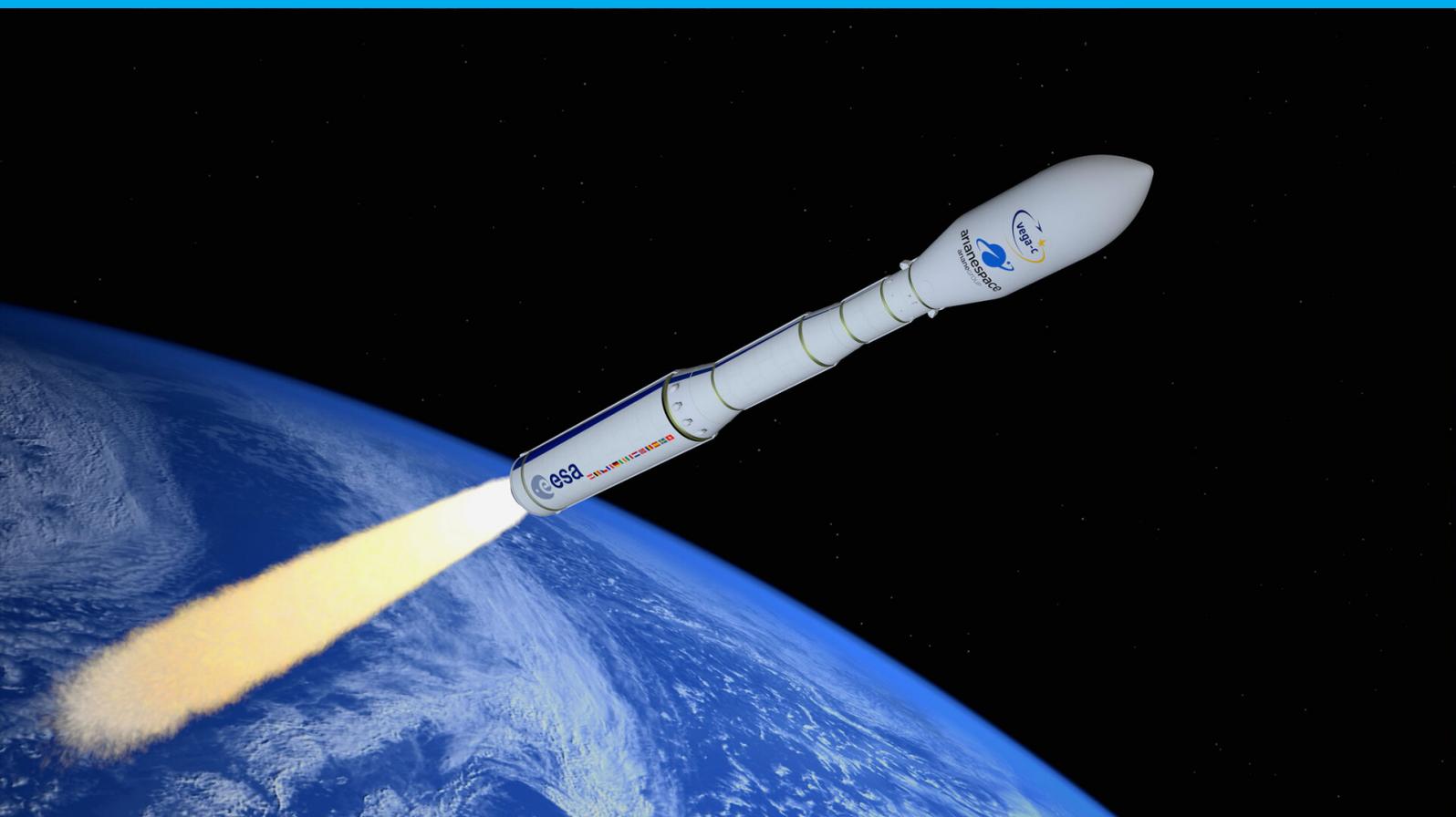


# System Studies and Re-entry Algorithms for reusable VEGA variant

Feasibility of return to launch  
site operation through vehicle  
selection and trajectory opti-  
mization

Alessandro Maria Laspina





# System Studies and Re-entry Algorithms for reusable VEGA variant

Feasibility of return to launch site operation  
through vehicle selection and trajectory  
optimization

by

Alessandro Maria Laspina

to obtain the degree of Master of Science  
at the Delft University of Technology,  
to be defended publicly on Tuesday August 30, 2022 at 9:30 AM.

Student number: 4805003  
Project duration: October 1, 2020 – August 2, 2022  
Thesis committee: Prof. dr. ir. M. Naeije, TU Delft, supervisor  
Dr. Ir. E. van Kampen, TU Delft  
Dr. E. Schrama, TU Delft

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

1

---

<sup>1</sup>Cover image from: <https://www.avio.com/sites/avio.com/files/styles/articles/public/2020-01/Vega%20C.jpg?itok=iQ2ymkn9>



# Preface

This Master's thesis on system studies and re-entry algorithms for a reusable VEGA variant covers topics of heuristic, and convex optimization. It has been completed to fulfill graduation requirements at the Delft University of Technology.

The project was undertaken with collaboration from AVIO, specifically Christophe Roux, who was influential in setting the direction of the research topic. Finalizing the research goals where difficult, but ultimately a reasonable scope was identified. Christophe was important in the completion of this thesis and therefore I would like to extend my thanks to him.

This thesis has been easily one of the most difficult projects in my life so far to conclude. I say that not because the topic may be extremely complex, but because of the turbulent period in which it was written in. During the COVID pandemic life was not easy and many changes in my personal life set back the work by a year. That being said, it has been the most worthwhile project in my academic career and even more so in my life, as there was no better way to learn the lesson I did in this period rather than in my future where I have to mention a special thanks to my family; you where always there besides me and at times where the only ones. Lastly, for an individual to me that needs no introduction, Marc Naeije deserves equally special recognition for always believing in me and helping me see the light at the end of tunnel. His role as a thesis supervisor does not do him justice, as he is an exemplary individual who at times provided more guidance than some of the algorithms written in this paper. Most of all, as one can tell, he also taught me that having a little bit of levity in a situation is not harmful, and there are far more important things than trying to stand out. Without him, this thesis would not have been possible.

To the reader, I hope that this document will be be a good read, and that it may serve future students well as a point of reference.

*Alessandro Maria Laspina*  
*Rome, August 2022*



# Abstract

RLV development can be considered as the modern step towards mission design due to financial and strategic decisions. In the past, reusability has been addressed however the level of maturity of the technology, both in terms of hardware and software was not yet reached. There are several aspects to developing a RLV, and these can be categorized into optimization of the LV, optimization of the trajectory, and cost analysis. To be able to determine the feasibility of the mission, it is not just necessary to develop a suitable configuration, but to also determine the physical feasibility of the trajectory. Several methods exist of which convex optimization is selected. This class of algorithms have risen in popularity in the regime of powered descent guidance, and present a desirable trade-off between performance and computational cost. An already existing algorithm, DESCENDO, for a two-staged vehicle CALLISTO purposed for a mission to a geo-synchronous orbit, is taken as reference. The algorithm is rewritten in YALMIP allowing it to perform more efficiently by saving computation time through creation of multiple controllers based on a discretized burn schedule. A potential candidate for reusability in the future is selected, which is a VEGA variant, considered as a two-staged vehicle with set requirements on the mission and configuration. Through closed-loop simulations, the feasibility of RTLS for a particular mission of this VEGA variant can be studied. The disciplines involved in the study include the launch vehicle optimization, engine sizing, preliminary ascent & descent, and 3-DoF simulations. Previous research at TU Delft on RLV has included the work Rozenmeijer, Vandamme, Van Kesteren, Miranda, and Contant, graduate students of the TU Delft Aerospace Engineering faculty. The work relied on the usage of the TUDAT C++ software environment and based its feasibility or reusability of operations through a cost-analysis. A shift in direction is taken away from cost-analysis to examine at a greater detail the physical feasibility of the trajectory for a nominal candidate RLV. This is done by examining the influence of simulator to guidance algorithm dynamics and guidance algorithm parameters. Moreover, a nominal payload class between 100 and 500 kg is selected to determine the configuration of the vehicle ideal for this mission. To be able to determine feasibility of RTLS, three metrics are considered, which are the final landing velocity, final landing position, and maximum dynamic pressure.

The study performs higher fidelity analysis only on the return phase, and as such the starting conditions for descent are determined through a preliminary design process by considering a drag-less ascent. This returns a starting altitude of around 26-30 km, with similar values for starting downrange position, and varying conditions of initial mass and velocity. For the preliminary descent, it is found that the metric of dynamic pressure does not reach more than around 60% of the limit imposed by the VEGA-C, which is similar for other VEGA variants. This coincides with research done with the CALLISTO vehicle. All in all, the best cases for these metrics and one included as an overall best case where candidates for the 400 kg payload class. The selection criteria for best cases of the preliminary descent involved the dynamic pressure, final velocity, and required propellant mass for descent. Moreover, the vehicle optimization results showed that this contained the most variation of vehicle characteristics, and as such it was deemed as a desirable class to work with for its flexibility in design. The best case burnt mass result was selected as the best case velocity required extensive propellant mass to burn for only a less than 7 m/s difference in result, which would not be indicative of what the convex algorithm could achieve due to dynamics involved in the preliminary experiment. This nominal candidate is then tested for various variations of controller tuning parameters combinations, burn schedules, and simulator/guidance frequencies. The results showed a clear desirable region for the final time of just between 300 and 310 seconds for return, favouring shorter burns. The solution envelope for the burn schedule showed gaps in zones of feasibility as well as optimality, suggesting some performance issues with the algorithm due to it failing to find solutions. Nevertheless this envelope is well defined and several feasible solutions existed. The influence of parameter tuning and simulator frequency was studied. It was determined that no set of guidance parameters could give an advantage over the other, but that some values did favour feasibility more. This is somewhat in contrast to the selection of frequencies, as despite the fact there was

also a large difference for higher frequency ratios between the Q3 to Q4 and Q0 to Q2, there is a noticeably trend that higher ratios are favoured. Moreover, a local optimal ratio of frequency of 10-1 was also found, and being the same ratio used for the other experiments as well as the CALLISTO study, provides more evidence that this effect is intended.

The uncertainties studied are for the initial state variations, errors in reading of position and velocities, and process time delays. It was noted that almost all the errors in the initial state variations shared similar distributions for ranges of values of the metrics. The overall majority returned feasible as well as the large minority of this returned optimal. Of little to no significance was the processing time delay, of which the overwhelming majority returned optimal results, and the rest where outliers whose process time factor where beyond the  $3\sigma$  limit imposed in the creation of normal random variables. The largest errors arose from the real-time uncertainty in the velocities and position, modelled after pseudo-range errors. Although results showed a high density in the feasible and optimal regions for metrics of final time and position, there was also a high density past the feasible regions. It can be considered that the feasibility of RTLS operations for such a VEGA vehicle is restricted by such errors as expected, but nevertheless results are promising in what can be the main steps to lead to an error analysis study by the use of state estimation techniques and testing various modifications made to the SOCP problem to improve performance.

# List of Abbreviations

ACT	Actuator
CEA	Chemical Equilibrium for Applications
CFD	Computational fluid dynamics
CFRP	Carbon fiber reinforced polymer
COESA	Committee on Extension to the Standard Atmosphere
CVX	Convex
DG	Descent guidance
DL	Dead-leaf
DoE	Design-of-experiment
DoF	Degrees-of-freedom
DRL	Down-range landing
ECEF	Earth Centered Earth Fixed frame
ECI	Earth Centered Inertial frame
ELV	Expendable launch vehicle
FB	Fly-back
GA	Genetic algorithm
GC	Guidance & control
GLOW	Gross lift-off weight
GNC	Guidance navigation control
GUI	Guidance
LP	Launch Pad frame
LV	Launch vehicle
MDF	Multidisciplinary design feasible
MDO	Multidisciplinary design optimization
PR	Propulsion
RK	Runge-Kutta
RLV	Reusable Launch Vehicle
RTLS	Return to launch site
SA	Standard atmosphere
SDK	Spacecraft dynamics & kinematics
SF	Safety factor
SIM	Simulator
SN	Sensors & navigation
SOCP	Second order cone programming
TB	Toss-back
TR	Trajectory
TUDAT	TU Delft Astrodynamics Toolbox
VEB	Vehicle equipment bay
VTHL	Vertical take-off horizontal landing
VTVL	Vertical take-off vertical landing
ZEM	Zero-error miss
ZEV	Zero-error velocity
UTS	Ultimated Yield Strength
YTS	Tensile Yield Strength



# List of Symbols

Subscript/Superscript symbol	Definition
0	Initial
<i>a</i>	Value at ambient
<i>A</i>	Ascent
<i>adapter</i>	Payload adapter
<i>b</i>	Burn
<i>b,e</i>	Bottom elliptical section
<i>burn</i>	Burnt (propellant)
<i>c</i>	Value at chamber
<i>c</i>	Construction
<i>cb</i>	Common Bulkhead
<i>co</i>	Coasting
<i>cr,f</i>	Fuel tank crown
<i>cr,o</i>	Oxidizer tank crown
<i>culm</i>	Culmination
<i>cvx</i>	Convex
<i>D</i>	Downwards component
<i>D</i>	Descent
<i>descent</i>	Descent
<i>dry</i>	Dry (inert)
<i>E</i>	East component
<i>e</i>	Value at exit
<i>E</i>	E frame
<i>e,f</i>	Fuel tank ellipsoidal end
<i>e,o</i>	Oxidizer tank ellipsoidal end
<i>Earth</i>	Earth
<i>ECEF</i>	Earth centered Earth fixed frame
<i>ECI</i>	Earth centered inertial frame
<i>eff</i>	Effective
<i>eng</i>	Engine
<i>f</i>	Final
<i>f</i>	Fuel
<i>fairing</i>	Payload fairing
<i>gauge,min</i>	Minimum tank gauge
<i>go</i>	Remaining time (time to go)
<i>gui</i>	Pertaining to guidance
<i>inert</i>	Inert
<i>inter</i>	Interstage
<i>k,f</i>	Fuel tank knuckle
<i>k,o</i>	Oxidizer tank knuckle
<i>LP</i>	Launch Pad
<i>M</i>	Missile
<i>M10</i>	M10 engine
<i>max</i>	Maximum

<i>meas</i>	Measured value
<i>MX</i>	MX engine
<i>N</i>	North component
<i>oa</i>	Overall
<i>ox</i>	Oxidizer
<i>p</i>	DESCO threshold value
<i>p</i>	Propellant
<i>pay</i>	Total payload
<i>pl</i>	Payload instruments
<i>pred</i>	Predicted value
<i>press</i>	Pressurant gas
<i>pt</i>	Process time
<i>real</i>	Real value
<i>saved</i>	Saved in stage 1
<i>sea</i>	Sea level
<i>sim</i>	Pertaining to simulator
<i>stage</i>	Vehicle stage
<i>struc</i>	Vehicle structure
<i>te</i>	Parasitic tank effects
<i>theory</i>	Theoretical value
<i>tot</i>	Total
<i>TPS</i>	Thermal protection system
<i>u</i>	Useful
<i>vac</i>	Vacuum
<i>VEB</i>	Vehicle equipment bay

Greek Unit	Definition	Unit
$\alpha$	Angle of attack	[deg]
$\delta$	Latitude	[deg]
$\tau$	Longitude	[deg]
$\varepsilon$	Structural Index	-
$\lambda$	Payload ratio	-
$\epsilon$	Construction mass ratio	-
$\epsilon$	Noise error	Value dependent
$\Lambda$	Mass ratio	-
$\mu$	Root of quadratic payload ratio	-
$\mu_{Earth}$	Earth gravitational constant	[kg m/s <sup>2</sup> ]
$\mu$	Mean value	Value Dependent
$\sigma$	Standard Deviation	Value Dependent
$\sigma$	Thrust acceleration norm	[N]
$\rho$	Density	[kg/m <sup>3</sup> ]
$\gamma$	Specific heat ratio	-
$\Omega$	Earth rotation vector	[rad/s]
$\omega$	Earth rotation rate	[rad/s]
$\Delta V$	Velocity Change	[m/s]
$\theta_{max}$	Maximum thrust direction angle	[deg]
$\eta_w$	Thrust acceleration difference	[m/s <sup>2</sup> ]
$\phi$	Propellant mass ratio	-

Roman Symbol	Definition	Unit
<i>A</i>	Speed of sound	[m/s]
<i>A</i>	Area	[m <sup>2</sup> ]
<i>a</i>	Acceleration	[m/s <sup>2</sup> ]

$c^*$	Characteristic Velocity	[m/s]
$C$	Coefficient	-
$C_F$	Thrust coefficient	-
$D$	Diameter	[m]
$d^*$	Approximated drag acceleration	[m/s <sup>2</sup> ]
$F$	Force	[N]
$f$	Frequency	[Hz]
$g_0$	Gravity at sea level	[m/s <sup>2</sup> ]
$h$	Altitude	[m]
$I_{sp}$	Specific Impulse	-
$k$	Factor	-
$k_{ellipse}$	Ellipse ratio	-
$L$	Length	[m]
$\dot{m}$	Mass flow rate	[kg/s]
$m_M$	Molar mass	[kg/mol]
$M$	Mach number	-
$M$	Big-M thrust acceleration coefficient	[m/s <sup>2</sup> ]
$N$	Number of stages	-
$N$	Discretized grid length	-
$N_{cvx}$	Number of successive convexifications	-
$n_M$	Number of moles	-
$OF$	Oxidizer to fuel ratio	-
$P$	Pressure	[Pa]
$\mathcal{P}$	Pseudo-range	[m]
$q$	Dynamic Pressure	[W/m <sup>2</sup> ]
$q$	Binary vector	-
$r$	Position	[m]
$r$	Radius	[m]
$R$	Earth center to spacecraft distance	[m]
$R$	Gas constant	[J/(kg k)]
$R^*$	Universal gas constant	[J/(K mol)]
$S$	Surface integral	-
$S_{ref}$	Reference surface area	[m <sup>2</sup> ]
$SF$	Safety factor	-
$SI$	Structural Index	-
$t$	Time	[sec]
$t$	Thickness	[m]
$T$	Temperature	[deg]
$T$	Thrust	[N]
$T_s$	Time step	[sec]
$U_e$	Exit velocity	[m/s]
$v$	velocity	[m/s]
$V$	Volume	[m <sup>3</sup> ]
$w$	Thrust acceleration	[m/s <sup>2</sup> ]
$\mathcal{W}_{\eta_w}$	Thrust norm difference weight factor	-
$x$	Distance	[m]
$X_e$	Downrange Distance	[m]
$Y_e$	Crossrange Distance	[m]
$z$	Logarithmic mass	-



# Contents

Abstract	v
List of Abbreviations	vii
List of Symbols	ix
List of Figures	xvii
List of Tables	xix
1 Introduction	1
1.1 Stakeholder Interests . . . . .	1
2 Research Goals	3
2.1 Literature Study Summary . . . . .	3
2.2 Research Gap . . . . .	5
2.3 Research Objectives . . . . .	6
2.4 Research Question . . . . .	7
3 Methodology	9
3.1 Design Variables . . . . .	10
3.2 Optimizer . . . . .	10
3.3 Constraints . . . . .	11
3.4 Processing . . . . .	11
4 Flight Mechanics	13
4.1 Reference Frames . . . . .	13
4.1.1 Kinematic Equations of Motion . . . . .	15
4.2 Environment . . . . .	16
4.2.1 Dynamics Models . . . . .	16
4.2.2 Atmospheric Models . . . . .	18
4.2.3 Atmospheric Model Selection Motivation . . . . .	20
5 Launcher Configuration	21
5.1 Propulsion . . . . .	22
5.2 Stages . . . . .	24
5.2.1 Propellant Tank . . . . .	24
5.2.2 Pressure System . . . . .	25
5.3 Interstage . . . . .	26
5.4 Payload Bay . . . . .	26
5.4.1 Fairing . . . . .	26
5.4.2 Payload Adapter . . . . .	26
5.4.3 Vehicle Equipment Bay . . . . .	26
5.4.4 Payload . . . . .	27
5.5 Landing Legs . . . . .	27
5.6 Launch Vehicle . . . . .	27

6	Launch Vehicle Optimization	29
6.0.1	NASA CEA Runs	29
6.0.2	RLV sizing	30
6.0.3	Missile DATCOM Database	37
7	Preliminary Ascent and Descent	39
7.1	Preliminary Ascent	39
7.1.1	Multi-stage Rocket	39
7.1.2	Culmination Altitude	41
7.1.3	Final Optimization Routine	41
7.2	Preliminary Descent	42
7.2.1	Guidance Law	43
7.2.2	Closed-loop Simulation	44
7.2.3	Trade-off Space	45
8	Closed-Loop Simulations	47
8.1	Architecture	47
8.1.1	DG (Descent Guidance)	47
8.1.2	ACT (Actuators)	48
8.1.3	SDK (Spacecraft Dynamics & Kinematics)	49
8.1.4	SN (Sensors & Navigation)	49
8.2	Design of Experiment Campaigns	50
8.2.1	DoE1- Burn Schedule and Final Time	50
8.2.2	DoE2- Algorithm Parameter Tuning	50
8.2.3	DoE3- Guidance and Simulation Frequencies	51
8.2.4	DoE4- Sensitivity Analysis	51
9	Guidance Algorithm	55
9.1	Convex Optimization	55
9.1.1	Least Squares Problem	56
9.1.2	Linear Programming Problem	56
9.1.3	Convex Optimization Problem	56
9.1.4	Guidance Strategy	57
9.2	DESCO	58
9.2.1	Problem Formulation	58
9.2.2	Procedure	63
9.2.3	Burn Schedule Definition	64
9.3	Verification	64
9.3.1	Burn Schedule Approach	65
9.3.2	Minimum Thrust Constraint	69
9.3.3	Computation Time	72
9.3.4	Atmospheric Model Selection	73
10	Results	75
10.1	Launch Vehicle Optimization	75
10.2	Engine Sizing	77
10.3	Preliminary Ascent & Descent	77
10.3.1	Preliminary Ascent	77
10.3.2	Preliminary Descent	79
10.4	Aerodynamics	80
10.5	Simulations	82
10.5.1	DoE 1	82
10.5.2	DoE 2	91
10.5.3	DoE 3	94
10.5.4	DoE 4	96

---

11	Conclusion	101
12	Recommendations	105
12.1	Research	105
12.2	Model Improvements	105
12.3	Software	106
A	Appendix	107
A.1	Vehicle Optimization	107
A.2	NASA CEA Run Database	108
A.2.1	M10 Engine	108
A.3	MX Engine	109
B	Pseudocode	111
B.1	Control allocation	112
B.1.1	$P_a$ thrust allocation scheme	112
B.1.2	Engine ignition thrust allocation scheme	113
B.2	DESCO	114
B.3	Simulator	115
B.3.1	SDK Block	115
B.3.2	SN Block	115
B.4	Algorithms	116
B.4.1	SOCP1t0	116
B.4.2	SOCP1	117
B.4.3	SOCP2t0	118
B.4.4	SOCP2	119
	Bibliography	121



# List of Figures

3.1	MDO discipline structure and order. . . . .	9
4.1	Two inertial reference frames, where $X'Y'Z'$ is moving with a constant velocity $W$ with respect to $XYZ$ . Taken from Wakker [66] . . . . .	14
4.2	From left to right: regions of zonal, tesseral, and both harmonics, taken from Wakker [66] . . . . .	16
5.1	GLOM breakdown of the RLV adapted from [50] . . . . .	21
5.2	M10 engine characteristics given by [29] . . . . .	22
5.3	MX (orange) and M10 (blue) engine configuration on vehicle body axis for first stage. . . . .	23
5.4	RLV geometry, variables, and material properties over simplified schematic for the 2*MX+M10 first stage and M10 second stage. Fuel tank, oxidizer tank, turbine, pump, and gas generator block diagram with valves shown on the first stage has been adapted from [71] . . . . .	28
6.1	Missile DATCOM procedure and code snippet for passing inputs through fprintf. . . . .	38
8.1	Simulator block diagram representation . . . . .	48
8.2	Time lag effects between guidance algorithm and simulator. . . . .	53
9.1	Verification trajectories. The points on the trajectory are selected so as to indicate a 6 (for $t_f = 300$ sec) or 5.6 (for $t_f = 280$ sec) second interval between the last point. . . . .	66
9.2	Thrust profile of pre-generated burn schedule approach for $t_f=300$ sec, $\%t_1=.1$ , and $\%t_2=.6$ . . . . .	67
9.3	Thrust profile of pre-generated burn schedule approach for $t_f=280$ sec, $\%t_1=.0$ , and $\%t_2=0$ . . . . .	67
9.4	Thrust profile of big-M approach burn scheduling for $t_f=280$ sec, $\%t_1=.0$ , and $\%t_2=0$ . . . . .	68
9.5	Verification trajectories with the added $T_{min}$ constraint on DESCO. The points on the trajectory are selected so as to indicate a 6 (for $t_f = 300$ sec) or 5.6 (for $t_f = 280$ sec) second interval between the last point. . . . .	70
9.6	Thrust profile of big-M approach burn scheduling for $t_f=280$ sec, $\%t_1=.0$ , and $\%t_2=0$ , with added $T_{min}$ constraint on DESCO. . . . .	71
9.7	Thrust profile of pre-generated burn schedule approach for $t_f=280$ sec, $\%t_1=.0$ , and $\%t_2=0$ , with added $T_{min}$ constraint on DESCO. . . . .	71
9.8	Thrust profile of pre-generated burn schedule approach for $t_f=300$ sec, $\%t_1=.1$ , and $\%t_2=.6$ , with added $T_{min}$ constraint on DESCO. . . . .	72
9.9	State history comparison for pre-generated burn schedule approach for $t_f=280$ sec, $\%t_1=0$ , and $\%t_2=0$ , using the LF 62 and HF 76 atmospheric models for guidance and simulator respectively. . . . .	73
10.1	$C_D$ and $C_L$ data for 802 RLV Id (best case $m_{burn,descent}$ . . . . .	80
10.2	$C_D$ and $C_L$ linear fit with minimum and maximum variation relative to CALLISTO . . . . .	81
10.3	DoE1 metric results. . . . .	82
10.4	Feasible to infeasible metric comparison for DoE 1 results. . . . .	83
10.5	Feasible cluster plots with line of best fit as scatter data for DoE1. . . . .	84
10.6	Density plot for variations of feasibility with respect to total evaluations for difference between $t_{gui}$ and $t_1 = t_{sim}$ . . . . .	86
10.7	Density plot for variations of feasibility with respect to total evaluations for difference between $t_{gui}$ and $t_2 = t_{sim}$ . . . . .	87
10.8	Distribution of optimal, feasible, and infeasible values of $\%t_1$ for DoE1. . . . .	87

10.9	Distribution of optimal, feasible, and infeasible values of % $t_2$ for DoE1. . . . .	88
10.10	Distribution of optimal, feasible, and infeasible values of % $t_f$ for DoE1. . . . .	88
10.11a	nice plot . . . . .	89
10.12a	nice plot . . . . .	89
10.13a	nice plot . . . . .	90
10.14	DoE3 $\frac{f_{sim}}{f_{gui}}$ variation results for $\ r_f\ $ . . . . .	94
10.15	DoE3 $\frac{f_{sim}}{f_{gui}}$ variation results for $\ v_f\ $ . . . . .	95
10.16	DoE 4 sensitivity analysis for $v_{N,0}$ . . . . .	97
10.17	DoE 4 sensitivity analysis for $v_{E,0}$ . . . . .	97
10.18	DoE 4 sensitivity analysis for $v_{D,0}$ . . . . .	98
10.19	DoE 4 sensitivity analysis for $m_0$ . . . . .	98
10.20	DoE 4 sensitivity analysis for $h_0$ . . . . .	99
10.21	DoE 4 sensitivity analysis for $X_0$ . . . . .	99
10.22	DoE 4 sensitivity analysis for $t_{pt}$ . . . . .	100
10.23	DoE 4 sensitivity analysis for $\sigma_{URE}$ . . . . .	100
B.1	Thrust control allocation scheme pseudo-code for optimization of minimized mass consumption based on maximum achievable thrust with pressure losses. . . . .	112
B.2	Thrust control allocation scheme pseudo-code with prioritization for engines that are already switched on. . . . .	113
B.3	DESCO pseudocode. . . . .	114
B.4	SDK block pseudocode. . . . .	115
B.5	SN block pseudocode. . . . .	115

# List of Tables

1.1	Stakeholder requirements for mission. . . . .	2
3.1	Design variables and required output for each discipline. . . . .	10
3.2	Optimizer and software used for each discipline. . . . .	11
3.3	Metric tolerances for 3-DoF simulations . . . . .	11
4.1	Standard atmosphere 1962 model constants. . . . .	19
4.2	Atmospheric model variations used for 3-DoF simulations and guidance algorithm. . . . .	20
5.1	M10 Engine characteristics grouped from stakeholder information and literature. . . . .	23
5.2	MX Engine characteristics grouped from stakeholder information and literature. . . . .	23
5.3	Relationships for a hydrolox propellant liquid engine from [69] . . . . .	24
6.1	Total variations for NASA CEA run. 3000 different points evaluated in total for varying $P_c$ and $\frac{P_c}{P_e}$	30
6.2	Flight conditions for Missile DATCOM runs. . . . .	37
8.1	DoE 4 sensitivity analyses for initial state variations. 3600 total cases. Each case used seed values of 10, 11, and 12 with the mersenne twister algorithm. . . . .	52
8.2	User equivalent range errors used for DoE 4 and regions of $h$ for activation. . . . .	54
9.1	Burn schedule characterization for SOCP problems based on number of times controllers require formulation . . . . .	64
9.2	Nominal parameters for SDK (spacecraft dynamics & kinematics) and DG (descent guidance) block. . . . .	65
9.3	Verification trajectories for different types of $\mathcal{T}_p$ and methods of generation. . . . .	65
9.4	Verification trajectories for different types of $\mathcal{T}_p$ and methods of generation, with the minimum thrust constraint $T_{min}$ as 10% of the maximum thrust of each engine. . . . .	69
9.5	Controller pre-load and build times for different burn scheduling approaches. . . . .	73
10.1	Measures of central tendency for GA results for 100kg payload class RLV. . . . .	75
10.2	Measures of central tendency for GA results for 200kg payload class RLV. . . . .	76
10.3	Measures of central tendency for GA results for 300kg payload class RLV. . . . .	76
10.4	Measures of central tendency for GA results for 400kg payload class RLV. . . . .	76
10.5	Measures of central tendency for GA results for 500kg payload class RLV. . . . .	76
10.6	Measures of central tendency for MX feasible candidates. . . . .	77
10.7	Measures of central tendency for M10 feasible candidates. . . . .	77
10.8	Measures of central tendency for preliminary ascent and descent starting conditions and criteria, for payload class of 100kg. . . . .	78
10.9	Measures of central tendency for preliminary ascent and descent starting conditions and criteria, for payload class of 200kg. . . . .	78
10.10	Measures of central tendency for preliminary ascent and descent starting conditions and criteria, for payload class of 300kg. . . . .	78
10.11	Measures of central tendency for preliminary ascent and descent starting conditions and criteria, for payload class of 400kg. . . . .	78

10.12	Measures of central tendency for preliminary ascent and descent starting conditions and criteria, for payload class of 500kg. . . . .	79
10.13	Preliminary descent nominal selection results. . . . .	79
10.14	Ranges of DoE 1 clusters for $\ \mathbf{v}_f\ $ vs. $\ \mathbf{r}_f\ $ . The signs in the parentheses in the ranges column corresponds to F for feasible, O for optimal, and I for infeasible. . . . .	83
10.15	Linear fit statistics of DoE 1 feasible clusters for $\ \mathbf{v}_f\ $ vs. $\ \mathbf{r}_f\ $ . . . . .	84
10.16	Measures of central tendency of feasible $\ \mathbf{v}_f\ $ vs. $\ \mathbf{r}_f\ $ clusters for % $t_1$ . . . . .	85
10.17	Measures of central tendency of feasible $\ \mathbf{v}_f\ $ vs. $\ \mathbf{r}_f\ $ clusters for % $t_2$ . . . . .	85
10.18	Measures of central tendency of feasible $\ \mathbf{v}_f\ $ vs. $\ \mathbf{r}_f\ $ clusters for % $t_f$ . . . . .	85
10.19	DoE 2 and 3 nominal candidates selected from DoE 1 results. . . . .	90
10.20	DoE 2 Measures of statistical tendency for variation of $N$ 's $\ \mathbf{r}_f\ $ . Asterix (*) denotes nominal value for experiments without variation of this parameter unless otherwise stated. . . . .	91
10.21	DoE 2 Measures of statistical tendency for variation of $N$ 's $\ \mathbf{v}_f\ $ . Asterix (*) denotes nominal value for experiments without variation of this parameter unless otherwise stated. . . . .	91
10.22	DoE 2 Measures of statistical tendency for variation of $N_{cvx}$ 's $\ \mathbf{r}_f\ $ . Asterix (*) denotes nominal value for experiments without variation of this parameter unless otherwise stated. . . . .	92
10.23	DoE 2 Measures of statistical tendency for variation of $N_{cvx}$ 's $\ \mathbf{v}_f\ $ . Asterix (*) denotes nominal value for experiments without variation of this parameter unless otherwise stated. . . . .	92
10.24	DoE 2 Measures of statistical tendency for variation of $\mathcal{W}_{\eta_w}$ 's $\ \mathbf{r}_f\ $ . Asterix (*) denotes nominal value for experiments without variation of this parameter unless otherwise stated. . . . .	92
10.25	DoE 2 Measures of statistical tendency for variation of $\mathcal{W}_{\eta_w}$ 's $\ \mathbf{v}_f\ $ . Asterix (*) denotes nominal value for experiments without variation of this parameter unless otherwise stated. . . . .	92
10.26	DoE 2 Measures of statistical tendency for variation of $\theta_{max}$ 's $\ \mathbf{r}_f\ $ . Asterix (*) denotes nominal value for experiments without variation of this parameter unless otherwise stated. . . . .	92
10.27	DoE 2 Measures of statistical tendency for variation of $\theta_{max}$ 's $\ \mathbf{v}_f\ $ . Asterix (*) denotes nominal value for experiments without variation of this parameter unless otherwise stated. . . . .	93
10.28	DoE 2 Measures of statistical tendency for variation of Nominal Candidate's $\ \mathbf{r}_f\ $ . . . . .	93
10.29	DoE 2 Measures of statistical tendency for variation of Nominal Candidate's $\ \mathbf{v}_f\ $ . . . . .	93
10.30	DoE 1 nominal candidate used for DoE 4. . . . .	96
A.1	M10 Engine sizing suitable candidates. . . . .	109
A.2	MX Engine sizing suitable candidates. . . . .	110

# Chapter 1

## Introduction

Reusability has been considered as an alternative to expendable launch vehicles as early as 1967, as Knauer identified it as "The Next Major Launch Vehicle Development". As with any development in the realm of Aerospace Engineering, its purposes are based on financial and or strategic decisions. Financial decisions are related to cutting costs or increasing profit margins, while strategic decisions seek end goals other than making money [49]. Intuitively it is understandable that a reusable launch vehicle is financially attractive by offering more flight opportunities for the launch service provider which in turn cut costs for developments of new architecture [68]. Under the same study by Knauer it was noted that a proposed modified Saturn V/S-IC for recovery could have contributed to 60% in savings when compared to the standard ELV (expendable launch vehicle) version of the Saturn V.

It wasn't until the 1970's where the first reusable launch system started development under NASA's Space Shuttle program. From its first and last flight between 1981 and 2011 [31] respectively, several reusable launch vehicle programs were contracted of which most never saw the light of day [49], one of the most noticeable family being the x-30 to 40 experimental launchers [33][34][35][36]. A noticeable paradigm shift in the field of reusability occurred with the introduction of SpaceX Falcon 9's rocket [37][52]. Not only was it able to produce successful flights but for the first time since the Kistler K-1 [32] had been contracted, there was a focus on VTVL semi-reusable systems rather than a completely reusable single-stage-to orbit LV. Later on in 2016 Blue Origin established itself as another key player in the RLV market having completed 4 tests of the first stage of the New Glenn semi-reusable launch vehicle [44].

As mentioned, the RLV market is segmented. The segmentation of the market calls for attention to specific sub categories of research. Beneath the top block representing the field of RLV study as a whole, lie subcategories. The most active in the research community are the subcategories of guidance, trajectory optimization, and studies on configuration.

The purpose of this literature study is to primarily serve as a requirement for graduation of the Aerospace Engineering masters at TU Delft, as well as a reference document when the student begins her/his thesis work. The report is structured into the following sections. First, the stakeholder requirements given by AVIO are detailed in the next chapter. Then, the section after deals with explaining flight mechanics, including reference frames, translational and rotational equations of motion. Section 4 details the environment of the problem, namely the acceleration, and atmospheric models relevant to a re-entry problem and briefly mentions how to select the appropriate ones. Optimal control theory follows, which describes the components of solving an optimal control problem; non-linear programming, differential equations and integration of functions, and non-linear algebraic equation systems. Section 6 explains the guidance and control relevant to a reusable rocket landing problem, detailing various guidance laws, and control methodologies, and selection strategies. Launcher configuration then follows, detailing individual systems of the launch vehicle and then the collective.

### 1.1 Stakeholder Interests

Founded in 1908, but officially known as "AVIO s.p.a" for the past 50 years, it started as an aeronautical company. Going through various rebranding from "Società Italiana Aviazione" (Italian Aviation Society) to Fiat Avio, and to what its known as today, being a leader in the space propulsion market. The work for which AVIO has been most well known for the last decade has been its contribution to the development of the VEGA (Vet-

tero Europeo di Generazione Avanzata) rocket family. The expertise spans across liquid, solid, and cryogenic propulsion for either military or commercial payload launches. While the origin of the word is merely a call to the company's original transition from Fiat Aviazione, then shortened to Fiat Avio, employers familiarize the name as an acronym for "Advanced Vision Into Orbit". Whether or not the company aims to explicitly characterize itself as that, it is nonetheless a fair representation based on the work that it has accomplished. The current VEGA family consists all of expendable launch vehicles (ELVs). An effort as arisen in a collaboration with the propulsion, structures, and guidance navigation and control (GNC) department at AVIO to begin research on a semi-reusable derivative of a new VEGA launch vehicle. Prior to this study, some internal cost and preliminary analysis for the design of the launch vehicle and its intended reusable purposes were already established. The aim for AVIO is to research into a two stage launch vehicle with the purposes of conducting missions to send a payload into a sun-synchronous orbit at altitudes of 600km. Among these requirements, several others have been provided by AVIO for this study. The requirements ID in the first column are used as short hand notation, whose notation is in the form of 'requirement-system-number'. The system portion between the two hyphens are notations for launch vehicle (LV), propulsion (PR), and trajectory (TR), each of which will be discussed in the relevant subsections of section.

<b>ID</b>	<b>Stakeholder Requirement</b>
REQ-LV-1	The structural index of the launch vehicle should be within the range of 0.08 and 0.12
REQ-LV-2	Propellant mass should be within the range of 60000 to 80000 kg
REQ-LV-3	The RLV shall be flown with up to 500kg of payload
REQ-PR-1	Throttling rate is limited to 50% of maximum per second per stage
REQ-PR-2	The first stage shall have two MX and one M10 engines
REQ-PR-3	The second stage shall have one M10 engine
REQ-TR-1	The trajectory should follow a RTLS to the French Guiana space center
REQ-TR-2	The first stage shall return considering a 45-55% drop off in velocity prior to its separation
REQ-TR-3	The payload shall be ejected at 600km of altitude
REQ-GC-1	The guidance and control maximum frequency is 25 Hz

Table 1.1: Stakeholder requirements for mission.

# Chapter 2

## Research Goals

In this section we represent a summary of the literature study taken, the research gap, research objectives, research questions, and finally the planning of the research. We note that some terminologies are used in a different manner based on the authors they have been cited from. Specifically, we refer to trajectory optimization to offline planning/guidance, which we previously referred to as open-loop guidance. The same is true for closed-loop guidance, which we refer to here as online planning/guidance.

### 2.1 Literature Study Summary

It is clear from the research encountered, the importance of coupled flight and G&C optimization was not studied in the early days of RLV development [51]. At the time of Boeing's preliminary research prior to the Moon landing the focus since then and for many decades was on MDO analysis. It is not surprising when it has become a consensus that the capabilities of guidance are limited by the onboard computing power. Indeed, the first form of planetary guidance was the Apollo polynomial guidance, which was easy to implement and offered a solution in polynomial time. However, this was not an effective guidance scheme even at the time, where findings point to Meditch's fuel-optimal algorithm based on the Pontryagin Maximum principle in 1964 [41]. Moreover, both where analytical methods, which inherently reduce the problem of powered landing to only a few constraints. Specifically, most analytical methods do not consider process and attitude constraints, have derived commands related to fixed  $t_{go}$  (time to go), and consider decoupled three axes accelerations [57].

As the amount of onboard computation power increased throughout the years, researchers were able to derive new numerical programming algorithms that saw the rise of the field of computational guidance. In the mid 2000's 3-DoF (degrees of freedom) studies with a fuel optimality objective were solved with nonlinear programming problems. Specifically, the direct method (DM) of pseudospectral (PS) discretization showed high accuracy, despite having computing speeds unsuitable for online guidance. A 3-DoF landing problem on Mars was then revisited with other convex (CVX) methods of semi-definitive programming (SDP) and second-order cone programming (SOCP). Studies then showed that convexification of the landing problem resulted in the same solution as with the Pontryagin Maximum Principle. Unlike PS algorithms, convex optimization algorithms compute solutions much faster.

Development of new strategies for solving the NLP powered descent (PD) problem has not been limited only to the optimization method used. Different characteristics have been studied, such as hazard avoidance [67][73], VTVL and VTHL configurations [59][68], as well as combinations of the above mentioned methods. More recently, reinforcement learning approaches have been used to generate guidance commands. The analytical zero error miss/zero error velocity (ZEM/ZEV) subject to improvements via a deep learning approach has been studied [22][24][28]. Reinforcement learning approaches require training of the data, which must be done offline, and in the case of the mentioned literature, it was shown that solutions could not adapt to model uncertainties. For any NLP method, there is a trade-off between computation time and solution accuracy. For G&C solutions must be available online, however it is also a possible strategy to use more computationally expensive methods to generate an initial trajectory, as in the case of reinforcement learning and PS methods. Even when compared to reinforcement learning approaches, a 3-DoF landing problem returned better fuel optimal conditions and end time with a PS solver [57][25].

Aside from SpaceX and Blue Origin's works, there have also been experimental studies on flight and G&C for

RLVs. DLR launched the ENTRAIN (European Next Reusable Ariane) to determine the financial and operational feasibility of different return options. These looked at how the choice of MPS, structures, hardware, VTHL and VTVL LV format influenced the trajectory [68]. CNES (Centre National d'Études Spatiales) have FROG, whose objective was experimenting of a control algorithm and adaptive control system. This demonstrator shares an overall objective with others such as the EAGLE by DLR, DTV by ESA and INCAS in analyzing control laws for retro propulsion, being vehicles of a VTVL format. Outside of Europe, exists the technology demonstrators Peacock from CALT (China Academy of Launch Vehicle Technology), and RLV-T3 and RLV-T5 from Linkspace. Additionally JAXA has the RV-X launch vehicle and is also in a joint effort with CNES and DLR on the CALLISTO (Cooperative Action Leading to Launcher Innovation in Stage Toss-back Operations), whose main focus is GNC through convex optimization [16][20][57].

As was already mentioned, one of the key growing focus for RLV studies has been GNC and trajectory optimization. This is both due to the historical growth of computational guidance capabilities and the research in the mathematics that has amalgamated. Furthermore, approaches at solving the powered descent problem have largely targeted Mars [5][6][8][27][46][47][58][60][60][62][63][74][75], Moon [23][30][38][41][39][56][67][72], or asteroid landings [57]. Within the context of RLV, the ideal scenario would be to operate in atmospheric conditions of the Earth. The Moon's surface can be considered as a vacuum, and Mars's atmospheric effects can usually be ignored. In addition, Earth return has disturbances of wind shears. Moreover, landing zones for Mars and Moon are usually designated in a region of km of magnitude, while Earth landing requires pinpoint precision. With regards to engine configurations, planetary landers have low thrust and wide throttling capability, whilst Earth landers have limited throttling, high thrust and high fuel consumption. Lastly, RLV on Earth have strict attitude constraints, while planetary landers are more compact vehicles whose attitude control is less complex.

Assessment of the configuration of RLV is typically done with planning of both its ascent and descent trajectory in mind. The focus remains on keeping the first stage to be reusable. However, the configuration of the first stage is determined by what payload it needs to carry before jettisoning. As such, a systems analysis is carried out of different configurations based on choices of gross lift-off mass (GLOM), stage lift-off mass (SLOM), stage structural indices, and loads during re-entry [68]. Mass estimation of the first stage can be done through empirical methods, and stage structural indices can be based off flight tested launchers. Load constraints choices are typically, maximum dynamic pressure, maximum product of dynamic pressure and angle of attack, and maximum acceleration. Based on the ascent profile, the  $\Delta V$  at descent is determined, as well as conditions for beginning of return trajectory [57].

Previously it was mentioned that the choice of mission profile was almost synonymous with the choice of configuration. For example, the FB approach is characterized by a jettisoning of the first stage in a residual atmosphere. This can influence the motion between the RLV and the expendable stages, which raises the priority for flight control at this stage. Because the maneuver aims to invert the horizontal speed a large portion of the time the vehicle will be in a residual atmosphere environment, and as such it is advantageous to use an air-breathing propulsion system. To cover the large downrange distances, the vehicle is fitted with wings to increase its lift to drag ratio and allow gliding capability over a Mach range of 0.3 to 6. The TB profile features a steep ascent profile, and as such the configuration of the RLV is close to an ELV. This means the vehicle has a lower lift to drag ratio, and focuses on controlling its kinetic energy through ballistic phases, which demand a suitable control strategy. Because of the slender cylindrical geometry of the RLV, the reusable stage could have tendency to tilt around its transversal axis and reach high angles of attack. To change the angle of attack, control algorithms using aerodynamic control surfaces can be used. The dead-leaf, a mix of the FB and TB profile, intends to exploit the aerodynamics to the greatest extent such that very few or none (complete glide-back) MPS re-ignitions are needed. As a result, the aerodynamics of the RLV is of most importance, featuring large wings and extensive flight profile requirements, to optimize its lift to drag ratio. While RLVs are oriented vertically for take-off, DL and FB approaches may favour a horizontal landing method (VTHL) as supposed to a vertical one with TB (VTVL) [17].

The work done in ENTRAIN focused on the use of wings for any reusable first stage launcher approach methods, RTLS (return to launch site) or DRL (down range landing), and take off and landing method, VTHL (vertical take-off horizontal landing) or VTVL (vertical take-off vertical landing). What was varied was the choice of engine for the return options (choice of launcher approach method and vehicle format). As a general rule, Earth landers require throttling capability, thus engines and propellants are limited to staged combustion cycles and gas generator cycles. In addition, VTHL RLVs require an air-breathing engine. VTVL also requires landing gear/legs, whose geometry can be design by scaling the masses of the Falcon 9 with its dry mass [68], as can be done for the fins, or other available flight tested RLV landing leg design. Other aerodynamic

control/surface choices can be the use of grid fins or conventional fins, body flaps, wings, and wing flaps. The aerodynamics of the vehicles can be determined using slender body theory, with the wing aerodynamics using empirical lifting line methods. Lift and drag profiles at varying Mach numbers and angle of attack are then determined using semi-empirical methods [11].

Aside from having a vertical or horizontal landing approach, the return target may also influence the configuration of the launcher. For a slender cylindrical body RLV configuration RTLS proves to be an unreasonable option for GTO (geostationary transfer orbit) missions [68]. More propellant may be required due to the increase of the amount of burns: boostback, re-entry, and landing burn. In case of DRL, only the boostback does not need to be performed since the horizontal velocity of the RLV after jettisoning does not need to be inverted. For FB and DL the choice may not be immediately obvious. FB and DL both have velocity inversions but at later stages of descent.

The main focus on NOA for PD has been on CVX optimization. This is an attractive strategy because of its accuracy and capability for online planning. Studies with other NOA methods for Earth landing RLV have both simultaneously considered ascent and descent phases. There are relatively more niche categories more closely associated with the descent phase. These include load relief algorithms, retropropulsive thrust plume effects, and optimization of engine restarts. Load relief algorithms intend to increase robustness to wind influences and allow a decrease of mass by removing the need to over-engineer the structure to account for safety concerns due to wind conditions. It is more of a practice with descent trajectory as the effect of load relief is influenced by the use of fins, usually inactive during ascent. Implementation of load relief algorithms saw improvements in command tracking and wind rejection by reducing the impact of wind on pitch and drift [12]. It is also shown that load relief is of most importance during peaks of maximum dynamic pressure. Retro-propulsion causes the plume to be highly under-expanded, interacting with the outside surface a VTVL RLV. During re-entry, the heat loads are known to be dispersed so that the maximum heat load on the vehicle decreases at the baseplate but increases on the walls of the VTVL RLV [21]. As RLV require liquid engines, the number of engine restarts should be minimized and can be optimized as a metric to determine the degree of reusability [40].

## 2.2 Research Gap

It is expected from [57] that future developments on studies of PD of RLV will revolve around improving the initial guess of the non-convex Earth landing problem. Additionally, it was mentioned that guidance algorithms need to balance between accuracy and computational speed. This trade-off is part of three more general issues not limited to algorithms but also the physical constraints imposed by the chosen configuration of the RLV. The latter is related to the first general issue, which is physical feasibility. This relates to whether or not the RLV virtually has a possible solution. Solutions are distinguished by two perspectives; the reachable and controllable set. Of importance here is the controllable set, which are the entry conditions that can be controlled to reach the desired final condition. These conditions are the boundary and the path and control constraints. The second is the accuracy of the environmental and dynamics model. Model uncertainties are always present, and are due to poor measurement accuracy, noise on measurements, unaccounted dynamic characteristics, and inaccurate model parameters. An appropriate selection for the environment models would be inclusion of gravity, tabulated atmospheric values (density, pressure, and temperature), and engine back pressure losses.

Inferences on possible improvements were also made in the literature review. Out of all methods for optimization of the offline or online solution, only bang-bang control, CVX, and RL methods consider a discontinuous inequality constraint for the thrust. TB strategies employ the use of liquid engines with limited throttling capability. Solutions shown for the thrust profiles of both CVX and PS methods for the descent phase indicate a large throttling capacity. PS methods have the ability to solve for a problem with different thrust constraints but requires splitting the problem into multiple phases. The issue then arises of giving a good initial condition or suitable search range for linking the ascent and descent. Other problem formulations, such as NP-hard problems also require defining the bounds of the thrust profile. It would be useful therefore to have a problem formulation and method that can consider multiple bounds for the thrust constraint without having to split the problem in multiple phases.

With regards to research done on reusable vehicles at TU Delft, the following has been completed. In 2012, Vandamme completed a thesis on research of the impact of launch assists on the effect on the performance

of launch vehicle design. The main recommendation was to include a multi stage launch vehicle design and subject it to the same MDO as was done [65].

In 2013, Van Kesteren followed this recommendations and modeled airplane-assisted launching with a cost model. A major difference considered the usage of solid rocket propulsion instead of liquid propulsion. Two different payload classes, 10kg and 2000kg were considered, yielding different  $\Delta V$  to go into orbit, resulting in difference in cost per flight and GLOM. The recommendations included here were to use boosters or multiple rocket engines and include models of liquid and hybrid rocket motors [64].

In 2015, Miranda completed research on the effect of hybrid rocket propulsion for airplane assisted launching and ground landing. Using the MDO methods from Vandamme and Van Kesteren, the results showed that hybrid rocket propulsion decreased GLOM of the launch vehicle [42]. The recommendation was made that a realistic model of separation of the launch vehicle from the assisting vehicle be made and introduce boosters and/or multiple rocket engines.

More recently, in 2017 Contant completed research on a VTVL design microlaunch vehicle. In June of 2020, Rozemeijer held the torch for the latest research in RLV's. He conducted a price per flight investigation in the cost effectiveness of toroidal staged microlaunch vehicles by comparison to a nominal VTVL microlaunch vehicle design [50].

The general trend for RLV studies at TU Delft have been the following. An MDO consisting of separated propulsion, geometry, and mass discipline, as well as trajectory and cost optimization disciplines. Over the years, the main candidate for conducting simulations and optimization has been the TU Delft Astrodynamic Toolbox (Tudat) using C++ libraries, along with trajectory optimization package PAGMO. Some level of guidance, limited to a polynomial guidance law, was considered. An additional recommendation from Rozemeijer was that the ignition of engines required precision so that the RLV did not spin out of control. This can be prevented with the addition of control algorithms. In addition, amongst all these studies the equations of the motion for the system were limited to the kinematics, and not the full body dynamics. This means the trajectory optimization studies were limited to 3-DoF.

Outside of TU Delft, several research on the guidance and control aspect of the PD have been previously mentioned. Most recently, Song et. al. [57] 2020 recommendations, and that of Simplício et. al. [53] in 2018, mention that the industry short term trends will be to combine open loop guidance solutions with control compensators for attitude control, and usage of closed loop laws, aided by offline trajectory solution waypoints. With this in mind, the research proposal is the following. An MDO approach is to be applied to a new RLV configuration, starting at the vehicle design discipline. Trajectory optimization is included as the next discipline and will couple both the ascent and descent phase of the vehicle using a preliminary approach of analytical and semi-analytical routines. Its purpose is two-fold; to match the recommendations of Song et. al. and Simplício et. al. for obtaining a nominal selection of trajectory starting conditions, but also in order to validate the configuration of the vehicle, insofar that it can actually achieve a mission, hence provide evidence for its physical feasibility. Different payload mass classes are to be analyzed. A convex optimization guidance algorithm is applied to give a reference thrust profile for the RLV to match with uncertainties involved in the actuator control allocation. Furthermore, sensitivity analysis experiments are run on initial state variations as well as influences on computational time delays, and real-time position and velocity errors.

In order to make the goals of the research project feasible, in contrary to aforementioned researches into reusability, a cost analysis is not considered, and only a single nominal VTVL configuration is selected. The restriction to a VTVL configuration is done based on the design requirements that VTHL require as mentioned by [17]; namely a large focus on optimization of aerodynamic surfaces, which would require the study to include an aerodynamics discipline to optimize the wings. Due to the fact that the study aims to build several routines from the ground up, it's scope will not include this. What is novel with respect to [20] is that the approach for convex guidance will use 3 instead of one single engine and will test the robustness of the guidance algorithm in this respect. Moreover, the RLV is tested for feasibility of RTLS assuming it is possible to achieve a 600km orbit with two stages.

## 2.3 Research Objectives

Following the research proposal stated before, a research objective and following Sub-Objectives can be identified

**Research Objective:** *Determine the feasibility of Reusable operations for the VEGARLV launcher by system*

*analysis of configuration, couple ascent and descent trajectory optimization, and a descent G&C algorithm*

The sub-objectives are the following:

**Sub-Objective 1:** *Define a VTVL RLV configuration based on the VEGA launcher characteristics by a systems engineering approach*

**Sub-Objective 2:** *Design a trade-off for the ascent and descent trajectory for the chosen RLV configuration by analytical orbit insertion estimates and semi-analytical guidance law implementation*

**Sub-Objective 3:** *Improve the online planning capabilities of the optimized trajectory by implementing a convex optimization guidance algorithm*

The feasibility keyword in the main research objective is answered by the degree to which the results of the sub-objective have been satisfactory. One key goal for the trajectory design is the ability for the missile to nullify its touchdown speed, and land on the launch pad (6m wide launch pad as reference [20]). The RLV must also have a completely vertical profile when landing. This check can be performed by analysis of the results of the trajectory optimization and guidance process. These two requirements fall under what was introduced as *physical feasibility*; the RLV can completed ascent and descent operations without breaking any path or boundary constraints.

Discipline feasibility is another criteria, for vehicle design, trajectory optimization, guidance, and control. The first objective has a preliminary role in determining these results. Vehicle Sizing discipline feasibility entails constructing the launch vehicle so that it matches the stakeholders requirements listed in table 1.1. Trajectory optimization discipline feasibility can be determined with preliminary analysis for the ascent and descent profile. Guidance discipline feasibility will require a physically feasible solution using the selected guidance method. This will be coupled with the actuators selected whose discipline feasibility will require selection of a nominal set of engines. Ideally, after achieving all feasibility requirements, we will be able to achieved inter-discipline feasibility, where several disciplines connect with one another.

## 2.4 Research Question

The main research question is formulated below:

**Research Question:** *To what extent is the VEGARLV configuration feasible for RTLS operations based on system sizing, offline planning, and online planning capabilities?*

The research question can be broken down into sub-questions.

**Sub-Question 1:** *What payload class can satisfy the structural index limit and achieve physically feasible ascent and descent operations?*

**Sub-Question 2:** *What type of burn schedule does the VEGARLV require for descent?*

**Sub-Question 3:** *What is the influence of tuning parameter choices on the guidance algorithm and its frequency relative to the simulation?*

Each research question is formulated so as to provide give insight into the internal goal of each sub objective. Research question 1 can be answered by the results of the first objective and also those of the second by selection of nominal candidates from the preliminary analysis of the ascent and descent profile. The second sub-question can be answered with the second and third objective, which is expected to yield different results as the 3-DoF simulation is more accurate. The last sub-question deals with the tuning of the simulation and guidance parameters and will be answered with the last objective.



# Chapter 3

## Methodology

Previous research at TU Delft on RLV ascent and re-entry studies have considered a MDO strategy (Multidisciplinary Design optimization). This trend has not been adapted in this study for the novel techniques and algorithms written. The mentioned previous research has primarily built upon an already existing environment written in tudat. However, because of the specific stakeholder requirements considered, it is more convenient to write anew the optimization strategy including software tools. There are still various approaches that are similar and independent on the programming environment used with respect to the previous studies, and previous contributions will be mentioned when necessary.

The multidisciplinary optimization problem considered in this study, at the monolithic level, still has requirements of discipline and interdiscipline feasibility. To achieve discipline feasibility, the stakeholder requirements must be respected at each of the different disciplines as shown in fig. 3.1. Some of these requirements influence more than one discipline, being REQ-TR-1 between preliminary optimization and the 3-DoF simulations. This being because the preliminary optimization uses different equations of motion and assumptions to model the descent trajectory, but the requirement that the RTLS launch pad (LP) should be at the French Guiana Kourou space center is still there. In this chapter, the known design variables, constraints, optimizers, and result processing techniques are detailed.

