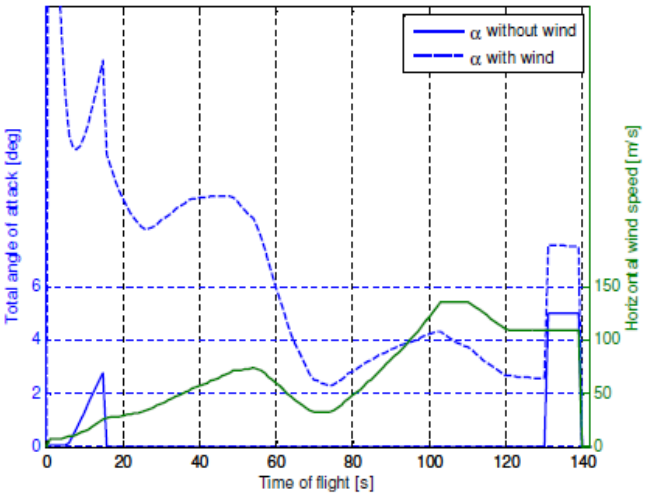


Figure 95: Total AoA for static controllability verification in no-wind and worst-case wind conditions.

One of the reasons identified for the optimistic evaluation of ELVs' performance is the lack of steering losses due to both wind and manoeuvres. These are definitely less relevant than gravity and drag losses, often by one or more orders of magnitude, but still represent a non negligible aspect if accuracies better than 10% are targeted. For this reason, approximate models for the evaluation of steering losses were introduced, based on the determination of the lateral thrust necessary to 1) balance the aerodynamic moment at each time instant and 2) allow for the required angular acceleration in case of pitch/yaw manoeuvres. As regards to the first aspect, the same models used for the static controllability analysis are employed, but the synthetic wind profile is used instead of the worst-case wind. This is because during an ascent to orbit, the wind velocity may reach the value of the steady-state envelope only at a given altitude and not throughout the whole flight. This reference altitude can be specified by the user, or is conservatively assumed to be ***12 km***, which determines a maximum wind in the [6-15] km region, where the effect of wind is largest.

As for the torque required to provide the necessary angular acceleration/deceleration during steering, the following manoeuvres are considered: a) initial pitch-over, with angular velocity defined by the pitch-over time and maximum angle of attack values (both optimization variables), and b) discontinuities in pitch and/or yaw after each jettison event, for which the angular velocity is assumed to be 0.5 deg/s. In both cases, the manoeuvre is considered to be constituted of a constant angular acceleration, with final angular velocity matching the required value, followed by a constant angular velocity phase and again a constant angular deceleration with null final velocity.

The manoeuvre's torque, applied only during the acceleration and deceleration phases, is summed to the aerodynamic balance torque to determine the total torque to be provided at each time instant by the TVC. Margins of ***1.2*** on the aerodynamic balance and ***2.0*** on the manoeuvres are added, to account for uncertainties in the models, and the required deflection angle is then computed as $sin(\delta_{TVC}(t)) = \left(M_{aerobalance}(t) + M_{manoeuvres}(t)\right) \Big/ \left(T(t) \cdot \left(X_{CoG}(t) - X_{pivot}\right)\right)$.

From the $\boldsymbol{\delta_{TVC}(t)}$, the instantaneous thrust in the axial direction is computed and applied in the equations of motion instead of the total thrust. Note that in order to compute the manoeuvres' torque, the longitudinal moment of inertia $\boldsymbol{I_{long}}$ is required. This is assumed to be the same for pitch and yaw also for non axi-symmetric launch vehicle's configurations, and is computed from the summation of the contributes of all launch vehicle components still not jettisoned. All components are

considered either thin-wall cylinders (payload fairing, payload adapter, dry stages and boosters) or solid cylinders (payload, liquid fuel, liquid oxidizer, solid grain), so that the inertias with respect to their centres are obtained as:

$$I = 1/12 \cdot m \cdot (3 \cdot r^2 + h^2) \quad for\ \ solid\ \ cylinders$$

$$I = 1/12 \cdot m \cdot (6 \cdot r^2 + h^2) \quad for\ \ thin-wall\ \ cylinders$$

These values are then shifted with respect to the overall launch vehicle's CoG using the parallel axes theorem under the same hypotheses defined for the CoG calculation in **Paragraph 4.5.6**. Note however how the contribute of the manoeuvres' torque to the total steering loss was found to be quite insignificant. For instance, **Figure 97** shows the TVC angle profile for both the P241 booster's nozzle and the Vulcain engine of EPC in a typical Ariane 5 flight to GTO. Most of the deflection is clearly due to aerodynamic moment compensation requirement, which peaks at the maximum wind instant, whereas manoeuvre-induced deflections are limited to less than 1 deg.

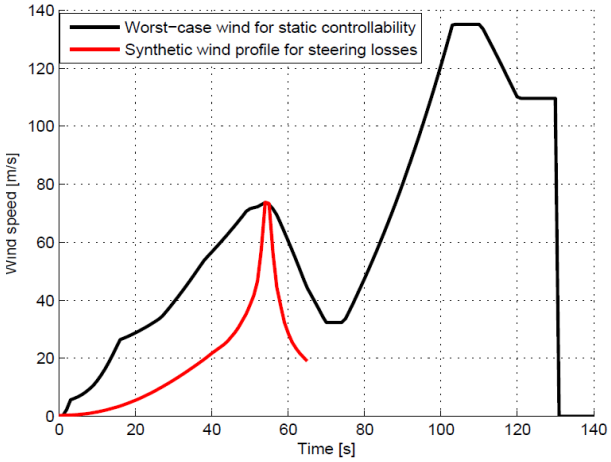


Figure 96: Worst-case and synthetic wind profiles versus time for a typical Ariane-5 ECA flight to GTO.

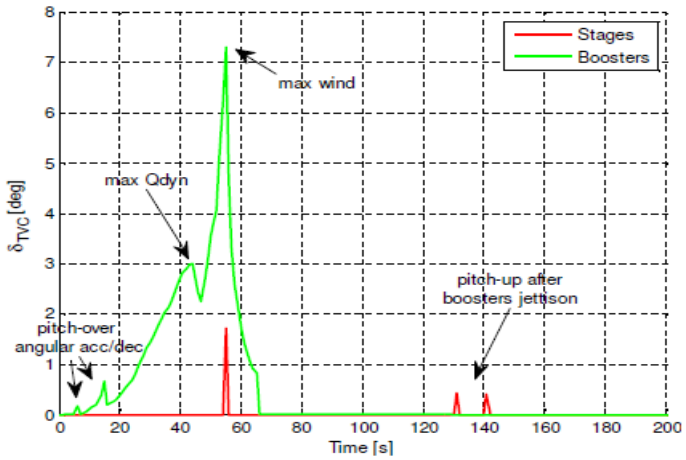


Figure 97: Required TVC angle profile for typical Ariane 5 ECA flight, accounting for both aerodynamic moments and manoeuvres.

6.4.2. Safety analysis for components re-entry

The most critical industrial requirement pointed out by ESA review is the safety analysis related to both the impact of the suborbital components of ELVs and the passivation and re-entry of orbital stages. For the latter aspect, provisions were described in the Propulsion discipline section to account for the EoL's ΔV in the propellant budget. As regards to boosters and lower stages that do not reach orbit instead, safety issues impose that the expected ground impact ellipse is not located within inhabited regions. This represents a very important constraint for many launch sites and target orbits, which was shown to sensibly affect the final performance of ELVs. To include this constraint in the model, some form of simulation of the re-entry of all suborbital components is necessary, for which two different models were conceived:

- Numerical integration of the same 3-DoF equations used for ascent, simplified by neglecting the lift force and therefore representing fully ballistic trajectories.
- Propagation of the Keplerian parameters from jettison to ground impact, complemented with empirical models developed from the 3-DoF simulations to estimate the downrange reduction due to drag.

In both cases, lift and other effects on the trajectory such as wind and off-nominal jettison conditions are neglected, due to the large uncertainties on these parameters. In order to reliably assess the impact ellipse taking into account all the possible uncertainties, a large number of Montecarlo simulations would be required, with computational times of maybe one order of magnitude larger than a full MDA. A Montecarlo set-up, including uncertainties on the jettison velocity, flight-path and heading, horizontal wind profile and direction, mass, angle of attack and sideslip, and possibly break-up analysis and statistical determinations of the fragments ballistic coefficient distribution, is therefore only left as a future development. Instead, it is considered sufficient for the purpose of safety constraints definition to assume a min-drag and max-drag ballistic coefficients, combined with a minor-to-major ellipse axes ratio, to determine the approximate impact ellipse's geometry.

Specifically, the algorithm for ground impact safety constraints was defined as follows:

1. Determine min and max drag coefficients by interpolation of a $C_D(AoA, L/D)$ database for cylinders obtained offline with Missile DATCOM. An average AoA of ***20 deg*** and ***65 deg*** for min and max drag are assumed, representative of typical attitudes in the ***[0-40] deg*** range for nose-first entries and in the ***[40-90] deg*** range for broadside entries.
2. Compute the min drag and max drag ballistic coefficients from the reference area (stage/booster cross-section), drag coefficient and jettison mass (inert mass including residuals) as: $C_b = M_{inert}/(Cd \cdot A_{ref})$.
3. Use either the 3-DoF ballistic integration or the Keplerian propagation to define the two impact points.

4. Compute the impact ellipse's minor axis from the major axis, with a guessed axes ratio of ***0.3***.
5. Overlap a population density map and obtain by 2D interpolation the population density at each of the 4 extremes of the ellipse. The 2015 population projection from the Gridded Population of the World, version 3 (GPWv3, **[131]**) is used as default density map. Population data is stored in data files with three different resolutions of 1, ½ and ¼ of a degree, or a user defined map can be provided for specific requirements.
6. Compare the highest obtained population density with the allowed maximum value, which is fixed to no persons per square km by default but can be modified by the user.

Note that a more correct constraint should be imposed on the probability of fatality along the complete flight profile of the launch vehicle (e.g. 10^{-6} as required by the standards). This approach was however not implemented because it requires the determination of population density and failure probability at each time instant along the flight, increasing the computational times and complexity of the model, with feedbacks from the reliability assessment to the trajectory. Though much simplified, the procedure described above is nevertheless sufficient to impose safety constraints which do affect the performance of a launch vehicle. Moreover, by properly modifying the density map file, drop zones in the ocean or in given areas of scarcely inhabited regions (e.g. launches from Baikonour) can also be enforced.

The dynamic model for the 3-DoF ballistic re-entry integration is taken without modifications from the ascent, by simply removing lift and thrust forces and allowing for a single integration phase from jettison to ground impact. A small ΔV of ***5 m/s*** with respect to the launch vehicle's velocity at jettison is assumed to be provided with retro-rockets, and the impact is considered for simplicity to always occur at null elevation. **Figure 98** and **Figure 99** show examples of the re-entry of the components of Ariane 5 ECA and VEGA. All stages of VEGA and Ariane's P241 boosters are in this case dropped into the Atlantic Ocean, but the EPC re-enters over Africa, making this trajectory unfeasible from safety point of view.

The dynamic model for the Keplerian propagation consists instead in the determination of the true anomaly and time a given orbit intersects the planet's surface, again assumed to be at null elevation, as follows:

$$\begin{cases} \theta_{impact} = acos\left(\frac{1}{e}\left(\frac{a\cdot(1-e^2)}{R_{planet}} - 1\right)\right) \\ t_{impact} = \sqrt{\frac{a^3}{\mu_{planet}}}\cdot(E_{impact} - e\cdot sin(E_{impact})) \quad with \quad tan(E_{impact}/2) = \sqrt{\frac{1-e}{1+e}}\cdot tan(\theta_{impact}/2) \end{cases}$$

This allows calculating through coordinate conversions the latitude δ and longitude λ of the impact point, from which the downrange distance from jettison to impact is obtained. The map deformation due to Mercator projection is also taken into account by using a scale factor equal $\pi\cdot R_{planet}\cdot sin(\pi/2-\delta)/180 \; km/deg$ on the vertical direction. An empirical evaluation of the ratio $k_{downrange} = \frac{downrange \; with \; drag}{downrange \; no \; drag}$ was performed using the ballistic 3-DoF model for Ariane 5 ECA and VEGA, showing that $\boldsymbol{k_{downrange}}$ can be reasonably approximated as a a function of $\boldsymbol{t_{impact}}$ and of the ballistic coefficient $\boldsymbol{C_b}$. Bi-dimensional fits were therefore developed, combining a power law regressions of $\boldsymbol{C_b}$ with a linear interpolation on $\boldsymbol{t_{impact}}$. The exploitation of the simplified Keplerian model allows for a 20-25% CPU time saving for each MDA, representing a sensible improvement in front of a limited loss of accuracy on the impact ellipse.

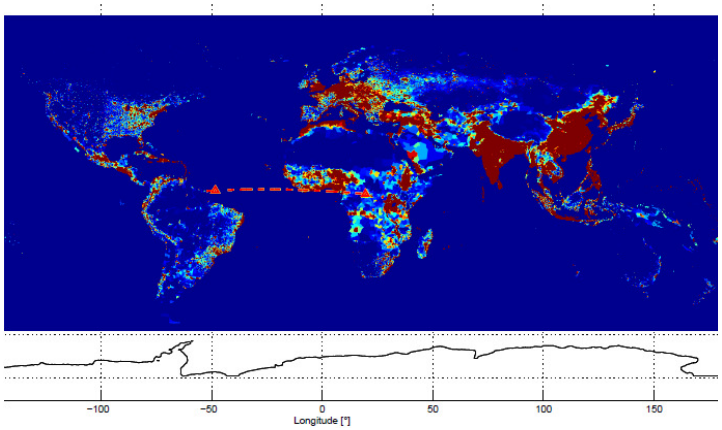


Figure 98: Example of unfeasible Ariane 5 ECA trajectory, with re-entry of ESC-A over populated areas of Africa. Red/ Yellow: max/min drag re-entry trajectories and impact points.

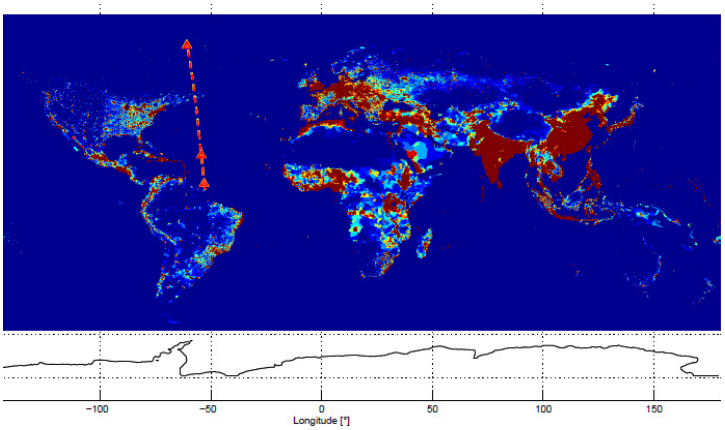


Figure 99: Example of feasible VEGA trajectory, with re-entry of all SP motors over the Atlantic Ocean. Red/Yellow: max/min drag re-entry trajectories and impact points.

6.5. Structural analysis and sizing

The structural analysis and sizing model was introduced specifically for the early preliminary models, with the purpose of sensibly improving the estimates of the inert masses of fairing, stages and boosters. The mass of several structural components can be computed within the Structures discipline, and is then added to that of the non-structural elements, assessed within the Weights discipline. In particular, SRM cases, fuel and oxidizer tanks, intertanks and interstages, forward and aft skirts, thrust frames, pad interface structures, payload adapter, payload fairing and boosters nose ogive are taken into consideration here.

Since the FEM approach was discarded due to the excessive computational times involved, an engineering-level structural analysis procedure was derived from a recent work defining a "beam approximation" for fuselage structures. Originally, this methodology was outlined by NASA in 1996 for aircrafts (**[133]**), and was afterwards adapted for the main airframe of in-line launchers at Georgia Tech in 2004 ([134] and **[135]**). The model developed here for generic ELVs extends the latter, including parallel configurations as well as a wider set of structural component types and complementing it with other classical analysis practices from **[109]** when needed. Few simplifying assumptions were also taken with respect to **[134]**, in particular considering all longitudinal cross sections as circular instead of elliptical (ELVs instead of RLVs).

The resulting structural analysis process, rather complicated, is composed of the following steps:

1. Define a set of load cases from Trajectory: on-pad, take-off, max Q_{dyn}, max $Q_{dyn} \cdot \alpha_{tot}$, max N_{aero}, max N_{ax} and N_{lat}.
2. For each load case:
   a. Define the distribution of masses along the longitudinal axis of the launcher's core and boosters sets, treated as separate beams throughout the structural analysis (up to three beams are therefore considered, one for the core and one for each of the boosters sets).
   b. Define the external aerodynamic and gravitational loads on each beam. Aerodynamic loads are distributed along the longitudinal axes through approximate assumptions.
   c. Define the transfer force in the axial and lateral direction between the core and the boosters set(s).
   d. Compute the primary external flight loads along the beam(s) from the inertial, aerodynamic and thrust loads. The three types of considered external loads are *axial force* ***P***, *shear force* ***T***, and *bending moment* ***M***.
   e. Compute the secondary external flight loads, that is ***(P,T,M)*** on secondary components such as payload adapter and thrust structures, which are not part of the main structural beam(s) carrying the primary loads.
3. Define the structural components to be sized in terms of geometry, longitudinal stations, loading conditions (*internal pressure loads*, *external primary and secondary flight loads*), and structural properties (material and stiffening).
4. For each longitudinal station of each structural component:
   a. Derive from all load cases the worst-case internal running loads: *hoop stress* $\boldsymbol{N_y}$ (circumferential), *axial compressive and tensile stresses* $\boldsymbol{N_{x,c}}$ and $\boldsymbol{N_{x,t}}$ (longitudinal), and *shear stress* $\boldsymbol{N_{xy}}$ (transverse).
   b. Estimate the required shell's and frames' thickness to withstand the worst-case running loads. For the shell, a *minimum material's gage* is imposed and three failure modes are considered: *ultimate strength*, *yield strength*, and *buckling*. To prevent general instability, the shell configurations are associated to longitudinal frames, whose smeared thickness is instead determined from *Shanley's criterion*.
5. From the calculated shell and frames thicknesses, determine the primary structural mass of each component through station-by-station integration. This process provides an estimate of the optimal structural weight, but does not account for non-optimum weights such as bulkheads, minor frames, coverings, fasteners, joints, and other issues.
6. Correlate the primary structural masses obtained through the above fundamental structural principals with existing launch vehicle weights, through linear regressions of available data. This allows taking into account the secondary structural masses, obtaining a good level of accuracy.

At the end of this procedure, the total masses of all structural components are summed to the non structural components (propulsion, avionics, ...) to determine the total inert masses of each stage, booster and payload fairing, which need to be compared with those at the previous iteration to check the convergence of the Trajectory-Structures sizing cycle. The CoG and inertia properties are at this point also updated, so that the following trajectory simulation can be executed on the basis of the up-to-date mass properties. More details on each of the above steps are given in the following paragraphs.

6.5.1. Load cases definition

The structural sizing of ELVs involves a large number of load cases, usually starting from the ground handling and continuing all the way to orbit insertion. However, for simplicity, only the following flight events are considered here:

- On-pad: dimensioning case for the pad interface structures, for which a ***2.0*** corrective factor with respect to the axial load derived from the launcher's weight is assumed, in order to account for ground winds.
- Take-off: may represent a dimensioning case in compression due to large mass of propellant still loaded in the tanks.
- Max $\boldsymbol{Q_{dyn}}$: usually corresponding to the maximum aerodynamic drag.

- Max $\boldsymbol{Q_{dyn}\cdot\alpha_{tot}}$: usually corresponding to the maximum aerodynamic bending moment.
- Max $\boldsymbol{N_{aero}}$: maximum lateral aerodynamic force, assuming the launcher is being subject to a worst-case lateral gust.
- Max $N_{ax,j}$, $N_{lat,j}$, $N_{axneg,j}$: maximum axial acceleration, lateral acceleration and negative axial acceleration, which all need to be checked for all phases j of the flight. For instance, the maximum acceleration for the second or third stage may be sensibly higher than for the first stage, or even a lower acceleration event may be more critical for a given stage with respect to the maximum acceleration instant, due to the presence of attached boosters.

In addition to these cases, loads at regular intervals of 10 s are checked along the whole trajectory, to verify that no relevant loading condition is neglected. An example of important load case identified with this method is the first thrust peak during the flights of both P241 and P80 boosters, more severe than the maximum acceleration event, which occurs close to burn-out and hence with a much smaller propellant loading. For all the $\boldsymbol{N_{lc}}$ selected load cases, all trajectory data influencing the structural sizing are stored, in particular: propellant loading and thrust of the different stages/boosters, gravity, pitch, lift, drag, CoP, CoG, AoA, wind speed, dynamic pressure, axial and lateral accelerations.

6.5.2. External forces

External aerodynamic and gravitational forces need to be computed for each load case in terms of axial and lateral components, since all loads are defined in the body reference frame. Gravitational accelerations are obtained from altitude and pitch angle as:

$$\begin{cases} g_{ax} = g(h(t))\cdot\sin(\theta) \\ g_{lat} = g(h(t))\cdot\cos(\theta) \end{cases}$$

whereas aerodynamic axial and lateral forces are derived from drag and lift as:

$$\begin{cases} N_{aero} = D\cdot\cos(\alpha_{wind}) - L\cdot\sin(\alpha_{wind}) \\ A_{aero} = L\cdot\cos(\alpha_{wind}) + D\cdot\sin(\alpha_{wind}) \end{cases}$$

Note the usage of the AoA including wind. This conservatively causes a rather large normal component (with typical wind AoA in the order of 5 deg at maximum dynamic pressure), resulting in similarly large bending moment. The direction of the lateral gravitational acceleration and aerodynamic normal force does not change the entity of the bending, and is therefore irrelevant.

With the knowledge of $\boldsymbol{g_{ax}}$, $\boldsymbol{g_{lat}}$, $\boldsymbol{A_{aero}}$, $\boldsymbol{N_{aero}}$, the balance of the external forces with the inertial accelerations recorded within the trajectory integration can be verified as follows:

$$\begin{cases} T_{tot} - A_{aero} - M\cdot g_{ax} = M\cdot n_{ax} \\ N_{aero} - M\cdot g_{lat} = M\cdot n_{lat} \end{cases}$$ where $\boldsymbol{T_{tot}}$ is the total thrust force, including stage and boosters.

Aerodynamic loads need then to be divided between the core and the booster sets, and to be distributed along the beams' longitudinal axis. $\boldsymbol{A_{aero}}$ and $\boldsymbol{N_{aero}}$ are simply split among core and booster sets proportionally to the products $\boldsymbol{C_D\cdot A_{ref}}$ and $\boldsymbol{C_L\cdot A_{ref}}$ of the different components. For each of the beams (core and boosters sets), the axial aerodynamic force is then partially applied at the tip of the launcher ($\boldsymbol{A_{aero,f}=0.7\cdot A_{aero}}$, representing all forms of drag except from friction drag), with the remaining $\boldsymbol{A_{aero,d}=0.3\cdot A_{aero}}$ uniformly distributed along the beam. The normal force is instead assumed to have a linear distribution from tip to bottom of the beam, computing the coefficients which ensure a CoP position matching that obtained from DATCOM.

Finally, the boosters and core exchange both axial forces and lateral forces during the initial ascent phases. Coherently with the structural model of Ariane 5, the axial force is assumed to be completely transferred at the FBA system, whereas the lateral forces are transferred at the ABA location (which is provided with piston dampeners). The introduction of the thrust loads of the boosters alleviates the compression loads aft of the attachment position. This is highlighted especially for large SP boosters, such as those of Ariane 5, for which the whole common bulkhead tank of the first stage is actually dimensioned for tensile loads instead of compression loads, as typical with most rocket structures. The net axial and lateral forces transferred during ascent from each booster to the core can be expressed as:

$$\begin{cases} F_{B,ax} = T_B - A_{aero,B} - M_B\cdot(g_{ax} + n_{ax}) \\ F_{B,lat} = N_{aero,B} + M_B\cdot(g_{ax} + n_{ax}) \end{cases}$$ with ($\boldsymbol{T_B}$, $\boldsymbol{M_B}$, $\boldsymbol{A_{aero,B}}$, $\boldsymbol{N_{aero,B}}$) being booster's thrust, mass and aerodynamic forces.

6.5.3. Mass distribution

The longitudinal distribution of all structural and non structural masses along the beams constituting the core and the booster sets is of large importance in the determination of the inertial loads and therefore of the structural masses. Each of the $\boldsymbol{N_{mi}}$ mass items is described in the model by four double values per each load case, building up four $\boldsymbol{N_{lc}}$**-by-**$\boldsymbol{N_{mi}}$ matrixes synthesizing the ELV's longitudinal inertia properties. For the j-th mass item at the k-th load case, these are:

- Mass $\boldsymbol{M_{jk}}$, which varies depending on the load case for propellants and may become null with jettison events.

- Start and end position $X_{min,jk}$ and $X_{max,jk}$, which are coincident for several mass items assumed to be concentrated: booster attachments, engine assemblies, pressurization systems, avionics and power systems, roll control system.
- Cantilever position $X_{cant,jk}$, which is null for regular items but defines the reaction station (location where the inertial loads for the full mass are introduced) for those mass items which are assumed to be cantilevered: engine assemblies and thrust structures, payload and payload adapter, pressurization system).

Liquid fuel and oxidizer are considered cantilevered only along the longitudinal axis: axial inertial loads are applied at the lower bulkhead, assuming that no axial force is transferred at the walls, whereas lateral inertial loads are applied throughout the full tank's length. The solid propellant grain is instead assumed to be bonded to the lateral surface of the casing, resulting in the axial inertial load also being distributed along the full length of the grain.

In total, 3 mass items are defined for the payload composite: *payload* ***PL***, *payload adapter* ***PLA*** and *payload fairing* ***PLF***. Besides, at most 17 mass items can be defined for each stage and boosters set: *solid grain case SpC*, *solid grain* ***Sp***, *oxidizer tank* ***OxT*** and *fuel tank* ***FT***, *oxidizer* ***Ox*** and *fuel* ***F***, *intertank* ***IT***, *interstage* ***IS***, *forward skirt* ***FSk*** and *aft skirt* ***ASk***, *thermal insulations* ***TI***, *unused propellants* ***UP***, *pressurization system* ***PS***, *engines assembly* ***EA***, *thrust frame* ***TF***, *forward* and *aft booster's attachments* ***FBA*** and ***ABA***, upper stage's *avionics and eps* ***AES***, *pad interface structure* ***PI***. Note that for j-th stage, the forward skirt is the short structure surrounding the top ellipsoidal dome of the upper tank (for LP) or of the grain case (SP). The interstage for stage j is then assumed to cover the length from the top of its upper tank or grain case up to the separation plane with (j+1)-th stage. Finally, the aft skirt of (j+1)-th stage starts from this separation plane and continues up to bottom ellipsoidal dome of the lower tank (LP) or grain case (SP) of (j+1)-th stage.

Several simplifying assumptions were taken for the definition of the longitudinal masses distribution, in particular:

- The weights of SP igniter and LP feed system are assumed concentrated in the EA mass item.
- Liquid tanks and solid grain case are assumed to be cylinders of length equal to the total tank's length, with uniform distribution of propellant, therefore neglecting the lower quantity of propellant in the domes.
- Booster attachment locations are taken as 20% of the booster's length for the ABA and 30% of the booster's nose ogive length for the FBA.
- TI and UP are assumed to be uniformly distributed over the whole lengths of the tanks and grain case.

Figure 100, **Figure 102** and **Figure 104** show mass distribution examples for Ariane 5 ECA (core and booster) and VEGA.

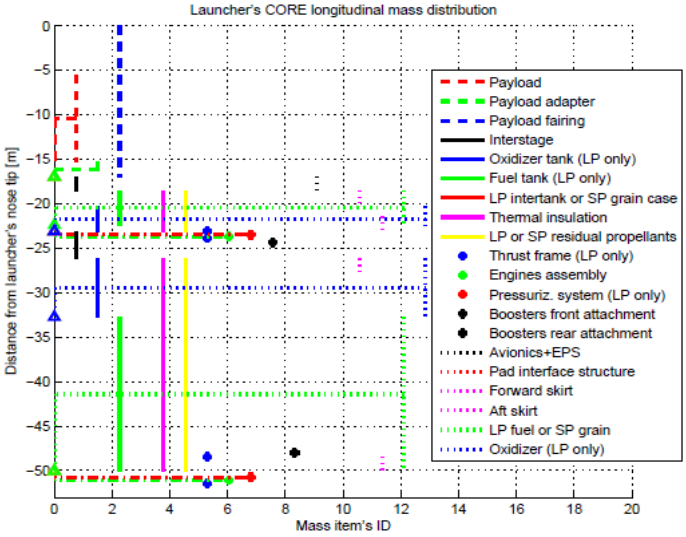


Figure 100: Mass distribution along Ariane's core beam. Star: concentrated mass. Triangle: reaction station of cantilever mass.

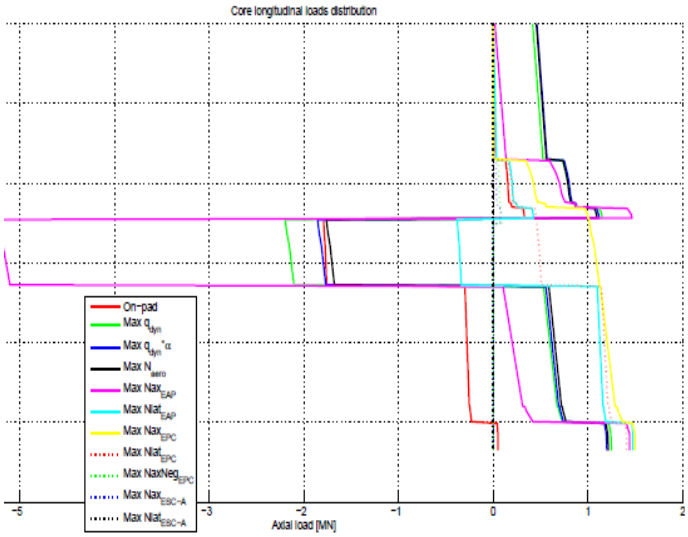


Figure 101: Axial load distribution along Ariane's core beam approximation. Negative values: tensile loads.

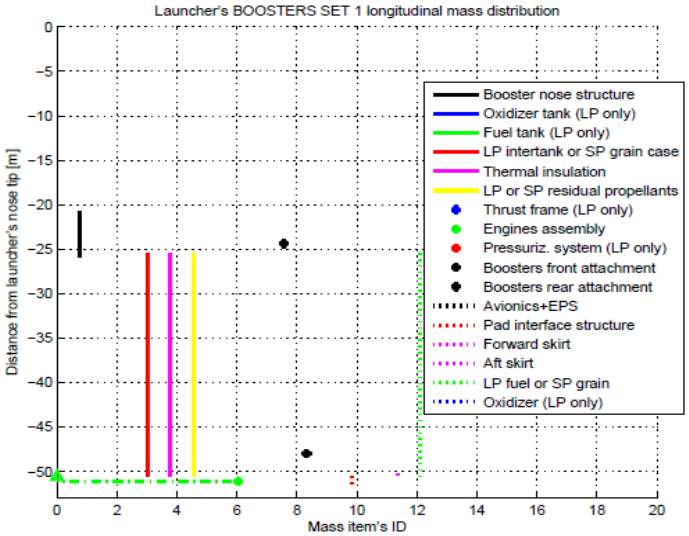


Figure 102: Mass distribution along Ariane's booster beam. Star: concentrated mass. Triangle: reaction station of cantilever mass.

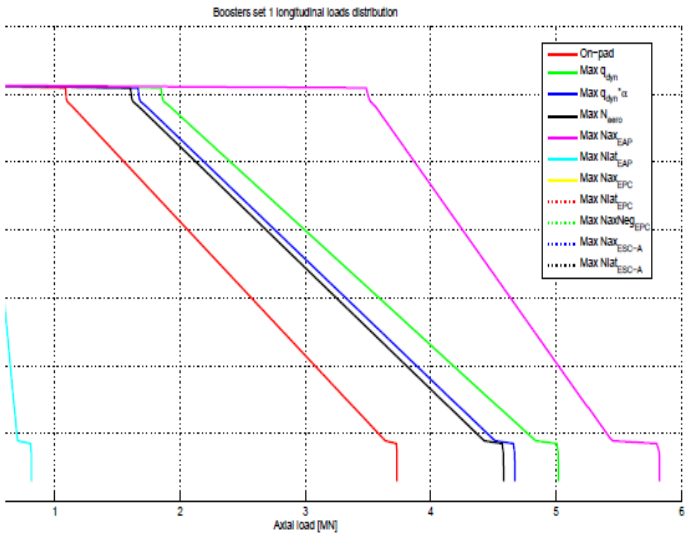


Figure 103: Axial load distribution along Ariane's boosters beam approximation. Negative values: tensile loads.

