Autonomous Guidance and Control for Precision Landing on Planetary Bodies

Convex Optimization Approach For Mars and Titan Case Studies

R.O. Mazouz

Faculty of Aerospace Engineering

Jet Propulsion Laboratory
California Institute of Technology

Delft University of Technology

Government Sponsorship Acknowledged

This research was partially carried out at the Jet Propulsion Laboratory, California Institute of Technology, and was sponsored by NASA.

Cover Image: Mars 2020 Mission Overview
https://mars.nasa.gov/mars2020/mission/overview/

Autonomous Guidance and Control for Precision Landing on Planetary Bodies

Convex Optimization Approach For Mars and Titan Case Studies

by

R.O. Mazouz

in fulfillment of the requirements for the degree of

Master of Science
in Aerospace Engineering

at Delft University of Technology,

to be defended publicly on Monday December 2, 2019 at 01:30 PM.

Student number:	4147146	
Project duration:	December 3, 2018 - December 2, 2019	
Supervisors:	Dr. ir. E. Mooij	TU Delft, Astrodynamics and Space Missions
	Dr. ir. M. Quadrelli,	NASA/JPL/Caltech, Robotics Modeling and Simulation
Thesis committee:	Dr. ir. W. van der Wal ,	TU Delft, Astrodynamics and Space Missions (Chair)
	Dr. ir. E. van Kampen,	TU Delft, Control and Simulation

This thesis is confidential and cannot be made public until December 31, 2020.

Jet Propulsion Laboratory
California Institute of Technology

TU Delft Delft University of Technology

Preface

This thesis report is written in fulfillment of the requirements for the degree of Master of Science in Aerospace Engineering at Delft University of Technology. The review provides an in-depth analysis of the engineering challenges of precision landing on planetary bodies, with a particular focus on Mars and Titan. The research that has been conducted at NASA's Jet Propulsion Laboratory (JPL), Department of Robotics Modeling and Simulation of Planetary Vehicles, focused on precision landing on Saturn's moon Titan, while Mars played a significant role in algorithm verification. Precision landing techniques anticipate on forthcoming rocket industry needs as defined by NASA. Achieving interplanetary pinpoint precision landing with an accuracy in the order of meters will greatly benefit the aerospace and technology industry, as well as the science community. Besides the enormous boost in acquired scientific knowledge, precision landing will provide a crucial link for human interplanetary travel in the (near) future.

Through the provision of assorted solutions this written file documents the results and conclusions of the conducted research. I would like to express my sincere gratitude to Dr. ir. E. (Erwin) Mooij at Delft University of Technology and Dr. ir. M. (Marco) B. Quadrelli at NASA's Jet Propulsion Laboratory for their guidance, navigation and collaboration in the process of establishing this thesis. Likewise, I would like to express my heartfelt appreciation to the ones closest to me.

Rayan Mazouz
November 18, 2019

Abstract

Autonomously landing a spacecraft on the surface of a planetary body with a degree of precision in the order of meters is highly challenging. Over the course of time, the landing ellipse, defined as the region with a 99% likelihood of where a space vehicle will land, has improved steadily but currently still has dimensions in the order of kilometers. The first and single Martian spacecraft that has performed a guided atmospheric entry and utilized precision landing technologies is the Mars Science Laboratory (MSL). The MSL probe and its focal point, the Curiosity rover, has thus been the most advanced mission yet to have flown to Mars. Nonetheless, space missions to the Red Planet have thus-far never landed following fuel optimal paths. Similarly, dispersions for landing on Titan with current technologies expand to hundreds of kilometers. The only reference mission to Titan is the Huygens probe, which has not utilized precision landing technologies nor optimal path planning. Besides, the goal of the Cassini-Huygens mission was to maximize descent time to augment scientific data retrieval of Titan's atmosphere. As part of the NASA Space Exploration Technology Directorate, a parafoil is now proposed for landing on Titan due to its cost effectiveness, ease of deployment, relatively low mass compared to the prospective payload and capabilities of precise autonomous delivery.

While considering all phases of Entry, Descent & Landing (EDL) and all elements of Guidance, Navigation & Control (GNC), the central focus of the research was put on the (powered and parafoil) terminal descent phase. The research core concerns a convex optimization programming approach to guarantee soft-landing. Throughout the discussion a distinction is made between lossless and successive convexification for optimal guidance. Both types have been simulated to either compute fuel optimal paths for powered Mars landing or pull-power optimal paths for parafoil Titan landing. By using the full capability of the spacecraft unprecedented precision may be achieved. This will enable engineers and scientists to reach the most alluring places on celestial bodies, thereby providing humanity a deeper understanding of the Universe.