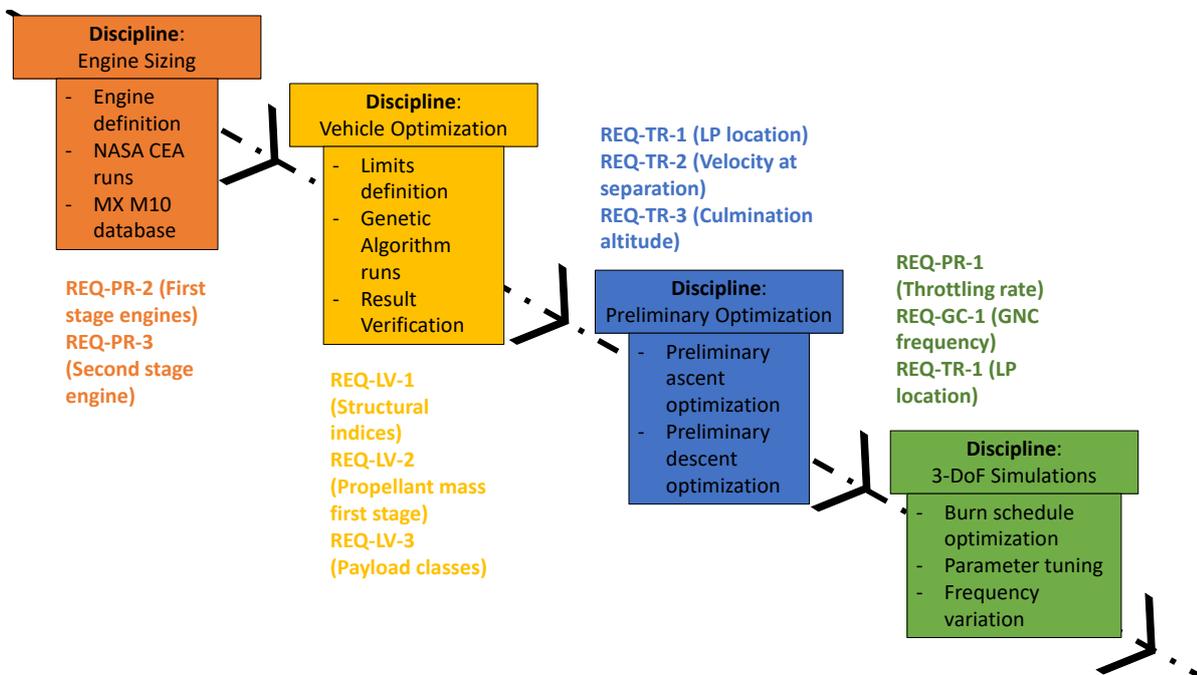


Figure 3.1: MDO discipline structure and order.

### 3.1 Design Variables

The design variables for each of the disciplines are inputs to their general optimization routine needed to solve the problem. These are shown in tab. 3.1.

<b>Discipline</b>	<b>Design Variables</b>	<b>Output</b>
Engine Sizing	$P_c$ $I_{sp}$ $\frac{P_e}{P_c}$	$D_e$ $P_e$
Launch Vehicle Optimization	$D_{stage,1}$ $m_{pay}$ $L_{c,ox}$ $k_{ellipse,1}$	RLV Geometry $C_D(\alpha, M)$ $C_L(\alpha, M)$
Preliminary Optimization	RLV Geometry $h_{culm}$ $m_{p,1}$ $m_{p,2}$ $m_{dry,1}$ $m_{dry,2}$	$\mathbf{x}_{0,descent}$
3-DoF Simulation(s)	RLV Geometry DESCO $\mathbf{x}_{0,descent}$ $C_D(\alpha, M)$ $C_L(\alpha, M)$	$\mathbf{x}_{f,descent}$

Table 3.1: Design variables and required output for each discipline.

For the engine sizing discipline, the inputs are given as ranges of chamber pressure  $P_c$ , specific impulse of the engines  $I_{sp}$ , and exit pressure to chamber pressure ratios  $\frac{P_e}{P_c}$ . The output is a database of feasible exit diameter  $D_e$  and pressures  $P_e$  of the engines. The  $I_{sp}$  are needed to interpolate with the nearest output  $I_{sp}$  of the the database formulated. The launch vehicle optimization discipline deals with the creation of a routine that builds the geometry and mass of the RLV. The first stage diameter  $D_{stage,1}$ , payload class mass  $m_{pay}$ , length of the oxidizer section  $L_{c,ox}$ , and the ellipse ratio of the common bulkhead tank of stage 1  $k_{ellipse,1}$  are the inputs to the problem. The output are the the RLV geometry, and as a end of optimization procedure, the coefficients of drag and lift  $C_D$  and  $C_L$  that are part of the aerodynamic analysis routine.

### 3.2 Optimizer

Each discipline will use a specific set of optimizers, which are ran with software specific for solving that problem. When referring to software, it also intended the MATLAB scripts and functions that are written, used interchangeably in this document with the word routine. Sometimes, the optimizer may be part of a specific toolkit shares the same software as other routines. An example of this is YALMIP, that is a SDP (semi-definitive programming) toolbox for MATLAB, which ECOS, the actual solver, is paired with. The 3-DoF simulation routine is also ran on MATLAB, and in fact the call to YALMIP is within it.

Discipline	Optimizer	Software
Engine Sizing	$I_{sp}$ interpolation with database	NASA CEA Run
Launch Vehicle Optimization	Genetic Algorithm	Launch Vehicle Optimization Routine
Preliminary Optimization	Factorial Experiment fmincon	Preliminary Optimization Routine
3-DoF Simulation(s)	Factorial Experiment fmincon (Actuator) ECOS (DESCO)	3-DoF Simulation Routine YALMIP

Table 3.2: Optimizer and software used for each discipline.

### 3.3 Constraints

The constraints to each discipline optimization problem are meant to narrow down the search space to a local optimal solution. Each discipline will have different motivations for their constraints whilst some are directly inherited by the stakeholder requirements.

### 3.4 Processing

When processing the results, due to the sequential structure of the optimization approach used for the study, it is important that nominal conditions carry over to the next discipline. Nominal conditions do not always equate to feasible and/or optimal candidates. In some experiments, it is possible that the optimal results are a result of the oversimplification of the problem, and as such when transferred across disciplines it may start to propagate errors due to inconsistencies between assumptions of different disciplines and problem set-up. The most intuitive example of this is between the preliminary optimization discipline, where the descent trajectory is calculated using a far less accurate model of dynamics, for the sake of reduced computation cost and to obtain a quantitative metric for a starting point of what would be the higher fidelity 3-DoF Simulation discipline.

For the 3-DoF simulations, the tolerances are shown in tab. 3.3. An assumption is made that the landing pad at Kourou has a radius of 6m. Moreover, it is considered that the limit for feasibility for the final velocity norm  $\|\mathbf{v}_f\|$  is considered as 6m/s. In reality, the final landing velocity is a function of the landing legs structure. However, not enough information is known about the RLV landing legs and as such it assumed that this can be considered acceptable for landing. This is considered also for the lateral velocity as a maximum velocity of 6 m/s will consider that any lateral movement will not drive the RLV away from the launch pad assuming that the final position norm  $\|\mathbf{r}_f\|$  is within feasible regions.

Metric	Optimal	Feasible	Less than feasible	Infeasible
$\mathbf{r}_f$ m	[0,1]	(1,6]	(6,10]	(10, $\infty$ ]
$\mathbf{v}_f$ m/s				
$q_{dyn,max}$ Pa	Determine from results			[100000, $\infty$ ]

Table 3.3: Metric tolerances for 3-DoF simulations



# Chapter 4

## Flight Mechanics

Flight dynamics refers to the aircraft/spacecraft's translational and rotational motion, described by a vector of position and velocities, as well as orientation in four dimensional space. This is based on Newton's Second Law  $F = ma$ , which allows calculation of the accelerations if the forces are known. In the sections below, the reference frames, equations of motion, and the models for the forces used are described.

### 4.1 Reference Frames

Before introducing the equations used to solve the dynamics of the problem, it is useful to know the concept of a reference frame and system. A reference frame is collection of points fixed relative to each other; in all reference frames at least one point exists which describes the origin. Depending on the type of dimensions, more points can exist that are fixed with respect to one another and the origin; these describe the axes in a reference system. In other words, a coordinate system uses the axes and origin of its reference frame to describe a set of point(s) [66]. The term coordinate system and reference system will be used interchangeably. For Engineering applications, it is common to have multiple reference frames and therefore reference systems pertaining to the problem. This happens when we look at space from a different observer. We may also want to reformulate the problem so that equations of motion (our differential equations) can be simplified. In the aerospace field, specifically in the problem of trajectory optimization, two reference frames are commonly used in problems relating to the Earth Moon Sun System. These are Earth Centered Inertial (ECI) and the Earth Centered Earth Fixed (ECEF) reference frames. One important caveat is that the way these reference frames are defined can differ in literature. For the purposes of this study, ECI is considered to be the reference frame with the origin at the center of a spherical Earth, the X axis pointing towards the Vernal Equinox, the Z axis towards the north pole at the time of the Vernal Equinox passing, and the Y axis completing the right hand rule. The ECEF frame's coordinate axis are rotating with respect to ECI. The origin is also at the center of the spherical Earth, and direction of rotation is assumed along the Z axis. In reality, the Earth's Z axis also rotates around an imaginary axis and wobbles about it. The first is due to a long-term effect known as precession, and the second a short term variation known as nutation. Precession results in a smooth long period motion of the ecliptic north pole (the real north pole at an instant in time) about the mean north pole, which takes 25800 years to complete. Nutation, whilst being a shorter term, still takes 18.6 years to complete a full cycle (the equator wobbles about the ecliptic plane, which is where the Earth's equator would be if neither effects existed). The time scale for this problem is in the magnitude of hundreds of seconds. [66] As both effects are dynamics that have significance over much larger time scales, they are excluded from this study. This explains why the only difference between our ECEF and ECI frames is a simple rotation (which takes 1 Earth day to complete) about the Z axis.

In simple mathematical formulation, we can put into symbols the difference between an inertial reference frame composed of axis  $XYZ$  and another inertial reference frame with a constant velocity this frame with axis  $X'Y'Z'$ . The constant velocity is with respect to the origins of the two frames. Figure 4.1 depicts the two reference frames. In mathematical formulation, the position of the two frames are related by equation 4.1.  $\mathbf{W}$  is the velocity of the second reference frame with respect to the first. Reference frames can also have different notions of time, which along with the relationship of position between  $XYZ$  and  $X'Y'Z'$  describe the Galilei

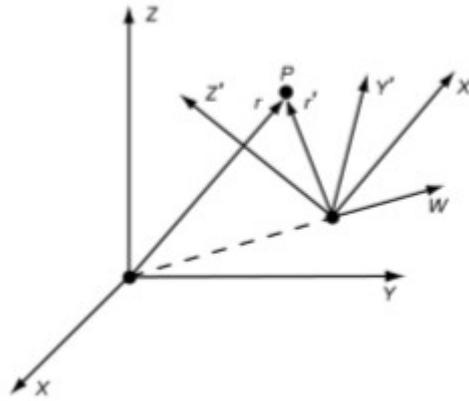


Figure 4.1: Two inertial reference frames, where  $X'Y'Z'$  is moving with a constant velocity  $W$  with respect to  $XYZ$ . Taken from Wakker [66]

transformations. Equation 4.2 shows the time difference where  $T$  is a time constant.

$$\mathbf{r}' = \mathbf{r} - \mathbf{W}(t - t_0) \quad (4.1)$$

$$t' = t + T \quad (4.2)$$

Transformations between different reference frames/systems are commonly performed with what is known as direction cosine matrix (DCM) also known as transformation matrices. Below the various reference frames used in this study will be discussed and DCMs are introduced where relevant. We denote transformation matrices with the letter  $C$ , with the subindex referring to the reference system you are converting to, and the superscript the reference frame you are converting to. We note that these matrices are square matrices and therefore the inverse of the matrix results in the reverse transformation.

#### ECI & ECEF

Equation 4.3 is the transformation matrix from ECI to ECEF, where  $\omega$  is the Earth rotation rate (72.9211585E-06 rad/s), and  $t$  is the time passed. This is represented by considering at time 0 both the ECI and ECEF frame being identical, and the ECI frame remaining fixed in space while the ECEF frame rotates around it.

$$\mathbf{C}_{ECEF}^{ECI} = \begin{bmatrix} \cos \omega t & \sin \omega t & 0 \\ -\sin \omega t & \cos \omega t & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.3)$$

The conversion to the ECEF frame can also be found by taking the distance from the center of the Earth in addition to the altitude  $R$ , the latitude  $\delta$ , and longitude  $\tau$ , as shown in equations 4.4 to 4.6.

$$r_{x,ECEF} = R \cos \delta \cos \tau \quad (4.4)$$

$$r_{y,ECEF} = R \cos \delta \sin \tau \quad (4.5)$$

$$r_{z,ECEF} = R \sin \delta \quad (4.6)$$

#### E-frame

The equations of motion are based on the E-frame. This set of equations considers the velocity in a local vertical and local horizontal relative to the Earth's center. The components are the north-pointing velocity  $v_N$ , the east pointing velocity  $v_E$  and the downwards pointing velocity  $v_D$ . To complement velocities, spherical coordinates are used for the position of the RLV. The environment considers a completely spherical Earth, which means that geodetic and geographic coordinates are the same. The longitude  $\delta$  of the RLV is measured with the Earth's vernal equinox pointing towards  $\delta = 0$  at  $t = 0$ . This considers no inclination of the orbital plane, and as such the latitude  $\phi$  is 0 at the equator.

### Angle of Attack Formulation

The study in the simulator considers a 3-DoF approach, and as such, rotational kinematics of the RLV are not considered. The angle of attack is defined as the between the rocket pointing vector and the velocity vector, in the ECI frame. This formulation is given in Siouris [55], and the angle of attack is given in eq. 4.7. The pointing vector in the ECI frame  $\mathbf{R}_{NT}$  is the unit vector of thrust, and the thrust magnitude vector  $\mathbf{R}_{NTm}$  is simply the thrust vector in ECI.

$$\alpha = \arccos \frac{\mathbf{R}_{NT} \cdot \mathbf{v}^{ECI}}{|\mathbf{R}_{NTm}| |\mathbf{v}^{ECI}|} \quad (4.7)$$

#### 4.1.1 Kinematic Equations of Motion

The equations of motion for the 3-DoF simulation are shown below from eq. 4.8 to 4.13. These have been taken from [43].

$$\dot{v}_N = \frac{F_x^E}{m} - 2\omega v_E \sin(\delta) - (\omega)^2 R \sin(\delta) \cos(\delta) - \frac{v_E^2 \tan(\delta) - v_N v_D}{R} \quad (4.8)$$

$$\dot{v}_E = \frac{F_y^E}{m} + 2\omega(v_D \cos(\delta) + v_N \sin(\delta)) + \frac{v_E}{R}(v_N \tan(\delta) + v_D) \quad (4.9)$$

$$\dot{v}_D = \frac{F_z^E}{m} - 2\omega v_E \cos(\delta) - \omega^2 R \cos^2(\delta) - \frac{v_E^2 + v_N^2}{R} \quad (4.10)$$

$$\dot{\delta} = \frac{v_N}{R} \quad (4.11)$$

$$\dot{\tau} = \frac{v_E}{R \cos(\delta)} \quad (4.12)$$

$$\dot{R} = -v_D \quad (4.13)$$

### LP Velocity and Position

Two different reference frames are used for the guidance algorithm and for the simulator. Whilst the simulator propagates the equations of motion as shown before in the E frame, the guidance algorithm considers position and velocity in LP frame. The equations for converting to this frame are given in 4.14 and 4.15 below, and have been taken from [20]. The subscript  $M$  denotes the missile location, whilst  $LP$  the launch pad location. The velocity in the LP frame can be converted directly with the DCM  $\mathbf{C}_{LP}^{NED}$  show in eq. 4.17 using the NED velocity  $v_{NED}$ . The position  $\mathbf{r}_{LP}$  requires calculating the ECEF to NED DCM as given in eq. 4.16.

$$\mathbf{r}_{LP} = \mathbf{C}_{LP}^{NED} \mathbf{C}_{NED}^{ECEF} (\mathbf{r}_{ECEF}(t) - \mathbf{r}_{ECEF}(0)) \quad (4.14)$$

$$\mathbf{v}_{LP} = \mathbf{C}_{LP}^{NED} \mathbf{v}_{NED} \quad (4.15)$$

$$\mathbf{C}_{NED}^{ECEF} = \begin{bmatrix} -\sin \delta \cos \tau & -\sin \delta \sin \tau & \cos \delta \\ -\sin \tau & \cos \tau & 0 \\ -\cos \delta \cos \tau & -\cos \delta \sin \tau & -\sin \delta \end{bmatrix} \quad (4.16)$$

$$\mathbf{C}_{NED}^{ECEF} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (4.17)$$

### Downrange and Crossrange distance

To be able to determine the relative ground track distances, the downrange and crossrange must be defined. These refer to the ground track distance relative to a 0 change in  $\tau$ , and 0 change in  $\delta$  for the downrange and crossrange, respectively. This helps give a more realistic representation of the final position norm metric  $\|\mathbf{r}_v\|$ , as using vector distances in the LP frame would assume cutting through the Earth to the final position from the LP point, thus being an underestimate. The eq. for downrange is given in 4.18 and crossrange in 4.19.

$$X_e = R_e \left( \sin \left( \frac{\pi}{2} - \delta_{LP} \right) (\tau_M - \tau_{LP}) \right) \quad (4.18)$$

$$Y_e = R_e \left( \sin \left( \frac{\pi}{2} - \tau_{LP} \right) (\delta_M - \delta_{LP}) \right) \quad (4.19)$$

## 4.2 Environment

The environment of the simulation considers the dynamics included, their constants, and the trajectory requirements. The sub-sections below will describe these dynamic models with a brief motivation for their inclusion.

### 4.2.1 Dynamics Models

When dealing with real world applications or trying to simulate real world applications in trajectory optimization, the forces being examined must be a function of the state parameters and some other constants. It must also be accurate enough so as to minimize the difference between the real acceleration value and the calculated value. In closed form solutions provided by Zarchan [71] to missile guidance, more commonly applied to 2D problems, the only acceleration considered is gravity as a constant  $9.81 \frac{m}{s^2}$ . This simplification allows an analytical form of the trajectory in x and y coordinates. Undoubtedly, closed-form solutions may provide results more efficiently and can be used as a preliminary experiment. However, the limitation of the dynamics to a constant gravitational acceleration is too restrictive. We will see in literature results of trajectory optimization that the effects of atmospheric drag at lower altitudes have a considerable effect on the trajectory [17]. Therefore, drag and lift will also be included.

The formulation of any acceleration on the RLV can be formulated by starting with Newton's second of law of motion. An important framework when considering this law is the reference frame in question. It is more practical and intuitive to solve and represent acceleration models and other parameters in terms of the ECI frame. In this frame, Newton's second law can be expressed as shown in 4.20, where  $v$  refers to the velocity,  $m$  the mass, and  $r$  the position.

$$\mathbf{F} = \frac{d}{dt}(m\mathbf{v}) = \frac{d}{dt}\left(m \frac{d\mathbf{r}}{dt}\right) \quad (4.20)$$

As shown in 4.1, a fixed reference frame (ECEF) can be reformulated into an inertial reference frame (ECI). Similarly, Newton's second law can be reformulated so that it is invariant in different inertial reference frames. Substituting Eq. 4.1 & 4.2 into 4.20 we arrive at Eq. 4.21.  $\mathbf{W}$  represents the velocity of the relative reference frame used in fig. 4.1. Of importance, is the second term of Eq. 4.21, which tells us that Newton's second law is invariant only when the change in mass of the system is zero.

$$\mathbf{F} = \frac{d}{dt}(m\mathbf{v}) - \mathbf{W} \frac{dm}{dt} \quad (4.21)$$



Figure 4.2: From left to right: regions of zonal, tesseral, and both harmonics, taken from Wakker [66]

### Thrust Force

Ideal rocket theory considers that a force is applied on the system by an expulsion of mass which decreases over time. It is important to note that the law of conservation of mass is still maintained (as all others), as in the mass of the missile plus the mass of the particles in the expulsion plume is constant. Applying Newton's second law of motion for this point mass system relative to an inertial reference frame we get Eq. 4.22, where  $\dot{m}$  is the mass flow leaving the rocket engine per unit time, and  $v_e$  is the effective exhaust velocity.

$$\mathbf{F}_T - \dot{m}\mathbf{v}_e = M \frac{d\mathbf{v}_{CoM}}{dt} \quad (4.22)$$

The form of thrust used by algorithms presented in this report is given by Eq. 4.23 [70].  $A_e$  is the exhaust area of the rocket nozzle, and  $P_e$  &  $P_a$  the pressure at the end of the rocket nozzle and the ambient pressure respectively. The second term in the equation does not seem intuitively to represent the final derived second law aside from the first term. The second term is the pressure drop that occurs when the nozzle is not able to achieve "optimum expansion". It can still be proven it is derived from the second law by considering the body of gas inside the combustion chamber and rocket nozzle. As shown by Zandbergen [70], drawing a boundary on the system consisting of the chamber and nozzle (resulting in a surface integral  $S_I$ ), conservation of momentum yields Eq. 4.24, where the first area integral can be simplified in mass rate times expulsion velocity.

$$\mathbf{F} = \dot{m}v_e + (P_e - P_a)A_e \quad (4.23)$$

$$\int_{S_I} (P_i - P_a)dS_i = \int_{A_e} \rho_e(U_e)^2 dA_e + \int_{A_e} (P_e - P_a)dA_e \quad (4.24)$$

The exhaust velocity  $v_e$  is given by eq. 4.25. Note that  $R$  refers to the ideal gas constant, whose value is the specific gas constant of the propellant and can be calculated by eq. 4.26 using the general gas constant  $R^*$  whose value is 8.314362618 J/molK [1], and the molar mass  $m_M$ .

$$v_e = \sqrt{2 \frac{\gamma}{\gamma - 1} R T_c \left(1 - \frac{P_e}{P_c}\right)^{\frac{\gamma - 1}{\gamma}}} \quad (4.25)$$

$$R = \frac{R^*}{m_M} \quad (4.26)$$

The known parameters of the rocket engines in this study are the sea-level gravity  $g_0$  and the specific impulse  $I_{sp}$ , taken as the specific impulse at vacuum  $I_{sp,vac}$ . Knowing the reference thrust  $T_{ref}$  to achieve, we can obtain the  $\dot{m}$  through eq. 4.27.

$$\dot{m} = \frac{T_{ref}}{I_{sp}g_0} \quad (4.27)$$

The ideal thrust equation, shown as 4.23 has some limitations when representing the behavior of overexpansion, that is when  $P_a < P_e$ . This is because if we assume that the exhaust velocity  $v_e$  is the one calculated at vacuum, and the only variables are  $\dot{m}$ ,  $P_a$  and  $P_e$ , the equation will yield a higher thrust value than achievable at optimum expansion when  $P_a = P_e$ . For this reason we assume two different formulations of eq. 4.23 by taking different assumptions of the MX and M10 engines. The first, more conservative approach, is considering that the engines have ideal altitude adapting nozzles, which experience optimal expansion for when  $0 \leq P_e \leq P_a$ . This is programmed as follows in eq. 4.28.

$$\mathbf{F}_T = \dot{m}v_e + \min[(P_a - P_e), 0]A_e \quad (4.28)$$

In the case that overexpansion is of interest, then a first-order approximation can be made by considering that  $\frac{P_e}{P_c}$  remains constant. The procedure to make this approximation of overexpansion loss is to first calculate  $\dot{m}$  with eq. 4.27. Then, the ideal rocket theory equation gives us another formulation for  $\dot{m}$ , as a function of  $\gamma$ ,  $P_e$ ,  $P_c$ ,  $R$ ,  $T_c$ , and  $A_e$ . Knowing  $\dot{m}$ , the exit to chamber pressure ratio can be found by manipulating eq. 4.29.

$$\dot{m} = A_e \frac{P_c}{\sqrt{RT_c}} \sqrt{\frac{2\gamma}{\gamma - 1} \left(\frac{P_e}{P_c}\right)^{\frac{2}{\gamma}} \left(1 - \frac{P_e}{P_c}\right)^{\frac{\gamma - 1}{\gamma}}} \quad (4.29)$$

The new  $P_e$  can be determined by multiplying  $P_c$  by the constant  $\frac{P_e}{P_c}$ , as shown in equations 4.30 and 4.31.

$$P_c = \dot{m} \sqrt{RT_c} \frac{A_e}{\sqrt{\frac{2\gamma}{\gamma - 1} \left(\frac{P_e}{P_c}\right)^{\frac{2}{\gamma}} \left(1 - \frac{P_e}{P_c}\right)^{\frac{\gamma - 1}{\gamma}}}} \quad (4.30)$$

$$P_{e,new} = P_c \frac{P_e}{P_c} \quad (4.31)$$

Finally, the new effective thrust can be calculated by re-evaluating  $v_e$  with eq. 4.25, and then plugging in  $P_{e,new}$  as  $P_e$  in eq. 4.23.

### Aerodynamic Forces

The aerodynamic forces on launch vehicle are the drag  $F_D$ , lift  $F_L$ , and side force  $F_S$ . For this study the model is simplified to exclude any side force as the sideslip angle  $\beta$  is treated as part of the effective angle of attack  $\alpha_{eff}$ . Eq. 4.7 is used to calculate this effective angle of attack ( $\alpha = \alpha_{eff}$ ). For the calculation of these forces, the atmospheric density  $\rho$ , coefficients of drag  $C_D$  and  $C_L$ , and cross-sectional reference area of the RLV  $S_{ref}$  are used. To obtain  $C_D$  and  $C_L$  the vehicle optimization routine's results are used as inputs for the geometry of the first stage. The coefficients are then formulated for varying  $\alpha_{eff}$ , and mach number  $M$ . For the purposes of this study,  $C_D$  and  $C_L$  are not functions of altitude  $h$  as the prior parameters are more influential and higher fidelity is not needed.

$F_D$  is calculated in the ECI frame by considering the relative velocity vector in the ECI frame  $V_{rel}$ , which is the velocity that the RLV experiences relative to the wind. The formula for  $V_{rel}$  is given in eq. 4.34. For  $F_L$ , due to the absence of vehicle rotational states, the velocity vector orthogonal to this is evaluated. This vector,  $v_{rel,\perp}$ , is given in eq. 4.33. To calculate the perpendicular vector relative to the  $v_{rel}$  and the direction of flight, the ECI velocity  $v^{ECI}$  and position  $r^{ECI}$  is used. The formulation is based on Rodrigue's rotation formula, used to rotate a 3D vector given an angle and a reference axis [15]. The formula requires calculation of the plane that skews the original vector to the new rotational direction. This plane  $\Pi_{v_{rel,\perp}}$  is simply the plane formed by  $r^{ECI}$  and  $v^{ECI}$ , as shown in eq. 4.32.

$$\Pi_{v_{rel,\perp}} = \frac{v_{rel} r^{ECI}}{\|v_{rel}\| \|r^{ECI}\|} \quad (4.32)$$

$$v_{rel,\perp} = v_{rel} \cos\left(\frac{\pi}{2}\right) + \Pi \sin\left(\frac{\pi}{2}\right) + (\Pi \cdot v_{rel})(1 - \cos\left(\frac{\pi}{2}\right)) \quad (4.33)$$

$$v_{rel} = v_{rel}^{ECI} - \Omega \times r^{ECI} \quad (4.34)$$

$$F_D^{ECI} = -\frac{1}{2} \rho S_{ref} C_D \|v_{rel}\|^2 \quad (4.35)$$

$$F_L^{ECI} = \frac{1}{2} \rho S_{ref} C_L \|v_{rel,\perp}\|^2 v_{rel,\perp} \quad (4.36)$$

### Gravity

The gravitational acceleration on the RLV is calculated in the E-frame. No perturbations effects like spherical harmonics are considered for this study, as the altitudes and duration of flight are presumed to not influence the dynamics significantly. The formula for gravitational force  $F_g$  is given in eq. 4.37. The gravity is dependent on the earth gravitational constant  $\mu_{Earth}$ , whose value is  $3.986 * 10^{14} \frac{m^3}{s^2}$ , the earth pointing vector which is the unit vector of the position vector in ECI, and the distance from the spacecraft to the center  $R$ , which is a state variable in the equations of motion 4.13.

$$F_g^{ECI} = -m \frac{\mu_{Earth}}{R^2} \frac{r^{ECI}}{\|r^{ECI}\|} \quad (4.37)$$

### 4.2.2 Atmospheric Models

Acceleration models may also be a function of parameters, which can be functions of state variables themselves. They can be represented by a tabulated form or an analytical form. Just like acceleration models, a trade off between realism and complexity of the model is imminent. Our data parameters, whose values are related as a function to the state parameters, are density  $\rho$ , and pressure  $P$ . These influence the drag and thrust respectively. The drag  $F_D$  is directly proportional to  $\rho$ , as seen in Eq. 4.35. The thrust is influenced by the pressure based on the exit nozzle pressure  $P_e$  and the ambient pressure  $P_a$ , which is the one that is calculated by a model.

### Exponential

The simplest and commonly used model for atmospheric effects is the exponential model. This is given by equation 4.38 [7].  $\rho_0$  is the density at sea level, which we define as  $1.225 \frac{kg}{m^3}$ .  $h$  is the altitude, and  $H$  is the scale height constant, which is 7200 m.

$$\rho = \rho_0 e^{-h/H} \quad (4.38)$$

The altitude can be found from ECI coordinates with Eq. 4.39, where  $R_e$  is the radius of the Earth (6378136 m).

$$h = \|\mathbf{r}\| - R_e \quad (4.39)$$

Similarly, the pressure can also be represented with an exponential model and with CIRA data (see below). The exponential model for the pressure is shown in equation 4.40.

$$P = P_0 e^{-h/H} \quad (4.40)$$

In principle, the only constants needed to calculate the relevant atmospheric data are the values at sea level and the scale height (H).

### CIRA

For a higher level of accuracy, there are the tabulated COSPAR international Reference Atmosphere (CIRA) models. Three types of models have been tabulated: CIRA low, CIRA mean, CIRA high. These are based on high ( $T_\infty = 696K$ ), low ( $T_\infty = 1310K$ ), and mean ( $T_\infty = 1020K$ ) variations of solar and magnetic activity as was done in [19]. The Committee on Space Research (COSPAR) developed this atmospheric reference models of temperature and densities from 0 to 2000km. The model accurately represents the features of different levels of the atmosphere: tropopause, stratopause, and mesopause. One reason such tabulated models may be more accurate is that they can represent discontinuities or irregular slopes of the function of the data parameter vs. the independent variable (height), which analytical equations may not. The low and high models are the values at seasons where either density or temperature are at an overall minimum or maximum [18].

### Standard Atmosphere 1962

The Standard Atmosphere 1962 model has been tabulated from an already existing database used in a optimization package called GPOPS-II, in one of its example problems for rocket re-entry. The data has been verified with the original report [? ]. The original data included ratios of  $\frac{P}{P_0}$  rather than the actual ambient pressure at sea level  $P_0$ . This is also true for the densities. As such, the constants used for this model are shown in table 4.1.

Constant	Definition	Value
$R \left( \frac{R^2 f t^2}{sec^2} \right)$	Air gas constant	1.71655e3
$\gamma$	Specific heat ratio	1.4
$P_0$ (psi)	Sea level pressure	14.695972
$\rho_0 \left( \frac{slug}{ft^3} \right)$	Sea level density	2.3768846e-3
$M_0 \left( \frac{g}{mol} \right)$	(Mean) Molecular weight at sea level	28.9644

Table 4.1: Standard atmosphere 1962 model constants.

### COESA 1976

The COESA (Committee on Extension to the Standard Atmosphere) 1976 model is included in MATLAB as a function. Below and above values of 0m and 84852m the function extrapolates values. The model includes values of atmospheric temperatures, speeds of sounds, pressures, and densities.

### NRLMSISE-00

The NRLMSISE-00 is a semi-empirical atmospheric model that contains definitions for atmospheric variables from the ground to the exobase. At the thermosphere, the model uses *Bates-Walker* equations that given temperature and density of species as an analytical function of altitude. It uses modifications of the equations of Hedin for solar flux calculations, oxygen profile, simulation of chemistry and dynamic flow effect on species, and thermal diffusion factors. For the mesosphere, the connection between the lower thermosphere and altitudes below 62.5 km are smoothed [45].

Configuration	Model		
	$P_a$	$A$	$\rho$
Low Fidelity 62	Standard Atmosphere 1962	Standard Atmosphere 1962	Standard Atmosphere 1962
Low Fidelity 76	COESA 1976	COESA 1976	COESA 1976
High Fidelity 62	$n_M$ with S.A. 1962, reference $T$ with NRLMSISE00	$\gamma, R$ with S.A. 1962, reference $T$ with NRLMSISE00	NRLMSISE00
High Fidelity 76	$n_M$ with COESA, reference $T$ with NRLMSISE00	$\gamma, R$ with S.A. 1962, reference $T$ with NRLMSISE00	

Table 4.2: Atmospheric model variations used for 3-DoF simulations and guidance algorithm.

### 4.2.3 Atmospheric Model Selection Motivation

Certain models are considered to have qualitatively a better representation of the real world dynamics. There are already some insights that can be made about the selection of aerodynamic models. A fidelity scale is created to rank the different models examined for this study. From highest to lowest fidelity, is NRLMSISE-00, COESA 1976, and Standard Atmosphere 1962. The issue with these atmospheric models is that they don't all contain the necessary information to completely model the atmospheric parameters, which are  $A$  for  $M$  calculation,  $P_a$  for engine back-pressure losses, and  $\rho$  for aerodynamic drag and lift. COESA 1976 and SA (Standard Atmosphere) 1962 models do contain this information. However, for NRLMSISE-00, to complete the environment models, it must be coupled with either COESA's or SA's data. This leads to two different additional models, for which will be named High Fidelity 62 for coupling NRLMSISE-00's data with SA and High Fidelity 76 for coupling with COESA. The HF (High Fidelity) 62 model is compiled by first obtaining the pressures and temperatures at the given altitude(s). NRLMSISE-00 provides  $T$  and  $\rho$ . Therefore, in order to obtain  $P_a$  using the model, the ideal gas law is used by considering a constant factor of number of moles times the gas constant  $n_M R$ . From the ideal gas law as given in eq. 4.41, we reformulate the equation to obtain the  $n_M$  as shown in eq. 4.42, for the SA  $T$  and  $P$  data, and eq. 4.44, for the COESA  $T$  and  $P$  data.

$$PV = n_M RT \quad (4.41)$$

$$n_{M,HF62} = \frac{P_{SA}}{RT_{SA}} \quad (4.42)$$

$$P_{a,HF62} = n_{M,HF62} RT_{NRLMSISE-00} \quad (4.43)$$

$$n_{M,HF76} = \frac{P_{COESA}}{RT_{COESA}} \quad (4.44)$$

$$P_{a,HF76} = n_{M,HF76} RT_{NRLMSISE-00} \quad (4.45)$$

To calculate  $A$ ,  $T$  on the reference model (SA or COESA) is used and plugged into eq. 4.46.

$$A = \sqrt{\gamma TR} \quad (4.46)$$

As for the other two models, CIRA and the exponential ones, their choice for not being included is the following. The exponential model is dependent on a scale height which can be modified. However, it does not contain information to form a fully dependent model for  $P$ ,  $\rho$ , and  $A$ , needed for the dynamics in the problem. This introduces the need for more dependencies from other model data which further increases the model pool. For the purposes of this study, the physical fidelity of the model is not so much of concern as is the discrepancy between a lower and higher (assumed) fidelity atmospheric model to mimic the uncertainties in the dynamics present in the real world GNC loop. For the CIRA model, the same is also true based on its dependency of atmospheric temperature regions.

# Chapter 5

## Launcher Configuration

The configuration of the launcher is in the early stages of development and will be concluded during the first steps of the thesis. AVIO is currently considering a two stage vehicle, whose configuration is based off a current work in progress M10 stage/engine, and a hypothetical MX, which has a thrust level six times more than the M10. Regardless of the chosen configuration, the selection of the propulsion system, and its own and the rest of the vehicles mass and sizing will be design variables in the equations of motions. These equations, particularly the thrust equation, consider the same assumptions of ideal rocket theory. Real rocket motors undergo other phenomenon, such as heat transfer from or to the expanding gas flow in the motor, non-ideal gas behavior, heterogeneous fluid flow, non-axial flow through the nozzle, flow separation, friction, shock waves, and chemical reactions to name a few. In an effort to bridge the gap between ideal rocket theory and real rocket performance, correction factors may be used.

In the later sections, the first and second stage geometry of the vehicles are detailed. These include sizing of the stage, interstages, and payload. They also include mass formulas used for the propellants, pressure system, structure, engine, payload fairing, payload internal bay, and interstage. When design parameters are not directly available from AVIO, they can be solved using relationships for geometry or equivalent theory. Regression analysis has also been completed from previous literature for some components of an ELV which will be assumed to work for an RLV here.

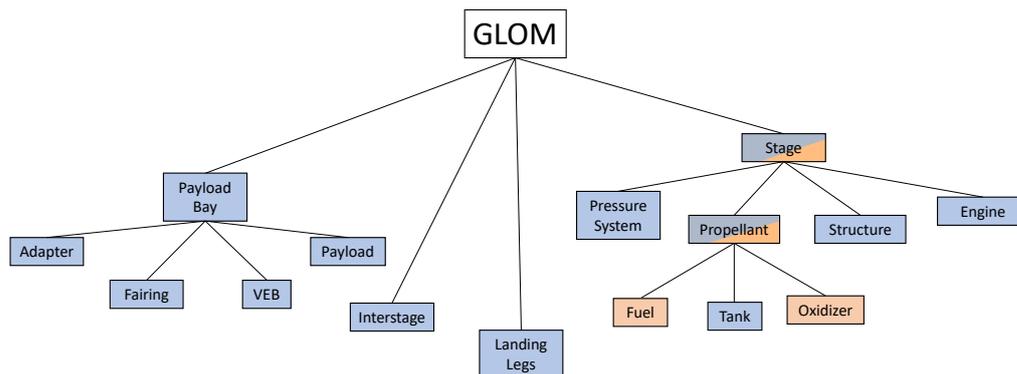


Figure 5.1: GLOM breakdown of the RLV adapted from [50]

Figure 5.1 shows the breakdown of the mass components that make up the gross lift off mass (GLOM) of the system. The blocks shaded in blue indicate that the system makes up part of the total inert mass  $m_{inert}$ ,

whilst the blocks shaded in light orange indicate that the component makes up part of the total wet mass, referred to as propellant mass  $m_{prop}$  of the system. We use the subindex  $i$  to denote the different stages of the component. Components of the first and second stage will have identical equations but different variables in some cases and result in different masses and geometries in the end. The payload is not referred to as a stage of the vehicle and therefore will have its own sizing equations and methods. This is the same for the interstage connecting stages 1 and 2.

## 5.1 Propulsion

The stakeholder requirements for the propulsion system are relisted below. We begin with the introduction of the M10 engine listed in the latter 2 requirements.

- REQ-PR-1 Throttling rate is limited to 50% of maximum per second per stage
- REQ-PR-2 The first stage shall have two MX and one M10 engines
- REQ-PR-3 The second stage shall have one M10 engine

The ideal rocket thrust equation is repeated down below again for convenience, which represents the total mass thrust based on ideal rocket theory. To model the thrust in the dynamics using this equation, information on  $\dot{m}$  determinable by the  $I_{sp}$ , the ambient pressure described by the atmospheric model in section 4.2.2, and the exit pressure  $P_e$  at the nozzle exit must be known. The latter parameters are properties of the engines used for this study, the M10 and MX.

$$F_T = (\dot{m}v_e + \frac{\pi D_e^2}{4} P_e) - \frac{\pi D_e^2}{4} P_a$$

The M10 engine is the first European methalox engine and stage that will be used for the new class of VEGA E and VEGA E light vehicles. The engine is a cryogenic engine running on methalox; liquid methane CH<sub>4</sub>(L) and liquid oxygen O<sub>2</sub>(L) propellant. Fig. 5.2 from [29] shown below details the characteristics of the stage/engine.



Figure 5.2: M10 engine characteristics given by [29]

A more recent research focus at AVIO has been the MX engine, which is meant to perform similarly to the M10 but with 6 times more the maximum thrust capability. The currently known details of the MX engine based on information provided by AVIO are listed in table 5.2. For the rotational equations of motion, it must be known what the effective moment about the RLV's longitudinal body axis  $i_B$  is; this points out from the bottom at the engines to the top of the nose cone of the payload. Fig. 5.3 shows a preliminary configuration of the two MX engines and the single M10 engine on the first stage of the launcher. The axis labeled  $j_B$  and  $k_B$  are the RLV body axis, with the  $i_B$  axis pointing into the page. For the second stage, no offset between the  $i_B$  axis and the engines exist, as the single M10 engine is centered.

Parameter	Value	Source
$I_{sp}$ (sec)	362	Stakeholder
$F_T$ (kN)	100	Stakeholder
$D_e$ (m)	TBD	Stakeholder
O/F ratio	3.4	[3]
$\rho_{ox}$ (kg/m <sup>3</sup> )	1.14	[2]
$\rho_f$ (kg/m <sup>3</sup> )	.423-.72	Lower bound:[2] Upper bound:[10]

Table 5.1: M10 Engine characteristics grouped from stakeholder information and literature.

Parameter	Value	Source
$I_{sp}$ (sec)	317	Stakeholder
$F_T$ (kN)	600	Stakeholder
$D_e$ (m)	TBD	Stakeholder
O/F ratio	TBD	Stakeholder
$\rho_{ox}$ (kg/m <sup>3</sup> )	1.14	[2]
$\rho_f$ (kg/m <sup>3</sup> )	.423-.72	Lower bound: [2] Upper bound: [10]

Table 5.2: MX Engine characteristics grouped from stakeholder information and literature.

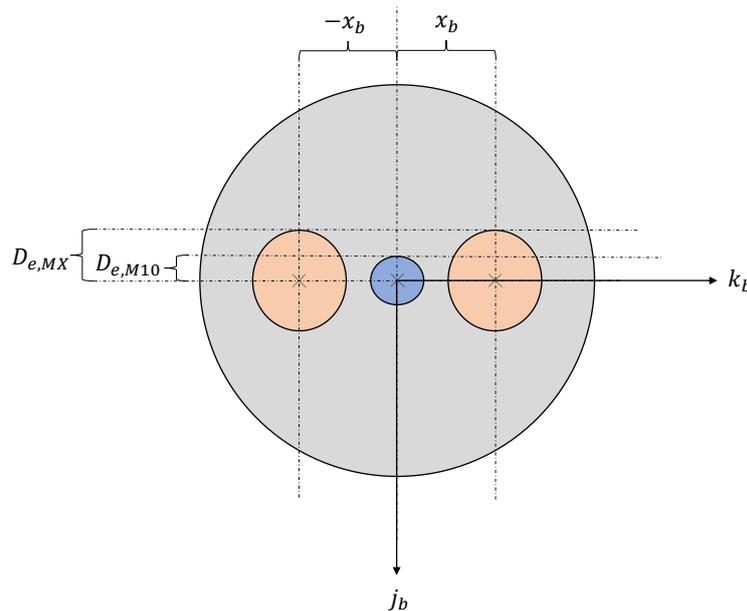


Figure 5.3: MX (orange) and M10 (blue) engine configuration on vehicle body axis for first stage.

In section 4.2.1 the equations for the thrust of the rocket using ideal rocket theory were detailed. The difference between this study and that of [42], [64], [65], and [50] is that some information on the propellant is already given by AVIO. What needs to be determined however is the engine mass and size. This can be done by usage of regression equations formulated by Zandbergen [69]. The M10 engine is a pump fed liquid engine. In [69], mass estimation relationships (MER) for engines running on hydrolox and kerolox were done, but not on methalox. The results however showed that the relationships for kerolox and hydrolox engines can be

used to reasonable accuracy for methalox engines. In addition, if the area ratio is known, the estimation of the length and diameter maybe further improved. The relationship for the mass and sizing of the engine. For the MX stage, whose relationships for sizing and mass found to be applicable for methalox cases are given in tab. 5.3.

Relationship	$R^2$	RSE	Number of Points
$D = .0455 T_{max}^{.2745}$	0.794	14.7%	21
$L = .1362 T_{max}^{.2279}$	0.783	17.6%	22
$M = 1.104 * 10^{-3} T_{max} + 27.702$	0.987	25.8%	25

Table 5.3: Relationships for a hydrolox propellant liquid engine from [69]

There are several parameters used in regression and showed in the Zandbergen study which give insight into the goodness of fit of regression relationships. RSE represents the relative standard error, and is determine by dividing the standard error of the estimate by the estimate itself, and multiplying the fraction by 100. The RSE indicates the reliability of the estimation; higher RSE means less reliable. The study considers that RSE's above 30% are to be considered carefully, despite there being no fixed threshold.  $R^2$  represents the coefficient of multiple determination. This is the degree to which the variation in the dependent variable observations are account for by the regression relationship. This is the difference between 1 and the fraction of the residual sum of squares  $SSR$  and the total sum of squares  $SST$ .

To determine the exit pressure of these engines, the NASA CEARUN online web-based interface for the Chemical Equilibrium with Applications (CEA) software can be used. This program allow calculation of chemical equilibrium compositions of propellants, in this case the methalox used in both engines. The inputs to the program are the  $O/F$ , fuel and oxidizer selection, and combustion chamber pressure  $P_c$ . The program then returns various chemical properties at different  $O/F$  and  $P_c$  ranges selected for the chamber and throat sections of the rocket nozzle. These include most importantly the temperature, pressures, and specific heat ratios  $\gamma$ . Moreover, they give general values of  $c^*$ ,  $C_T$ , and  $I_{sp}$  which are useful for cross validation given the specific impulse and  $O/F$  is already provided by stakeholders.

## 5.2 Stages

The first stage of the launch vehicle consists of two MX engines, and one M10 engine. The second stage is a M10 stage/engine. As mentioned, the MX engine is supposed to produce 6 times more the amount of thrust than the M10. The performance parameters for the thrust, namely the  $O/F$  ratio,  $I_{sp}$ , and  $T_{max}$  are already known. The length of the stages is given by Eq. 5.1.

$$L_{stage,i} = L_{cb,i} + 2r_{press,i} + L_{eng,i} \quad (5.1)$$

The parameters are the following: common bulkhead tank length  $L_{cb,i}$ , circular pressure tank radius  $r_{press,i}$ , and length of the engine  $r_{engine,i}$ . The common bulkhead design for a propellant tank maximizes the amount of volume available for propellant. SpaceX's Falcon 9 and Rocket Lab's Electron RLV's both use common bulkead propellant tanks for Kerolox engines. A design parameter associated with this design is the ellipse ratio  $k_{ellipse}$ , which specifies the eccentricity of the ellipse at the top and bottom of the tank. This factor will be used to determine the diameter of the common bulkhead tank  $D_{cb,i}$ , length of common bulkhead, length of interstage  $L_{inter}$ , and payload adapter  $L_{adapt}$ , and finally mass of common bulkhead tanks  $m_{tank,i}$ . The sections following will detail the procedures of estimating these parameters.

### 5.2.1 Propellant Tank

The diameter for the common bulkhead tank is given in Eq. 5.2.

$$D_{cb,i} = \frac{V_{cb,i}}{\pi \left( \frac{SL-1}{4k_{ellipse}} + \frac{1}{6k_{ellipse}} \right)} \quad (5.2)$$

The diameter requires knowledge of the total volume of the tank, which can be determined based off the knowledge of the  $O/F$  ratio and the total propellant volume itself, shown in Eq. 5.3. The useful volume of the

tank is limited by the effects of ullage, boil-off, and other effects. The correction factor  $k_{te}$  for these parasitic tank effects on the volume is selected as .1 from the work of [50] and [65] and will be selected here as well.

$$V_{cb,i} = (1 + k_{te})(m_{fl,i}\rho_f + m_{ox,i}\rho_{ox}) \quad (5.3)$$

The length of the common bulkhead tank is found by taking the length of both the oxidizer portion (Eq. 5.5), the fuel portion (Eq. 5.6), with the final term adding the extra elliptical portion at the top of the oxidizer tank 5.4).

$$L_{cb,i} = L_{ox,i} + L_{fl,i} + \frac{D_{cb,i}}{k_{ellipse}} \quad (5.4)$$

$$L_{ox,i} = \frac{4V_{ox,i}}{\pi D_{cb,i}^2} - \frac{2D_{cb,i}}{3k_{ellipse}} \quad (5.5)$$

$$L_{f,i} = \frac{4V_{f,i}}{\pi D_{cb,i}^2} \quad (5.6)$$

The mass of the common bulkhead tank can be found by introducing a new parameter for the design factor of ellipsoidal heads called  $E'$ , given in equation 5.7.

$$E' = 2k_{ellipse} + \frac{1}{\sqrt{k_{ellipse}^2}} \ln \left( \frac{k_{ellipse} + \sqrt{k_{ellipse}^2 - 1}}{k_{ellipse} - \sqrt{k_{ellipse}^2 - 1}} \right) \quad (5.7)$$

The mass of the common bulkhead tank can then be written as a function of this parameter, as well as the the local thickness buildup factor  $k_{pt}$  considered for piping and tank connections as shown in equation 5.8. The density of the propellant tank material is given as  $\rho_{cb}$

$$m_{cb,i} = (1 + k_{pt}) \left( \frac{3\pi r_{cb,i}^2 t_{cb,i} E' \rho_{cb,i}}{2k_{ellipse}} + 2t_{cb,i} \pi r_{cb,i} (L_{cb,ox,i} + L_{cb,fl,i}) \rho_{cb,i} \right) \quad (5.8)$$

To accommodate the usage of the cryogenic propellants, tanks need thermal protection systems (TPS). Based on the work done for a multidisciplinary design optimization of ELV, equations 5.9 and 5.10 show the masses needed for both oxidizer and fuel tanks respectively[13]. These are based off H2(L) but will be assumed to be applicable to CH4(L) as well.

$$m_{ox,TPS,i} = .9765(\pi D_{cb,i} L_{ox,i} + \pi D_{cb,i}^2) \quad (5.9)$$

$$m_{f,TPS,i} = 1.2695(\pi D_{cb,i} L_{fl,i} + \pi D_{cb,i}^2) \quad (5.10)$$

## 5.2.2 Pressure System

The final component that adds considerable mass to the stage is the pressure system. Once again, a correction factor  $k_{press}$  is taken here equivalent to 0.1 for connections of piping and local thickness. The mass of the pressure system (Eq. 5.11) is determined by both the mass of the pressurant gas  $m_{g,press}$  and the pressurant tank  $m_{t,press}$ , shown in equations 5.12 and 5.13 respectively.

$$m_{sys,press} = m_{t,press} + m_{g,press} \quad (5.11)$$

$$m_{g,press} = 1.1 \frac{P_{pt} V_{pt}}{R_{g,press} T_{g,press}} \frac{\gamma}{1 - P_{pt}/P_{press}} \quad (5.12)$$

$$m_{t,press} = (1 + k_{press})(4\pi r_{press}^2 t_{press} \rho_{press}) \quad (5.13)$$

$$V_{g,press} = \frac{m_{g,press} R_{g,press} T_{g,press}}{P_{press}} = \frac{4\pi r_{press}^3}{3} \quad (5.14)$$

$$t_{press} = SF \frac{P_{press} r_{press}}{2\sigma_{y,press}} \quad (5.15)$$

### 5.3 Interstage

The interstage is used to connect the first stage to the second stage. Following the work of [50], slanted interstages by selecting their length, conical angle, and required thickness can be designed. Interstages are slanted in order to accommodate the difference in diameters of the first and second stage. An extra 10 cm are account for the engine size to size the interstage, as shown in Eq. 5.16.

$$L_{inter} = L_{eng,2} + .1 \quad (5.16)$$

The conical angle of interstage section  $\Delta_{inter}$  can be found by Eq. 5.17.

$$\Delta_{inter} = \arctan\left(\frac{|D_{cb,1} - D_{cb,2}|}{2L_{inter}}\right) \quad (5.17)$$

In order to determine the mass of the interstage, the thickness of a conical shell must be determined. The thickness is modeled after the largest load the RLV is expected to experience, which is assumed to occur at take-off. The thrust at take off  $F_{T,TO}$  considers the maximum buckling load the interstage will experience and thus the thickness of the shell can be determined by rearranging Eq. 5.18.

$$F_{T,TO} = k_{buckling} \frac{2\pi E(t_{inter}/SF)^2}{\sqrt{3 - (1 - \nu^2)}} \cos^2 \Delta_{inter} \quad (5.18)$$

The mass of the interstage is found by treating it as a conical frustum as given by Eq. 5.19, with base diameter  $D_{cb,1}$  and top diameter  $D_{cb,2}$ .

$$m_{inter} = \pi t_{inter} \rho_{inter} \frac{|D_{cb,1} - D_{cb,2}|}{2} \sqrt{\left(\frac{|D_{cb,1} - D_{cb,2}|}{2}\right)^2 + L_{inter}^2} \quad (5.19)$$

There is another structure present in rockets that covers the forward skirt to the interstage, and the aft skirt to engines. This allows smoothing of the surfaces to result in a better aerodynamic shape of the RLV. The length of this stage (Eq. 5.20) does not influence the total length of the RLV, but it does add to its mass, given by Eq. 5.21.

$$L_{struc} = \frac{2r_{cb,1}}{k_{ellipse}} + 2r_{press} \quad (5.20)$$

$$m_{struc} = 2\pi r_{cb} L_{struc} t_{struc} \rho_{struc} \quad (5.21)$$

### 5.4 Payload Bay

The payload bay is comprised of the vehicle equipment bay mass  $m_{VEB}$ , the payload adapter mass  $m_{adapt}$ , the payload mass  $m_{payload}$ , and the fairing mass  $m_{fairing}$ . The equivalent sizing method and selections are detailed below.

#### 5.4.1 Fairing

The payload fairing mass  $m_{fairing}$  is based off the VEGA-C vehicle [4]. This is a parabolic shaped fairing.

#### 5.4.2 Payload Adapter

The payload adapter uses the MER shown in Eq. 5.22, which is a function of the payload mass  $m_{payload}$  [13].

$$m_{adapter} = .004775 m_{payload}^{1.0132} \quad (5.22)$$

#### 5.4.3 Vehicle Equipment Bay

The vehicle equipment (VEB) mass is given by the MER for the VEB is shown in Eq. 5.23 [13]. The inert mass in the equation refers to the inert masses of the first and second stage combined. This includes everything but the propellant masses (fuel, and oxidizer) shown in fig.5.1, and the payload bay mass.

$$m_{VEB} = .3672 m_{inert,1+2}^{.6798} \quad (5.23)$$

### 5.4.4 Payload

The payload mass for the configuration has two options; 1000 kg, and 2000kg. This is restricted by REQ-LV-2.

- REQ-LV-2 The RLV shall be flown in two payload classes: 1000 kg and 2000kg with the rest of the architecture, aside the propellant masses, remaining the same.

The reason for this follows from the work of [14] as was identified only one RLV exists in this range, being Electron's. This payload mass will allow more tuning given the restriction of structural index imposed by REQ-LV-1. A clarification is made here that this requirement allows remodeling of the adapter mass which can be estimated by the MER in Eq. 5.22, which is a function of the  $m_{payload}$  itself.

## 5.5 Landing Legs

A key difference in the configuration of an ELV to that of a RLV is the inclusion of landing legs. These technically take part of the overall system of the first stage, but have been separated here as their own system to simplify the graph shown in fig.5.1. Following the work of [14], it was mentioned that landing legs are stowed against the first stage outer circumference, does not have effect on the aerodynamics of the RLV. Previous research for landing leg mass estimation considered it being 10% of the first stage inert mass  $m_{inert,1}$ . This was used in Tartabini's study where the extra 10% was considered an additional increase from a ELV configuration to a RLV [61]. Blau's work validates this assumption using the estimations of the Falcon 9's vehicle mass, resulting in an increase in 10% of the GLOM due to the landing gears [9]. Price et. al. also saw a similar 9.41% increase for a Mars-lander with the inclusion of landing gears [48].

## 5.6 Launch Vehicle

A simplified schematic of the RLV and its associated geometry and design variables are shown in fig. 5.4. From now on, this RLV will be referred to as the VEGARLV, with the number indicating the number of stages. For each stage, and the entire launch vehicle, the configuration must adhere to REQ-LV-1 repeated below.