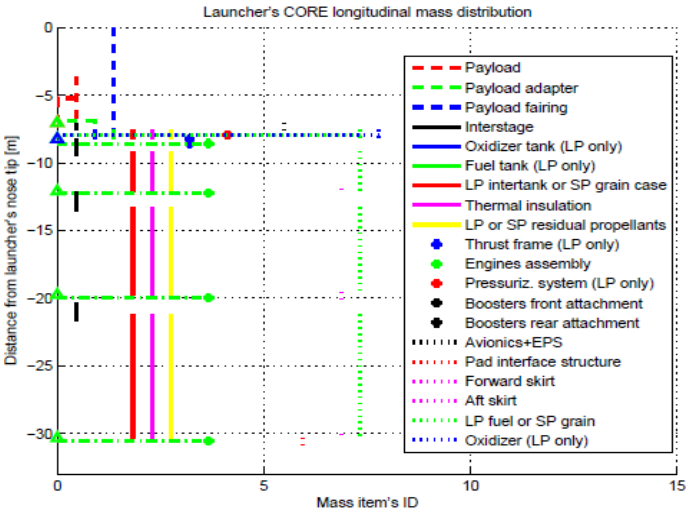


Figure 104: Mass distribution along VEGA's core beam. Star: concentrated mass. Triangle: reaction station of cantilever mass.

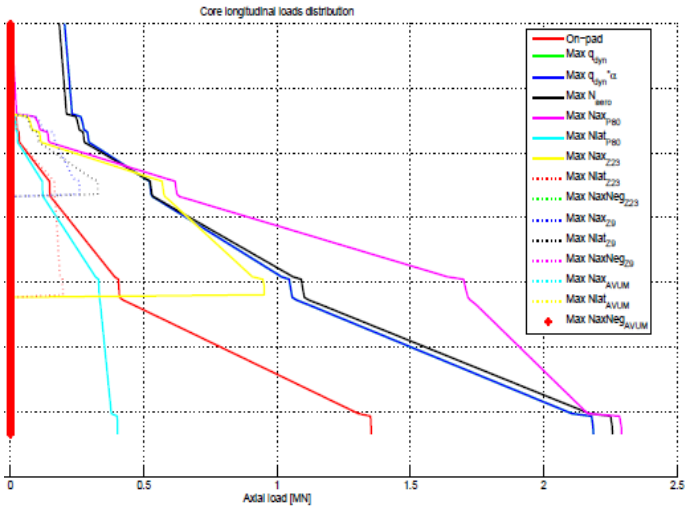


Figure 105: Axial load distribution along VEGA's core beam approximation. Negative values: tensile loads.

6.5.4. Axial, shear and bending flight loads

The *primary external flight loads* are intended as the *axial force P, shear force T and bending moment M* computed along the beam(s) for each load case and caused by the combination of gravitational, inertial and aerodynamic loads, as well as by the forces transferred from the boosters to the core and vice-versa. The total thrust at the bottom of the core and of the boosters closes the axial load diagram, whereas the lateral thrust provided by the TVC closes the diagram of shear force and bending moment. Longitudinal stations ***(X_j, R_j)*** along each beam are first defined, with a default distance of 20 cm. Then, following classical loads analysis practice, ***(P, T, M)*** loads are computed for each station j starting from the tip of the beam, considering contributes of all masses multiplied by the sum of gravitational and inertial accelerations, of the aerodynamic forces and of the booster-core transfer loads, aft of the attachment locations. ***P*** and ***T*** are therefore defined as:

$$\begin{cases} P_j = A_{aero,f} + A_{aero,d}\cdot\dfrac{X_j}{L} + (n_{ax} + g_{ax})\cdot\displaystyle\sum_{k=1}^{N_{mi}}(m_k\cdot\Delta X_{jk}) \pm F_{ax,B}\cdot(X_j > X_{F_{ax,B}}) \\ T_j = 2\cdot n_{aero,tip} + \dfrac{n_{aero,tip} - n_{aero,base}}{L}X_j + (n_{lat} + g_{lat})\cdot\displaystyle\sum_{k=1}^{N_{mi}}(m_k\cdot\Delta X_{jk}) \pm F_{lat,B}\cdot(X_j > X_{F_{lat,B}}) \end{cases}$$

where ΔX_{jk} defines whether the station X_j is located forward, within or aft mass item M_k as follows:

$$\begin{cases} \Delta X_{j,k} = \dfrac{m_k \cdot (X_j - X_{min,k})}{X_{max,k} - X_{min,k}} & if \;\; X_{min,k} \le X_j \le X_{max,j} \;\; (M_k \;\; distributed) \\ \Delta X_{j,k} = 1 & if \;\; X_j \ge X_{max,k} \;\; (M_k \;\; non \;\; cantilevered) \; or \; X_j \ge X_{cant,k} \;\; (M_k \;\; cantilevered) \\ \Delta X_{j,k} = 0 & if \;\; X_j \le X_{min,k} \;\; (M_k \;\; non \;\; cantilevered) \; or \; X_j \le X_{cant,k} \;\; (M_k \;\; cantilevered) \end{cases}$$

The bending moment M can afterwards be obtained by integration of the shear diagram, with the addition of concentrated moments introduced at the reaction points of the cantilevered items. These moments tend to alleviate the bending load for items placed on top of the reaction station (e.g. fairing) and to increase it for overhanging items (e.g. engines). Examples of axial loads distributions are shown in **Figure 101**, **Figure 103** and **Figure 105**, again for Ariane 5's core and boosters and for VEGA.

In addition to the primary flight loads calculated along the beams, which will be used to size those structural components defined as primary (i.e. tanks/case, interstages, intertanks, structural skirts, …), the *secondary external flight loads* are also computed, which will be used to size components defined as "secondary" since they are not part of the main load paths. In particular, these are the *payload adapter*, the *thrust frames* for all liquid propellant stages, and the *aerodynamic aft skirts*. For each of these components, P, T and M are separately computed. The payload adapter for example must withstand the gravitational and inertial loads due to the payload mass as well as its own mass, as follows:

$$\begin{cases} P = (M_{PL} + M_{PLA}) \cdot (g_{ax} + n_{ax}) \\ T = (M_{PL} + M_{PLA}) \cdot (g_{lat} + n_{lat}) \\ M = T \cdot (X_{CoG,PL} - X_{CoG,PLA}) \end{cases}$$

whereas the thrust frames must withstand axial loads due to both the thrust and the gravitational/inertial loads on the EA. Additionally, quite severe shear and bending loads originate for the maximum deflection angle, which result in significant lateral thrusts. The formulas used to estimate the loads acting on TF structures are therefore the following:

$$\begin{cases} P = T_{eng} - (M_{eng} + M_{TF}) \cdot (g_{ax} + n_{ax}) \\ T = T_{eng} \cdot sin(\delta_{TVC,max}) + (M_{eng} + M_{TF}) \cdot (g_{lat} + n_{lat}) \\ M = T \cdot (X_{CoG,eng} - X_{CoG,TS}) \end{cases}$$

Finally, aerodynamic skirts only need to support their own weight, plus a small contribute of internal pressure due to aerodynamic reasons which is conservatively taken as ***0.1 bars*** in this study.

6.5.5. Structural components definition

In order to proceed with the sizing of the structural components, they first have to be properly characterized in terms of location along the beams, geometric properties (i.e. X and R coordinates of a number of longitudinal sections), employed material and structural configuration. To tackle the design of any generic ELV's architecture, two structural components are considered for the payload composite - ***PLF*** and ***PLA*** - and at most nine structural components for each stage/booster: ***SpC***, ***OxT***, ***FT***, ***IT***, ***IS***, ***FSk***, ***ASk***, ***PI*** and ***TF***. The other mass items described in **Paragraph 6.5.3** are considered non structural elements, and the mass assessed in the other disciplines (Propulsion or Weights) is not changed within Structures.

For each of the above structural components, the start and end position are taken from the mass items $\boldsymbol{X_{min}}$ and $\boldsymbol{X_{max}}$ matrixes, and their length is divided in longitudinal stations with a wider spacing with respect to the beams' stations, by default ***50 cm***. In this case the discretization does not only have a computational purpose: the distance between stations also represents the length of the structural panels, over which the thickness is assumed to be constant.

Following the geometric outline, the loading condition at each station k of each structural component j is defined by setting two variables $\boldsymbol{p_{ext,jk}}$ and $\boldsymbol{p_{int,jk}}$ as follows:

- $\boldsymbol{p_{ext,jk}=0}$: station k of component j does not carry any external load ***(P,T,M)***. This is the case for the upper and lower domes of liquid tanks and solid motor cases, or for the whole liquid tanks in case of multiple or enclosed tanks.
- $\boldsymbol{p_{ext,jk}=1}$: station k of component j carries the primary external loads ***(P,T,M)***, to be interpolated from the beam approximation. This is the case for the cylindrical sections of liquid tanks and solid motor cases and for the interstages, intertanks, forward and aft structural skirts, and pad interface structures.
- $\boldsymbol{p_{ext,jk}=2}$: station k of component j carries the secondary external loads ***(P,T,M)***, as described above for the payload adapter, the thrust frames and the aerodynamic skirts.
- $\boldsymbol{p_{int,jk}=0}$: station k of component j does not have to withstand any internal pressure.
- $\boldsymbol{p_{int,jk}=1}$: station k of component j has to withstand an internal pressure taken as the Maximum Expected Operating Pressure (MEOP) for solid motor cases or as the sum of ullage and head pressures for oxidizer/fuel liquid tanks.

The combination of the pressure and flight loads determines the loading condition at each station of each structural component. Examples for Ariane 5 ECA and VEGA are reported in **Figure 106**, **Figure 107** and **Figure 108**, showing the

main load carrying structure as well as the aforementioned secondary structures. The primary load carrying structure of Ariane 5's core, rather complicated, is constituted by the following sequence of structural components from the top down: PLF, VEB, ESC-A's FSk, cylindrical section of ESC-A's hydrogen tank, ESC-A's aft skirt, EPC's IS and FSk (i.e. in the European industry terminology, JAVE structure), and EPC's integral common bulkhead oxygen-hydrogen tank. ESC-A's oxygen tank is enclosed in the hydrogen tank, and therefore only carries the internal pressure loads, whereas secondary structural components are EPC's aft aerodynamic skirt (Ariane 5 is connected to the pad through the boosters), the thrust structures for both stages, and the PLA.

The last properties that must be defined for each structural component are the structural material and the stiffening configuration. Two optimization variables are used for this purpose, ***SM*** and ***SC*** than can be frozen by the user to impose a given selection. The materials available by default, with mechanical properties shown in **Table 32**, are Al alloy 7075-T6, Ti alloy 6Al- 4V, low alloy steel 4340, Al-Li alloy 2195 and carbon-epoxy composite, or a custom material can be externally provided. Three stiffening concepts are instead available besides the shell configuration without stiffeners: simple integrally stiffened shell, Z-stiffened shells, and truss-core sandwich shell. These are characterized by different values of buckling efficiency, minimum gage parameter and frame stiffness coefficient, reported in **Table 33**. Note that in case the loads analysis shows that a structural component is never subject to compression, an unstiffened shell configuration is assumed for minimum weight, without considering the value given by ***SC***.

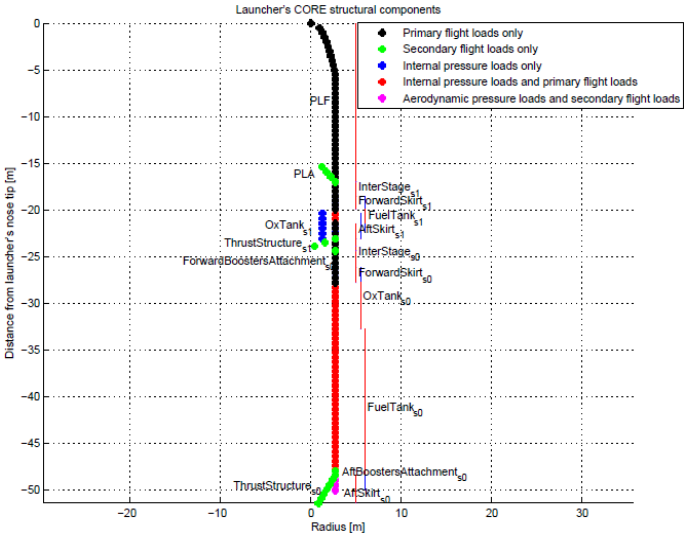


Figure 106: Structural components distribution and loading conditions along Ariane 5 ECA's core beam approximation.

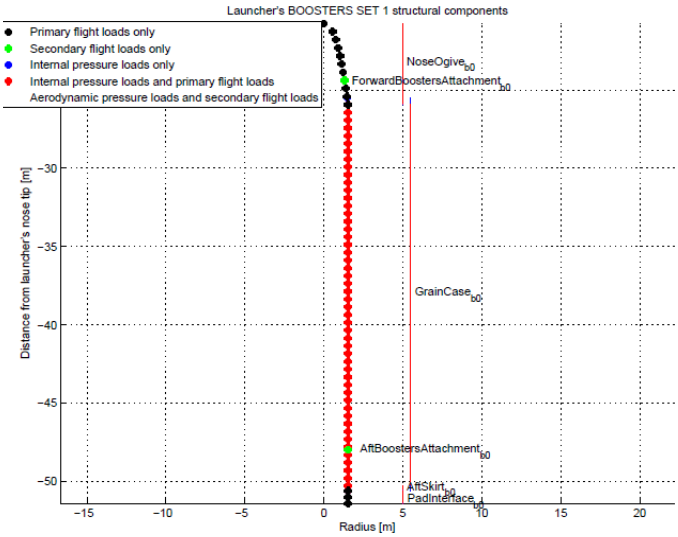


Figure 107: Structural components distribution and loading conditions along Ariane 5 ECA's booster beam approximation.

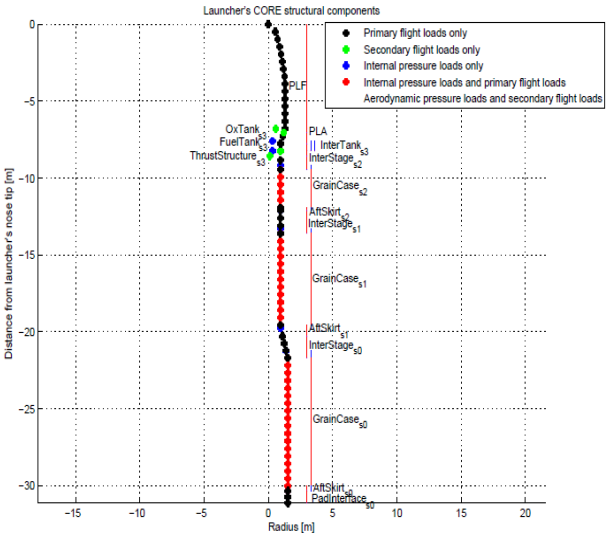


Figure 108: Structural components distribution and loading conditions along VEGA's beam approximation.

Material	E [GPa]	σ_{YTS} [MPa]	σ_{UTS} [MPa]	ρ [kg/m^3]	$t_{min,gage}$ [mm]
Al 7075-T6	72	505	570	2810	0.203
4340 Steel	200	1496	1793	7833	0.254
Ti 6Al-4V	110	828	1030	4420	0.305
Al-Li 2195	77	690	710	2600	0.203
CFRP	70	530	810	1600	0.500

Table 32: Assumed values of Young modulus *E*, tensile yield strength σ_{YTS} and ultimate strength σ_{UTS}, density ρ and minimum gage thickness for the five selected materials. Data are mostly taken from online sources

Configuration	ε	K_f	K_{mg}
Integrally stiffened shell	0.656	5.24	2.463
Z-stiffened shell	0.911	5.24	2.475
Truss-core sandwich shell	0.605	5.24	4.310

Table 33: Assumed values of buckling efficiency ε, stiffness coefficient K_f and minimum gage parameter K_{mg} for the three selected structural configurations. Data are taken from the work in Shanley ([137]).

6.5.6. Internal running loads: axial compression, axial tensile, hoop and shear

The worst-case running loads at each longitudinal station of each structural component can be derived from the station's geometry, internal pressure ($\boldsymbol{p=p_{head}+p_{ullage}}$), external primary or secondary flight loads ($\boldsymbol{P}$, $\boldsymbol{T}$, $\boldsymbol{M}$), and mechanical properties due to both structural material and configuration. Three types of internal running loads are considered for the sizing of thin-walled structural components of ELVs:

- Axial compression stress $\boldsymbol{N_{x,c}}$ [N/m]: originated from the axial compression force $\boldsymbol{P_c}$ and from the bending moment $\boldsymbol{M}$ of the external flight loads.
- Axial tensile stress $\boldsymbol{N_{x,t}}$ [N/m]: originated from tensile force $\boldsymbol{P_t}$, bending moment $\boldsymbol{M}$, and internal pressure $\boldsymbol{p}$.
- Hoop (circumferential) stress $\boldsymbol{N_y}$ [N/m]: for which internal ullage and head pressures are the only cause.
- Shear stress $\boldsymbol{N_{xy}}$ [N/m]: originated from shear forces of the external loads.

Arbitrarily considering all loads ($\boldsymbol{p}$, $\boldsymbol{P_c}$, $\boldsymbol{P_t}$, $\boldsymbol{T}$, $\boldsymbol{M}$) to have a positive sign, the following equations are used to obtain the internal running loads:

$$N_{x,t}=\frac{P_t}{2\cdot\pi\cdot R}+\frac{M}{\pi\cdot r^2}+\frac{p\cdot r}{2};\quad N_{x,c}=\frac{P_c}{2\cdot\pi\cdot R}+\frac{M}{\pi\cdot r^2}-\frac{p\cdot r}{2};\quad N_y=\begin{cases}p\cdot r & \text{for cylindrical tanks}\\ p\cdot r/2 & \text{for spherical tanks}\end{cases};\quad N_{xy}=\frac{T}{\pi\cdot R}$$

At this point, the worst-case running loads per each station of each structural component can be extracted from the load cases, treating tensile and compression loads separately due to the different involved failure modes. Some ELV components are subject only to compression loads, as typically (but not always) intertanks, interstages or similar structures, others only to tensile loads, generally solid motor cases and sometimes LP tanks. A contingency factor equal to ***1.5*** is applied to conservatively cover all uncertainties. Additional Factors of Safety (FoS) will be taken into account in the sizing of the structural components following standard NASA practice, as presented in the next paragraph.

6.5.7. Structural components sizing

The structural design is assumed to be of the shell-frames type: *shell panels* withstand tensile, hoop and shear stresses, *shell stiffeners* (integrally stiffened/Z-stiffened/sandwich) prevent compression buckling of the panels, and *ring frames* prevent general instability failure of the fuselage. A detailed description of the sizing procedure for this type of structures is given in references **[133]** (pp. 11-18) and **[134]** (pp. 19-26) hence only a short summary including the basic equations are given below. A material minimum gage restriction is imposed, and the failure modes considered are yield strength, ultimate strength and buckling. For each station of each structural component, the procedure is the following:

1. Compute minimum gage shell thickness $\boldsymbol{t_{s,mg}}$, simply obtained from the minimum material thickness $\boldsymbol{t_{min,gage}}$ as:
   $t_{s,mg}=t_{min,gage}\cdot K_{mg}$
2. Compute the minimum shell thickness for yield strength $\boldsymbol{t_{s,YTS}}$ using classical Von Mises theory to obtain the yield limit stress and the material's $\boldsymbol{\sigma_{YTS}}$ to derive the necessary thickness:
   $t_{s,YTS}=N_{YTS}/\sigma_{YTS}\quad with\ \ N_{YTS}=FoS_{YTS}\cdot\sqrt{N_x^2+N_y^2-N_x\cdot N_y+N_{xy}^2}\quad [N/m]$
   If the component is subject to both compression and tensile axial loads, $\boldsymbol{N_{x,c}}$ and $\boldsymbol{N_{x,t}}$ are separately substituted in the Von Mises equation in combination with the worst-case $\boldsymbol{N_y}$ and $\boldsymbol{N_{xy}}$, and the resulting largest Von Mises stress N_{YTS} is used to determine the thickness. $\boldsymbol{FoS_{YTS}=1.0}$ is assumed as by NASA standard for yield in unmanned vehicles (**[109]**), which would typically increase to 1.1 for manned vehicles.
3. Compute the minimum shell thickness for ultimate strength $\boldsymbol{t_{s,YTS}}$, using Maximum Principal Stress criterion to obtain the ultimate limit stress and the material's $\boldsymbol{\sigma_{UTS}}$ to derive the necessary thickness:
   $t_{s,UTS}=N_{UTS}/\sigma_{UTS}\quad with\ \ N_{UTS}=FoS_{UTS}\cdot\left(0.5\cdot(N_x+N_y)\pm\sqrt{0.25\cdot(N_x-N_y)^2+N_{xy}^2}\right)\quad [N/m]$
   As for the yield criterion, both $\boldsymbol{N_{x,c}}$ and $\boldsymbol{N_{x,t}}$ loads are tested. The FoS for ultimate strength in unmanned vehicles is set to ***1.15*** for SRM cases and ***1.25*** for other flight structures (increasing to 1.25/1.4 for manned systems).
4. Compute the minimum shell thickness for buckling $\boldsymbol{t_{s,B}}$, using a minimum weight equation for wide column stiffened shells first determined by Crawford and Burns in 1963 (**[136]**):
   $t_{s,B}=\sqrt{N_{x,c}\cdot L_f/\epsilon\cdot E}$
   where the thickness depends on the compression load $\boldsymbol{N_{x,c}}$, frames spacing $\boldsymbol{L_f}$, buckling efficiency $\boldsymbol{\varepsilon}$ and material's stiffness $\boldsymbol{E}$. The frames spacing is taken from the work of Shanley ([137], 1960) as the optimum spacing which minimizes the sum of shell thickness $\boldsymbol{t_{s,B}}$ and smeared equivalent thickness of the frames $\boldsymbol{t_f}$, therefore minimizing the total weight of buckling critical structures:
   $L_{f,opt}=\sqrt{6\cdot R^2\sqrt{C_f\cdot\pi\cdot\epsilon/K_f}}$

where Cf = 1/16000 is the empirically determined, non-dimensional, *Shanley's constant* and $\boldsymbol{K_f}$ is the aforementioned frames stiffness coefficient depending on the structural configuration.

5. Select the shell's thickness as: $\boldsymbol{t_s = max(t_{s,mg}, t_{s,YTS}, t_{s,UTS}, t_{s,B})}$.
6. If the shell is *not buckling critical* (i.e. $\boldsymbol{t_s}$ is determined by one of the other failure criteria, hence $t_s \neq t_{s,B}$), the frames spacing can be increased to reduce their mass, up to the point where the shell actually becomes critical. $\boldsymbol{L_f}$ is thus recomputed by inverting the first equation above as: $L_f = t_S^2 \cdot \epsilon \cdot E / N_x$
7. Compute the frames smeared equivalent thickness $\boldsymbol{t_f}$ from Shanley's criterion as: $t_f = 2 \cdot R^2 \cdot \sqrt{C_f \cdot \pi \cdot N_x / E \cdot K_f \cdot L_f^3}$.
8. Get the total equivalent thickness of the section as $\boldsymbol{t_{eq} = t_s + t_f}$.

When the equivalent thickness $\boldsymbol{t_{eq}}$ of each longitudinal station of a given structural component is available, it is possible to easily compute its primary structural mass by integration of the thicknesses as:

$$M_{ps} = \rho \cdot \sum_{j=1}^{N_{stations}-1} S_{lat,j} \cdot max(t_{eq,j}, t_{eq,j+1})$$

where the lateral surface $\boldsymbol{S_{lat}}$ is computed assuming linearly varying radius between sections (i.e. conical lateral surface, if radius is not constant). A few special cases are related to the following structural components:

- For spherical tanks, the thickness is assumed constant at its maximum value (i.e. max head pressure at the bottom).
- For the intertank structures in spherical tanks configuration, which represent the supporting structures of the tanks, the thickness is again assumed constant but a structural support plane is added besides the lateral surface, of diameter equal to the stage's diameter.
- Boosters nose ogive and boosters aft skirts are assumed to be reinforced for concentrated loads at the structural connections with the core, therefore resulting in an arbitrary reinforcement factor of ***2.0*** on the calculated thickness.

6.5.8. Structural model calibration with existing launcher structures

The component weights determined with the complex structural analysis procedure presented in the previous paragraphs is based on fundamental structural principals for minimum weight, hence can only reflect the optimum primary structural mass of the components. In reality, a large number of factors contribute to increasing the actual mass much above the value computed with these methods. Shanley provides in **Chapter 12** of **[137]** a good overview of such factors, stating that the overall correction factor may be in the order of 2.0 (100% weight increase) or more for conventional aircraft structures.

The main reasons for non-optimum weights can be summarized in the following key points:

- Structural joints and fasteners.
- Doublers and other secondary structures for joints and welds.
- Tapered profiles discretization, which applies mostly to aircraft wings.
- Standard gages for which the required thickness cannot be actually employed.
- Reinforcements due to cut-outs and local loading.
- Bulkheads and other secondary elements.
- Construction or cost motivations for the selection of non-optimum design.
- Other elements typical of rocket vehicle structures such as tanks sealing, keel beams, fail-safe straps, attachment fittings, pressure webs, ...

All of these factors are impossible to be evaluated with analytical methodologies, hence must be aggregated in a single corrective factor - empirically calibrated – which allows determining the secondary, or non-optimum, structural mass of the components as: $\boldsymbol{M_{ss} = K_{ss} \cdot M_{ps}}$. This approach may at first sight seem particularly artificial: after going through the ordeal of defining a complex structural model with analytical sizing relations, the resulting underestimated mass figures are arbitrarily corrected by an empirical factor which aims at matching the actual values. However, this methodology ensures reasonable accuracies in terms of absolute mass estimates and, more importantly, is capable of representing the basic physics at the basis of structural sizing. This is a very relevant advantage over purely empirical WERs, since it ensures that much more reliable structural design information is used, and justifies the exploitation of similar approaches in **[134]** and **[135]** for in-line launchers and in **[138]** for aircraft fuselages.

The tuning of the coefficient $\boldsymbol{K_{ss}}$ is nevertheless critical for the accurate estimation of the structural weights with the developed methodology; hence, different values were correlated for different types of structural components, on the basis of available mass information for the European launch vehicles Ariane 5 ES, Ariane 5 ECA and VEGA. Results for this regression analysis are presented in Table 4.6, which summarizes the structural coefficients for the different types of structures. Note that several coefficients are very large since they do include also the weight of non structural subsystems, as for example the separation systems and pyrotechnics for boosters nose ogives and aft skirts. The correction factor for SP motors is instead actually negative (i.e. structural mass estimated with optimum-weight principals is larger than actual

mass), due to the fact that the value of MEOP is known with good accuracy, hence the contingency factor of ***1.5*** on the loads uncertainty results in an overestimation of the structural weight.

Component type	Integral OxT/FT	Non-integral OxT/FT	SpC	IT/IS/FSk/ASk/PI	PLF	TF
$\mathbf{K_{ss}}$	0.490	1.318	-0.008	4.211	1.870	7.594

Table 34: Correlated values of the structural correction factor for non-optimum weights in different types of components.

Singing Don't worry, about a thing, 'Cause every little thing is gonna be all right!

CHAPTER 7

EARLY PRELIMINARY MODELS VALIDATION

The validation of the early preliminary models followed the same procedure developed for the conceptual models, with the goal of ensuring a fair comparison between the two successive modelling steps and evincing useful information regarding the applicability of the developed multidisciplinary analysis for real-world applications. In this chapter, few examples of disciplinary-level validation are first given for selected aspects related to the enhancements introduced for V2. Then, an outlook on the sensitivity analyses is followed by the main results of MDA and MDO tests for Ariane 5 ECA and VEGA, in direct comparison with the quantitative figures presented in **Chapter 5**.

7.1. Stand-alone disciplines validation and sensitivity analyses

7.1.1. Propulsion

For the propulsion discipline, most of the model improvements are related to the introduction of new modelling features, enabling for instance a more realistic representation of the SRMs thrust and performance profiles, a physical correlation of the effect of varying tanks pressure on both pressurization system mass and pumps cavitation, and the exploitation of specific propulsion technologies such as multiple thrust chambers, extendable and submerged nozzles. For LREs, these modifications determined the necessity to update specific impulse corrections and inert mass estimation models, resulting however in very small changes in accuracy (i.e. slight increase in M_{eng} accuracy for LREs).

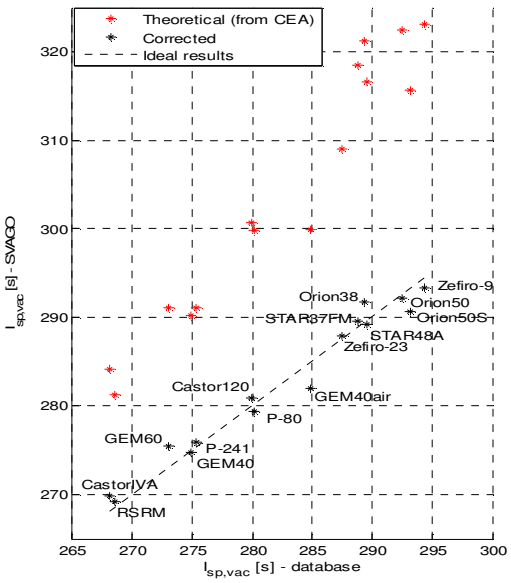


Figure 109: SRM models validation, theoretical and corrected $I_{sp,vac}$ for Avio and ATK engines.

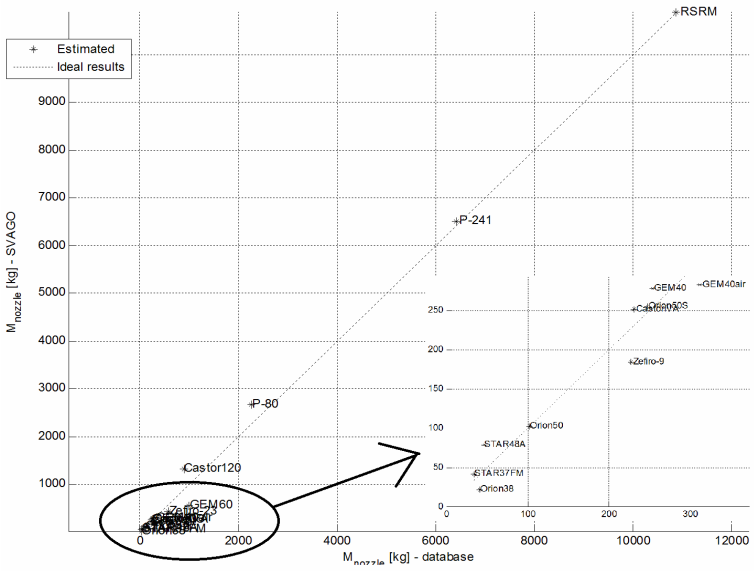


Figure 110: SRM models validation, nozzle mass for Avio and ATK engines, including a zoom for small motors (<300 kg nozzle).

On the contrary, sensible improvements were obtained in the assessment of both $I_{sp,vac}$ and M_{nozzle} for SRMs, mainly due to the availability of more real-world data from ATK website. These spurred the development of new correlations of the I_{sp} loss factor with the nozzle's expansion ratio and of the nozzle's mass with the engine's vacuum thrust. In front of maximum

errors with V1 models in the order of 2% on $I_{sp,vac}$ and 40% on M_{nozzle}, the validation process (15 SRMs from Avio and ATK) highlighted very significant improvements in accuracy, with (μ=-0.03%, σ=0.27%, M=0.41%) for $I_{sp,vac}$ and (μ=+1.14%, σ=14.35%, M=24.47%) for M_{nozzle}. **Figure 109** and **Figure 110** summarize these validation results.

As regards to the pressurization system and pump cavitation analyses, very few quantitative data are available for existing launchers, but is enough to understand the order of magnitude of the accuracy of the developed models. In particular, detailed information was retrieved for the pressurization systems of NASA's Saturn V and Space Shuttle (from historical archives[47]) as well as for Ariane 5 ECA/ES and VEGA. These include mass figures for the He gas/tanks of LOx systems (EPC, ESC-A, S-IC) and of the N_2O_4/MMH systems (EPS, AVUM) and the required masses of evaporated propellant for cryogenic stages (S-IC LH_2, S-II LO_x and LH_2, ET LO_x and LH_2, EPC LH_2, ESC-A LH_2). Validation with respect to these systems showed maximum errors on the single components in the order of 30%. However, the total pressurization system mass of the different stages is reasonably close to the actual figures, as reported in **Table 35**. Results in terms of total $M_{PressSys}$ are therefore satisfactory, and additional modelling efforts should be focused on evaporated LH_2 and heated He mass estimation, which are showing the largest inaccuracies.