Distinctive from the guidance system, the navigation entity assumes perfect state knowledge. Nonetheless, to cope with the requirements of Terrain Relative Navigation (TRN), a pointing constraint is included in the guidance scheme. The control segment of the GNC system is closed using a Proportional Integral Derivative (PID) controller. Nominal simulations pointed out that soft landing is entrenched while adhering to defined control and state constraints. The simulations have been verified by directly comparing the results to industry's current best practise (for Mars landing). All simulations have been carried out using CVX in Matlab. The obtained results are found to be (near) identical compared to the literature, and hence the designed algorithm has been verified. The active GNC system, for which three sources of errors were induced, accordingly managed to guide the spacecraft to the designated target. The three included sources of error were thrust misalignment, thrust magnitude errors and navigational sensing discrepancies.

As part of the research conducted at NASA/JPL/Caltech, the algorithm has been extended to become compatible with landing on the Saturn's moon Titan. Simultaneously, this allowed for a robustness assessment of the algorithm. A total of three simulation cases were investigated. The first two are based on three degree-of-freedom point-mass dynamics and are differentiated between (1) plain conditions and (2) wind conditions with an enforced glide-slope constraint and a pointing constraint. The latter (3) encompasses high-fidelity 6DoF dynamics with conditions similar to (2). Both the 3DoF and 6DoF simulations show promising results for on-board implementation. The soft-landing is guaranteed while adhering to imposed mission constraints. The control time history remains within desired limits, as well as the aerodynamic angles. The terminal landing is enforced such that the payload lands in parallel direction to the gravity vector.

The algorithm has been extended for obstacle avoidance compatibility based on multi-glideslope constraints. Specifically, when approaching the destination target, roughness of terrain might raise the need for avoiding path obstacles detrimental to the mission. Two extensive sensitivity analyses have been conducted, coupled to parafoil and powered descent. Anew the simulations show promising results for incorporating such guidance system in actual flight. The next steps in future research are the implementation of the 9DoF high-fidelity model with NASA/JPL DARTS (Dynamics Algorithms for Real-Time Simulation) to test for real-time hardware design and visualize terminal descent within the software. Actual testing in C++ is also advised to guarantee fast convergence. Including thrusters and on-board fuel (to a certain allowable mass) is also advised for Titan terminal descent to increase control authority. To avoid stall and destruction of the parafoil through mechanical loads, minimum and maximum velocity magnitudes should be imposed, respectively. These steps, amongst others, will eventually lead to the realization of precision landing on Mars and Titan.

Contents

List of Acronyms

Acronym	Description
AAE	Aeroassist and Atmospheric Entry
AIAA	American Institute of Aeronautics and Astronautics
CVX	Convex Optimization Programming
Caltech	California Institute of Technology
CoG	Center of Gravity
CoM	Center of Mass
DARTS	Dynamics Algorithms for Real-Time Simulation
DSN	Deep Space Network
DoF	Degree of Freedom
EDL	Entry Descent and Landing
ESA	European Space Agency
EoM	Equations of Motion
G-FOLD	Guidance for Fuel Optimal Large Diverts
GNC	Guidance Navigation and Control
GPS	Ground Positioning System
IEEE	Institute of Electrical and Electronics Engineers
IMU	Inertial Measurement Unit
IPM	Interior Point Method
JPL	Jet Propulsion Laboratory
MSL	Mars Science Laboratory
MSR	Mars Sample Return
Matlab	Matrix Laboratory
MoI	Moment of Inertia
NASA	National Aeronautics and Space Administration
PDG	Powered Descent Guidance
PID	Proportional Integral and Derivative
SCVX	Successive Convex Optimization
TRN	Terrain Relative Navigation