- RQ-LV-1 The structural index of the entire launch vehicle and each individual stage should be within the range of .08 and .12

The structural index  $SI$  can be calculated by Eq. 5.24, where  $m_{inert}$  can be defined to be the inert, dry mass of the entire launch vehicle or the an individual stage's  $m_{inert,i}$  inert mass. The same is true for the propellant mass  $m_{prop}$  which when considering the SI of a single stage Eq. 5.24 uses the propellant mass of that stage  $m_{prop,i}$ .

$$SI = \frac{m_{inert}}{m_{inert} + m_{prop}} \quad (5.24)$$

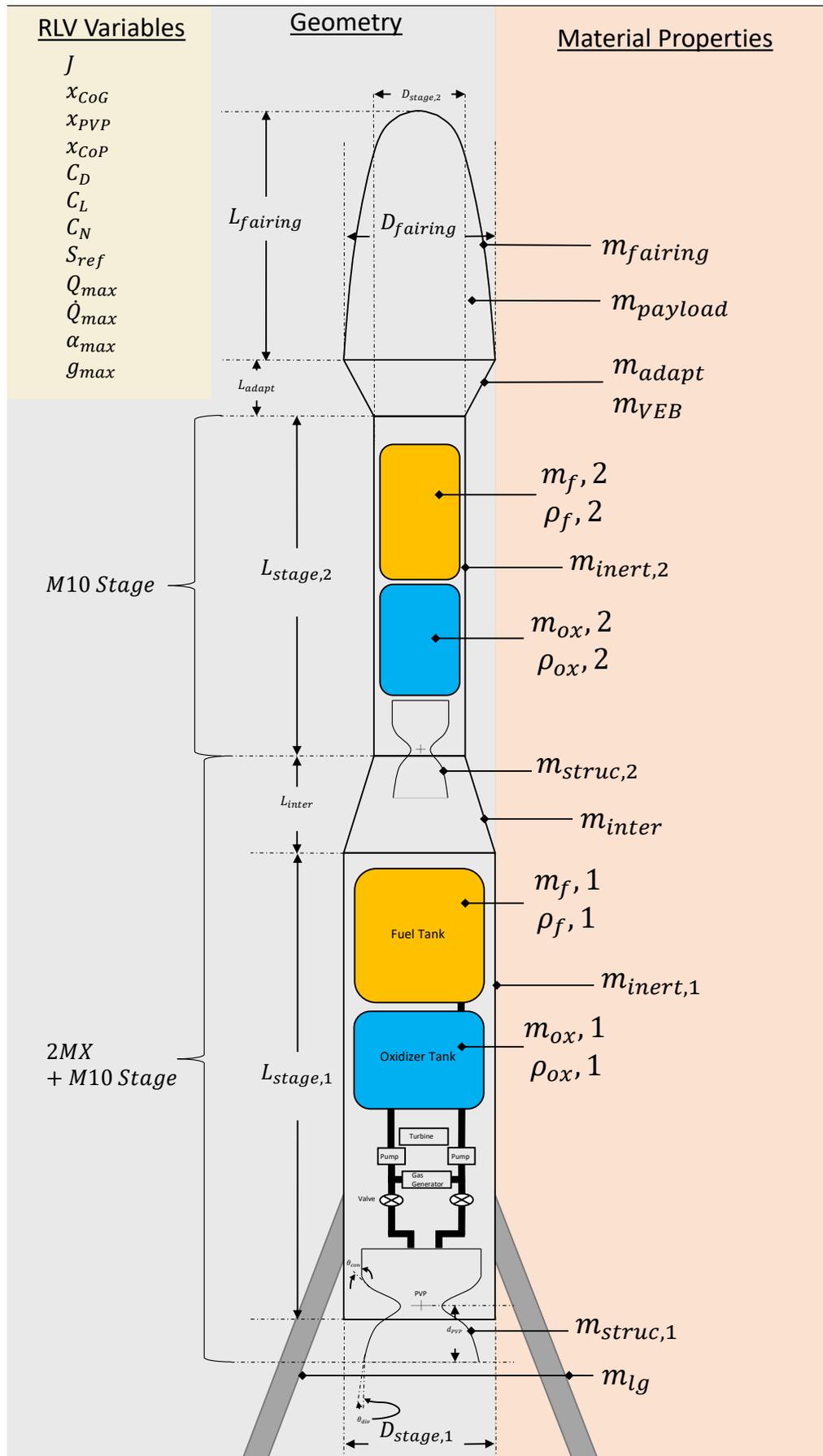


Figure 5.4: RLV geometry, variables, and material properties over simplified schematic for the 2\*MX+M10 first stage and M10 second stage. Fuel tank, oxidizer tank, turbine, pump, and gas generator block diagram with valves shown on the first stage has been adapted from [71]

## Chapter 6

# Launch Vehicle Optimization

The launch vehicle optimization process includes defining the geometry and performance of the RLV. This includes vehicle sizing of both the second and first stages, as well as the payload. Whilst the trajectory optimization and the purposes of this study are to examine feasibility of return trajectories of the first stage, the entire RLV, according to available statistical and tabulated data of mostly VEGA family launchers, is generated. This is because in the preliminary study for the selection of the starting conditions for descent an analytical ascent trajectory will be calculated with the entire RLV. Additionally, MX and M10 engine characteristics are finalized through the process of engine sizing. Moreover, the main procedure involving optimization of the RLV will include sizing and mass determination of both the dry and wet mass of the vehicle. Such formulas, which have been presented in section 6 are interdependent on other RLV characteristics. The most isolated optimization procedure is the mass and engine sizing itself which is done through tabulation of data available on NASA CEA. The rest of the vehicle is optimized using heuristical optimization, with the inclusion of non-linear optimization for sub-problems involved in this procedure. This section will lead off with the explanation of how the data from NASA CEA run was compiled, followed by how the multiobjective genetic algorithm was written to complete the optimization procedure.

### 6.0.1 NASA CEA Runs

NASA CEA Run was given inputs of chamber to exit pressure ratio, propellant mixture ratio, and chamber pressure of both the M10 and MX engines. This was done once as a single problem. The minimum and maximum values of the each of the inputs for the different engines and the corresponding intervals is displayed in table 6.1. The intuition behind the selection of these ranges was done based on the compiled data of liquid propulsion engines given by Rozemeijer[50] and Contant[14].

The chemical equilibrium problem type to derive characteristics of the engines was selected as the rocket problem. Next, the pressures, which are considered as the chamber pressures of the engine, the fuel and oxidizers, the oxidizer to fuel ratio, and the ratio of chamber to exit pressures are selected. The only constant for the M10 engine is the oxidizer to fuel ratio, and the assumption has been made that this oxidizer to fuel ratio is also true for the MX, given that it is meant to be an upscaled version of the M10. The process of generating all the data requires multiple user runs as the maximum number of values that can be given to the chamber to exit pressure ratio is 16. Following work from other students, the range of feasible chamber to exit pressures range from an order of magnitude of  $10^3$  to  $10^6$ . Runs with ratios lower and higher than this were attempted, but results ultimately showed that they were not feasible.

Whilst NASA CEA run generates and tabulates data for the engines, it cannot select a specific combination of values for these variables given a particular specific impulse that the user wants. Therefore, the second procedure after generation of data is to transport the results of the runs in MATLAB or any other software that can directly import the output data. NASA CEA also outputs the specific impulse for these runs. Moreover, it outputs the specific heat ratio that is needed to solve equation 6.1. This is the Vandekerckhove function and it is only dependent on the specific heat ratio. Next, the ideal thrust coefficient is calculated, dependent on the exit to chamber pressure ratio, specific heat ratio, and the Vandekerckhove result as seen in equation 6.2. In equation 6.14 the ideal exhaust velocity of the nozzle given the chamber temperature and the ideal gas constant, as well as the Vandekerckhove result is calculated. Finally, equation 6.4 shows how to obtain the ideal specific impulse of the engine using the result of the thrust coefficient and exit velocity, as well as the sea

level gravitational acceleration. This final value is crucial because it allows us to determine, with a given error margin, the engine characteristics of the MX and M10 that are most loyal to the given stakeholder standards. The way to do this is to calculate the quality factor as defined by Zandbergen [69], of the specific impulse as shown in equation 6.5.

$$\Gamma = \sqrt{\gamma} \left( \frac{2}{\gamma+1} \right)^{\frac{\gamma+1}{2(\gamma-1)}} \quad (6.1)$$

$$C_{F,id} = \Gamma \frac{\frac{\sqrt{2\gamma}}{\gamma-1} \frac{\gamma-1}{\gamma}}{1 - \frac{P_e}{P_c}} \quad (6.2)$$

$$c_{id} = \frac{1}{\Gamma} \sqrt{RT_c} \quad (6.3)$$

$$I_{sp,id} = \frac{C_{F,id} c_{id}}{g_0} \quad (6.4)$$

$$Z_{sp} = \frac{I_{sp,real}}{I_{sp,id}} \quad (6.5)$$

For convenience of the reader that the MX and M10 real specific impulses are 317 and 364 respectively. This process of backward engineering allows us to obtain a range of possible M10, and MX engine configurations. Finally, with the viable candidates separated from the rest of the CEA results, the engine diameters are then ultimately calculated in the following approach. A parameter  $K$  is calculated with the known exit velocity  $U_e$  that is given by the CEA run results. This also uses the exit exit pressure, chamber pressures and temperatures, specific heat ratio, and specific impulse quality factor, as show in equation 6.5. Then, with the knowledge of the vacuum thrust of the engines, equation 6.9 calculates the exit area, which then is converted into diameter as shown in equation 6.7, given the nozzle of the engine is perfectly circular.

$$U_e = \sqrt{2 \frac{\gamma}{\gamma-1} \frac{R_A}{M} T_c \left( 1 - \frac{P_e}{P_c} \frac{\gamma-1}{\gamma} \right)} \quad (6.6)$$

$$D_e = \sqrt{\frac{4A_e}{\pi}} \quad (6.7)$$

$$K = Z_{sp} \frac{P_c}{\sqrt{RT_c}} \left( \sqrt{\frac{2\gamma}{\gamma-1} \left( 1 - \frac{P_e}{P_c} \frac{2}{\gamma} U_e + P_e \right)} \right) \quad (6.8)$$

$$A_e = \frac{F_{T,vac}}{K} \quad (6.9)$$

Variable	$P_c$ (bar)	O/F	$\frac{P_c}{P_e}$
Range	5:5:100	3.4	10:10:480 500:100:10000 10000:1000:15000
No. of inputs	20	1	150

Table 6.1: Total variations for NASA CEA run. 3000 different points evaluated in total for varying  $P_c$  and  $\frac{P_c}{P_e}$

## 6.0.2 RLV sizing

The vehicle sizing procedure is encapsulated in a multiobjective genetic algorithm. The general overview of the procedure is as follows. The inputs to the genetic algorithm are the first stage diameter, common bulkhead end ellipse ratio, and the length of the oxidizer section. The selection of these 3 components has been chosen so as to reflect their variability with respect to other factors. These variables have a continuous analytical relationship with other factors, and as such it is advantageous to set up the search space in the GA for such variables. There are other variables that are used in this procedure but because of their discrete

nature they are called from outside the genetic algorithm. As such multiple runs of the GA to cover all the possible combinations of these variables is selected. The first step is the selection of the MX and M10 engines based on the first stage diameter. This can be determined simply by considering a 10cm distance between motors assembled as seen in figure 6.1 and whose overall diameter must encompass the diameter of the first stage as seen in equation. Next, the first stage propellant tank is sized, followed by the second stage. At this point, the algorithm reaches a criteria check to determine whether the structural index is within a certain threshold. If it is not, the structural index is improved by decreasing the propellant mass. If it is, the structural mass (dry mass) of the vehicle is increased instead. Finally, the propellant mass is checked again to ensure the most optimal structural ratio for the candidate possible. The genetic algorithm does this for every single candidate solution. Below, further detail on how these procedures are carried out are detailed. The majority of the relations used in this procedure are given by Huzel and Huang.

### MX and M10 fitting

The first calculation when beginning to evaluate the objective function of a candidate solution is determining which, out of the available engine choices, is the most optimal. Two criteria determine this choice. First, from the tabulated data of feasible engine configurations a mesh is created out of the two variables: ID of M10 and MX. All of the possible combinations are then subjected to a check to see whether the engine nozzles do not extend beyond the first stage diameter of the RLV. We recall that this is an input to the RLV from the overall multiobjective GA procedure. Then, the second criteria is to determine which of these has the most optimal thrust at sea level. Equation 6.10 shows the criteria that is evaluated for this condition.

$$\min |P_{sea,tot} = 2(P_{a,sea} - P_{e,MX})\left(\frac{\pi}{4}D_{e,MX}^2\right) + (P_{a,sea} - P_{e,M10})\left(\frac{\pi}{4}D_{e,M10}^2\right)| \quad (6.10)$$

This criteria finds the minimum value of the relationship, which corresponds to the configuration that will have the lowest exit pressure losses at sea level. The exit pressures and diameters of the M10, and MX engines are all tabulated matrices in the procedure and therefore renders this a simple row and column index search. This then gives the equivalent values of the pressures and diameters, as well as the engine ID that is stored in the tabulated results.

The purpose of doing this, which is true amongst many other procedures in this discipline, is to attempt to create candidates that adhere to best the main purpose of the study. This is to determine feasibility for the RLV of the first stage. When beginning descent the RLV experiences little to no pressure losses due to the pressure differential between nozzle exit and environment. However, at around 4000-5000 meters of altitude, pressure losses begin to amount, and considering from previous studies of re-entry, the RLV begins a steadier and slower decline at this atmospheric region. This is in part due to the aerodynamic forces that are building up but additionally also due to the fact that the RLV needs to take into account all of the errors that have build up prior in the descent (guidance solution failure, model mismatch, sensitivity study errors). As a result, and again as reported in [20], the required thrust can spike up to the maximum achievable with similarly written convex algorithms. It is this combination of increased dynamics and need to correct previous errors that warrants careful attention to the selection of exit pressures earlier in the study. This procedure attempts to partly cover this responsibility by making the RLV operational at lower altitudes. It is certainly more intuitive to create a use-case of the engine from scratch, that is select an available thrust that functions more correctly, but the scope of this study calls for the use of the MX and M10 engines specifically.

### Stage Sizing

The sizing of the first stage is based on the common bulkhead characteristics as reported by Huzel and Huang. A common bulkhead configuration refers to a launch vehicle body where the head of the stage serves both as the junction to the upper stage and also as the head of the propellant tank its stage. As its name suggests, the oxidizer and fuel section are separated by a common bulkhead which together look like a capsule shaped tank. Individually, both tanks have a cylindrical portion and an elliptical portion which may be protruding outwards for the head of the tank (fuel portion) or inwards (oxidizer portion). These two sections when referring to the lengths  $l$  physical characteristics such as density  $\rho$  are denoted by subscripts  $c$  and  $e$  for cylindrical and elliptical, respectively. Moreover, this design has already been chosen on Rocket Lab's Electron and by a previous study on re-entry [50]. The common bulkhead ends are ellipsoidal shaped, and have limited eccentricity. In fact, their eccentricity is limited by the relationship reported by Huzel and Huang of Combined stress factor  $K$  against ellipse ratio  $k$ . The ellipse ratio of the tank is given by equation 6.11. This is however

given as a direct input to the genetic algorithm, bounded between the limits of 1 (completely spherical) and 2 (highest eccentricity). The variable  $D_{e,cb,1}$  denotes the diameter of the elliptical end of the common bulkhead, with the subscript index after the second comma denoting that it is for the first stage. Only the end that interfaces with the interstage is elliptical, whilst the lower portion is spherical. The purpose of doing this is to simplify the design as introducing another point of eccentricity introduces not just one but many new variables for the GA to handle.

$$k_{ellipse} = \frac{D_{cb,1}}{D_{ellipse,cb}} \quad (6.11)$$

Next, the mass of the oxidizer and fuel section of the propellant tanks is calculated using relations 6.12 and 6.13 respectively. The variable  $l_{co}$  refers to the length of the cylindrical portion of the oxidizer section.

$$m_{ox} = \rho_{ox} \left( \left( \frac{8\pi}{3} a^2 \right) b + \pi a^2 l_{co} \right) \quad (6.12)$$

$$m_f = \frac{m_{ox}}{OF} \quad (6.13)$$

The volume of the fuel can then be calculated using equation 6.14.

$$V_f = \frac{m_f}{\rho_f} \quad (6.14)$$

Then, the length of the cylindrical portion of the fuel section is calculated via equation 6.15.

$$l_{cf} = \frac{V_f}{\pi D_{cb,1}^2} \quad (6.15)$$

The thickness  $t_c$  of this cylindrical section is then calculated with equation 6.16. SF refers to a scale factor that has been chosen as 1.5 for this study and also adheres to most selections used in [13].

$$t_c = SF \frac{P_{p,t_f} a}{\sigma_y} \quad (6.16)$$

Finally, the design factor of the ellipsoidal tanks is completely dependent on  $k_{ellipse}$  and it is given by equation 6.11. This can then be used to find the mass of the elliptical ends of the oxidizer and fuel sections as given in equations 6.17 and 6.18 respectively.

$$m_{e,o} = \pi D_{cb,i}^2 t_{e,o} E' \frac{\rho_{ppt}}{2k_{ellipse}} \quad (6.17)$$

$$m_{e,f} = m_{e,o} \frac{t_{e,f}}{t_{e,o}} \quad (6.18)$$

The length of the engines are the only parameters related to the engines that are not determine in the engine sizing procedure but are rather fixed constants based off the relationship found by Zandbergen for liquid propulsion systems using regression analysis, and these are given in equation 6.19 and 6.20.

$$l_{eng} = .1362 F_{T,vac}^{(2.2279)} \quad (6.19)$$

The same is true for the masses of engines, another relation wholly dependent on the vacuum thrust.

$$m_{eng} = (1.104E - 3) F_{T,vac} + 27.702 \quad (6.20)$$

Ultimately these calculations allow us to find the total structural (dry) mass of the first stage, given by equation 6.21. Note that this equation also incorporates the thermal protection system mass given by equations 6.22 and 6.23. Moreover, the factor of 1.1 in front of the 2 relates to the landing leg mass that incorporates 10% of the total stage 1 mass as discussed in section 5.5.

$$m_{struc,1} = 1.1 * 2\pi D_{cb,1} t_c (l_{c,f} + l_{c,o} + 2D_{e,1}) \rho_c + m_{tps} \quad (6.21)$$

$$m_{ox,TPS,i} = .9765 (\pi D_{cb,i} L_{ox,i} + \pi D_{cb,i}^2) \quad (6.22)$$

$$m_{f,TPS,i} = 1.2695(\pi D_{cb,i} L_{f,i} + \pi D_{cb,i}^2) \quad (6.23)$$

The various thicknesses of the tank include the wall thickness at the knuckle  $t_k$ , the wall thickness at the crown  $t_{cr}$ , the wall thickness on the ellipsoidal tank-end  $t_e$ , wall thickness at the spherical tank-end  $t_s$ , and wall thickness of a cylindrical tank section  $t_c$ . The calculation for these thicknesses are shown from equation 6.24 to 6.29. These thicknesses naturally have varying calculations based on section of oxidizer of fuel and are denoted by the corresponding subindex in the equation set. We recall that the reason for this is because the oxidizer section contains a lower head that is spherical, and an elliptical one at the top. The fuel section instead has an elliptical head at the top and an indented ellipse at the bottom.

$$t_{k,o} = \frac{SF KP_{pt,o} D_{cb,1}}{\sigma_\gamma} \quad (6.24)$$

$$t_{cr,o} = \frac{SFP_{ppt,o} k_{ellipse} D_{cb,1}}{2\sigma_\gamma} \quad (6.25)$$

$$t_{e,o} = \frac{t_{k,o} + t_{cr,o}}{2} \quad (6.26)$$

$$t_{k,f} = t_{k,o} \frac{P_{pt,f}}{P_{pt,o}} \quad (6.27)$$

$$t_{cr,f} = \frac{t_{cr,o} P_{pt,f}}{P_{pt,o}} \quad (6.28)$$

$$t_{e,f} = \frac{t_{k,f} + t_{cr,f}}{2} \quad (6.29)$$

The interstage is the section that connects the first stage to the second stage and its length can be determined adding a 10cm margin to the value of the length of the engine fitted on the second stage (M10). As a result, the relation for the interstage length is given in equation 6.30.

$$l_{inter} = l_{M10} + .1 \quad (6.30)$$

After completion of the first iteration of calculation of the first stage parameters, the second stage is evaluated using the same equations. Note that the second stage uses the same  $k_{ellipse}$  from the first stage, whose reasoning returns back to the point of keeping the program simple and reducing the number of variables for the genetic algorithm to propagate. For the purposes of this experiment, the interest is not in generating multiple feasible results because even a single result will still allow variation of parameters of the RLV such as propellant in the tank. Moreover, aside from the general continuous variables that the GA is fed, there are also discrete variables which further increase multiplicity in the results (e.g. changing the material density will unequivocally vary the mass, and even if the mass remains the same across different materials, then it strengthens the argument that it may be the size restrictions that drive the RLV design, not the mass. The opposite is also true).

Finally, the first iteration of the RLV of the candidate solution can be generated by calculating the total inert mass given in equation 6.31. Note that this relationship also has a factor of 1.1 in the front, and this time it is applied to the total inert mass of the RLV has a margin for wiring, tubing and other components apart from the ones mentioned.

$$m_{inert,tot} = 1.1 \sum_{i=1}^2 (m_{struc,i} + m_{b,e,i} + m_{m,e,i}) + 2m_{M10} + m_{MX} \quad (6.31)$$

## Payload Sizing

The payload sizing and mass depends on the payload mass of the instruments chosen (a discrete parameter), the total inert mass of the stages given by equation 6.31, and the diameter of the second stage  $D_{cb,2}$ . Following Castellini's work [13], we can use regression relationships for sizing the various components that encompass the payload: payload fairing, vehicle equipment bay, instruments (also simply referred to as payload), and the payload adapter. As was shown in [13], reusable launch vehicles may have multiple types of forms, with the most typical for multistage vehicles being a larger diameter of the first stage and smaller diameter of the second, with an increase in diameter for the payload fairing. Amongst mass considerations, it was noted that there are also aerodynamic factors that come into play. For the purposes of this study, the aerodynamics of

the payload fairing and hence the first and second stack of the launch vehicle are not of importance as the preliminary analysis used to obtain the starting conditions for descent do not take them into play. Furthermore, in case any of the candidate's solution was taken to analyse its aerodynamic feasibility the payload fairing may be easily morphed into a different design with negligible changes in the overall structural index of the vehicle. This is assuming that the payload itself is a volume of a cylindrical shape whose radius is freely adaptable to suit a require size and and shape of the fairing for aerodynamic stability. The diameter of the fairing is taken as the largest value between 2.62, which is reported in [4] for the VEGA-C, and the diameter of the first stage. The diameter of the payload (instruments) is shown in equation 6.40, where the factor  $k_{D_{pl}}$  is taken from [13], and is 1.12.

$$D_{pl} = \frac{D_{fairing}}{k_{D_{pl}}} \quad (6.32)$$

The lengths of the fairing and adapter can also be calculated with such regression relationships from [13]. These are given in equations 6.33 and 6.34.

$$l_{fairing} = \frac{1}{2}2.313D_{pl} + .15D_{pl} \quad (6.33)$$

$$l_{adapt} = .15D_{pl} \quad (6.34)$$

The payload fairing can be calculated by obtaining first the density of the fairing, as given by equation 6.37. Then, the area of the fairing, which has been chosen from [4] to be a parabolic shape that represents best that of the VEGA-C is given by equation 6.36.

$$\rho_{fairing} = \frac{(\frac{D_{fairing}}{2})^2 + l_{fairing}^2}{2\frac{D_{fairing}}{2}} \quad (6.35)$$

$$A_{fairing} = l_{fairing}\pi(\sqrt{\rho_{fairing}^2 - l_{fairing}^2} + \frac{\rho_{fairing}^2}{2} \arcsin \frac{l_{fairing}}{\rho_{fairing}} - l_{fairing}(\rho_{fairing} - \frac{D_{fairing}}{2})) \quad (6.36)$$

The mass of the fairing is then given by equation 6.37, which is equal to  $A_{fairing}$  times a factor of 13.3 given by regressional analysis.

$$m_{fairing} = 13.3A_{fairing} \quad (6.37)$$

The vehicle equipment bay, adapter, and total mass of the payload is then given by equations 6.38, 6.39, and 6.40 respectively.

$$m_{veb} = .3762m_{inert,tot}^{.6798} \quad (6.38)$$

$$m_{adapt} = .0477526m_{pay}^{1.01317} \quad (6.39)$$

$$m_{p,tot} = m_{pl} + m_{fairing} + m_{veb} + m_{adapt} \quad (6.40)$$

The mass of the interstage can be calculated by determining first the slant angle that is formed between the first and second stage as shown in equation 6.42. This gives the mass relationship shown in equation 6.43, using the thickness formula for the interstage as shown in equation 6.41. The slant angle equation considers both a increasing and decreasing diameter from stage 1 and stage 2, as noticed by the absolute term of the difference between the two diameters.

### Structural Indices

The structural index, whose formula for the total index is shown in equation 6.46, is evaluated for each candidate GA individual with the first and second stages as shown in equations 6.44 and 6.45 respectively. This equation includes yet another 1.1 factor for the total dry mass of the first stage. This is applied again to account for a margin of error that considers weight of tubing, instruments, and pipes connecting the stage to the engines. Note that the first letter of the subscripts of the masses  $m_{b,e,l}$ ,  $m_{m,e,l}$  and  $m_{t,e,l}$  refer to the bottom lid, middle section, and top lid of the stage, respectively, which are just treated as flat plates of equal diameter to the stage. The general formula of equation 6.46 is still applicable here as the structural indices are always taken as the inert mass divided by the total mass.

$$t_{inter} = (F_{vac,tot,1} \sqrt{3(1 - t_{gauge,min}^2)}) \left( \frac{SF^2}{k_{buck} 2\pi E \cos \delta_{inter}^2} \right) \quad (6.41)$$

$$\delta_{inter} = \frac{\arctan|D_{cb,1} - D_{cb,2}|}{2l_{inter}} \quad (6.42)$$

$$m_{inter} = \pi t_{inter} \rho_{inter} \left( \frac{D_{cb,1} - D_{cb,2}}{2} \right) \sqrt{\frac{|D_{cb,1} - D_{cb,2}|^2}{2} + l_{inter}^2} \quad (6.43)$$

$$\epsilon_1 = \frac{((m_{M10} + m_{MX}^2) + 1.1(m_{b,e,1} + m_{m,e,1} + m_{t,e,1} + m_{struc,1}))}{m_{ox} + m_f + ((m_{M10} + 2m_{MX}) + 1.1(m_{b,e,1} + m_{m,e,1} + m_{t,e,1} + m_{struc,1}))} \quad (6.44)$$

The structural index of the second stage can similarly be calculated as given in equation 6.45.

$$\epsilon_2 = \frac{1.1(m_{struc,2} + m_{b,e,2} + m_{m,e,2} + m_{t,e,2}) + m_{eng,2}}{1.1(m_{struc,2} + m_{b,e,2} + m_{m,e,2} + m_{t,e,2}) + m_{eng,2}} \quad (6.45)$$

$$\epsilon_{tot} = \frac{m_{inert,tot} + m_{pl}}{m_{ox,tot} + m_{f,tot} + m_{pl} + m_{inert,tot}} \quad (6.46)$$

### Center of Gravity of First Stack

The center of gravity of the vehicle first stack is calculated to be used for calculation of the von mises stress criterion validation for the shell thickness of the first and second stages. The calculation for this is included in appendix A.1.

### First Von Mises Stress Validation

Von mises stress criterion is used to obtain the minimum required stress for the launcher to not fail under axial, tensile, and compressive (buckling) forces and moments at launch. There are two validations for each GA candidate and the first is triggered after calculation of  $\epsilon_2$ . Before describing the equations, and important note must be made here as to why the first validation begins with the second stage and not the first.  $N_{UTS}$ ,  $N_{YTS}$ , use  $N_{x,t}$  and  $N_{x,c}$  to calculate  $t_{s,YTS}$ ,  $t_{s,UTS}$ , and  $t_B$ . What these have in common is that they use  $M_{bending}$  which is dependent on  $x_{cg,I}$ . This is dependent on the entire vehicle as it is part of the first stack. As a result, in order to apply von mises stress criterion to improve the structural indices, it must be done from the top up as changing  $x_{cg,I}$  will change the values of the von-mises calculations. This is due to the fact that the second stage has imposed limitations on the wet mass whilst the first stage is allowed to have a variable wet mass. An important point is to be made here about the selection of  $F_{T,vac,1}$  as the force that induces the bending moment. As Castellini [13] mentions,  $M_{bending}$  is due to external flight loads, and therefore should be imparted by aerodynamic forces. A far-reaching assumption is made here that the maximum bending moment will be equal to the thrust force. The first reason this is made is related to the fact that in earlier runs that did not have von-mises stress validation, the structural index of the vehicle was low, thus the thrust to weight ratios of candidate solutions was high. This means that the thrust can be used as a very conservative estimate. When faced with the alternative of using a safety factor to cover for the lower value of moment the vehicle is built to resist, this is hindered by the computational cost of the procedure. No aerodynamic data of the vehicle is generated in the GA run to avoid longer load times.

### Second Von Mises Stress Validation

Once the first calculations of the candidate are complete, a check is made on  $t_{c,1}$  and  $\epsilon_1$ . This check was added as a result of previous testing done with the procedure where without it there was a tendency for the GA to find solutions that were below the structural index minimum requirement of 0.8. This was noted because the equations given by Huzel and Huang give a small value of  $t_c$ , and that it does not vary much with an increase in length of the vehicle. A low value of thickness not only reduces the mass of the vehicle thus making it fall below the minimum requirement of structural index. It can also render the vehicle more susceptible to damages under loads at lift-off where they are considered to be greatest due to the anti-parallel forces of thrust and weight, which both are maximized in this period of flight. As such, deviating from historical procedures used for these types of re-entry studies [50][14][65][64] that optimized RLV sizes and masses, a calculation of the von-mises required thickness is made. This procedure is done by Castellini [13], and considers the worst-case internal running loads of axial compression, axial tensile, hoop, and shear. There are also secondary flight loads that act on the vehicle's thrust frame but their equations consider forces that factor in the mass of the engines only, and as such result in much smaller values of force. Instead, by considering the tensile

forces, bending moment  $M$ , and radius of the first stage  $D_{cb,1}$ , we can derive the shear stress  $N_{x,c}$  and axial compression stress  $N_{x,c}$  formula shown in equation 6.50 and 6.51 respectively.

$$P_t = F_{T,vac,tot,1} - m_{eng,tot,1}(11g_0) \quad (6.47)$$

$$P_c = P_t \quad (6.48)$$

$$T_{xy} = \frac{F_{T,vac,tot,1}}{\pi \frac{D_{cb,1}}{2}} \quad (6.49)$$

$$N_{x,t} = \frac{P_t}{2\pi \frac{D_{cb,1}}{2}} + \frac{M}{\pi \left(\frac{D_{cb,1}}{2}\right)^2} - \frac{pD_{cb,1}}{4} \quad (6.50)$$

$$N_{x,c} = \frac{P_c}{2\pi \frac{D_{cb,1}}{2}} + \frac{M}{\pi \left(\frac{D_{cb,1}}{2}\right)^2} + \frac{pD_{cb,1}}{4} \quad (6.51)$$

To determine whether  $t_{c,1}$  needs to be improved, it is compared to the minimum gage shell thickness  $t_{s,mg}$  (equation 6.54), minimum shell thickness for yield strength  $t_{s,YTS}$  equation 6.60, minimum shell thickness for ultimate tensile strength  $t_{s,UTS}$  equation 6.61, and finally minimum shell thickness for buckling  $t_{s,B}$  (equation 6.58). Furthermore, for  $t_{s,YTS}$  and  $t_{s,UTS}$  the shear stress  $N_{xy}$ , and hoop stress  $N_y$  formulas are given in equation 6.53, and 6.52 respectively. For  $N_{UTS}$  and  $N_{YTS}$  both are evaluated using values of  $N_{x,c}$  and  $N_{x,t}$  and picking the result with the highest stress. There are also safety factors used in this equations, which are  $SF_{YTS} = 1.1$  and  $SF_{UTS} = 1.25$ . The ultimate yield strength  $\sigma_{YTS}$  and ultimate tensile strength  $\sigma_{UTS}$  are both discrete parameters fed to the GA and varied across different GA runs.

$$N_y = P_{internal} \frac{D_{cb,1}}{2} \quad (6.52)$$

$$N_{xy} = \frac{2T_{xy}}{\pi D_{cb,1}} \quad (6.53)$$

$$t_{s,mg} = t_{min,gage} K_{mg} \quad (6.54)$$

$$t_{s,YTS} = \frac{N_{YTS}}{\sigma_{YTS}} \quad (6.55)$$

$$t_{s,UTS} = \frac{N_{UTS}}{\sigma_{UTS}} \quad (6.56)$$

$$N_{UTS} = SF_{UTS} \left( \frac{1}{2} \sqrt{N_x + N_y} \pm \sqrt{\frac{1}{4} (N_x - N_y)^2 + N_{xy}^2} \right) \quad (6.57)$$

$$t_{s,B} = \sqrt{\frac{N_{x,c} L_f}{\epsilon E}} \quad (6.58)$$

$$N_{YTS} = 1.1 \sqrt{N_x^2 + N_y^2 - N_x N_y N_{xy}^2} \quad (6.59)$$

$$t_{s,YTS} = \frac{N_{YTS}}{\sigma_{YTS}} \quad (6.60)$$

$$t_{s,UTS} = \frac{N_{UTS}}{\sigma_{UTS}} \quad (6.61)$$

In order to calculate  $t_{s,B}$  the optimal frame spacing  $L_{f,opt}$  must be calculated using equation 6.62. This is dependent on the buckling efficiency  $\epsilon_{buckling}$ , Shanley's constant  $C_f$  (1/16000), and frame stiffness coefficient  $K_f$ , the latter which is a discrete variable dependent on the material selection chosen for the GA run.

$$L_{f,opt} = \sqrt{6 \left( \frac{D_{cb,1}}{2} \right)^2 \sqrt{C_f \pi \frac{b_{eff}}{K_f}}} \quad (6.62)$$

Ultimately, a new first stage shell thickness is calculated  $t_{c,new}$  that takes the largest value out of all the aforementioned thickness calculations. If the new shell thickness is greater than the old then  $m_{struc,1}$  is recalculated using  $t_{c,new}$ . Note that from equation 6.63, if the new thickness is not greater than the old thickness, and the  $\epsilon_1$  was less than the minimum requirement, an  $SF$  of 1.5 is applied to the old value, and recalculation is done using this  $t_{c,new}$ .

$$t_{c,new} = \begin{cases} \max(t_{s,mg}, t_{s,YTS}, t_{s,UTS}, t_{s,B}) & \text{if } t_c < t_{s,mg}, t_{s,YTS}, t_{s,UTS}, t_{s,B} \\ t_{c,new} = SF t_c & \text{else} \end{cases} \quad (6.63)$$

### 6.0.3 Missile DATCOM Database

In order to complete the design of the RLV, aerodynamic coefficients are generated to define  $F_D$  and  $F_L$  in the closed loop trajectory. The software used for generating these trajectories is missile DATCOM. This software has been used in precedent research at TU Delft on RLV [50]. The programme is a stand-alone executable code that reads input in a FORTRAN format, and then returns the aerodynamic derivatives that can be used to calculate the aerodynamic characteristics of the RLV. This software uses empirical and semi-empirical relations and is considered as a preliminary design tool. The purpose for using this program is to focus on the scope of the research to show mainly feasibility on RTLS based on the guidance strategy. As such, a higher fidelity model to generate the coefficients is not needed here. Below, the procedure for generating the aerodynamic database of coefficients for the RLV is discussed.

#### Flight Conditions

The flight conditions for points at which to evaluate the aerodynamic coefficients must first be selected. These are shown in table 6.2, and are largely based on the work done by Rozenmeijer[50]. One difference however, is that because this study deals with re-entry, and that specifically of a fly-back RTLS,  $\alpha$  can vary considerably, especially at the beginning when the downrange velocity inversion has to be applied. This is also true for the fact that contrary to an ascent profile, the downwards acceleration  $\dot{v}_D$  in the NED frame can also vary considerably with respect to the velocity, as the thruster needs to fire in free fall whilst its nose is pointing up. In a 3-DoF environment, this can mean a range of  $\alpha$  that varies from 180deg to  $-180$ deg. As such, the range for  $\alpha$  value is increased to cover this. The altitude is kept at a constant of 30000 m as it is assumed that compared to the other input of  $M$  and  $\alpha$  it does not influence significantly the coefficients.

Parameter	Ranges
$M$	0, 0.3, 0.6, 0.8, 0.9, 0.95, 1.0, 1.05, 1.1, 1.2, 1.3, 1.4, 1.6, 1.8, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0
$\alpha$ (deg)	-180:5:-15, -10:1:10, 15:5:180
$\beta$ (deg)	0
$h$ (m)	30000

Table 6.2: Flight conditions for Missile DATCOM runs.

#### Generating Script File

In order to generate the aerodynamic database, a script file for each unique RLV solution must be created. Meta-programming is used by calling missile DATCOM within a MATLAB script and passing the inputs using the fprintf command. The overarching procedure is shown in fig. 6.1. LAFT and DAFT are set 0 which are the boattail or flare length and diameter. Since only the return flight for the first stage will be required for missile DATCOM, these parameters are not further discussed here. The TRUNC=.FALSE command corresponds to the type of input geometry that DATCOM receives. With this option, DATCOM must receive values of longitudinal and radial distance from the respective axis of the vehicle. Nevertheless, as mentioned, as the first stage's shape very nearly represents a cylinder, the calculation of these coordinates is trivial and depends entirely on radius and length of the first stage.

#### Running Missile DATCOM

Each DATCOM run has a limited number of inputs that it can handle, and therefore multiple scripts are created to cover the entire range of flight conditions. Once a case is finished running, the output file generated is opened and the aerodynamic coefficients are extracted from the derivatives of the output file. The procedure for this has been written and verified by Rozenmeijer [50], and is the same used here. Once all the conditions for the selected RLV configurations is run, the data is stored in a cell structure as a MATLAB file and loaded onto the simulator when needed.

#### Interpolation of Results

When calling the aerodynamic database, the values of  $C_D$  and  $C_L$  are interpolated based on the flight conditions of  $M$  and  $\alpha$ . MATLAB's interp2 function is used for two-dimensional interpolation. A value of 100 is

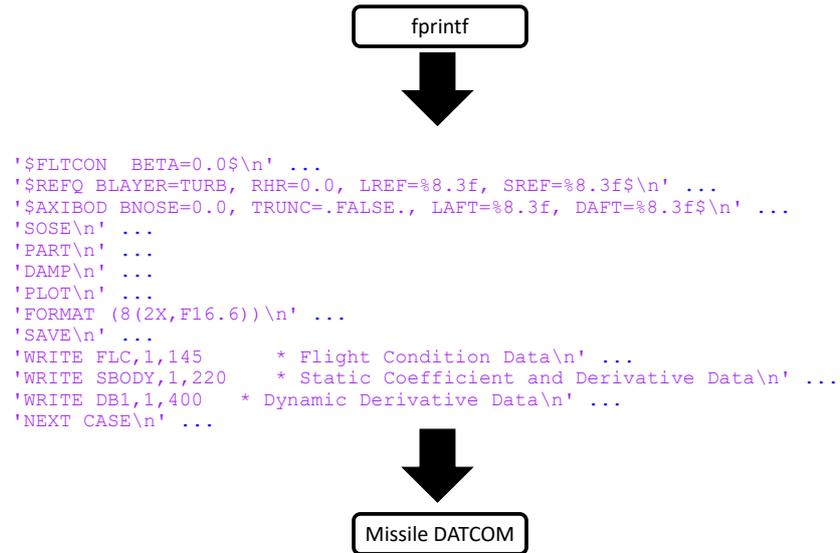


Figure 6.1: Missile DATCOM procedure and code snippet for passing inputs through fprintf.

appended to the  $M$  vector used for interp2, in the case that there is a state for which the Mach number is greater than 5.

# Chapter 7

## Preliminary Ascent and Descent

This section deals with the preliminary phase to calculate nominal launch vehicle descent starting conditions. This is done by the usage of analytical methods to calculate the optimum ascent trajectory by evaluating metrics of performance, and using analytical descent methods to determine the preliminary feasibility by varying the final time and candidate ascent trajectory solutions. The chapter is structured by first introducing the methods for evaluating the ascent trajectory of the launch vehicle, and then the method to evaluate a feasible descent trajectory. These results of these approaches will carry on over to the closed-loop RTLS simulations as a starting point for selection of the range of parameters to use in the DoE's for the guidance algorithm.

### 7.1 Preliminary Ascent

The scope of the project for evaluating ascent performance is not concerned with the holistic feasibility of the RLV for reaching the target. We interpret for this section of the study as having the second stack of the RLV reach 600km with a threshold of -1 to 1 km. The purpose of this is two-fold; one to avoid the solver used to exclude solutions that are slightly large or smaller than the intended target due to some round-off or truncation error, and secondly because it is taken as a value of sensible target payload insertion altitude error. For the latter because the mass of the propellant is a variable that is both a function of the ascent and descent, the optimization approach will be to set it as an objective to maximize.

The optimization method used for evaluating the ascent trajectory is entirely dependent on obtaining a value for propellant available for descent and ensuring that the target orbit is achievable. As such, the scope of the project concerns itself with using simple, analytical approaches to find the solution of such a problem.

To explain this optimization approach, the following sections break down the problem by first explaining the different characteristic quantities assigned to the multi-stage rocket, then the original lagrange approach as introduced in the reader is described, and finally the form used for this study is introduced.

#### 7.1.1 Multi-stage Rocket

The multi-stage rocket is categorized in different stages and sections. The latter term has already been used previously in this document to refer to the vehicle stack, and thus will be used interchangeably. For the  $i^{th}$  rocket section the characteristic quantities of concern are the payload ratio  $\lambda_i$ , the construction mass ratio  $\epsilon_i$ , and the mass ratio  $\Lambda_i$ . These terms are all formulated in equations 7.1, 7.2, and 7.3, respectively. The propellant mass ratio  $\phi_i$  can be used to express  $\Lambda_i$  and is given in equation 7.4.

$$\lambda_i = \frac{m_{u_i}}{m_{0_i}} = \frac{m_{c_{i+1}}}{m_{0_i}} \quad (7.1)$$

$$\epsilon_i = \frac{m_{c_i}}{m_{c_i} + m_{p_i}} \quad (7.2)$$

$$\Lambda_i = \frac{m_{0_i}}{m_{0_i} - m_{p_i}} = \frac{1}{1 - \phi_i} \quad (7.3)$$

$$\phi_i = \frac{m_{p_i}}{m_{0_i}} \quad (7.4)$$

Only  $\lambda_i$ ,  $\Lambda_i$ , and  $\phi_i$  refer to the rocket section, whilst  $\epsilon_i$  refers to the rocket stage. The variable  $m$  refers to the mass and the subscripts  $c$ ,  $p$ ,  $0$ , and  $u$  refer to the construction, propellant, initial, and useful mass respectively. The useful mass is intended as the total vehicle payload mass. The ideal end velocity can also be obtained from all the rocket sections by equation 7.5, where  $c_{eff}$  is the effective exhaust velocity and along with  $\epsilon_i$  are known constants.

$$v_{e_N} = \sum_{i=1}^N (c_{eff_i} \ln \Lambda_i - \sum_{i=1}^N (c_{eff_i} \ln (\lambda_i(1 - \epsilon_i) + \epsilon_i)) = f(\lambda_i) \quad (7.5)$$

In order to maximize equation 7.5 we introduce a subsidiary requirement by acknowledging that the total payload ratio  $\lambda_{tot}$  is equal to the product of all payload ratios as shown in equation 7.6.

$$\lambda_{tot} = \prod \lambda_i \quad (7.6)$$

We rewrite this requirement by taking the difference of the two terms as shown in equation 7.7.

$$g(\lambda_i) = \sum_{i=1}^N \ln \lambda_i - \ln \lambda_{tot} = 0 \quad (7.7)$$

This can then be converted into a lagrange multiplier problem where we wish to find the optimal value of two functions that are tangent to one another at a certain optima. This is shown in equation 7.8.

$$F(\lambda_i) = f(\lambda_i) + \mu g(\lambda_i) \quad (7.8)$$

In order to find the optima, we wish to maximize  $f(\lambda_i)$  and that involves taking the derivative as shown in equation 7.9.

$$\frac{\partial F(\lambda_i)}{\partial \lambda_i} = \frac{\partial f(\lambda_i)}{\partial \lambda_i} + \mu \frac{\partial g(\lambda_i)}{\partial \lambda_i} = 0 \quad (7.9)$$

Plugging equation 7.7 and 7.8 into this expression gives the equation for each  $\lambda_i$ , as shown in equation 7.10, which can also be reformulated as shown in equation 7.11.

$$-(c_{eff})_i \frac{1 - \epsilon_i}{\lambda_i(1 - \epsilon_i) + \epsilon_i} + \mu \frac{1}{\lambda_i} = 0 \quad (7.10)$$

$$\lambda_i = \frac{\mu \epsilon_i}{c_{eff,i}} - \mu(1 - \epsilon_i) \quad i = 1, 2, \dots, N \quad (7.11)$$

This equation gives the optimal value of  $\lambda_i$  given a value of  $\mu$ . Additionally, by knowledge of the fact that the sum of the products of  $\lambda_i$  gives  $\lambda_{tot}$ , equation 7.12 can be used to find the optimal lagrange multiplier.

$$\prod_{i=1}^N \lambda_i = \prod_{i=1}^N \left( \frac{\mu \epsilon_i}{c_{eff,i}} - \mu(1 - \epsilon_i) \right) = \lambda_{tot} \quad (7.12)$$

Equation 7.12 can then be expanded to show the general equation used for calculating  $\lambda_{tot}$  by replacing all the indices  $i$  with their appropriate stack number and taking the sum of the products. This is shown in equation 7.13.

$$\lambda_{tot} = \frac{-\epsilon_1 \epsilon_2 \mu^2}{(\epsilon_1 + \epsilon_2 - \epsilon_1 \epsilon_2 - 1) \mu^2 + (c_1 + c_2 - \epsilon_1 c_1 - c_1 \epsilon_2 - c_2 \epsilon_1 - c_2 \epsilon_2 + c_1 \epsilon_1 \epsilon_2 + c_2 \epsilon_1 \epsilon_2) \mu - c_1 c_2 + c_1 c_2 \epsilon_1 + c_1 c_2 \epsilon_2 - c_1 c_2 \epsilon_1 \epsilon_2} \quad (7.13)$$

Finally, by evaluating expression 7.13 and collecting like terms, we arrive at the equation that can be used to solve for  $\mu$ , which is a quadratic, as shown in eq. 7.14.

$$\lambda_{tot} = (\epsilon_1 \epsilon_2 \mu^2) / ((c_{eff,1} - \mu)(c_{eff,2} - \mu)(\epsilon_1 - 1)(\epsilon_2 - 1)) \quad (7.14)$$

This results in two possible values and the selection for the correct value is given by virtue of the fact that the payload ratio needs to remain within a sensible value of magnitude, and that it must be positive.

Once the optimum propellant mass of the vehicle is determined, the equations that govern the position and thus the propellant burnt to reach orbit in the ascent phase are accounted for.

### 7.1.2 Culmination Altitude

For a multi-stage rocket, the  $(i+1)^{th}$  rocket section can be considered to burn directly after the ejection of the  $i^{th}$  stage. However, it is also possible to introduce a coasting arc between the ignition of two stages. In two different cases, the influence of a coasting arc can either increase the culmination altitude or decrease it. The culmination altitude  $h_c$ , is defined at the point where the vertical velocity of the vehicle is zero. In the case of no aerodynamic drag, the relationship between  $h_c$  and coasting time  $t_{co}$  is indirectly proportional and linear. With aerodynamic drag an optimal  $t_{co}$  can be found for culmination altitude. This  $h_{co}$  will nevertheless be lower than the one without vacuum at any  $t_{co}$ . For this study, it is decided that aerodynamic drag is not included in the evaluation of  $h_{co}$  for two different purposes. First of all,  $h_{co}$  corresponds to the starting altitude for descent. It is considered worthwhile to make this over-prediction as it accounts for the overall distance of the vehicle to the LP. It is therefore advantageous to set up a nominal trajectory for the closed-loop simulation where the distance covered is greater than expected. Secondly, as the method is analytical, it may be highly inaccurate to account for drag using a term that does not require some level of numerical iteration. The drag force is a non-conservative and therefore depends on the altitude variation against time which may be non-linear. Estimation as a linear function may cause an under-estimate of  $h_{co}$  resulting in a less than nominal candidate for benchmark selection of starting conditions.

For a two stage vehicle, the culmination altitude can be found given the equation shown in 7.15. The changes in altitude from each stage ( $\Delta h_1, \Delta h_2$ ).

$$h_c = \Delta h_1 + \Delta h_2 + \Delta V_1 t_{b,2} \frac{(\Delta V_1 + \Delta V_2)^2}{2g_0} \quad (7.15)$$

To determine the different  $\Delta V$  for each stage we can use equation 7.16 for vertical flight in a constant gravity field and vacuum.

$$(\Delta v_e)_i = -(c_{eff,i}) \ln \lambda_i (1 - \varepsilon_i) + \varepsilon_i + \varepsilon_i - g_0 t_{b,i} \quad (7.16)$$

### 7.1.3 Final Optimization Routine

There is a fundamental difference between the objective of the original  $h_c$  problem included in the reader and that for this study. Recall that for ascent the objective is to save propellant mass and reach a target  $h_c$  within a certain threshold. Therefore, the original lagrange problem does not conserve itself only with finding an optimal  $\mu$ , but also maximizing a variable. As a result, the multi-stage rocket problem needs to be altered to include a new variable,  $m_{saved}$  which is the propellant mass that is not burnt in stage 1 for ascent operations. Furthermore, it is no longer valid to exclude  $t_{co}$  as the objective is not to maximize or even reach a set  $h_c$ , but it is to do it with as little propellant as possible. As a result,  $t_{co}$  is predicted no longer to be zero.

For this reason, the equations to solve for  $h_{co}$  and  $\mu$  are recomposed with  $t_{co}$  and  $m_{saved}$  in mind. For  $m_{saved}$ , it simply becomes a matter of substituting  $m_{p,1}$  in the original equations with  $m_{p,1} + m_{saved}$ .

The new  $h_c$  is given in equation 7.17 introducing the relevant gravity losses to the vehicle during the culmination arc.

$$h_c = \Delta h_1 + \Delta V_1 t_{co} - \frac{1}{2} g_0 t_{co}^2 + \Delta h_2 + (\Delta V_1 - g_0 t_{co}) t_{b,2} + \frac{(\Delta V_1 + \Delta V_2 - g_0 t_{co})^2}{2g_0} \quad (7.17)$$

We note that as shown in equation 7.18  $c_{eff,i}$  is a function of  $I_{sp}$  and is divided by a factor of 1000 to account for the change of units in km (which is what the reader used in the original equations). We rewrite the payload ratio with their respective indices as shown in equations 7.19, and 7.20 for the first and second stack of the vehicle. Note the inclusion of the term  $m_{saved}$  in equation 7.19.

$$c_{eff,i} = \frac{I_{sp,i} g_0}{1000} \quad (7.18)$$

$$\varepsilon_1 = \frac{m_{c,1} + m_{saved}}{m_{c,1} + m_{p,1}} \quad (7.19)$$

$$\varepsilon_2 = \frac{m_{c,2}}{m_{c,2} + m_{p,2}} \quad (7.20)$$

The optimization problem can be formulated as shown in equation 7.21. The two variables are  $t_{co}$  and  $m_{saved}$ , where the objective is set as  $-m_{saved}$ , as *fmincon* interprets the objective to be minimized.

$$\begin{aligned}
& \min(m_{saved}, t_{co}) - m_{saved} \\
\text{s.t. } h_{culm} = & -I_{sp,1}g_0 \log \frac{\lambda_1}{\lambda_2} \left(1 - \frac{m_{c,1} + m_{saved}}{m_{c,1} + m_{p,1}}\right) + \frac{m_{c,1} + m_{saved}}{m_{c,1} + m_{p,1}} \frac{m_{p,1} + m_{saved}}{\frac{F_{T,vac,1}}{I_{sp,1}g_0}} + \\
& \Delta V_1 t_{co} - \frac{1}{2}g_0(t_{co}^2) + \\
& \left(-I_{sp,2}g_0 \log \lambda_2 \left(1 - \frac{m_{c,2}}{m_{c,2} + m_{p,2}}\right) + \frac{m_{c,2}}{m_{c,2} + m_{p,2}} - \right. \\
& \left. g_0 \frac{m_{p,2}}{\frac{F_{T,vac,2}}{I_{sp,2}g_0}} \frac{m_{p,2}}{\frac{F_{T,vac,2}}{I_{sp,2}g_0}} + \right. \\
& \left. (\Delta V_1 - g_0 t_{co} \frac{m_{p,1} + m_{saved}}{\frac{F_{T,vac,1}}{I_{sp,1}g_0}} + \frac{(\Delta V_1 + \Delta V_2 - g_0 t_{co})^2}{2g_0})\right) \\
\Delta V_1 = & ((I_{sp,1}g_0) \log \lambda_1 (1 - \varepsilon_1) + \varepsilon_1 - g_0 t_{b,1}) \sin \theta_{pitch} \\
\Delta V_2 = & ((I_{sp,2}g_0) \log \lambda_2 (1 - \varepsilon_2) + \varepsilon_2 - g_0 t_{b,2}) \sin \theta_{pitch} \tag{7.21} \\
\lambda_1 = & \frac{\mu \frac{m_{c,1} + m_{saved}}{m_{c,1} + m_{p,1}}}{\left(\frac{I_{sp,1}g_0}{1000} - \mu\right) \left(1 - \frac{m_{c,1} + m_{saved}}{m_{c,1} + m_{p,1}}\right)} \\
\lambda_2 = & \frac{\mu \frac{m_{c,2}}{m_{c,2}} + m_{p,2}}{\left(\frac{I_{sp,2}g_0}{1000} - \mu\right) \left(1 - \frac{m_{c,2}}{m_{c,2} + m_{p,2}}\right)} \\
t_{b,1} = & \frac{m_{p,1} - m_{saved}}{\frac{F_{T,vac,1}}{I_{sp,1}g_0}} \\
t_{b,2} = & \frac{m_{p,2}}{\frac{F_{T,vac,2}}{I_{sp,2}g_0}} \\
t_{co} \geq & \mathbf{0} \\
599000 \leq & h_{culm} \leq 601000
\end{aligned}$$

## 7.2 Preliminary Descent

The preliminary descent procedure follows from the results of the preliminary ascent by using the nominal starting conditions for descent to determine the feasibility of operations by varying time to reach the LP. This is done as the guidance algorithm to be implemented is a finite horizon problem, where the final time must be an input and not a result of the optimization. Through the preliminary ascent procedure, multiple starting points are obtained also by virtue of the fact that the culmination altitude is bounded inequality on both sides rather than an equality constraint. By varying final time and using different conditions for ascent, a trade off can be made using the results of the descent to further refine the nominal trajectory used. The preliminary descent, like the ascent procedure, also considers a reduced complexity of the problem for the purpose that it is intended to provide a starting point to where the DoE's efforts to optimize the guidance controller's should be spent. However, unlike preliminary ascent, whilst the guidance law is analytical, the trajectory is numerically propagated in time, yet without aerodynamic drag. Aerodynamic drag is also excluded here as the procedure is meant to generate a fast trade-off analysis and due to the fact that the guidance laws do not take into account the drag acceleration of the vehicle. It is worthwhile to mention that the same approach could be done by an open-loop fashion using the convex guidance algorithm as the optimization problem rather than using analytical guidance laws. However, the size of the discretized grid of the problem would have to be exceptionally large.

In the following sections the guidance law used for this procedure is explained, followed by the closed-loop simulation set up. Next, the approach to generate trade off contour maps is explained along with the method to narrow down nominal trajectories for use in the closed-loop simulations is explained.

### 7.2.1 Guidance Law

The guidance law used in this study is the ZEM/ZEV (zero-error miss and zero-error velocity). The ZEM/ZEV law falls under the class of optimal feedback guidance algorithms as described by [53]. This approach was also used in [20] to generate a trade-off space rapidly relative to the time for higher fidelity simulations to evaluate. In short, optimal feedback guidance laws allow definition of the final velocity, which in this case is set to 0. To arrive at the definition of the ZEM/ZEV formulation, we first start with the constrained terminal velocity guidance law (CTVG).

#### Constrained-Terminal Velocity Guidance (CTVG)

We define a cost function for minimizing the integral of the square of the acceleration given by Eq. 7.22. To derive a law that optimizes the equation the trajectory, whilst keeping a closed-form solution, a Hamiltonian function is used. The Hamiltonian is the sum of the kinetic and potential energy in a system. For this problem it is defined by Eq. 7.23.

$$J = \frac{1}{2} \int_{t_0}^{t_f} \mathbf{a}^T \mathbf{a} dt \quad (7.22)$$

$$\mathbf{H} = \frac{1}{2} \mathbf{a}^T \mathbf{a} + \mathbf{p}_r^T \mathbf{v} + \mathbf{p}_v^T (\mathbf{g} + \mathbf{a}) \quad (7.23)$$

The symbols  $\mathbf{p}$  denote the co-states of the problem, also known as the auxiliary or ad-joint. Taking the negative result of the partial derivative of the Hamiltonian with respect to the states, the co-states can be found. Nulling the Hamiltonian's derivative with respect to the acceleration minimizes the cost function as stated by the Pontryagin maximum principle [53]. This is done in equation 7.24.

$$\begin{aligned} \mathbf{p}_r \dot{}(t) &= -\frac{\partial \mathbf{H}}{\partial \mathbf{r}(t)} = 0 \\ \mathbf{p}_v \dot{}(t) &= -\frac{\partial \mathbf{H}}{\partial \mathbf{v}(t)} = -\mathbf{p}_r(t) \\ \frac{\partial \mathbf{H}}{\partial \mathbf{a}(t)} &= 0 \end{aligned} \quad (7.24)$$

The last line of this equation can be substituted by the integrated terms of the co-state vectors given in equation 7.25 to give the optimal control solution in Eq. 7.28, and the states (Eq. 7.26 & 7.27). We rewrite these new state functions as a function of their respective co-states as shown in 7.28, and then substitute the solution in 7.25 to obtain Eq. 7.29 which is the constrained terminal velocity guidance (CTVG) law [26][53].

$$\begin{aligned} \mathbf{p}_r(t) &= \mathbf{p}_r(t_f) \\ \mathbf{p}_v(t) &= t_{go} \mathbf{p}_r(t_f) + \mathbf{p}_v(t_f) \end{aligned} \quad (7.25)$$

$$\mathbf{v}(t) = \frac{t_{go}^2}{2} \mathbf{p}_r(t_f) + t_{go} \mathbf{p}_v(t_f) - t_{go} \mathbf{g} + \mathbf{v}_f(t) \quad (7.26)$$

$$\mathbf{r} = -\frac{t_{go}^3}{6} \mathbf{p}_r(t_f) - \frac{t_{go}^2}{2} \mathbf{p}_v(t_f) + \frac{t_{go}^2}{2} \mathbf{g} - t_{go} \mathbf{v}_f + \mathbf{r}_f \quad (7.27)$$

$$\begin{aligned} \mathbf{p}_r(t_f) &= \frac{6(\mathbf{v} - \mathbf{v}_f)}{t_{go}^2} + \frac{12(\mathbf{r} - \mathbf{r}_f)}{t_{go}^3} \\ \mathbf{p}_v(t_f) &= -\frac{2(\mathbf{v} + 2\mathbf{v}_f)}{t_{go}} - \frac{6(\mathbf{r} - \mathbf{r}_f)}{t_{go}^2} + \mathbf{g} \end{aligned} \quad (7.28)$$

$$\mathbf{a} = \frac{6(\mathbf{r}_f - (\mathbf{r}(t) + t_{go} \mathbf{v}))}{t_{go}^2} + \frac{4(\mathbf{v}_f - \mathbf{v}(t))}{t_{go}} - \mathbf{g} \quad (7.29)$$

### Zero Effort Miss (ZEM) & Zero Effort Velocity (ZEV)

ZEM & ZEV are useful concepts when trying to understand how guidance laws work. ZEM distance is defined as the separation between the target and missile at the end of flight, without any further accelerations. This of course would mean that the trajectories would be straight lines. Eq. 7.30 shows how the ZEM distance is calculated, with  $t_{go}$  being the time to go.

$$\mathbf{ZEM} = \mathbf{r} + \mathbf{v}t_{go} \quad (7.30)$$

Both concepts can also simply be expressed as the difference between the final state (position for ZEM, velocity for ZEV) as given by Eq. 7.32 & 7.31. Note that for this formulation, we have also included the effect of gravity.

$$\mathbf{ZEV}(t) = \mathbf{v}_f - [\mathbf{v}(t) + \int_{t_f}^{t_0} \mathbf{g}(\tau) d\tau] \quad (7.31)$$

$$\mathbf{ZEM}(t) = \mathbf{r}_f - [\mathbf{v}(t)t_{go} + \int_{t_f}^{t_0} (t_f - \tau)\mathbf{g}(\tau) d\tau] \quad (7.32)$$

ZEM and ZEV by themselves do not automatically form a guidance law. However, as detailed by [26], a derivation of a new guidance law may exist by using these two new concepts. In fact, the CTVG, FTVG, and PNG law can be reformulated using this concept as shown in Eq. 7.33, 7.34, and 7.35 respectively.

$$\mathbf{a}(t) = \frac{6}{t_{go}^2(t)} \mathbf{ZEM}(t) - \frac{2}{t_{go}(t)} \mathbf{ZEV}(t) \quad (7.33)$$

$$\mathbf{a}(t) = \frac{3}{t_{go}^2(t)} \mathbf{ZEM}(t) \quad (7.34)$$

$$\mathbf{a}(t) = \frac{n}{t_{go}^2(t)} \mathbf{ZEM}(t) \quad (7.35)$$

The first two terms of the final form are the most useful result after all substitutions. The integers 6 and -2 that appear in front of the two terms in Eq. 7.33 are actually the optimal gains for what is more generally known as terminal velocity guidance. The control vector for the acceleration for this general guidance law is given by Eq. 7.29 ( $K_1 = 6$  and  $K_2 = -2$  for the optimal solution which is also known as CTVG) [53].

$$\mathbf{a}(t) = K_1 \frac{\mathbf{ZEM}}{t_{go}^2} + K_2 \frac{\mathbf{ZEV}}{t_{go}} \quad (7.36)$$

### 7.2.2 Closed-loop Simulation

The closed-loop simulation involves the descent guidance, and SDK (spacecraft dynamics and kinematics) blocks only. The problem formulation is named as RTL1, that includes a varying gravity field, in a Flat Earth environment, with 3-DoF. Moreover mass propellant dynamics are turned off to give a conservative metric of the available maximum thrust acceleration to the vehicle during descent. The maximum available thrust actually increases as more propellant is burnt but is not included here for the following reasons. The analytical ZEM/ZEV guidance law does not follow a discrete burn schedule but rather, from previous results shown by [53], starts with a high thrust which decays and then rises back up again. In fact, given that the formulation of the analytical guidance law uses the integration of the acceleration as a cost function in its derivation, it is intuitive to see how the cost function will have a local minima and never reach a discrete value. Therefore, for this limitation, we turn off propellant depletion dynamics to counteract this discrepancy and overestimation of the performance of the vehicle compared to the convex algorithm. Additionally the rotation of the Earth is accounted for by account for the distance covered over the selected final time, as well as the burn time of the first stage during ascent. This is shown in equation 7.37.

$$X_{eD,F} = X_{eA,F} \delta t_{b,1} + \omega R_e t_f \quad (7.37)$$

The closed loop simulation parameters for both SDK and DG block are shown in table 9.2. The one-to-one ratio of  $f_{sim}$  to  $f_{gui}$  is done so as not to account for update delays and that the best possible trajectory is achieved as the main priority is assessing the feasibility of RTL1 in a reduced fidelity and not that of the guidance law. The integration scheme used is RK4 (runge-kutta 4).