Total $M_{PressSys}$	ET	S-IC	S-II	EPC	EPS	AVUM
Actual [kg]	1713	5573	1947	390	188	28
Estimated [kg]	1860	5494	1734	367	226	24
Error [%]	+8.6	-1.4	-10.9	-5.9	+20.2	-11.9

Table 35: Validation of the total pressurization system mass for launch vehicle stages of known design data.

Since no detailed WBS is available for existing engines, it is not possible to validate the WER for boost TPA mass. Nevertheless, the reliability of the cavitation analysis can be simply assessed by verifying the capability to predict the necessity of boost pumps on existing engines. Execution of the turbopump analysis on all pump-fed LREs in database highlighted how no GG or EC engine requires a boost pump. For SC LREs instead, the use of boost pumps is correctly represented in all cases except for the Russian 11D58MF and RD-275M engines. Although the model could be improved with the introduction of a more complete engine cycle analysis, it is deemed sufficient for the early preliminary level of detail since a complete bottom-up physics-based mass estimation methodology would also be required for achieving sensible improvements in assessment of M_{eng}.

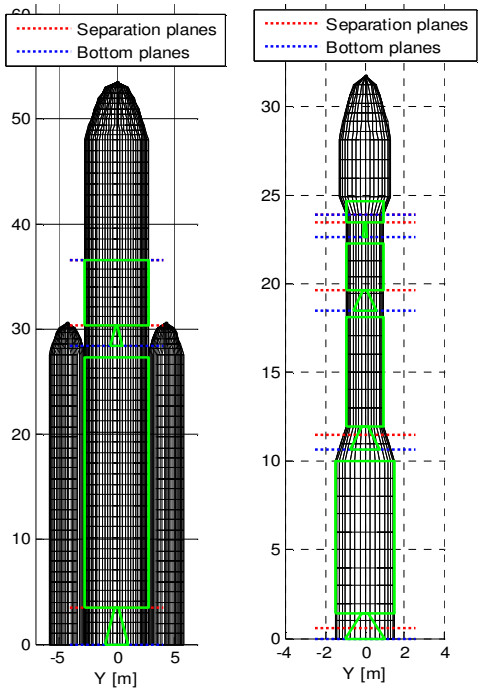


Figure 111: External geometry and internal planes definition for Ariane 5 ECA and VEGA.

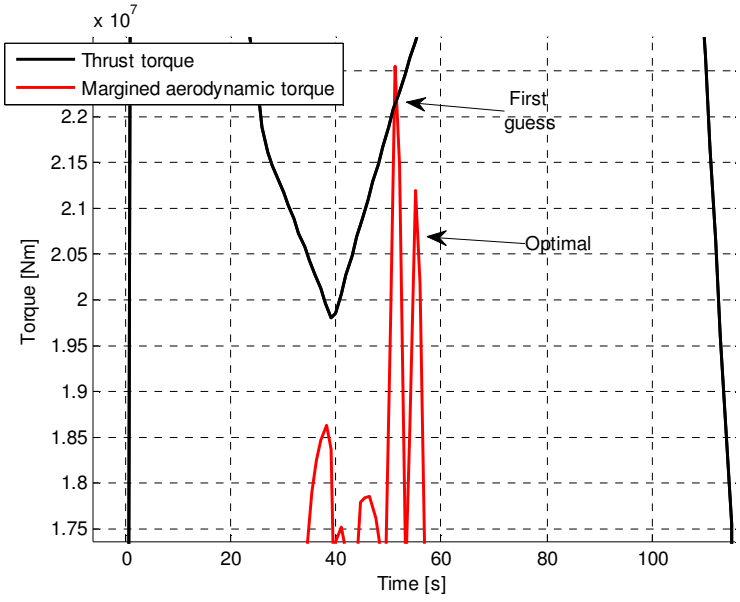


Figure 112: Available thrust torque and required torque for static controllability (with a margin of 1.5) for Ariane 5 ECA's typical GTO flight.

[47] Saturn V: www.apollosaturn.com/saturnv.htm, Space Shuttle: http://science.ksc.nasa.gov/shuttle/technology/sts-newsref/stsref-toc.html

7.1.2. Geometry, Weights and Structures

The most relevant enhancement in the Geometry models lays in the definition of the separation plane from one stage to the next, which strongly affects the structural mass of interstage, forward and aft skirts. **Figure 111** represents the external and internal geometries of Ariane 5 ECA and VEGA, including the positions of the bottom and separation planes of each stage. Note the under-fairing configuration of AVUM upper stage, which is covered for most of its structure by the PLF.

The weight estimation validation for V2 involved assessing both the first guess inert masses from the Weights discipline and the converged values from the Trajectory-Structures iterative cycle. Due to the much longer set-up process and the unavailability of detailed WBS for non European launchers, the validation was only repeated for Ariane 5 ECA/ES and VEGA, totalling 3 PLFs (Ariane 5 long, Ariane 5 short, VEGA), 4 liquid stages (EPC, EPS, ESC-A, AVUM) and 4 solid motors (P241, P80, Z23, Z9). Examples of WBS obtained with the two mass estimation approaches are reported in **Table 36** and **Table 37** (EPC and P80), whereas the statistical figures of total inert mass accuracy are presented in **Table 38**, together with those for the other disciplinary output parameters from Propulsion and Aerodynamics. The inert masses of lower and upper LP stages are condensed in a single entry, as well as those of SP stages/boosters, due to the limited number of empirical data points, which would result in totally unreliable statistical information if further split.

EPC[48]	Actual design data	Weights discipline estimates (WERs)	Structures discipline estimates
Usable propellant	**173300 kg**	**173300 kg**	**173300 kg**
Unused propellant	2600 kg guessed 1.5%	2228 kg	nsc[49]
Interstage	1743 kg ESA internal	1102 kg	1666 kg
Ox + Fuel tank	4989 kg ESA internal[50]	7336 kg	6571 kg
Thermal insulation	NA	499 kg	nsc
Thrust frame	1754 kg ESA internal	1770 kg	1755 kg
Engines assembly	2370 kg ESA internal	2370 kg	nsc
Pressuriz. system	390 kg ESA internal[51]	543 kg	nsc
Booster attachment	1010 kg ESA internal	802 kg	nsc

Table 36: WBS for Ariane 5 ECA's EPC cryogenic core stage: actual data, mass estimates from Weights and Structures.

P80	Actual design data	Weights discipline estimates (WERs)	Structures discipline estimates
Usable propellant	**87732 kg**	**87732 kg**	**87732 kg**
Unused propellant	632 kg	632 kg	nsc
Interstage	321 kg www.elv.it	161 kg	420 kg
SP case	4770 kg [52]	6501 kg	5681 kg
Thermal insulation	NA	309 kg	nsc
Engines assembly	2638 kg [53]	2607 kg	nsc
Pad interface	228 kg www.elv.it	693 kg	233 kg

Table 37: WBS for VEGA's P80 SRM first stage: actual data, mass estimates from Weights and Structures.

Discipline	Parameter	Description	E [%]	M [%]	μ [%]	σ [%]
Propulsion	$I_{sp,vac,LP}$ [s]	Vacuum specific impulse, LREs	**0.98**	**1.60**	**-0.02**	**0.71**
Propulsion	$I_{sp,vac,SP}$ [s]	Vacuum specific impulse, SRMs	**0.31**	**-0.50**	**-0.03**	**0.27**
Propulsion	A_e [m^2]	Nozzle exhaust area (for $I_{sp}(h)$)	14.63	30.11	-0.80	15.37
Aerodynamics	C_D	Drag coefficient	9.35	81.80	4.28	9.27
Aerodynamics	C_L	Lift coefficient	10.40	98.47	9.10	14.27
Weights	$M_{inert,LP}$ [kg]	LP stages total inert mass	**8.13**	**-27.38**	**-2.94**	**8.47**
Weights	$M_{inert,SP}$ [kg]	SP stages/boosters total inert mass	**7.81**	**+16.36**	**+4.12**	**6.91**
Weights	M_{PLF} [kg]	Payload fairing mass	**6.93**	**+18.60**	**+1.33**	**8.55**

Table 38: Summary of V2 disciplinary level validation results: statistical figures of errors in the estimation of relevant outputs of each discipline: $E = mean(|e|)$, $M = max(|e|)$, $\mu = mean(e)$, $\sigma = stdev(e)$ **. In bold: values modified from V1 to V2 models.**

[48] Data for Ariane 5 stages/boosters are from internal ESA references; data for VEGA are from www.elv.it and published papers.
[49] nsc = non structural component, whose weight is not changed with respect to the weight estimation in the second column.
[50] 4989 kg is referred to Ariane 5G version, containing less propellant than 5E (lower α_P of Vulcain 1), hence may be underestimated.
[51] 390 kg is for the Liquid Helium Subsystem (SSHeL) for LOx pressurization only, evaporated H_2 for LH_2 pressurization is not included.
[52] Computed as inert mass from **[139]** minus total engine mass (nozzle+TVC+igniter), hence includes both structure and insulation.
[53] 2637.5 kg = 2249.0 kg for the nozzle (**[140]**) + 258.5 kg for the TVC system (**[150]**) + 130.0 kg for the igniter (**[139]**).

7.1.3. Trajectory

The main enhancements to the Trajectory discipline are the wind modelling, used for static controllability verification and steering losses evaluation, and the safety assessments related to the ground impact of suborbital boosters and stages. Although no relevant data for quantitative comparison with respect to actual trajectories exist, a qualitative verification of the implemented approaches was performed.

Static controllability verification is identical to V1, except for the calculation of the AoA, which accounts for a worst-case horizontal wind as already described in **Paragraph 6.4.1** and depicted in **Figure 95**. The wind definition is conservative in both intensity (sum of the maximum steady-state and wind gust at each altitude) and direction (wind coming from the back in case of the initial pitch-down manoeuvre), determining very large values of AoA especially in the first seconds of flight (balanced by the low Q_{dyn}). For steering losses evaluation, this is not realistic since the vehicle's aerodynamic loads are generally alleviated by turning it into the wind, thereby reducing α and/or β at the expense of some trajectory deviation. For controllability purposes however, this worst-case AoA is assumed to compensate for the approximate nature of the static verification approach, resulting in a much more strict controllability constraint with respect to V1. **Figure 112** shows an extract from Ariane 5 ECA's ascent, for which the control margin is negative for the first guess trajectory obtained with a fast PSO run and is then adjusted within WORHP optimization, reaching feasibility.

Re-entry safety constraints were instead introduced to avoid trajectories with massive launcher components re-entering the atmosphere on populated areas over the Earth. Qualitative verification of the re-entry and safety models highlighted how optimized VEGA trajectories do not require modifications to match safety constraints, since for polar launches all SRMs fall in the Atlantic Ocean. For Ariane 5 ECA launches to GTO on the contrary, the EPC falls on inhabited regions of central Africa for all optimal trajectories, such as those shown in **Section 5.4.1**. The application of the safety constraint allows reshaping the trajectory so that EPC's impact ellipse is moved westerly. Different runs were tried, starting the optimization both from the optimal solution of the problem without safety constraints and from new PSO initial guesses. As shown in **Figure 113**, this leads to the impact point being either in the Gulf of Guinea or west of Liberia, both feasible for the predefined population density map. If the first option is deemed acceptable, the payload loss is very limited, from 10187 kg to 10171 kg. If instead the second is preferred for safety considerations, the penalty is much larger, the payload being reduced to 9505 kg. Note that the population density map is extremely non-smooth, hence the satisfaction of the safety constraint may in several cases result in local optimality. Artificial smoothing of the population density map could lead the optimizer out of populated areas without getting trapped in locally feasible areas of the search space. Similarly, the introduction of arbitrarily modified maps to represent specific drop corridors or other regions of feasible component impacts could be exploited to enable for example launches to specific inclinations such as from Baikonour.

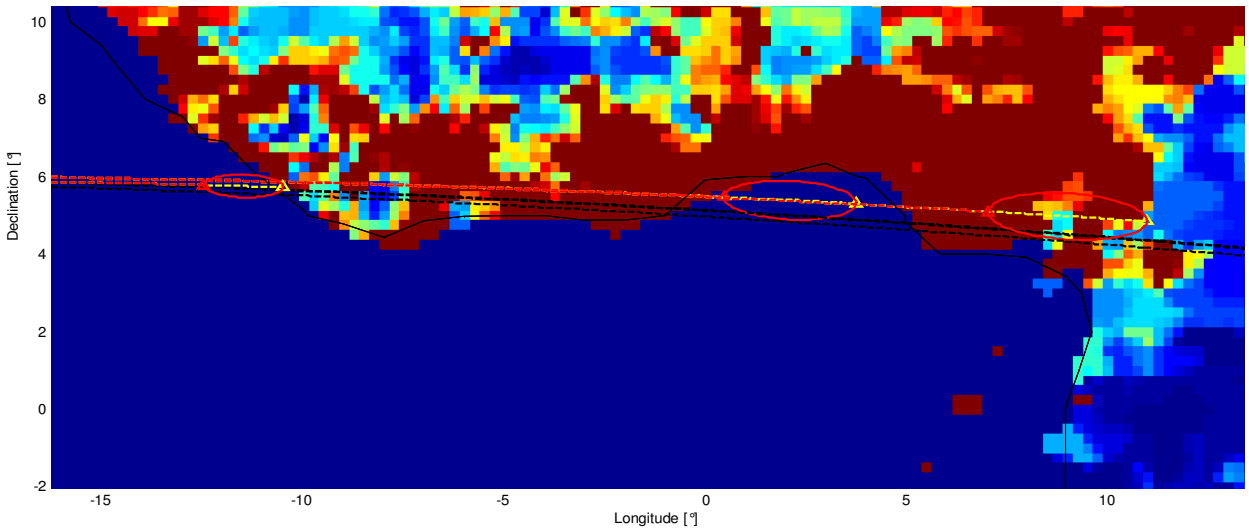


Figure 113: Ariane 5 ECA's ground track, zoom on EPC's re-entry region. The three impact ellipses are 1) no safety constraints (right), 2) active safety constraint, optimization started from solution 1) (centre), and 3) started from a PSO first guess (left).

The main purpose of the upgrades to the Trajectory discipline was the improvement of the payload performance assessment, which was shown to be optimistic with V1 models (9% and 14% overestimation of Ariane 5 ECA's and VEGA's payload). As a consequence of these trajectory models enhancements, as well as of the realistic representation of SRMs' thrust profiles, new estimates were obtained, matching very closely the reference performances: ***10187 kg*** against ***10050 kg*** from manual for Ariane 5 ECA (***1.4% error***), and ***1573 kg*** against ***1500 kg*** for VEGA (***4.8 % error***). A residual overestimation of less than ***5%*** could not be eliminated, and could be due to inaccuracies in the launchers' design data (inert masses, I_{sp}, ...) as well as to the 3-DoF nature of the model and lack of wind impact on the trajectory simulation.

7.1.4. Sensitivity analyses

As for V1, one-variable-at-a-time and Montecarlo analyses use statistical information on the disciplinary level errors (***E***, ***M***, ***μ***, ***σ*** from **Table 38**) as inputs for assessing the effect on the global performance. **Table 39** summarizes the results for the one-variable-at-a-time SA, without reporting data for A_e, C_D and C_L whose disciplinary accuracy was not affected and for which therefore the same results of **Table 21** apply. By the comparison of the results obtained for V1 and V2 models, it first can be highlighted how the sensitivities of the payload mass to variation of the inputs parameters are very similar. The only exception regards the sensitivities to the inert masses of VEGA's SRMs P80, Z23 and Z9, which decrease from [25.0 to 1, 7.0 to 1, 3.5 to 1] for V1 to [14.9 to 1, 6.3 to 1, 3.2 to 1] for V2. Although trajectory optimization robustness issues could in part be the cause for such difference, it is also realistic that since the applied ΔParameter in V2 is much smaller than in V1 (7.8% instead of 22.5%), the resulting payload sensitivity decreases. In any case, the close matching of $\partial M_{PL} / \partial Param$ between the two versions supports the consistency of the models and of the sensitivity analysis process. Moreover, from the numerical values reported in **Table 39**, a significantly lower criticality of the inert masses can be evinced with respect to V1 models. In particular, the estimated average errors on disciplinary outputs determine global PL errors of at most ***~5%*** for Structures and Weights, ***~3%*** for Propulsion and ***~1%*** for Aerodynamics.

Table 40 shows instead ***μ***, ***σ*** and expectable payload ranges from Montecarlo analyses, in comparison also with the values from manual and from fixed design trajectory optimization. By a comparison with V1 figures (**Table 22**), the spread of the PL performance for Gaussian-varying input parameters is lowered from ***~8%*** and ***~16%*** for Ariane and VEGA to ***~5%*** and ***9%***, consistently with the better 1σ accuracy of the disciplinary outputs (especially the inert masses). The resulting payload ranges are reasonably close to the actual values and both biased on the same side, for a worst-case 1σ expectable error ***~12%***. In summary, although all weight estimation models (structural components, propulsion subsystems, other non structural elements) would still benefit from enhancements such as the introduction of high fidelity methods (e.g. FEM), SA on the developed early preliminary design MDA seem to suggest that the overall performance accuracy is satisfactory in light of the very limited computational times involved (***<2 s*** for each MDA, see next Section).

Component	***Parameter***	***ΔParameter***		***ΔM_{PL}***		$\partial M_{PL} / \partial Param$
ESC-A	$M_{inert,LP}$	+529 kg	+8.1%	-544 kg	**+5.4%**	-1.028 kg/kg
Ariane PLF	$M_{inert,PLF}$	+185 kg	+6.9%	-15 kg	-0.2%	-0.082 kg/kg
EPC	$M_{inert,LP}$	+1407 kg	+8.1%	-544 kg	**+5.3%**	-0.387 kg/kg
EAP	$M_{inert,SP}$	+2507 kg	+7.8%	-298 kg	-2.9%	-0.119 kg/kg
ESC-A	$I_{sp,vac}$	+4.2 s	+1.0%	+126 kg	+1.2%	+30.1 kg/s
EPC	$I_{sp,vac}$	+4.1 s	+1.0%	+275 kg	+2.7%	+67.0 kg/s
EAP	$I_{sp,vac}$	+0.8 s	+0.3%	+29 kg	+0.3%	+36.2 kg/s
AVUM	$M_{inert,LP}$	+58 kg	+8.1%	-58 kg	**-3.6%**	-0.999 kg/kg
VEGA PLF	$M_{inert,PLF}$	+37 kg	+6.9%	-5 kg	-0.3%	-0.143 kg/kg
P80	$M_{inert,SP}$	+664 kg	+7.8%	-44 kg	-2.8%	-0.067 kg/kg
Z23	$M_{inert,SP}$	+213 kg	+7.8%	-34 kg	-2.1%	-0.158 kg/kg
Z9	$M_{inert,SP}$	+111 kg	+7.8%	-35 kg	-2.2%	-0.312 kg/kg
AVUM	$I_{sp,vac}$	+3.0 s	+1.0%	+6 kg	+0.4%	+2.0 kg/s
P80	$I_{sp,vac}$	+0.8 s	+0.3%	+7 kg	+0.4%	+9.1 kg/s
Z23	$I_{sp,vac}$	+0.8 s	+0.3%	+9 kg	+0.6%	+11.5 kg/s
Z9	$I_{sp,vac}$	+0.9 s	+0.3%	+14 kg	+0.9%	+15.1 kg/s

Table 39: Representative extracts of the results of one-variable-at-a-time SA for V2 models.

	Ariane 5 ECA		**VEGA**	
Payload mass from manual	10050 kg	-	1500 kg	-
Payload mass for fixed actual design	**10187 kg**	**+1.4%**	**1573 kg**	**+4.8%**
Payload mass distribution mean value μ_{PL}	10272 kg	+2.2%	1552 kg	+3.5%
Payload mass distribution stdev σ_{PL}	724 kg	5.3%	132 kg	8.8%
Payload mass expectable ($1\sigma_{PL}$) range	**[9548; 10996] kg**	**[-5.0; +9.4]%**	**[1420; 1684] kg**	**[-5.3; +12.3]%**

Table 40: Ariane 5 ECA's and VEGA's payload performance from manual, fixed design and Montecarlo distributions (V2).

7.2. Multi-Disciplinary Analysis for existing launchers

With the wide range of improvements developed for most disciplinary areas, MDA processes of Ariane 5 ECA and VEGA testify a significantly better capability of the early preliminary models to assess the performance of existing European launchers. MDA results are presented in **Table 41** and **Table 42** for Ariane 5 ECA and VEGA, focusing on weights, geometry and system outputs, since propulsion and aerodynamic parameters are reasonably close to those shown in

V1's validation (for OTS engines, actual propulsion system data are substituted to the computed values). The key points evinced from the analysis of the MDA outputs are the following:

- The assessment of the lengths of both Ariane 5 and VEGA stages/boosters is drastically improved, due to the better representation of the interstage sections (skirts modelling), the introduction of the under-fairing configuration, and a more accurate definition of the divergent angles in case of VEGA SRMs. In particular, the total heights of Ariane's core, Ariane's boosters and VEGA are very well predicted (errors ***<1 m***). ESC-A's length seems to be overestimated by 1.85 m, but this includes in SVAGO models the whole section from the bottom of HM-7B's nozzle up to the interface with the VEB, whereas the numerical figure from Ariane 5's manual allocates much of the section around HM-7B to the lower stage. In fact, EPC's length is underestimated with respect to this model, balancing the longer ESC-A. Moreover, since there are virtually no disengagement issues, the separation plane is assumed to be placed at the nozzle's top, therefore reducing the total length carried with the upper stage by 3.44 m. For VEGA, the total launcher's height is reduced by almost 5 m with respect to the estimates with the conceptual models. A ~0.7 m overestimation remains, possibly due to uncertainties in the motors' FF and in the nozzles' cone angle.
- The increased accuracy in the lengths assessment and the skirts modelling, allowing to better divide the structural mass among the different stages, combine with the newly developed structural analysis module to provide WBS for both Ariane 5 ECA and VEGA much closer to reality. Even though the total wet masses are actually farther from the actual value with respect to what was obtained with the conceptual models, the accuracy on the single components is largely improved. For instance, the error on ESC-A's and AVUM's inert masses is **<5%**, which is particularly important in light of the large system level influence hold by upper stages.
- As a result of the above improvements, as well as of the less optimistic performance assessment with the new Trajectory models, the payload optimization nested within the MDAs ensures an extremely good matching of the launchers' actual performance. Specifically, Ariane's and VEGA's performances are overestimated by respectively 3.6% and 5.6%, which represents a very good accuracy for the implemented engineering-level analyses. Most importantly, the offset with respect to real-world figures is on the same side for both launchers, which have a completely different architecture and design, suggesting that the developed models may ensure fair automatic trade-offs in terms of performance among different ELV configurations.
- Since cost and reliability models were not modified, the same considerations reported in **Section 5.4** apply here.

Ariane 5 ECA output parameters	**MDA**	**Actual**
Mass properties		
Payload fairing mass [kg]	2670	2675
Payload adapter mass [kg]	411	400
Upper stage inert mass [kg]	6574	6504
Lower stage inert mass [kg]	16435	17300
Boosters inert mass [kg] (each)	33348	32098
Geometry		
Payload fairing ogive length [m]	5.55	5.55
Payload fairing total length [m]	17.00	17.00
VEB length [m]	1.55	1.56
Upper stage length [m]	6.56	4.71
Lower stage length [m]	28.40	30.52
Boosters length [m]	32.76	33.59[54]
Total launcher length [m]	**53.51**	**53.79**
Reference aerodynamic area [m^2]	**37.51**	**37.51**
Global design objectives		
Payload mass to reference GTO [kg]	**10409**	**10050**
Gross Take-Off Weight [kg]	**771153**	**766383**
Total Cost per Launch [M€]	**180.5**	**173.0**
Mission Success Probability	**0.926**	**0.970**

VEGA output parameters	**MDA**	**Actual**
Mass properties		
Payload fairing mass [kg]	539	529
Payload adapter mass [kg]	54	60
1st stage inert mass [kg]	9881	8497
2nd stage inert mass [kg]	3128	2725
3rd stage inert mass [kg]	1250	1416
Upper stage inert mass [kg]	681	715
Geometry		
Payload fairing total length [m]	7.88	7.88
1st stage length [m]	10.65	10.80
2nd stage length [m]	7.87	7.55
3rd stage length [m]	4.16	3.63
Upper stage length [m], outside PLF	1.20	1.21
Upper stage length [m], total	2.02	2.04
Total launcher length [m]	**31.75**	**31.07**
Reference aerodynamic area [m^2]	**7.21**	**7.21**
Global design objectives		
Payload mass to reference GTO [kg]	**1585**	**1500**
Gross Take-Off Weight [kg]	**139789**	**138089**
Total Cost per Launch [M€]	**35.1**	**30.0**
Mission Success Probability	**0.975**	**0.980**

Table 41 (left) and Table 42 (right): Ariane-5 ECA and VEGA *output* design parameters, results from the MDA compared with the actual values. *Input* design variables (e.g. M_{prop}, ...), user parameters (e.g. D_{PLF}) and Propulsion outputs are not reported here. For the source of the actual ELVs' data, see the 4th column of the analogous tables in Section 5.4.

[54] Including nozze extending from the case (31.61 m case only)

7.3. Single-Objective MDO for existing launchers

As for V1, the "small" MDO problems for European launchers were used to verify the capabilities of the integrated engineering models and optimizers to improve the design of ELVs. Note that for V2, the trajectory load parameters q_{dyn}, n_{ax} and $q_{heat,max}$ lose significance, as the structural analysis directly uses the full trajectory data to derive the relevant load cases. For this reason, Ariane 5 ECA and VEGA MDO problems are reduced to 4 ($M_{prop,EPC}$, $M_{prop,ESC\text{-}A}$, $M_{prop,P241}$, D_{P241}) and 7 ($M_{prop,P80}$, $M_{prop,Z23}$, $M_{prop,Z9}$, $M_{prop,AVUM}$, D_{P80}, D_{Z9}, K_{UF}) design variables in addition to the ***10*** and ***14*** trajectory control variables. The numerical values of the design variables and system level outputs for the optimal design solutions obtained with global PSO runs and local WORHP refinement are reported in **Table 43** and **Table 44**. **Figure 114** provides instead an outlook of such solutions in terms of external geometry. A few remarks can be schematically drawn:

- The best solutions found for Ariane 5 ECA and VEGA after local refinement have a GTOW of respectively ***711.6 tons*** and ***134.2 tons***, showing limited improvements (***1.2%*** and ***2.1%***) obtained with WORHP over the best PSO-1D runs. The correction to be applied to the MDA's GTOW due to performance assessment errors can be obtained from the sensitivity of the payload to the total mass $\partial PL/\partial GTOW$, estimated from multi-objective min GTOW vs max PL optimization runs. These sensitivities are larger for V2 models (***22.4 kg/ton*** and ***24.9 kg/ton***) with respect to V1 (***17.0 kg/ton*** and ***18.8 kg/ton***), so that the corrected net design improvement brought by V2 MDO equals ***~5.8%*** for Ariane and ***~1.6%*** for VEGA. This confirms how VEGA's actual design appears more mass-optimal than Ariane's, as already found with V1 models. However, the introduction of the analytical structural sizing prevents from the inert masses' decrease which was the outcome of the synthetic reduction of trajectory load parameters q_{dyn}, n_{ax} and $q_{heat,max}$ within V1 models. This determines the much more limited impact of MDO on the design, with GTOW improvements mostly obtained through re-allocation of the propellant masses.
- The MDO processes with V2 models support the qualitative trends in the allocation of the propellant mass for the different stages/boosters evinced from V1 validation. For Ariane, $M_{prop,ESC\text{-}A}$ is increased up to the predefined bound of +30.% (18.7 tons[55]) whereas $M_{prop,P241}$ is only reduced by ~7%. This low reduction in SP boosters propellant is justified by the active Q_{dyn} path constraint, which prevents from further reducing M_{prop} when the thrust profile is frozen (P241 is modelled as an OTS SRM in V2, with fixed $T^*(t)$). For VEGA, $M_{prop,AVUM}$'s upper bound is also hit, while the variation of propellant in the SRMs qualitatively follows that of V1.
- Since the effect of MDO is not as large as with V1 models, the external geometry of Ariane and VEGA is after global and local optimization is not largely changed. In particular, the diameter of P241 boosters is only reduced by few cm, because the nozzle's diameter is fixed (P241 is a frozen OTS engine) and hence smaller diameters would violate the geometric interference constraints. The optimized design of VEGA is instead significantly thinner than the original design for aerodynamic advantages (approximately 60 cm longer although 5.6 tons lighter)
- Regarding costs and reliability, the considerations outlined for V1 models still apply, with the CpL being reduced together with the GTOW and the MSP almost unchanged with variations of the launcher's dimensions. The introduction of the structural sizing however determines a higher sensitivity of the inert mass to the propellant mass of upper stages. Since the upper stage's inert mass is a major driver of VEGA's cost, this causes a 0.6 M€ increase in the CpL in front of a reduction of ~5.6 tons in GTOW.

	Actual	**MDA**		**PSO optimal**		**WORHP optimal**	
$M_{prop,EPC}$ [tons]	173.3	173.3	**0**	149.1	-14.0%	145.6	-16.0%
$M_{prop,ESC\text{-}A}$ [tons]	14.4	14.4	**0**	18.7	+29.9%	18.7	+29.9%
$M_{prop,P241}$ [tons] (each)	240.1	240.1	**0**	227.4	-5.3%	224.8	-6.4%
D_{P241} [m]	3.05	3.05	**0**	3.00	-1.6%	2.99	-2.0%
$M_{inert,EPC}$ [kg]	17300	16435	-5.8 %	14777	-14.9%	14596	-16.1%
$M_{inert,ESC\text{-}A}$ [kg]	6504	6574	+1.5 %	6744	+3.1%	6564	+3.1%
$M_{inert,P241}$ [kg] (each)	32098	33348	+3.7 %	32034	-0.3%	31782	-0.9%
L_{core} [m]	53.79	53.51	-0.5 %	51.21	-4.8%	50.74	-5.7%
L_{P241} [m]	33.59	32.76	-2.5 %	32.24	-4.0%	32.20	-4.1%
PL to GTO [kg]	**10050**	**10409**	+3.6 %	**10050**	**-**	**10050**	**-**
GTOW [tons]	**766.4**	**771.2**	+0.6 %	**719.4**	-6.1%	**711.6**	-7.2%
CpL [M€]	**173.0**	**180.5**	+4.3 %	**174.8**	+1.0%	**173.6**	+0.4%
MSP []	**0.970**	**0.926**	-4.5 %	**0.930**	-4.1%	**0.931**	-4.0%

Table 43: V2 results for Ariane 5 ECA small MDO problem: optimization variables values and system level outputs from global and local minimum mass MDO, in comparison with actual launcher's design and MDA.

[55] The fact that sensible performance enhancements can be achieved with an enlargement of the cryogenic upper stage is confirmed by the design of ESC-B stage for the future Ariane 5 ME version, which assumes 28 tons of propellants.

	Actual	MDA		PSO optimal		WORHP optimal	
$M_{prop,P80}$ [tons]	87.8	87.8	-	80.7	-8.1%	79.9	-9.0%
$M_{prop,Z23}$ [tons]	23.8	23.8	-	25.6	+7.6%	26.6	+11.8%
$M_{prop,Z9}$ [tons]	10.6	10.6	-	13.0	+22.6%	10.3	-2.8%
$M_{prop,AVUM}$ [kg]	550	550	-	715	+30.0%	715	+30.0%
D_{P80} [m]	3.03	3.03	-	3.17	+4.6%	3.09	+2.0%
D_{Z9} [m]	1.90	1.90	-	1.95	+2.6%	1.80	-5.3%
K_{UF} []	0.355	0.355	-	0.426	+20.0%	0.426	+20.0%
$M_{inert,P80}$ [kg]	8497	9881	+16.3%	9524	+12.1%	9333	+9.8%
$M_{inert,Z23}$ [kg]	2725	3128	+14.8%	3288	+20.7%	3396	+24.6%
$M_{inert,Z9}$ [kg]	1416	1250	-11.7%	1422	+0.4%	1217	-14.1%
$M_{inert,AVUM}$ [kg]	715	681	-4.8%	726	+1.5%	728	+1.8%
L_{core} [m]	31.07	31.75	+2.2%	31.08	+0.0%	32.36	+4.2%
PL to GTO [kg]	**1500**	**1585**	+5.7%	**1500**	-	**1500**	-
GTOW [tons]	**138.1**	**139.8**	+1.2%	**137.1**	-0.7%	**134.2**	-2.8%
CpL [M€]	**32.0**	**35.1**	+9.7%	**36.1**	+12.8%	**35.7**	+11.6%
MSP []	**0.980**	**0.975**	-0.5%	**0.977**	-0.3%	**0.977**	-0.3%

Table 44: V2 results for VEGA small MDO problem: optimization variables values and system level outputs from global and local minimum mass MDO, in comparison with actual launcher's design and MDA.