Nomenclature

Greek Symbols

Symbol	Description	Units
α	angle of attack	rad
α	mass depletion rate	kg/s
β	angle of sideslip	rad
γ	flight-path angle	rad
γ_{gs}	glide-slope constraint	rad
Γ	slack variable	
δ	latitude	rad
δ_a	normalized asymmetric parafoil deflection	-
δ_{max}	maximum gimbal angle	rad
δ_s	normalized symmetric parafoil deflection	-
η	tilt angle	rad
ρ	density	kg/m^3
θ	pointing angle	rad
σ	bank angle	rad
χ	heading	rad
ω_{cb}	rotational rate of the central body	rad/s
$\boldsymbol{\omega}$	rotation vector	rad/s
τ	longitude	rad
ϕ	cant angle	rad
χ	heading	rad

Roman Symbols

Symbol	Description	Units
$\mathbf{A}$	state coefficient matrix	–
AR	aspect ratio	–
b	wingspan	m
c	chord	m
h	height	m
$\mathbf{B}$	control coefficient matrix	–

$\mathbf{C}$	output transmission matrix	–
$\mathbf{C}$	transformation matrix	–
C_D,C_L,C_S	aerodynamic force coefficient	–
C_l,C_m,C_n	aerodynamic moment coefficient	–
$\mathbf{D}$	direct transmission matrix	–
D	drag	N
$\mathbf{d}$	surface landing target	m
$\mathbf{F}$	force vector	N
$\mathbf{g}$	gravity vector	m/s^2
g_0	Earth surface gravity	m/s^2
g_M	Mars surface gravity	m/s^2
g_T	Titan surface gravity	m/s^2
I_{sp}	specific impulse	s
I_{xx}, I_{yy}, I_{zz}	inertia about the principal axes x, y, z	kg m^2
$\mathbf{I}$	inertia tensor	kg m^2
$\mathbf{I}$	normalized inertia tensor	–
$\mathscr{L}$	roll	Nm
L	lift	N
m	mass	kg
$\mathscr{M}$	pitch	Nm
$\dot{m}$	mass flow	kg/s
$\mathscr{N}$	yaw	Nm
p	roll rate	rad/s
q	pitch rate	rad/s
q	quaternion	–
$\bar{q}$	dynamic pressure	N/m^2
$\mathbf{q}$	quaternion vector	–
r	modulus of position vector	m
r	yaw rate	rad/s
$\mathbf{r}$	position vector	m
$\mathbf{r}_{cm}$	position vector of the CoM	m
R	radius	m
S	side force	N
S_{ref}	reference area	m^2

t	time	s
$\mathbf{T}_B(t)$	commanded thrust	N
T	modulus of thrust vector	N
$\mathbf{T}$	thrust vector	N
$\mathbf{u}$	control vector	–
V	modulus of velocity vector	m/s
$\mathbf{v}$	velocity vector	m/s
$\mathbf{W}$	wind vector	m/s
x, y, z	Cartesian position components	m
$\mathbf{x}$	state vector	–
$\mathbf{y}$	output vector	–

Indices

Symbol	Description	Units
0	initial condition	
A	aerodynamic	
A	airspeed based	
AA	aerodynamic frame, airspeed based	
AG	aerodynamic frame, groundspeed based	
B	body frame	
c	commanded	
cb	central body	
cm	centre of mass	
f	final	
G	gravitational	
G	groundspeed based	
G	guidance frame	
I	planetocentric inertial frame	
M	Mars	
P	propulsion frame	
R	planetocentric rotating frame	
T	Titan	
T	thrust	
TA	trajectory frame, airspeed based	
TG	trajectory frame, groundspeed based	
V	vertical frame	
W	wind frame	
x, y, z	along X-, Y- and Z-axis	

1 INTRODUCTION

This report, for which this chapter serves an introductory purpose, has been established to outline the motives and results regarding the thesis research conducted on the matter of autonomously landing a spacecraft on the surface of a planetary body with a degree of precision in the order of meters. The research focuses on Mars and Titan (moon of Saturn), however, the discussion may potentially be extended to landing on other bodies (i.e., planets, moons, planetoids, etc.) in the Solar System and beyond. Guidance, Navigation and Control (GNC) systems combined with autonomous trajectory optimization techniques are in particular the areas of interest. Included in the background information delineated in Section 1.1 are an overview of past missions that have successfully landed on the Martian and Saturn's moon surface, the scientific objectives for enabling pinpoint precision landing and the challenges that come along with it. The first section is concluded with a brief overview of the consequential upcoming Mars 2020 mission and the proposed Titan parafoil mission, marking the latest developments in the robotic Solar System exploration program of NASA. Based on the presented background information a problem statement is derived. This statement is presented in Section 1.2 along with the entrenched research question that naturally follows from it. In the final section an overview of the rest of this thesis is outlined.