### 7.2.3 Trade-off Space

Each trajectory starting point for descent will be varied based on the preliminary ascent solutions and a final time for reaching the LP when considering  $t_0$  as the beginning of descent. The trade-off contour maps aim to reflect the three different primary metrics for evaluating feasibility of RTLS operations. These are velocity touchdown norm  $\|\mathbf{v}_f\|$  in m/s, required propellant mass for burn  $m_{burn}$  in kg, and maximum dynamic pressure  $Q_{dyn,max}$  in kPa. These three metrics can provide information on the feasibility of different starting points, hence, ascent trajectories, as well as  $t_f$  to touchdown. An important note must be made on the position touchdown norm  $\|\mathbf{r}_f\|$ , which is excluded as the position of the LP with the ZEM/ZEV law was noted to always remain within the optimal bounds considered for landing, and as such is not included here.  $\|\mathbf{v}_f\|$  can provide information on the control strictness required for landing. An intuitive understanding as to why this metric is more sensitive in this preliminary analysis may be due to the fact that velocity variations as a percentage of the maximum velocity the guidance law computes over the trajectory are greater and more prevalent than position variations. This can be seen in equation 7.26 as  $\mathbf{v}(t)$  is dependent on the factor of  $t_{go}$ , and as such when the final time is reached, the final velocity may not have reached target yet. Instead,  $\mathbf{r}(t)$  is directly related to  $\mathbf{a}(t)$ , and therefore the thrust acceleration is still prevalent.



# Chapter 8

## Closed-Loop Simulations

The main method that the feasibility for RTLS for the VEGA vehicle designed for this study will be assessed on, is its performance in a 3-DoF simulator. This simulator architecture has been written entirely in MATLAB. Large scale DoEs (Design of Experiment) are run to examine three different criteria for feasibility: physical feasibility, model accuracy, and real-time performance. Separate DoE's are run to focus on varying certain aspects of the RTLS strategy mainly through variation of the guidance algorithm. Whilst each DoE focuses on a particular aspect of the mission, the results may give insights that relate to more than one of the criteria. The rest of this chapter will begin by describing the architecture of the software used to run the closed-loop simulations. The closed-loop simulator will be broken down into each of the high-level components and explained in detail. Next, the parameters chosen for the DoE's and their motivation is given, along with expectation of results and hypothesis. Within the section of DoE's campaign will also include the method for sensitivity analyses, considered as the 4th main DoE in this study. Finally, verification that the closed-loop simulator works as intended will be evidenced.

### 8.1 Architecture

The architecture of the simulator, whilst programmed completely in MATLAB code, is built in mind representing the block diagram structure shown in fig. 8.1. The simulator can be intuitively broken down into separate blocks that are responsible for different procedures in the closed-loop simulations. The architecture aims to replicate the influence of errors on the control  $\mathbf{u}$  and state  $\mathbf{x}$  from the data measured by sensors of the RL. The simulation begins by first calculating the descent guidance (DG) which takes in inputs of real and measured states  $\mathbf{x}_{real}$  and  $\mathbf{x}_{meas}$ . The actuator also receives this information, and additionally the required control, which is the real (expected) control  $\mathbf{u}_{real}$ , which in this case is the magnitude and thrust direction of the engines. Next,  $\mathbf{u}_{meas}$  and  $\mathbf{x}_{real}$  is fed to the spacecraft dynamics & kinematics block (SDK), which then converts  $\mathbf{x}_{real}$  into  $\mathbf{x}_{meas}$ . The closed-loop system is completed by passing the new  $\mathbf{x}_{meas}$  into the next step of the guidance algorithm at the next guidance step.

#### 8.1.1 DG (Descent Guidance)

The descent guidance block incorporates the procedure of determining whether or not the current simulation step coincides with a guidance step. The simulator runs at a fixed guidance and simulation step size, and it is the simulation step size that drives the process ( $f_{sim} \geq f_{gui}$ ). The only variable influencing the descent guidance step size other than  $f_{gui}$  is the process time for calculating the guidance solution. This includes the entire guidance solution from  $i_{cvx} = 1$  to  $N_{cvx} + 1$  as  $N_{cvx}$  is the number of successive convexifications, and apart from those SOCP1 is always invoked. The function containing the controller evaluation is entitled DESCO-Descent Envelope with Successive Convex Optimization. This name references loosely the DESCENDO (descending over successive envelopes using successive convexification-based optimization) controller, albeit with a different acronym to distinguish them being two different controllers but essentially executing the same function, with similar if not identical architectures (apart from being written by two different users). After running DESCO, the process time is saved. At the first guidance step, thus at the beginning of the simulation, the trajectory gets a free guidance solution. This solution will also set the initial thrust acceleration of the engine. The point at which the next guidance step is computed is dependent upon three things: the

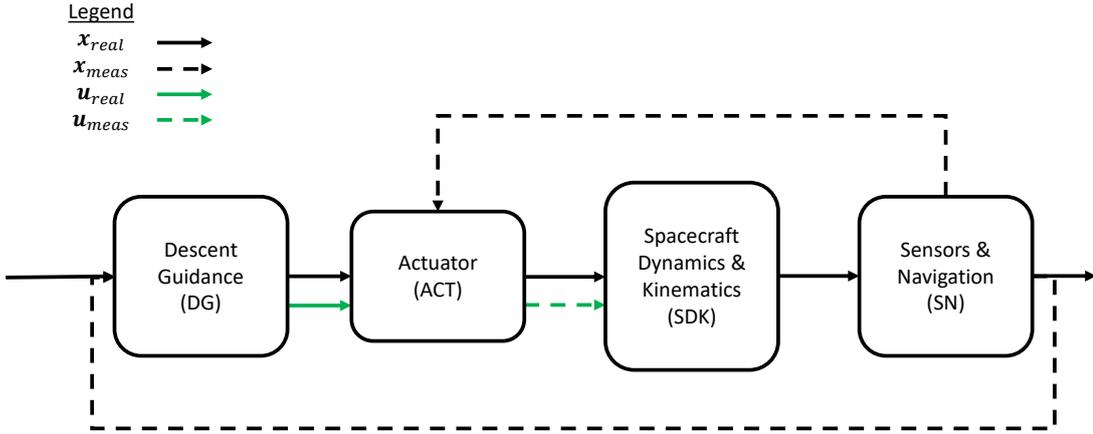


Figure 8.1: Simulator block diagram representation

guidance frequency delay  $\frac{1}{f_{sim}}$ , the process time delay  $t_{pt}$ , and the simulation frequency delay. The latter is the effect that is caused by the remaining time from the start of the last guidance step  $t_{gui, i_{gui}}$  with the added frequency and process time delay, and the time until the next simulation step  $t_{sim, i_{sim}+1}$ . The subscript  $i_{sim}$  and  $i_{gui}$  refer to the simulation time and guidance time index respectively. Therefore, unless  $\frac{f_{sim}}{f_{gui}}$  is a whole number and that there is no  $t_{pt}$ , there will always be simulation frequency delay.

The guidance step determination section of the block also saves a flag `guidance_step` to keep track of whether a guidance solution has been recently calculated or not. This step is then fed to the SN block to determine whether a bias and error has to be applied to the guidance time. The analogy purpose of doing so is that the virtual guidance controller aims to replicate the functionalities of real-time GNC hardware. Every time the guidance clock measures the time at which the solution is calculated, there may be some delay due to bias  $\epsilon_t$  and (random) noise  $\eta_t$ . Note that the delays mentioned before are not errors and they are always present even in the case of a perfect controller.

Once a guidance solution has been evaluated, it is stored and kept until the simulation time  $t_{sim}$  is greater than all the time delays and errors. This is where another flag `guidance_read` registers whether a guidance solution has been recently updated or not. All of the information from the guidance solution obtained from DESCOCO is stored as a cell structure named  $U_{ref}$ . This cell structure replaces  $U_{ref,0}$ , which is the structure that is used in the actuator block, when the flag is 0 and when the combined guidance delay and error time  $\Delta t_{gui}$  and  $t_{gui}$  is less than  $t_{sim}$ . If it is the case, `guidance_read` is set 1 or else 0. The purpose of this flag is to avoid re-updating  $U_{ref}$  when a new guidance solution has yet not been calculated. After updating the solution, a diagnostics check is passed. This check is done at  $i_{sim} = 1$  and it is meant to determine the overall feasibility of the simulation case. If DESCOCO fails to calculate the SOCP1 solution at  $t_{sim} = t_{gui} = 0$ , then the `sim_break` flag is set to 1. This will be used to evaluate the simulation termination criteria after the SN block.

### 8.1.2 ACT (Actuators)

The inputs from the DG block used to calculate the reference acceleration come from the cell structure  $U_{ref}$ . These are the reference discretized time grid  $t_{ref}[k]$ , and thrust acceleration  $\mathbf{w}[k]$ . In order to calculate the reference acceleration, the  $t_{ref}[k]$  is interpolated with  $\mathbf{w}[k]$  with MATLAB's `interp1`. A reference thrust acceleration to achieve  $a_{ref}$  is obtained which is then passed on to a function that calculates the throttling %'s of the M10 %  $T_{M10}$  and MX %  $T_{MX}$  engines.

Following this, a check is made to determine that there is propellant ( $m_{meas} \geq m_{dry}$ ). If not, then the throt-

thrusting %'s are set 0. Otherwise, depending on the value of  $\|a_{ref}\|$  the thrust to achieve  $T_{tvc}$  is split up amongst the 2 types (3 in total) engines. The actuator strategy here aims to match, whenever possible,  $I_{sp,oa}$  that is used in DESCO to determine the mass flow rate in the SOCP problems. As such, the first if statement in this routine check whether  $T_{tvc}$  is greater than the overall combined minimum thrust of all engines. If so, then an `fmincon` sub-routine is called.

If the overall combined minimum thrust of the engines is greater than  $T_{tvc}$ , the thrust acceleration is then checked to match the lowest possible output thrust by the RLV, which is  $T_{M10,min}$ . If the check is valid, and  $T_{tvc}$  if it is not greater than  $2T_{MX,min}$ , the MX engines are used, otherwise, it resorts to the M10. Otherwise,  $a_{ref}$  is too low and the engines are not turned off or switched off. The reason a threshold is not applied is because in testing it was determined that if ever  $a_{ref}$  was lower than  $T_{M10,min}$ , then it was because of residual values from the SOCP solver. This is true in cases where there is a thrust-on-off-on or off-on sequence where certain constraints are turned off in the problem, but ECOS may return very small values close to 0. It may also be true when a free burn schedule ( $q[k] = 1 \forall k$ ) is applied. In preliminary testing, there was no incidence where this caused  $a_{ref}$  values close to but below  $T_{M10,min}$  to be set to 0 in the actuator block. Moreover, second priority is given to firing the M10 engine as for the nominal engine configuration the loss of thrust compared to the overall thrust achievable by the M10 at sea level pressure is much greater than that of the MX.

If the overall combined minimum thrust of the engines does not exceed  $T_{tvc}$  then a `fmincon` sub-routine is called "TVC Opt.". This sub-routine contains checks that were done in the higher level routine explained before. However, instead of determining whether  $T_{tvc}$  is in the bounds of the minimum thrusts of the various possible combinations of engine throttling, it checks whether the exit pressures  $P_e$  can compensate for the ambient pressure  $P_a$ . As with before, M10 is the first priority due to its lower potential at lower altitudes. If M10 can provide a positive overall thrust relative to the ideal thrust equation, then the following can occur. If  $P_{e,MX} \geq P_a$  and  $P_{e,M10} \geq P_a$ , meaning the exit pressures of all engines are greater than ambient, all engines are used. The rest of the statements, visible in figure B.1 in appendix B.1.2, adapt the equality constraints of the problem based on the relationship of  $P_e$  with respect to  $P_a$ .

### 8.1.3 SDK (Spacecraft Dynamics & Kinematics)

The spacecraft dynamics & kinematics block is associating with the real state values and simulation of the environment of the spacecraft. The pseudo-code for this block is shown in fig. B.4. The environment is built from the equations of motion, listed in equations 4.8 to 4.13. The SDK block also contains information about the integrator used, in this case RK4, and also propagates  $t_{sim}$  to the next simulation index. Moreover, once the real state is propagated to the next simulation index, the block also calculates auxiliary data which is included in a vector containing  $\alpha_{eff}$ ,  $Q_{dyn}$ ,  $M$ ,  $C_d$ ,  $C_l$ ,  $\rho$ ,  $v_{rel,x}$ ,  $v_{rel,y}$ ,  $v_{rel,z}$ ,  $P_a$ . This information can be used to evaluate trajectories once completed. The purpose of saving these variable is that they can be helpful in debugging when NaN's or Inf values are reached. Looking at the state history from the solution output cannot intuitively explain why NaN's occur if they do. Rather, looking at the data that is used to propagate the states, such as thrust acceleration data and aerodynamic data, can give insight into whether the models used have been programmed incorrectly.

### 8.1.4 SN (Sensors & Navigation)

In the case of a perfect system with no delays, real state  $x_{real}$  is equivalent to the  $x_{meas}$  state. The pseudo-code for this block is shown in fig. B.5. For each block, either one of these vectors are fed. The DG and ACT block reads  $x_{meas}$ , and SDK & SN read  $x_{real}$ . It should be made clear that the ACT block serves as a virtual actuator controller, as the pressure model is the same that is used for DG. SDK then takes the reference  $\%T_{M10}$  and  $\%T_{MX}$  and determines whether that is achievable based on the real environment  $P_a$ . The reason that ACT is separated from  $x_{real}$  is so that as the SDK block already calculated the atmospheric properties ( $P_a$ ,  $\rho$ ,  $A^*$ ), it avoids having to call the atmospheric function twice within one simulation index. This call in fact is one of the most expensive routines apart from the DESCO function and as such was deemed sensible to not include this in ACT. This is one of the few exceptions where the GNC block diagram in the simulator differs from its real life counterpart, as actuators will always optimize the control based on the readings sensors pick up, but the execution is dependent on the real limits of the system ( $P_{a,real}$  and not  $P_{a,ideal}$ ).

## 8.2 Design of Experiment Campaigns

In this section we will discuss the main DoE campaigns used to determine the feasibility of RTLS operations for the nominal RLV candidate. Additionally, other DoE campaigns are carried out to examine the robustness of the algorithm to different burn schedules and final times. Moreover, interactions between  $f_{sim}$  and  $f_{gui}$  are examined to determine what effect the delay on guidance update has on the physical feasibility of the solution. Last of all, The sensitivity analysis, considered as the final main DoE to examine the effects of the algorithm with input errors and model mismatch is carried out.

Throughout each DoE there is an evolution of the nominal trajectory that is carried over to the next DoE. This process of sequential development aims to cover as many different cases as possible and of interest are solutions that are just outside the bounds of feasibility for metrics of  $\|\mathbf{r}_f\|$  and  $\|\mathbf{v}_f\|$ . The purpose of selecting nominal trajectories under this criteria is to discover cases that may improve the feasibility of the solutions. In these DoE, except for the sensitivity analysis, the dynamic models remain constants, and as such there is no reason to carry over feasible results over the next experiment. Understanding what types of values variables should undertake based on the type of starting conditions (specifically burn schedule and  $t_f$ ) can serve as a useful first-hand approach to development of automatic gain selection once enough data is gathered.

### 8.2.1 DoE1- Burn Schedule and Final Time

The first major design of experiments for high fidelity simulations involves studying the influence of the burn schedule and final time on the feasibility metrics  $\|\mathbf{r}_f\|$ ,  $\|\mathbf{v}_f\|$ , and  $Q_{dyn}$ . In determining the feasibility of operations, varying  $t_f$ ,  $t_1$ , and  $t_2$  gives insight into both the physical feasibility of RTLS and that of DESCOCO. Moreover, as was done in [20], this provides a trade-off map between feasibility metrics and solution metrics  $t_f$  and  $m_{burn}$ . Whilst the objective of the SOCP problems are concerned with maximizing the final mass, the analysis can nonetheless extend to optimizing the final time indirectly by varying the parameters in the DoE. The reason for selecting these parameters as the first step for high fidelity simulations is to include parameters that are universal to a convex optimization approach for RTLS. Moreover, nominal values with reasoning for choice have already been provided for DESCENDO, and the algorithm used here shares many similarities. Furthermore, as the burn schedule(s), thus  $q\forall t$  requires a unique controller for every unique value of  $q$ , it makes sense to focus initial efforts on debugging and verification of the multiple controller approach by varying  $t_f$ ,  $t_1$ , and  $t_2$ , where the highest number of controller cases for  $q$  are likely to be covered.

The number of controllers needed to be generated for this DoE is 191 for each SOCP problem. There are in total four (2 at  $t = 0$ , and 2 more for  $t > 0$ ). This results in a total of 764 controllers.

### 8.2.2 DoE2- Algorithm Parameter Tuning

The second DoE concerns itself with optimizing parameters that are mostly exclusive to the SOCP problem(s). This involves  $w_{\eta_w}$ ,  $\theta_{max}$ ,  $N_{dg}$ , and  $N_{cvx}$ , with nominal trajectory selection. The nominal trajectories are chosen by analysing the feasible solutions for  $\|\mathbf{r}_f\|$  (thus within 6 meters), and infeasible in  $\|\mathbf{v}_f\|$ . Then, the maximum value of  $\|\mathbf{v}_f\|$  for this cluster of cases is obtained and used to extend the search space for  $\|\mathbf{v}_f\|$  by setting this maximum value as the new  $v_{tol,f}$ . The same procedure is also done for cases that are feasible in  $\|\mathbf{v}_f\|$  but infeasible in  $\|\mathbf{r}_f\|$ . This procedure is repeated and both  $v_{tol,f}$  and  $r_{tol,f}$  is extended and saved for each iteration until when the number of cases of the infeasible metric starts showing a non-linear correlation with the % increase of the last iteration's  $v_{tol,f}$  or  $r_{tol,f}$  value. Finally, for each of the ranges between each iteration of clusters, a random value is sampled. No cross-sampling (that is mixing the ranges of the two metrics  $\|\mathbf{r}_f\|$  and  $\|\mathbf{v}_f\|$ ) is done as it is deemed that enough nominal trajectories can be obtained this way. The minimum number of trajectories accepted will be 4, with a maximum of 9, meaning each metric will need to have at least 2 iterations passed.

The reasoning for this type of sampling is to obtain a solution envelope that is worthwhile examining and focusing on results that do are not feasible, by introducing variations on variables that control both the size of the problem, modify the objectives, and alter the constraints. To reduce/increase the size of the problem,  $N_{dg}$  variations can be applied to increase/decrease the discretization grid. This will increase the solution time but may also provide a more robust control by increasing the accuracy of the prediction horizon (although the prediction horizon remains the same). More points are evaluated in the grid and as such more variations of the control can be applied which may also result in a smoother control profile, and as such if noticeable variations occur in the trajectory the control profile does not have to alter as much. A non-smooth control profile

can result in a prediction and control horizon that is non-smooth and as such variations on the dynamics which the convex problem does not respond well too might introduce more steep control profiles in an effort to find a feasible solution. The objective for SOCP2 can be controlled by varying  $\mathcal{W}_{\eta_w}$ , increasing or decrease the weight of the reference thrust vector on the new thrust vector computed by the current  $i^{th}$  SOCP2 iteration. This in itself can also relax constraints and allow more freedom in calculating the prediction horizon. On the other hand increasing the value may also be beneficial by providing an already existing prediction horizon which may be useful especially if the problem is more robust. In fact, one can see that increases in  $N_{dg}$  and  $\Xi_{\eta_w}$  may work hand-in-hand in improving the feasibility of the solution. Last of all,  $\theta_{max}$  provides a value that can be used to increase/decrease the control envelope of the solution. This can result both in cases where a more restricted direction profile can result in improved or reduced feasibility, the same is true for a more relaxed direction profile. In the first case, reduced the direction profile can reduced the search space by restraining the results to a smaller direction envelope, meaning that it is more likely that between successive convexification iterations but also between different guidance steps the direction profile is more likely to remain the same. With a wider direction profile if one solution calculates a reference thrust acceleration vector that is noticeably larger than the previous ones, we have to rely that the change in dynamics in future simulation steps will allow that thrust value to be accepted. Indeed, the expectation from decreasing  $\theta_{max}$  will be either improving the number of feasible guidance steps calculated along the trajectory or completely worsening the overall feasibility of the solution.

DoE 2 generates controllers based on  $N_{dg}$  and burn schedule. In total, 10720 controllers are generated, and processes are segmented by the value of  $N_{dg}$  to reduce the size of the cell structure used to store the controllers when passing the controllers to the input structure of the simulator.

### 8.2.3 DoE3- Guidance and Simulation Frequencies

This DoE aims to analyse the effect of  $f_{sim}$  and  $f_{gui}$ . The purpose of this experiment is to determine whether, due to the model mismatch of the SOCP problem and the environment of the simulator, the physical feasibility of the solution is impacted. There may be a bottleneck, where increasing  $f_{sim}$  any further does not cause any noticeable differences in the trajectory. For very high fidelity simulations, a fast update time of the dynamics is key in replicating the real world effect that guidance delay has on the RTLS. These simulations will also be run at a fixed time step with RK4 and such is done so that the focus remains on the selection of these parameters and not on the influence that the integrator may have when dealing with variable step sizes. Finding these bottlenecks can be useful for understanding how high  $f_{sim}$  needs to be without needlessly increasing it to the point where no real impact on the feasibility occurs.

The choice of  $f_{gui}$  is limited by the stakeholder requirement where the guidance controller has a limit of 25Hz. Increasing the guidance frequency up to these values may not provide an effective strategy to studying the coupled effects of  $f_{sim}$  and  $f_{gui}$ . This is because a perfect simulator would have a  $f_{sim}$  that approaches infinity. As such, applying very high values of  $f_{gui}$  would require a sensible increase in  $f_{sim}$  as well for the results to mean anything. Therefore, it is especially true that the ratio  $\frac{f_{sim}}{f_{gui}}$  be taken into account as well. For example, a choice of 10 to 1 may provide a less precise reference solution than 60 to 25. However, it is proves more useful in determining the influence of model and parameter mismatch between the environment and the guidance algorithm.

### 8.2.4 DoE4- Sensitivity Analysis

The sensitivity analysis aims to determine the degree of uncertainty to which the algorithm can still obtain feasible results. The approach used here is a one-at-a-time sensitivity analysis. In addition to uncertainties and discrepancies fed to the DG block and the SDK block, the variation of the initial starting conditions will also be analysed. Moreover, an additional time delay source is added to the guidance algorithm, and that is the processing time delay. To generate the random variables, each separate sensitivity experiment uses 3 different seeds, thus a total of 9 seeds are used. The random number algorithm used is the mersenne twister with MATLAB's rand and normrnd function, if random variables with a normal distribution are required. The methods for how these experiments are set are discussed below.

Parameter	Nominal Value	Variation	Total Variations
$v_{N,0}$ (m/s)	0		300
$v_{E,0}$ (m/s)	90.06073	$\pm 10\% \ \mathbf{v}_{NED,0,nominal}\ $	300
$v_{D,0}$ (m/s)	-456.025		300
$h_0$ (m)	26008.95	$\pm 10\% h_{0,nominal}$	900
$m_0$ (kg)	30254.15	$\pm 10\% m_{p,nominal}$	900
$X_0$ (m)	29938.44	$\pm 10\% X_{0,nominal}$	900

Table 8.1: DoE 4 sensitivity analyses for initial state variations. 3600 total cases. Each case used seed values of 10, 11, and 12 with the mersenne twister algorithm.

### Initial State Variations

The variation of the initial state can help expand the search space of feasibility for RTLS of the launch vehicle by compensating for the approximated methods of finding the initial state through the preliminary ascent section of the study. The parameters that are varied and their values are shown in table 8.1. The seeds used for creating the random variables are 10, 11 and 12, each used to generate an equal number of cases for each variable. The motivation for selecting these variables and their variation is as follows. A nominal variation of 10% is considered and the approach used here is one of montecarlo sampling. The values are generated with a nominal distribution, with the mean equation to the nominal values of each variable plus an increase or decrease of 10%, which corresponds to  $3\sigma$ . The general equation for generating the variation of the nominal parameters is shown in eq. 8.1 and 8.2.

$$\mu = \text{Nominal} \quad (8.1)$$

$$\sigma = \frac{\text{Nominal}}{3} \quad (8.2)$$

The value of 10% is assumed to represent the variations that result from the previous disciplines of preliminary ascent and descent tasked with finding the nominal trajectory(ies). Moreover, it also takes into account the degree of error associated with using these methods. We are more interested however in errors of smaller degree, and for that reason a normal distribution is used here to increase the pool of variations that are closest to the nominal values.

### Processing Time Delay

The second set of variables studied in the sensitivity analyses is the processing time delay  $t_{pt}$ . Fig. 8.2 shows what errors can be present in the simulator for an example case where  $f_{sim}/f_{gui} = 2$ .

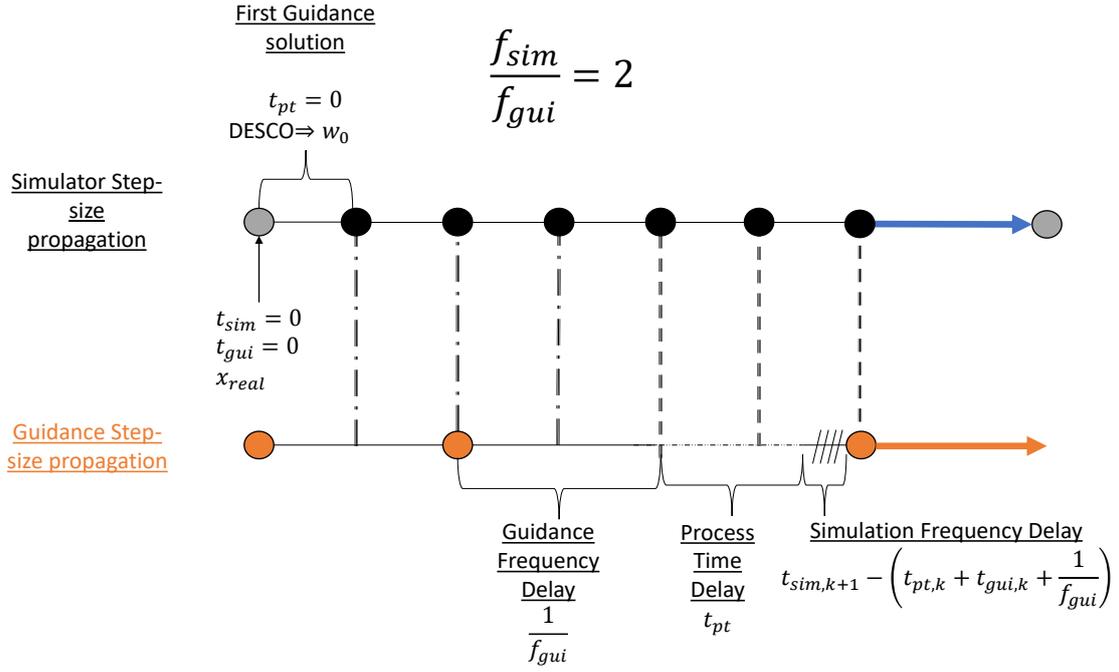


Figure 8.2: Time lag effects between guidance algorithm and simulator.

This sensitivity analysis aims to study the effects of the processing time delay, by varying the total processing time for the DESCOCO algorithm by a factor of .001 to 10. The motivation for this is that it is assumed that the most powerful on-board guidance computer will be able to compute the solutions 10 times faster than what the machine that ran the simulations can achieve. The opposite is true for the slowest guidance computer. This factor is simply considered as a heuristical value and may actually vary from what the actual VEGA on-board computer can achieve. However, due to the lack of information on the hardware of the guidance computer, we resort to this assumption. The distribution of variation however is made to favour the case where a more powerful computer is available. Moreover, because this parameter does not deal with differences that are within the  $\pm 10\%$  variation of the nominal parameter, but rather deals with variations that are order of magnitude difference, the random variables are passed through an exponential filter which results in an absolute value. In order to do this, the normal distribution is first calculated using the log equivalent of the ranges.

### Pseudo-range Errors

The real time errors are modelled as errors that stem from the calculation of the pseudo-range  $\mathcal{P}$ . The pseudo-range is assumed to be the distance that the position information of the RLV travels from the RLV to the ground-station (assumed as the LP) and back, to inform the guidance controller on-board its position relative to the LP. The information is assumed to be transmitted as an electromagnetic wave, with negligible doppler shift effects and other time varying effects. The pseudo-range  $\mathcal{P}$  can be calculated as show in eq. 8.3. The distances between the missile position (subscript  $M$ ) and LP  $\Delta x_{M,LP}$ ,  $\Delta y_{M,LP}$ , and  $\Delta z_{M,LP}$  are given in eq. 8.4 to 8.6.

$$\mathcal{P} = \sqrt{(\Delta x_{M,LP})^2 + (\Delta y_{M,LP})^2 + (\Delta z_{M,LP})^2} \quad (8.3)$$

$$\Delta x_{M,LP} = x_M^{ECEF} - x_{LP}^{ECEF} \quad (8.4)$$

$$\Delta y_{M,LP} = y_M^{ECEF} - y_{LP}^{ECEF} \quad (8.5)$$

$$\Delta z_{M,LP} = z_M^{ECEF} - z_{LP}^{ECEF} \quad (8.6)$$

The errors considered in the pseudo-range stem from the UERE (User Equivalent Range Errors). These errors are shown in tab. 8.2. The UERE total error is taken as the sum of these errors based on the region of altitude  $h$ .

Error Source	Error Range (m)	$h$ threshold (m)
Satellite Clocks	$\pm 2$	[0, $\infty$ ]
Orbits Errors	$\pm 2.5$	
Receiver Noise	$\pm .3$	
Multipath	$\pm 1$	
Ionospheric Delays	$\pm 5$	[48000, $\infty$ ]
Tropospheric Delays	$\pm .5$	[8000, 15000]

Table 8.2: User equivalent range errors used for DoE 4 and regions of  $h$  for activation.

Using these errors, the ECEF position of the missile can be calculated by equation 8.7. The error  $\epsilon_{r,ECEF}$ . As with the previous experiments, the random error values are generated with a normal distribution with a mean of 0 and  $\sigma$  that varies between 0 and 3 with a uniform distribution, equivalent to the sum of all UERE sources given the current  $h$ . The seed used for variations of  $\sigma$  are 10, 11, and 12.

$$\mathbf{r}_{M,error}^{ECEF} = \mathbf{r}_M^{ECEF} + \epsilon_{r,ECEF} \quad (8.7)$$

The error in position is calculated by considering  $\delta_{meas}$  and  $\lambda_{meas}$  as given in equations 8.8 and 8.9.

$$\delta_{meas} = \arctan\left(\frac{r_{y,M,error}}{r_{y,M,error}}\right) \quad (8.8)$$

$$\lambda_{meas} = \arctan\left(\frac{r_{z,M,error}}{\|r_{x,y,M,error}\|}\right) \quad (8.9)$$

The measured pseudo-range can then be calculated by eq. 8.10.

$$\mathcal{P} = \|(\mathbf{r}_{LP}^{ECEF} - \mathbf{r}_{M,error}^{ECEF})\| \quad (8.10)$$

Finally, the velocity errors can be found by considering the signal travel distance to and from the ground station and with the speed of light as a constant of 299792458 m/s. The equation for the measured NED velocity is given in 8.11.

$$\mathbf{v}_{NED,meas} = \mathbf{v}_{NED,real} + \dot{\mathbf{v}}_{NED,real} \frac{2\mathcal{P}}{c} \quad (8.11)$$

With equations 8.11, 8.8, and 8.9, the errors for the experiments can be calculated.

# Chapter 9

## Guidance Algorithm

The guidance algorithm used for this study and subsequent variations are based on convex programming. Within the convex programming spectrum, the guidance algorithm sets up the problem as a second-order cone program (SOCP). The algorithm is a rewritten version of the DESCENDO algorithm [54], using MATLAB and the YALMIP toolbox to construct an optimizer object, which is by all intents and purposes a controller, to solve the RTLS problem in a closed-loop.

This chapter starts with a basic overview of the convex programming problem, and the subset of SOCP problems. It then introduces the problem set up for the SOCP1 problem, and subsequently the SOCP2 problem, explaining the procedures of successive and lossless convexification. Following this, the practical improvements made in the algorithm, entitled DESCO (Descent Envelope With Successive Convex Optimization), written for this project, over the original DESCENDO algorithm will be highlighted. Then, the general pseudo-code procedure for the algorithm is introduced. Lastly, a validation of the program with sample problems will be highlighted to show that the algorithm meets the constraint requirements imposed in the optimizer build up.

### 9.1 Convex Optimization

Convex optimization is a class of mathematical optimization where the objective function and constraints are convex. Intuitively, this means that the function that describes the objectives and constraints has a segment between any points on the graph that can be drawn, and has the function below this segment. This is mathematically portrayed by the relations shown by eq. 9.1 to 9.2.

$$\begin{aligned} f_i(\alpha x + \beta y) &\leq \alpha f_i(x) + \beta f_i(y) \\ \forall x, y \in \mathbb{R}^n \\ \forall \alpha, \beta \in \mathbb{R} \end{aligned} \tag{9.1}$$

$$\begin{aligned} \alpha + \beta &= 1 \\ \alpha &\geq 0 \\ \beta &\geq 0 \end{aligned} \tag{9.2}$$

An important distinction is that eq. 9.1's first statement may seem similar to that of the objective for a linear problem, which is instead denoted by an equal sign rather than an inequality, as shown in eq. 9.3.

$$f_i(\alpha x + \beta y) = \alpha f_i(x) + \beta f_i(y) \tag{9.3}$$

This difference also highlights the fact that a convex problem has less strict requirements than a linear problem. Nonetheless, a linear program is a convex programming problem, and we can consider convex optimization as a class of linear optimization. Following this, we can form a general description of the convex optimization problem with the objective described by eq. 9.4.

$$\min f_0(x) \tag{9.4}$$

This statement presumes that the functions  $f_0, \dots, f_m$ , which are the  $m$  constraints and objective functions, are convex, and therefore satisfy eq. 9.1 and 9.2. A special subclass of convex programming problems are the least-squares programming problems, and the linear programming problems. These are discussed in the following subsections below.

### 9.1.1 Least Squares Problem

The least squares problem has no constraints, therefore  $m = 0$  in eq. 9.4. Its objective function is instead of the form given in eq. 9.5, where  $a_i^T$  are the rows of  $A$ , and  $x$  is the decision variable vector.

$$\begin{aligned} \min f_0(x) &= \|Ax - b\|_2^2 = \sum_{i=1}^k (a_i^T x - b_i)^2 \\ &x \in \mathbb{R}^n \\ &A \in \mathbb{R}^{k \times n} \\ &k \geq n \end{aligned} \quad (9.5)$$

These types of problems can be solved by forming a system of linear equations of the form shown in eq. 9.6, giving us the solution, obtainable analytically, in eq. 9.7.

$$(A^T A)x = A^T b \quad (9.6)$$

$$x = (A^T A)^{-1} A^T b \quad (9.7)$$

These types of problems have been solved for decades, and their solution times are restrained by Moore's law: they will decrease exponentially in the future with the advent of more powerful computer architectures. Generally speaking, the least squares problem can be solved with a time that is roughly equal in order of magnitude to  $n^2 k$ .

### 9.1.2 Linear Programming Problem

Linear programming problems are problems where the objective and constraint functions are linear. The general formulation of the linear programming problem is given in eq. 9.8. The parameters  $c$ , and  $a_i$  are vectors and  $b_i$  are scalars which are  $\in \mathbb{R}$ , and specify the objectives and constraint functions, respectively.

$$\begin{aligned} \min c^T x \\ \text{subject to } a_i^T x \leq b_i, i = 1, \dots, m \end{aligned} \quad (9.8)$$

In other words, linear constraints are constraints of the form shown in 9.8, where on both sides of the inequality the expressions are linear polynomials.

The difference between linear programming and least-squares problems does not end in their method of definition of objectives and constraints. Linear programming problems cannot always be solved analytically, especially if we impose several constraints. Nevertheless, they both share the similarity in that both problems have been explored in academics for a long time, and therefore there are numerical methods which can be used to solve such problems. A rule of thumb in identifying the complexity of solving a linear program is taking order of magnitude of  $n^2 m$ , assuming  $m \geq n$ .

### 9.1.3 Convex Optimization Problem

By combining the characteristics of the linear programming problem and the least squares problem, we can obtain the convex optimization problem. That is, we have a problem of the form given by eq. 9.9, where we have both an objective function and inequality and equality constraints,  $f_i$  and  $h_i$ , respectively.

$$\begin{aligned} \min f_0(x) \\ \text{subject to } f_i \leq 0, i = 1, \dots, m \\ h_i(x) = 0, i = 1, \dots, p \end{aligned} \quad (9.9)$$

The inequality constraints and objective functions must be convex, whilst the equality constraint function can be affine. This means that the function, which contains the set  $C \subseteq \mathbb{R}^n$ , has any two distinct points in  $C$  lying in  $C$ , or in other words, any two points in  $C$  are a linear combination that is part of  $C$ .

### Quadratic Optimization Problems

A convex optimization problem is called a quadratic program (QP) if it has affine constraint functions and a quadratic objective function. The general formulation of the QP is given in eq. 9.10.

$$\begin{aligned}
 & \min \frac{1}{2} x^T P x + q^T x + r \\
 & \text{subject to } Gx \leq h \\
 & \quad Ax = b \\
 & \quad P \in \mathbb{S}_+^n \\
 & \quad G \in \mathbb{R}^{m \times n} \\
 & \quad A \in \mathbb{R}^{p \times n}
 \end{aligned} \tag{9.10}$$

In the QP problem, a convex quadratic function is minimized over a polyhedron, which contains a feasible set for which the optimal value of the decision variable  $x$ , may lie within or on the bounds of.

A special class of QP programs are the quadratically constrained quadratic programs (QCQP). These have multiple intersecting ellipsoids whose mutual set describes the feasible region for which the optimal value of  $x$  may lie in. QP may also be reformulated into linear programs when setting  $P = 0$ , whilst QCQP may be reformulated into QP by setting  $P_i = 0 \forall i$ .

### Second-order cone programming

Closely related to the QCQP problem is the second-order cone programming problem (SOCP), whose general formulation is given by eq. 9.11.

$$\begin{aligned}
 & \min f^T x \\
 & \text{subject to } \|A_i x + b_i\|_2 \leq c_i^T x + d_i, i = 1, \dots, m
 \end{aligned} \tag{9.11}$$

For SOCP problems, it holds true that  $A \in \mathbb{R}^{k \times n}$ , which is called the second-order cone constraint. The newly defined optimization variable  $g$ , and  $x$ , are respectively  $\in \mathbb{R}^p$  and  $\in \mathbb{R}^n$ . As the name suggests, SOCP problems contain intersecting cones, rather than general polyhedrons, which contain the feasible set for which the optimal value of both these decision variable lies in. The quadratic objective function, for which we want to find the optimum, intersects with these second-order, or in other words, quadratic, cones.

#### 9.1.4 Guidance Strategy

Convex optimization problems have gained traction over years not just as a general mathematical procedure to solve any sort of problems with given equality, inequality constraints and an convex objective function. There is evidence to show that in the field of the solving fuel-optimal trajectory problems with state and control constraints they have become an increasingly popular strategy to solve these sorts of problems. Moreover, and most importantly, with the increase of power in modern computers and software to reliably solve multiple classes of convex optimization problems, they have gained popularity for their ability to solve such problems in a closed-loop fashion.

The guidance strategy, which this project attempted to replicate, is based on the DESCENDO (descending over extended envelopes using successive convexification-based optimization) algorithm. Aside from being a SOCP, the distinguishing characteristics of the algorithm are its usage of the procedures of lossless and successive convexification.

#### Lossless Convexification

Lossless convexification is used to 'relax' non-convex constraints in a convex form. The keyword lossless indicates that no region of the feasible space of the solution disappears through the process of convexification. The two dynamics which are non-convex are the propellant depletion dynamics, and the thrust magnitude and pointing constraints. Both of these have previously been studied and proven to work [54]. The propellant depletion dynamics is convexified by change of coordinates. We introduce the new variable  $z(t)$ , which is the log of the mass of the launch vehicle  $\hat{m}(t)$ . The mass and mass flow rate can then be written as shown by eq. 9.12 and 9.13 respectively.

$$z(t) = \ln \hat{m}(t) \tag{9.12}$$

$$\dot{z}(t) = \frac{\dot{\hat{m}}(t)}{\hat{m}(t)} \quad (9.13)$$

The second source of non-convexification requires the introduction of another optimization constraint using two new changes of coordinates, denoted by  $\mathcal{W}(t)$  and  $\sigma(t)$ . These two new constraints are shown in eq. 9.14 and 9.15.

$$\mathcal{W}(t) = \frac{\mathbf{T}_{CVX}(t)}{\hat{m}(t)} \quad (9.14)$$

$$\sigma(t) = \frac{\|\mathbf{T}_{CVX}(t)\|}{\hat{m}(t)} \quad (9.15)$$

### Successive Convexification

Successive convexification is the process of introducing non-convex constraints, or objective functions, as pre-determined values of a previous convex problem, into a new convex problem. The DESCENDO algorithm was written with the inclusion of successive convexification in mind as part of the work done by Jerez et al. [20]. In it, the optimality of the solution is increased by introducing aerodynamic effects in a second procedure based on the state history of the first solution. We rewrite the force drag term included in eq. 4.35 with the newly introduced  $z(t)$  variable to formulate the drag dynamics for the successively convexified iteration(s), as shown in eq. 9.24 as the drag acceleration.

$$d^*(t) = \frac{1}{2} \rho^* S_{ref} C_D^* \frac{\|\mathbf{v}^*(t)\|}{e^{z^*(t)}} \quad (9.16)$$

## 9.2 DESCO

DESCO is meant to take all functioning characteristics of the DESCENDO algorithm but improve its computational efficiency through the use of a different framework for solving convex optimization problems. In the original paper of DESCENDO, CVX, a MATLAB software for disciplined convex programming was used to build the optimizer routine that computed the trajectory given the inputs of the state of the vehicle. The same principle is mostly true in the DESCO algorithm, however the software YALMIP has been adopted to reduce the overhead time in building the problem. YALMIP is able to create a controller beforehand, that is prior to the launch of the simulation, in a process that takes nonetheless no more than a few seconds. The key point here however is that for the amount of cases to be run in the design of experiments whose purpose is to select the most robust tuning parameters of the algorithm, these few seconds can become hundreds if not thousands of hours. In fact, in comparison to CVX, YALMIP's feature of building a controller means that successive iterations and calls to the algorithm within a simulation will take tenths of seconds rather than whole seconds.

Additionally, whilst the original DESCENDO algorithm has not been procured to be compared with DESCO, the controller framework in YALMIP allows the usage of Big-M reformulations that is more of a programming technicality than a feature like lossless convexification and successive convexification. Big-M reformulations involves the process of converting a non-convex or binary (logic) constraint to a set of constraints using a change of coordinates by introduces additional binary variables and binary constraints. In an intuitive sense, this allows the usage of **if** statements, whose influence in DESCO will be explained in section 9.3.

### 9.2.1 Problem Formulation

The SOCP (second-order cone programming) problem is introduced in this section. The equations of motion used for calculating the state (mass, velocity, and position) are in the LP frame. These equations consider a very simple euler scheme where the position is propagated through the discretized points in time by its proportional relationship to the current velocity, and the rate of change of velocity. The time interval between each discretized points is governed by the relationship in eq. 9.17. In this equation the time interval  $T_s$  is related to the final time  $t_f$ , current time  $t$ , and number of discretization points  $N$ .

$$T_s = \frac{t_f - t}{N - 1} \quad (9.17)$$

The equations of motion for the launch vehicle in the LP frame do not consider non-inertial effects, precisely because the conversion from the algorithm's frame of reference to the simulator's equation of motion frame

of reference (E frame), and vice-versa, do not require them. The discretized dynamics equations are given in equations 9.18 to 9.20.

$$\mathbf{r}[k+1] = \mathbf{r}[k] + T_s \mathbf{v}[k] + \frac{T_s^2}{3} (\mathbf{a}[k] + \frac{\mathbf{a}[k+1]}{2}) \quad (9.18)$$

$$\mathbf{v}[k+1] = \mathbf{v}[k] + \frac{T_s}{2} (\mathbf{a}[k] + \mathbf{a}[k+1]) \quad (9.19)$$

$$z[k+1] = z[k] - \frac{1}{I_{sp,oa} g_0} \frac{T_s}{2} (\sigma[k] + \sigma[k+1]) \quad (9.20)$$

The variables  $\mathbf{a}$  and  $\sigma$ , are surrogate variables, and are defined slightly different for the initial problem solution in the algorithm than the successive convexification iterations.

### SOCP1 & SOCP2

The DESCO algorithm, like its original counterpart DESCENDO, works by solving a simplified RTLS problem with no drag dynamics with the dynamics equations listed in 9.18 to 9.20. This procedure takes place in SOCP1, where nevertheless lossless convexification is applied. Successive convexification however, is handled by SOCP2, and its subsequent iterations. In this procedure, what otherwise would be nonconvex terms are introduced as a vector of already determined values for each different time step. This convexifies (turns linear) nonlinear dynamics by introducing them as an input rather than as a term to solve. By pairing this constraint with a regularization parameter in the thrust history that penalizes large changes in the thrust acceleration vector, it renders neighbouring iteration of solutions not too different from one another. There are additional constraints imposed upon SOCP2, including a trust region constraint, and additional boundary conditions, which are described below. The algorithms for SOCP 1 and SOCP 2 are near exact to the ones used in [54] and are shown in appendix B.4.2 and B.4.4, including their counterparts for  $t = 0$  in B.4.1 and B.4.3.

### Objective Functions

The DESCO algorithm's main objective is to minimize the fuel mass expended. Considering the change of coordinates introduced in equation 9.12 for the mass, this means that we must maximize the final value of  $z$ , and in discrete terms  $z[N]$ . This is shown in equation 9.21, where an equivalent form of the expression using the newly introduced  $\sigma$  variable for thrust acceleration constraint is included. Indeed, whilst the earlier form of the objective function is used for the algorithm, these two expressions are identical, albeit numerical errors and for the purposes of clarity we adopt the more direct approach that means more explicitly maximizing the final mass of the vehicle.

$$\max z[N] \quad (9.21)$$

A design choice for SOCP2 is made to augment, by regularization (that is by adding additional linear terms), the objective function. The SOCP2 objective function is then described by eq. 9.22.

$$\max z[N] - \mathcal{W}_{\eta_w} \sum_{k=1}^N \eta_w[k] \quad (9.22)$$

The summation term contains a weight factor  $\mathcal{W}_{\eta_w}$  where  $\eta_w$  is the norm of the difference between thrust profiles of the first SOCP1 solution or previous successive convexification iterations. The reason for included the weight factor is so one can balance the importance of the original objective term (log of the final mass  $z[N]$ ) and the differences in thrust profiles. The reason for regularizing the function in the first place is to allow for solutions to not vary, or vary significantly, depending on the mission and thrust profile achievable, whether previous thrust profiles are as reliable as newly calculated ones. With this phraseology it may be obvious that regularizing the term prevents us from finding an optimal solution only with respect to the final mass, and this is also in fact quite clear from the expression itself. However, as the very purpose of successive convexification is to introduce non-convex constraints are readily evaluated terms, each SOCP2 algorithm with the inclusion of the newly added dynamics of drag will unequivocally generate a different thrust profile using the same objective function. Whilst the algorithm has control rate constraints in order to function with the characteristics of the MX and M10 launchers, new iterations of DESCO have no obligation to keep the same profile of the previous iterations. That is, without the regularization term successive convexifications will cause rapidly changing thrust profiles that will still adhere to the control rate constraints, but is not favorable for guidance when errors are introduced as the thrust profile is less predictable. This can cause a non-smooth

thrust profile which may be more efficient in reaching maximizing fuel mass and landing. However, because of this increased freedom in the thrust profile the RLV may favour an increased number of burns over single longer smooth burns. This may also cause multiple burns than scheduled, and in real life applications, delays due to transient time in engine starts and stops may accumulate errors in the long run and cause infeasible landings.

### Surrogate Variables

A surrogate variable is used to express another variable or an expression containing two or more variables. This is done for simplifying expressions in the programming of the algorithm for ease of readability for the user. In the dynamics eq. 9.18 to 9.20 the surrogate variable is the acceleration  $\mathbf{a}[k]$ . This variable has different expressions for the SOCP1 and SOCP2 algorithm, given that the latter introduces the drag term. For SOCP1, the surrogate variable is given in eq. 9.23. The gravity  $\hat{\mathbf{g}}(t)$  is the gravitational acceleration vector in the LP frame taken at the time of the first index of the algorithm's iteration.

$$\mathbf{a}[k] = \mathbf{w}[k] + \hat{\mathbf{g}}(t) \quad (9.23)$$

SOCP2 introduces the drag term that is expressed in a continuous formulation as eq. 9.16. In the discrete formulation this can be written using eq. 9.24. The asterix superscripts denote that it is a supplied (already known) vector. Note that the drag is not a vector quantity for each time index, and that the drag is in fact approximated as the overall aerodynamic force on the vehicle in all axis of the LP frame.

$$d_i^*[k] = \frac{1}{2} \rho_i^*[k] S_{ref} C_{D,i}^*[k] \frac{\|\mathbf{v}_i^*[k]\|}{\exp z_i^*[k]} \quad (9.24)$$

The directional vector in which the drag acts is antiparallel to the RLV velocity vector, and thus, for the SOCP2 algorithm the surrogate variable for acceleration is written as given in eq. 9.25.

$$\mathbf{a}[k] = \mathbf{w}[k] - \hat{\mathbf{g}}(t) - d_i^*[k] \mathbf{v}[k] \quad (9.25)$$

Both these expressions are written as equality constraints in the respective algorithms. On the contrary, a surrogate variable, introduced as part of the lossless convexification procedure,  $\sigma$ , is expressed as a surrogate variable for the norm of the thrust acceleration vector as an inequality constraint. This constraint is expressed in eq. 9.26.

$$\|\mathbf{w}[k]\| \leq \sigma[k] \quad (9.26)$$

### Boundary Conditions

The objective of the algorithm however needs to be guided by constraints, as introduced in the convex optimization problem formulation in eq. 9.4. Boundary conditions refer to constraints that are placed on the final and or initial index. The requirements for RTLS as mentioned in table 3.3 is to execute a landing with a sufficiently low final velocity and deviation from the launch site. Whilst there is a margin of error for both requirements, only the final velocity has an inequality constraint in the SOCP1 problem. This is to ensure that any errors in position do not build up upon the existing deviation from the exact landing coordinates from the algorithm's solution. The boundary conditions for the position and velocity of the RLV are expressed in eq. 9.27. We note that continuous terms with the hat indicate single values fed to the algorithm at time  $t$  which may be non-zero.

$$\begin{aligned} \mathbf{r}[1] &= \hat{\mathbf{r}}(t) \\ \mathbf{r}[N] &= \mathbf{r}_f \\ \mathbf{v}[1] &= \hat{\mathbf{v}}(t) \\ \mathbf{v}[N] &\leq \mathbf{v}_f \end{aligned} \quad (9.27)$$

For mass, the first boundary condition is placed at the starting time of the algorithm iteration, which is the natural log of the mass at that time. The final time index constraint for the mass is expressed as an inequality constraint for which the variable  $z$  must be greater than or equal to the natural log of the dry mass of the vehicle. These two constraints are expressed in eq. 9.28.

$$\begin{aligned} z[1] &= \ln \hat{m}(t) \\ z[N] &\geq \ln m_{dry,1} \end{aligned} \quad (9.28)$$

Moreover, because the guidance algorithm does not take into account attitude angles, both the algorithm and the simulator for this study will calculate all needed angles to establish the approximated orientation of the RLV from the thrust vector. Therefore, constraints on the final thrust vector in the LP frame have to be placed. Furthermore, for the purposes of this study the thrust profile must be as smooth as possible when firing. This is because rapid throttling of the engines can result in an unreliable thrust profile when considering the real-world application.

In addition to the regularization term in the objective function, an initial condition on the thrust vector based on either the vehicle starting condition at a non-zero time of the simulation must be placed based on the vehicle's current thrust acceleration. This means that the final landing condition, considering that the thrust vector's z axis in the LP frame corresponds to the vehicle's longitudinal position, and is antiparallel to the vehicle's x axis in the body frame, must result in zero for the thrust acceleration's x and y axis and greater than or equal to zero for the z axis (vertical landing). The thrust acceleration term  $w$  is then bounded for the first time index in the algorithm to its current acceleration. For the remainder of constraints a similar methodology follows, since the first time index of the algorithm is an input and not solution of the first future state index (which would be the second time index). The thrust acceleration boundary conditions are expressed in eq. 9.29.

$$\begin{aligned} \mathbf{w}[1] &= \hat{\mathbf{w}}(t) \\ \mathbf{w}[N] &\geq 0 \end{aligned} \quad (9.29)$$

### Control and Control Rate Constraints

When viewing the SOCP procedures as solving an optimal control problem with decision variables and control variables, the only control variable that is present is the thrust acceleration vector  $\mathbf{w}$ . This vector must be bounded by equivalent magnitude and rate constraints. In addition, as mentioned before, for both the simulator and algorithm this study assumes the orientation of the RLV based on the direction of thrust. As a result, a direction constraint on the firing angle must be imposed. This direction constraint is expressed in eq. 9.30. The variable  $\theta_{max}$  is a constant that can be adjusted that indicates the angle between the longitudinal of the vehicle and the vertical direction (LP frame z axis).

$$\mathbf{w}_z[k] \geq \frac{\|\mathbf{w}_{x,y}[k]\|}{\tan\theta_{max}} \quad (9.30)$$

The magnitude of the thrust is expressed using the  $\sigma$  variable as shown in eq. 9.31. The minimum and maximum thrust in Newtons is indicated by  $T_{min}$  and  $T_{max}$  respectively. Note that the fixed mass is used for all time indices of the algorithm to preserve the convexity. This approximation as stated by Simplicio et. al. [20], is not restrictive because the actual limits themselves may be adjusted to better represent the the real maximum and minimum thrust achievable by the propulsion system. Moreover, as the simulation progresses closer to the landing point, the deviation between the mass of the initial index and that of the final index becomes less. This approximation may further benefit as well by increasing the number of indices for each SOCP algorithm run as it realistically only affect the final index solutions as between closer spaced indices the mass is meant to drop off less when firing.

$$\frac{T_{min}}{\hat{m}(t)} \leq \sigma[k] \leq \frac{T_{max}}{\hat{m}(t)} \quad (9.31)$$

In addition to thrust magnitude constraints, we also supply thrust rate constraints using a forward discretization scheme. This means that, as we begin with the first time index as an already predetermined value for thrust and state of the RLV, we start calculating the thrust rate from the second time index of the SOCP problem. Given the thrust rate constraints  $\dot{T}_{min}$  and  $\dot{T}_{max}$  for the minimum and maximum N thrust per second (N/s), respectively, the thrust rate constraint is expressed as shown in eq. 9.32.

$$\sigma[k] - T_s \frac{\dot{T}_{max}}{\hat{m}(t)} \leq \sigma[k+1] \leq \sigma[k] + T_s \frac{\dot{T}_{max}}{\hat{m}(t)} \quad (9.32)$$

It is noted that the aforementioned constraint was of high importance in reducing chattering without forcing restrictions on eq. 9.26.