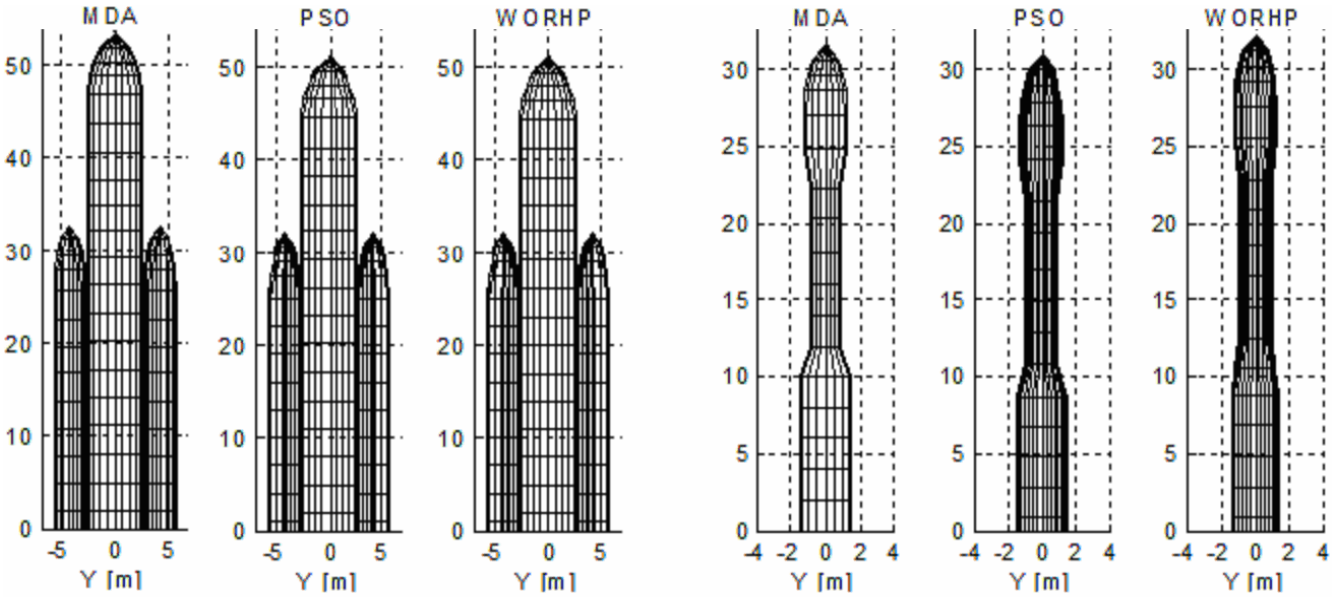


Figure 114: Geometry of Ariane 5 ECA (left) and VEGA (right) with V2 models, comparison of MDA, best solution from global PSO-1D MDO runs and from local WORHP MDO refinement.

7.4. Comparison of Conceptual and Early preliminary models

This paragraph summarizes the validation results from **Chapters 5** and **7**, with the aim of outlining a comparison of the conceptual level and early preliminary level models, on the basis of performances, cost and reliability estimates as well as of the required CPU times. Specifically, the following results are considered and reported in **Table 45** for both V1 and V2:

- Fixed Design estimated PL: obtained from trajectory optimizations with fixed launcher design data, defines the accuracy of the trajectory, guidance and control models in assessing the performance (PL mass) of a given ELV.
- Expectable 1σ PL error range: obtained from Montecarlo sensitivity analyses, defines the accuracy on the estimated performance in terms of maximum (1σ) expectable variability of the PL mass due to disciplinary modelling errors.
- MDA estimated PL: obtained from MDA processes with nested trajectory optimizations, defines the overall accuracy of the multidisciplinary models in assessing the performance (PL mass) of ELVs with given architecture and design.
- MDA estimated GTOW/CpL/MSP: also obtained from MDA processes, defines the accuracy of the multidisciplinary models in assessing the total launch mass, total cost per launch and overall mission success probability.
- MDA required CPU time: defines the required computational effort and hence computational efficiency of the MDA.

- MDO estimated GTOW/CpL/MSP: obtained from single-objective (min GTOW) small MDO problems, defines the capability of the MDO to reduce the mass of a given launcher's architecture.

From a critical analysis of this comparison, few key aspects need to be stressed. First, the two-steps modelling process allowed the definition of a wide range of enhancements, directly stemming from the validation of the conceptual level multidisciplinary design cyle. Hence, several missing functionalities were enabled and sensible improvements in accuracy were achieved in most of the disciplines. For instance, the errors on $I_{sp,vac}$ and M_{nozzle} for SP systems, on the launcher components' lengths and on M_{inert} for all ELV stages/booster types were significantly diminished. The incidental one-variable-at-a-time sensitivity analyses showed how the criticality of the weight models was definitely reduced with the introduction of the new structural sizing module. The inert masses remain the largest cause of inaccuracies, but the global performance error due to average errors on a single parameter is only ***~5%***, against ***~3%*** for propulsion performances and ***~1%*** for aerodynamic coefficients. Montecarlo runs resulted instead in narrower 1σ PL variability ranges, so that the maximum expectable error on the global performance was lowered from ***~19%*** with V1 models to ***~12%*** with V2 models. At the same time, the error on the PL mass from MDAs was shown to be only ***~4%*** and ***~6%*** for Ariane 5 ECA and VEGA, representing – especially for Ariane 5 - a relevant accuracy improvement with respect to the initial modelling. More importantly, a performance bias on the same side (i.e. optimistic) for two ELVs with radically different architectures suggests that trade-offs obtained with MDO could be entrusted. Finally, MDO processes for small ELV design problems highlighted how the effect of the optimization on the GTOW is much more limited with the enhanced models. This is however to be seen as a good sign, since the large GTOW obtained with V1 models were partially synthetic in nature, due to the exaggerated effect on the structural masses of the trajectory load parameters.

In summary, the modifications and additions introduced in the multidisciplinary design models for ELVs significantly improved their behaviour in terms of overall accuracy in performance and confidence to be placed in the design solutions and trade-offs obtained with MDO. This was obtained at a limited price in terms of computational effort: no high fidelity analyses were included, nor were other relatively computationally intensive engineering methods such as linear aerodynamics, 6 DoF trajectory, integrated Montecarlo simulations or bi-dimensional propulsive system analyses. As a consequence, computational times for both MDAs and MDOs were roughly doubled from V1 to V2 models, a figure which seems acceptable in light of the above mentioned advantages.

	Actual		V1C1 (Conceptual)		V2C1 (Early preliminary)	
	Ariane 5 ECA	VEGA	Ariane 5 ECA	VEGA	Ariane 5 ECA	VEGA
Fixed Design PL	10050 kg	1500 kg	10944 kg (+8.9%)	1715 kg (+14.3%)	10187 kg (+1.4%)	1573 kg (+4.8%)
Expectable 1σ PL error	-	-	[+4.0; +19.2]%	[-15.7; +16.1]%	[-5.0; +9.4]%	[-5.3; +12.3]%
MDA PL	10050 kg	1500 kg	12472 kg (+24.1%)	1403 kg (-6.5%)	10409 kg (+3.6%)	1585 kg (+5.6%)
MDA GTOW	766.4 tons	138.1 tons	764.8 tons (-0.2%)	139.4 tons (+0.9%)	771.2 tons (+0.6%)	139.8 tons (+1.2%)
MDA CpL	173.0 M€	32.0 M€	171.6 M€ (-0.8%)	34.8 M€ (+8.7%)	180.5 M€ (+4.3%)	35.1 M€ (+9.7%)
MDA MSP	0.966	0.980	0.927 (-4.0%)	0.975 (-0.5%)	0.926 (-4.1%)	0.977 (-0.3%)
MDA CPU time	-	-	~0.6 s	~0.5 s	~1.4 s (2 Traj-Stru iter., no re-entries)	~1.2 s
MDO GTOW	-	-	505.8 tons (-34.0% wrt actual ***-20.5%*** MDO effect)	131.7 tons (-4.6% wrt actual ***-8.9%*** MDO effect)	711.6 tons (-7.2% wrt actual ***-5.8%*** MDO effect)	134.2 tons (-2.8% wrt actual ***-1.6%*** MDO effect)
MDO CpL	-	-	137.4 M€ (-20.6% wrt actual)	33.2 M€ (+10.7% wrt actual)	173.6 M€ (+0.4% wrt actual)	35.7 M€ (11.6% wrt actual)
MDO MSP	-	-	0.926 (-4.1% wrt actual)	0.975 (-0.5% wrt actual)	0.931 (-3.6% wrt actual)	0.977 (-0.3% wrt actual)
MDO CPU time [56]	-	-	~15 h (~4h + ~1h global+local, x3)	~13 h (~3h + ~1h global+local, x3)	~32 h (~9h + ~2h global+local, x3)	~27 h (~7h + ~2h global+local, x3)

Table 45: Summary of the system level results obtained with V1C1 and V2C1 models for the 1) Montecarlo sensitivity analyses: *Expectable 1σ PL error*; 2) Trajectory optimizations with fixed launcher design: *Fixed Design PL*; 3) MDA processes: *MDA PL*, *GTOW*, *CpL*, *MSP* and *CPU time*, and 4) MDO processes: *MDO GTOW*, *CpL*, *MSP* and *CPU time*.

[56] Intended as the single-processor CPU time required for 3 global optimization runs (with any GO algorithm), each involving 50000 MDA evaluations, and the related 3 local refinement processes with WORHP.

It's just mind-blowingly awesome. I apologize, and I wish I was more articulate,
but it's hard to be articulate when your mind's blown—but in a very good way.

CHAPTER 8

APPLICATIVE CASE STUDIES

Chapters 4 to **7** focused on the engineering multidisciplinary analysis of ELVs, providing a detailed description of the two-steps modelling and validation procedure, with the aim of highlighting strengths and weaknesses of the MDA approach. Results from MDO processes were also introduced, showing the capability of the global and local algorithms to improve the design of existing European launchers. However, only small MDO problems for minimum launch mass were tackled, in order to analyze the effect of few selected design parameters.

This chapter builds upon the already presented design optimization aspects, to provide a broader overview of the MDO functionalities and of the potential of the developed MDO environment for the design of launch vehicles. First, the same small Ariane 5 and VEGA design problems are reused to compare the MDF and IDF formulations and to show examples of multi-objective performance-only MDO. Then, larger MDO problems are introduced by extending Ariane 5 and VEGA models, considering both single and multi-objective optimization. Finally, a different applicative scenario taken from the FLPP is described, showing an example of practical application of the MDO environment.

Note that all results presented in this Chapter are referred to V2 multi-disciplinary models and therefore represent the most realistic design solutions which can be obtained with the developed MDO environment.

8.1. MDF versus IDF problem formulation

MDF and IDF formulations for generic MDO problems were introduced in **Section 2.2**, including a theoretical definition, an overview of state-of-art applications and some comparative results presented by different authors. Then, their specific application to the early preliminary design of ELVs in the present research was detailed in **Section 6.1**. In numerous examples of MDF/IDF applications from recent literature, IDF was shown to provide significant advantages in terms of computational efficiency, in light of the reduced CPU time per each objective function evaluation ensured by the elimination of iterative loops. However, this cannot be considered a result of general validity: as with most optimization related issues, benefits and drawbacks of different problem formulations strongly depend on the specific problem at hand.

In order to fairly compare MDF and IDF approaches, the computational resources allocated for each single-objective min GTOW optimization run with PSO-1D were maintained constant, and the outcome in terms of achieved objective function values was compared. The small MDO problems for Ariane 5 ECA and VEGA already defined in **Chapter 8** were used for this purpose, since the relative simplicity of the model ensures an easier comparison. Note that freezing the computational resources means letting the IDF optimizations run for a larger number of iterations, since each function evaluation is faster to execute. The exact value of this "speed-up" effect of the IDF approach varies depending on the specific launcher being designed as well as on stochastic effects. In fact, the Trajectory-Structures iterative cycle, which is the cause of the longer MDF computational times, is only executed if no violations of the design constraints are found in the previous disciplines. Hence, an optimization run detecting less feasible solutions will result in a lower IDF speed-up. The average increase in computational effort per each function evaluation with MDF approach with respect to IDF was anyway assessed to be ***1.97*** for Ariane 5 and ***1.88*** for VEGA, as the mean value of 5 different runs per each case.

These IDF speed-up figures are actually rather limited with respect to other applicative scenarios, for which the number of iterations in MDF mode may be very high. In the developed MDA for ELVs on the contrary, usually 2 and at most 3 Trajectory-Structures iterations are sufficient to close the design cycle with 1% tolerance on the inert masses. Besides, in several occasions the iterative cycle is not executed at all, further decreasing the speed-up potential of IDF formulation. For this reason, the drastic computational advantage often found in literature was not highlighted in the present application, as shown in **Figure 115** for Ariane 5 and **Figure 116** for VEGA. In particular, the normalized launch mass convergence history against the normalized CPU time does not permit to draw any definitive conclusion on the advantage of one

formulation over the other. As expectable, a larger number of iterations (i.e. marks in the figures) are required to achieve design feasibility in IDF mode due to the additional compatibility constraints, especially for Ariane 5, but the faster execution partially balances this effect. A non negligible advantage of MDF formulation in terms of final objective function values can be evinced for the Ariane 5 case, but this is not confirmed on VEGA's MDO problem, for which similar convergence histories were obtained.

More tests were attempted with different settings of PSO-1D parameters (in total, 3 optimization runs per each formulation for 5 different PSO-1D settings), showing a limited advantage (***<5%***) of MDF in 8 of 10 cases, with IDF having a small edge (***<1%***) in the remaining two cases. In conclusion, MDF and IDF approaches show a similar behaviour in terms of computational efficiency on Ariane 5 and VEGA small MDO problems. Nevertheless, a limited but non negligible advantage of the former was highlighted by the available testing results. In combination with the higher robustness ensured by the built-in multidisciplinary feasibility, this suggests that MDF formulation may be better suited for the specific applicative case of ELVs' design optimization. Although more extensive testing on other launch vehicles could provide more conclusive insights, this result was considered sufficient in light of the long computational times involved in such further study, hence the MDF formulation was selected for all other MDO applications described in this chapter.

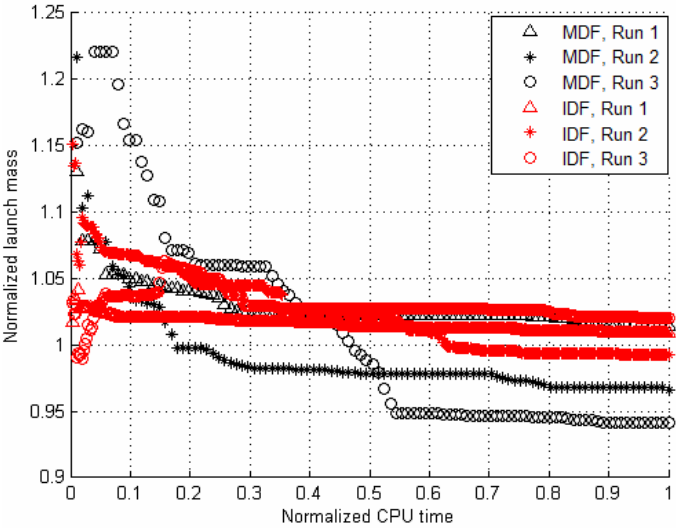


Figure 115: Normalized objective vs CPU time convergence with PSO-1D algorithm for Ariane 5 ECA small MDO problem, comparison of MDF and IDF problem formulations.

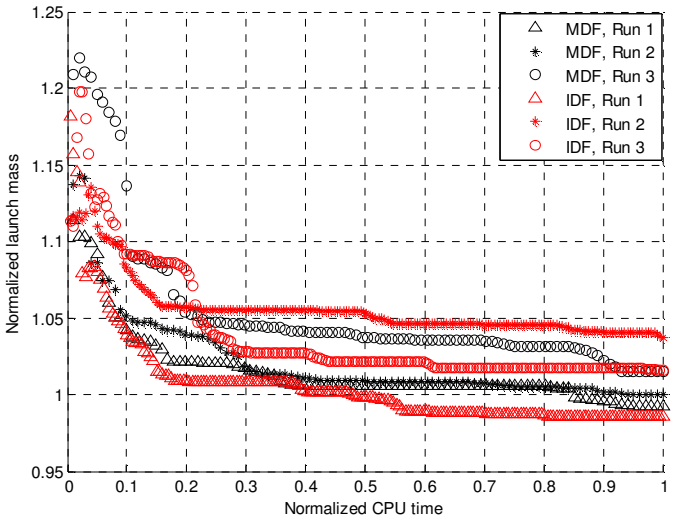


Figure 116: Normalized objective vs CPU time convergence with PSO-1D algorithm for VEGA small MDO problem, comparison of MDF and IDF problem formulations

8.2. Minimum launch mass vs maximum payload mass optimization

Multi-objective MDO for min GTOW vs. max PL was already introduced for the conceptual and early preliminary MDO validation, as a mean to obtain an approximate evaluation of the derivative $\partial PL/\partial GTOW$, which represents the effect of varying the payload requirement on the total mass of the system. Within the validation procedure, linear regressions of $\partial PL/\partial GTOW$ were used to scale the GTOW obtained from the MDA in order to derive the actual design improvement brought by minimum mass single-objective MDO processes. This type of multi-objective optimization is however extremely useful also in real world design practices: although a reference PL mass is one of the key design requirements, it is extremely useful to understand the sensitivity of the launch vehicle being designed to variations of such target. For this purpose, MDO appears a perfect approach, since it permits the simultaneous optimization of multiple solutions of different size, eventually achieving a Pareto front including design information encompassing a wide range of payload requirements.

To give an impression of the quality of the Pareto fronts which can be achieved with the developed MDO environment, **Figure 117** and **Figure 118** show the non dominated solutions obtained for the small MDO problems of Ariane 5 ECA and VEGA. After some preliminary testing on different ELVs' MDO cases, it was recognized that DG-MOPSO provides generally better Pareto fronts with respect to the other multi-objective algorithms, in terms of both convergence and coverage. For this reason, and since the focus of the present research is on the engineering aspects rather than on the optimization methodologies, all multi-objective results shown in the continuation of this Chapter are referred to three

different runs of DG-MOPSO. Although at least 5 or maybe 10 runs would allow for a better evaluation of the stochastic effects, the execution of 3 runs was deemed sufficient in light of the long computational times involved[57].

Since both objectives are purely performance based and the optimal solutions for minimum launch mass and maximum payload are characterized by similar designs scaled to different sizes, the small MDO problems are large enough to provide good Pareto fronts, as shown in **Figure 117** and **Figure 118**. In each run, a swarm of 500 particles was allowed to evolve for 50 iterations, initializing one of the particles with the best single-objective solution found (including local refinement, see **Section 7.3**), which is highlighted in the figures as (PSO-1D+WORHP) solution. This allows obtaining non dominated fronts from the three runs very close to each other, although the different particles spread differently on the overall Pareto front. It is however to be remarked that the entire Pareto front could not be identified with this set-up, since the input payload range was [6-14] tons for Ariane and [1-2] tons for VEGA.

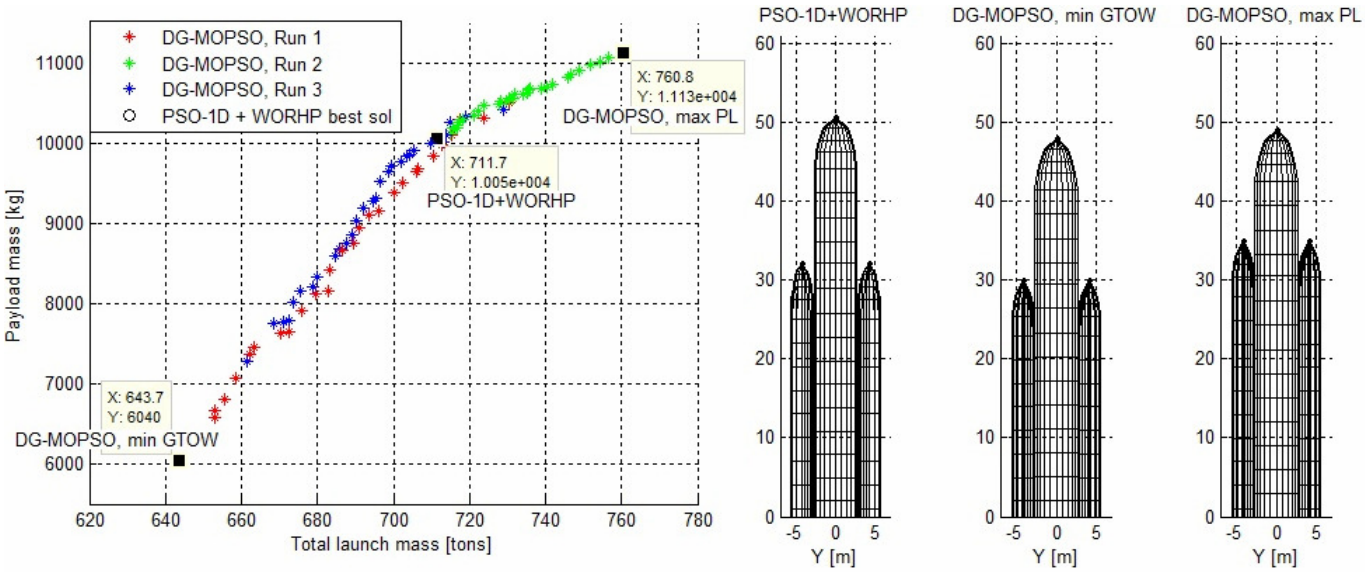


Figure 117: Small Ariane 5 ECA MDO problem: min GTOW vs. max PL Pareto fronts from 3 DG-MOPSO runs and related external geometries, in comparison with the single-objective min GTOW solution identified with the label "PSO-1D+WORHP".

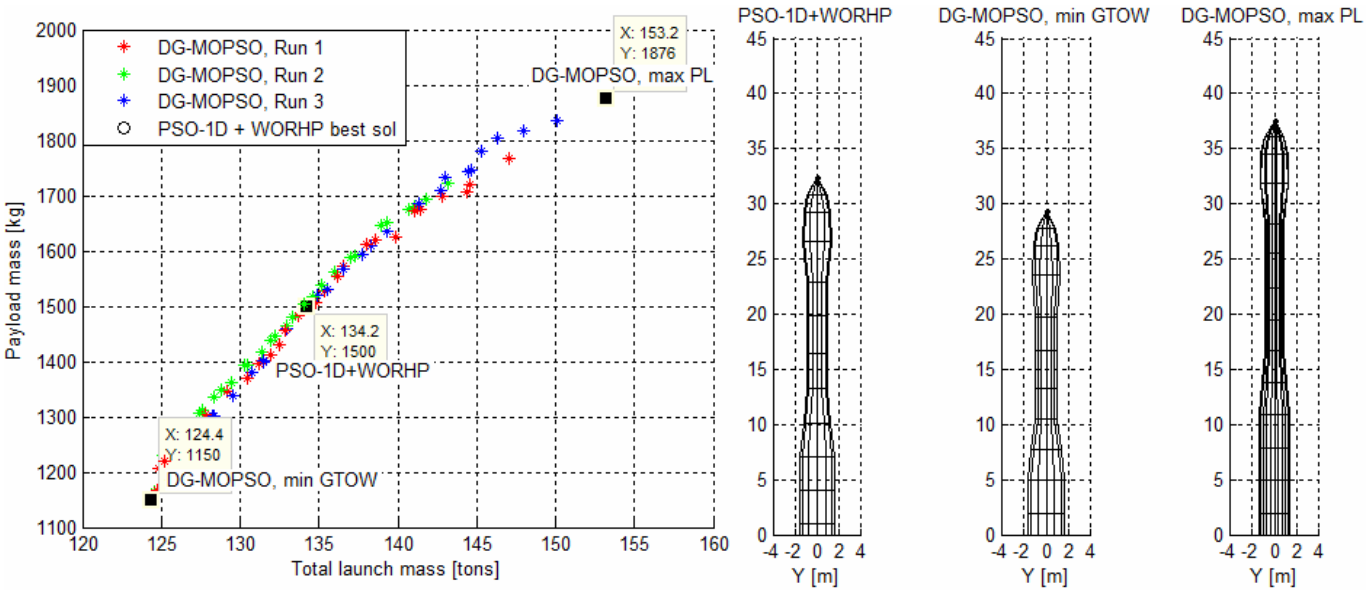


Figure 118: Small VEGA MDO problem: min GTOW vs. max PL Pareto fronts from 3 DG-MOPSO runs and related external geometries, in comparison with the single-objective min GTOW solution identified with the label "PSO-1D+WORHP".

[57] ***<10 hours*** per each run involving 50000 function evaluations (e.g. 100 particles for 500 iterations) on a single processor, reduced to ***<5 hours*** with parallel OpenMP implementation for the exploitation of two cores.

8.3. Ariane 5 ECA and VEGA large MDO problems

The "small" MDO problems for Ariane 5 ECA and VEGA allowed verification of the behaviour of the MDO process through simple optimization runs, requiring relatively short computational times. However, only the staging characteristics (i.e. propellant masses for each stage and boosters set) and some basic geometric parameters were optimized, therefore resulting in limited design changes brought by MDO. Moreover, since the launcher's architecture, the entire propulsion systems for all stages/boosters as well as the other technologies are completely frozen in the small MDO problems, it is not possible to identify Pareto fronts including non performance objectives such as CpL or MSP.

In order to assess the capability of further improving the performance of European ELVs and to attempt to identify performance vs cost Pareto fronts, *"large"* MDO problems were defined for Ariane 5 ECA and VEGA, by widening the bounds of some optimization variables and unlocking the design of the propulsion systems of each stage and booster. The main characteristics of the new MDO set-up for Ariane 5 ECA and VEGA are the following:

- All discrete optimization variables are maintained frozen, so that the launcher's architecture, propulsion technologies and structural lay-out are unchanged with respect to the actual design.
- M_{prop} ranges for ESC-A and AVUM are extended up to ***+50%***, whereas ***±30%*** is kept for all others.
- The length-over-diameter ratio of the PLFs $(L/D)_{nose}$ is allowed to vary in a ***±50%*** range.
- LP tanks pressure p_{tanks} and pressurization gas tank pressure p_{press} are added to the optimization problem, respectively in ***±50%*** and ***±30%*** ranges.
- OTS LREs are substituted by new design LP engines, with all discrete variables frozen to the OTS values. Nominal thrust T^*, chamber pressure p_{cc}, expansion ratio ε and mixture ratio α_P are optimized in ***±30%*** ranges.
- OTS SRMs are substituted by new design SP engines, with all discrete variables frozen to the OTS values. Nominal thrust T^*, chamber pressure p_{cc} and expansion ratio ε are optimized in ***±30%*** ranges, with normalized thrust and I_{sp} profiles, ***SF***, ***FF*** and ***WF*** assumed to be identical to that of the original SRMs (SRM_{type}***=custom***).
- Contraction ratio A_{cc}/A_t, nozzle's divergent angle δ_{div} and TVC angle δ_{TVC} are maintained constant at the values specified for the OTS engines, since their influence on the global performance is rather limited.

In total, ***20*** and ***23*** continuous design optimization variables for Ariane and VEGA problems result from the above settings, to be added to the customary ***10*** and ***14*** optimizable parameters for the ascent trajectories. In the next paragraphs, results are first presented for the single-objective minimum GTOW optimization in comparison with those obtained for the "small" MDO problem, then some examples of multi-objective optimization are reported.

8.3.1. Ariane 5 ECA, minimum GTOW single-objective MDO

The minimum launch mass design solutions for Ariane 5 ECA obtained with PSO-1D and with a subsequent WORHP refinement show a much lower GTOW with respect to the case of the small MDO problem. In fact, ***658.4 tons*** from the global runs and ***625.2 tons*** from local refinements represent a sensible improvement over the 711.6 tons reported in **Section 7.3**. Considering the correction for $\partial PL/\partial GTOW$, PSO-1D and WORHP optimization processes reduce the total launch mass by ***~17%*** with respect to the original design. The significant design enhancement comes from several modifications (see **Table 46** and **Figure 119** for numerical details and external geometry), which can be summarized in few key points:

- *Modification of the upper stage's HM-7B engine*: p_{cc} and T^* are increased from ***37*** to ***50 bars*** and from ***63*** to ***82 kN***, maintaining similar ε and α_P and compensating for the larger chamber pressure with an augmented pressurization gas pressure (from ***226*** to ***278 bars***). This determines a larger inert propulsion system mass (from 364 kg to 406 kg) but with the twofold advantage of enhanced I_{sp} and lower gravity losses in the upper stage's flight, therefore providing a sensible net performance gain.
- *Reduction of the P241 boosters' size*: since T^* is allowed to be scaled within the large MDO problem, the boosters' size can be drastically reduced without excessively increasing the trajectory loads (as was the case within the small MDO problem). Hence, M_{prop} and T^* are both lowered by approximately 30%, constituting the largest GTOW saving. The nozzle's expansion ratio is also reduced from 11.3 to 8.4, raising $I_{sp,sea}$ due to lower jettison altitude.
- *Stretching of the P241 boosters' size*: the diameter of the boosters is pushed to its lower bound of ***2.44 m*** (from the original ***3.05 m***), with a resulting increase in length to ***37.5 m***. Although at first sight this may be ascribed to aerodynamic reasons, the real cause is actually structural: the compression loads on the launcher's core are relieved by the tensile force transferred by the boosters, so that the entire lower composite in the original design (MDA) is sized for tensile strength. In the optimized design, this positive effect is also passed on to the upper stage, since the boosters are connected only ***10 cm*** below the PLF's jettison section (i.e. the boosters-core geometric interference constraint is active). The structural mass of ESC-A's skirts and of the VEB thus turns out to be much lower with respect to the original design, with almost ***1.2 tons*** saved on M_{inert} in spite of the ***2.4 tons*** higher propellant loading.

- *Stretching of the core's size*: although $M_{prop,EPC}$ is lowered by 27 tons and α_P is increased for propulsive performance reasons, the total length of the core is almost the same. This is due to a stretching of both the upper stage (more propellant, longer nozzle) and especially of the PLF's nose, which is pushed almost to the upper bound mainly for aerodynamic reasons (an almost 10% decrease in the average C_D is registered, in front of a limited $M_{inert,PLF}$ penalty).

	MDA	Small MDO	Large MDO bounds [LB; UB]		Large MDO PSO-1D		Large MDO WORHP	
$M_{prop,EPC}$ [tons]	173.3	145.6	121.3	225.3	152.5	-12.0%	146.6	-15.4%
$M_{prop,ESC-A}$ [tons]	14.4	18.7	14.4	21.6	16.5	+14.6%	16.8	+16.7%
$M_{prop,P241}$ [tons]	240.1	224.8	168.1	312.2	201.4	-16.1%	189.4	-21.1%
D_{P241} [m]	3.05	2.99	2.44	3.66	2.51	-17.7%	**2.44**	-20.0%
$L/D_{PLF,nose}$ []	1.028	-	0.514	1.542	1.382	+34.4%	1.431	+39.2%
T^*_{EPC} [kN]	1293.3	-	905.3	1681.3	1283.1	-0.8%	1224.9	-5.3%
$\alpha_{P,EPC}$ []	6.1	-	5.62	6.58	**6.58**	+7.9%	6.30	+3.3%
ε_{EPC} []	59.5	-	30.0	70.0	45.2	-24.0%	46.5	-21.8%
$p_{cc,EPC}$ [bars]	115.0	-	89.5	120.0	104.8	-8.9%	109.8	-4.5%
$p_{tanks,EPC}$ [bars]	3.0	-	2.2	3.5	**2.2**	-26.7%	**2.2**	-26.7%
$p_{press,EPC}$ [bars]	23.0	-	16.1	29.9	**29.9**	+30.0%	**29.9**	+30.0%
T^*_{ESC-A} [kN]	63.0	-	31.5	94.5	69.4	+10.2%	82.3	+30.6%
$\alpha_{P,ESC-A}$ []	5.0	-	4.85	5.15	4.89	-2.2%	4.90	-2.0%
ε_{ESC-A} []	83.1	-	58.2	108.0	81.3	-2.2%	83.8	+0.8%
$p_{cc,ESC-A}$ [bars]	37.0	-	30.0	50.0	48.5	+31.1%	**50.0**	+35.1%
$p_{tanks,ESC-A}$ [bars]	3.0	-	2.2	3.5	2.9	-3.3%	2.9	-3.3%
$p_{press,ESC-A}$ [bars]	226.0	-	158.2	293.8	268.7	+18.9%	278.2	+23.1%
T^*_{P241} [kN]	6395.0	-	4476.5	8313.5	5096.7	-20.3%	4874.3	-23.8%
ε_{P241} []	11.0	-	7.7	14.3	8.5	-22.7%	8.4	-23.6%
$p_{cc,P241}$ [bars]	64.0	-	56.8	71.2	63.1	-1.4%	61.6	-3.8%
$M_{inert,EPC}$ [tons]	16435	14596	-	-	13712	-16.6%	13473	-18.0%
$M_{inert,ESC-A}$ [kg]	6574	6564	-	-	5278	-19.7%	5391	-18.0%
$M_{inert,P241}$ [kg]	33348	31782	-	-	27111	-18.7%	25379	-23.9%
$M_{inert,PLF}$ [kg]	2675	-	-	-	2902	+8.5%	2935	+9.7%
L_{core} [m]	53.51	50.74	-	-	53.01	-0.9%	53.10	-0.8%
$L_{PLF,nose}$ [m]	5.55	-	-	-	7.46	+34.4%	7.73	+39.3%
L_{P241} [m]	32.76	32.20	-	-	37.81	+15.4%	37.50	+14.5%
$I_{sp,sea,EPC}$ [s]	317.8	-	-	-	322.7	+1.5%	327.9	+3.2%
$I_{sp,vac,EPC}$ [s]	431.0	-	-	-	421.4	-2.2%	425.7	-1.2%
$I_{sp,sea,P241}$ [s]	247.3	-	-	-	248.9	+0.6%	248.4	+0.4%
$I_{sp,vac,P241}$ [s]	275.3	-	-	-	270.7	-1.7%	270.4	-1.8%
$I_{sp,vac,ESC-A}$ [s]	446.0	-	-	-	451.8	+1.3%	452.3	+1.4%
max q_{dyn} [kPa]	38.6	44.1	-	-	35.9	-7.0%	38.1	-1.3%
max n_{ax} [g]	3.74	4.15	-	-	3.57	-4.5%	3.60	-3.7%
GTOW [tons]	**771.2**	**711.6**	-	-	**658.4**	-14.6%	**625.2**	-18.9%
CpL [M€]	**180.5**	**173.6**	-	-	**158.9**	-12.0%	**155.8**	-13.7%
MSP []	**0.926**	**0.931**	-	-	**0.900**	-2.7%	**0.901**	-2.7%

Table 46: Comparison of MDA, small MDO and large MDO design solutions for Ariane 5 ECA. In bold: optimization variables which have hit either the lower or upper bound.