1.1. BACKGROUND INFORMATION

Autonomously landing a spacecraft on the surface of a celestial body with a high degree of precision (in the order of meters) is very challenging. To gain a better understanding of the current advances, a brief historical overview of space vehicles that have successfully landed on the Martian and Titanian surface is given in Section 1.1.1. Over the course of time, the landing ellipse, defined as the region with a 99% likelihood of where a space vehicle will land, has improved steadily but is currently still measured in the order of kilometers (Blackmore, 2017). The question that may rightfully arise is why increasing the precision is needed if these past missions have already proven to be successful. This matter is addressed from a scientific perspective in Section 1.1.2. The challenges that come along with pinpoint landing on a planet, and Mars in particular, form a key aspect of this conducted research and are highlighted in Section 1.1.3. In the last parts of this background information overview the discussion is tilted to the Mars 2020 and the Titan parafoil mission, the latest developments in NASA's exploration journey to the Red Planet and Saturn's moon, respectively.

1.1.1. HISTORICAL OVERVIEW

Over the last decades multiple flyby and orbital missions launched towards Mars with the intent of gaining a better understanding of the neighbor planet. Several spacecraft have even successfully landed on the Martian surface within a predefined landing ellipse. The five most significant of these missions are Viking, Pathfinder, Opportunity & Spirit, Phoenix and Curiosity (MSL) (Braun and Manning, 2007). In case of Titan, the only, yet very significant mission, has been Cassini-Huygens. As interplanetary navigation improved over time, the landing ellipses tightened steadily. In this section some characteristics of Viking, MSL and Cassini-Huygens are highlighted, as they are part of NASA's Large Strategic Science Missions (National Research Council, 2006). Brief insights are provided into the key Entry, Descent and Landing (EDL) elements of these costliest and scientifically most capable spacecraft missions.

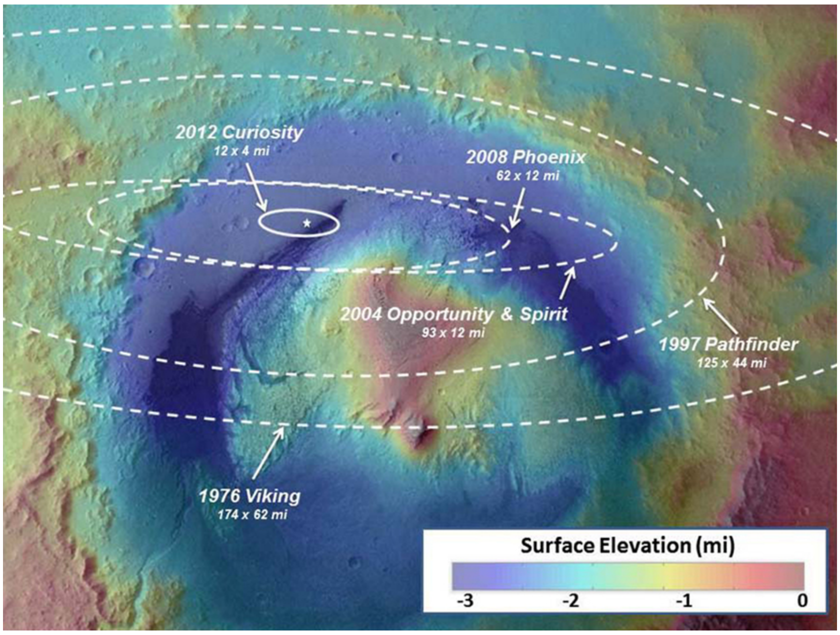


Figure 1.1: Landing ellipses of successful landings on Mars. Image credit: NASA/JPL-Caltech/ESA

VIKING

Viking has been the first spacecraft to safely touch down on Mars and has been important in measuring the planet's atmospheric properties firstly (Soffen and Snyder, 1976). These properties are driving factors in entry vehicle design as they indicate how the enormous kinetic energy is dissipated by friction due heating. Due to the high number of uncertainties, a statistical approach was used to design the GNC system (Holmberg et al., 1980). During terminal descent the spacecraft used gravity-turn guidance (dual altitude-velocity contour) with limitations in terms of CPU and memory (Ingoldby, 1978). This system was verified using a relatively low accuracy of five-hundred Monte Carlo simulations, and consequently a rather large ellipse emerged ($\approx 280 \times 100$ km). Nevertheless, Viking became crucial in laying the foundation and till this day serves as the backbone of planetary (mainly parachute) landing technologies (Braun and Manning, 2007).