### Big-M Reformulation

The thrust constraint itself does not represent entirely the functional characteristics of the propulsion system. For any optimal control problem, due to the limits on both the MX's and M10's throttling capability ranges, an additional constraint must be imposed. In order to do this, the SOCP problems identify prespecified burn and coasting intervals where this constraint is turned off and on. From a programming perspective, the thrust constraint inequality, or for that matter, all inequalities represent  $N$  constraints in the problem. This means that the convex optimization problem handles the constraints at each time index, and does not generalize a constraint over the whole time span. Whilst this may indicate a problem in the efficiency in solving the algorithm, it actually allows us to specify constraints based on the time index, or in other words the burning periods.

As introduced at the start of this section, Big-M reformulations allow us to specify when the magnitude constraint should be applied. In a logical formulation, constraints shown by eq. 9.31, 9.32, and 9.30 are to be interpreted as shown in eq. 9.33.

$$\begin{cases} \mathbf{w}_z[k] \geq \frac{\|\mathbf{w}_{x,y}[k]\|}{\tan\theta_{max}}, \frac{T_{min}}{\hat{m}(t)} \leq \sigma[k] \leq \frac{T_{max}}{\hat{m}(t)} & \text{if } T_s(k-1) \in \mathcal{T}_p \\ \mathbf{w}[k] = \mathbf{0}, & \text{otherwise} \end{cases} \quad (9.33)$$

The pre-specified burn time is part of the set  $\mathcal{T}_p$ . In logical sense, we describe this with an if, or otherwise approach. In a programming sense, using Big-M formulation, we introduce a new variable  $q$ , which is a predetermined vector of dimension  $1 \times N$ . This is a binary vector for which a value of 1 at the corresponding index indicates that the time is within the set of  $\mathcal{T}_p$ , and 0 indicates it isn't, so there is no thrust. This variable  $q$  however, cannot be used as a binary factor to turn off or on the constraints by multiplying the left or right hand side of the constraints to result in a zero or non zero value (by virtue of  $1 - q[k]$ ), since multiplication of two convex variables would result in a non-convex problem. Rather, the inequalities must be driven to zero by using an offset, which as the name of the procedure implies, is  $M$ . In selecting  $M$ , the goal is to choose a number that is sufficiently big for which the expression may never reach. In order to choose such a number we have to look at which constraint has the fewest number of terms on the left and right hand inequalities, and in this case it is the thrust acceleration vector constraint on the first line of eq. 9.33. The vector  $\mathbf{w}$  is bounded by the maximum and minimum achievable thrust, not in terms of magnitude, but as a vector quantity. This means that the minimum acceleration is the negative of the maximum achievable magnitude of acceleration (not the minimum value of the thrust acceleration magnitude). By indicating that 1's denote burn periods and 0's denotes coasting, the thrust acceleration constraint is written programmatically in YALMIP as shown in 9.34. The value of  $M$  is then simply the maximum achievable acceleration, which will occur at the lowest possible mass value ( $m_{dry,1}$ ) and the highest thrust value  $T_{max}$ , as shown in eq. 9.35.

$$-Mq[k] \leq \mathbf{w}[k] \leq Mq[k] \quad (9.34)$$

$$M = \frac{T_{max}}{m_{dry,1}} \quad (9.35)$$

The direction constraint and thrust rate acceleration constraint can then be augmented with the terms  $M(1 - q[k])$  on the left and both sides of the inequalities, respectively. This is shown in eq. 9.36, and 9.37.

$$M(1 - q[k]) + \mathbf{w}_z \geq \frac{\|\mathbf{w}_{x,y}[k]\|}{\tan\theta_{max}} \quad (9.36)$$

$$-M(1 - q[k]) + \frac{T_{min}}{\hat{m}(t)} \leq \sigma[k] \leq M(1 - q[k]) + \frac{T_{max}}{\hat{m}(t)} \quad (9.37)$$

### Trust Region Constraint

SOCP2's objective function is regularized by the inclusion of the summation of differences in a newly generated SOCP's  $\mathbf{w}$  solution and the old. The term  $\eta_w[k]$ , is used to express this difference as an inequality constraint shown in eq. 9.38. The parameter  $\mathbf{w}^*[k]$  is the thrust acceleration profile of the previous SOCP iteration.

$$\|\mathbf{w}[k] - \mathbf{w}_i^*[k]\| \leq \eta_w[k] \quad (9.38)$$

This constraint, as mentioned by Simplicio et. al.[20], makes it easier for the algorithm to converge by restricting how much the thrust solution may change, and is called the trust region constraint (TRC). An emphasis

should be noted on the ease of convergence as it is an inequality constraint, whereas an equality constraint will be interpreted as more numerically stringent on the algorithm. Nevertheless, the algorithm will choose, if it is able to find the optimal solution, a value of  $\eta_w[k]$  that is equal to and not greater than or equal to the left hand side of the expression as it directly results in a better value of the objective function (eq. 9.22).

### Flight Path Constraint

An additional constraint in SOCP2 is the inclusion of a subsurface flight constraint and a flight path constraint (FPC). Eq. 9.39 covers both grounds of these requirements, where the RTLS trajectory is bounded to the interior of a shrinking cone [20], whose geometry is determined by the first index position of the RLV ( $\hat{\mathbf{r}}(t)$ ).

$$\mathbf{r}[k] \geq \frac{\mathbf{r}_z(\hat{t})}{\|\hat{\mathbf{r}}_{x,y}(t)\|} \|\mathbf{r}_{x,y}[k]\| \quad (9.39)$$

It is important to note that the inclusion of this constraint only in the SOCP1 algorithm was a design choice made in DESCENDO and carried over to DESCO, as the authors noted that feasible SOCP1 solutions could only be found by removing this constraint. Nevertheless, it was also noted that SOCP1 would find a solution that did not break this constraint despite it not being programmed in the algorithm.

## 9.2.2 Procedure

As mentioned before, DESCO is a rewrite of the DESCENDO algorithm by Simplicio et. al.[20], with the most noticeable difference is the usage of a different convex optimization software, YALMIP, instead of CVX. The interface between the simulator and the DESCO algorithm is shown in fig. B.3. Due to the algorithm's similarity with DESCENDO, the flowchart has been adapted from Simplicio et. al., safe for the inclusion of the initialization of the SOCP controllers and the inclusion of the SOCP at time 0 problems. Lines of green colors indicate a logical true to the diamond yellow shaded figures, which are logical statements. Lines of orange color indicate a logical false. Lines going away from blue boxes, which are calculation procedures are dotted to be visually more recognizable. At the SOCP2 feasibility check, a single line is drawn for simplicity but it is made note that the 'SOCP2t0' algorithm is called at  $t = 0$  just as the SOCP1t0. We recall that these algorithms do not have the initial thrust acceleration constraint  $\mathbf{w}_0$  and as such the optimization problem is tasked with finding the current thrust acceleration too.

Prior to the beginning of the simulation, the static inputs for the SOCP controllers are declared in a script. These static inputs include the weight for the regularization term in SOCP2  $\mathcal{W}_{\eta_w}$ , the maximum  $T_{max}$  and minimum  $T_{min}$  achievable thrust, the height threshold for SOCP2 to activate  $h_p$ , the number of indices in the SOCP problem  $N$ , the number of successive convexifications  $N_p$ , the minimum thrust rate  $\dot{T}_{min}$ , and maximum thrust rate  $\dot{T}_{max}$ , overall specific impulse  $I_{sp,oa}$ , and maximum direction angle between vehicle longitudinal axis and ground  $\theta_{max}$ . At any simulation time step, for which the time  $t_{sim}$  always corresponds to  $t$  (the governing frequency is the simulation frequency  $f_{sim}$ , a check is made to determine whether the guidance algorithm is used or not. At  $t = 0$ , this guidance step is always taken, and at other time steps the inequality in 9.40 is treated as a logical statement whose true result will mean that the guidance algorithm solution is computed.

$$t_{gui} + \frac{1}{f_{gui}} \leq t_{sim} + \frac{1}{f_{sim}} \quad (9.40)$$

At any time other than  $t = 0$ , if a new guidance solution is not generated, a stored guidance solution will be interpolated for based on the current time  $t$ . Each time a solution is generated, it replaces the previous (if existing) solution. SOCP2 solutions prioritize over SOCP1 solutions, but only for the same guidance step.

If the algorithm determines that a guidance step is to be made, it prepares the problem by initially trajectory discretization and sets up the vector  $q$  needed as part of the Big-M reformulation to determine burn periods. Each guidance step is fed with updates dynamic inputs, which are the current RLV position and velocity in LP frame coordinates, as well as the previous solution's state history and thrust profile, as well as the corresponding time vector, used in case the solution to SOCP1 is infeasible for the given time step. The current mass of the vehicle  $m$  is also provided. The burn time periods of the vehicle is also specified  $t_{b,i}$ , where the first value of the index  $i$  indicates the end of the first burn, and subsequent values alter between starts and stops unless the final burn lasts until the final time  $t_f$ . This subscript is used interchangeably with  $t_1$  and  $t_2$  which also indicate the end and beginning times of the first and second burns respectively. The drag coefficient  $C_d$  of the vehicle is computed by calling the aerodynamic database stored and interpolating for  $\alpha$  and

Burn Schedule $\mathcal{T}_p$ Approach	Controller build count	Software
If statement	$i_{gui}$	CVX, YALMIP
Big-M	$i_{gui}$	CVX, YALMIP
	1	YALMIP
Pre-built controllers	All unique $q$ vectors	CVX (undocumented), YALMIP

Table 9.1: Burn schedule characterization for SOCP problems based on number of times controllers require formulation

$M$ . The atmospheric density  $\rho$  is computed based on the chosen atmospheric model which are detailed in section 9.3. Lastly, the final time is indicated to complement the definition of  $q$  with the burn periods  $t_{b,i}$ . Following trajectory discretization, a logical check is made on whether the guidance time  $t_{gui}$  is 0. As mentioned, both the simulation time and guidance time match at the start of the simulation, so that guidance solution is always calculated at the start. If true, then the t0 variation of the SOCP controllers will be used. As mentioned, a different controller for the initial time needs to be used in order to calculate the initial thrust vector, whilst the standard SOCP1 and SOCP2 controllers will require it as an input from the previous guidance step solution. Once the controller has been identified, the solution, with the respective dynamic inputs is calculated. If the SOCP1t0 controller returns an infeasible solution, the RTLS problem is handled as infeasible and therefore the simulation is terminated. If feasible, and also in the case of SOCP1, the altitude threshold check indicated in eq. 9.41 is made.

$$h \geq h_p \quad (9.41)$$

If true, an index  $i$  is initialized to 1 which will correspond to the index representing the number of successive convexifications made +1. At  $i = 1$ , no successive convexifications have been made, and therefore the first lossless convexification solution will be generated by introducing the drag term using the dynamic inputs of  $\rho$  and  $c_D$ , as well as the state history of the SOCP1 (or SOCP1t0) solutions. If feasible, the stored solution is updated, and the index  $i$  is increased by 1. This loop is repeated until  $i$  exceeds  $N_p + 1$ . If the solution is infeasible at any index  $i$ , the most recent stored solution will be interpolated and used in the simulator.

### 9.2.3 Burn Schedule Definition

The burn schedule can have a considerable impact on the feasibility of the solution. There is a limitation however in the approach in [54], whereby the software employed, CVX, to construct the SOCP problem, requires building of the problem/controller and cannot take in so called 'dynamic' inputs, which vary between each  $t_{gui}$  step. The example of what can be deemed dynamic inputs are the initial states  $\mathbf{r}_0$ , velocity  $\mathbf{v}_0$ , mass  $m$ , initial time  $t_0$ , final time  $t_f$ , the burn schedule array  $q$  and additionally for SOCP2 the approximated drag acceleration  $d^*$ . Static inputs are variables that are used to tune the controller, which are the thrust history difference weight factor  $\mathcal{W}_{\eta_w}$ , altitude threshold  $h_p$ , number of discretization points  $N$ , number of successive convexifications  $N_{cvx}$ , maximum thrust rate  $\dot{T}_{max}$ , overall specific impulse  $I_{sp,oa}$ , maximum thrust direction angle  $\theta_{max}$ , and minimum and maximum thrust  $T_{min}$  and  $T_{max}$ . For this study these 'static' inputs, as the name suggest, are open-loop variables and do not change during the trajectory.

YALMIP is another software package for MATLAB that allows creation of SOCP problems. Within the toolbox, there is an option to use an 'optimizer' structure to set up a problem as a controller which can be saved and then called to solve the problem with different inputs. This can save time for DoE's as the controller only needs to be generated once. This type of approach also exists in CVX however it is not a documented procedure and little to no support or examples exist for its application. As a result, it was not explored in this study. In the original DESCENDO study, CVX used if statements to declare different constraints for different steps in the discretized grid. We note that these constraints are switched on based on the  $\mathcal{T}_p$ , and in programming sense are the 1's (thrust on) or 0's in the input  $q$ . Table 9.1 shows the different approaches identified for this study in generating the controllers.

## 9.3 Verification

In this section verification procedures to determine that the algorithm runs as intended are detailed. These verification procedures were run as independent tests from the DoE's of the 3-DoF simulation discipline. The verification tests ran involve varying  $h_p$ , as well as varying the atmospheric models between the DG (plus ACT) and SDK blocks to determine the suitable model selection. For these tests, the feasibility of the two

Block	Variable	Value
SDK	$T_{min,M10}$	10kN
	$T_{min,MX}$	60kN
	$T_{max,M10}$	100kN
	$T_{max,MX}$	600kN
	$f_{sim}$	10Hz
DG	$\theta_{max}$	70deg
	$\mathcal{W}_{\eta_w}$	.0001
	$T_{min}$	130kN
	$T_{max}$	1300kN
	$f_{gui}$	1Hz
	$\dot{T}_{max}$	1kN/s

Table 9.2: Nominal parameters for SDK (spacecraft dynamics &amp; kinematics) and DG (descent guidance) block.

Trajectory	$\ r_f\ $ (m)	$v_{N,f}$ (m/s)	$v_{E,f}$ (m/s)	$v_{D,f}$ (m/s)
Pre-generated $\mathcal{T}_p$ , $t_f = 300$ , $\%t_1 = .1$ , $\%t_2 = .6$	0.007915	0.003206	0.001491	0.406295
Pre-generated $\mathcal{T}_p$ , $t_f = 280$ , $\%t_1 = 0$ , $\%t_2 = 0$	0.030835	0.003101	-0.00246	0.57546
Big-M $\mathcal{T}_p$ , $t_f = 280$ , $\%t_1=0$ , $\%t_2=0$	0.122493	0.00676	1.313176	1.21859

Table 9.3: Verification trajectories for different types of  $\mathcal{T}_p$  and methods of generation.

different YALMIP controller optimizer approaches is analysed (Big-M and controller pre-generation), as well as the adherence of the thrust profiles to the two difference ACT schemes formulated (prioritize  $P_a$ , prioritize running engine). The parameters used for the SDK and DG blocks are show in table 9.2. Additional tests where run on  $\dot{T}_{max}$  but results showed an overwhelming evidence that the nominal value of 1 kN/s as used in [54] faired much better as values in the search space from .5 to 100kN/s all resulted in infeasible results. As such this parameter was not studied further. Two different burn schedules and  $t_f$  where considered for both Big-M and the pre-generated controller methods. The first criteria was  $t_f = 300$  sec, with  $\%t_1 = .1$  and  $\%t_2 = .6$ . This criteria was chosen with hindsight after completing partially some runs from DoE1. The second trajectory selected was with  $t_f = 280$  sec, and no strict burn schedule ( $\%t_1 = 0$ ,  $\%t_2 = 0$ ). The final time was tested in grid search from 250 seconds to 350 seconds with increments of 10 seconds. It was determined that from 250-260 sec the solution was infeasible at  $t = 0$ , whilst at 270 it was feasible, and 280 was the first point of optimality.

### 9.3.1 Burn Schedule Approach

The trajectory profile with the strict burn schedule for the big-M approach fails to find a feasible solution with the SOCP1t0 algorithm. Other tests are ran for  $t_f$  in the range of 280 to 400 with increments of 10, all yielding infeasibility at  $t = 0$  or largely infeasible solutions with  $\|r_f\|$  and  $\|v_f\|$  in the tens of thousands of meters and hundreds of m/s respectively. The controller fails to find optimal solutions soon after the first few guidance iterations. This indicates that there is an inherit problem with the way that the constraints are rewritten in the big-M approach for burn schedules. A possible reason for this may be the fact that the big-M problem would need to have constraints with discrete and non-smooth profiles for the bounds.

The rest of the results for both the big-M approach and pre-generated controllers is shown in tab. 9.3. The results show that all trajectories return optimal values, except for the big-M trajectory which yields a slightly less than optimal value for velocity. The feasibility of the big-M approach without a strict burn schedules reinforces the idea that the binary type constraints on the problem may impose numerical problems far too great for ECOS to solve. This reasoning could also be applied to the fact as to why the choice of  $\dot{T}_{max}$  is extremely sensitive.

The trajectories are plotted and shown in fig. 9.1. The following insights can be made from the figure. The big-M approach without a strict burn schedule seems to have the most direct approach to the LP, in the sense that it does not overshoot the longitude. The ground track profile shows however that there is more crossrange (latitude) variation compared to the pre-generated approach.

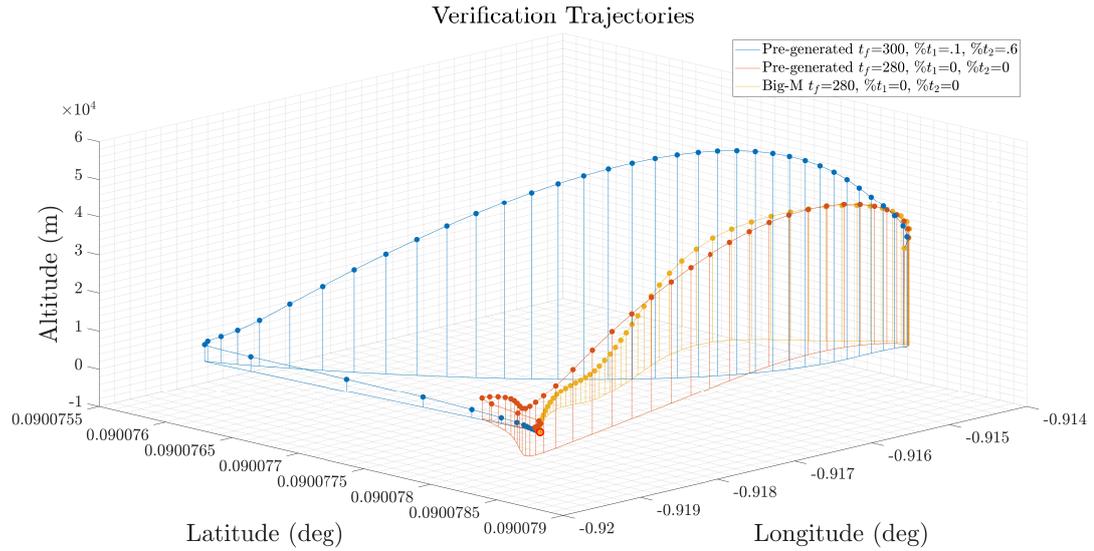


Figure 9.1: Verification trajectories. The points on the trajectory are selected so as to indicate a 6 (for  $t_f = 300$  sec) or 5.6 (for  $t_f = 280$  sec) second interval between the last point.

In all cases, what can be noticed is that at the point of approach there seems to be more time taken (indicated by the increasing number of points which denote 10 second intervals on the trajectory curves). And exception to this is for the pre-generated unrestricted burn profile which shows a slow down at the lowest longitude point, where an angle is formed to redirect the RLV towards the LP. A rapid thrust profile is followed, which then ultimately leads to a slower and more controlled thrusts towards the LP.

The pre-generated restricted burn schedule trajectory (blue) shows a different strategy to reach the LP. For the majority of the flight, the points are somewhat equally distributed, up until where a maximum latitude point is reached. Interestingly, this trajectory favours to distance itself respective to the crossrange of the LP and redirected itself to land laterally. A rapid burn sequence is then followed after redirection, and ultimately more controlled burns near the LP.

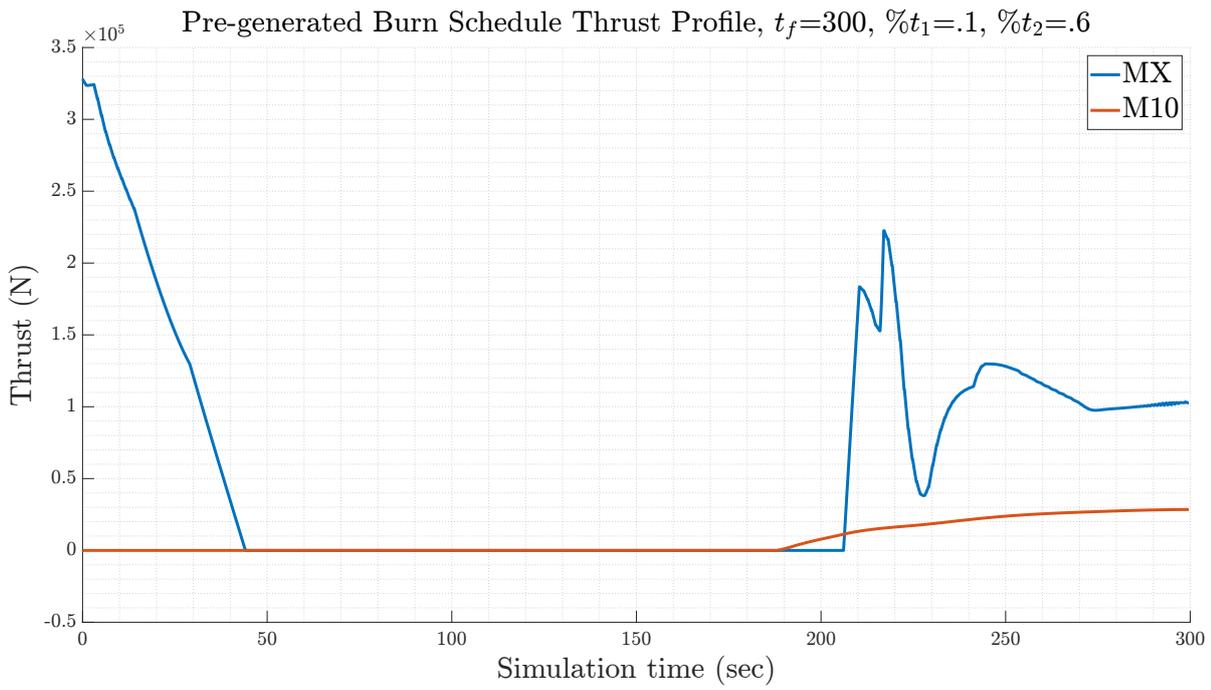


Figure 9.2: Thrust profile of pre-generated burn schedule approach for  $t_f=300$  sec,  $\%t_1=.1$ , and  $\%t_2 = .6$ .

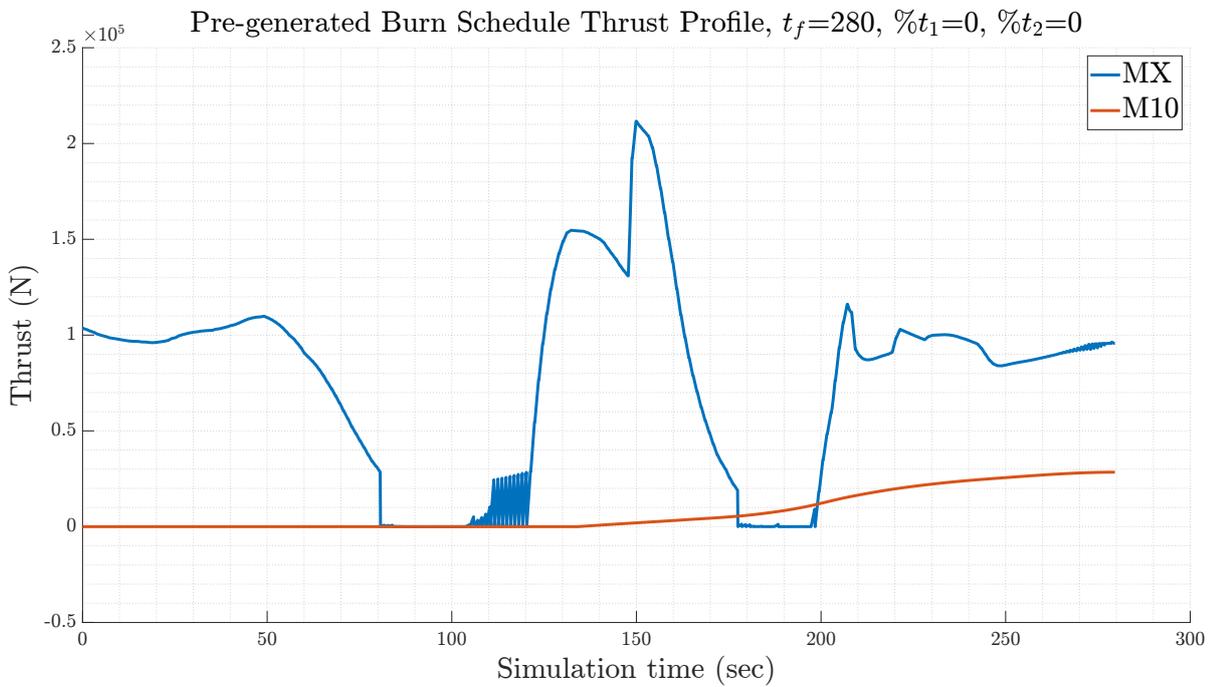


Figure 9.3: Thrust profile of pre-generated burn schedule approach for  $t_f=280$  sec,  $\%t_1=0$ , and  $\%t_2=0$ .

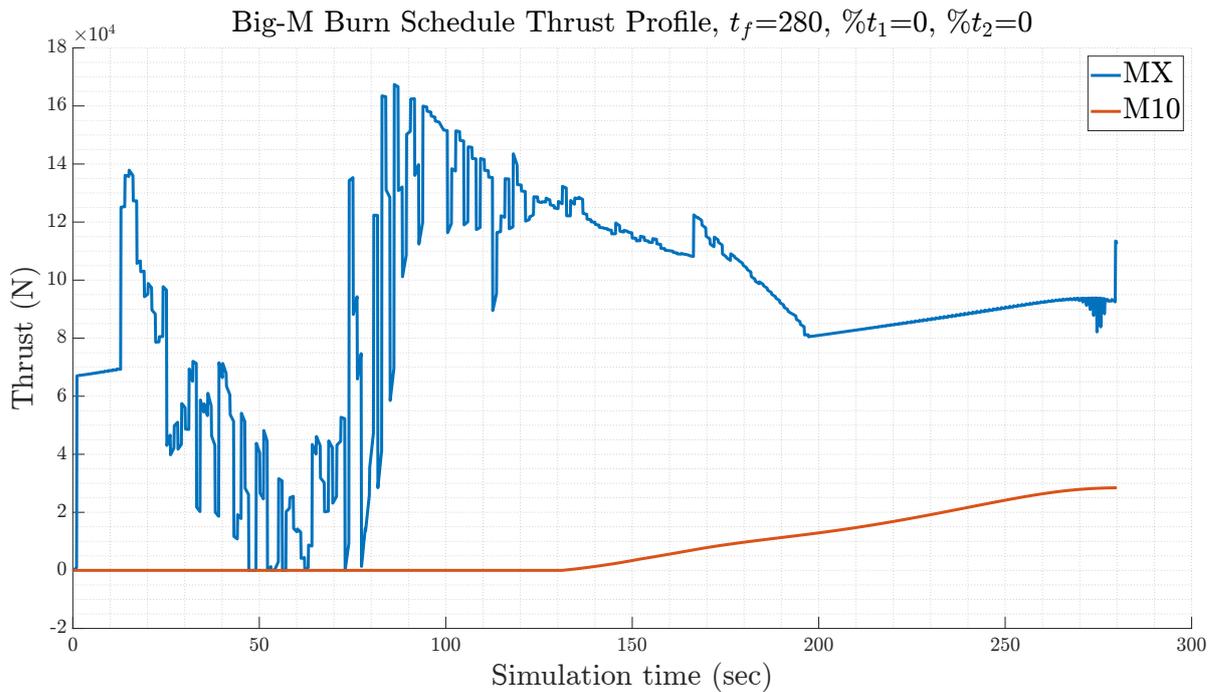


Figure 9.4: Thrust profile of big-M approach burn scheduling for  $t_f=280$  sec,  $\%t_1=0$ , and  $\%t_2=0$ .

The reasoning as to why the different trajectories with different controller burn scheduling approaches have different strategies for return may be answered by looking at the thrust profiles. Fig. 9.2 to 9.4. Fig. 9.4 shows a non-smooth thrust profile for the MX engine, with a smooth M10 profile that switches on at 130 sec. It can also be seen that the MX engine does not switch on immediately at the start of the simulation, which when looking at the diagnostics shows that the first few guidance iterations fail to find a feasible solution. The thrust rate constraint in SOCP1 and SOCP2 is not respected as well, indicating that the big-M constraints are not representative of the original problem. Between  $t = 5$  and  $t = 120$  a large majority of the SOCP1 guidance solution are solved with numerical problems or fail to find a solution. Past the point the solutions begin to improve, which is followed by a more smoothed out thrust profile. Despite the lack of adherence to  $\dot{T}_{max}$ , the maximum thrust the MX achieves is only around 162kN.

In comparison, the pre-generated burn schedule approach produces a much smoother thrust profile as shown in fig. 9.4. There are three particular points of non-smoothness, one being at an engine switch on at around 105 sec up until 120 sec. The other is at 178 sec to 200 sec, and the last the final 15 seconds before landing. No particular pattern related to the diagnostics of the solutions reflects this behavior, aside from the former of the three points where SOCP2 solutions result in feasible with numerical problems. Nevertheless, three clear burn regions can be identified:  $t = 0$  to  $t = 80$ ,  $t = 120$  to  $t = 178$ , and  $t = 200$  to  $t = 280$ . This result is favourable because it shows that the algorithm is capable of producing a smoothed thrust profile, and also shows the contrast between what can be achieved with analytical guidance law such as ZEM/ZEV whereby a zero thrust acceleration is possible for multiple instances of time.

Fig. 9.2 shows the thrust profile for the pre-generated burn schedule with fixed  $t_{b,i}$ . This result shows no non-smoothness apart from a very small degree towards landing. This is expected as near landing the algorithm is more likely to vary its thrust due to the way that the discretized grid is set up, such that as the  $t_{gui}$  approaches  $t_f$  the time step size decreases. Interestingly, the control allocation scheme allocates the thrust initially only to M10 at  $t_{b,2}$ . This is in contrast to all trajectories where for  $t_{b,1}$  only the MX is firing. At the initial point of the trajectory, based on the nominal control allocation scheme shown in fig. B.1, both the MX and M10 are favoured to thrust. It seems however that the fmincon solution favours only the switch on of the MX engine, despite the fact that firing the M10 engine would require less mass by virtue of its higher  $I_{sp}$ . Nevertheless the initial thrust as determined by the algorithms are all greater than  $T_{M10,max}$ , safe for the big-M approach. The level of controllability and smoothness as shown in the pre-generated trajectories is also therefore in partial merit to the control allocation scheme, which does not switch between usage of MX and M10 engines when crossing the  $T_{M10,max}$  (100kN) threshold. This behaviour however is not expected and presents an opportunity to revisit the fmincon problem that minimizes the mass flow rate. Nevertheless,

Trajectory	$\ r_f\ $ (m)	$v_{\{N, f\}}$ (m/s)	$v_{\{E, f\}}$ (m/s)	$v_{\{D, f\}}$ (m/s)
Pre-generated $\mathcal{T}_p$ , $t_f = 300$ , $\%t_1 = .1$ , $\%t_2 = .6$	0.073674	0.005404	-0.00244	0.859375
Pre-generated $\mathcal{T}_p$ , $t_f = 280$ , $\%t_1 = 0$ , $\%t_2 = 0$	0.032975	0.003059	-0.00026	0.581003
Big-M $\mathcal{T}_p$ , $t_f = 280$ , $\%t_1=0$ , $\%t_2=0$	0.09013	0.008428	0.753591	1.330494

Table 9.4: Verification trajectories for different types of  $\mathcal{T}_p$  and methods of generation, with the minimum thrust constraint  $T_{min}$  as 10% of the maximum thrust of each engine.

the set burn schedule with pre-generation of controller strategy shows the most favourable outcome out of all three approaches.

A trend can be noticed between these three trajectories, in the context of controllability and resulting trajectory. Whilst the big-M approach does not adhere to the SOCP problem's constraint, it can be deemed to have the highest level of controllability out of all three solutions. By having a control profile that is more adjustable, the maximum thrust achieved does not need to be as high. In contrast, introducing a working  $\dot{T}_{max}$  constraint returned a much smoother thrust profile, with identifiable regions of  $t_{b,i}$  despite the fact that none were fixed. Moreover, with the introduction of fixed  $t_{b,i}$  the thrust profile eventually smoothed out. This result shows the importance of the  $\dot{T}_{max}$  parameter and can provide more evidence to support the difficulty in finding multiple working values that allow the algorithm to work. It was also deemed important in making DESCENDO work in [54]. In short, increase level of control does not indicate a reduced quality of the solution, but rather can help validate the algorithm to its real-world application. Whilst it is true that REQ-PR-1 has a  $\dot{T}_{max}$  of 50% which is far beyond what any of the profiles show to reach (not including switch on or offs), for the validation of the guidance algorithm it is important to show smoothness of the thrust profile to add an increased level of predictability of the solution and better translation to the actual capabilities of the engines and actuators of a physical RLV. Moreover, when subjecting the algorithm to noise, it is unfavourable to deal with an irregular thrust profile, as at a point of infeasibility, the SOCP problems may have more difficulty determining the next feasible solution by virtue of the fact that neighbouring solutions themselves already have a large discrepancy between them. This added level of controllability also explains the direct approach that the big-M trajectory has compared to the other. Intuitively, it may be favourable in fact to approach the LP with some negative downrange or crossrange so as to allow space to make final manoeuvres rather than approach the target too rapidly, risking an overshoot from the control that may cause the solution to become infeasible due to the last few seconds of thrust. This is especially true when we impose a restriction of  $T_{min}$  which is described in the next section.

### 9.3.2 Minimum Thrust Constraint

In order to avoid the presence of non-smooth regions of thrust, especially lower values near  $T = 0$ , we impose a  $T_{min}$  condition on the thrusters of 10% of their maximum thrust. The results for the verification trajectories with this condition are shown in tab. 9.4. All results show feasibility, with optimality for velocities for the pre-generated methods, as well as their  $\|r_f\|$ . A slight drop off of  $v_{E,f}$  in the big-M approach can be noticed, albeit with an equally slight increase of its  $v_{D,f}$ .

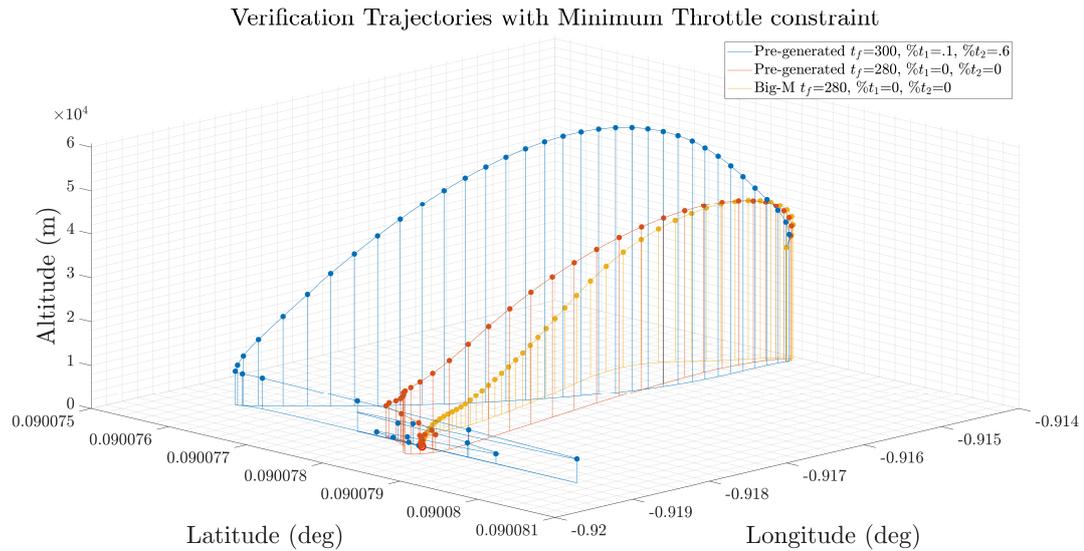


Figure 9.5: Verification trajectories with the added  $T_{min}$  constraint on DESCO. The points on the trajectory are selected so as to indicate a 6 (for  $t_f = 300$  sec) or 5.6 (for  $t_f = 280$  sec) second interval between the last point.

The improved final condition may be attributed to the fact that as discussed before, the pre-generated  $\mathcal{T}_P$  trajectories distance themselves from the LP during the landing to allow for some degree of correction, whilst the big-M approach rapidly approaches it. This can also be seen in this scenario with the added  $T_{min}$ , as shown in fig. 9.5. The pre-generated  $\mathcal{T}_P$  approach fixed fixed  $t_{b,i}$  shows an interesting deviation from its counterpart without  $T_{min}$  constraints. The trajectory seems to oscillate across the latitude point of the LP, which varies a maximum of  $4e-6^\circ$ , roughly 3.9m if we convert this distance with eq.4.19. This presents an interesting scenario where the algorithm essentially hovers above the LP until it can arrive at the designated  $t_f$  with the right conditions. This reinforces the idea that with reduced controllability and set bounds the solution can improve in accuracy. In fact, even in this case both pre-generated trajectory share a better solution that the big-M approach which again favours a direct landing without distancing itself during the final moments. An advantage of this may be presented in the context of safety of operations when landing. If the algorithm favours a landing approach that takes up more distance it may be a safety hazard in the event that a failure occurs. However, in this case a large deviation from the LP coordinates at the point of landing is not present, with the larger distance being covered by the red curve of a few hundred meters west of the LP.

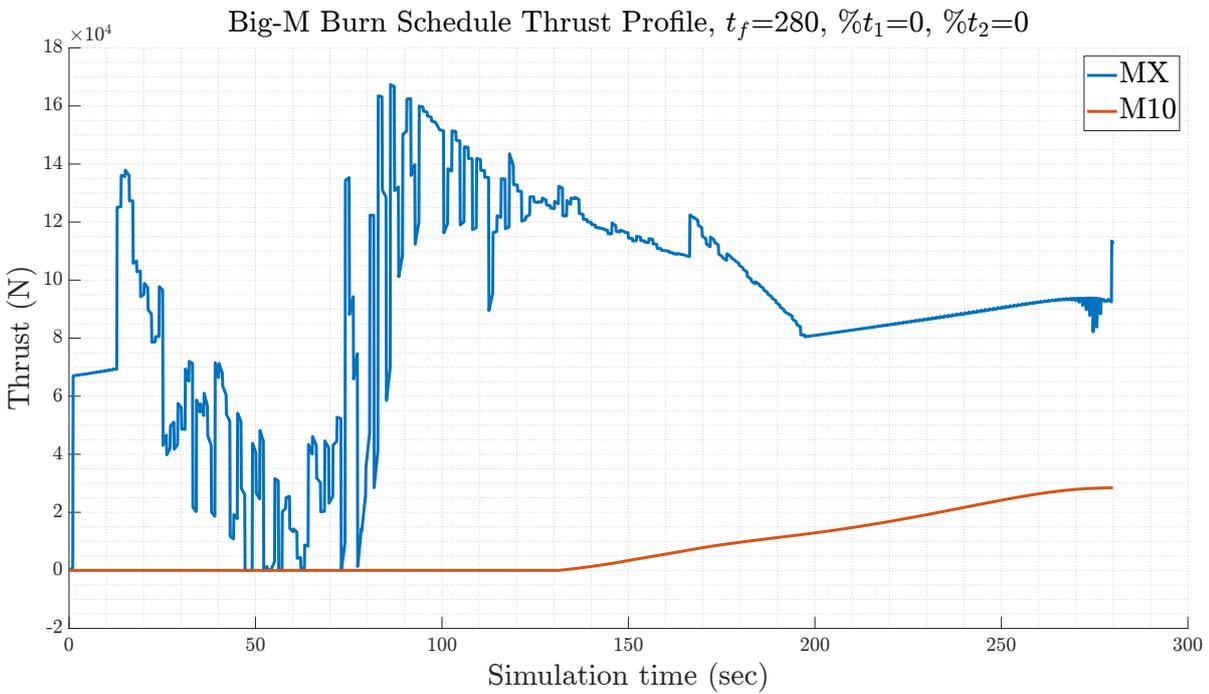


Figure 9.6: Thrust profile of big-M approach burn scheduling for  $t_f=280$  sec,  $\%t_1=0$ , and  $\%t_2=0$ , with added  $T_{min}$  constraint on DESCO.

Fig.9.8 shows the thrust profile of the big-M approaches with added  $T_{min}$  constraint. The profile now shows a switch a on of the M10 in the earlier stages of flight, and non-smooth control for both engines, until the final 120 seconds where the M10 retains its smoothness. The  $t_f$  had to be increased to 300 sec, with 290 sec providing an infeasible solution. Overall the profile shows an undesirable type of control whereby translating it to a real world actuator system may not be practical.

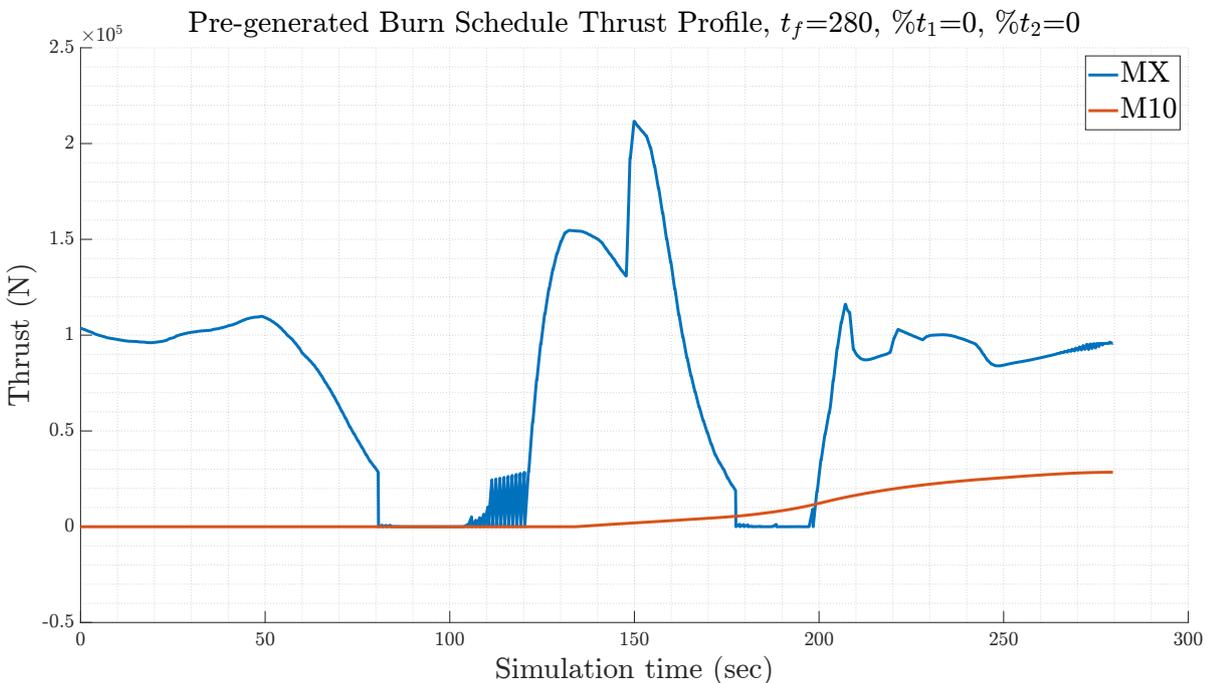


Figure 9.7: Thrust profile of pre-generated burn schedule approach for  $t_f=280$  sec,  $\%t_1=0$ , and  $\%t_2=0$ , with added  $T_{min}$  constraint on DESCO.

Fig. 9.7 shows the pre-generated  $\mathcal{T}_p$  without set  $t_{b,i}$ . The solution does shows adherence to the  $T_{min}$

constraint, also noted by the fact that its diagnostics return no infeasible solutions. The profile shows that the control allocation scheme can distribute lower levels of thrust to the M10, whilst allowing the MX to cover the remaining thrust, as well as understanding when to switch on or off either engines. The difference between its counterpart without  $T_{min}$  is in that several switch on and off of the MX and M10 are noticeable. Increasing the amount of restarts for a liquid rocket engine may increase the amount of complexity of the design if pyrotechnic igniters are used, and requires further analysis through reliability testing[70].

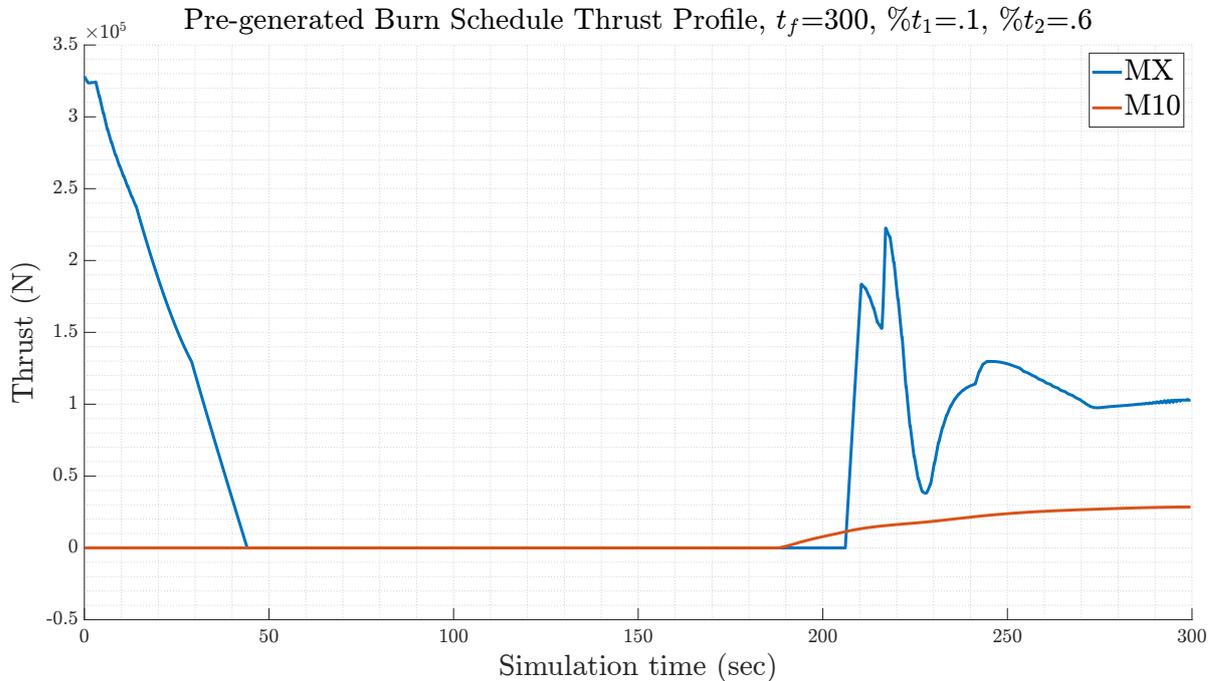


Figure 9.8: Thrust profile of pre-generated burn schedule approach for  $t_f=300$  sec,  $\%t_1=.1$ , and  $\%t_2=.6$ , with added  $T_{min}$  constraint on DESC0.

Fig. 9.8 shows the pre-generated  $\mathcal{T}_p$  trajectory with fixed  $t_{b,i}$ . Several restarts for the MX (4) and even more so for M10 (10) are noticeably for  $t_{b,2}$ . The number of restarts does not present itself an issue but the time span in which they occur, between  $t = 225$  sec to  $t = 265$  sec is fairly brief. Whilst the control profile itself is smooth in periods of switch on, and adheres to all thrust constraints, rapid switch on and off can also present issues when translating the control to a physical actuator control system in case noise is added. In contrast with fig. 9.6 and 9.4, the overall thrust in this region however is a smooth profile, and it is the control allocation scheme that experiences issues due to the variations of  $P_a$  when entering lower altitudes. This causes a change in priority between engine should be switched on. Ultimately however, the landing stage shows a return to a more smoothed distribution of thrust across the two engine types. This thrust profile also shows the highest reached thrust for the MX engine out of the previous verification tests, and reinforces the call for higher thrust values when more constraints are applied. The SOCP solutions are all deemed feasible for both SOCP1 and SOCP2 albeit a few between  $t_{b,1}$  and  $t_{b,2}$  that return infeasible. Nevertheless, the first feasible solutions at  $t_{b,2}$  delay the thrusting of the engines for about 25 seconds ( $t_{b,2}=25$ ). This is again evidence to show that the algorithm can favour a coasting period before re-ignition.

### 9.3.3 Computation Time

Table 9.5 shows the overall computation times needed for the verification trajectories without set  $t_{b,i}$ 's. The big-M approach has an overall much smaller preliminary time to generate the controllers (4 in total: SOCP1t0, SOCP1, SOCP2t0, SOCP2). The average time taken however is noticeably greater than the pre-generation approach, for both SOCP1 and SOCP2. This strategy allows more effective execution of large scale DoE's, but also for physical feasibility of the mission when testing the delay due to a simulated processing time to compute the problem covered in the sensitivity analysis in section 8.2.4.

Approach	Preliminary Time over 3 runs (sec)		Average time per SOCP 1 (sec)	Average time per SOCP2 (sec)
Big-M	4.7210		.1187	.0458
Pre-generation approach $N = 20$ , $t_f = 280$ sec, $\%t_1=0$ , $\%t_2=0$	Controller Build	Controller Load	.0729	.0227
	423.8331	75.6543		

Table 9.5: Controller pre-load and build times for different burn scheduling approaches.

### 9.3.4 Atmospheric Model Selection

For selection of the nominal atmosphere models, the pre-generated unrestricted  $t_{b,i}$  trajectory is ran with all possible combinations of guidance and simulator atmospheric model choices. These are shown in table 4.2, where the LF models are used for the guidance algorithm, and the HF for the simulator. The results for the variation of states can be seen in fig. 9.9.

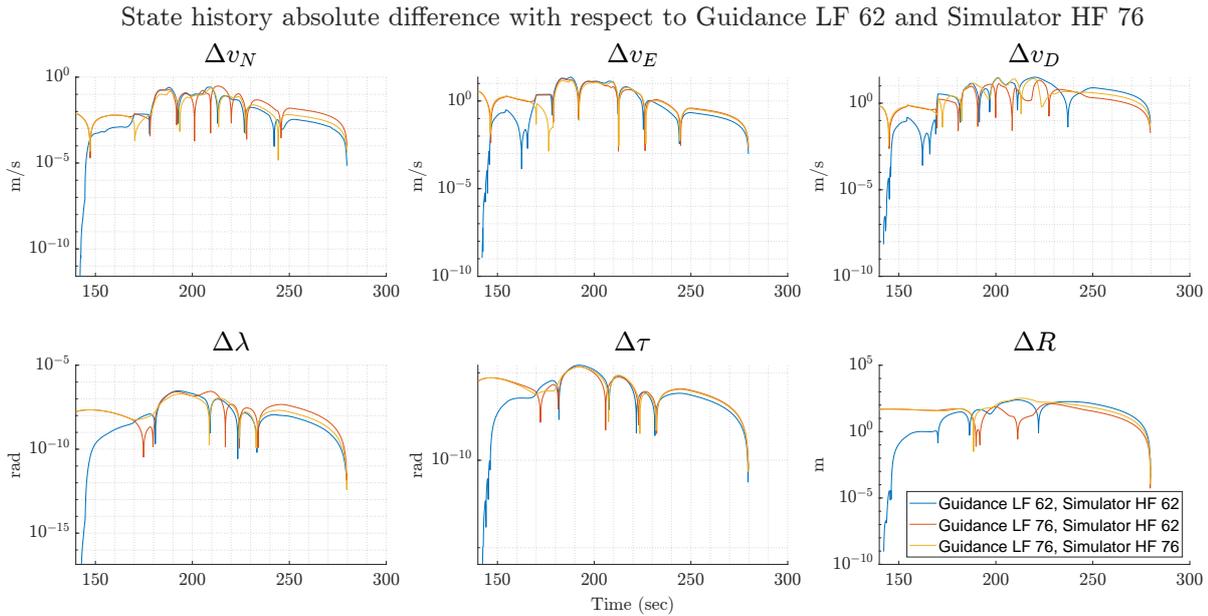


Figure 9.9: State history comparison for pre-generated burn schedule approach for  $t_f=280$  sec,  $\%t_1=0$ , and  $\%t_2 = 0$ , using the LF 62 and HF 76 atmospheric models for guidance and simulator respectively.

The figure shows that compared to the LF 76 and HF 76 selection, there is a consistent variation of the states since  $t = 0$ , which has been cut off from the graph because previous values are less than 1 or 2 orders of magnitude for all states than what is visible in the given time range. The purpose of the graph is to determine whether a selection of models results in a larger discrepancy in states compared to other model selections. The results do not show a noticeably difference in state histories.

The LF 76, and HF 62 selection shows the most difference for the majority of the trajectory except for  $v_D$  and  $R$ . The LF 76 HF 76 model sits somewhere in between the two.

The models do not impact the feasibility of the trajectory, as the dips at 280 seconds, the value of  $t_f$ , indicate the differences start decreasing. The latitude  $\lambda$  shows at most a peak of a few 10's of meters of variation at around 180 sec to 210 sec. The most noticeably differences are for  $v_D$  and  $R$ , where up to 231 m of difference is noticeably at around 220 sec when comparing the nominal model to LF 76 and HF 76. The former shows a maximum peak of around 33 m/s at 220 sec. When looking at only these two most influential differences across all state histories, the hierarchy of difference between models matches with what is expected, from 240 sec to 280 sec. This shows the blue curve on top of the yellow, which is on top of the red. With the nominal model being LF 62 and HF 76, switching the SA 1962 for the COESA 1976 models with the LF 76 and HG 62 shows the lowest state difference for  $v_D$  and  $R$ , which is the opposite in this time range for the other states, which is the expected outcome, as a direct switch in models for guidance simulator should represent the greatest state history difference. It is possible that as  $v_D$  and  $R$  are variables that are directly impacted by the variations in the model selection due to the fact that all models are functions of the altitude and that  $\dot{R} = -v_D$ . The downwards velocity may build up greater differences compared to the LF 62 and HF 76. This effect then carries on to  $R$  as it starts to follow opposite the trend of the other states because of its state equation. This

effect then snowballs into  $v_D$  as its state equation as given in 4.10 is related to  $R$ , and as such these two states follow the opposite trend in state differences with respect to the nominal atmospheric model.

In short, the results show the expected behaviour in differences in thrust history based on the qualitative fidelity of the models. This is an exception for  $v_D$  and  $R$ , possibly due to the proportionality of  $\dot{R}$  to  $v_D$ . The nominal selection of LF 62 for the guidance algorithm and HF 76 for the simulator is kept and used for all simulations.

# Chapter 10

## Results

This chapter presents all the results of the three DoE's and the disciplines of Launch Vehicle Optimization, Engine Sizing, and preliminary ascent & descent. For each DoE campaign and discipline, the motivation for the selection of nominal candidates to pass on to the next disciplines is explained.

### 10.1 Launch Vehicle Optimization

The results of the launch vehicle optimization procedure return a total of 1499 final candidates after 500 generations which respect all requirements for all payload classes. Tables 10.1 to 10.5 show the measures of central tendency results for the various payload classes. All but the 500 kg payload class runs returned 300 feasible candidates, which returned 299 instead. All payload classes cover the entire span of possible  $m_{p,1}$  except for 500 kg, which has the lowest minimum of 58371.27 kg. There is also quite a noticeable difference between the minimum and the rest of the quantiles for all payload classes for  $m_{p,1}$ . From a jump of 400 to 500kg, there is an increase in the minimum propellant of 8371 kg. The value of 400kg also presents an interesting case for the minimum value of  $m_{dry,1}$ , where the greatest payload increases the minimum mass by around 823.66 kg. This is two order of magnitudes different than the differences between the smaller payload classes. Moreover, 400kg presents the largest  $\sigma$  for all variables included in the table. The usefulness of selecting a nominal candidate for in this class is that when examining the variation of propellant mass for the sensitivity study, there may be an added degree of reliability that the increase or decrease in propellant mass may correspond to feasible configuration. This translates to both the variation in  $m_{dry,1}$  and  $m_{p,1}$ , as with a 3-DoF simulation, the final mass can correspond to any combination of dry and propellant mass, as long as it greater than the former. This is also true for the fact that the 400 kg class contains candidates feasible for the maximum and minimum propellant mass requirements.

The smaller variation with respect to  $\mu$  is found in  $m_{pay}$ , for all payload classes. This result is desirable as the added payload mass apart from the instruments itself should not be too dependent on the inputs to the launch vehicle optimization problem, aside from  $m_{pl}$  itself. Another small relative difference in % comes from  $m_{dry,2}$ . There is statistical evidence to show that the effects of mass variation can be focused on  $m_{dry,1}$  and  $m_{p,1}$ . As an example, for  $m_{pay} = 100$ ,  $\sigma$  for  $m_{dry,2}$  corresponds to roughly 4.5145 % of  $\mu$ . For  $m_{p,1}$ , it is 6.3029%, for  $m_{pay}$  it is 1.9675 %, and for  $m_{dry,1}$  it is 7.2825 %.

Variables	$\mu$	$\sigma$	CI		Quantile				
			2.5%	97.5%	0	.25	.5	.75	1
$m_{dry,1}$ (kg)	7606.731	553.9627	7543.791	7669.672	5381.787	7709.429	7801.191	7844.051	7869.603
$m_{dry,2}$ (kg)	1146.725	51.76849	1140.843	1152.607	931.1554	1152.119	1165.569	1172.128	1176.243
$m_{p,1}$ (kg)	68701.69	4330.192	68209.7	69193.68	50000	69973.13	69992.53	69997.13	69999.99
$m_{pay}$ (kg)	443.6766	8.729242	442.6848	444.6684	408.5555	445.2785	446.7338	447.4155	447.8241

Table 10.1: Measures of central tendency for GA results for 100kg payload class RLW.