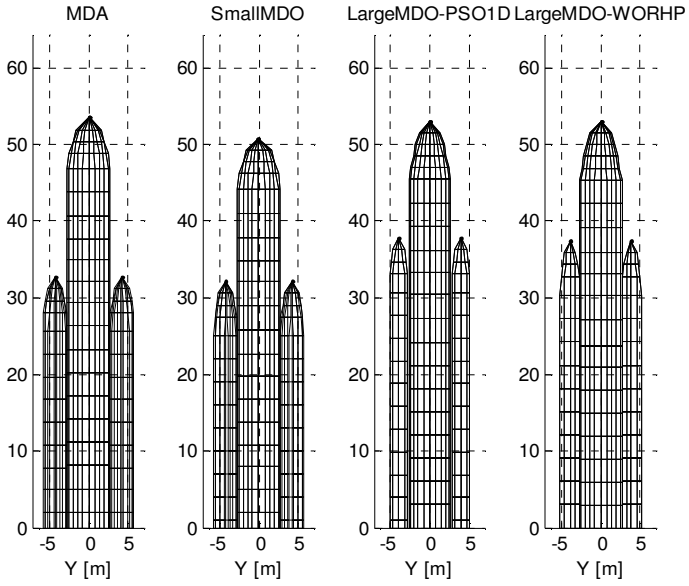


Figure 119: Geometry of Ariane 5 ECA from a) MDA, b) best sol. for small MDO problem, c) best PSO-1D sol. for large MDO problem and d) best WORHP sol. for large MDO problem

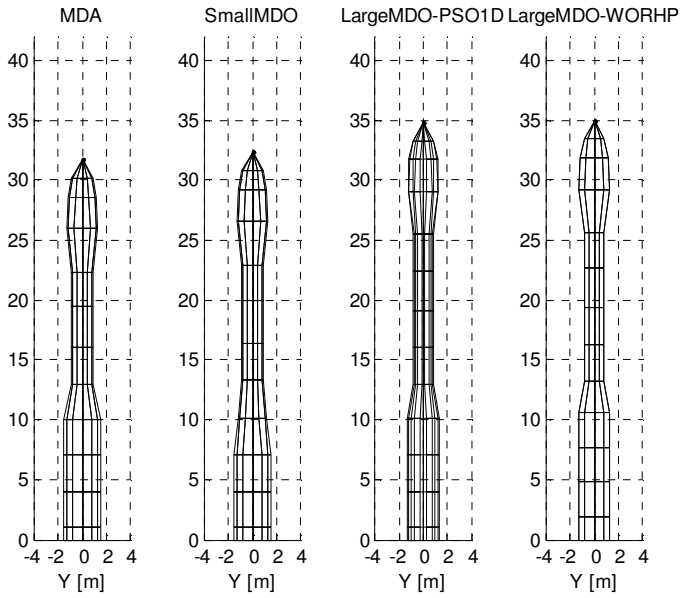


Figure 120: Geometry of VEGA from a) MDA, b) best solution for small MDO problem, c) best PSO-1D solution for large MDO problem and d) best WORHP solution for large MDO problem.

8.3.2. VEGA, minimum GTOW single-objective MDO

As for Ariane, the enlargement of the MDO problem determines a relevant increase in the GTOW saving brought by the MDO process, with a minimum launch mass of ***125.1 tons*** from the global runs further reduced to ***122.8 tons*** by WORHP refinement. This corresponds to a corrected ***10.0%*** design enhancement contributed by MDO, in front of the mere ***1.6%*** of the small MDO problem. **Table 47** reports the numerical figures for the optimal solutions found and **Figure 120** presents the resulting external geometries.

The most important design changes implemented by the MDO process for VEGA are the following:

- The optimization definitely favours a strong relief of the structural loads, both in terms of SRMs internal pressures and flight loads. Specifically, the internal pressures of P80, Z23 and Z9 are reduced by ***7%***, ***19%*** and ***16%*** whereas lower nominal thrusts (***-18%***, ***-6%***, ***-12%***) contribute to a nominal trajectory with $Q_{dyn,max}$***=44.5 kPa*** and $n_{ax,max}$***=4.23 g*** against ***53.6 kPa*** and ***4.56 g*** of the original design. The combination of relieved internal and external structural loads determines very significant structural mass savings on all the stages, which offsets the lower specific impulses. Note however that the advantage of reduced chamber pressures in the SRMs should be verified with higher fidelity analyses, as it may very well be artificial in nature: for instance, even limited errors in the sensitivities of case mass and specific impulse to p_{cc} can easily be the cause of the bias towards lower values if combined together.
- With all inert masses decreased, the propellant loadings are sensibly reduced for P80 and Z23 stages, which produce the larger GTOW savings, whereas Z9 SRM remains of the same size and AVUM's tanks are enlarged, though not until the upper bound.
- The diameters of P80 and Z9 are both lowered by almost ***15%***, which turns out in a longer though lighter launch vehicle (***35.03 m*** instead of ***31.75 m***). The main cause for this behaviour again appears to be aerodynamic in nature, suggesting that controllability issues may still be underestimated even with the model enhancements in V2.

A last remark applicable to both Ariane and VEGA regards costs and reliability. As already highlighted for the small MDO problems, a lower CpL is a side effect of the GTOW minimization for both launchers. However, the MSPs of both Ariane and VEGA are drastically cut with the large problem. This is a synthetic result: the failure rates of HM-7B and RD-869 are known and are loaded from database if defined as OTS. However, in the enlarged MDO problem, modifications to these engines are allowed through the definition as new LRE, which causes the failure rates to be computed from component build-up methods, which appear particularly pessimistic.

	MDA	Small MDO	Large MDO bounds [LB; UB]		Large MDO PSO-1D		Large MDO WORHP	
$M_{prop,P80}$ [tons]	87.8	79.9	61.4	114.1	78.4	-10.7%	76.7	-12.6%
$M_{prop,Z23}$ [tons]	23.8	26.6	16.7	31.0	20.4	-14.3%	20.3	-14.7%
$M_{prop,Z9}$ [tons]	10.6	10.3	7.4	13.7	11.3	+6.6%	10.9	+2.8%
$M_{prop,AVUM}$ [kg]	550	715	550	825	611	+11.1%	629	+14.4%
D_{P80} [m]	3.03	3.09	2.42	3.64	2.69	-11.2%	2.60	-14.2%
D_{Z9} [m]	1.90	1.80	1.52	2.28	1.64	-13.7%	1.65	-13.2%
K_{UF} []	0.355	0.426	0.284	0.426	0.371	+4.5%	0.376	+5.9%
$L/D_{PLF,nose}$ []	1.400	-	0.700	2.100	1.387	-0.9%	1.423	+1.6%
T^*_{P80} [kN]	2787.1	-	1950.9	3623.2	2309.9	-17.1%	2273.6	-18.4%
ε_{P80} []	16.0	-	11.2	20.8	16.0	+0.0%	15.6	-2.5%
$p_{cc,P80}$ [bars]	95.0	-	78.5	111.5	87.3	-8.1%	88.6	-6.7%
T^*_{Z23} [kN]	1048.6	-	734.0	1363.2	962.2	-8.2%	990.3	-5.6%
ε_{Z23} []	25.0	-	17.5	32.5	23.7	-5.2%	23.8	-4.8%
$p_{cc,Z23}$ [bars]	106.0	-	86.2	120.0	**86.2**	-18.7%	**86.2**	-18.7%
T^*_{Z9} [kN]	300.1	-	210.1	390.1	280.0	-6.7%	265.3	-11.6%
ε_{Z9} []	56.0	-	39.2	72.8	50.8	-9.3%	50.0	-10.7%
$p_{cc,Z9}$ [bars]	83.0	-	70.1	95.9	**70.1**	-15.5%	**70.1**	-15.5%
T^*_{AVUM} [N]	2385	-	1193	3578	1788	-25.0%	1790	-24.9%
$\alpha_{P,AVUM}$ []	2.00	-	1.85	2.15	1.97	-1.5%	1.99	-0.5%
ε_{AVUM} []	81.5	-	57.1	106.0	79.8	-2.1%	82.6	+1.3%
$p_{cc,AVUM}$ [bars]	20.0	-	15.7	22.4	17.4	-13.0%	16.8	-16.0%
$p_{tanks,AVUM}$ [bars]	35.6	-	28.8	40.0	31.1	-12.6%	30.0	-15.7%
$p_{press,AVUM}$ [bars]	310.0	-	217.0	403.0	255.9	-17.5%	259.8	-16.2%
$M_{inert,P80}$ [kg]	9882	9333	-	-	8264	-16.4%	8128	-17.7%
$M_{inert,Z23}$ [kg]	3128	3396	-	-	2335	-25.4%	2352	-24.8%
$M_{inert,Z9}$ [kg]	1250	1217	-	-	1144	-8.5%	1101	-11.9%
$M_{inert,AVUM}$ [kg]	681	728	-	-	603	-11.5%	602	-11.6%
M_{PLF} [kg]	529	-	-	-	534	+0.9%	541	+2.3%
$I_{sp,sea,P80}$ [s]	251.1	-	-	-	249.5	-0.6%	250.3	-0.3%
$I_{sp,vac,P80}$ [s]	280.1	-	-	-	279.2	-0.3%	278.9	-0.4%
$I_{sp,sea,Z23}$ [s]	243.4	-	-	-	242.6	-0.3%	242.5	-0.4%
$I_{sp,vac,Z23}$ [s]	287.5	-	-	-	287.2	-0.1%	287.2	-0.1%
$I_{sp,vac,Z9}$ [s]	294.4	-	-	-	293.2	-0.4%	293.2	-0.4%
$I_{sp,vac,AVUM}$ [s]	314.7	-	-	-	319.7	+1.6%	320.0	+1.7%
L_{tot} [m]	31.75	32.36	-	-	34.81	+9.6%	35.03	+10.3%
$L_{PLF,nose}$ [m]	3.64	-	-	-	3.61	-0.8%	3.70	+1.6%
max Q_{dyn} [kPa]	53.6	53.6	-	-	44.3	-17.4%	44.5	-17.0%
max n_{ax} [g]	4.56	4.53	-	-	4.41	-3.3%	4.23	-7.2%
GTOW [tons]	**139.8**	**134.2**	-	-	**125.1**	-10.5%	**122.8**	-12.2%
CpL [M€]	**35.1**	**35.7**	-	-	**33.6**	-4.3%	**33.2**	-5.4%
MSP []	**0.977**	**0.977**	-	-	**0.940**	-3.8%	**0.939**	-3.9%

Table 47: Comparison of MDA, small MDO and large MDO design solutions for VEGA. In bold: optimization variables which have hit either the lower or upper bound.

8.3.3. Multi-objective MDO

Although multi-objective optimization for purely performance objectives was shown to be extremely effective, significant difficulties were encountered when trying to optimize the design of Ariane 5 and VEGA launch vehicles for cost and reliability besides launch mass. In particular, even with the large MDO problem described in this section, it is not possible to identify performance versus reliability trade-offs, since the mission success probability is almost insensitive to variations of the continuous design variables but only to different launcher architectures or different technologies (especially propulsive) employed in the different stages and boosters. For this reason, only cost versus performance trade-offs are presented here.

Figure 121 and **Figure 122** show the best Pareto fronts achieved with three runs of DG-MOPSO of 100000 function evaluations each. Note how the Pareto fronts are extremely narrow, only encompassing about 1.6 tons and 0.3 M€ for Ariane and about 0.3 tons and 0.1 M€ for VEGA. This is a consequence of the strong dependence of the assessed CpL from the inert masses of the different stages, which is also proportional to the total mass at launch. Although technological factors also influence the cost of the different components (e.g. a higher structural efficiency or larger engine's thrust result in a

higher technical quality factor multiplying the mass-based CER), this is not sufficient – at least for the two examined cases - to obtain relevant differences among the Pareto solutions. Nevertheless, sound trade-offs in certain design variables can be identified. For instance, the minimum mass and minimum cost solutions for Ariane 5 case are mostly identical, except for a much lower thrust of the first stage engine (1.1 MN against 1.3 MN in vacuum), which reduces the performance but ensures a lower engine's development and production cost. For VEGA instead, the propellant loading of the first stage with respect to the second and third constitutes the main difference among the different solutions. In fact, the cost model for both development and production of SRMs includes two different CERs, one provided by ESA for small strap-on boosters covering Z23 and Z9 and one for larger booster motors directly obtained from TRANSCOST. In order to continuously connect the two curves of development or production costs as a function of the loaded propellant mass, the CER for large SRMs was modified as in **Figure 123**. Since the resulting cost model presents a slope discontinuity when shifting from one CER to the other (for M_{prop}=40 tons), minimum cost solutions for VEGA tend to favour a smaller P80 with respect to minimum cost solutions, because the cost of Z23 and Z9 stages is evaluated with the "cheaper" small motors CER.

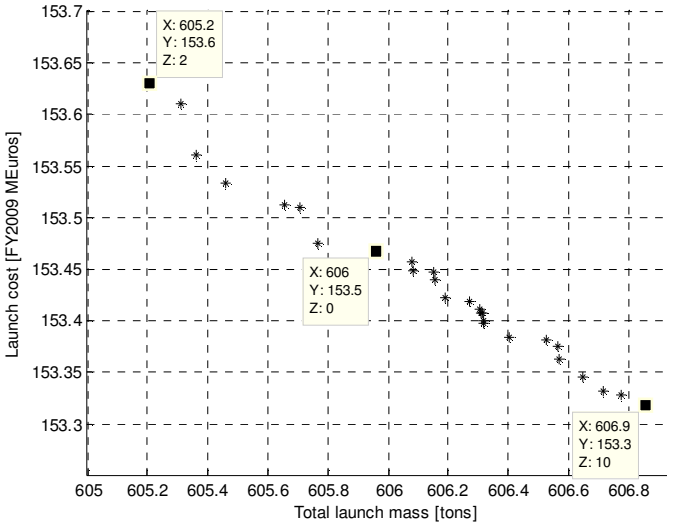


Figure 121: Large MDO problem for Ariane 5 ECA, multi-objective GTOW vs. CpL Pareto front.

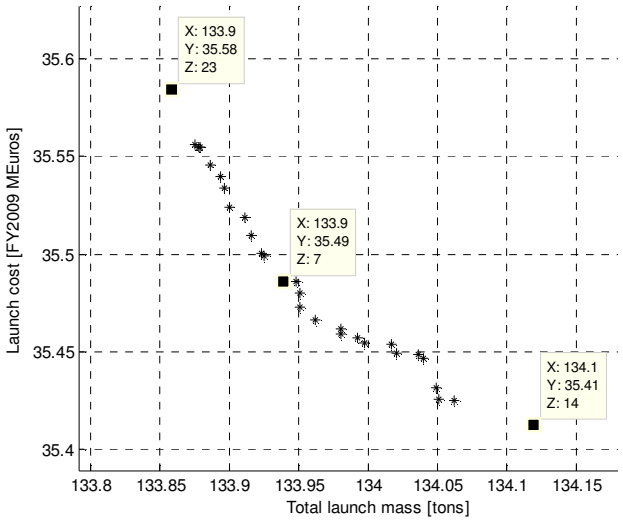


Figure 122: Large MDO problem for VEGA, multi-objective GTOW vs. CpL Pareto front.

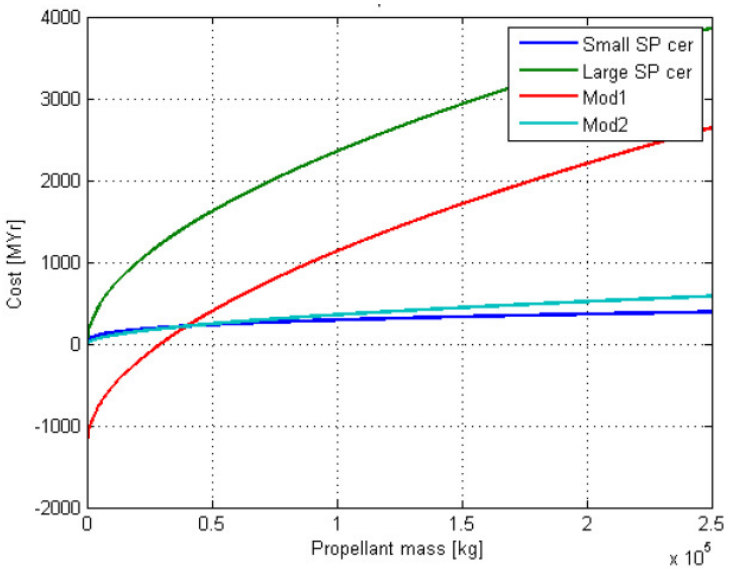


Figure 123: SRMs development cost models: original CERs for small and large motors, plus modified curves to match the two profiles. Selected CER is the blue curve for small motors up to 40 tons of propellant and the red curve Mod1 for larger systems.

Another issue related to the cost (and reliability) assessments is related to the tuning of the model parameters. In fact, virtually all CERs and failure rate relations are calibrated with empirical correction factors which depend on the set of data points used for the correlation, and additional coefficients are employed to describe specific effects on cost/reliability of technological trade-offs or other design choices. As a consequence, performance versus cost versus reliability Pareto fronts

are strongly dependent on these empirical parameters. To exemplify this behaviour, the same MDO process for Ariane 5 ECA was repeated after re-tuning the CER for large SRMs, artificially reducing by 50% both development and production costs for P241 boosters. The results of the new MDO runs are shown in **Figure 124** and **Figure 125** in terms of Pareto fronts and external geometries of four representative Pareto solutions. It is clear how modifying the CER radically changes the shape and extent of the non dominated front, now spanning 45 tons and 3 M€ from the minimum mass to the minimum cost solutions. This is mostly the effect of the halved cost for the SP systems, which is exploited by the optimizer to achieve a min CpL design with minimum propellant in both cryogenic stages (121 and 14 tons), at the expense of a larger quantity of SP in the boosters (218 tons). This is sided by modified thrust levels, with the core stage's engine reduced to less than 1 MN in vacuum, with associated lower costs, compensated by higher SRMs thrust. Note that this modification is reflected in the external geometry of the vehicles, with the boosters for the minimum cost solution being larger in volume however shorter in length due to a lower slenderness with respect to the minimum mass design.

Although this kind of trade-off between the dimensions of cryogenic and solid propellant systems is sound in a performance vs. cost perspective, the numerical results are questionable, being dependent on the tuning of the cost parameters. In real-world design exercises, design trends identified with such approach could still be useful, but verification of actual CpL advantages should be externally verified with bottom up cost estimation methodologies. Further research is therefore certainly necessary in this area, though the development of thrust-worthy cost and reliability models to be used in MDO for early design activities appears extremely challenging, due both to the lack of detailed cost information on existing systems and to the inherent limitations of historical-based approaches.

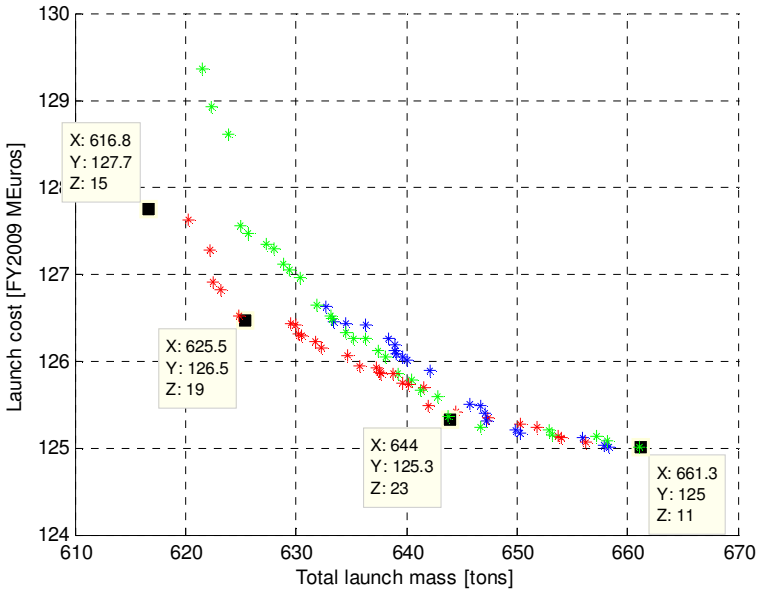


Figure 124: Large MDO problem for Ariane 5 with modified cost model., multi-objective GTOW vs. CpL Pareto front.

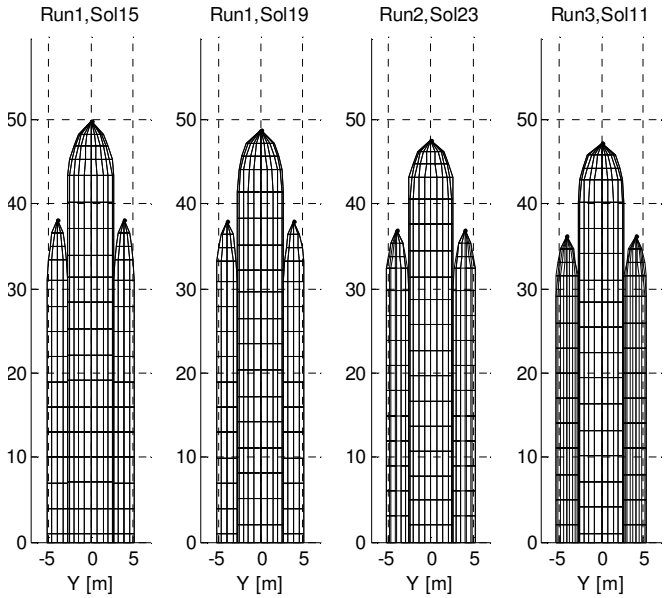


Figure 125: External geometry for the four solutions shown in Figure 124, where Z identifies the solution's ID.

8.4. MDO of free-architecture ELV

In order to provide an impression of the capabilities of the MDO environment for search space exploration of arbitrary launcher configurations, an example of global optimization for the same GTO mission of Ariane 5 ECA but without constraining the design of the stages and boosters to specific architectures is provided. Several architectural and technological trade-offs were unlocked for this test:

- Linear configuration or parallel configuration with 2, 3 or 4 boosters of the same design.
- Core stage propulsion with 1, 2, 3 or 4 engines, either Vulcain-2 or of new cryogenic SC or GG design.
- Upper stage propulsion with 1 or 2 OTS engines, with selection of one of the following EC LREs: European Vinci, US RL-10B or Russian RD-0146.
- Variable diameter, possibly up to three different diameters (i.e. VEGA-like), with non-under fairing upper stage.
- Upper stage tanks configuration (enclosed tanks or common bulkhead) and structural materials (Al 7075 or Al-Li 2195 for the tanks, Al-7075 or CFRP for the interstage, intertank and thrust frame structures).
- Solid rocket boosters structural material (Steel or CFRP for the case, Al 7075 or CFRP for the other structures).
- Pad interface structure with the core or with the boosters.

The resulting MDO problem has in total ***45*** optimization variables: 4 for the system level design, 16 for the lower stage, 8 for the upper stage, 7 for the boosters and 10 for the trajectory. Although several single and multi-objective MDO runs were attempted, the min GTOW versus max MSP results are presented here, as an example of the introduction of reliability in multi-objective optimal design. The Pareto fronts obtained in three runs of 200000 function evaluations each are shown in **Figure 126**, with the external geometry, reliability profile and design data reported respectively in **Figure 127**, **Figure 128** and **Table 48**. Whereas one of the runs only returned a single solution, the other two converged towards a common Pareto front, spanning about 100 tons in GTOW but less than 1% in foreseen reliability. In fact, all solutions are characterized by the same concept, featuring a core stage with 2 Vulcain-2, an upper stage with 2 RL-10B and the presence of 2 boosters. The variability of launch mass and success probability is mainly due to the relative size of the cryogenic stages and of the solid boosters: in order to improve reliability, solutions on the right of the front allocate more SP mass to the boosters, resulting in higher GTOW but shorter, and hence less risky, flights of the two cryogenic stages.

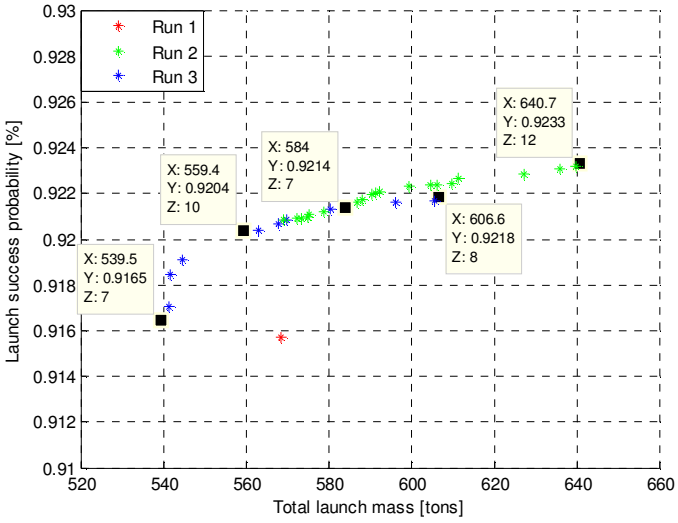


Figure 126: Free-architecture launcher MDO problem for GTO mission, multi-objective GTOW vs. MSP Pareto front.

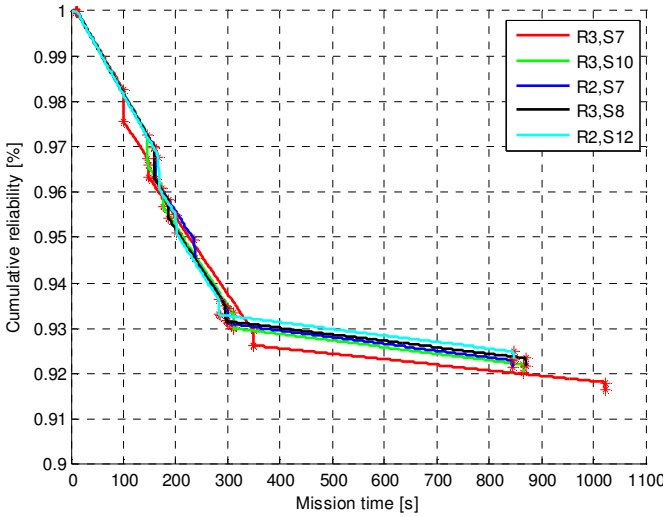


Figure 127: Reliability vs. time profile for the five solutions shown in Figure 126, where Z identifies the solution's ID.

All solutions converge towards common choices for most of the technological trade-offs: 2 boosters and 2 engines for each stage, Vulcain-2 and RL-10B engines, pad interface with the boosters (for lower structural mass on the first stage), separate bulkhead tanks with a lower diameter and advanced materials for the upper stage (again, reduced weight), and LOx forward for the first stage (for controllability). It is worth noting how none of the discrete variables which have larger system level effect originates a separate branch of the Pareto front. For instance, no non dominated solutions were identified with a linear configuration, probably as a consequence of the high failure rate of cryogenic systems, which more than offsets the risk advantage from avoiding the boosters' jettison event. This confirms how the tuning of the risk models, exactly as for the cost models, is a particularly critical issue for multi-objective MDO, certainly requiring further studies and probably a dedicated calibration on a case-by-case basis to obtain reasonable results.

Note also that the minimum mass solution for the free architecture problem has a GTOW of ~540 tons, constituting a sensible improvement with respect to the minimum mass solutions achieved for Ariane 5's large (~625 tons) and small (~539.5 tons) MDO problems. This confirms how an enlargement of the design freedom allows in general to improve the design objective, even though specific technological solutions - for instance common bulkhead Al-Li upper stage tanks - then need to be verified in terms of feasibility and affordability.

As a final note, larger MDO problems were also attempted, including for example the choice among 2 or 3 stages and 1 or 2 sets of boosters as well as even broader allowed options of propulsive and structural technologies. However, in these cases the MDO process failed to converge to globally optimal areas, providing significantly different solutions at each global run when allowing reasonable computational times (i.e. not more than 24 hours of computation per run). This is due to the fact that, when the architecture of the launcher is not constrained in any way, the resulting optimization domain is extremely vast and non smooth, hence requiring huge populations and numerous iterations to perform full search space exploration. Since only shared memory parallel computing on dual core architectures was exploited, solutions to these problems could not be found, although the extension of the optimization architecture to high performance computing may very well enable such attempts.

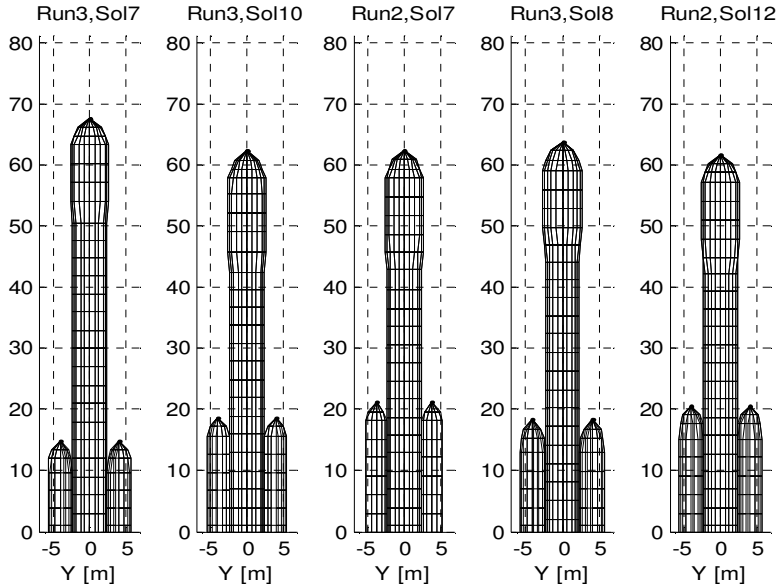


Figure 128: External geometry for the five solutions shown in Figure 126, where Z identifies the solution's ID.