MARS SCIENCE LABORATORY

The Mars Science Laboratory (MSL) and its focal point, the Curiosity rover, has been the most advanced Martian mission yet. The descent maneuver was dubbed "seven minutes of terror", adverting to the time available to slow down the spacecraft travelling at about 21,000 km/h at the start of the EDL phase to zero at the end of it. The sequence being fully autonomous, the timing being perfect and a newly adapted powered descent technology to guarantee safe touchdown of heavier payloads: the sky crane system. The MSL became the first interplanetary space mission to use precision landing technologies during EDL (Way et al., 2013). The combination of both advanced guidance and navigation made the landing ellipse constraint shrink significantly ($\approx 20 \times 6.4$ km) and now serves as a pathway to pinpoint precision landing.

CASSINI-HUYGENS

Cassini-Huygens, a joint venture between the Jet Propulsion Laboratory, the European Space Agency and the Italian Space Agency, aimed for getting to know the Saturnian system, i.e., Saturn, its rings and its numerous moons. The robotic spacecraft, one the most ambitions missions ever launched into deep space, consisted of the orbit Cassini designed by the Americans and the lander Huygens designed by the Europeans. Powerful on-board cameras and instruments gave the spacecraft the ability to take detailed images of light spectra and high accuracy measurements of atmospheric properties (Atkinson et al., 2007). This was conceivable as the instruments were capable of measuring atomic particles, densities, electrical charges, magnetic fields and mass. After reaching the Saturnian system in July 2004, Cassini started beaming valuable data, providing scientist a much deeper understanding of the Saturnian area. Huygens was developed to re-enter into the largest moon of Saturn: Titan. The landing sequence consisted of a parachute descent phase and after a successful touchdown it became to most distant lander in space travel history.

1.1.2. Need for Precision Landing

Conclusive from the provided historical overview of the five most successful missions to the Red Planet is the fact that currently the accuracy for landing on Mars is still measured in the order of kilometers. The necessity of increasing the precision to the order of meters is addressed in this section from a scientific perspective. This discussion about the need for precision includes a portray of the benefits in terms of scientific objectives, the way in which human exploration of the Earth's neighbor planet is aided and the contributions to the sustainable exploration of space. Often space missions aid science in the quest for knowledge in a specific field and hence the respective scientists precept most of the requirements. However, at times the engineering requested for certain space missions is not possible (yet) due to the level of complexity being too high.

Scientific Objectives

A set of (uniquely) determined engineering constraints related to a mission's architecture governs the selection of landing sites. Scientists have to abide this framework of acceptable landing sites settled by these constraints, and from that select the most relevant location. These constraints are often based on engineering capabilities of the respective vehicle. A space mission is designed in such a way that the delivery vehicle is capable of safely landing the payload. These proposed landing sites are currently inherently safe because of the fact that the landing ellipses have dimensions measured in kilometers (Thomson and el Baz, 2014).

Generally speaking, the larger the proposed landing ellipse, the more challenging it becomes to select a suitable landing region. Examples of such areas are deserts, oceans, or lowlands with little elevation difference. A drawback of these sites is the fact that they are not the most interesting in terms of science. To obtain higher quality scientific output, spacecraft would have to be able to land in an inherently unsafe environment with highly varying elevation, i.e., pinpoint precision landing becomes an absolute necessity. A whole new world would open up if fields near caves and valleys can be explored. The ability to settle with permanent outposts on bodies throughout the Solar System becomes within reach and planetary samples can return to Earth for further investigation, the latter for which the Mars 2020 is preparing. The establishment of such ideals is only possible if current systems are further advanced to push through the current boundaries.

Human Exploration

It is the fervent wish of mankind to break through the boundaries of science and establish whether life beyond Earth exits. The next step in the exploration is a human exhibition send to Mars, chosen because it provides the quickest way to find an answer within the current engineering capabilities (Zubrin, 2014). The Martian surface is extremely variable and a rendezvous with prior positioned assets is needed. Landing within the proximity of these assets has two important reasons. These assets contain elements vital for supporting human life such as power systems, oxygen generators and water supplies. Furthermore, these assets may be near special human exploration zones and regions of resources used for exiting scientific in-situ observations.