Variables	$\mu$	$\sigma$	CI		Quantile				
			2.5%	97.5%	0	.25	.5	.75	1
$m_{dry,1}$ (kg)	7702.689	576.0237	7637.242	7768.136	5340.135	7819.859	7910.236	7950.439	7978.701
$m_{dry,2}$ (kg)	1162.021	53.01285	1155.998	1168.044	951.8506	1168.65	1182.112	1188.457	1193.066
$m_{p,1}$ (kg)	68763.25	4097.879	68297.66	69228.85	50000	69976.95	69992.23	69997.01	69999.91
$m_{pay}$ (kg)	550.3822	9.024853	549.3568	551.4076	512.9626	552.2009	553.6305	554.2702	554.721

Table 10.2: Measures of central tendency for GA results for 200kg payload class RLV.

Variables	$\mu$	$\sigma$	CI		Quantile				
			2.5%	97.5%	0	.25	.5	.75	1
$m_{dry,1}$ (kg)	7520.461	522.078	7461.144	7579.779	5297.128	7610.85	7725.32	7769.16	7795.273
$m_{dry,2}$ (kg)	1135.544	46.09642	1130.307	1140.782	943.2143	1137.401	1154.532	1161.304	1165.763
$m_{p,1}$ (kg)	68695.17	3904.478	68251.55	69138.79	50000	69967.87	69991.03	69996.65	69999.99
$m_{pay}$ (kg)	652.7006	8.167361	651.7726	653.6286	617.3672	654.075	655.9034	656.6021	657.0226

Table 10.3: Measures of central tendency for GA results for 300kg payload class RLV.

Variables	$\mu$	$\sigma$	CI		Quantile				
			2.5%	97.5%	0	.25	.5	.75	1
$m_{dry,1}$ (kg)	7563.23	612.7882	7493.606	7632.854	5254.154	7672.336	7704.111	7873.786	7997.259
$m_{dry,2}$ (kg)	1173.643	67.34537	1165.991	1181.294	928.8036	1185.169	1200.974	1206.294	1210.752
$m_{p,1}$ (kg)	68539.85	4471.482	68031.81	69047.89	50000	69971.82	69992.66	69997.49	69999.99
$m_{pay}$ (kg)	759.0157	9.77548	757.905	760.1264	721.7938	761.0568	761.5607	763.4987	765.4495

Table 10.4: Measures of central tendency for GA results for 400kg payload class RLV.

Variables	$\mu$	$\sigma$	CI		Quantile				
			2.5%	97.5%	0	.25	.5	.75	1
$m_{dry,1}$ (kg)	7603.646	381.0357	7560.28	7647.012	6078.02	7650.484	7744.883	7786.385	7820.872
$m_{dry,2}$ (kg)	1194.148	53.68117	1188.039	1200.258	1016.899	1200.197	1215.388	1221.794	1227.404
$m_{p,1}$ (kg)	69259.6	2355.019	68991.58	69527.62	58371.27	69969.6	69992.02	69996.79	69999.99
$m_{pay}$ (kg)	865.1837	6.15545	864.4832	865.8843	840.7916	865.9689	867.4835	868.1445	868.6969

Table 10.5: Measures of central tendency for GA results for 500kg payload class RLV.

## 10.2 Engine Sizing

The results of the engine sizing procedure are shown in tab. 10.7, and 10.6. The values of  $D_e$  are lower for all measures of central tendency values and quantiles for the M10 engine except for the minimum. The same is true for  $P_e$ . This gives the MX a greater potential in maintaining maximum thrust at lower altitudes if we consider  $P_e$  as a maximal achievable value and assume the engines have altitude compensation abilities. We note that the feasible configurations of the M10 contain a much smaller range of errors for  $I_{sp}$ . Moreover, the confidence intervals for both engines on  $\%I_{sp,error}$  is fairly large. This being 37.5% and 23.5% of  $\mu$  for the MX and M10 engines respectively. The MX also has a wider spread in  $\frac{P_e}{P_a}$  with a 11.6907% difference, and 1.8252% for the M10. For  $P_c$ , the differences from the confidence intervals to the  $\mu$  are 17.38% for MX, and 14.439% for the M10. This is evidence that the sample points of chosen for the MX as inputs to the CEA run problem may not be representative of the entire distribution. The  $\%I_{sp,error}$  variation also supports this hypothesis, and therefore the number of points for the MX case should be increased to obtain a better representation of the engine characteristics. However, the wide distribution of the MX's engine outputs demonstrate that, along with its generally higher  $P_e$ , serves as a suitable candidate to be considered as an engine with an altitude compensating nozzle.

The total number of feasible MX candidates are 42, and the total for M10 are 59. The smaller pool of candidates for MX may also explain the larger disparity in the CI, as a larger sample may reduce this. There is an increasing difference between quantiles of the MX for  $P_e$ . The maximum feasible value results in a 110 kPa candidate, which would cover sea-level altitudes (101.325 kPa). To determine the variation of  $\%I_{sp,error}$  relative to  $P_e$ , we can look at tab. A.2 and A.2. The MX candidates show a decrease in  $\%I_{sp,error}$  as  $P_e$ , whilst the inverse is true for M10. This result is ideal as the MX engines are closer to  $I_{sp,theory}$  with larger  $P_e$  making them an effective choice for near sea-level altitude. Whilst the M10's error increase with higher  $P_e$ , it's shorter variation of error makes up for this.

Variable	$\mu$	$\sigma$	CI		Quantile				
			2.5%	97.5%	0	.25	.5	.75	1
$D_e$ (m)	1.244731	0.694935	1.028175	1.461288	0.589972	0.803284	1.026194	1.426183	3.573345
$P_e$ (Pa)	41006.67	29448.05	31830.01	50183.33	2690	17350	33800	57300	110000
$P_c$ (Pa)	5345881	2981428	4416803	6274959	2690	17350	33800	57300	110000
$\frac{P_e}{P_a}$	0.007553	0.002834	0.00667	0.008436	0.003407	0.005653	0.006245	0.010372	0.011145
$\%I_{sp,error}$	-2.50762	3.017659	-3.44799	-1.56726	-8.06462	-4.44131	-2.77354	0.317805	0.894548

Table 10.6: Measures of central tendency for MX feasible candidates.

Variable	$\mu$	$\sigma$	CI		Quantile				
			2.5%	97.5%	0	.25	.5	.75	1
$D_e$ (m)	1.121321	0.616043	0.960779	1.281862	0.620621	0.726445	0.899157	1.277063	3.154245
$P_e$ (Pa)	5495.532	3224.08	4655.332	6335.732	416.6571	2620.832	5309.014	8165.8	11234.55
$P_c$ (Pa)	5101889	2826726	4365240	5838538	416.6571	2620.832	5309.014	8165.8	11234.55
$\frac{P_e}{P_a}$	0.001041	7.28E-05	0.001022	0.00106	0.000844	0.001007	0.001065	0.001095	0.001138
$\%I_{sp,error}$	-0.25681	0.231872	-0.31724	-0.19639	-0.57579	-0.43302	-0.30915	-0.11878	0.299995

Table 10.7: Measures of central tendency for M10 feasible candidates.

## 10.3 Preliminary Ascent & Descent

The preliminary ascent and descent results are split into the 5 payload classes to help better identify if a suitable candidate exists, and give insight to begin answering the first research question.

### 10.3.1 Preliminary Ascent

The preliminary ascent measures of central tendency are shown in tables 10.8 to 10.12. The CI for all of the variables and payload classes are not much greater than 1% in the worst cases, making it a statistically accurate representation of the distribution of results. This could be to the uniform grid space used for variation of the variables  $k_{h_0,descent}$  and  $t_f$ . The highest relative variation lies in starting descent velocities  $v_{x,0,descent}$

and  $v_{z,0,descent}$ . These are dependent entirely on  $\lambda$  (mass fraction), which is a function of the RLV candidate geometry. When it comes to the variation of  $h_0$  and  $t_f$ , whilst  $t_f$  remains the same for all payload classes,  $h_{0,descent}$  varies with the solution case due to  $k_{h_{0,descent}}$ . The smallest  $\sigma$  lies for the 500 kg payload class, the largest for 100 kg. This trend is also true for all other variables, and there is a jump that occurs at 400 kg where the highest variation for  $X_{0,descent}$ ,  $m_{0,descent}$ , and starting velocities is present.

There is statistically significant variation of the quantile distribution for  $m_{0,descent}$  across different payload classes. What is interesting is the spread of solutions for the 400 kg payload class. It was seen that for the launch vehicle optimization it was also the highest class that covered the entire  $m_{p,1}$  range. As mentioned before, the 400 kg payload class is an optimal nominal candidate for its widespread characteristics. When carrying out the sensitivity analysis, having a wider variation ensures that the actual RLV itself can support an increase or decrease of  $m$  whether it be of its  $m_{dry}$  or  $m_p$ . In a 3-DoF environment this is not of concern as the mass split between dry and wet only starts to influence the trajectory if the trajectory requires complete exhaustion of propellant. In 6-DoF, this would influence the moment of inertia of the vehicle and thus impact the e.o.m. This is also true for variations of the velocities,  $X_{0,descent}$ , and  $h_{0,descent}$ .

Variables	$\mu$	$\sigma$	CI		Quantile				
			2.5%	97.5%	0	.25	.5	.75	1
$h_{0,descent}$ (m)	28644.19	538.7293	28642.26	28646.12	26086.7	28676.65	28715.23	28787.19	29397.77
$X_{0,descent}$ (m)	41140.69	2589.384	41131.42	41149.95	29938.53	41902.27	41913.81	41916.59	41918.3
$m_{0,descent}$ (kg)	41957.58	2702.612	41947.91	41967.25	30381.79	42696.43	42798.91	42841.47	42868.29
$t_f$ (sec)	400.0838	115.3999	399.6708	400.4967	200.0009	300.2378	400.0117	500.0686	599.9978
$v_{x,0,descent}$ (m/s)	-32.7341	28.25129	-32.8352	-32.633	-41.2844	-41.2	-41.1088	-40.8218	91.13656
$v_{z,0,descent}$ (m/s)	390.8203	17.01486	390.7594	390.8812	384.1059	384.4699	385.0809	386.4441	458.2496

Table 10.8: Measures of central tendency for preliminary ascent and descent starting conditions and criteria, for payload class of 100kg.

Variables	$\mu$	$\sigma$	CI		Quantile				
			2.5%	97.5%	0	.25	.5	.75	1
$h_{0,descent}$ (m)	28470.36	447.1603	28468.76	28471.96	26084.97	28469.23	28505.93	28575.64	29395.63
$X_{0,descent}$ (m)	41177.33	2450.411	41168.56	41186.1	29938.53	41904.71	41913.35	41916.23	41918.02
$m_{0,descent}$ (kg)	42084.32	2605.701	42074.99	42093.64	30340.13	42811.47	42907.06	42948.45	42977.05
$t_f$ (sec)	400.0749	115.4018	399.662	400.4879	200.0016	300.1146	400.3028	499.63	599.9999
$v_{x,0,descent}$ (m/s)	-33.5946	26.57582	-33.6897	-33.4995	-41.728	-41.647	-41.5436	-41.2836	90.76682
$v_{z,0,descent}$ (m/s)	388.104	17.16176	388.0426	388.1655	381.2667	381.654	382.2411	383.6049	457.4855

Table 10.9: Measures of central tendency for preliminary ascent and descent starting conditions and criteria, for payload class of 200kg.

Variables	$\mu$	$\sigma$	CI		Quantile				
			2.5%	97.5%	0	.25	.5	.75	1
$h_{0,descent}$ (m)	28570.21	422.4747	28568.7	28571.72	26018.69	28563.23	28601.1	28674.06	29220.77
$X_{0,descent}$ (m)	41136.65	2334.797	41128.3	41145.01	29938.45	41898.97	41912.76	41916.13	41918.19
$m_{0,descent}$ (kg)	41868.04	2457.418	41859.25	41876.84	30297.13	42597.73	42719.52	42766.91	42792.9
$t_f$ (sec)	399.7879	115.5293	399.3745	400.2013	200.0006	299.6421	399.9285	499.6664	599.9969
$v_{x,0,descent}$ (m/s)	-33.1183	24.99358	-33.2078	-33.0289	-41.5165	-41.4357	-41.3281	-41.0092	90.41368
$v_{z,0,descent}$ (m/s)	389.7572	16.3566	389.6987	389.8158	382.5985	382.9586	383.6052	385.3261	456.7553

Table 10.10: Measures of central tendency for preliminary ascent and descent starting conditions and criteria, for payload class of 300kg.

Variables	$\mu$	$\sigma$	CI		Quantile				
			2.5%	97.5%	0	.25	.5	.75	1
$h_{0,descent}$ (m)	28348.22	509.3182	28346.4	28350.04	26003.96	28305.78	28488.64	28517.9	29070.41
$X_{0,descent}$ (m)	41043.39	2673.815	41033.83	41052.96	29938.43	41901.08	41913.65	41916.47	41918.08
$m_{0,descent}$ (kg)	41833.15	2828.587	41823.03	41843.28	30254.15	42667.15	42702.01	42857.79	42995.38
$t_f$ (sec)	399.7454	115.4477	399.3323	400.1585	200.0028	299.6478	399.6429	499.6766	599.9995
$v_{x,0,descent}$ (m/s)	-32.4141	29.01521	-32.5179	-32.3103	-42.1507	-41.708	-41.6016	-41.5121	90.06097
$v_{z,0,descent}$ (m/s)	387.6096	18.31798	387.544	387.6751	378.4426	380.2963	381.92	382.4065	456.0256

Table 10.11: Measures of central tendency for preliminary ascent and descent starting conditions and criteria, for payload class of 400kg.

Variables	$\mu$	$\sigma$	CI		Quantile				
			2.5%	97.5%	0	.25	.5	.75	1
$h_{0,descent}$ (m)	28389.65	154.7304	28389.1	28390.21	27914.34	28310.28	28348.82	28434.96	28969.13
$X_{0,descent}$ (m)	41474.44	1408.161	41469.39	41479.48	34952.75	41899.58	41913.07	41915.89	41917.83
$m_{0,descent}$ (kg)	42233.45	1543.433	42227.91	42238.98	35263.66	42634.12	42741.52	42784.39	42812.66
$t_f$ (sec)	399.8611	115.6002	399.4468	400.2755	200.002	299.7292	399.6077	500.0428	599.9974
$v_{x,0,descent}$ (m/s)	-37.3843	14.44425	-37.4361	-37.3325	-42.0663	-41.9846	-41.8843	-41.5982	30.61817
$v_{z,0,descent}$ (m/s)	384.2406	12.03191	384.1975	384.2837	379.0375	379.4398	380.0458	381.4889	434.8955

Table 10.12: Measures of central tendency for preliminary ascent and descent starting conditions and criteria, for payload class of 500kg.

### 10.3.2 Preliminary Descent

The results for the chosen nominal candidates are shown in table 10.13. The majority of the candidates are for the 400 kg payload class. The best case  $Q_{max}$  candidate is for 500 kg. When looking at the variation of  $Q_{max}$ , none of the results show a value close enough to the vehicle limit, 110 kPa. This result coincides with what was shown by Simplicio [54], where RTLS trajectories experience a much lower  $Q_{dyn}$  than DRL. We can therefore narrow down the selection criteria and look at the other candidates. As such, this metric is not much of importance to us and this serves as a validation that it is the other metrics of  $m_{burn}$ ,  $\|\mathbf{v}_f\|$  and  $\|\mathbf{r}_f\|$  that are critical in determining feasibility of RTLS operations. The latter is shown to have no variation except for candidates whose propellant mass far exceeds that available on board. We recall that preliminary descent was done without propellant depletion dynamics and therefore as a first step it is determined if any of the potential candidates selected fall into this region. The candidate with the selection criteria for the best case  $\|\mathbf{v}_f\|$  falls into this region, consuming 1668.15 kg over its maximum capacity, and with the candidates  $m_{0,descent}$  being 30254.18 kg, this goes well beyond its capability to return. It is expected that solutions with a lower  $\|\mathbf{v}_f\|$  may require more expulsion of propellant. Due to the way that the ZEM/ZEV guidance calculates the reference thrust, such that it is never non-zero, this can amount to trajectories where  $m_{burn}$  far exceeds  $m_{0,descent}$ .

It is the convex guidance will be more robust and therefore the propellant consumption and  $\|\mathbf{v}_f\|$  will be decreased due to a greater degree of control achievable by imposing a burn schedule. Therefore, it may be worthwhile to look at candidates that can already optimize  $m_{burn,descent}$ . For this candidate,  $m_{0,descent}$  is 30254.15 kg, well within the available propellant used for descent. The other metrics of  $Q_{max}$  and  $\|\mathbf{v}_f\|$  are also not exceedingly larger than their respective optimal cases. Moreover, from the discussion before, it is expected that a final velocity 14.672 m/s can be further reduced by implement convex guidance.

The progression of the 400 kg payload class as the most suitable candidate for is continued here. For this reason, and amongst the usefulness of all the metrics discussed, the best case  $m_{burn,descent}$  solution's conditions are used as nominal conditions for the closed-loop simulations. The best case overall selection criteria shares values that are very similar to this case. However ultimately it is not selected as it has a slight higher propellant burn requirement and is subject to an optimal selection of  $Q_{max}$ , which is nevertheless a metrics that does not exceed the feasible range.

Selection Criteria	$m_{dry,1}$ (kg)	$m_{p,1}$ (kg)	$m_{dry,2}$ (kg)	$m_{burn,descent}$ (kg)	$Q_{max}$ (Pa)	$\ \mathbf{v}_f\ $ (m/s)	RLV ID	Sol. ID	Payload Class (kg)
Best case $Q_{max}$	7297.204	69264.78	1234.302	31247.19	9457.682	9.477059	497	806	500
Best case $\ \mathbf{v}_f\ $	5254.186	50000	1070.45	51688.15	23234.31	2.378602	340	887	400
Best case $m_{burn,descent}$	5254.154	50000	1070.518	17361.55	23234.28	14.67268	802	294	400
Best case overall	5254.176	50000	1070.483	17400.11	23234.29	13.74467	800	769	400

Table 10.13: Preliminary descent nominal selection results.

## 10.4 Aerodynamics

The aerodynamics coefficients for the first stage of the nominal RLV id of 802 that corresponds to the best case  $m_{burn,descent}$  selection criteria as per analysis of the preliminary guidance results is shown in fig. 10.1. Compared to the CALLISTO's first stage there are some noticeable similarities in the  $C_D$  curve. The maximum value is at around 10.9, for a Mach number of 1.1, whereas the CALLISTO peaks at around 5.5 for 1.6. There is a noticeable increase in the spike magnitude for  $C_D$  for the transonic region, and this may be due to the fact as identified by Van Kesteren, and Pagano, and analysed by Martjin, that Missile DATCOM 97's wave drag in the transonic regime is overestimated. The solution to this was to assume uniform diameter, which indeed is the case here as the first stage is modelled as a uniform diameter cylinder. As such, these effects are not present in the results. This large difference in  $C_D$  could be attributed if there was a larger radius of the nominal candidate of 1.1637 m, but it is indeed smaller than the radius of the CALLISTO which is 1.4m. It could be attributed to the reference surface area  $S_{ref}$ , but since the shapes of the the CALLISTO is of the same topology for the nominal RLV, the formula is dependent on radius and thus it is smaller nonetheless.

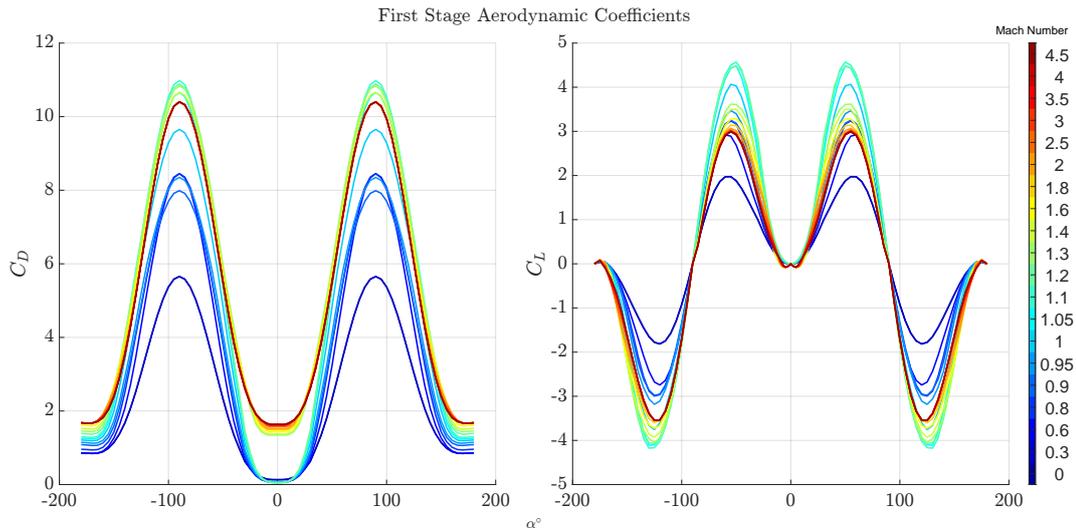


Figure 10.1:  $C_D$  and  $C_L$  data for 802 RLV Id (best case  $m_{burn,descent}$ )

The aerodynamic coefficients computed by computational fluid dynamics method have been graciously supplied by Simplicio, for the CALLISTO vehicle. By considering that the relative shapes of the first stages of the two vehicles are identical, an error fit curve can be made for each  $M$  and  $\alpha$  value available for CALLISTO compared to all the feasible RLV solutions. This is shown in fig. 10.2.

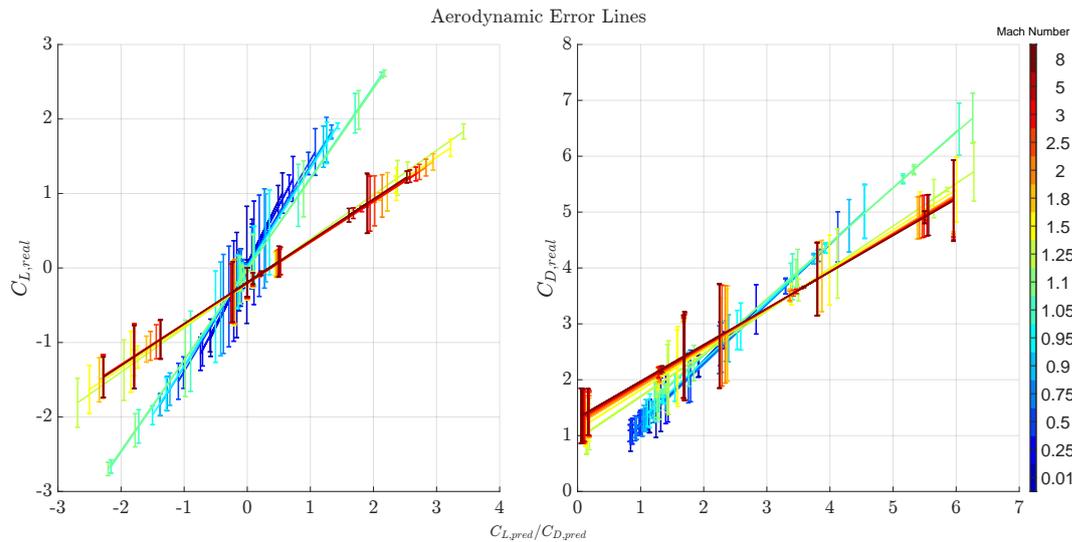


Figure 10.2:  $C_D$  and  $C_L$  linear fit with minimum and maximum variation relative to CALLISTO

The fit curve with minimum and maximum variations is generated by considering the predicted drag  $C_{D,pred}$  and lift  $C_{L,pred}$  coefficients which are taken from the RLV optimization results, have a fit with the real drag  $C_{D,real}$  and lift  $C_{L,real}$  data. This analysis considers a conservative assumption that the relative shapes of the launch vehicles are similar enough, and the primary dimensions of length and diameter are not too different to be able to compare the two via a linear fit. There is in fact a noticeable trend that both real coefficients increase as the predicted coefficients increase as well. For lower  $M$  numbers this spread is most quantifiable due to the smaller variation of the drag and lift relative to  $\alpha$ .

It can be seen for lift that the errors are largest and most accumulated at  $M \leq 1.05$ , and as  $C_{L,pred}$  and  $C_{L,real}$  approaches zero. This may be because of the fact that the lift can cross between positive and negative values and the relative difference becomes larger due to the sign change.

For  $C_D$ , the errors are most accumulated at this same  $M$  region but are smaller. Larger variations are present for  $M > 2$ , and for  $C_D > 4$ , for some values of  $M > .95$  there are large variations. For  $C_L$ , it can be seen that there are two main clusters of lines at different  $M$ , one for  $.01 \leq M \leq 1.1$ , and another for  $1.25 \leq M \leq 8$ . This tells us that there may be a noticeable difference in the methods used to calculate the two coefficients between the CFD and Missile DATCOM results, independent of the geometry of the RLV. The gap between this region is above the transonic just at the border of the transonic regime, which may indicate that the uniform diameter of the RLV is still not a method to bypass the wave drag overestimation in Missile DATCOM 97. This is also true for  $C_L$ , indicating that the flow separation may be also incorrectly estimated due to the poor accuracy of the wave drag estimation, as the latter induces the former.

By using this data, it is possible to run a sensitivity analysis on the aerodynamic errors by varying them along the linear fit. This can serve useful in order to bridge the gap between the higher fidelity CFD data, expected from a RLV of that form, for the transonic regime. For the purposes of this study, this result is not further investigated and is left for future research.

## 10.5 Simulations

In examining the DoE 1 results we are interested in obtaining solutions that are both feasible in all metrics, and solution envelopes that are highly correlated with as few variations of variables as possible. To do so, the data is first visually inspected to determine if there are any trends. The most intuitive inspection can be done by plotting the results in a 3-D scatterplot with  $\|\mathbf{r}_f\|$  on the x axis,  $\|\mathbf{v}_f\|$  on the y axis. Trends on this plot can be spotted as clusters, where the data is more dense in certain regions and less dense in others. Indeed, sparsity of this data visualization will give insight into the robustness of the algorithm, and sensibility of the problem. To identify clusters, the “kmeans” MATLAB function can be used whose input takes the data and sorts it according to the number of groups specified. If the sorting does not match the result expected via visual inspection, the plot range can be trimmed to focus on specific areas and repeat the process until all desired clusters are identified. Once the clusters are separated and solution indices obtained, measures of central tendency can be used on the variable variations and the results of the metrics themselves in that region. These measures provide insight on the statistical distribution of variable variations. This can provide a means to generating a trade-off space for selection of certain variables to feasibility of results.

### 10.5.1 DoE 1

In the first DoE, the following data trends are identified. Using kmeans, it is not possible to segment the data because of the poor quality of the solution, but is visible on fig. 10.3 that there are distinct clusters. Through inspection of the scatter plot, it is possible to distinguish four different clusters with the log scale for x and y. These clusters are manually identified with the ranges as given in table 10.14.

### DoE 1- Metric Plots

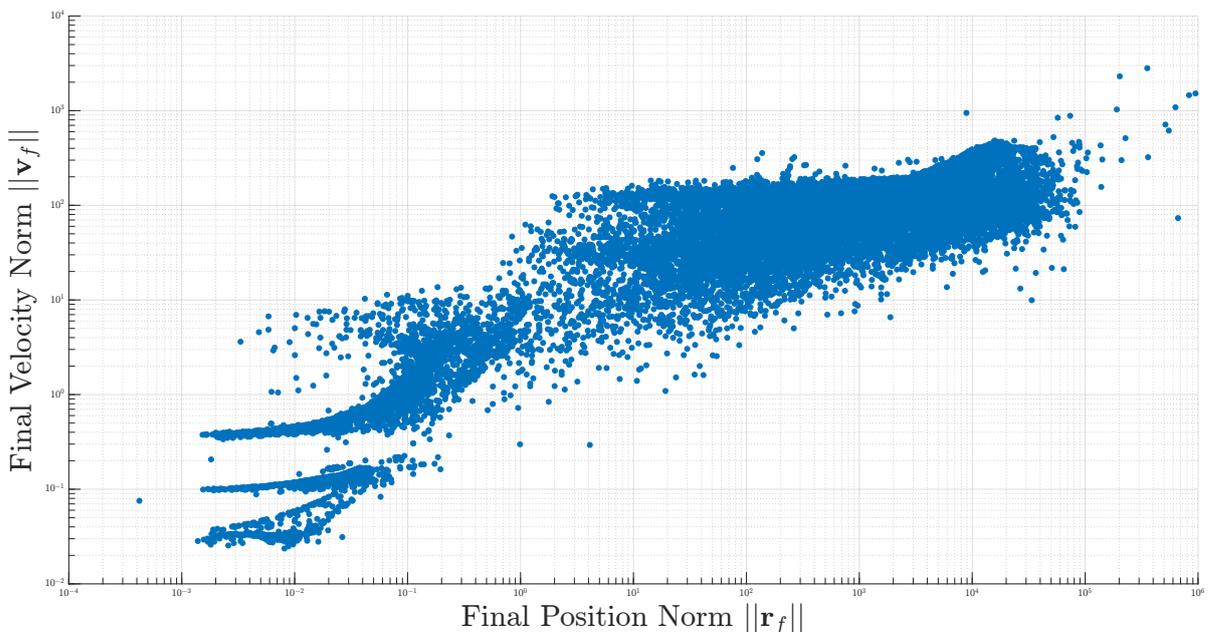


Figure 10.3: DoE1 metric results.

Cluster	Ranges and Feasibility		Number of Solutions
	$\ \mathbf{r}_f\ $	$\ \mathbf{v}_f\ $	
1	(.2,∞) (F&O)	[0,2) (F&O)	5080
2	[0,.04) (O)	(.09,.3) (F&O)	1237
3	[0,.03) (O)	[0,.09)	343
4	(10,∞) (I)	(6,∞) (I)	51263

Table 10.14: Ranges of DoE 1 clusters for  $\|\mathbf{v}_f\|$  vs.  $\|\mathbf{r}_f\|$ . The signs in the parentheses in the ranges column corresponds to F for feasible, O for optimal, and I for infeasible.

When analysing the selection of cluster ranges to solutions within metric tolerances, fig. 10.4 shows that there is an overwhelming majority of solutions that are infeasible in  $\|\mathbf{v}_f\|$  but feasible in  $\|\mathbf{r}_f\|$ , than vice versa. This tells us that the metric which is hardest to meet for these tolerances is the final velocity. It is expected that this result may have something to do with the conditions imposed on the SOCP1 problem  $v_{f,z}$  constraint, which is an equality constraint set to 0, different from the original DESCENDO problem. This could indicate that there may be failures to achieve this condition for  $h \leq h_p$  and for when landing occurs, meaning that it could also impact the result of  $\|\mathbf{r}_f\|$ .

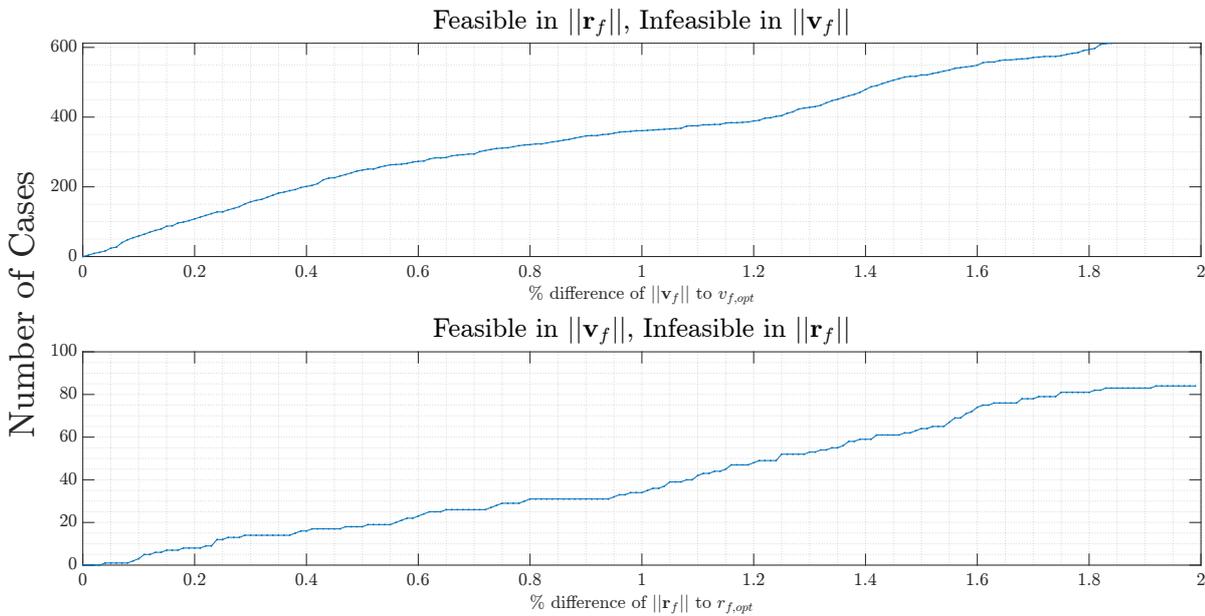


Figure 10.4: Feasible to infeasible metric comparison for DoE 1 results.

Due to the disproportionately larger infeasibility in  $\|\mathbf{v}_f\|$ , the final landing velocity criteria could be too strict. An additional piece of evidence in this is the large separation between the chosen clusters that are examined and the rest of the infeasible solutions. This can be seen from equivalent density in fig. 10.3. Even so, the majority of points in the chosen clusters are in 1 out of the feasible cluster regions (1-3), and the other 2 feasible clusters represent a small percentage of the total 61300 solutions. The reason this provides evidence for the fact that strict  $v_{f,z}$  equality constraint may reduce the reliability of the algorithm is that there is no continuity in the solution envelope. By continuity, we refer to the length to which these feasible cluster zones extend to. By looking at 10.5, a plot that shows the 3 feasible clusters with a dotted line of best linear fit through them, one can see that 1 and 2 are clearly recognizable trends as opposed to cluster 4 (in fig. 10.3). The coefficients for the linear fit were found with MATLAB's polyfit function and their value along with  $R^2$  coefficient is shown in table 10.15 ( $C_1$  represents the slope and  $C_2$  the intercept for a linear polynomial of the form  $\|\mathbf{v}_f\| = C_1\|\mathbf{r}_f\| + C_2$ ). Cluster's 1 and 2 show a strong correlation whilst 3's is fairly weak. When referring to continuity, the high regression coefficient indicates that this cluster of data does not appear out of coinci-

dence but rather is a region where results are predictable. This trend can provide useful data about what burn schedule choices are more optimal than others.

## DoE 1- Metric Plots for Feasible Cluster Zones

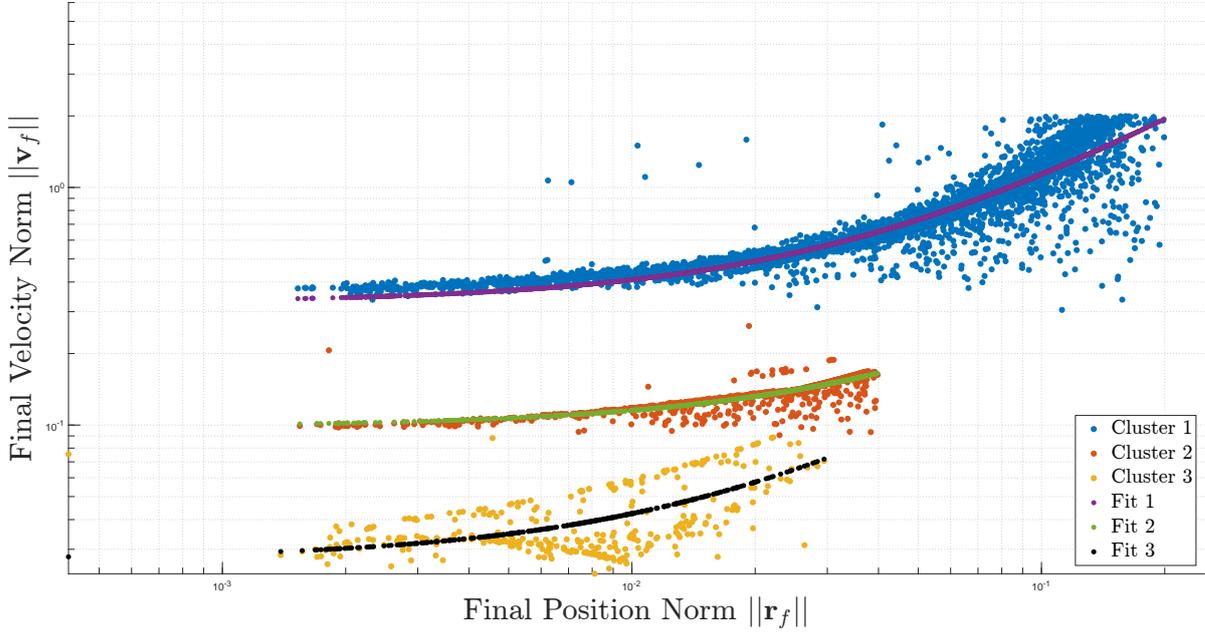


Figure 10.5: Feasible cluster plots with line of best fit as scatter data for DoE1.

Cluster	Linear Fit Statistics		
	$C_1$	$C_2$	$R^2$
1	0.328074	8.076176	0.777951
2	0.098558	1.670712	0.737537
3	.027252	1.514048	0.389108

Table 10.15: Linear fit statistics of DoE 1 feasible clusters for  $\|v_f\|$  vs.  $\|r_f\|$

Tables 10.16 to 10.18 show the measures of central tendency for  $t_{b,i}$ , and  $t_f$  for the three feasible cluster zones. The first out of these three shows the distribution of  $\%t_1$ . The value of 1 for Q4 represents the free burn schedule solution. The higher confidence interval is larger for all 3 clusters than the smaller one, indicating that the distribution of  $\%t_1$  in this cluster is possibly towards the left. The results amongst the different clusters are fairly similar. Cluster 1 seems to have the lowest number of solutions that fall in higher values of  $\%t_1$ , both due to its lowest 97.5% confidence interval, Q3, and second lowest  $\sigma$ . Cluster 3 is somewhere in between, whilst cluster 2 shows the most variation.

Tab. 10.17 shows the measures of central tendency for  $\%t_2$ . The value of 2 for Q4 is the one that represents the free burn schedule. In contrast to the  $\%t_1$  results, this shows the the highest value for Q3 for cluster 1, and cluster 2 has the least variation both in terms of its C.I., and quantile distributions.

Tab. 10.18 shows the cluster results for  $t_f$ . Cluster 1 covers the larger range Q0-Q4, despite having the smallest gap of C.I. between 2.5% and 97.5%. Cluster 2 and 3 are mostly identical.

These results show that with Cluster 1 containing the most data, and the lowest variation for the burn schedule parameters, it is the region where the highest predictability of results can be obtained. In other words, it presents a suitable design region for trading off  $t_f$  as an alternative metric with final position and final velocity wanted.

$\ v_f\ $ vs. $\ r_f\ $	$\mu$	$\sigma$	C.I.		Quantile				
			2.5%	97.5%	Q0	Q1	Q2	Q3	Q4
cluster	0.292984	0.288166	0.297802	0.175154	0.016327	0.163265	0.261224	0.391837	1
2	0.316361	0.304819	0.327903	0.206913	0.016327	0.130612	0.277551	0.489796	1
3	0.285179	0.265247	0.30511	0.187671	0.016327	0.114286	0.244898	0.457143	1

Table 10.16: Measures of central tendency of feasible  $\|v_f\|$  vs.  $\|r_f\|$  clusters for  $\%t_1$ .

$\ v_f\ $ vs. $\ r_f\ $	$\mu$	$\sigma$	C.I.		Quantile				
			2.5%	97.5%	Q0	Q1	Q2	Q3	Q4
cluster	0.302542	0.297142	0.307942	0.19633	0.016327	0.130612	0.293878	0.457143	2
2	0.260076	0.249014	0.271138	0.19831	0.016327	0.097959	0.212245	0.408163	2
3	0.281145	0.258505	0.303785	0.213175	0.016327	0.130612	0.228571	0.440816	2

Table 10.17: Measures of central tendency of feasible  $\|v_f\|$  vs.  $\|r_f\|$  clusters for  $\%t_2$ .

$\ v_f\ $ vs. $\ r_f\ $	$\mu$ (sec)	$\sigma$	C.I. (sec)		Quantile (sec)				
			2.5%	97.5%	Q0	Q1	Q2	Q3	Q4
cluster	307.7441	306.1413	309.3468	58.27025	210	270	290	330	640
2	322.1423	319.9756	324.309	38.84271	230	300	310	340	480
3	328.2507	324.3576	332.1438	36.65671	250	310	320	340	470

Table 10.18: Measures of central tendency of feasible  $\|v_f\|$  vs.  $\|r_f\|$  clusters for  $\%t_f$ .

It is entirely possible that there may be a point along the trajectory that is common to the majority of cases of DoE 1 where it will ‘make’ or ‘break’ the feasibility. This means that the feasibility of the solution is highly dependent on how the guidance algorithm performs at that point. One way to examine this is to check the distribution of the burn schedule across the different feasibility regions.

It can be seen in the feasibility envelope plots, especially for T1 vs T2 (10.12, that there are numerous gaps. Between  $\%t_1$ ,  $\%t_2$ , and  $t_f$ , the main observation is that the latter is the one that has the most control on the physical feasibility of the solution.  $\%t_1$  and  $\%t_2$  have more uniform distributions than  $t_f$  for the feasibility and optimality, as seen on fig. 10.8, fig. 10.9, and fig. 10.10. What can be taken from these plots is that the duration of the first burn tends to be much smaller for optimal and feasible results than the second burn. Moreover, there is an optimal final time region, as seen from the infeasibility distribution in  $t_f$ , that is smoother compared to the others. This can also be seen in the feasibility envelope plots, where there is a line of optimal results that covers nearly the entire  $\%t_1$  and  $\%t_2$  region, that starts at  $t_f = 300/310$  sec (fig. 10.13 and 10.11), which coincidentally is the bin for which the lowest number of infeasible results were present as seen in fig. 10.10. The reasoning for the fact that selection of  $t_f$  can represent more clearly the feasibility of the solution can be down to how these variables are treated in the guidance algorithm.  $\%t_1$  and  $\%t_2$  are variables that are passed into a burn schedule vector  $q$ , which are then represented in a discretized grid. Whenever using binary vectors or discretized grids in closed-loop systems, there may be some discrepancies in future solutions based on the rounding errors that occur between the cut off of a thrusting and non-thrusting point, or vice versa. For example, it is possible that a decimal second difference in the starting time between one guidance solution and the next causes the vector of thrust schedule to switch an element from a 0 to 1, even though it underestimates the burning period. That switch from a 0 to 1 means that the next guidance solution will consider a thrusting duration that is not a difference in time equal to the difference between the last guidance solution time and the current, but it will be a whole time step in the solution difference, which when dealing with grids that are relatively smaller than the simulation grid, may be in the same order of magnitude or one smaller than the final time. Let us consider an example of  $t_f = 300$ . The first time index in the guidance solution is 59, and the next is 61.5 seconds. We set  $\%t_1 = .3$ . In the previous guidance solution, the time was less than the cutoff threshold for  $t_1$ , which would be 60 seconds. Therefore, this solution will generate a vector with a grid for  $N=20$  that will overcompensate the thrust duration by  $(300-60)/19-1$  (1 for considering an  $f_{gui}$  of 1), that is roughly 11.6316 seconds longer than the predicted thrusting period. This effect may be even more exacerbated when we have a lower  $f_{gui}$ , and the guidance steps near  $t_1$  evaluate the solution further

away from that point.

On fig. 10.6, the y axis are the % of infeasible solutions compared to the total solutions for that time step difference between  $t_1$  and  $t_{gui}$ . The x axis represents the difference of the guidance time step and the time of the second burn point start. It can be seen for the first two plots, which is  $N_{cvx} = 1$  and  $N_{cvx} = 2$ , there is a slight increase from the left and right side in the number of 100 % infeasible solutions. Therefore, there is evidence to show that there is a statistically significant increase in the number of infeasible solutions when you approach the  $t_2$  point.

Primarily the location of infeasible results occurs seconds before  $t_1$  and in the zero thrust phase between  $t_1$  and  $t_2$ . Evidence of this can be seen as in  $t_1$  we note that there is a spike in infeasible results after .55 seconds, which indicate the first guidance solution after entering the  $t_1$  to  $t_2$  region as the guidance frequency was 1 second. This error could occur because the reference thrust value recorded by the ACT block does not coincide with the thrust requirement of the first guidance solution calculated after passing  $t_1$ . This means that the guidance solution reads that the vehicle is still thrusting, whilst in reality it does not want it to thrust and so the input thrust acceleration does not match the constraints of the SOCP problem for the next guidance step.

The other possibility, is that there is an issue with the thrust rate constraints when you have to switch off and switch on. These constraints, when you assume a minimum thrust value that is greater than the achievable thrust rate over one step in the discretized grid, can cause infeasibility. The solution would be to relax the problem and introduce an additional constraint for when thrust stops or/and thrust recontinues to allow any thrust rate or an increased thrust rate magnitude. Alternatively, a new input to the problem could be introduced which is a variable time step vector. There is an issue however of introducing non-convex inputs, and the time step vector as the discretized grid over time would no longer be convex. By introducing a variable time step, you can take into account the regions where the thrust switches on and/or off and increase the time step to account for thrust magnitude change needed to follow the solution. Moreover, a variable custom time step can also prevent the time over-prediction effects discussed before by reducing the time step at the point of approach of  $t_1$  and  $t_2$ . The issue of non-convex inputs like a custom step size vector is that it forces a non-smooth bound for which the solver may have difficulties finding a solution.

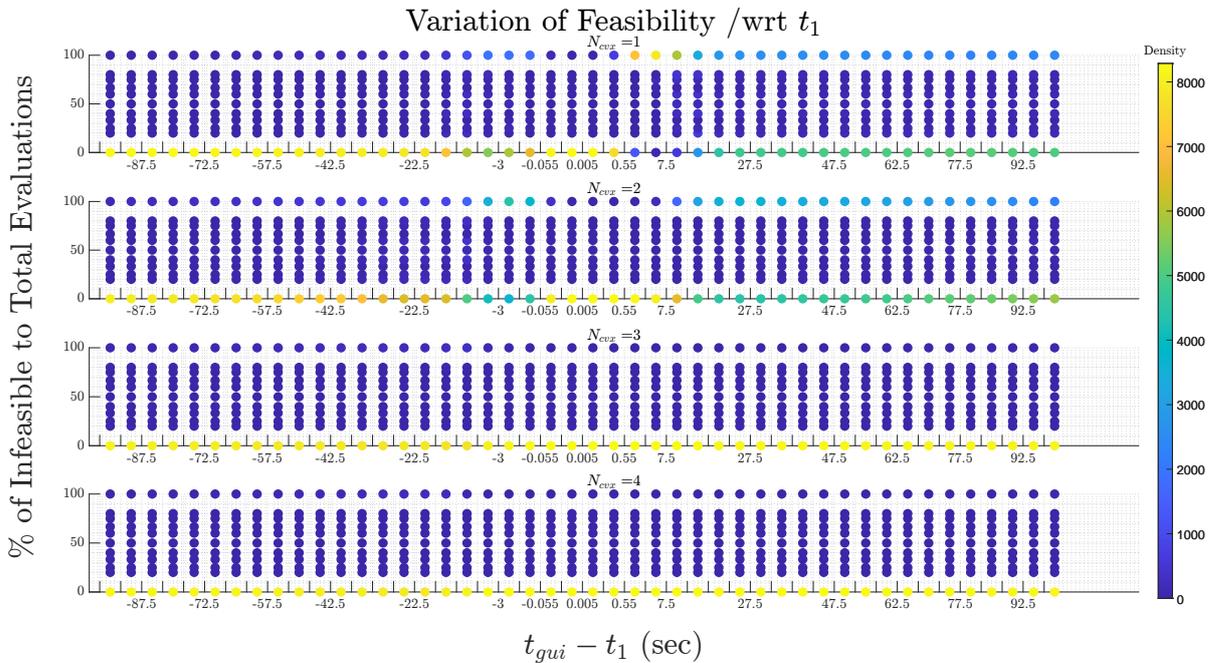


Figure 10.6: Density plot for variations of feasibility with respect to total evaluations for difference between  $t_{gui}$  and  $t_1 = t_{sim}$ .

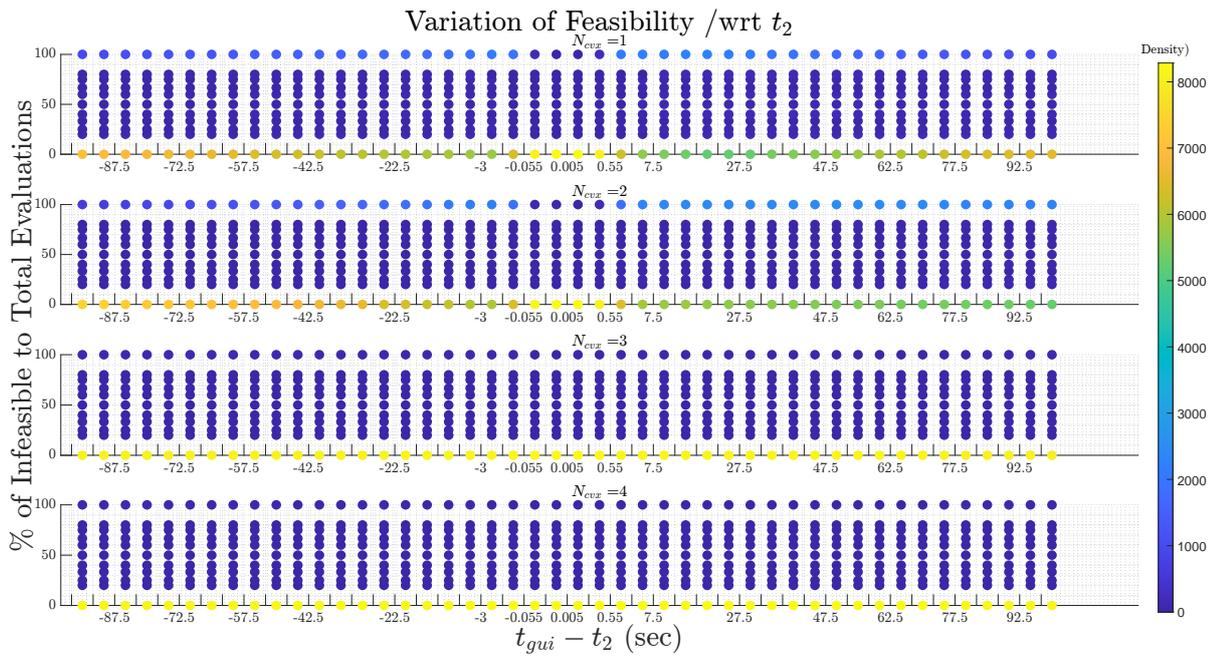


Figure 10.7: Density plot for variations of feasibility with respect to total evaluations for difference between  $t_{gui}$  and  $t_2 = t_{sim}$ .

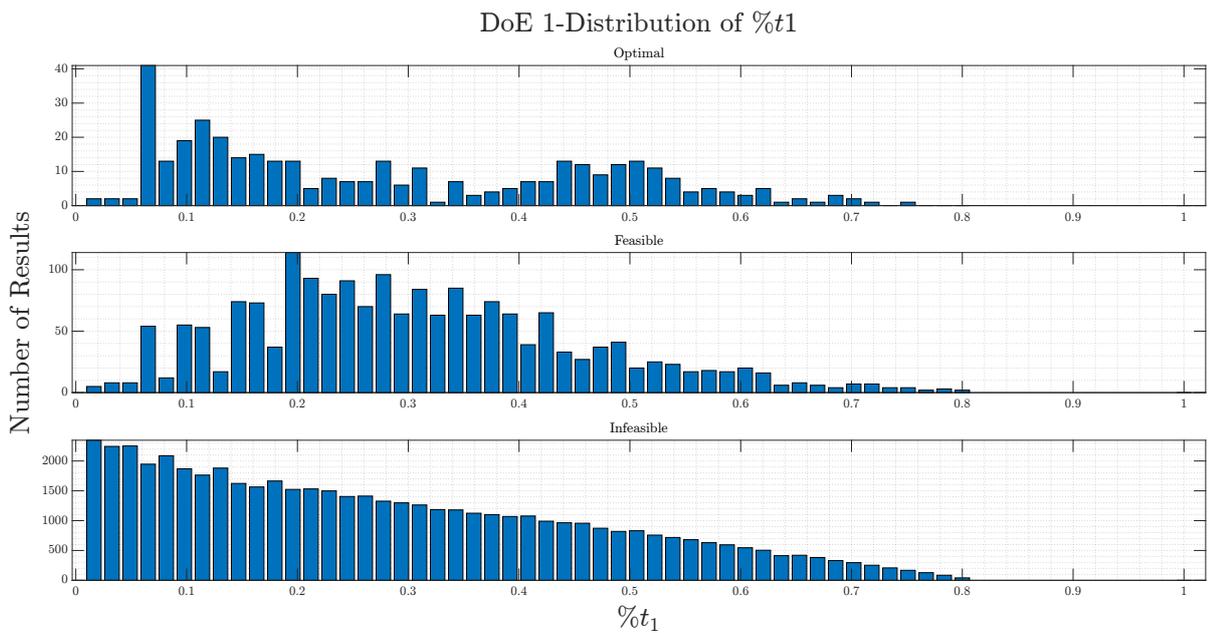


Figure 10.8: Distribution of optimal, feasible, and infeasible values of  $\%t_1$  for DoE1.

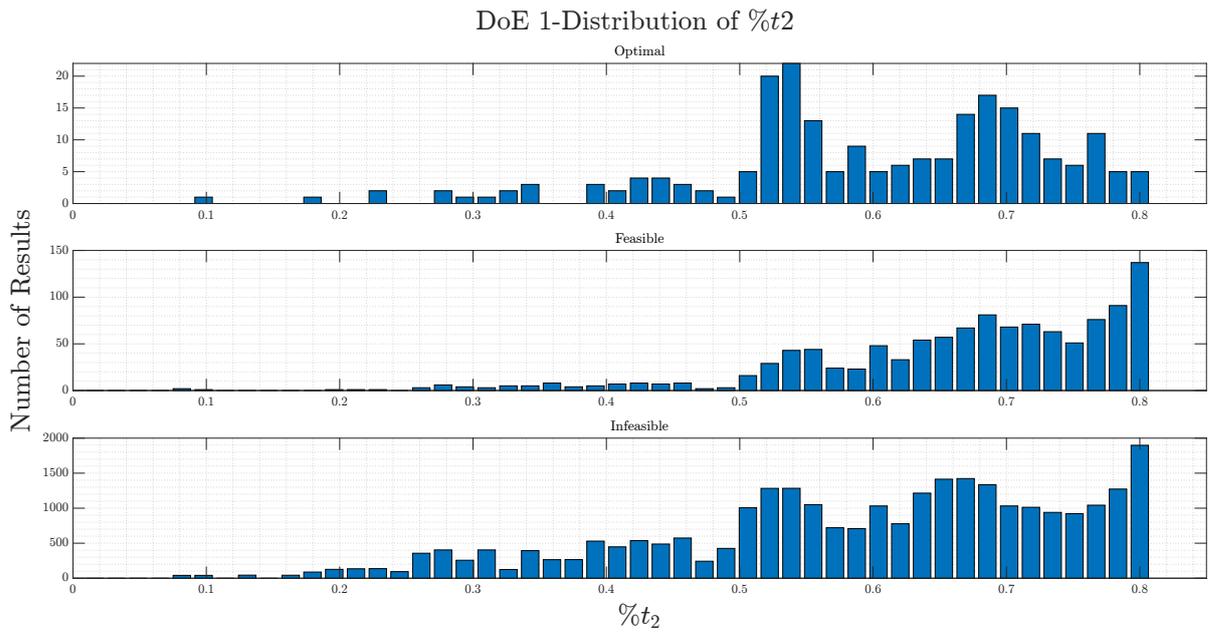


Figure 10.9: Distribution of optimal, feasible, and infeasible values of %t<sub>2</sub> for DoE1.

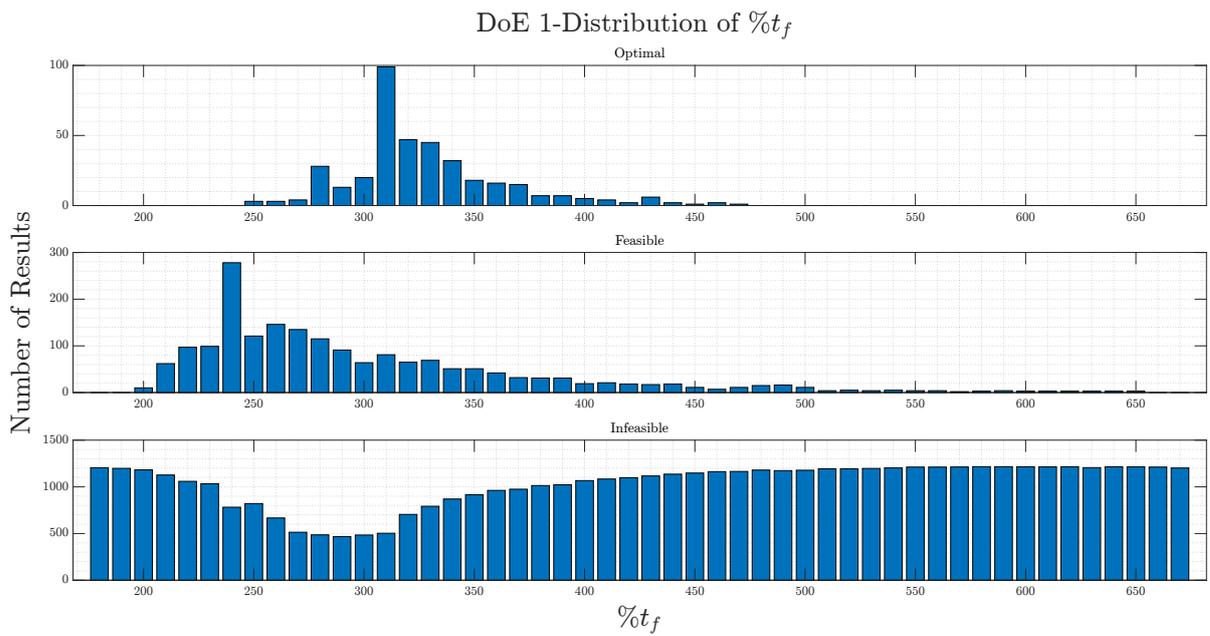


Figure 10.10: Distribution of optimal, feasible, and infeasible values of %t<sub>f</sub> for DoE1.