Parameter	Bounds	R3,S7	R3,S10	R2,S7	R3,S8	R2,S12
$N_{boosters}$	0 or [2-4]	2	2	2	2	2
PIC	[1; 2]	2	2	2	2	2
$L_{nose,PLF}$	[2.8-8.3]	4.41	4.98	4.36	4.66	4.64
D_{US} *[m]*	As PLF or [4.0-5.4]	4.78	4.77	4.74	4.63	4.67
D_{LS} *[m]*	As US or [4.0-6.5]	As US	As US	As US	As US	As US
D_{BS} *[m]*	[1.5-3.5]	3.36	3.12	2.96	3.49	3.38
L_{core} *[m]*	-	67.7	62.3	62.3	63.8	61.8
L_{BS} *[m]*	-	14.9	18.7	21.2	18.4	20.6
$(L/D)_{core}$ *[]*	-	14.2	13.1	13.1	13.8	13.2
$(L/D)_{BS}$ *[]*	-	4.4	6.0	7.1	5.3	6.1
$M_{prop,LS}$ *[tons]*	[100-250]	210.1	186.4	180.3	177.8	169.6
$M_{prop,US}$ *[tons]*	[14-40]	31.8	26.2	25.6	27.0	26.6
$M_{prop,BS}$ *[tons]*	[80-300]	111.8	137.9	151.3	161.9	181.4
LS Tanks Type	Common bulkhead	-	-	-	-	-
LS Tanks Arrang.	Ox/Fuel aft	Fuel aft	Fuel aft	Fuel aft	Fuel aft	Fuel aft
US Tanks Type	Common/Enclosed	Enclosed	Enclosed	Enclosed	Enclosed	Enclosed
US Tanks Type	Ox aft	-	-	-	-	-
US tanks mat.	Al 7075 / Al-Li	Al-Li	Al-Li	Al-Li	Al-Li	Al-Li
US other mat.	Al 7075 / CFRP	CFRP	CFRP	CFRP	CFRP	CFRP
$M_{prop,LS}$ *[tons]*	-	21.5	19.6	22.0	21.6	21.3
$M_{prop,US}$ *[tons]*	-	6.9	6.4	6.4	6.4	6.7
$M_{prop,BS}$ *[tons]*	-	16.2	16.0	17.1	18.5	20.4
LS Propulsion	Vulcain 2 or new design LRE	Vulcain 2	Vulcain 2	Vulcain 2	Vulcain 2	Vulcain 2
$N_{eng,LS}$	[1-4]	2	2	2	2	2
US Propulsion	HM-7B, Vinci, RL-10B, RD-0146	RL-10B	RL-10B	RL-10B	RL-10B	RL-10B
$N_{eng,US}$	[1-2]	2	2	2	2	2
BS Propulsion	New SRM (P241 thrust profile)	-	-	-	-	-
T^{*}_{BS} *[MN]*	[2.0-8.0]	4.38	3.58	3.49	3.86	4.06
$p_{cc,BS}$ *[bars]*	[56.0-104.0]	86.5	82.1	85.8	81.0	84.3
ε_{BS} *[]*	[7.0-20.0]	10.3	12.1	9.5	11.9	11.4
$n_{ax,max}$ *[g]*	-	3.84	4.36	4.08	4.14	4.16
$Q_{dyn,max}$ *[kPa]*	-	42.0	31.6	30.8	30.5	31.0
GTOW [tons]	-	539.5	559.4	584.0	606.6	640.7
MSP []	-	0.916	0.920	0.921	0.922	0.923
CpL [M€]	-	138.6	140.2	145.6	147.9	152.8

Table 48: Min GTOW - max MSP free-architecture MDO problem: comparison of design data of the five solutions in Figure 126.

8.5. Future Launchers Preparatory Program case study

ESA's Future Launchers Preparatory Program (FLPP) was mentioned in the introduction as one of the main reasons for the selection of ELVs as the core application of the MDO methodology for the present research. FLPP is currently the main European framework for all activities related to the conceptual design of the next generation of European ELVs, with the aim of reducing the cost-per-kilo to orbit by half with respect to Ariane 5. Besides, the development of an entire family of launchers through modular design, rather than of a single type of vehicle, is at the core of the development philosophy with the purpose of drastically improving the launch system's flexibility. In this context, the last MDO application presented here is taken directly from FLPP studies, considering published requirements as a starting point for the set-up of the MDO problems.

According to recent information (**[142]**, **[143]**), mainly two families of launchers are being considered: 1) a so-called Building Block (BB) concept, featuring the assembly of existing or slightly adapted European LP stages or SRMs, and 2) a Next Generation Launcher (NGL) concept, gathering brand new lower composites derived from VEGA's technology for SRMs and/or Ariane's cryogenic liquid propulsion (though LO_X/CH_4 combinations are also being investigated in parallel) in combination with an upper stage based on Vinci EC engine. In order to minimize the recurring cost, a two-stage launcher architecture is always preferred, with the addition of strap-on boosters to increase the payload performance.

Since the BB approach is not particularly well suited to the MDO methodology, a study case from the NGL concept was selected, in particular encompassing the architectures labelled as NGL-HH-GG and NGL-HH-SC, which are mentioned in reference **[142]** to be the most promising. These are constituted of a LO_x-LH_2 upper stage with Vinci expander cycle engine and a LO_x-LH_2 (H) lower stage with a new design engine, either based on staged combustion (NGL-HH-SC) or on the more classical gas generator technology (NGL-HH-GG).

As a consequence of this selection, the high-level requirements taken from FLPP as a starting point for the MDO set-up are the following:

- Target performance of the in-line two-stage configuration: ***3 tons*** to GTO (from Kourou).
- Flexible extended missions with addition of a variable number of strap-on boosters, specifically up to ***5*** and ***8 tons***.
- Payload environment similar to that of Ariane 5: $n_{ax,max,PL}$***<4.55 g*** and $q_{heat,max,PL}$***=1135 W/m²***.
- Main cryogenic stage using a newly designed cryogenic engine of either SC or GG feed system.
- Upper stage using a cryogenic Vinci EC engine.
- Main stage and upper stage both featuring a common bulkhead tank configuration.
- Upper stage exploiting under-fairing configuration.
- Strap-on boosters based on Vega SRM technology.
- Possible in-air ignition of part of the strap-on boosters, in order to match the path constraints.
- Specific launch cost halved with respect to Ariane 5 ECA (i.e. approximately from 17 k€/kg to 8.5 k€/kg to GTO.
- Target reliability: 0.99

On top of these specifications, four additional trade-offs are left to the optimizer in the MDO problem: a) the use of two Vulcain-2 engines in the first stage instead of a single new LRE; b) the introduction of throttle capability in the new lower stage's LRE; c) the design of a classical (i.e. non under-fairing) configuration for the upper stage; and d) the exploitation of advanced Al-Li materials for the upper stage tanks. The first trade-off is motivated by the possible necessity to avoid the large development cost associated with a new high-thrust LRE, the second by the requirement of matching the axial acceleration and dynamic pressure path constraints, and the final two are related to the minimization of the upper stage's inert mass, which largely influences the overall payload performance. From synthesis of the specified requirements and reasonable guesses of the optimization variables' ranges[58], **Table 49** was compiled to summarize the MDO problem definition for both the inline ELV and the parallel ELV with strap-on boosters. In case of classical upper stage configuration, the 12 m PLF is assumed for 3 tons and 5 tons missions whereas the 17 m PLF is assumed for the 8 tons mission. In case of under-fairing configuration, these lengths are increased to include part of the upper stage.

In order to design the complete ELVs' family for 3, 5 and 8 tons performance targets, a rather complex study flow was developed, already shown in the introduction (Figure 4) as an example of the industrial applicability of the MDO research. The design of the three launcher configurations was executed sequentially, first targeting the baseline 3-tons in-line version and then adding the boosters for the 8-tons and 5-tons configuration. Note that a better approach would be represented by the concurrent optimization of all three configurations, since the sequential approach introduces artificial constraints in the design of the strap-on boosters, in particular with respect to the flight loads. However, the simultaneous MDO of different launcher configurations was not implemented due to time limitations.

[58] For instance, M_{prop} ranges are taken around the baseline values defined in **[142]**, T^*_{core} was derived from an assessment of the range of GTOW and a 1.5 lift-off T/W, and $T^*_{booster}$ is comprised between slightly less than Z23 and slightly more than P80.

The FLPP study case was already introduced in **Chapter 1** to exemplify the advantage of MDO methodology in terms of committed effort for ELVs conceptual design studies, with the caveat that the confidence in the achieved design solutions was to be subject to the evaluation of the expectable disciplinary errors and related global sensitivities. After the exhaustive validation effort described in **Chapters 5** and **7**, some examples of quantitative results are presented below, although the full study cannot be reported here for length limitations.

Area	Parameter	Type	Value/bounds
Program	$N_{launches}$, $N_{years,ops}$	fixed	6 launches/year, 20 years
Mission	M_{PL}	fixed	3/5/8 tons
	D_{PL}	fixed	4.57 m (i.e. 5.4 m PLF)
	L_{PL}	fixed	6.85/9.82[1] m (i.e. 12/17 m PLF)
	M_{PLA}	fixed	300/500 kg for 3&5/8 tons missions
Loads	$n_{ax,max}$, $q_{dyn,max}$, $q_{heat,max}$	fixed	4.55 g, 57 kPa, 40 MW/m^2
Architecture	N_{stages}	fixed	2
	$N_{boosters}$	fixed/discrete opt.var.	0 for inline or variable from 2 to 8.
Upper stage	*Prop*	fixed	LOx-LH2
	M_{prop}	continuous opt. var.	[10-30] tons
	N_{eng}, *OTS*, OTS_{ID}	fixed	1 Vinci OTS LRE
	p_{tanks}, p_{press}	continuous opt. vars.	[2.0-3.5] bars and [150-300] bars (heated He for LO_x)
	D_{const}	fixed	Yes
	UFC	boolean opt. var.	Yes or No
	k_{UF}	continuous opt. var.	[0.4-0.7]
	TT, *TA*	fixed	2 (common bulkhead) and 1 (LO_x aft)
	$SM_{main/sec}$	discrete opt. var.	1 (Al 7075) or 4 (Al-Li 2195)
	SC_{main}, SC_{sec}	fixed	0 (unstiffened) and 1 (integrally stiffened)
Lower stage	*Prop*	fixed	1 (LOx-LH_2)
	M_{prop}	continuous opt. var.	[100-180] tons
	N_{eng}, *OTS*, OTS_{ID}	boolean opt. var.	Yes or no (2 Vulcain-2 or 1 new design LRE)
	Feed	discrete opt. var.	GG or SG *(if new design)*
	T^*, α_P, ε	continuous opt. var.	[1.5-3.0] MN, [4.7-6.5], [30-80] *(if new design)*
	p_{cc}	continuous opt. var.	[30-120] or [80-270] bar *(if new GG or SC)*
	Nozzle, δ_{div}, *TVC*, δ_{TVC}	fixed	2 (bell nozzle), δ_{div}=12 deg, 1 (hydraulic), δ_{TVC}=7 deg
	p_{tanks}, p_{press}	continuous opt. vars.	[2.0-3.5] bars and [15-40] bars (heated He for LO_x)
	ThC	boolean opt. var.	[60-100]% or 100% only *(if new design)*
	D_{const}	fixed	Yes
	TT, *TA*	fixed	2 (common bulkhead) and 2 (LH_2 aft)
	$SM_{main/sec}$	fixed	1 (Al 7075)
	$SC_{main/sec}$	fixed	1 (integrally stiffened)
Boosters	*Prop*	fixed	4 (HTPB-AP-Al, Avio formulation)
	M_{prop}	continuous opt. var.	[10-30] tons
	SRM_{type}	fixed	1 (custom, P80 thrust profile)
	T^*, p_{cc}	continuous opt. vars.	[0.2-1.5] MN and [50-110] bar
	ε	continuous opt. var.	[7-15] for ground-lit, [15-40] for air-lit
	Nozzle, δ_{div}, *TVC*, δ_{TVC}	fixed	1 (cone nozzle), δ_{div}=15 deg, 2 (electric), δ_{TVC}=7 deg
	D	continuous opt. var.	[1.8-3.0]

Table 49: Fixed parameters and optimization variables (including boundaries) for the FLPP MDO problems set-up.

When assembling the data reported in **Table 49**, the MDO problem for the inline launcher only (3-tons case) is constituted of ***16*** design optimization variables and ***13*** trajectory optimization variables as follows:

- Propellant loading: $M_{prop,LS}$, $M_{prop,US}$
- Upper stage's geometry and structural material: *UFC*, K_{UF}, $SM_{main,US}$
- Upper stage's tanks pressures: $p_{tanks,US}$, $p_{press,US}$
- Lower stage's propulsion system: OTS_{LS}, $Feed_{LS}$
- Lower stage's new design cryogenic engine: T^*, α_P, ε, p_{cc}
- Lower stage's throttle: *ThC*
- Lower stage's tanks pressures: $p_{tanks,LS}$, $p_{press,LS}$
- 4 pitch-over parameters: $\Delta\theta_{PO}$, t_{PO}, $t_{PO,decay}$, $\Delta\psi_{PO}$
- 2 yaw parameters during exoatmospheric flight: $\Delta\psi_{LS}$, $\Delta\psi_{US}$
- 4 pitch parameters during exoatmospheric flight: $\Delta\theta_{LS}$, $\Delta\theta_{BTL,i}$, ζ_{BTL}, $\theta_{BTL,f}$
- 3 linear interpolation nodes for the lower stage's thrust throttle, if a throttleable engine is selected.

The best Pareto front obtained for the minimum mass versus minimum cost problem with DGMOPSO is reported in **Figure 129** as an example of the achieved FLPP results. It is worth noting that all high performance technologies are

exploited in the design solutions presented in the front, in particular: a) under fairing upper stage configuration with common bulkhead tanks in Al-Li 2195, ensuring very low inert masses; and b) single, new design, high chamber pressure (from ~200 to ~230 bars) SC engine with very high thrust levels (from 2.6 to 2.8 MN sea level) for maximum propulsive performance (I_{sp} from ~383 to ~395 s at sea level, from ~451 to ~460 s in vacuum). This allows obtaining extremely low GTOW values, ranging ***from 174 to 181 tons***, with propellant masses ***from 116 to 139 tons*** for the lower stage and ***from 13 to 29 tons*** for the upper stage. In fact, the ratio of the propellant loadings constitutes the largest difference in design among the solutions in the Pareto-optimal front, with a push towards higher upper stage masses for performance and higher lower stage masses for costs. In any case, the figures provided above appear sensibly lower than those mentioned in the official studies. In fact, ***150 tons*** and ***26 tons*** of propellant are reported in **[142]** and **[143]** for the lower and upper stages, with a total launch mass of ***226 tons***. These are however obtained with more conservative assumptions, such as traditional Al-2219 tanks, 150 bars of SC engine's chamber pressure and only 2.5 MN thrust at sea level.

It is also worth noting that two trajectory constraints impose important requirements on the design. In particular, the static controllability becomes critical in the atmospheric phases of the flight, since no boosters are available to aid the core's TVC. For this reasons, a large maximum deflection angle of the new engine's TVC has to be exploited (***~12 deg***), resulting in the controllability profile of **Figure 130**. Besides, the axial acceleration constraint of 4.55 g forces the use of a new engine with throttle capabilities, which are exploited as shown in **Figure 131** to reduce the thrust towards the end of the burn.

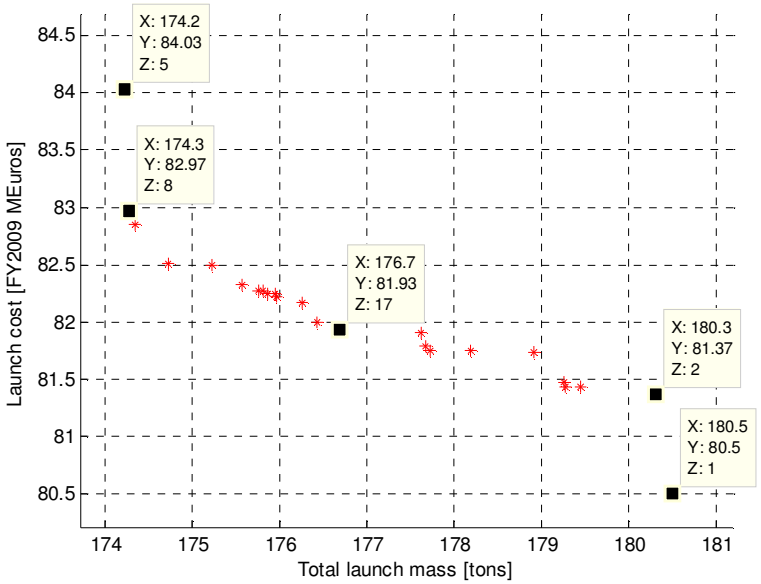


Figure 129: Min. GTOW- min CpL Pareto front for FLPP-3 tons MDO problem.

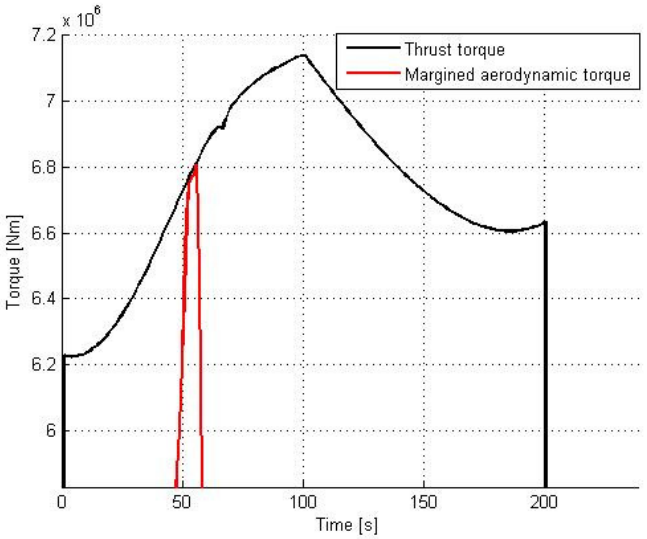


Figure 130: Example of controllability profile for the achieved optimal FLPP design solutions in Figure 129.

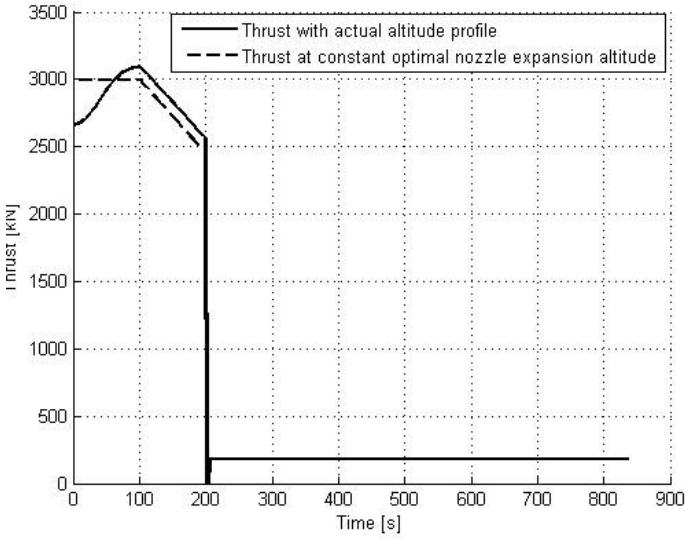


Figure 131: Example of thrust throttle profile for the achieved optimal FLPP design solutions in Figure 129.

We came all this way to explore the moon, and the most important thing is that we discovered the Earth.

CHAPTER 9

CONCLUSIONS AND FUTURE DEVELOPMENTS

9.1. Research conclusions

This PhD dissertation introduced MDO as a disruptive technological innovation if massively applied within industry to the conceptual and early preliminary design of complex systems. In fact, MDO has the potential for drastically improving both the design process and the design quality. With MDO tools assisting system engineers, the design efforts in terms of man-hours and hence financial costs can be largely reduced. Moreover, a much wider investigation of the design space can be performed, possibly identifying significantly better design solutions with respect to traditional fixed point iterations process, as regards to LCC or any other figure of merit. Hence, the goal of the research was set as the development and quantitative assessment of a MDO environment capable of tackling the early design phases of ELVs, which were selected as the primary application due to the current European relevance (FLPP, VEGA evolutions, Ariane 5 ME, ...).

The focus of the research was on the engineering modelling aspects, with the idea of attempting a detailed analysis of the accuracy of the developed methods, both in terms of disciplinary errors and of system level sensitivities. Although a rather significant number of developments documented in recent years have been focused on the application of MDO to the design of different types of STS, the evaluation of accuracy and reliability of such models to the extent described in the present research is a rather new concept, to the author's knowledge.

From the initial study of the state-of-art, several significant challenges connected to the MDO methodology were identified in the areas of modelling, hardware and software, and mathematics. These are summarized in **Table 50**, together with solutions which were proposed throughout this dissertation to deal with each of the highlighted issues, showing how the most important aspects enabling MDO can be successfully tackled with today's technology. Nevertheless, the difficult compromise between analysis fidelity and computational effort represents a massive obstacle, having in large part contributed to prevent from successful widespread industrial application of MDO in the past 20 years. The main question proposed for the present research is therefore centred on this issue - whether it is possible to develop relatively simple models permitting fast MDA cycles while still ensuring sufficient accuracy to place confidence in the achieved design solutions. Unfortunately, a straightforward answer could not be found: the required computational effort surely matches the original target (i.e. ***<2 s*** for a complete V2 MDA), but it is not nearly as easy to measure the accuracy requirement. It can however be said that the two-steps process described in **Chapters 4** through **7**, involving modelling, validation and critical analysis of the results, proved to be of key importance for the incremental improvement of the models, ensuring a sensible enhancement in accuracy at a manageable price in computational effort. As a result, the final (V2) models are characterized by accuracies on the global performance which should be sufficient in most of the cases to fairly compare two significantly different design solutions through MDO.

To provide a clear impression of the suitability of the MDO environment for the initial design of ELVs, a comprehensive overview of the main research achievements, lessons learned and remaining open points are schematically summarized:

- Through the two-steps modelling and validation on European launchers, a wide range of design functionalities encompassing all relevant disciplines was developed and implemented, which appears adequate for the conceptual and early preliminary level design of ELVs. In particular, ELVs can be optimized for performance, cost and/or reliability, including geometric, structural, trajectory and safety constraints covering most of the conceptual design aspects currently investigated within industry.
- Careful incremental modelling allows achieving sufficient subsystem-level accuracy in most areas as regards to the global performance error caused by disciplinary outputs, without introducing high fidelity methods. In particular,

one-variable-at-a-time SA highlighted how the total inert masses of stages/boosters (structural elements, propulsion components, other non structural subsystems) remain the most critical parameters even after the development of the analytical structural sizing module. Worst-case system level errors due to average errors on a single mass figure are in the order of ***~5%*** with V2 models. Propulsive parameters ($I_{sp,vac}$ and A_e) and aerodynamic coefficients (C_D and C_L) have a lower effect on payload performance (***~3%*** and ***~1%***), the former due to a rather high accuracy and the latter to low sensitivity.

Area	Challenge	Solution
Modelling	Find a suitable compromise between disciplinary analyses fidelity and required computational effort (e.g. engineering-level vs. high fidelity).	2-steps modelling procedure, highlighting the critical aspects to be improved in the second step from a quantitative analysis of the results of the first step.
	Develop well-posed interdisciplinary interfaces and consistent multi-disciplinary problem formulation.	Use of C++ object-oriented programming paradigm to define clear interfaces and extendable modules.
	Capture (at least partially) the human expertise in the multidisciplinary design models.	Interviews at ESA-ESTEC of disciplinary experts, with the obtained hints directly integrated in the models.
	Define a parametric geometry model in common to all involved disciplines (aerodynamics, structures,...).	Use of LaWGS format for engineering-level tools (no high fidelity analyses, simplified geometry definition).
	Model the non-technical disciplines (cost estimation, reliability assessment, schedule and programmatics).	Use of open literature (TRANSCOST) complemented by ESA expertise for costs and reliability assessment.
	Verify and validate the developed MDA models with respect to real-world data.	4-steps validation procedure, aimed at identifying the main weaknesses and assessing the accuracy of the models, repeated for both modelling steps for comparison.
HW & SW	Procure and integrate the required disciplinary codes (proprietary, open source, commercial, ...), often in different languages and coding philosophies.	Use of open source *legacy* code only (i.e. source code available and free to modify), interfaced with I/O pipe files and write/read scripts for simplicity.
	Ensure version control of complex software as well as management, storing and communication of large amounts of data.	Use of svn for version control and of xml database structure with an automatic parser to C++ objects. All interdisciplinary communication only through a central management routine.
	Allow for interactive machine-human interface, to ease the introduction of the designer's expertise.	Reverse communication during optimization, provisions for easy set-up of successive global and local optimization runs.
	Provide large computational resources, such as parallel processing or GPU computing.	Master-slave parallelization of GO algorithms with OpenMP for shared memory computing, exploiting dual-core machines.
Mathematics	Implement an efficient GO/LO architecture, involvin procurement or development of several algorithms.	Integration of GO algorithms from previous research and of a large SQP library developed at a collaborating institution.
	Select a suitable MDO architecture and problem formulation, depending on problem characteristics.	Exploitation of BBO with both MDF and IDF formulations (MDF slightly superior), NOL attempted but inefficient.
	Define surrogate models of specific analyses if CPU effort for the involved disciplines is too large.	Found not to be necessary for engineering-level analyses, left as a future development (after introduction of high fidelity).

Table 50: Main challenges of MDO methodology identified within the research, including the implemented or proposed solutions.

- Trajectory optimization allowing for accurate and robust assessment of ELVs' payload mass is one of the key aspects of MDO, enabling fair comparison of the performance of different designs. Significant work in this area (described in **Sections 4.6** and **6.4**) led to the definition of environmental, dynamics (3-DoF), guidance and control models ensuring very good performance assessment accuracy and robustness. Besides, the introduction of several enhancements in the trajectory models' smoothness (**[A11]**) allows the exploitation of local optimization with WORHP NLP solver. The local approach is definitely more efficient on the trajectory subproblem, leading to faster and robust performance assessments in case of ascent optimizations executed as stand-alone processes (i.e. fixed design ELV) or nested in a MDA cycle. Validation cases provided payload estimates within ***5%*** of the actual performances of existing launchers. This residual error could not be eliminated, and is possibly due to inaccuracies in the design data as well as to residual modelling issues (e.g. 3-DoF dynamics, lack of wind impact on the EoMs, ...).
- Cost and reliability estimates appear at a first sight rather accurate. For instance, both the overall CpL and the spread among development, production and operations assessed through MDA cycles match the approximate figures openly available for Ariane 5 ECA and VEGA launch systems. Besides, although a pessimistic evaluation of the cryogenic engines' failure rate leads to underestimated reliability for Ariane 5 ECA with respect to its impressive launch record, the computed MSP of VEGA is very close to the target 98% reliability. Nevertheless, accuracy on these test cases is sided by significant issues related to the exploitation of the cost and reliability models in multi-objective MDO, as described below.

- Montecarlo simulations represent a powerful method to assess the expectable range of performance error, given the launcher's architecture and estimates of the Gaussian curves (i.e. μ and σ) describing the inaccuracies in the disciplinary parameters. From the analyses, the 1σ payload mass ranges for the European test cases appear sufficiently narrow for the considered level of detail, with an error envelop lowered from ***±19%*** to ***±12%*** from V1 to V2. At the same time, the accuracy in the payload performance assessment through the MDA cycle is extremely good, with only ***+4%*** and ***+6%*** errors for Ariane 5 ECA and VEGA. More importantly, the performance bias from both Montecarlo SA and MDA processes is on the same side for the two European ELVs (i.e. slightly optimistic). Since Ariane and VEGA have radically different architectures, this suggests that trade-offs obtained with MDO can be entrusted even if the difference in performance is lower than the mentioned ***~12%***.
- To summarize the comparison of V1 and V2 models, the upgrades significantly improve the behaviour in terms of overall global performance accuracy and confidence to be placed in the design solutions. This is obtained at a limited price in terms of computational effort: no high fidelity analyses were included, nor were other relatively computationally intensive methods such as panels aerodynamics, 6 DoF dynamics, integrated Montecarlo simulations or bi-dimensional propulsive system analyses. As a consequence, computational times for both MDAs and MDOs are only roughly doubled from V1 to V2 models, a figure which seems acceptable in light of the above mentioned advantages.
- Small and single-objective (min GTOW) MDO problems for Ariane 5 ECA and VEGA were employed to verify the potential of MDO to improve the design of existing launch vehicles. Although VEGA's design appears more GTOW-optimal with respect to Ariane's, a positive net effect of MDO is obtained for both ELVs. Such effect is much more limited with V2 models, passing from ***~20%*** and ***~9%*** to ***~6%*** and ***~2%***. This is however the outcome of an improvement in the models' accuracy, since the larger GTOW reductions obtained within V1 are partially synthetic in nature: the exaggerated effect on the structural masses of the trajectory load parameters leads the optimizer to strongly reduce $q_{dyn,max}$, $n_{ax,max}$ and $q_{heat,max}$, allowing for artificially low inert masses.
- Local refinement of global optimization runs was shown to reduce the objective function value, but only to a limited extent (***<2%***). Moreover, different starting points always lead to different local optima, suggesting that the smoothness of the MDA objective and constraint functions should definitely be improved. Since stand-alone local trajectory optimization is satisfactory from this point of view, efforts should be focused on the multidisciplinary models definition, although the multi-modality of the constraints and especially of the objective function may be difficult to limit.
- A comparison of two different problem formulations, MDF and IDF, points at a different behaviour within V2's MDA models for ELVs with respect to typical literature results. In fact, a limited but non negligible advantage of the straightforward MDF approach was highlighted by the available testing results, mostly due to the very low number of required iterations of the Trajectory-Structures loop. Also taking into account that the built-in multidisciplinary feasibility ensures higher robustness, MDF formulation seems better suited than IDF for the specific applicative case of ELVs' design optimization, although more extensive testing on other launch vehicles could provide more conclusive insights.
- The enlargement of the MDO problem brings significant reductions in the GTOW of Ariane 5 ECA and VEGA design. In particular, allowing modifications to the propulsion systems results in several physically sound design changes (e.g. increase in HM-7B's chamber pressure, stretching of P241 boosters, ...), although others should be verified with more detailed analyses (e.g. VEGA SRMs' internal pressure decrease). When the MDO problem is further enlarged by unlocking several discrete optimization variables (i.e. "free-architecture" MDO case), additional advantages in terms of design objectives can be achieved, although each new technological solution needs to be a posteriori verified for technical feasibility and affordability.
- As far as purely performance objectives are concerned (i.e. GTOW and PL mass), multi-objective MDO provides very useful design information. For example, min GTOW vs. max PL optimization can be exploited to understand the sensitivity of the launch vehicle being designed to variations of the payload performance target, which is an extremely important figure in real world design exercises.
- The introduction of cost and reliability objectives in MDO is instead more complicated. Specifically, mathematical features intrinsic to the CpL and MSP calculation constitute a serious obstacle for global multi-objective optimization. In fact, although technical quality factors contribute to including technological considerations, cost estimation through CERs represents a mass-based approach, which often results in cost reductions when minimizing mass and vice versa. Moreover, the reliability of ELVs is quite insensitive to the values of all continuous design variables, being mostly affected by architectural and technological aspects. Finally, several tuning parameters are defined for each of the CERs and related aggregating functions as well as for the failure rates calculation. These strongly affect the LCC and MSP of launch systems, so that the achieved Pareto fronts radically change according to their values. Examples shown within this research include the effects of large SRMs' technology factor and cryogenic systems failure rate respectively on mass vs. cost and mass vs. reliability Pareto fronts. As a consequence,

the parameters' tuning is a particularly critical open point for MDO, which needs to be further addressed with additional testing on a larger number of test cases.

- Models tuning is a critical aspect also for performance-related disciplines. In fact, the good behaviour of the models highlighted in the previous points is subject to the consideration that only European ELVs were tested, and the same ELVs had been often used for models calibration due to lack of information on other systems. In particular, inert weights and lengths assessments are more prone to experimental tuning than other modelling aspects. Further testing on ELVs with widely different underlying technologies therefore appears necessary to complete the validation of the models by understanding the level of accuracy which results from extrapolations to non-European designs (e.g. US government and commercial vehicles, Russian launchers, ...).
- Another current limitation lays in the inability of the MDO environment to converge to globally optimal areas in case of very large MDO problems. This is experienced when the architecture of the launcher is not constrained in any way (e.g. variable number of stages and booster sets, variable technologies in each stage/booster), resulting in a very wide and non smooth search space. In theory, global optimization should be able to tackle these problems as well, but the number of required model evaluations increases drastically with the dimensions of the problem. Since only shared memory parallel computing on dual core architectures is exploited, solutions to these problems could not be easily obtained. High performance deployment thus constitutes another critical point to be further researched, possibly opening the possibility to uncover relevant trade-offs in the architectural design of ELVs, presently prevented by computational resources.
- Also requiring HPC, the impact of the introduction of high fidelity information in the MDA cycle represents another prominently interesting open point to be evaluated. In fact, high fidelity methods should further reduce the errors highlighted here for Ariane 5 and VEGA, besides allowing to tackle more generic configurations without the risk of completely missing the value of specific design parameters due to extrapolation in the engineering level models. In particular, coarse FEM in combination with analytical failure modes analysis seems the most indicated high fidelity methodology to be introduced, in light of the criticality of the calculation of the structural masses.

In conclusion, this research highlighted how the development and tuning of reliable MDO environments is a very complex task, requiring large efforts in most engineering areas, as well as computer science and mathematics. However, detailed investigations showed that reasonable accuracies and physically sound design modifications can be obtained through the MDO approach, even when exploiting only fast engineering level models to avoid resorting to HPC. With today's computational resources, enabling introduction of high fidelity information in the design cycle, MDO guided by human expertise is a powerful approach for the initial design phases of launchers and other STS. Although the huge initial investment in terms of development and personnel training is a major obstacle to widespread industrial application, it is therefore the opinion of the author that the resulting benefit in terms of design quality is worth the effort, with the potential of contributing to the long term goal of achieving low cost access to space.