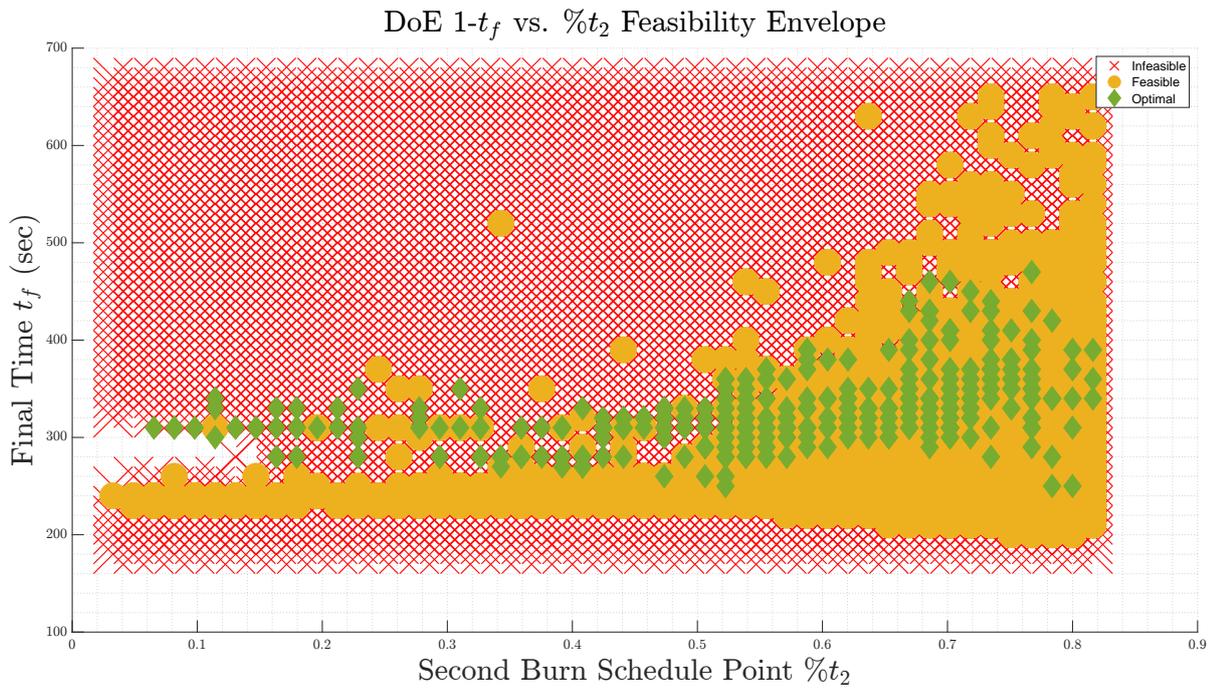


Figure 10.11: a nice plot

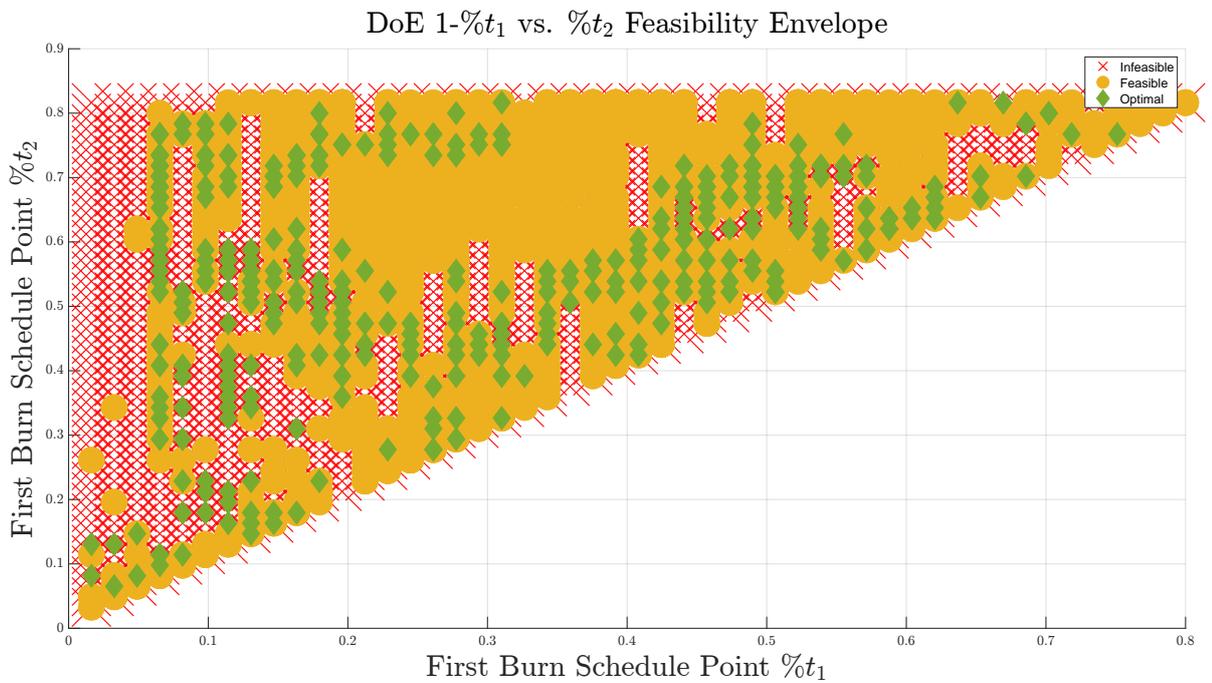


Figure 10.12: a nice plot

Nominal Candidate	$\%t_1$	$\%t_2$	$t_f$ (sec)	$\ \mathbf{r}_f\ $ (m)	$\ \mathbf{v}_f\ $ (m/s)
1	0.310204	0.620408	380	7830.334	105.9759
2	0.293878	0.620408	190	6753.126	258.1761
3	0.081633	0.195918	500	391.4016	73.50938
4	0.097959	0.473469	250	0.003318	0.354625
5	0.244898	0.391837	200	4253.586	225.6671
6	0.42449	0.604082	610	1120.967	135.9666

Table 10.19: DoE 2 and 3 nominal candidates selected from DoE 1 results.

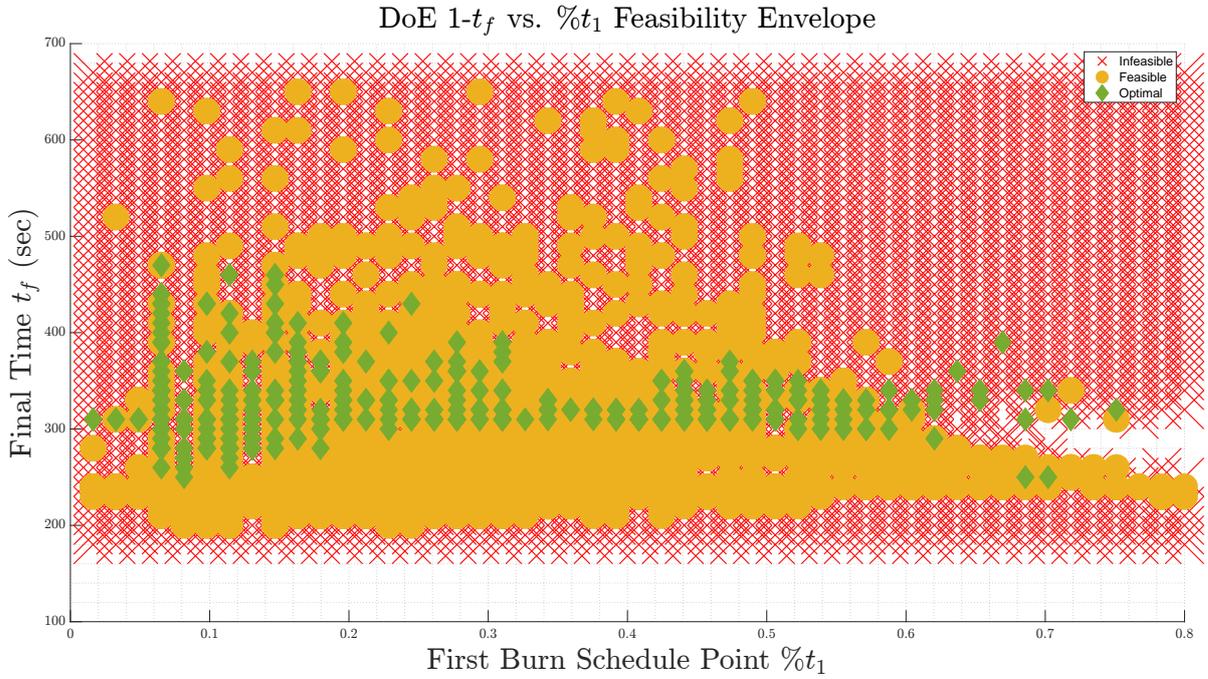


Figure 10.13: a nice plot

The nominal candidates selected from DoE1 are shown in table 10.19. One optimal candidate (4) is selected amongst a variety of infeasible candidates for both metrics of  $\|\mathbf{r}_f\|$  and  $\|\mathbf{v}_f\|$ . The purpose of this is to determine both how variations of guidance algorithm parameters, and frequencies of guidance and simulator can both improve or decrease performance.

## 10.5.2 DoE 2

The measures of central tendency for results of DoE 2 are shown in tables 10.20 to 10.29 for metrics of  $\|\mathbf{r}_f\|$  and  $\|\mathbf{v}_f\|$ . The results for variations of discretized grid size  $N$  show that there is a direct decrease for all measures as  $N$  increases except for Q4 for  $N = 50$ . The increase of  $N$  shows a statistically significant decrease in the metric of both metrics and as such it is a variable that can be simply optimized against computation time and not an optimal value based on the quality of the solution.

The number of successive convexifications distribution shows some improvement as  $N_{cvx}$  increases for both metrics but there is no clear statistical trend that increasing this improves the solution.  $N_{cvx}$  is dependent on the robustness of the problem, and the present problems with burn schedule overestimation or underestimation due to the discretization of the grid as discussed in section 10.5.1 may take control over the would-be improvement of refining the dynamics of the problem.

The reference thrust difference weight factor  $\mathcal{W}_{\eta_w}$  shows that decreasing the value below the nominal of  $1e-4$  can improve the quality of the solution as  $\sigma$ ,  $\mu$  and all but Q4 show improvements most noticeably for  $\|\mathbf{r}_f\|$  rather than  $\|\mathbf{v}_f\|$ . There is not enough statistical evidence to show that there is a more favorable weight factor when aiming for increasing feasibility of results.

The maximum thrust direction angle  $\theta_{max}$  shows noticeable improvement for  $\|\mathbf{r}_f\|$  for Q1 and Q2 as the angle increases. The other quantiles however show no clear statistical trend.  $\sigma$  also decreases for increasing  $\theta_{max}$  for both metrics. One possible explanation for this is that as explained in section 10.5.1, increasing the control profile can improve or decrease the quality of the solution as increasing the search space may provide with more local optimal that may be far less optimal than a global one. Therefore, it may be that there are solutions that can benefit from this and those that otherwise suffer from an increase of range in the control profile. The lack of trend in the higher quantiles supports this as this variable can only target a select few candidates.

The variation of the nominal candidate results show that nominal candidate 4 is likely infeasible, 3 may or may not be improved further by change of burn schedule or frequencies, whilst 6 shares the same possibility for only its  $\|\mathbf{v}_f\|$ . 3 out of 6 candidates have been turned from infeasible to feasible, and 2 out of these 3 are optimal. The results show that there is statistically significant evidence that tuning the parameters of the algorithm(s) can improve modestly the solution feasibility. There is however very strong evidence that there is no optimal set of guidance variables that guarantee feasibility or even offer an advantage over others, as all the  $\sigma$  for all variables and metrics are near the same order of magnitude or even greater in value than the  $\mu$ , showing evidence that there is a large statistical spread in the data.

N	$\mu$ (m)	C.I. (m)		$\sigma$ (m)	Quartile (m)				
		2.5%	97.5%		Q0	Q1	Q2	Q3	Q4
10	5111.949	4707.92	5515.978	5030.757	0.002202	866.7187	4779.555	7449.332	53543.22
20 (*)	2752.684	2463.952	3041.416	3601.172	0.002039	296.7763	1225.889	4253.586	43782.82
30	1886.395	1592.595	2180.194	3472.313	0.002121	55.51481	324.4138	3141.408	38433.11
40	1493.267	1312.108	1674.426	2259.485	0.002177	33.86605	300.6431	3349.357	21271.49
50	1466.817	1275.718	1657.915	2383.453	0.004546	16.50744	154.8919	3011.659	24867.11

Table 10.20: DoE 2 Measures of statistical tendency for variation of  $N$ 's  $\|\mathbf{r}_f\|$ . Asterix (\*) denotes nominal value for experiments without variation of this parameter unless otherwise stated.

N	$\mu$ (m/s)	C.I. (m/s)		$\sigma$ (m/s)	Quartile (m/s)				
		2.5%	97.5%		Q0	Q1	Q2	Q3	Q4
10	179.4001	172.378	186.4221	87.43501	0.098074	122.9181	156.8576	259.0109	321.0363
20 (*)	140.416	133.1112	147.7207	91.10759	0.096568	70.89255	136.4173	223.9699	407.4824
30	110.9422	103.7843	118.1001	84.59682	0.132558	27.64788	119.0705	175.4399	401.7398
40	103.434	97.21499	109.653	77.56556	0.156819	28.30501	113.666	142.6744	302.3612
50	93.97785	87.7684	100.1873	77.44666	0.068698	17.49764	79.42482	134.9321	297.2325

Table 10.21: DoE 2 Measures of statistical tendency for variation of  $N$ 's  $\|\mathbf{v}_f\|$ . Asterix (\*) denotes nominal value for experiments without variation of this parameter unless otherwise stated.

$N_{cvx}$	$\mu$ (m)	C.I. (m)		$\sigma$ (m)	Quartile (m)				
		2.5%	97.5%		Q0	Q1	Q2	Q3	Q4
1	2489.167	2232.435	2745.899	3545.349	0.002121	88.98436	645.1872	3982.37	36693.54
2	2612.915	2326.478	2899.351	3952.857	0.004546	80.4165	652.2184	4089.98	43782.82
3 (*)	2554.221	2295.754	2812.688	3569.302	0.003318	89.68192	717.6272	4089.98	38433.11
4	2560.168	2274.637	2845.7	3937.665	0.002039	65.38899	581.6893	4089.98	53543.22

Table 10.22: DoE 2 Measures of statistical tendency for variation of  $N_{cvx}$ 's  $\|r_f\|$ . Asterisk (\*) denotes nominal value for experiments without variation of this parameter unless otherwise stated.

$N_{cvx}$	$\mu$ (m/s)	C.I. (m/s)		$\sigma$ (m/s)	Quartile (m/s)				
		2.5%	97.5%		Q0	Q1	Q2	Q3	Q4
1	132.3646	126.1585	138.5706	85.70274	0.096568	74.66148	127.4415	208.4941	321.0363
2	122.6118	116.0102	129.2134	91.10233	0.132558	30.44079	120.9239	208.4941	321.0363
3 (*)	125.7799	119.3	132.2599	89.48546	0.068698	54.34426	119.6808	209.1375	401.7398
4	122.841	116.2444	129.4376	90.97131	0.10051	44.34124	119.1164	208.7086	407.4824

Table 10.23: DoE 2 Measures of statistical tendency for variation of  $N_{cvx}$ 's  $\|v_f\|$ . Asterisk (\*) denotes nominal value for experiments without variation of this parameter unless otherwise stated.

$\mathcal{W}_{\eta_w}$	$\mu$ (m)	C.I. (m)		$\sigma$ (m)	Quartile (m)				
		2.5%	97.5%		Q0	Q1	Q2	Q3	Q4
5e-5	2338.04	2065.229	2610.85	3368.262	0.002039	36.07185	483.617	3867.274	21271.49
1e-5	2398.988	2128.526	2669.45	3339.265	0.002121	43.53461	565.7526	3925.833	17381.3
5e-4	2529.085	2237.556	2820.614	3599.365	0.002177	94.27853	654.5969	4037.186	36693.54
1e-4 (*)	2753.8	2394.569	3113.032	4423.877	0.003099	125.2312	774.4151	4174.517	53543.22
5e-3	2751.573	2433.402	3069.745	3928.307	0.002202	145.9501	995.0729	4174.517	38433.11

Table 10.24: DoE 2 Measures of statistical tendency for variation of  $\mathcal{W}_{\eta_w}$ 's  $\|r_f\|$ . Asterisk (\*) denotes nominal value for experiments without variation of this parameter unless otherwise stated.

$\mathcal{W}_{\eta_w}$	$\mu$ (m/s)	C.I. (m/s)		$\sigma$ (m/s)	Quartile (m/s)				
		2.5%	97.5%		Q0	Q1	Q2	Q3	Q4
5e-5	125.3056	117.9694	132.6418	90.57642	0.068698	35.94071	123.8846	208.9231	321.0363
1e-5	123.7866	116.4619	131.1113	90.43437	0.10051	34.02277	123.0964	208.4941	321.0363
5e-4	126.6497	119.5288	133.7707	87.91845	0.133319	53.30454	123.4727	208.4941	321.0363
1e-4 (*)	130.6916	123.5794	137.8037	87.58547	0.096568	55.32583	124.9416	210.0604	407.4824
5e-3	123.1037	115.7789	130.4285	90.43564	0.098074	39.92772	119.2422	208.4941	401.7398

Table 10.25: DoE 2 Measures of statistical tendency for variation of  $\mathcal{W}_{\eta_w}$ 's  $\|v_f\|$ . Asterisk (\*) denotes nominal value for experiments without variation of this parameter unless otherwise stated.

$\theta_{max}$ (deg)	$\mu$ (m)	C.I. (m)		$\sigma$ (m)	Quartile (m)				
		2.5%	97.5%		Q0	Q1	Q2	Q3	Q4
60	4358.505	3976.006	4741.003	4766.671	0.013526	283.5801	3194.993	6923.69	53543.22
65	3108.225	2784.398	3432.052	3967.288	0.002352	150.8284	1656.455	5092.764	43782.82
70 (*)	2302.587	2011.398	2593.776	3570.522	0.002039	73.50129	461.6045	3602.856	28718.24
75	1554.123	1320.344	1787.902	2866.567	0.002121	35.6368	278.6946	2656.491	38433.11
80	1425.834	1244.471	1607.197	2260.133	0.002367	19.41456	369.2721	2198.534	21271.49

Table 10.26: DoE 2 Measures of statistical tendency for variation of  $\theta_{max}$ 's  $\|r_f\|$ . Asterisk (\*) denotes nominal value for experiments without variation of this parameter unless otherwise stated.

$\theta_{max}$ (deg)	$\mu$ (m/s)	C.I. (m/s)		$\sigma$ (m/s)	Quartile (m/s)				
		2.5%	97.5%		Q0	Q1	Q2	Q3	Q4
60	157.5478	150.2672	164.8284	90.73044	0.102111	98.01395	149.3964	237.7441	321.0363
65	134.8692	127.2163	142.5222	93.75855	0.099199	49.92667	128.692	216.8418	307.6048
70 (*)	117.424	110.1412	124.7069	89.3011	0.068698	31.43861	120.2011	196.5842	407.4824
75	110.0344	102.9959	117.073	86.30581	0.098074	20.09658	107.8844	167.8672	401.7398
80	109.1643	103.0078	115.3209	76.72207	0.11471	37.93855	114.9561	145.2916	302.3612

Table 10.27: DoE 2 Measures of statistical tendency for variation of  $\theta_{max}$ 's  $\|v_f\|$ . Asterisk (\*) denotes nominal value for experiments without variation of this parameter unless otherwise stated.

Nominal Candidate	$\mu$ (m)	C.I. (m)		$\sigma$ (m)	Quartile (m)				
		2.5%	97.5%		Q0	Q1	Q2	Q3	Q4
1	778.6982	603.7976	953.5988	1990.553	2.415856	138.2475	225.1506	535.339	32129.91
2	1304.409	1057.302	1551.515	2812.329	0.002039	0.030016	0.171962	2225.272	38433.11
3	4197.048	3875.47	4518.625	3432.136	43.53461	1039.831	3264.239	6909.861	12826.72
4	6249.823	5999.185	6500.461	2852.52	3445.321	3867.274	5378.154	7449.332	14849.13
5	1764.881	1460.895	2068.868	3452.725	0.002202	3.45679	183.7577	2040.63	24867.11
6	1221.034	852.6798	1589.389	4188.046	4.095149	79.93218	282.7009	905.7538	53543.22

Table 10.28: DoE 2 Measures of statistical tendency for variation of Nominal Candidate's  $\|r_f\|$ .

Nominal Candidate	$\mu$ (m/s)	C.I. (m/s)		$\sigma$ (m/s)	Quartile (m/s)				
		2.5%	97.5%		Q0	Q1	Q2	Q3	Q4
1	90.43588	86.59584	94.27592	43.7037	2.865491	50.86814	91.58956	132.1213	195.878
2	69.09564	60.70671	77.48456	95.47474	0.068698	0.528899	2.027921	172.3836	407.4824
3	179.9864	172.6132	187.3595	78.69246	17.49764	100.002	186.2821	244.8735	282.8889
4	241.5605	238.5768	244.5441	33.95708	196.5842	216.415	230.5565	259.0109	321.0363
5	52.68945	47.89497	57.48392	54.45634	0.098074	4.78785	34.88128	102.2621	223.5115
6	127.8486	125.6996	129.9975	24.43288	12.29632	118.5587	124.889	136.2558	304.5971

Table 10.29: DoE 2 Measures of statistical tendency for variation of Nominal Candidate's  $\|v_f\|$ .

### 10.5.3 DoE 3

The results for the variation of  $f_{sim}$  to  $f_{gui}$  are shown in fig. 10.14 for  $\|\mathbf{r}_f\|$  and fig. 10.15 for  $\|\mathbf{v}_f\|$ . When plotting the quantiles for solution metrics against the ratios of  $f_{sim}$  and  $f_{gui}$ , a better representation of the effects of both parameters can be obtained as realistically it is this ratio and not just the change in individual parameters that reflects how well the guidance algorithm can cope with a higher fidelity simulation. Immediately, some interesting trends are present. There is a trend that occurs at  $\frac{f_{sim}}{f_{gui}}=20$  where both metric's Q4 stretch out several orders of magnitude below Q3, and the solutions become increasingly more widespread for order of magnitude of Q1 and Q0. The median of these results also shows a noticeably but more modest decrease for both metrics, and interestingly it remains mostly stable at around  $10^3$  for  $\|\mathbf{r}_f\|$  and  $10^2$  for  $\|\mathbf{v}_f\|$ . There is therefore statistical evidence to show that there is a performance increase noticeably beginning at this frequency ratio. An interesting outlier lies in  $\frac{f_{sim}}{f_{gui}}=10$  which is the nominal ratio used for all DoE's. In this range lie the possible combinations of 10-1, 100-10, and 200-20. All possible combinations are included in the Q0-Q1 interquartile range. This showcases proof that the anomaly may in fact be a local optimal region. There is more evidence to support that this point is optimal as with DESCO the simulations where also ran with  $\frac{f_{sim}}{f_{gui}}=10$ , with the authors mentioning that it was selected after testing.

For lower values of  $\frac{f_{sim}}{f_{gui}}$ , there is less statistical certainty about their performance as  $\|\mathbf{v}_f\|$  contains various outliers (red cross points), and the quartiles do not even show feasibility for either  $\|\mathbf{v}_f\|$  or  $\|\mathbf{r}_f\|$ . Having a matching  $f_{sim}$  and  $f_{gui}$  should in theory improve the performance as for every step that the e.o.m. of the spacecraft are propagated, a new guidance solution is calculated. This means that the feedback loop updates more frequently, and it is counterintuitive to see it degrade performance. Looking at the values of the exit flags for each DESCO call, there is no clear indication that the solution quality is affected, as it is not straightforward to compare diagnostics results for simulations with different frequencies of simulation and guidance. In order to yield any valuable information about these low ratios, more tests should be conducted and with the removal of SOCP2 to remove the influence of non-convex dynamics. It is possible that the fast feedback loop between guidance and simulator causes issues in the reference thrust vector term in the objective because the values may approach zero and those zeros in the convex optimization controller can induced rounding errors which build up and eventually cause the controller to produce infeasible SOCP runs and an infeasible solution.

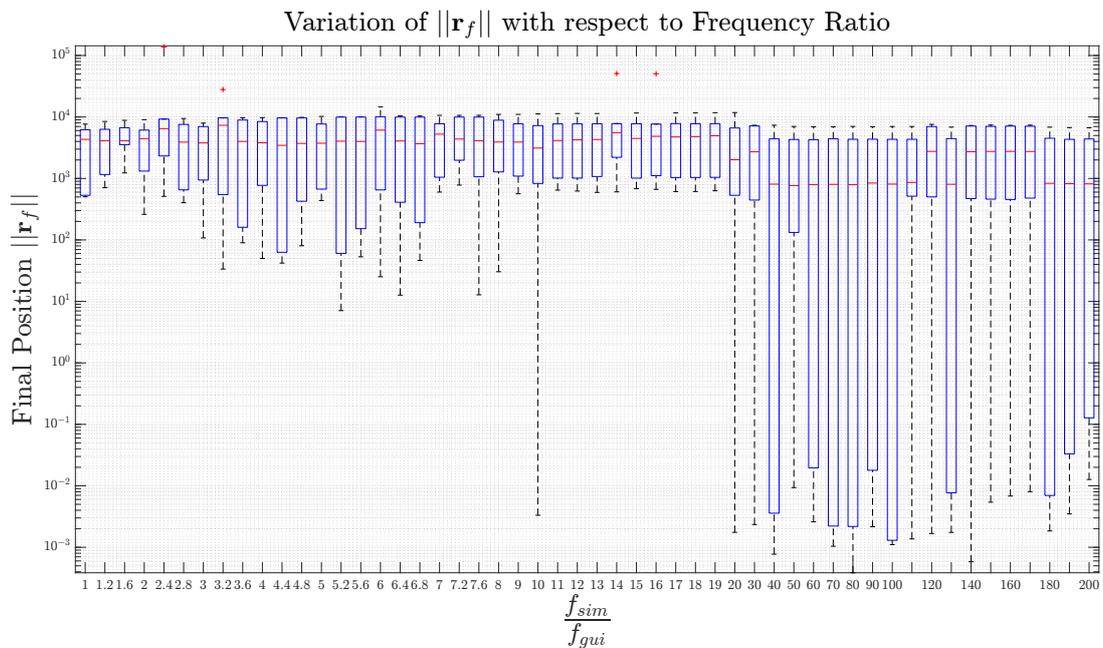


Figure 10.14: DoE3  $\frac{f_{sim}}{f_{gui}}$  variation results for  $\|\mathbf{r}_f\|$

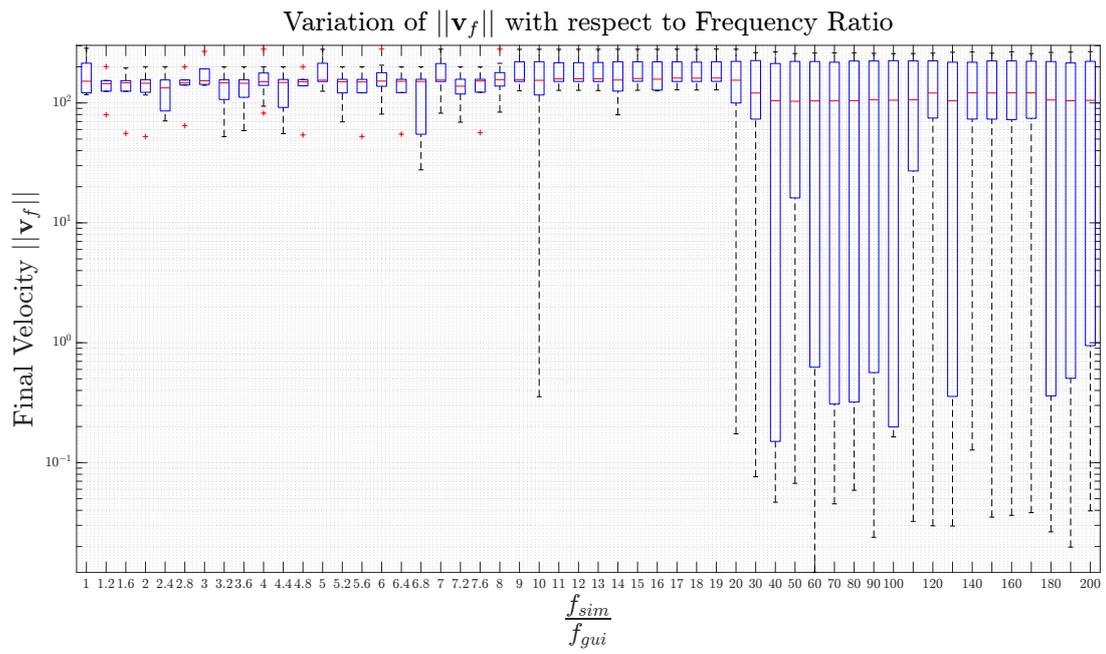


Figure 10.15: DoE3  $\frac{f_{sim}}{f_{gui}}$  variation results for  $\|\mathbf{v}_f\|$

DoE 1 Results	
$\mathbf{r}_f$ (m)	$\mathbf{v}_f$ (m/s)
4.2091e-4	.0754

Table 10.30: DoE 1 nominal candidate used for DoE 4.

### 10.5.4 DoE 4

The nominal candidate chosen from DoE 1 for the sensitivity analysis is shown in table 10.30. This candidate was chosen to be the most optimal such that any errors could only amount reasonably to a direction away from optimality. The plots from fig. 10.16 to 10.23 are grouped as three different graphics. On the top left is the distribution relative to the range difference of range of the metric value for  $\|\mathbf{r}_f\|$ . On the bottom left is the same plot but for  $\|\mathbf{v}_f\|$ . On the right side the density of the distribution for the absolute velocity difference  $|(\|\mathbf{v}_{f,nominal}\| - \|\mathbf{v}_f\|)|$  against the absolute position difference  $|(\|\mathbf{r}_{f,nominal} - \mathbf{r}_f\|)|$  is plotted as different colored shaded regions, with the single results overlaid on top as black dots. This plot aims to show where the greatest region of plots are present. The reason for this is that the wide majority of sensitivity analysis results do not demonstrate a clear fit for error against uncertainty. From the previous DoE's, results show that there isn't a clear set of optimal guidance algorithms parameter values that universally work for all trajectories.

For  $v_{N,0}$ ,  $v_{E,0}$ , and  $v_{D,0}$  sensitivity results the results are fairly widespread. The majority of results are mostly optimal. The range of  $\|\mathbf{r}_f\|$  spans across more order of magnitude than  $\|\mathbf{v}_f\|$  for all the velocity cases. The distributions for all velocity elements are fairly similar. The  $v_{N,0}$  component is the only one that is non-zero at the nominal condition. Nevertheless, the same peaks are noticeable for  $\|\mathbf{r}_f\|$  at  $10e-3 \leq \|\mathbf{r}_f\| < 100e-3$ , and  $100 \leq \|\mathbf{r}_f\| < 1e+3$ . This is also true for  $1 \leq \|\mathbf{v}_f\| < 10$ . It also exhibits an increase in the two distributions to the right. This is evidence that the norm of the velocity, and not the increase in magnitude of one of the elements that influences the errors in  $\|\mathbf{v}_f\|$  and  $\|\mathbf{r}_f\|$ . This provides useful evidence that the assumption that the off-plane velocity occurs, and that the assumption of the flat earth such that no crossrange is incurred is ideal. This result is fortuitous for the fact that the velocity loss at separation can safely be assume to be a loss of norm and not specific in any of the velocity elements.

The distribution for  $\|\mathbf{r}_f\|$  and  $\|\mathbf{v}_f\|$  for all other variable variations follow the same trend for velocities. The most distinguished distribution is the one for  $\sigma_{URE}$ , where for  $\|\mathbf{r}_f\|$  after the peak the distribution is monotonically decreasing. There is also a considerable number of cases beyond 10m/s for the absolute velocity difference. The distribution of the scatter plot of fig. 10.23 show a much wider spread for  $.3 \text{ m/s} \leq \|\mathbf{v}_{f,nominal}\| - \|\mathbf{v}_f\| < 20 \text{ m/s}$ , and  $.3 \text{ m} \leq \|\mathbf{r}_{f,nominal}\| - \|\mathbf{r}_f\| < 20 \text{ m}$ . This variation causes a visible difference in the density spread of solutions. Previous variables show a somewhat log-log linear relationship, meaning that as  $|(\|\mathbf{r}_{f,nominal}\| - \|\mathbf{r}_f\|)|$  increases, the rate at which  $|(\|\mathbf{v}_{f,nominal}\| - \|\mathbf{v}_f\|)|$  increases slows down. This is a useful insight because it allows focusing on improvement of one of the metrics and not both simultaneously when turning a non-feasible candidate into a feasible one. There is a caveat to this however for  $X_0$ ,  $h_0$ ,  $m_0$  and the velocities, most noticeably  $v_{N,0}$ ,  $v_{D,0}$ , at the  $.3 \text{ m/s} \leq \|\mathbf{v}_{f,nominal}\| - \|\mathbf{v}_f\| < 3 \text{ m/s}$ , and  $.09 \text{ m} \leq \|\mathbf{r}_{f,nominal}\| - \|\mathbf{r}_f\| < .3 \text{ m}$  region, where there is a nonlinear log-log relationship. This instead shows for the optimal and feasible region there is an increase in the rate of velocity errors. This evidence matches the one found in DoE 1 where there where more noticeably more infeasible solutions in  $\|\mathbf{v}_f\|$  that where feasible in  $\|\mathbf{r}_f\|$  than vice versa. Nevertheless, results in this curved cluster are feasible and almost all optimal for  $\|\mathbf{r}\|$ .

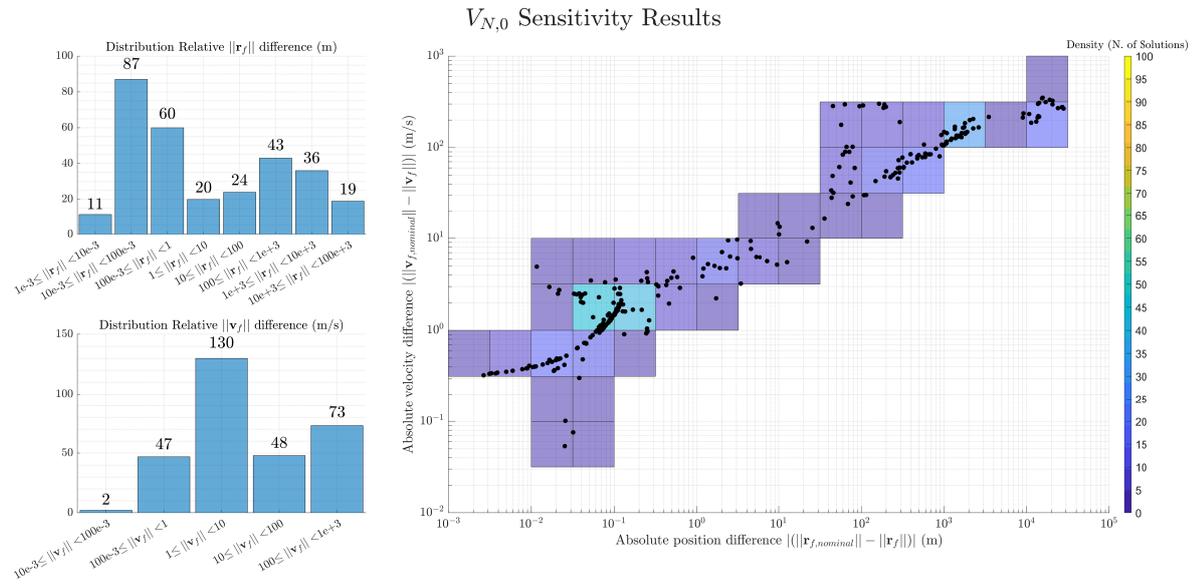


Figure 10.16: DoE 4 sensitivity analysis for  $v_{N,0}$

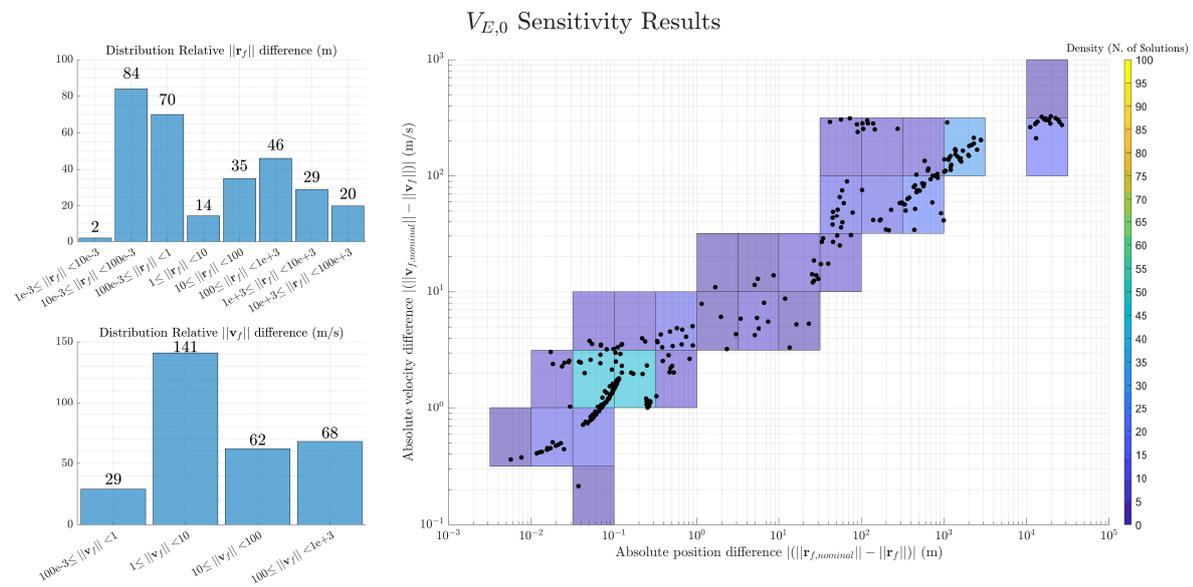


Figure 10.17: DoE 4 sensitivity analysis for  $v_{E,0}$

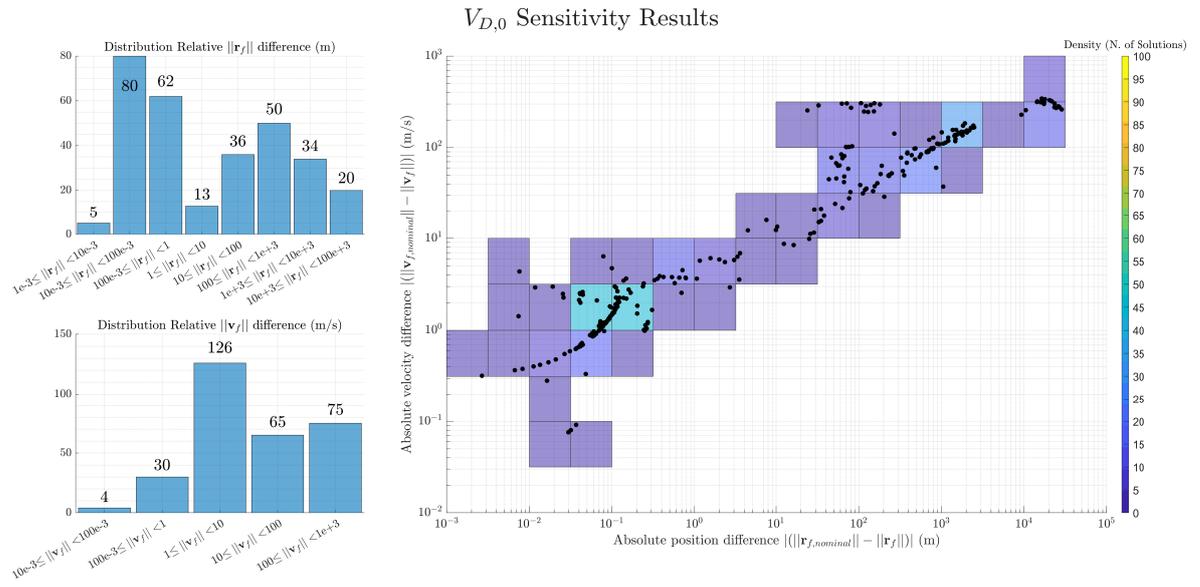


Figure 10.18: DoE 4 sensitivity analysis for  $v_{D,0}$

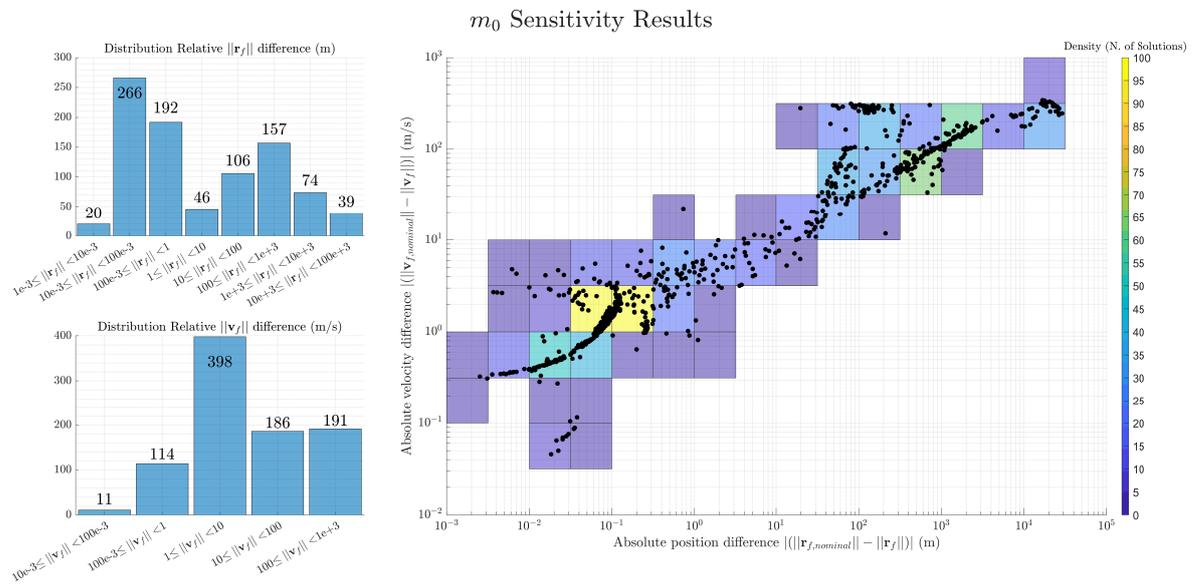


Figure 10.19: DoE 4 sensitivity analysis for  $m_0$

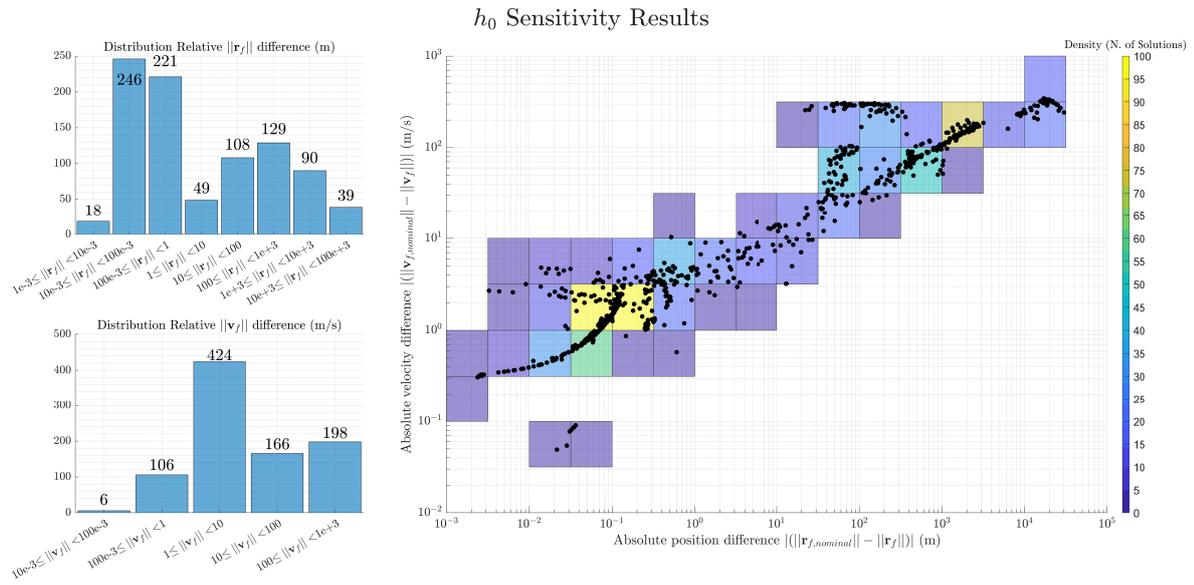


Figure 10.20: DoE 4 sensitivity analysis for  $h_0$

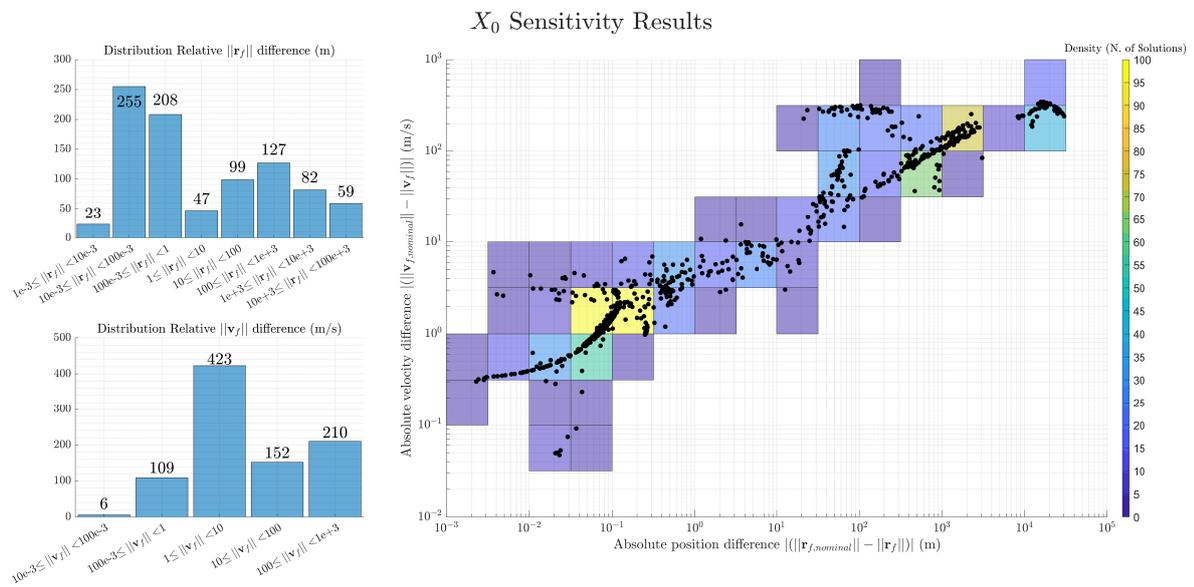


Figure 10.21: DoE 4 sensitivity analysis for  $X_0$

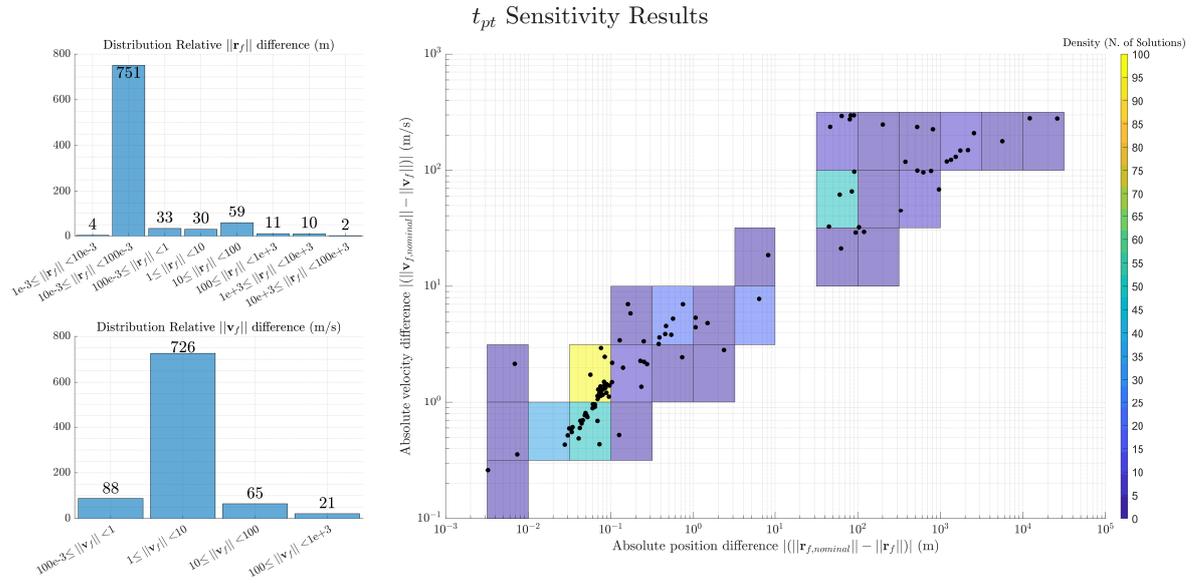


Figure 10.22: DoE 4 sensitivity analysis for  $t_{pt}$

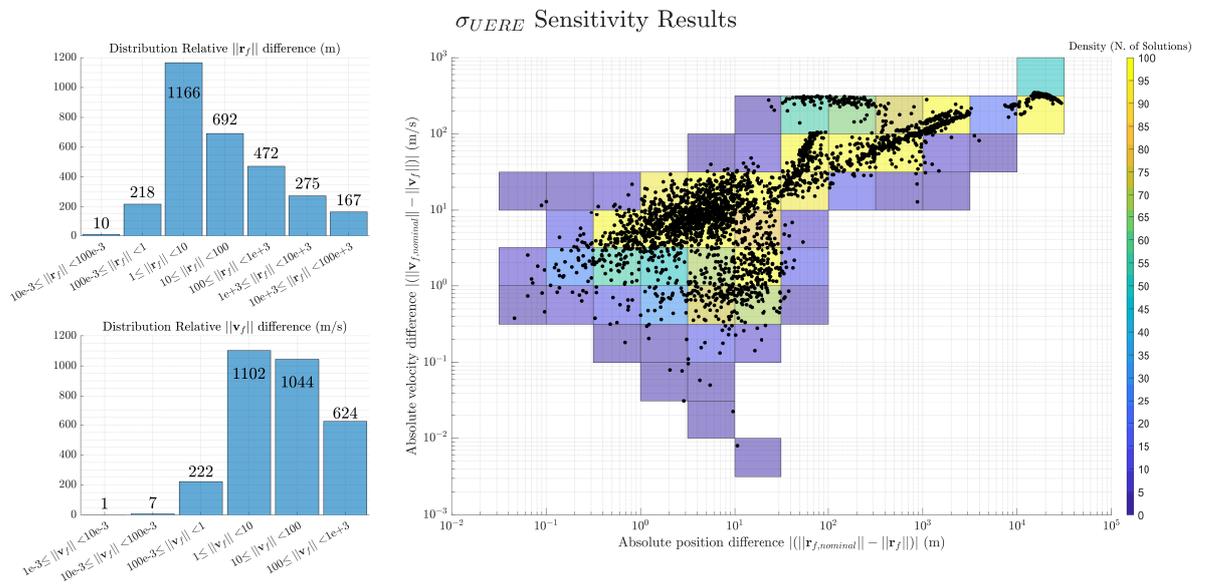


Figure 10.23: DoE 4 sensitivity analysis for  $\sigma_{URE}$

# Chapter 11

## Conclusion

In this chapter the conclusions of the research are covered, and the research questions and subquestions are answered.

With the rise of convex optimization guidance algorithms, and the research gap for a VEGA launch vehicle to enter the RLV market, this research aimed to analyse the feasibility of RTLS by design of a two-staged VEGA variant for reusable operations. The vehicle is optimized based on 5 different payload classes within 100 to 500 kg to determine the optimal payload class based on system and mission requirements for vehicle. Due to the specific requirements and use-case of the MX and M10 engines, a launch vehicle optimization routine is written from the ground up with a dual objective form, one for vehicle mass with priority to maximize propellant mass but minimize dry mass, and the other for vehicle center of gravity position. The latter is implemented for rendering feasibility of results as translatable as possible to a 6-DoF environment by over-engineering the procedure that may improve controllability of the thrusters due to the increase of pivot action. The engine sizing is selected so as to match the diameter of the first stage given the bounds selected prior to the launch vehicle optimization discipline. The nominal engine is then selected as one for which the nominal operating point is as close as possible to the atmospheric conditions at sea level, by assuming it operates as an ideal engine with an altitude adapting nozzle, such that only under-expansion losses are incurred. A similar approach to that of the CALLISTO RLV is then applied, by writing a similar variant of the DESCENDO guidance algorithm, as the DESCO. The usage of convex optimization in this algorithm considers first a solution with no additional dynamics other than thrust acceleration and gravity acceleration. Through successive convexification, non-convex dynamics, drag in this case, can be applied to obtain a higher fidelity solution. This will then yield, through the creation of a GNC block process, a 3-DoF environment where the guidance algorithm is used in a closed-loop fashion and grant the ability to answer the following research questions:

**Research Question:** *To what extent is the VEGARLV configuration feasible for RTLS operations based on system sizing, offline planning, and online planning capabilities?*

This research-question was then broken down to help examine the feasibility of operations at each different discipline to give insight prior to the final phase of simulations, whether there is a good indication that the RLV design is feasible for operations through the following sub-question:

**Sub-Question 1:** *What payload class can satisfy the structural index limit and achieve physically feasible ascent and descent operations?*

**Sub-Question 2:** *What type of burn schedule does the VEGARLV require for descent?*

**Sub-Question 3:** *What is the influence of tuning parameter choices on the guidance algorithm and its frequency relative to the simulation?*

To be able to answer these research sub-questions, the optimization strategy for this project was sequential in nature and discipline feasibility was the factor that allowed propagation to the next discipline. This is in contrast to previous approaches used at TU Delft due to the fact that the toolkits, routines, and algorithms written have been created and not built upon already existing software architecture, except for optimizers and

solver packages. Moreover, because the focus is on the feasibility of return through a closed-loop approach of simulations, thus in real-time, the context of cost evaluation is not considered as feasibility of operations is directly bounded by set stakeholder mission and vehicle requirements. The disciplines involved in this approach where the engine sizing, launch vehicle optimization, aerodynamics, preliminary ascent and descent analysis, and 3-DoF trajectories. The constraints set to these discipline's inputs where guided by the constraints of the stakeholder requirements and already existing data. Software used to carry out optimization of these disciplines involved the usage of NASA CEA (Chemical Equilibrium with Applications) for engine sizing, Missile DATCOM 97 for aerodynamics, YALMIP and ECOS as the SDP (semi-definitive programming) package and solver respectively for the guidance algorithm in the 3-DoF simulations, and fmincon for optimization of the preliminary analysis of ascent.

The optimization procedure began with the building of a feasible set of candidates of M10 and MX engine. The former whose information was available online on the AVIO website and the latter is currently a work in progress with a know  $I_{sp}$  value and whose rest of characteristics was reverse engineered by cross-referencing this with  $I_{sp,meas}$  from multiple NASA CEA evaluations. Results showed that variation of the error in  $I_{sp}$  varied more for the MX than that of the M10, and that a more refine grid search would be necessary to obtain more accurate values. In constrast to the M10 however, there was an inversely proportional relation between the  $\%I_{sp,error}$  and  $P_e$  for the MX, which makes it an appealing choice for an engine to operate at sea-level conditions when a high thrust is needed to nullify the final velocity  $\|\mathbf{v}_f\|$ . The two engines do complement each other in the context of controllability as despite the fact that the M10 has a lower achievable thrust, its higher  $I_{sp}$  means less mass is consumed and more precision can be acquired as both engines are restricted to operating within 10% to 100% of their maximum thrust. Nevertheless, the optimal candidate chosen for the M10 still allowed more than 70% of its maximum thrust during operation at sea-level pressures, making a good choice for redirecting the RLV to land with a vertical control profile.

From the results of the launch vehicle optimization, all payload classes can satisfy the structural index limit. There is however an optimal payload class compared to the others for its desirable spread of characteristics of  $m_{p,1}$  and starting conditions as shown in tables 10.11 and 10.13. The preliminary descent results show that perhaps due to the larger possible variation of the RLV at this payload class, it renders it more robust as a choice as it was deemed a suitable candidate for 3 out of the 4 selection criteria for preliminary descent. The other criteria considered the best case  $Q_{max}$ , but it was validated, in similarity with results from the CALLISTO study, that at an altitude of around 26 – 30 km the maximum dynamic pressure does not come close to the maximum that the vehicle can achieve, when considering the same maximum as the VEGA-C launch vehicle. The larger variability of vehicle characteristics for the 400 kg payload class also makes it a suitable candidate for sensitivity analysis as when selecting a nominal candidate the intention is to make sure it operates well in all possible conditions and not optimal just at one.

The aerodynamics results show large variations in peaks of  $C_D$  and  $C_L$  compared to a similar design such as the CALLISTO, despite the fact the nominal candidate and the majority of the candidates across different payload classes have a smaller first stage radius and as a consequence  $S_{ref}$  as well. It was noted that between the pre-transonic and transonic region, the error curves where clustered at two different groups for both  $C_D$  and  $C_L$ . This may indicate that despite the fact that the vehicle diameter is uniform, which should alleviate the effects of wave drag overestimation, there may still be a large degree of inaccuracy at this  $M$  region. Nevertheless, there is a moderately strong fit between the drag and lift coefficients between the CALLISTO and nominal RLV for this study, paving the way to sensitivity analysis of the errors in the aerodynamic coefficients. As the project built the foundation for various tools and routines to test the feasibility of operations for the VEGA variant bounded by the requirements of the study, it is worthwhile to mention also that the over-engineering of the launch vehicle optimization process had proved a desirable addition. With the inclusion of the center of gravity term as the second objective to maximize relative to the position of the thrusters of the first stage, the results yielded an overall smaller diameter and little to no statistically significant change in other 3-DoF dependent characteristics of the RLV, relative to tests ran without the second objective. This may help translate the procedure with more ease and increase the usefulness of the results when implementing a 6-DoF simulation.