9.2. Future developments

Despite the large effort undertaken for this research, MDO is such a large field that the numerous areas of investigation (problem formulation and decomposition, MDO architectures, optimization methods and other mathematical techniques, engineering applications) were only partially covered. Hence, a relevant number of research topics can be identified for future study, which can be divided in three main groups: 1) short terms developments of the MDO environment for ELVs; 2) further improvements in the design fidelity of the models; and 3) extension to other classes of space vehicles.

Short terms developments include most of the aspects pointed out in the previous section as weaknesses still to be addressed in the implemented MDO environment, in particular:

- Verification of the multi-disciplinary models on non European technologies, such as US government (e.g. Delta IV, Atlas V, ...) and US commercial (Falcon family, Antares, Liberty, ...) ELVs, Russian (Soyuz, Proton, Dnepr, ...) and Japanese (H-II) launchers, or systems from emerging countries (China, India, Brazil). These can straightforwardly be studied with the models described in this dissertation, although very specific details may require further tuning. As an example, only an approximate geometry of Soyuz launch vehicle can be obtained (**Figure 132**) since no provisions are implemented for conical booster structures, even though this does not largely affect the estimated performances.

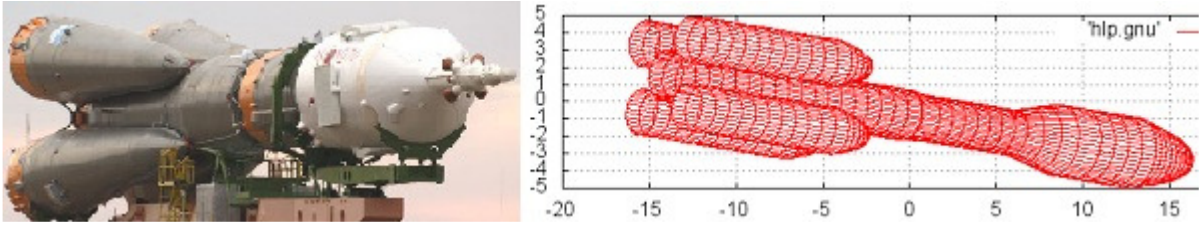


Figure 132: Soyuz launch vehicle in photo and as drawn with the developed models, in its 2.1B version.

- Attempts to improve the tuning of the cost and reliability parameters, which was shown to be particularly critical as well as complicated in this research. Besides, several optimization variables strictly related to non-performance aspects were initially introduced, but their actual influence on costs and reliability was not deeply investigated. These include ***LPM***, ***SSM***, ***SM***, ***RL***, ***LCE***, ***TOE***, $N_{eng,tests}$, which should be connected to the CpL and MSP of launch systems in order to allow for fair trade-offs against sheer performance.
- Exploitation of HPC for MDO problem enlargement. This can be accomplished either through Message Passage Interface (MPI) technology for distributed memory architectures or through modern GPU computing paradigms (e.g. CUDA or OpenCL).

Improvements in the accuracy of the models for the design of ELVs constitute a much broader research line, possibly leading to the development of a Version 3 of the MDA cycle. In particular, the following aspects should be covered in the various disciplines, being mainly constituted of analyses taken into consideration for the early preliminary models described for V2 but then discarded for time and/or computational effort constraints:

- *SP grain optimization*: introduction of a fast subsystem optimization nested in the MDA to match given thrust profile constraints, such as corridors for satisfying performance, max dynamic pressure or max axial acceleration values.
- *Solid grain domes modelling*, from both the structural and the propulsive law point of view.
- *Component-level mass estimation* for LP and SP systems: very complex upgrade requiring development of physics-based sizing tools for LP combustion chambers and LP/SP nozzles (structural and thermal analyses with different cooling options), as well as of a full engine cycle analysis for turbopump systems. This includes pressure balance equations, analytical relations for the assessment of the main turbomachinery parameters, and CEA analyses to compute the gas generators and pre-burners performances. From the basic parameters of all the power pack components, an estimation of the inert masses can then be performed with empirical or analytical models. Finally, analytical sizing tools for TVC, vernier engines and RCS systems should be developed, as well as relations to estimate the effect of throttle and restart capability on the inert masses (see **[144]** and **[145]** for some examples of the development of analytical sizing methods for rocket engines).
- *Geometry upgrades*: introduction of additional tanks configurations (e.g. toroidal, multiple cylinders, ...) and of optimizable interstage geometries for non constant diameter cores (e.g. cone angles, tolerances).
- *Aerodynamics panel methods*: replacement of DATCOM with a combination of panel impact methods for hypersonic speeds, subsonic/supersonic linear aerodynamics methods, and semi-empirical viscous relations. Note however that the accuracy advantages of this approach (which may be limited especially for the CoP position) have to be carefully weighted against the increase in computational times and the rather large development effort.
- *High fidelity structural models*: although the analytical structural sizing improved the accuracy in the estimation of the inert masses, this remains a critical aspect. Further enhancement of the analytical model can be specified, such as the definition of neglected load cases (e.g. ground winds on pad, pull-up manoeuvre for launchers integrated horizontally) or the estimation of the structural flexibility of ELVs, which could be used as constraint to limit the aerodynamic push towards thinner vehicles. Nevertheless, as already mentioned, Structures discipline seems the most indicated for a quantitative assessment of the introduction of high fidelity information, such as coarse FEM (see for example the approach in **[146]**).
- *Montecarlo analyses for trajectory uncertainties*, with the aim of statistically determining both the ground impact ellipse in case of stages/boosters re-entries and the orbital insertion accuracy for the payload release. Both aspects can be used as constraints or even objectives in a MDO or trajectory optimization scenario.
- *Improved safety analyses* with respect to two main aspects: 1) introduction of NASA's Debris Assessment Software (DAS)[59] for the calculation of the orbit's lifetime in case of upper stages and the estimation of the debris distribution for the break-up of suborbital components; and 2) calculation of the probability of fatalities along the vehicle's flight path caused by ground impacts, as line integration of the population density multiplied by the failure probability at each time instant. As an alternative, artificial smoothing of the population density map or arbitrarily modified maps to represent specific drop corridors could improve the current safety constraint definition.
- *Dynamic controllability verification*: since static controllability analyses were shown to be often inadequate to pose significant constraints to other launcher design aspects (e.g. aerodynamic stretching), introduction of dynamic controllability verification through 6-DoF simulations should be considered. In light of the relevance of this aspect, a prototype was developed connecting an open source multibody analysis software (*MBDyn*[60]) within a Matlab-Simulink environment to the outputs of the MDO process. Results from this approach, involving the modelling of nozzle and main vehicle as separate bodies, were encouraging in terms of controllability verification potential. However, control law issues and long computational times represent obstacles to the introduction of this method in the MDO loop, therefore further studies are recommended in this area.

[59] http://orbitaldebris.jsc.nasa.gov/mitigate/das.html
[60] http://www.aero.polimi.it/~mbdyn

- *Thermal modelling*: all thermal insulation and protection issues were considered in V1 and V2 models only through historical WERs. Introduction of physics based methods such as nodal analyses should be investigated, in order to improve the TPS masses accuracy and to allow for additional thermal conditioning considerations such as boil-off assessment for restartable cryogenic upper stages.

The implementation of a third modelling step including the above items would most likely require the complication of the MDO architecture and the exploitation of some form of HPC. In fact, the dimensions of the disciplinary search spaces and the computational effort involved in the new analyses would make the MDA cycle particularly prone to problem decomposition. For instance, subspace optimizations could be introduced in Propulsion (grain shape design optimization) and Structures disciplines (structural details optimization), possibly investigating architectures such as CO or BLISS. Moreover, if FEM is added to the design loop, MPI or GPU implementation appears mandatory, and meta-modelling methods also should be considered for reducing the computational effort.

The last proposed research line is even broader, considering the extension of the MDO environment to other classes of space vehicles. Several new applications can be conceived, the most closely related to ELVs being the introduction of man-rating functionalities, such as launch escape system, crew interfaces and safety requirements. Alternatively, the implementation of aerodynamic, weight, structural, guidance and control models for capsule shaped REVs certainly requires a larger effort (see **[A3]** and **[A6]** for a preliminary modelling). However, this represents a step towards merging ascent and re-entry vehicles, eventually leading to the MDO of reusable launch systems. Finally, MDO could be extended to orbital spacecrafts, integrating existing mission analysis libraries with multidisciplinary design models to introduce selected aspects of satellite and probe design in global trajectory optimization problems. As a last remark, the combination of these widely different applicative scenarios can also be considered, tackling the concurrent optimization of multiple branches of space missions constituted of different types of vehicles. Although presenting very challenging issues, this is an extremely interesting new concept, enabling the search for globally optimal solutions of complete interplanetary missions. Examples span from composite spacecrafts including an orbital and a planetary entry module to complex Mars Sample Return architectures with launch, interplanetary travel, Mars entry, Mars ascent, Earth return and re-entry.

APPLICABLE DOCUMENTS

The reports listed below represent final deliverables to ESA as part of the PRESTIGE documentation, and give further details on all design, modelling, validation and application phases of the research. All documents were written by the author and a colleague from the University of Bremen, Annalisa Riccardi. The author was in charge of all the engineering sections of the documentation, as well as of the sections on the overall MDO architecture and results. Annalisa Riccardi was instead in charge of all sections related to optimization methods and algorithms, GUI and software architecture. Labels in the last column define the area of each document, in particular M for Mathematics, E for Engineering, or M+E for both.

	Document	**Reference**	**Title**	
[A1]	Technical Note D1a	D1a.pdf	"Concurrent Design Optimization architectures and techniques"	M+E
[A2]	Technical Note D1b	D1b.pdf	"Launch Vehicles Design Environment - Applications selection and disciplines review"	E
[A3]	Technical Note D2	D2.pdf	"Launch Vehicles Design Environment - Engineering Models"	E
[A4]	Technical Note D2b	D2b.pdf	"Cost and Reliability Models"[61]	E
[A5]	Technical Note D3	D3.pdf	"Launch Vehicles Design Environment - Applicability of CDO/MDO techniques"	M
[A6]	Technical Note D6	D6.pdf	"Launch Vehicles Design Environment - Parametric Relations and Optimization Variables"	E
[A7]	SRD	SRD.pdf	"SVAGO Software Requirement Document"	E
[A8]	ADD	ADD.pdf	"SVAGO Architectural Design Document"	M
[A9]	VVD, Version 1	VVD1.pdf	"SVAGO Version 1 Verification and Validation Document"	E+M
[A10]	VVD, Version 2	VVD2.pdf	"SVAGO Version 2 Verification and Validation Document"	E+M
[A11]	TOS	TOS.pdf	"Trajectory Optimization Study with SVAGO"	E+M
[A12]	FLPP	FLPP.pdf	"Future Launchers Preparatory Program case study with SVAGO"	E
[A13]	User Manual	UM.pdf	""SVAGO Version 1 User's Manual"	E+M
[A14]				

[61] This document is separated from Technical Note D2 since it was prepared by Paolo Martino, a MSc student who performed his thesis work in the development of cost and reliability models for PRESTIGE research.

BIBLIOGRAPHY

[1]	Stine D.D., "The Manhattan Project, the Apollo Program, and Federal Energy Technology R&D Programs: A Comparative Analysis", Congressional Research Service's Report for Congress RL34645, June 30, 2009.
[2]	Sobieszczanski-Sobieski J., "Multidisciplinary Optimization for Engineering Systems: Achievements and Potential", NASA TM 101566, March 1989.
[3]	Hammond W.E., "Design Methodologies for Space Transportation Systems", AIAA Educational Series, 1st Edition, 2001.
[4]	Malone B., "On the Financial Impact of MDO on the Corporation", 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Atlanta, 4-6 Sept. 2002, AIAA 2002-5495.
[5]	Diedrich A., Hileman J., Tan D., Willcox K., and Spakovszky Z., "Multidisciplinary Design and Optimization of the Silent Aircraft", 44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January 2006, AIAA 2006-1323.
[6]	Berens T., "HISAC – High speed aircraft SSBJ", AIAA 2009 Year Review.
[7]	Sobieszczanski-Sobieski J., "A Linear Decomposition Method for Large Optimization Problems", NASA TM 83248, February 1982.
[8]	Sobieszczanski-Sobieski J. and Barthelemy J.F., "Improving Engineering System Design by Formal Decomposition, Sensitivity analysis, and Optimization", Proceeding of the International Conference of Engineering Design, Hamburg, West Germany, August 1985.
[9]	Wrenn G.A. and Dovi A.R., "Multilevel Decomposition Approach to the Preliminary Sizing of a Transport Aircraft Wing", AIAA Journal of Aircraft, Vo1.25. No. 7, pp. 632-638, July 1988.
[10]	Bindolino G., Ghiringhelli G.L. and Ricci S., "A Multi-Level Procedure For The Preliminary Sizing Of Aircraft Wing-Box Structures", XVIII National AIDAA Congress, September 2005.
[11]	Sobieszczanski-Sobieski J., "Optimization by Decomposition: a Step from Hierarchic to Non-Hierarchic Systems", NASA report N89-25149, 1989.
[12]	Bloebaum C.L., Hajela P. and Sobiesezczanski-Sobieski, J., "Non-Hierarchic System Decomposition in Structural Optimization", Engineering Optimization, Vol. 19, pp. 171-186, 1992.
[13]	Kroo I., Altus S., Braun R. Gage P. and Sobieszczanski-Sobieski J., "Multidisciplinary Optimization Methods for Aircraft Preliminary Design", AIAA 94-4325-CP, 1994.
[14]	Braun R. and Kroo I., "Development and Application of the Collaborative Optimization Architecture in a Multidisciplinary Design Environment", Proceedings of the ICASE/NASA LangleyWorkshop on Multidisciplinary Design Optimization, Hampton, Virginia, March 1995.
[15]	Braun R., "Collaborative Optimization: an architecture for large-scale distributed design", PhD dissertation, Stanford University, Department of Aeronautics and Astronautics, May 1996.
[16]	Braun R., Gage P., Kroo I. and Sobieszczanski-Sobieski J., "Implementation and Performance Issues in Collaborative Optimization", AIAA 96-4017, 1996.
[17]	Sobieszczanski-Sobieski J., Agte J.S. and Sandusky R.R.Jr., "Bi-Level Integrated System Synthesis (BLISS)", NASA TM-1998-208715, August 1998.
[18]	Sobieszczanski-Sobieski J., Emiley M.S., Agte J.S. and Sandusky R.R.Jr., "Advancement of Bi-Level Integrated System Synthesis (BLISS)", NASA TM-2000-210305, December 2000.
[19]	Sobieszczanski-Sobieski J., Altus T.D., Phillips M. and Sandusky R.R.Jr., "Bi-Level Integrated System Synthesis (BLISS) for Concurrent and Distributed Processing", AIAA 2002-5409, 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Atlanta, Georgia, September 2002.

[20]	Sobieszanski-Sobieski, J., Altus, T., Phillips, M., Sandusky, R., "Bi-Level Integrated System Synthesis (BLISS) for Concurrent and Distributed Processing," AIAA Journal, Vol. 41, No. 10, 2003, pp. 1996-2003.
[21]	Cramer E.J., Frank P.D., Shubin G.R., Dennis J.E. Jr. and Lewis R.M., "On Alternative Problem Formulations for Multidisciplinary Design Optimization", AIAA 92-4752.
[22]	Padula S.L., Alexandrov N. and Green L.L., "MDO Test Suite at NASA Langley Research Center", AIAA 96-4028.
[23]	Balling L.J. and Wilkinson C.A., "Execution of Multidisciplinary Design Optimization Approaches on Common Test Problems", 6th Symposium on Multidisciplinary Design and Optimization, Bellevue, WA, September 1996.
[24]	Alexandrov M.N. and Lewis R.M., "Analytical and Computational Properties of Distributed Approaches to MDO", 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis & Optimization, AIAA 2000-4718.
[25]	Brown N.F. and Olds J.R., "Evaluation of Multidisciplinary Optimization (MDO) Techniques Applied to a Reusable Launch Vehicle", 43rd AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January 2005, AIAA 2005-707.
[26]	Tedford N.P. and Martins J.R.R.A "On the Common Structure of MDO Problems: a Comparison of Architectures", 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Portsmouth, Virginia, September, 2006.
[27]	De Wit A.J. and Van Keulen F., "Numerical Comparison of Multi-Level Optimization Techniques", AIAA 2007-1895.
[28]	Yi S.I., Shin J.K and Park G.J., "Comparison of MDO methods with mathematical examples", Structural and Multidisciplinary Optimization, Vol. 35, pp. 391402, 2008.
[29]	Olds J.R., "Multidisciplinary Design of a Rocket-Based Combined-Cycle SSTO Launch Vehicle using Taguchi Methods", AIAA/AHS/ASEE Aerospace Design Conference, Irvine, CA, February 1993, AIAA 93-1096.
[30]	Olds J.R., "System Sensitivity Analysis Applied to the Conceptual Design of a Dual-Fuel Rocket SSTO", 5th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Panama City, September 1994, AIAA 94-4339.
[31]	Braun R.D., Powell R.W., Lepsch R.A., Stanley D.O and Kro I.M., "Comparison of Two Multidisciplinary Optimization Strategies for Launch-Vehicle Design", Journal of Spacecraft and Rockets, Vol. 32, No. 3, May-June 1995.
[32]	Braun R., Moore A.A. and Kroo I., "Use of the Collaborative Optimization Architecture for Launch Vehicle Design", 6th Symposium on Multidisciplinary Design and Optimization, Bellevue, WA, September 1996.
[33]	Braun R., Moore A.A. and Kroo I., "Collaborative Approach to Launch Vehicle Design", Journal of Spacecraft and Rockets, Vol. 34, No. 4, July-August 1997.
[34]	Acton D.E. and Olds J.R., "Computational Frameworks for Collaborative Multidisciplinary Design of Complex Systems", AIAA 1998- 4942.
[35]	Young D.A., Olds J.R., Hutchinson V., Krevor Z., Pimentel J., Reeves J.D., Sakai T. and Young J., "Centurion: A Heavy-Lift Launch Vehicle Family for CisLunar Exploration", AIAA Paper 2004-3735.
[36]	Olds J.R., Bradford J., Charania A., Ledsinger L, McCormick D. and Sorensen K., "Hyperion: An SSTO Vision Vehicle Concept Utilizing Rocket-Based Combined Cycle Propulsion", AIAA 1999-4944.
[37]	Steffes S.R. and Olds J.R., "Magel: An Advanced Partially Reusable Launch Architecture Concept", 12th AIAA/AHI Space Planes and Hypersonic Systems and Technologies Conference, Norfolk, Virginia, December 2003.
[38]	Kokan T., Olds J.R., Hutchinson V. and Reeves J.D., "Aztec: A TSTO Hypersonic Vehicle Concept Utilizing TBCC and HEDM Propulsion Technologies", AIAA 2004-3728.
[39]	Young D.A., Kokan T., Clark I., Tanner C. and Wilhite A., "Lazarus: A SSTO Hypersonic Vehicle Concept Utilizing RBCC and HEDM Propulsion Technologies", 14th AIAA/AHI Space Planes and Hypersonic Systems and Technologies Conference, Canberra, November 2006.
[40]	Durant N., Dufour A.., Pain V., Baudrillard G., and Schoenauer M., "Multidisciplinary Analysis and Optimisation Approach for the Design of Expendable Launchers", 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, September 2004.
[41]	Jean-Marius T., "Multidisciplinary Optimization for Early System Design for Expandable Launchers", AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, September 2006.
[42]	Balesdent M., Bérend N. and Dépincé P, "Optimal Design of Expendable Launch Vehicles using Stage-Wise MDO Formulation", 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference, 13-15 Sep 2010, Fort Worth, Texas, AIAA 2010-9324.
[43]	Mollmann C., Wiegand A., Dalheimer M., Martinez Barrio A., and Kauffmann J., "Multidisciplinary Design

	Optimisation of Expendable Launchers in ASTOS", 4-th International Conference on Astrodynamics Tools and Techniques, Madrid, Spain, May 2010.
[44]	Krishnan R., Sistla R. and Dovi A.R., "High Speed Civil Transport Design Using FIDO", NASA CR-1999-209693, October 1999.
[45]	Walsh J.L., Townsend J.C., Salas A.O., Samareh J.A., Mukhopadhyay V. and Barthelemy J.F., "Multidisciplinary High-Fidelity Analysis And Optimization Of Aerospace Vehicles, Part 1: Formulation", AIAA 2000-0418.
[46]	Walsh J.L., Weston R.P., Samareh J.A., Mason B.H., Green L.L. and Biedron R.T., "Multidisciplinary High-Fidelity Analysis And Optimization Of Aerospace Vehicles, Part 2: Preliminary Results", AIAA 2000-0419.
[47]	Monell D., Mathias D., Reuther J. and Garn M., "Multi-Disciplinary Analysis for Future Launch Systems Using NASA's Advanced Engineering Environment (AEE)", AIAA Paper 2003-3428.
[48]	Monell D., Verhage M., Vander Kam J., Mathias D., Reuther J. and Garn M., "The Advanced Engineering Environment (AEE) Project for NASA's Next Generation Launch Technologies (NGLT) Program", AIAA Paper 2004-0202.
[49]	Theisinger J.E. and Braun R.D., "Multiobjective Hypersonic Entry Aeroshell Shape Optimization", AIAA 2008-5873.
[50]	Johnson J.E., Lewis M.J. and Starkey R.P., "Coupled Entry Heat Shield/Trajectory Optimization for Lunar Return", AIAA 2008-6557.
[51]	Hanninen P.G., Lavagna M.R., Finzi A.E. and Marraffa L., "Preliminary design global optimization for space vehicles during atmospheric manoeuvres", AIAA 2005-3271.
[52]	Armellin R., Lavagna M., Starkey R.P. and Lewis M.J., "Aerogravity-Assist Maneuvers: Coupled Trajectory and Vehicle Shape Optimization", Journal of Spacecraft and Rockets, Vol. 44, No. 5, September–October 2007.
[53]	Padula S.L. and Gillian R.E., "Multidisciplinary Environments: A History of Engineering Framework Development", 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Virginia, Sept. 2006, AIAA 2006-7083.
[54]	Phoenix Integration, "Improving The Engineering Process With Software Integration", ModelCenter Technical White Paper, February 2003, available at www.phoenix-int.com
[55]	Simulia Products, "Isight 5.6 Datasheet", December 2011, available at: www.simulia.com/products/isight.html
[56]	Technosoft Inc, "The Adaptive Modeling Language. A Technical Perspective, 2003, available at www.technosoft.com/aml.php
[57]	Szedula J.A., "FASTPASS - A tool for launch vehicle synthesis", AIAA 96-4051.
[58]	Carty A., "An Approach to Multidisciplinary Design, Analysis and Optimization For Rapid Conceptual Design", AIAA 2002-5438.
[59]	Moore K.T., Naylor B.A. and Gray J.S., "The Development of an Open Source Framework for Multidisciplinary Analysis and Optimization", AIAA 2008-6069.
[60]	Balling L.J. and Sobieszczanski-Sobieski J., "Optimization of Coupled Systems: A Critical Overview of Approaches", AIAA Journal, Vol. 34, No. 1, pp. 6-17, January-February 1996.
[61]	Ledsinger L.A. and Olds J.R., "Optimized Solutions for Kistler K-1 Branching Trajectories Using Multidisciplinary Design Optimization Techniques", Journal Of Spacecraft And Rockets, Vol. 39, No. 3, May-June 2002.
[62]	Vinko T., Izzo D., Bombardelli C., "Benchmarking different global optimisation techniques for preliminary space trajectory design", International Astronautical Congress, Paper IAC-07-C1.3.01, September 2007.
[63]	Deb K., "Evolutionary Algorithms for Multi-criterion optimization in Engineering Design", Evolutionary Algorithms in Engineering and Computer Science, 1st Edition, John Wiley & Sons, Chichester, UK, June 1999, pp. 135-161.
[64]	Nocedal J. and Wright S.J., "Numerical Optimization", 2nd Edition, Springer, New York, 2006.
[65]	Chachuat B., "Nonlinear and Dynamic Optimization: From Theory to Practice", 1st Edition, Automatic Control Laboratory, EPFL, Switzerland, 2007.
[66]	Haupt R.L. and Haupt S.E., "Practical Genetic Algorithms", 2nd Edition, Wiley, New Jersey, 2004.
[67]	Kennedy J., Eberhart R.C. and Shi Y., "Swarm Intelligence", Morgan Kaufmann Publishers, San Francisco, 2001.
[68]	Castellini F., "Global Optimization Techniques for Space Missions Design", MSc Thesis, Politecnico di Milano, April 2008, available at http://publishing.yudu.com/Freedom/Anpj3/FrancescoCastelliniS
[69]	Castellini F. and Lavagna M., "Comparative Analysis of Global Techniques for Performance and Design Optimization of

	Launchers", Journal of Spacecraft and Rockets, in print.
[70]	Coello Coello C. A., "An Updated Survey of Evolutionary Multi-Objective Optimization Techniques: State of the Art and Future Trends", Proceedings of the 1999 Congress on Evolutionary Computation-CEC99, IEEE, Washington, DC, July 1999.
[71]	Ingber L., "Simulated Annealing: practice versus theory", Mathematical Computing Modeling, Vol. 18, No. 11, December 1993, pp. 29-57.
[72]	Coello Coello C.A., Reyes-Serra M., "Multi-Objective Particle Swarm Optimizers: a survey of the state-of-the-art" International Journal of Computational Intelligence Research, Vol. 2, Issue 3, 2005.
[73]	Dorigo M., "Ant colony optimization theory: A survey", Theoretical Computer Science, Vol. 344, Issue 2-3, November 2005.
[74]	Wolpert D.H., and Macready W.G., "No free lunch theorems for optimization", IEEE Transactions on Evolutionary Computation, Vol. 1, Issue 1, April 1997.
[75]	Zitzler E., Deb K., and Thiele L., "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results", IEEE Transactions on Evolutionary Computation, Vol. 8, Issue 2, 2000.
[76]	Coello Coello C.A., Pulido G.T., and Lechuga M.S., "Handling multiple objectives with Particle Swarm Optimization", IEEE Transactions on Evolutionary Computation, Vol 8, Issue 3, 2004.
[77]	Eberhart R. and Kennedy J., "Particle Swarm Optimization", Proceedings of the IV IEEE International Conference on Neural Networks, IEEE, Washington, DC, 1995.
[78]	Clerc M. and Kennedy J., "The Particle Swarm - Explosion, Stability, and Convergence in a Multidimensional Complex Space", IEEE Transactions on Evolutionary Computation, Vol. 6, No. 1, February 2002.
[79]	Coello Coello C.A., Toscano Pulido G. and Lechuga M.S., "Handling multiple objectives with Particle Swarm Optimization", IEEE Transactions on Evolutionary Computation, Vol 8, Issue 3, 2004.
[80]	Knowles J.D., Corne D.W., "Approximating the Non-dominated Front Using the Pareto Archived Evolution Strategy", IEEE Transactions on Evolutionary Computation, Vol. 8, Issue 2, March 2000.
[81]	Deb K., Agrawal S., Pratap A., and Meyarivan T., "A fast elitist non-dominated sorting genetic algorithm for Multi-Objective Optimization: NSGA-II", Proceedings of Parallel Problem Solving from Nature VI Conference, Springer, New York, September 2000.
[82]	Dorigo M., Di Caro G. and Gambardella L. M., "Ant Algorithms for Discrete Optimization", Artificial Life, Vol. 5, No. 2, 1999.
[83]	Dorigo M. and Socha K., "Ant Colony Optimization for continuous domains", European Journal of Operational Research, Vol. 185, No. 3, 2008.
[84]	Najera A.G., "Extending ACOR to solve multi-objective problems", Proceedings of the 2007 UK Workshop on Computation Intelligence, London, UK, July 2007.
[85]	Nikolayzik T., Büskens C. and Gerdtsz M., "Nonlinear large-scale Optimization with WORHP", 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference, Fort Worth, Texas, September 2010, AIAA 2010-9136.
[86]	Sutton G.P., and Biblarz O., "Rocket Propulsion Elements", 7th Edition, John Wiley and Sons Inc., 2001.
[87]	Huzel D.K. and Huang D.H., "Design of Liquid Propellant Rocket Engines", NASA SP-125, 2nd Edition, 1972.
[88]	Miller W.H., "Solid Rocket Motors Performance Analysis and Prediction", NASA SP-8039, 1971.
[89]	Isakowitz S.J., "AIAA International Reference Guide to Space Launch Systems", 3rd Edition, AIAA, 1998.
[90]	Koelle D.E., "Handbook of Cost Engineering for Space Transportation Systems: TRANSCOST-Model 7.2 (Rev2007)", TransCostSystems (TCS), 2007.
[91]	Ricciardi A., "Complete Geometric Analysis of Cylindrical Star Grains", AIAA 89-2783.
[92]	Hartfield R., Jenkins R., Burkhalter J. and Foster W., "A Review of Analytical Methods for Solid Rocket Motor Grain Analysis", AIAA 2003-4506.
[93]	Gordon S. and Mc Bride B.J., "Computer Program for Calculation of Complex Chemical Equilibrium Compositions and Applications - Volume I: Analysis", NASA Reference Publication 1311, October 1994.
[94]	Gordon S. and Mc Bride B.J., "Computer Program for Calculation of Complex Chemical Equilibrium Compositions and Applications - Volume II: Users Manual and Program Description", NASA Reference Publication 1311, June 1996.

[95]	Hoggard L. and Leahy J. "Liquid Engine Design: Effect of Chamber Dimensions on Specific Impulse", NASA report M09-0679, 2009.
[96]	Mehta G., Stone W. and Ingram C., "Comparative Testing of Russian Kerosene and RP-1", AIAA 1995-2962.
[97]	O'Leary R.A. and Beck J.E., "Nozzle Design", Pratt & Whitney Rocketdyne's Engineering Journal of Power Technology, Spring 1992, available at www.pwrengineering.com/articles/nozzledesign.htm
[98]	Stark R.H., "Flow Separation in Rocket Nozzles, a Simple Criteria", 41st AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit, Tucson, Arizona, July 2005, AIAA 2005-3940.
[99]	ESA CDF Study Report, "Heavy Lift Launch Vehicle", Report CDF-36(A), April 2005.
[100]	Rohrschneider R.R., "Development of a Mass Estimating Relationship Database for Launch Vehicle Conceptual Design", AE 8900 Paper, Georgia Institute of Technology, April, 2002.
[101]	Whitehead J.C., "Single Stage to Orbit mass budgets derived from propellant density and specific impulse", AIAA 1996-3108.
[102]	Craidon C.B., "A Description Of The Langley Wireframe Geometry Standard (Lawgs) Format", NASA TM-85767, February 1985.
[103]	Vukelich S.R., Stoy S.L., Burns K.A. et al., "Missile DATCOM, Volume I - Final Report", McDonnel Douglas Technical Report AFWAL-TR-86-3091.
[104]	Hammond W.E., "Design Methodologies for Space Transportation Systems", AIAA Educational Series, 2001.
[105]	Blake W.B., "Missile DATCOM Users Manual - 1997 Fortran 90 Revision", Air Force Research Laboratory AFRL-VA-WP-TR-1998-3009.
[106]	Blake W.B., "Missile DATCOM: 1997 Status And Future Plans", AIAA 97-2280.
[107]	Sooy T.J. and Schmidt R.Z., "Aerodynamic Predictions, Comparisons, and Validations Using Missile DATCOM (97) and Aeroprediction 98 (AP98)", Journal Of Spacecraft And Rockets, Vol. 42, No. 2, March-April 2005.
[108]	Glatt C.R., "WAATS - A computer program for Weights Analysis of Advanced Transportation Systems", NASA Contractor Report CR-2420, December 1974.
[109]	Bruhn E.F., "Analysis and Design of Flight Vehicle Structures", Tri-State Offset Company, Orlando, 1973.
[110]	Bruhn E.F., Orlando J.I. and Meyers J.F., "Analysis and Design of Missile Structures", Tri-State Offset Company, Orlando, 1967.
[111]	Wild P.M. and Vickers G.W., "Analysis of filament-wound cylindrical shells under combined centrifugal, pressure and axial loading", Composites Part A 28A (1997) 47-55, 1996.
[112]	Castellini F., Lavagna M. and Erb S.: "Concurrent optimization of launcher architectures and ascent trajectories with global optimization algorithms". 58th International Astronautical Conference, Glasgow, October 2008.
[113]	Castellini F. and Lavagna M., "Comparative Analysis of Global Techniques for Performance and Design Optimization of Launchers". Accepted for publication on the *Journal of Spacecraft and Rockets*.
[114]	International Organization for Standardization, "Flight Dynamics – Concepts, Quantities and Symbols", International Standard ISO 1151, 1988
[115]	American National Standards Institute, Recommended Practice for Atmospheric and Space Flight Vehicle Coordinate Systems, American National Standard ANSI/AIAA R-004-1992, Feb. 1992
[116]	Detra R.W., Kemp N.H. and Riddell F.R., "Addendum to heat transfer to satellite vehicles reentering the atmosphere", Jet Propulsion, Vol. 27, No. 12, December 1957.
[117]	Bradford J.E., Charania A., and Germain B.St., "REDTOP-2: Rocket Engine Design Tool Featuring Engine Performance, Weight, Cost, and Reliability", AIAA 2004-3514.
[118]	Ariane 5 User's Manual (Issue 5, Revision 1 – July 2011), available at www.arianespace.com
[119]	VEGA User's Manual (Issue 3, Revision 0 – March 2006), available at www.arianespace.com
[120]	Shi Y. and Eberhart R.C., "Empirical Study of Particle Swarm Optimization", Proceedings of the 1999 Congress on Evolutionary Computation (CEC-99), Washington, July 1999.
[121]	Van Den Bergh F., "An analysis of Particle Swarm Optimizers", PhD dissertation, University of Pretoria, Nov. 2001.
[122]	Trelea I.C., "The particle swarm optimization algorithm: convergence analysis and parameter selection", Information

	Processing Letters, No. 85, 2003.
[123]	"ATK Space Propulsion Products Catalog", May 2008, available at www.atk.com.
[124]	Kawahara G. and McKleskey S.F., "Titanium Lines, Composite Overwrapped Pressure Vessels", AIAA Paper 1996-2751.
[125]	"Turbopump systems for Liquid Rocket Engines", NASA SP-8107, August 1974.
[126]	Szabo S., Berns J.A. and Stofan A.J., "Centaur Launch Vehicle Propellant Utilization System", NASA TN D-4848, October 1968.
[127]	Inter-Agency Space Debris Coordination Committee, "Space Debris Mitigation Guidelines", October 2002
[128]	Behruzi P., Michaelis M. and Khimeche G., "Behavior of the Cryogenic Propellant Tanks during the First Flight of the Ariane 5 ESC-A Upper Stage", AIAA 2006-5052.
[129]	"Terrestrial Environment (Climatic) Criteria Handbook For Use In Aerospace Vehicle Development", NASA-HDBK-1001, August 2000.
[130]	Barret C., "Design of Launch Vehicle Flight Control Augmentors and Resulting Flight Stability and Control", NASA Technical Paper 3704, August 1997.
[131]	Gridded Population of the World - version 3 (GPWv3) website at http://sedac.ciesin.columbia.edu/gpw/index.jsp
[132]	NASA's freeware Debris Assessment Software (DAS) website at http://orbitaldebris.jsc.nasa.gov/mitigate/das.html
[133]	Ardema M.D., Chambers M.C., Patron A.P., Hahn A.S, Miura H., and Moore M.D., "Analytical Fuselage and Wing Weight Estimation of Transport Aircraft", NASA Technical Memorandum 110392, May 1996.
[134]	Hutchinson V.L., "Analytical Structural Weight Estimation of Conceptual Launch Vehicle Fuselage Components with the Georgia Tech Structural Tool for Rapid Estimation of Shell Sizes (GT-STRESS)", MSc Thesis, Georgia Tech, May 2004.
[135]	Hutchinson V.L., and Olds J.R., "Estimation of Launch Vehicle Propellant Tank Structural Weight Using Simplified Beam Approximation", 40th IAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Fort Lauderdale, Florida, July 2004, AIAA 2004-3661.
[136]	Crawford R.F., and Burns A.B., "Minimum Weight Potentials for Stiffened Plates and Shells", AIAA Journal, Vol. 1, No. 4, April 1963.
[137]	Shanley F.R., "Weight Strength Analysis of Aircraft Structures", Second Edition, New York: Dover Press, 1960.
[138]	Cavagna L., Da Ronch A., and Ricci S., "NeoCASS Next generation Conceptual Aero Structural Sizing", 6th European Framework - SimSAC Report, February 2009.
[139]	Bianchi S., Bonnet M. and Trippi A., "Vega Launch Vehicle Propulsion Systems - An Overview of the 2004 Development Status", AIAA-2004-4212.
[140]	Fischer H., Fransen S., Thirkettle A. and Kiryenko S., "A Dynamic Analysis Tool for Europe's Small Launcher VEGA", European Conference on Spacecraft Structures, Materials and Mechanical Testing, Noordwijk, Netherlands, May 2005.
[141]	Carnevale C. and Resta P.D., "Vega Electromechanical Thrust Vector Control Development", 43rd AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit, Cincinnati, OH, July 2007, AIAA 2007-5812.
[142]	Pilchen, G. M., Breteau J., Caruana J., Kauffmann J., Ramusat G., Sirbi A. and Tumino G., "Future Launchers Preparatory Programme (FLPP) – Preparing For The Future Through Technology Maturation And Integrated Demonstrators: Status And Perspectives", 59th International Astronautical Congress, Glasgow, UK, October 2008.
[143]	Letourneur, Y., Leleu F., Pinard D. et al., "Status of Next Generation Expendable Launcher concepts within the FLPP programme", 59th International Astronautical Congress, Glasgow, UK, October 2008.
[144]	Nasseri S.A., "Development Of An Algorithm And An Integrated Program For The Preliminary Sizing Of Liquid Propellant Rocket Engines", IAC- 11.C4.1.12, 62nd International Astronautical Congress, Cape Town, October 2011.
[145]	Casiano M.J., Hulka J.R. and Yang V., "Liquid-Propellant Rocket Engine Throttling: A Comprehensive Review", Journal of Propulsion and Power, Volume 26, Issue 5, 2010.
[146]	Collier C., Yarrington P., Pickenheim M. and Bednarcyk B., "An Approach to Preliminary Design and Analysis", 48th Structures, Structural Dynamics, and Materials Conference, Honolulu, Hawaii, April 2007, AIAA 2007-2176.
[147]	Makinen R., Neittaanmaki A. Periauxb J., Sefriouibc M. and Toivanena J., "Parallel genetic solution for multiobjective MDO", 1996

[148]	Kandil A. and El-Rayes K., "Parallel genetic algorithms for optimizing resource utilization in large-scale construction projects", Journal of Construction Engineering and Management, May 2006.
[149]	Branke J., Deb K. and Reddy M., "Parallelizing Multi-Objective Evolutionary Algorithms: Cone Separation", KanGAL Report Number 2004017, 1996.
[150]	Streichert F., Ulmer H. and Zell A. "Parallelization of multi-objective evolutionary algorithms using clustering algorithms", 2005.

Annex 1

COMPUTATIONAL TIMES

An overview of the computational effort required for the different launch vehicle analyses is reported in **Table A1**. All computational times are referred to a single processor of type ***Intel Core2 Duo T6500 processor (2.10 GHz, 2 MB L2 Cache, 800 MHz FSB)*** with ***4096 MB DDR2 (800 MHz) RAM***. OpenMP parallel implementation of all global algorithms allows for approximately linear speed-up in both standalone global trajectory optimization and global MDO process. No parallelization scheme is instead available for the MDA, local trajectory optimizations or MDO local refinements (both with WORHP). Data are reported in **Table A1** for both the cases including and not including the writing to file of the MDA outputs. The first is only used in stand-alone MDA processes or as final post-processing of MDOs, involving also visualization and printing of the geometry as well as generation of the detailed aerodynamics, trajectory, structural, cost and reliability output files. This mode therefore requires much longer times than the case without output printing, which is actually used within the MDO processes.

Computational efforts for a MDA depend on the architecture of the launcher being investigated and the intended target orbit. The number of propelled components to be analyzed determines the computational time for all disciplines, in particular the number of executions of the external code CEA, which largely affects (20-30%) the total MDA time for V1 models. On the other hand, the final orbit to be reached determines the duration of the ascent trajectory to be integrated, hence the related computational time, which is of the same order of magnitude of the full design cycle in V1. In V2 instead, iterations of the Trajectory-Structures disciplines drive the computational times, although 2 iterations are sufficient in most of the cases with 1% tolerances on the inert masses. For V2, CPU times are referred to an analysis configuration which was chosen as a good compromise between accuracy and efficiency. In particular, several analysis upgrades can be selected, that result in larger computational loads: 1) full ballistic analysis for the SP motors, including a loop of CEA executions for each pressure at which the specific impulse needs to be evaluated. This allows for a limited accuracy enhancement at the price of a 3 to 6 seconds increase in CPU time for each MDA. 2) Analysis of more aerodynamic configurations, requiring multiple calls to Missile DATCOM and thus multiplying Aerodynamics' computational effort. 3) Simulation of the reentry trajectories with full trajectory integration rather than Keplerian model. This involves respectively 95% and 110% increase in the trajectory model execution for Ariane 5 ECA and VEGA. Since however the impact ellipse and hence safety constraint are often significantly affected, CPU times with full reentry integration are also shown in **Table A1** within brackets in the Trajectory and Total columns.

Summarizing, a full MDA cycle within a MDO process requires ***0.5-0.6 seconds*** with V1 models, ***0.8-0.9 seconds*** for V2-IDF models and ***1.2-1.4 seconds*** for V2-MDF models. The total run time for MDO processes can be estimated by multiplying these figures for the total number of foreseen MDA evaluations (i.e. number of particles by number of iterations), with a ***0.8-0.9*** correction factor accounting for a percentage of runs which are not executed to completion due to infeasibility. A detailed review of the computational effort required for standalone trajectory optimization, including a comparison of global (PSO) and local (WORHP) approaches, is instead reported in **[A11]**.

Model	Case	Write	Prop.	Geom.	Aero.	WCR	Trajectory	Struct.	Total excl. trajectory	Total (no iter.)	Total (2 iter.)	Total (3 iter.)
V1	Ariane 5	no	0.2 s	0.0 s	0.2 s	0.0 s	0.1 s (-)	-	0.4 s	**0.5 s**	-	-
V1	VEGA	no	0.3 s	0.0 s	0.1 s	0.0 s	0.1 s (-)	-	0.4 s	**0.5 s**	-	-
V1	Ariane 5	yes	0.2 s	2.9 s	0.2 s	0.0 s	0.4 s (-)	-	3.3 s	3.7 s	-	-
V1	VEGA	yes	0.3 s	1.4 s	0.1 s	0.0 s	0.4 s (-)	-	1.8 s	2.2 s	-	-
V2	Ariane 5	no	0.2 s	0.0 s	0.2 s	0.0 s	0.1 s (0.6 s)	0.4 s	0.8 s	**0.9 s** (1.4 s)	**1.4 s** (2.4 s)	**1.9 s** (3.4 s)
V2	VEGA	no	0.3 s	0.0 s	0.1 s	0.0 s	0.2 s (0.7 s)	0.2 s	0.6 s	**0.8 s** (1.3 s)	**1.2 s** (2.2 s)	**1.6 s** (3.1 s)
V2	Ariane 5	yes	0.2 s	3.3 s	0.3 s	0.0 s	0.7 s (2.1 s)	1.3 s	5.1 s	5.8 s (7.2 s)	7.8 s (12.6)	9.8 s (14.0)
V2	VEGA	yes	0.3 s	1.5 s	0.2 s	0.0 s	0.7 s (2.2 s)	0.4 s	2.4 s	3.1 s (4.6 s)	4.2 s (7.2 s)	5.3 s (9.8 s)

Table A1: Computational times for MDA processes of Ariane 5 ECA and VEGA launch vehicles, including spread between design disciplines (WCR=Weights, Costs, Reliability) and trajectory integration, also showing different cases for number of Trajectory-Structures iterations.

Annex 2

OPEN-MP PARALLELIZATION OF POPULATION-BASED GLOBAL ALGORITHMS

In the last decades, High Performance Computing (HPC) has taken a prominent role in most scientific fields. CFD or FEM are the most typical engineering applications requiring parallel deployment, but countless other simulation or optimization processes benefit from the exploitation of multi-core architectures. When the computational requirements are not excessively large, shared-memory machines are often the ideal choice in terms of such architectures, although the term HPC does not properly applies due to the limited speed-up potential. In fact, standard hardware already available, or at least available at a low cost, can be used to sensibly reduce computational times. Besides, the effort required for parallel coding of scientific programs in shared memory applications is much more limited than in distributed memory architectures, as in modern computer clusters. In this context, *OpenMP*[62] is a widespread API for shared memory multi-thread programming in C/C++ and Fortran, which allows to deploy parallel applications on dual-core/quad-core laptops or desktops as well as on more performing workstations (typically up to 32 cores). When compared to "real" HPC paradigms, such as *MPI* for distributed memory clusters or *CUDA* and *OpenCL* for GPU computing, the main advantage of OpenMP lays in the very limited amount of required parallelization effort, resulting in large coding time savings.

In another area of computer science, global optimization algorithms were born to complement the traditional gradient-based optimization techniques, which are known to converge to local optima for problems characterized by large and non smooth search spaces, multiple objectives and/or discrete variables. Global methods, usually based on the mimicking of natural phenomena (Genetic Algorithms, Particle Swarm Optimization, ...), are capable of solving these problems, but require very high numbers of function evaluations. This prevents their application to models with long evaluation times, unless parallel computing is exploited. Fortunately, parallel implementation of global algorithms is as much straightforward as it is critical for their practical applicability to large problems. In fact, most global techniques are population-based, involving the contemporary evaluation of a large number of solutions. Only the values of optimization variables, constraints and objectives need to be exchanged among the processes, and communications only occur when the evolution of the population is carried out at each iteration. This results in an intrinsically parallel algorithmic architecture, characterized by almost linear speed up, so that population-based optimization algorithms have been classified among the *"embarrassingly parallel"* problems.

Due to constraints in both hardware resources and available time, the presented research could not exploit HPC platforms, but it was decided to combine the simplicity of OpenMP and the suitability of population-based methodologies to parallelization for at least exploiting the dual- and quad- core architectures of modern personal computers. Parallel exploitation of global optimization algorithms is a very cost-efficient approach, enabling almost linear speed-up for any applicative scenario with an extremely limited effort in terms of both man-hours and hardware costs. Although several more complex parallelization schemes have been developed (see **[147]**, **[148]**, **[149]** and **[150]** for examples of diffusion, island, cone-separation and clustering models, whereas a short review can be found in **[69]**), the most straightforward master-slave model was implemented for its simplicity and the more than satisfactory obtained speed-up.

Figure A1 presents the most generic logic scheme applicable to all population-based GO algorithms, showing how the model evaluation block (i.e. evaluation of the N_S solutions to be performed at each iteration) is common to all techniques. This is the only part of the code involved in the parallelization with the Master-Slave model, which consists in performing all algorithmic operations with the master thread only and only parallelizing the model evaluations' *for* cycle. As highlighted in **Figure A2**, this requires an extremely limited amount of new code, though allocation of the cycle resources needs some consideration. As a result, since basically all the computational effort is spent in the *for* cycle when the computational time for each model evaluation exceeds ***0.01-0.1 s*** - which is the case for all applicative scenarios - the resulting speed-up is nearly linear (see **Figure A3** for an example of CPU usage in sequential and parallel executions).

[62] For the full specifications of OpenMP see www.openmp.org, whereas for useful tutorials see for example http://openmp.org/mp-documents/ntu-vanderpas.pdf and https://computing.llnl.gov/tutorials/openMP .

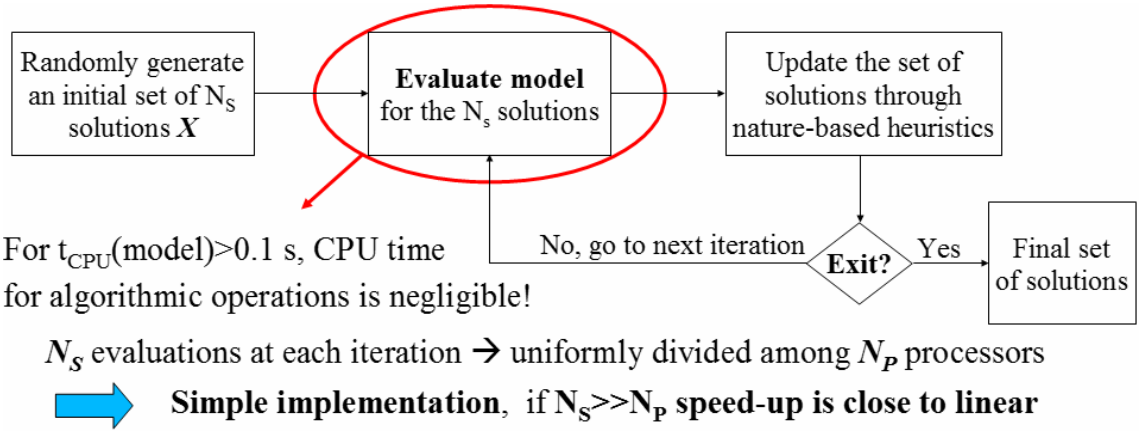


Figure A1: Master-Slave parallelization model for population-based GO algorithms, only requiring parallelization of a for cycle.

Sequential code

```
Eigen::VectorXd var(m_NVAR), objconst(m_NOBJ+m_NCONS);
double penalty=0;




for(int i=0;i<popsize;i++)
    {
    cout << "Iteration " << iternumber << ", particle


    //1. Copy current solution to temp. vector:
    for(int j=0;j<m_NVAR;j++)
        var(j)=pop[i][j];
    //2. Evaluate model:
    objconst = m_model(var);
    //3. Copy cstrs from tmp vector & eval penalty:
    penalty=0;
    for(int j=0;j<m_NCONS;j++)
        {
        Cpop[i][j]=objconst[j+m_NOBJ];
        penalty+=constrViolation(objconst[j+m_NOBJ],j
        }
    //4. Copy objs from tmp vector & apply penalty:
    for(int j=0;j<m_NOBJ;j++)
        Jpop[i][j]=objconst[j]+penalty*hugeCstrVio();
    }
```

OpenMP parallel code

```
Eigen::VectorXd var(m_NVAR), objconst(m_NOBJ+m_NCONS);
double penalty=0;
int chunk=1;
#pragma omp parallel default(shared) private(var,objconst,penalty)
    {       //BEGIN PARALLEL REGION//
    var.resize(m_NVAR); objconst.resize(m_NOBJ+m_NCONS);
    int nthreads=omp_get_num_threads();
    #pragma omp for schedule(dynamic,chunk)
        for(int i=0;i<popsize;i++)
            {
            if(nthreads==1) cout << "Iteration " << iternumber <<
            else            cout << "Iteration " << iternumber <<
                                    ", processor " << omp_get_thre
            //1. Copy current solution to temp. vector:
            for(int j=0;j<m_NVAR;j++)
                var(j)=pop[i][j];
            //2. Evaluate model:
            objconst = m_model(var);
            //3. Copy cstrs from tmp vector & eval penalty:
            penalty=0;
            for(int j=0;j<m_NCONS;j++)
                {
                Cpop[i][j]=objconst[j+m_NOBJ];
                penalty+=constrViolation(objconst[j+m_NOBJ],j);
                }
            //4. Copy objs from tmp vector & apply penalty:
            for(int j=0;j<m_NOBJ;j++)
                Jpop[i][j]=objconst[j]+penalty*hugeCstrVio();
            }
    }       //END PARALLEL REGION//
```

Figure A2: Sequential and parallel C++ codes for the model evaluation loop applicable to any population-baseed GO algorithms.

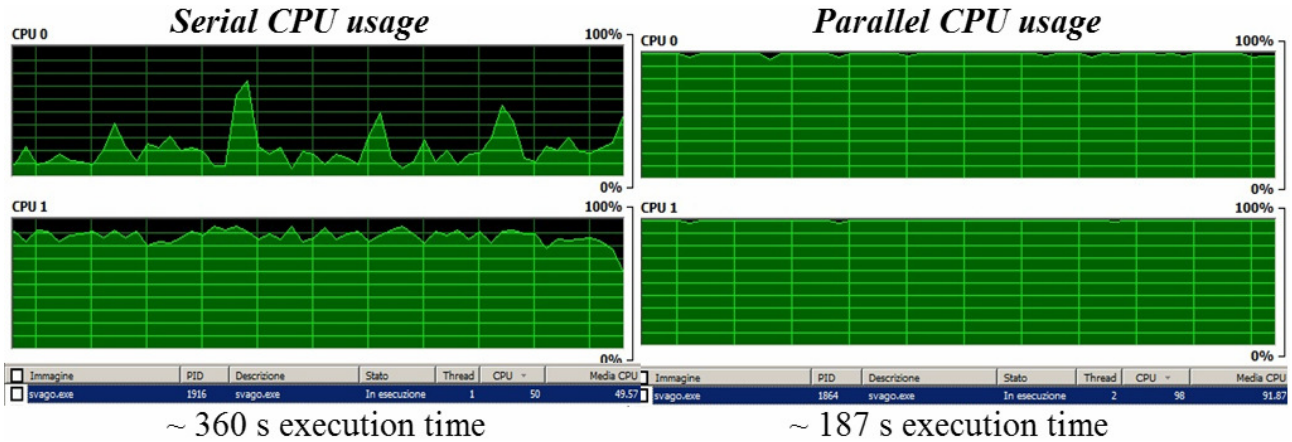


Figure A3: Example of CPU usage for sequential and parallel execution of PSO-1D on mathematical benchmark cases with synthetic increase of t_{CPU}*(model)* through a *0.1s* "wait" instruction, showing a 1.93 speed up for the 2 available processors.

Annex 3

SOFTWARE ARCHITECTURE AND USER INTERFACE

The MDO environment described within this PhD dissertation was developed in C++ making heavy use of Object Oriented programming for ensuring flexibility and modularity, resulting in the Space Vehicles Analysis and Global Optimization (SVAGO) tool. The software's architecture is described in details in the Architectural Design Document (**[A8]**) whereas the GUI and execution details are given in the User's Manual (**[A13]**). A synthesis of the main aspects of the software is however given here (see **Figures A4**, **A5** and **A6** for the main layers, architecture and GUI screenshot example):

- Input/output interfaces are handled trhough an *XML* database structure, which contains all information (user inputs, optimization variables, constraints and objectives, and all the disciplinary outputs) on the launch being designed.
- A *GUI* based on Qt technology was developed for the inputs definition in order to ease the set-up of MDO problems. The GUI automatically writes the XML file including information provided by the user, but is presently limited to V1 models, whereas manual editing of the XML file is necessary to include V2 settings.
- Information in the XML is automatically parsed to C++ objects defining a Central System Database (CSD), storing all data regarding the user input selections, mission parameters, vehicle design data, optimization settings and output results. All communications from one subsystem to another occur through this system database, so that no direct interfaces have to be defined among the subsystems. This hierarchic structure simplifies the implementation of the CSI, the modification/replacement of an existing subsystem and the addition of new subsystems, since only the new interfaces between the subsystem and the CSD along with the new execution rules for the CSI have to be defined.
- A Central System Intelligence (CSI) manages all information flow between subsystems and with the optimization algorithms, implementing the rules defined in the DSM of each Version to call the different disciplinary analysis in the correct order and to take/return data from/to the optimization algorithms.
- Multiple global and/or local runs with different options for better user control of the optimization process are allowed, also featuring a reverse communication protocol during optimizations which allws the user to inspect the current results before deciding whether to continue or stop the process.
- Numerical outputs of the design solution(s) returned by the MDA/MDO processes are inspected directly from the XML file(s), exploiting external XML visualizers such as Eclipse for enhanced readability.
- Main output messages are printed to screen in the GUI (with customizable level of detail), which also contains a window for 2D/3D geometry visualization from *3-View* and *Silhouette*. Matlab scripts are also used to automatically plot results from the disciplinary output files (e.g. Pareto fronts, trajectories, structural loads, aerodynamics ...).

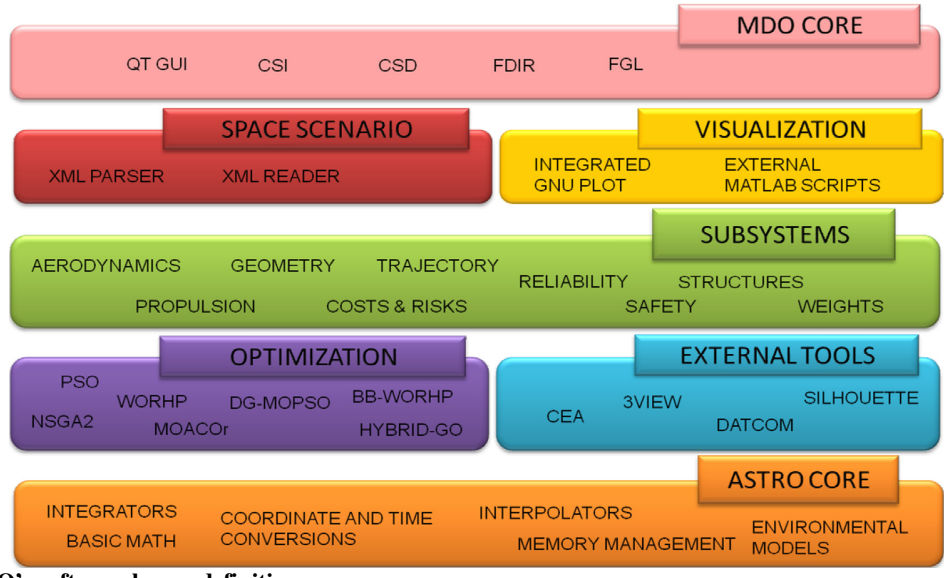


Figure A4: SVAGO's software layers definition.

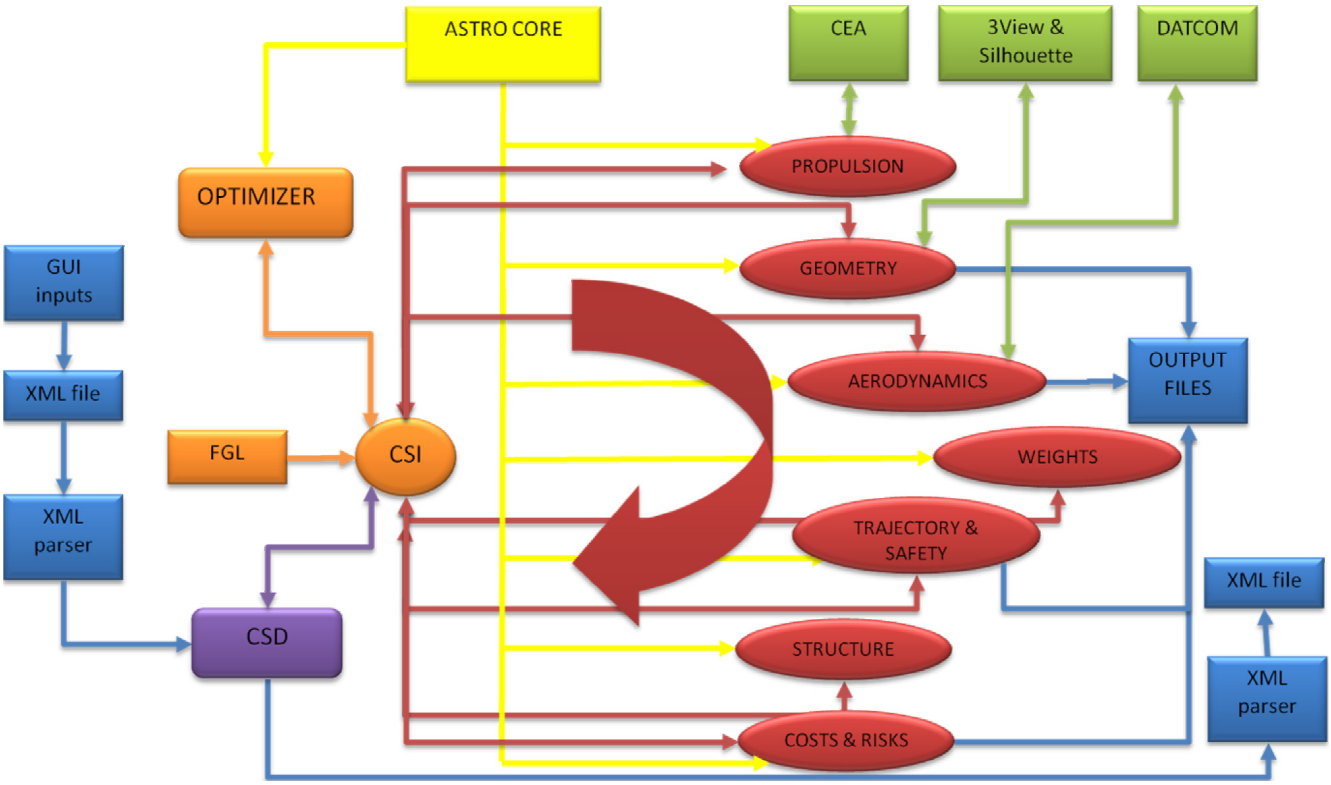


Figure A5: SVAGO's main software architecture. CSI: Central System Intelligence. CSD: Central System Database.

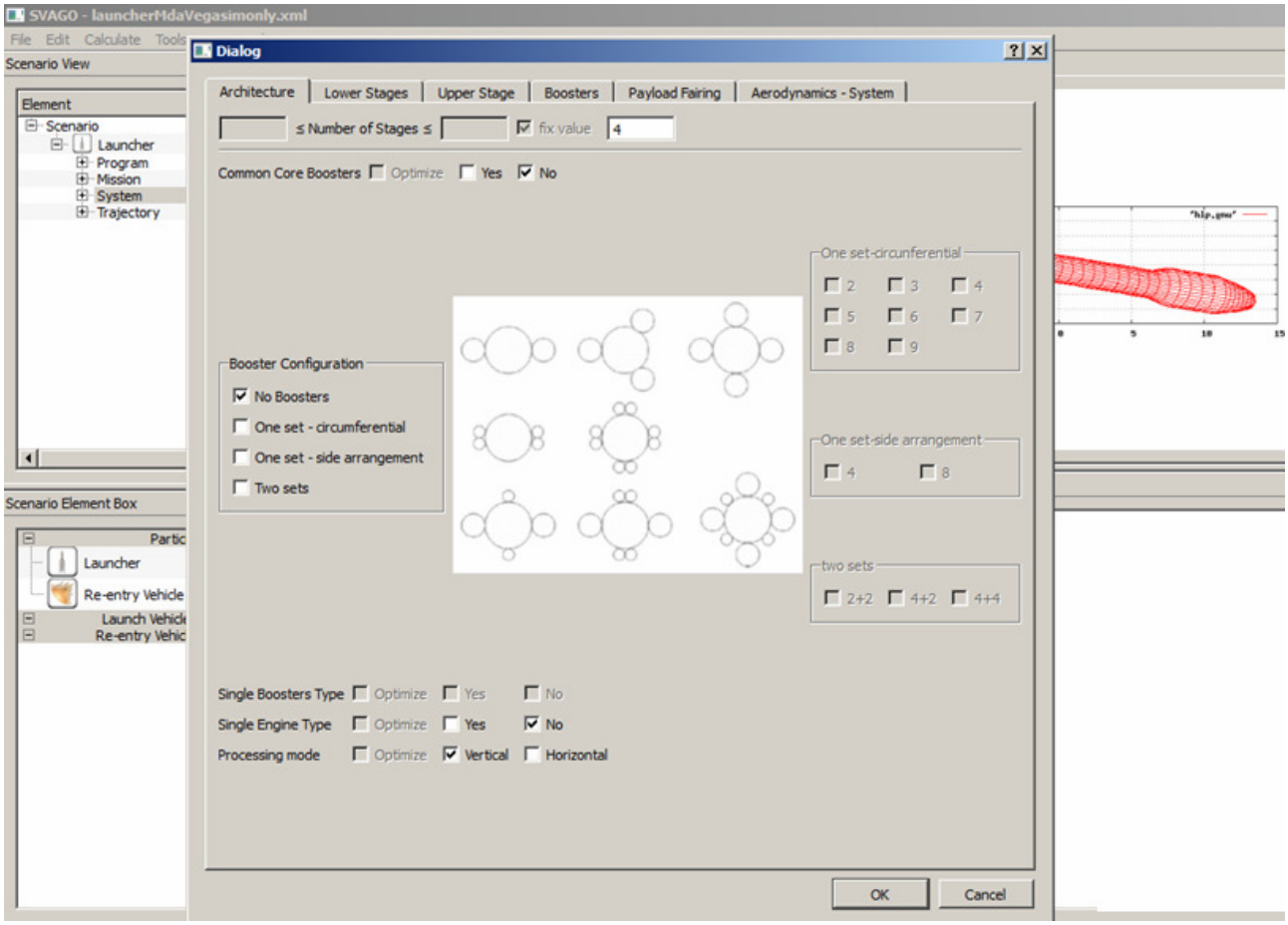


Figure A6: Example of GUI window from SVAGO software tool