The inclusion of the burn schedule pre-processing proved to be an effective method to circumvent complications to the algorithm by usage of big-M formulations. This procedure allocates the time needed to build the controllers before the beginning of the descent and is only a function of the guidance and launch vehicle  $T_{max}$ ,  $\%T_{min}$ ,  $\$T_{max}$  and  $m_{dry,1}$  parameters, and no information about the position or velocity of the RLV at the time of descent is needed to build these controllers. By applying a different controller at  $t = 0$  an initial thrust value was also able to be formulated, granting more control at the initial condition without employing

an analytical approximation of the initial thrust that may worsen performance.

The DoE results showed that there are several feasible and optimal results for values of  $\%t_1$ , particularly in the range of .1 to .5. In this region, the results for  $t_f$  optimality fluctuates but the envelope encapsulates well from 300 to 310 seconds. Taking into account the possible time delay effects that may influence the burn schedule at points of approach near  $\%t_1$  and  $\%t_2$  for  $t_{gui}$ , this may be a good nominal selection to counteract these effects as feasibility of the solution is most robust to variation of the burn schedule at this point.

Overall, results showed that the variation of  $\frac{f_{sim}}{f_{gui}}$  had a greater statistically significant influence on the quality of the solution than tuning of the parameters. In other words, the former can be deemed as more of a global improvement to the feasibility than the latter. The spread of the variation of the tuning parameter results showed that whilst improvement was possible by selecting certain values for  $\mathcal{W}_{\eta_w}$ ,  $\theta_{max}$ , and  $N$ . The former two of the three showed improvement but not widespread. The thrust difference objective may not be as influential of a term for the given values of  $\mathcal{W}_{\eta_w}$  considered. The maximum direction angle  $\theta_{max}$  may exhibit improvement on the solution but is highly dependent on the robustness of the algorithm. Since the maximum number of evaluations considered was high (10000) for each SOCP iteration, and with the results showing some noticeably improvement in Q2 but not higher quartiles, this variable is also dependent on the path taken and not global optimal value exists for the nominal cases considered. This was explained by the fact that increasing the control profile may decrease the performance by settling for a local optimum as opposed to a global optimum within a smaller range that would be harder to find for the solver because of the increase in bounds. Nevertheless, when 3/6 nominal candidates achieved feasibility by tuning the parameters and 2 of these achieve optimality. The difference between parameter tuning and control of simulator and guidance frequency is that whilst both may be dependent on the robustness of the controller, the results showed that there is a more well defined region for which  $\frac{f_{sim}}{f_{gui}}$  optimal value exists, and even a local optimum at  $\frac{f_{sim}}{f_{gui}} = 10$ , which may be preferred as a trade off between computation time of the trajectory and feasibility. Nevertheless, it is important to mention the difference in the available variation of the parameters as the maximum number of variations per variable in DoE 2 was only 6, which may reduce the level of insight due to a small sample.

The sensitivity analysis results show the most variation for the chosen set of values of  $\sigma_{URE}$ . These log-log relationship between the final velocity and final position differences are more spread out for this analysis, and the majority of simulations are outside feasibility range for both metrics. The initial state variations show very similar distribution shapes for both metrics and also exhibit a cluster at optimal values of absolute position difference and absolute velocity difference. Interestingly, the relationships for velocities,  $h_0$ ,  $m_0$ , and  $X_0$  show a linear log-log relation, meaning that the position errors are more likely to increase in proportion to the velocity errors. The added  $t_{pt}$  show that the algorithm can exhibit good robustness for both metrics, and the majority of the solutions are in the feasible region of velocity difference and optimal for position difference. In conclusion, to a certain extent the feasibility of operations for the VEGA variant of the nominal launch vehicle design is limited in its robustness to variations of the trajectory due to  $\sigma_{URE}$  errors. However, as no state estimation and correction procedure has been adopted, the overwhelming conclusion is that to a larger extent the VEGARLV is a suitable candidate for return given the number of feasible and optimal solutions found across the different DoE's. A more optimal selection of frequencies may improve the performance of the algorithm indirectly as well. The selection of engine, vehicle, and guidance algorithm show both discipline and interdiscipline feasibility and in this context the RTLS can certainly be deemed as a feasible operation subject to correct planning and tuning. The tuning parameter choices, within the variations considered, did not seem to have as great an effect on the feasibility of the solution as the choice of  $\frac{f_{sim}}{f_{gui}}$ . The most important tuning parameter choice was the burn schedule parameters  $\%t_1$ ,  $\%t_2$ , and  $t_f$ , whose feasibility envelope was well defined, and a clear optimal and safe  $t_f$  of 310 sec was identified.



# Chapter 12

## Recommendations

Throughout the document several recommendations have been given on how to improve performance or augment the disciplines with better models and data. For convenience, these are restated here and are separated into three different categories: recommendations on future research that can stem from this project, model improvements, and software advice. The sections below will discuss these.

### 12.1 Research

This research was tasked with determining the feasibility by subjecting the disciplines to set stakeholder requirements and as such all routines had to be written from scratch. With the availability now of the verified and validated results, different studies can be conducted that build upon the available launch vehicle optimization, simulator, and guidance algorithm and study behavior at the lower level. The list mentions the possible studies that can be conducted to further explore feasibility of RTLS.

- A verification of the launch vehicle optimization for different sets of stakeholder requirements has not been carried out. This would verify the procedure for any type of launch vehicle, allowing the research to expand into different payload classes.
- Convex optimization techniques are not the only method to generate the closed-loop trajectory. Pseudo-spectral techniques identified in the literature study may also be an alternative to generate the trajectory, and with their acceptance of non-linear dynamics, can also be used as a reference offline solution.
- When landing, retro-propulsion may interact with the landing legs by the variation of the thrust plume's geometry. This effect may influence landing performance and can be studied further with CFD software to better represent the landing procedure.
- The SOCP problem is not restricted to just SDP and Pseudo-spectral optimization. MILP (mixed-integer linear programming) approaches also exist which allow non-convex constraints.

### 12.2 Model Improvements

- The preliminary analysis of the ascent flight is only an analytical approximation coupled with an fmincon procedure. This does not include atmospheric drag and it was seen that  $t_{co}$  far exceeding what was deemed reasonable for a vehicle with an orbit insertion at 600km. Previous research at TU Delft carried out optimization of the ascent profile with tudat and such can be done here as well to improve the set of initial conditions for descent. Alternatively, a convex guidance algorithm for ascent can also be created and the overall optimization of the 3-DoF simulation be a coupled ascent and descent flight. In this method the analytical equations for ascent can be kept but included in the equations a drag term that gives a more moderate  $h_{0,descent}$ .
- A 6-DoF simulation can be applied to examine the effects of rotational motion. Currently the 3-DoF approximates  $\alpha$  as the angle of the thrust and velocity vectors, and not rotational states which are coupled with the kinematic motion.

- With the overall optimization of the set of all disciplines being sequential in nature, an iterative approach coupling the feasibility of RLV and trajectory together could prove useful to give insights into how the variation of the RLV can influence RTLS.
- Engine over-expansion was not considered as it was assumed the MX and M10 behaved optimally for all  $P_e > P_a$ . An analysis of similar liquid propulsion engines can be carried out to obtain estimations of the thrust coefficients at various nozzle conditions.
- Wind perturbations and other effects such as  $J_2$  gravity was not considered for this study but their influence may be detrimental to the performance of the algorithm.
- Inclusion of other dynamics in SOCP problem such as pressure and lift was studied but not extensively. Whilst results showed no real improvement over the original formulation, with added effects like wind it may be worthwhile to add them especially so since their influence can easily translate to the convex formulation by adding an additional term to the velocity equations.

### 12.3 Software

- There exists various different solvers that can be used to solve SOCP problems aside from ECOS. Most noticeably are MOSEK and GUROBI. It was mentioned by Simplicio in a conversation that the former of the two was less robust but provided better solutions when it did work.
- Missile DATCOM 97 showed that between the CALLISTO and the nominal RLV, despite the difference in sizes, with the geometries being similar, there was an unexpectadly higher  $C_D$  and  $C_L$ . A more precise technique to calculate the aerodynamic coefficients such as computational fluid dynamics (CFD) can be used to obtain a more accurate aerodynamic modelling of the vehicle.
- The simulation is run on a single core. With MATLAB's parallel toolbox capabilities, the simulator can be rewritten to allow parallel computation of the environment and the guidance algorithm simultaneously.
- Whilst the software written and research completed presents a novel use case for a VEGA variant of the launch vehicle, extensive work had to be put in to create and verify these tools. A suggestion for future students would be to focus on already existing tool-boxes such as tudat, and built upon only the necessary components in C++. YALMIP is applicable with C++ and the transcription of the problem would not be too different from that in MATLAB.

# Appendix A

## Appendix

### A.1 Vehicle Optimization

The center of gravity of the first vehicle stack was evaluated to be use to carry out von-mises stress criterion on both the second and first stage during lift off. The point of origin for the center of gravity location is the base plate of the first stage, not the tips of the engine nozzles. Moving along the longitudinal axis, towards the payload, therefore, corresponds in a positive change of the center of gravity location of the first stack. The center of gravity lateral coordinate is 0 as the vehicle is perfectly symmetrical. To calculate  $x_{cg,I}$ ,  $m_I$  must first be calculated as given by equation A.1.1. Note that the variable for length  $L$  is uppercase to denote the cumulative distance from the origin to center of mass of that structure (denoted by the subscript), and not the length of the portion of the RLV that the subscript refers to.

$$\begin{aligned}
 m_I = & \\
 & 2m_{MX} + 2m_{M10} \\
 & + (1.1 \sum_{i=1}^2 ((m_{b,e,i} + m_{m,e,i} \\
 & + m_{t,e,i} + m_{struc,i} m_{f,i} + m_{ox,i}))) \\
 & + m_{inter} + m_{adapt} + m_{fairing} + m_{pay}
 \end{aligned} \tag{A.1.1}$$

$$\begin{aligned}
 x_{cg,I} = & \\
 & ((m_I l_I - (2m_{MX} L_{MX} + m_{M10} L_{M10} + m_I L_{eng} \\
 & + m_{f,1} L_{f,1} + m_{m,e,1} L_{m,e,1} + m_{ox,1} L_{ox,1} + m_{t,e,1} L_{t,e,1} \\
 & + m_{struc,1} L_{struc,1} + m_{inter} L_{inter} + m_{M10} L_{M10,2} + m_{b,e,2} L_{b,e,2} \\
 & + m_{f,2} L_{f,2} + m_{b,e,2} L_{b,e,2} + m_{ox,2} L_{ox,2} + m_{t,e,2} L_{t,e,2} + m_{struc,2} L_{struc,2} \\
 & + m_{pay} L_{pay} + m_{fairing} L_{fairing} + m_{pl} L_{pl})) / (m_I)
 \end{aligned} \tag{A.1.2}$$

$$L_{mx} = \frac{l_{MX}}{2} \tag{A.1.3}$$

$$L_{M10,1} = \frac{l_{M10}}{2} \tag{A.1.4}$$

$$L_{eng} = \max(l_{MX}, l_{M10}) \tag{A.1.5}$$

$$L_{ox,1} = \frac{h_{ox,1}}{2} + L_{eng} \tag{A.1.6}$$

$$L_{b,e,1} = L_{eng} + l_{ox,1} \tag{A.1.7}$$

$$L_{f,1} = \frac{h_{f,1}}{2} + L_{b,e,1} \tag{A.1.8}$$

$$L_{t,e,1} = L_{b,e,1} + l_{inter} \quad (\text{A.1.9})$$

$$L_{struc,1} = L_{eng} + \frac{l_{struc,1}}{2} \quad (\text{A.1.10})$$

$$L_{inter} = \frac{l_{inter}}{2} + L_{t,e,1} \quad (\text{A.1.11})$$

$$L_{b,e,2} = L_{t,e,1} + l_{inter} \quad (\text{A.1.12})$$

$$L_{ox,2} = \frac{h_{ox,2}}{2} + L_{b,e,2} \quad (\text{A.1.13})$$

$$L_{b,e,2} = l_{ox,2} + L_{b,e,2} \quad (\text{A.1.14})$$

$$L_{f,2} = L_{b,e,2} + \frac{h_{f,2}}{2} \quad (\text{A.1.15})$$

$$L_{t,e,2} = L_{b,e,2} + l_{f,2} \quad (\text{A.1.16})$$

$$L_{struc,2} = L_{inter} + \frac{l_{inter}}{2} + \frac{l_{struc,2}}{2} \quad (\text{A.1.17})$$

$$L_{pay} = L_{inter} + \frac{l_{inter}}{2} + l_{struc,2} + \frac{l_{pay}}{2} \quad (\text{A.1.18})$$

$$L_{fairing} = L_{inter} + \frac{l_{inter}}{2} + l_{struc,2} + l_{pay} + \frac{l_{fairing}}{3} \quad (\text{A.1.19})$$

$$L_{pay} = L_{fairing} \quad (\text{A.1.20})$$

$$L_I = L_{pay} + \frac{2}{3} l_{pay} \quad (\text{A.1.21})$$

## A.2 NASA CEA Run Database

### A.2.1 M10 Engine

$P_c$ (Pa)	$D_e$ (m)	$P_e$ (Pa)	$I_{sp}$ Error (%)
493460	0.620621	416.6571	-0.57579
493460	0.621999	416.8333	-0.55669
493460	0.639181	417.7875	-0.53526
986920	0.646423	904.2286	-0.53202
986920	0.647573	906.35	-0.52932
986920	0.65617	909.5889	-0.51794
1480400	0.664067	1384.2	-0.51654
1480400	0.665138	1418.3	-0.49825
1480400	0.67809	1431.033	-0.49681
1973800	0.685389	1928.886	-0.4887
1973800	0.686377	1935.217	-0.47928
1973800	0.698205	1949.5	-0.47701
2467300	0.706471	2459.043	-0.4487
2467300	0.707624	2464.9	-0.44542

2467300	0.724485	2495.443	-0.43522
2960800	0.732326	2997	-0.42639
2960800	0.733123	3002.033	-0.42141
2960800	0.749632	3047.214	-0.41472
3454200	0.751593	3536.7	-0.41106
3454200	0.759395	3549.3	-0.40114
3454200	0.779668	3558.613	-0.39962
3947700	0.781794	3969.071	-0.38883
3947700	0.789535	4165.757	-0.38072
3947700	0.816001	4195.183	-0.36099
4441200	0.823639	4647.3	-0.35596
4441200	0.824797	4735.114	-0.35335
4441200	0.861284	4763.7	-0.35224
4934600	0.863481	5036.471	-0.3374
4934600	0.875561	5189.114	-0.3202
4934600	0.899157	5309.014	-0.30915
5428100	0.908355	5571.486	-0.30705
5428100	0.92052	5748.029	-0.29923
5428100	0.948899	5781	-0.29075
5921500	0.951448	6305.614	-0.24636
5921500	0.959421	6325.55	-0.23571
5921500	1.010387	6459.3	-0.21867
6415000	1.013564	6889.333	-0.21719
6415000	1.035662	7043.543	-0.20611
6415000	1.094998	7086.767	-0.1882
6908500	1.096279	7451.517	-0.17983
6908500	1.09802	7626.5	-0.17946
6908500	1.182894	7671.4	-0.17142
7401900	1.19081	7998.9	-0.14594
7401900	1.191702	8018.5	-0.12733
7401900	1.305517	8214.9	-0.11593
7895400	1.31272	8594.333	-0.09971
7895400	1.314116	8625.9	-0.06591
7895400	1.474592	8857.233	-0.064
8388800	1.47945	9136.529	-0.05682
8388800	1.481619	9166.2	-0.0483
8388800	1.71818	9389.6	-0.01301
8882300	1.72507	9737.467	0.050444
8882300	1.74397	9772.825	0.093359
8882300	2.147836	10039.43	0.143824
9375800	2.151275	10274.86	0.155041
9375800	2.153537	10316.05	0.172127
9375800	3.150404	10580.91	0.287088
9869200	3.153645	11178.61	0.289101
9869200	3.154245	11234.55	0.299995

Table A.1: M10 Engine sizing suitable candidates.

### A.3 MX Engine

$P_c$ (Pa)	$D_e$ (m)	$P_e$ (Pa)	$I_{sp}$ Error (%)
493000	0.589972	2690	-8.06462

493000	0.606165	3380	-7.57434
987000	0.62259	4110	-7.50978
987000	0.640677	5580	-7.50136
987000	0.669982	8970	-7.13124
1480000	0.69163	10720	-6.82925
1970000	0.715105	11370	-6.50812
1970000	0.7424	13933.33	-6.49355
2470000	0.772378	14460	-4.5422
2470000	0.774758	16450	-4.50034
2960000	0.803284	17350	-4.44131
2960000	0.812388	17500	-4.31498
2960000	0.815941	18500	-4.27255
3450000	0.829504	18800	-4.25732
3950000	0.858589	24700	-4.15153
4440000	0.864882	25100	-3.96295
4440000	0.880555	28600	-3.94847
4440000	0.90464	29400	-3.83792
4930000	0.958281	29600	-3.4344
4930000	0.995	31000	-3.25067
4930000	1.011656	33300	-3.16099
5430000	1.040732	34300	-2.38609
5430000	1.069103	34500	-0.12567
5430000	1.086031	40966.67	0.029593
5920000	1.099583	46033.33	0.120111
6420000	1.123258	46300	0.158267
6910000	1.186343	47900	0.173561
7400000	1.227845	51133.33	0.179192
7400000	1.34615	52300	0.184089
7900000	1.386152	53900	0.223261
7900000	1.403545	55400	0.25369
8390000	1.426183	57300	0.317805
8390000	1.432619	59800	0.346156
8390000	1.559837	63600	0.35975
8880000	1.626339	68933.33	0.361474
8880000	1.760612	74233.33	0.369802
8880000	1.768171	79466.67	0.410428
9380000	2.018236	84800	0.543742
9380000	2.499681	93200	0.557367
9380000	2.952031	98700	0.600822
9870000	3.132548	104000	0.795794
9870000	3.573345	110000	0.894548

Table A.2: MX Engine sizing suitable candidates.



## Appendix B

# Pseudocode

### B.1 Control allocation

#### B.1.1 $P_a$ thrust allocation scheme

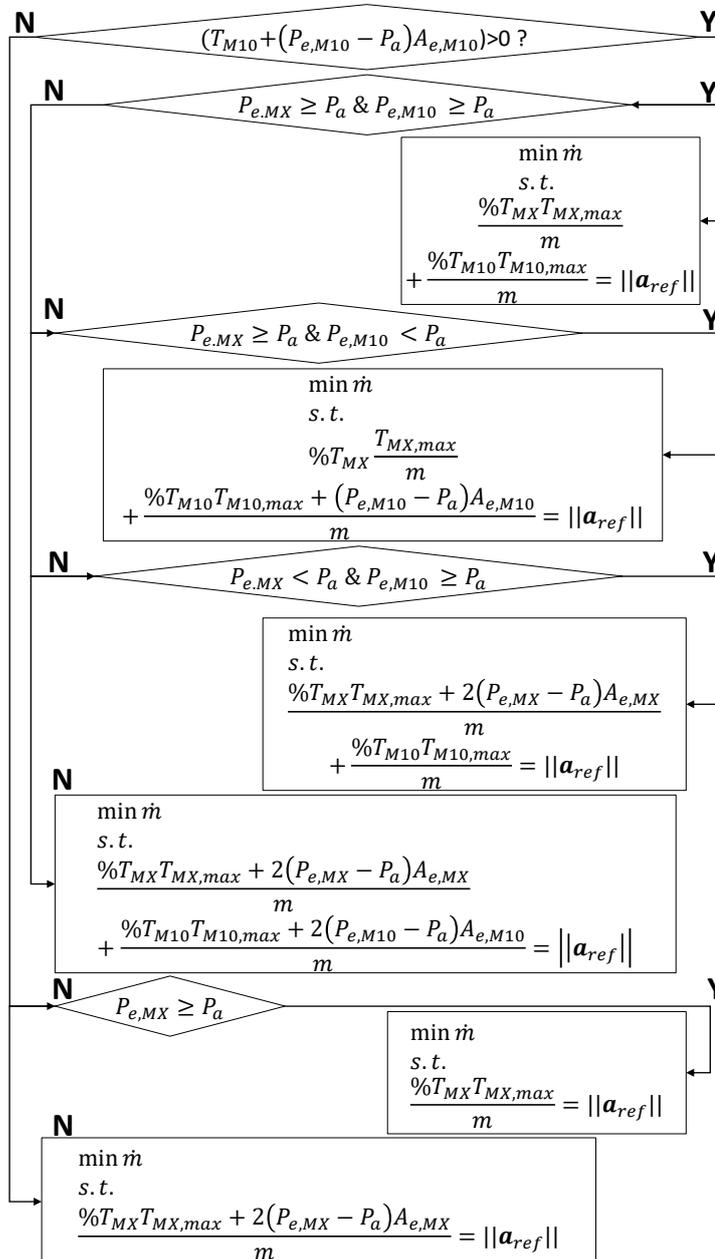


Figure B.1: Thrust control allocation scheme pseudo-code for optimization of minimized mass consumption based on maximum achievable thrust with pressure losses.

**B.1.2 Engine ignition thrust allocation scheme**

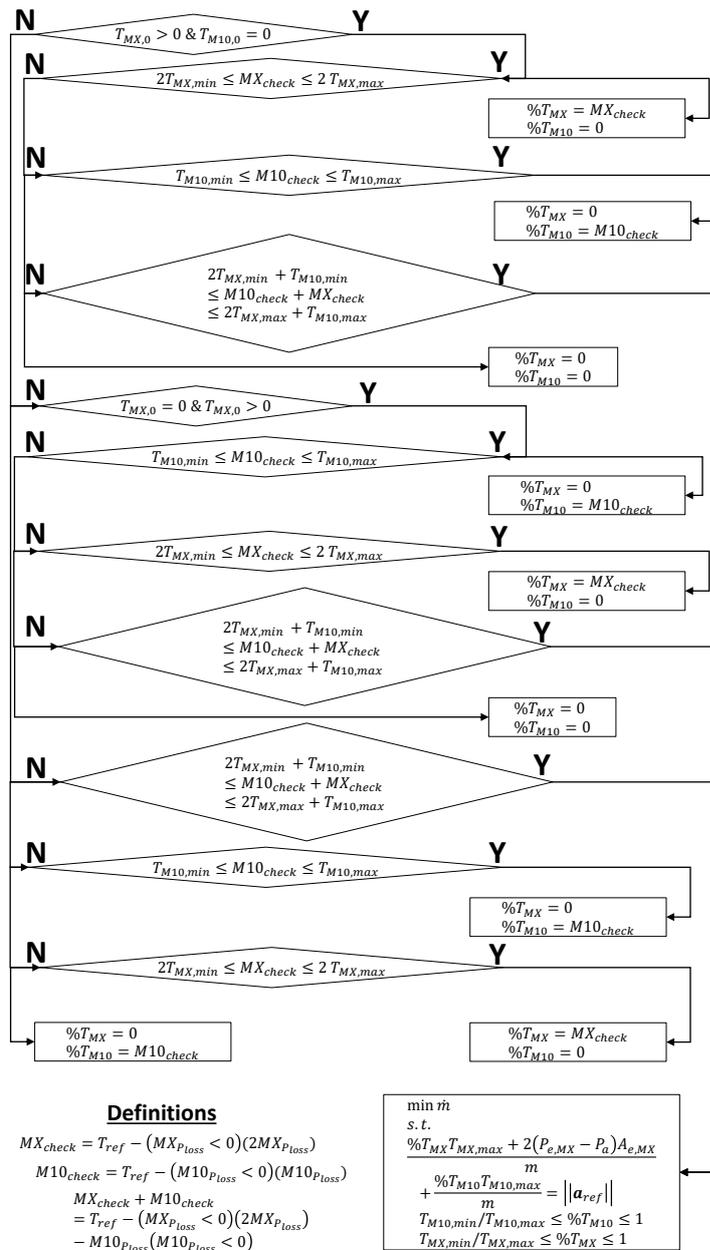


Figure B.2: Thrust control allocation scheme pseudo-code with prioritization for engines that are already switched on.

## B.2 DESCO

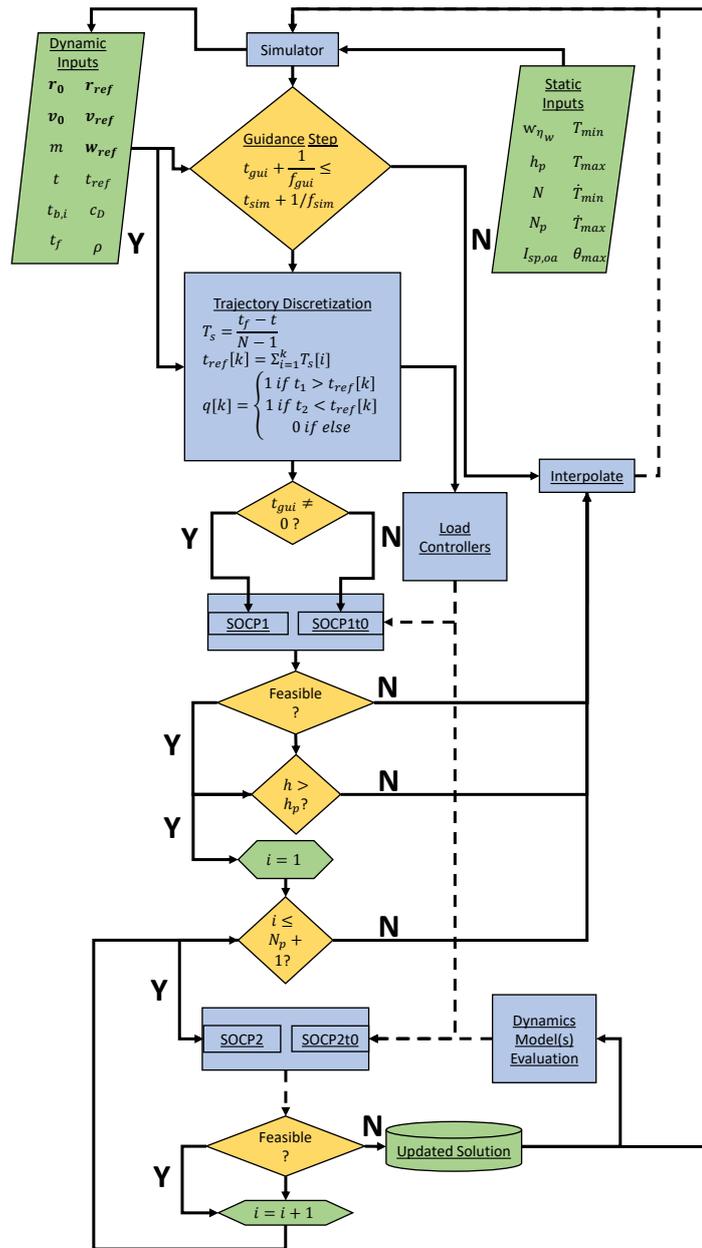


Figure B.3: DESCO pseudocode.

### B.3 Simulator

#### B.3.1 SDK Block

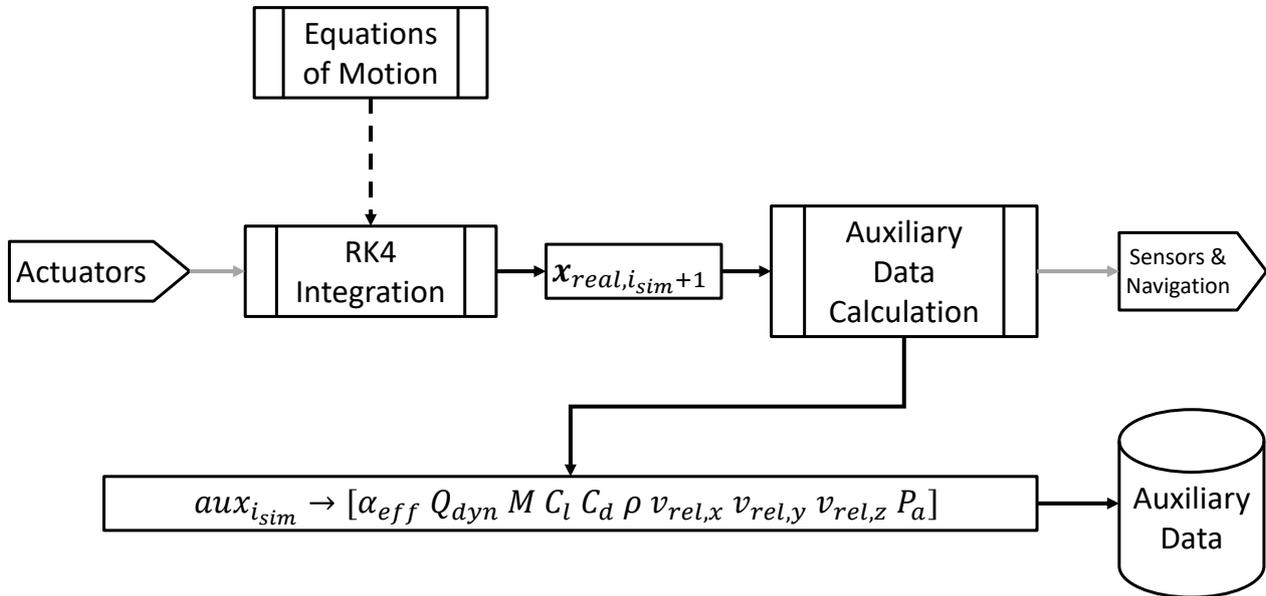


Figure B.4: SDK block pseudocode.

#### B.3.2 SN Block

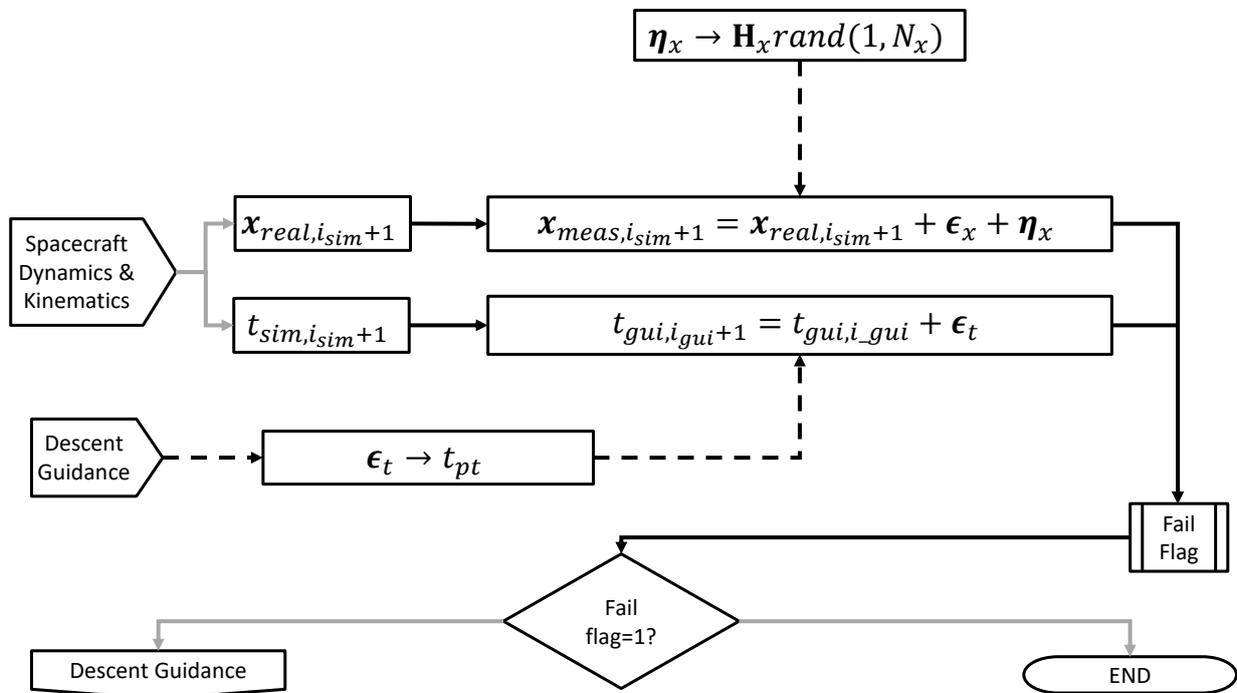


Figure B.5: SN block pseudocode.

## B.4 Algorithms

All algorithms from this section have been adapted from the work of [54].

### B.4.1 SOCP1t0

$$\begin{aligned}
 & \min(\mathbf{w}, \sigma) - z[N], \text{ s.t.} \\
 & \quad \underline{\text{Boundary conditions}} \\
 & \quad z[1] = \log \hat{m}(t), \mathbf{r}[1] = \hat{\mathbf{r}}(t), \mathbf{v}[1] = \hat{\mathbf{v}}(t) \\
 & \quad \mathbf{r}[N] = \mathbf{r}_f, \mathbf{v}[N] = \mathbf{v}_f, \mathbf{w}_{x,y}[N] = \mathbf{0}, \mathbf{w}_z[N] \geq \mathbf{0} \\
 & \quad \underline{\text{Dynamics equations}}, \forall k \in [1, \dots, N-1] \\
 & \quad \mathbf{r}[k+1] = \mathbf{r}[k] + T_s \mathbf{v}[k] + \frac{T_s^2}{3} (\mathbf{a}[k] + \frac{\mathbf{a}[k+1]}{2}) \\
 & \quad \mathbf{v}[k+1] = \mathbf{v}[k] + \frac{T_s}{2} (\mathbf{a}[k] + \mathbf{a}[k+1]) \\
 & \quad z[k+1] = z[k] - \frac{1}{I_{sp} g_0} \frac{T_s}{2} (\sigma[k] + \sigma[k+1]) \\
 & \quad \underline{\text{Surrogate variables}}, \forall k \in [1, \dots, N] \\
 & \quad \mathbf{a}[k] = \mathbf{w}[k] + \hat{\mathbf{g}}(t) \\
 & \quad \|\mathbf{w}[k]\| \leq \sigma[k] \\
 & \quad \underline{\text{Control constraints}}, \forall k \in [1, \dots, N-1] \\
 & \quad \begin{cases} \mathbf{w}_z[k] \geq \frac{\|\mathbf{w}_{x,y}[k]\|}{\theta_{max}}, & \text{if } T_s(k-1) \in \mathcal{T}_P \\ \mathbf{w}[k] = \mathbf{0}, & \text{else} \end{cases} \\
 & \quad \underline{\text{Control rate constraints}}, \forall k \in [1, \dots, N-1] \\
 & \quad \sigma[k] - T_s \frac{\dot{T}_{max}}{\hat{m}(t)} \leq \sigma[k+1] \leq \sigma[k] + T_s \frac{\dot{T}_{max}}{\hat{m}(t)}
 \end{aligned} \tag{B.4.1}$$

## B.4.2 SOCP1

$$\begin{aligned}
& \min(\mathbf{w}, \sigma) - z[N], \text{ s.t.} \\
& \quad \underline{\text{Boundary conditions}} \\
& z[1] = \log \hat{m}(t), \mathbf{r}[1] = \hat{\mathbf{r}}(t), \mathbf{v}[1] = \hat{\mathbf{v}}(t), \mathbf{w}[1] = \hat{\mathbf{w}}(t) \\
& \mathbf{r}[N] = \mathbf{r}_f, \mathbf{v}[N] = \mathbf{v}_f, \mathbf{w}_{x,y}[N] = \mathbf{0}, \mathbf{w}_z[N] \geq \mathbf{0} \\
& \quad \underline{\text{Dynamics equations}}, \forall k \in [1, \dots, N-1] \\
& \mathbf{r}[k+1] = \mathbf{r}[k] + T_s \mathbf{v}[k] + \frac{T_s^2}{3} (\mathbf{a}[k] + \frac{\mathbf{a}[k+1]}{2}) \\
& \quad \mathbf{v}[k+1] = \mathbf{v}[k] + \frac{T_s}{2} (\mathbf{a}[k] + \mathbf{a}[k+1]) \\
& z[k+1] = z[k] - \frac{1}{I_{sp} g_0} \frac{T_s}{2} (\sigma[k] + \sigma[k+1]) \\
& \quad \underline{\text{Surrogate variables}}, \forall k \in [1, \dots, N] \\
& \quad \mathbf{a}[k] = \mathbf{w}[k] + \hat{\mathbf{g}}(t) \\
& \quad \|\mathbf{w}[k]\| \leq \sigma[k] \\
& \quad \underline{\text{Control constraints}}, \forall k \in [1, \dots, N-1] \\
& \left\{ \begin{array}{l} \mathbf{w}_z[k] \geq \frac{\|\mathbf{w}_{x,y}[k]\|}{\theta_{max}}, | : \frac{T_{min}}{\hat{m}(t)} \leq \sigma[k] \leq \frac{T_{max}}{\hat{m}(t)}, \text{ if } T_s(k-1) \in \mathcal{T}_P \\ \mathbf{w}[k] = \mathbf{0}, \text{ else} \end{array} \right. \\
& \quad \underline{\text{Control rate constraints}}, \forall k \in [1, \dots, N-1] \\
& \sigma[k] - T_s \frac{\dot{T}_{max}}{\hat{m}(t)} \leq \sigma[k+1] \leq \sigma[k] + T_s \frac{\dot{T}_{max}}{\hat{m}(t)}
\end{aligned} \tag{B.4.2}$$

### B.4.3 SOCP2t0

$$\begin{aligned}
& \min(\mathbf{w}, \sigma) - z[N] + \mathcal{W}_{\eta_w} \sum_{k=1}^N \eta_w[k], \text{ s.t.} \\
& \quad \underline{\text{Boundary conditions}} \\
& \quad z[1] = \log \hat{m}(t), \mathbf{r}[1] = \hat{\mathbf{r}}(t), \mathbf{v}[1] = \hat{\mathbf{v}}(t) \\
& \quad \mathbf{r}[N] = \mathbf{r}_f, \mathbf{v}[N] = \mathbf{v}_f, \mathbf{w}_{x,y}[N] = \mathbf{0}, \mathbf{w}_z[N] \geq \mathbf{0} \\
& \quad \underline{\text{Dynamics equations}}, \forall k \in [1, \dots, N-1] \\
& \quad \mathbf{r}[k+1] = \mathbf{r}[k] + T_s \mathbf{v}[k] + \frac{T_s^2}{3} (\mathbf{a}[k] + \frac{\mathbf{a}[k+1]}{2}) \\
& \quad \mathbf{v}[k+1] = \mathbf{v}[k] + \frac{T_s}{2} (\mathbf{a}[k] + \mathbf{a}[k+1]) \\
& \quad z[k+1] = z[k] - \frac{1}{I_{sp} g_0} \frac{T_s}{2} (\sigma[k] + \sigma[k+1]) \\
& \quad \underline{\text{Surrogate variables}}, \forall k \in [1, \dots, N] \\
& \quad \mathbf{a}[k] = \mathbf{w}[k] + \hat{\mathbf{g}}(t) \\
& \quad \|\mathbf{w}[k]\| \leq \sigma[k] \\
& \quad \underline{\text{Trust region constraints}}, \forall k \in [1, \dots, N] \\
& \quad \|\mathbf{w}[k] - \mathbf{w}_{i_{cvx}}^*[k]\| \leq \eta_w[k] \\
& \quad \underline{\text{Flight path constraints}}, \forall k \in [1, \dots, N-1] \\
& \quad \mathbf{r}_z[k] / \text{geq} \frac{\hat{\mathbf{r}}_z(t)}{\|\hat{\mathbf{r}}_{x,y}(t)\|} \|\mathbf{r}_{x,y}[k]\| \\
& \quad \underline{\text{Control constraints}}, \forall k \in [1, \dots, N-1] \\
& \quad \begin{cases} \mathbf{w}_z[k] \geq \frac{\|\mathbf{w}_{x,y}[k]\|}{\theta_{max}}, & \text{if } T_s(k-1) \in \mathcal{T}_P \\ \mathbf{w}[k] = \mathbf{0}, & \text{else} \end{cases} \\
& \quad \underline{\text{Control rate constraints}}, \forall k \in [1, \dots, N-1] \\
& \quad \sigma[k] - T_s \frac{\dot{T}_{max}}{\hat{m}(t)} \leq \sigma[k+1] \leq \sigma[k] + T_s \frac{\dot{T}_{max}}{\hat{m}(t)}
\end{aligned} \tag{B.4.3}$$

## B.4.4 SOCP2

$$\begin{aligned}
& \min(\mathbf{w}, \sigma) - z[N] + \mathcal{W}_{\eta_w, \Sigma_{k=1}^N} \eta_w[k], \text{ s.t.} \\
& \quad \text{Boundary conditions} \\
& z[1] = \log \hat{m}(t), \mathbf{r}[1] = \hat{\mathbf{r}}(t), \mathbf{v}[1] = \hat{\mathbf{v}}(t), \mathbf{w}[1] = \hat{\mathbf{w}}(t) \\
& \mathbf{r}[N] = \mathbf{r}_f, \mathbf{v}[N] = \mathbf{v}_f, \mathbf{w}_{x,y}[N] = \mathbf{0}, \mathbf{w}_z[N] \geq 0 \\
& \quad \text{Dynamics equations, } \forall k \in [1, \dots, N-1] \\
& \mathbf{r}[k+1] = \mathbf{r}[k] + T_s \mathbf{v}[k] + \frac{T_s^2}{3} (\mathbf{a}[k] + \frac{\mathbf{a}[k+1]}{2}) \\
& \quad \mathbf{v}[k+1] = \mathbf{v}[k] + \frac{T_s}{2} (\mathbf{a}[k] + \mathbf{a}[k+1]) \\
& z[k+1] = z[k] - \frac{1}{I_{sp} g_0} \frac{T_s}{2} (\sigma[k] + \sigma[k+1]) \\
& \quad \text{Surrogate variables, } \forall k \in [1, \dots, N] \\
& \quad \mathbf{a}[k] = \mathbf{w}[k] + \hat{\mathbf{g}}(t) \\
& \quad \|\mathbf{w}[k]\| \leq \sigma[k] \\
& \quad \text{Trust region constraints, } \forall k \in [1, \dots, N] \\
& \quad \|\mathbf{w}[k] - \mathbf{w}_{i_{cvx}} * [k]\| \leq \eta_w[k] \\
& \quad \text{Flight path constraints, } \forall k \in [1, \dots, N-1] \\
& \quad \mathbf{r}_z[k] / g e q \frac{\hat{\mathbf{r}}_z(t)}{\|\hat{\mathbf{r}}_{x,y}(t)\|} \|\mathbf{r}_{x,y}[k]\| \\
& \quad \text{Control constraints, } \forall k \in [1, \dots, N-1] \\
& \quad \begin{cases} \mathbf{w}_z[k] \geq \frac{\|\mathbf{w}_{x,y}[k]\|}{\theta_{max}}, & \text{if } \frac{T_{min}}{\hat{m}(t)} \leq \sigma[k] \leq \frac{T_{max}}{\hat{m}(t)}, \\ \mathbf{w}[k] = \mathbf{0}, & \text{else} \end{cases} \\
& \quad \text{Control rate constraints, } \forall k \in [1, \dots, N-1] \\
& \quad \sigma[k] - T_s \frac{\dot{T}_{max}}{\hat{m}(t)} \leq \sigma[k+1] \leq \sigma[k] + T_s \frac{\dot{T}_{max}}{\hat{m}(t)}
\end{aligned} \tag{B.4.4}$$



# Bibliography

- [1] Fundamental physical constants. URL <https://physics.nist.gov/cgi-bin/cuu/Value?r>. Last accessed September 2020.
- [2] Rocket propellants, 2008. URL <http://braeunig.us/space/propel.htm>. Last accessed September 2020.
- [3] M10 engine, 2020. URL <https://www.avio.com/m-10>. Last accessed September 2020.
- [4] Vega c light, 2020. URL <https://www.avio.com/vega-future-applications>.
- [5] B. Açıkmeşe and S. Ploen. A powered descent guidance algorithm for mars pinpoint landing. *AIAA Guidance, Navigation, and Control Conf and Exhibit*, 2005. URL <https://doi.org/10.2514/6.2005-6288>.
- [6] B. Açıkmeşe and S. Ploen. Convex programming approach to powered descent guidance for mars landing. *J Guid Contr Dynam* 30, pages 1353–1366, 2007.
- [7] M. N. Berberan-Santos, E. N. Bodunov, and L. Pogliani. On the barometric formula.
- [8] L. Blackmore, B. Açıkmeşe, and D. Scharf. Minimum-landing-error powered-descent guidance for mars landing using convex optimization. *J Guid Contr Dynam* 33, pages 1161–1171, 2010.
- [9] P. Blau. Falcon 9 v1.1 & f9r launch vehicle overview. URL <http://www.spaceflight101.net/falcon-9-v11.html>. Last accessed October 2020.
- [10] H. Burkhardt, M. Sippel, A. Herbertz, and J. Klevanski. Kerosene vs methane: A propellant tradeoff for reusable liquid booster stages. *Journal of Spacecraft and Rockets*, 2004. URL <https://arc.aiaa.org/doi/pdf/10.2514/1.2672>.
- [11] L. Bussler, J. Wilken, S. Stappert, M. Sippel, I. Dietlein, and E. Dumont. Assessment of vtvl and vthl reusable first stages. *HiSST: International Conference on High-Speed Vehicle Science Technology*, 2018.
- [12] L. Bussler, J. Wilken, S. Stappert, M. Sippel, I. Dietlein, and E. Dumont. New control functionalities for launcher load relief in ascent and descent flight. *8TH EUROPEAN CONFERENCE FOR AERONAUTICS AND AEROSPACE SCIENCES (EUCASS)*, 2019.
- [13] F. Castellini. Multidisciplinary design optimization for expendable launch vehicles. *PhD thesis*, 2012.
- [14] S. Contant. Design and optimization of a small reusable launch vehicle using vertical landing techniques. URL <https://repository.tudelft.nl/islandora/object/uuid:58cb0a6e-e9ac-4ce5-9051-5ddd863e98a8>.
- [15] Jian S. Dai. Euler–rodriques formula variations, quaternion conjugation and intrinsic connections. *Mechanism and Machine Theory*, 2015.
- [16] A.P. de Mirand, J. Bahu, and O. Gogdet. Ariane next, a vision for the next generation of ariane launchers. *Acta Astronautica* 170, pages 735–749, 2020.
- [17] J. Desmariaux, E. Bourgeois, A.P. de Mirand, J. Bahu, and P. Baiocco. Investigations on flight control for rocket first stage recovery. *67<sup>th</sup> Int Astronautical Congress*, 2016.

- [18] G. Di Campli. Models of the earth's upper atmosphere. *COSPAR International Reference Atmosphere*, 2012.
- [19] G. Di Campli. Vega launchers' trajectory optimization using a pseudospectral transcription. *TU Delft Master Thesis of Aerospace Engineering*, 2017.
- [20] E. Dumont, T. Ecker, C. Chavagnac, L. Witte, J. Windelberg, and J. Klevanski. Callisto - reusable vtlv launcher first stage demonstrator. 2018. URL [https://elib.dlr.de/119728/1/406\\_DUMONT.pdf](https://elib.dlr.de/119728/1/406_DUMONT.pdf).
- [21] T. Ecker, F. Zilker, E. Dumont, Karl, Sebastian, and K. Hannemann. Aerothermal analysis of reusable launcher systems during retro-propulsion reentry and landing. 2018. URL <https://elib.dlr.de/120072/>.
- [22] L. Furfaro and R. Linares. Waypoint-based generalized zem/zev feedback guidance for planetary landing via a reinforcement learning approach. *3<sup>rd</sup> IAA Conf on Dynamics and Control of Space Systems*, pages 401–416, 2017.
- [23] R. Furfaro, S. Selnick, and M. Cupples. Nonlinear sliding guidance algorithms for precision lunar landing. *21<sup>st</sup> AAS/AIAA Space Flight Mechanics Meeting*, pages 945–964, 2020.
- [24] B. Gaudet, R. Linares, and R. Furfaro. Integrated guidance and control for pinpoint mars landing using reinforcement learning. *Adv Astron Sci*, pages 3135–3154, 2018.
- [25] B. Gaudet, R. Linares, and R. Furfaro. Deep reinforcement learning for six degree-of-freedom planetary landing. *Advances in Space Research*, pages 1725–1741, 2020. URL <https://doi.org/10.1109/TAC.1964.1105758>.
- [26] M. Hawkins, Y. Guo, and B. Wie. Spacecraft guidance algorithms for asteroid intercept and rendezvous missions. *International Journal of Aeronautical and Space Sciences*, 13:154–169, 2012.
- [27] X. Jiang, R. Furfaro, and S. Li. Integrated guidance for mars entry and powered descent using reinforcement learning and gauss pseudospectral method. *4<sup>th</sup> IAA Conf on Dynamics and Control of Space Systems*, pages 761–774, 2018.
- [28] X. Jiang, R. Furfaro, and S. Li. Integrated guidance for mars entry and powered descent using reinforcement learning and gauss pseudospectral method. *4<sup>th</sup> IAA Conf on Dynamics and Control of Space Systems*, pages 761–774, 2018.
- [29] D. Kajon, D. Liuzzi, M. Rudnykh, D. Drigo, L. Arione, N. Ierardo, and A. Sirbi. Development of the liquid oxygen and methane m10 rocket engine for the vega-e upper stage. *8<sup>TH</sup> EUROPEAN CONFERENCE FOR AERONAUTICS AND SPACE SCIENCES (EUCASS)*, 2019. URL <https://www.eucass.eu/doi/EUCASS2019-0315.pdf>.
- [30] A. Klumpp. Apollo lunar descent guidance. *Automatica* 10, pages 133–146, 1974. URL [https://doi.org/10.1016/0005-1098\(74\)90019-3](https://doi.org/10.1016/0005-1098(74)90019-3).
- [31] Mansfield C. L. Sts-135: The final voyage. URL [https://www.nasa.gov/mission\\_pages/shuttle/shuttlemissions/sts135/launch/sts-135\\_mission-overview.html](https://www.nasa.gov/mission_pages/shuttle/shuttlemissions/sts135/launch/sts-135_mission-overview.html).
- [32] Wade M. Kistler k-1, . URL <http://www.astronautix.com/k/kistlerk-1.html>.
- [33] Wade M. x-30, . URL <http://www.astronautix.com/x/x-30.html>.
- [34] Wade M. x-33, . URL <http://www.astronautix.com/x/x-33.html>.
- [35] Wade M. x-34, . URL <http://www.astronautix.com/x/x-34.html>.
- [36] Wade M. x-37, . URL <http://www.astronautix.com/x/x-37.html>.
- [37] Wall M. Wow! spacex lands orbital rocket successfully in historic first,, . URL <http://www.space.com/31420-spacex-rocket-landing-success.html>.

- [38] L. Ma, Z. Shazo, and W. Chen. Trajectory optimization for lunar soft landing with a hamiltonian-based adaptive mesh refinement strategy. *Adv Eng Softw* 100, pages 266–276, 2016. URL <https://doi.org/10.1016/j.advengsoft.2016.08.002>.
- [39] L. Ma, Z. Shazo, and W. Chen. Trajectory optimization for lunar rover performing vertical takeoff vertical landing maneuvers in the presence of terrain. *Acta Astron* 146, pages 289–299, 2018. URL <https://doi.org/10.1016/j.actaastro.2018.03.013>.
- [40] L. Ma, K. Wang, and Z. Xu. Trajectory optimization for powered descent and landing of reusable rockets with restartable engines. *69<sup>th</sup> Int Astronautical Congress*, 2018.
- [41] J. Meditch. On the problem of optimal thrust programming for a lunar soft landing. *IEEE Trans Autom Contr*, pages 477–484, 1964. URL <https://doi.org/10.1109/TAC.1964.1105758>.
- [42] F. Mirande. Design optimization of ground and air-launched hybrid rockets. *Master of Science Thesis in Aerospace Engineering*, 2015.
- [43] J. A. Mulder, W. H. J. J. Van Staveren, J. C. Van der Vaart, E. De Weerd, C. C. De Visser, A. C. In't Veld, and E. Mooij. *Lecture Notes AE3202, Flight Dynamics*. TU Delft, 2013.
- [44] Blue Origin. Introducing new glenn. URL <https://www.youtube.com/channel/UCVxTHEKKLxNjGcvVaZindlg>.
- [45] J. M. Picone, A. E. Hedin, and D. P. Drob. Nrlmsise-00 empirical model of the atmosphere: statistical comparisons and scientific issues. *Journal of Geophysical Research*, 107, 2002. URL <https://doi.org/10.1029/2002JA009430>.
- [46] S. Ploen, B. Açıkmeşe, and A. Wolf. A comparison of powered descent guidance laws for mars pinpoint landing. *AIAA/AAS Astrodynamics Specialist Conf and Exhibit*, 2006.
- [47] R. Prakash, P. Burkhart, and A. Chen. Mars science laboratory entry, descent, and landing system overview. *IEE Aerospace Conf*, pages 1–18, 2008. URL <https://doi.org/10.1109/AERO.2008.4526283>.
- [48] H.W. Price, R.D. Braun, R. Manning, and E. Sklyanski. A high-heritage blunt-body entry, descent, and landing concept for human mars exploration. *54<sup>th</sup> AIAA Aerospace Sciences Meeting*, pages 1–16.
- [49] Webb R. Is it worth it? the economics of reusable space transportation. *ICEAA 2016 International Training Symposium*, 2016.
- [50] M. Rozemeijer. Toroidal staged microlaunch vehicle. *Master of Science Thesis in Aerospace Engineering*, 2020.
- [51] Knauer J. S. Reusability — the next major launch vehicle development. *SAE Transactions, 1967, Vol. 75, SECTION 2: Papers 660167–660458 (1967)*, pages 892–901, 1967.
- [52] M. Sagliano. Generalized hp pseudospectral-convex programming for powered descent and landing. *JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS Vol. 42, No. 7*, pages 1562–1570, 2019.
- [53] P. Simplício, A. Marcos, E. Joffre, M. Zamaro, and N. Silva. Review of guidance techniques for landing on small bodies. *Progress in Aerospace Sciences*, 2018. URL <https://doi.org/10.1016/j.paerosci.2018.10.005>.
- [54] P. Simplício, A. Marcos, and S. Bennani. Guidance of reusable launchers: Improving descent and landing performance. *Journal of Guidance, Control and Dynamics*, 2019.
- [55] G. M. Siouris. *Missile Guidance and Control Systems*. Springer, 2003.
- [56] Z. Song, D. Zhao, and X. Lu. Terminal attitude constrained guidance and control for lunar soft landing. *Acta Astron Sci* 153, pages 137–147, 2015.
- [57] Z. Song, C. Wang, S. Theil, D. Seelbinder, M. Sagliano, X. Liu, and Z. Shao. Survey of autonomous guidance methods for powered planetary landing. *Springer Nature 2020*, 2020.

- [58] R. Sostaric and J. Rea. Powered descent guidance methods for the moon and mars. *AIAA Guidance, Navigation, and Control Conf and Exhibit*, 2005. URL <https://doi.org/10.2514/6.2005-6287>.
- [59] S. Stappert, J. Wilken, M. Sippel, and I. Dietlein. Evaluation of european reusable vtl booster stages. *2018 AIAA SPACE and Astronautics Forum and Exposition*, 2018.
- [60] M. Szmuk, U. Eren, and B. Aıkmee. Successive convexification for mars 6-dof powered descent landing guidance. *AIAA Guidance, Navigation, and Control Conf*, 2018. URL <https://doi.org/10.2514/6.2017-1500>.
- [61] P.V. Tartabini, J.R. Beaty, R.A. Lepsch, and M.G. Gilbert. Payload performance analysis for a reusable two-stage-to-orbit vehicle.
- [62] U. Topcu, J. Casoliva, and K. Mease. Fuel efficient powered descent guidance for mars landing. *AIAA Guidance, Navigation, and Control Conf and Exhibit*, 2005. URL <https://doi.org/10.2514/6.2005-6286>.
- [63] U. Topcu, J. Casoliva, and K. Mease. Minimum-fuel powered descent for mars pinpoint landing. *J Spacecr Rock 44*, pages 324–331, 2007. URL <https://doi.org/10.2514/1.25023>.
- [64] M. Van Kesteren. Air launch versus ground launch: a multidisciplinary design optimization study of expendable launch vehicles on cost and performance. *Master of Science Thesis in Aerospace Engineering*, 2013.
- [65] J. Vandamme. Assisted-launch performance analysis using trajectory and vehicle optimization. *Master of Science Thesis in Aerospace Engineering*, 2012.
- [66] K. F. Wakker. *Fundamentals of Astrodynamics*. Faculty of Aerospace Engineering, Delft University of Technology.
- [67] C. Wang and Z. Song. Rapid trajectory optimization for lunar soft landing with hazard avoidance. *Adv Astron Sci 161*, pages 885–900, 2018.
- [68] J. Wilken, S. Stappert, L. Bussler, M. Sippel, and E. Dumont. Future european reusable booster stages evaluation of vthl and vtl return methods. *69th International Astronautical Congress (IAC)*, pages 892–901, 2018.
- [69] B. T. C. Zandbergen. Simple mass and size estimation relationships of pump fed rocket engines for launch vehicle conceptual design. *6<sup>th</sup> European Conference for Aeronautics and Space Sciences (EU-CASS)*, 2015.
- [70] Barry T.C. Zandbergen. *Thermal Rocket Propulsion*. TU Delft, 2018.
- [71] P. Zarchan. *Tactical and strategic missile guidance*. Reston, VA: American Institute of Aeronautics and Astronautics, 2019.
- [72] B. Zhang, S. Tang, and B. Pan. Multi-constrained suboptimal powered descent guidance for lunar pinpoint soft landing. *Aerosp Sci Technol 48*, pages 203–213, 2016.
- [73] H. Zhang, J. Liang, and X. Huang. Autonomous hazard avoidance control for chang’e-3 soft landing. *Advances in Space Research*, pages 559–568, 2014. URL <https://doi.org/10.1360/092014-51>.
- [74] Y. Zhang, Y. Guo, and G. Ma. Collision avoidance zem/zev optimal feedback guidance for powered descent phase of landing on mars. *Adv Space Res 59*, pages 1514–1525, 2017.
- [75] L. Zhou and Y. Xia. Improved zem/zev feedback guidance for mars powered descent phase. *Adv Space Res 54*, pages 2446–2455, 2014.