Even though a manned expedition to Mars might be decades away, several test flights must be conducted to gain confidence in the developed systems. A successful human exploration mission to the Red Planet thus comes with the quest of finding a way to land 40-80 tonnes on the Martian surface (with a possible elevation) within the proximity of prior positioned assets (± 10 m). However, as reality strikes, today's engineers are battling to land a rover of 1 ton within an accuracy of 10 km. To have realistic plans for landing humans on Mars, the mass landing capability needs a magnitude increase of order two, while the landing accuracy needs a magnitude increase of order four. As systems need to get more advanced to land more weight on Mars with higher precision, solely relying on heritage systems is not a possibility.

Sustainability Prospect

From a perspective of sustainability, precision landing demonstrated on Earth has shown that rockets, after payload delivery to space, can land back, refuel and be reused (instead of the conventional ocean dumping) (Blackmore, 2017). Phenomenal engineering artwork has been applied several times by SpaceX to be able to land their rockets on predefined locations. One might rightfully wonder why precision landing has been accomplished on Earth but seems so complex on Mars and Titan. The short answer would be that the latter two are GPS denied environments, making planetary landing navigation extremely hard. However, the problem has deeper roots that need to be addressed in the quest of interplanetary pinpoint landing. The various reasons behind this increased complexity are discussed in the next section. Nonetheless, the precision landing by SpaceX does serve as a demonstration of key technologies. The ability of the GNC system to tackle dispersions of random nature is truly remarkable. Abreast the increase in precision, the technology serves a greater purpose for the aerospace industry, namely, allowing for a more sustainable practice of spaceflight. This is especially important in these times where the industry inserts an enormous pressure on the globe.

1.1.3. PRECISION LANDING CHALLENGES

Autonomously landing spacecraft on a planetary body with high prescribed precision is very challenging. NASA even refers to the EDL phase of a Mars mission as the seven minutes of terror (Braun and Manning, 2007). Landing precisely on Titan is rather difficult due to wind gusts on the Saturnian moon. However, there are a lot of (potential) benefits to be derived from planetary pinpoint landing. The report of the U.S. National Research Council on National Space Technology Roadmap and Priorities even identifies precision landing as a top priority and critical EDL capability for the near future. This section elaborates on the challenges of precision landing and the respective research focus aspects. It is important to keep in mind that the challenges are interrelated to an increase in maximum payload mass, an increase in spacecraft reliability and a decrease of development costs. Based on the dual combination of the NASA roadmap for EDL technologies (Munk et al., 2015) and other literature sources, the main challenges have been identified and grouped:

Atmospheric environment:

- The atmosphere of Mars is too thin to cause decent deceleration, but thick enough for tremendous heat (thermo-mechanical loads) induced by friction at hypersonic speeds (Lissauer and de Pater, 2013).
- Lack of understanding of the aerodynamics, aeroheating environment, winds and density variations (Sostaric, 2010). Especially for Titan due to the fact that Cassini-Huygens is the only reference mission.
- The atmosphere of Titan is very thick, in fact, Titan has a surface density value higher than four times the value of Earth. On the other hand, however, the gravity value is much lower (Quadrelli et al., 2019). This unique combination causes significantly different aerodynamic performances.

Guidance, navigation and control:

- Full autonomy necessary because of time delay between Earth and Mars/Titan. The Deep Space Network (DSN) cannot be utilized for these landing sequences.
- Increased navigation errors because Mars/Titan are GPS denied environments.
- Strain on GNC systems, because of the abrupt reduction of velocity and release of kinetic energy (Mars).
- Strain on the GNC systems due to heavy gusts of wind (Titan).
- No margin for error in the sense that the first attempt must be immediately right.
- End-to-end validation of GNC systems in Martian or Titanian conditions is not possible.

Physical constraints:

- Limited on-board fuel capacity and thus limited hovering capabilities (Mars).
- Limited on-board power and thus limited canopy pull capabilities (Titan).
- Limited aerodynamic control due to the inherent design of the spacecraft (Mars).
- Lower ballistic coefficients allow the spacecraft to decelerate to higher altitudes. While a low ballistic coefficient is important for high mass vehicles in the thin Martian atmosphere, in terms of design it quite difficult to achieve lower coefficients (Putnam and Braun, 2016).

Surface hazards:

- Four main surface characteristics are identified as potential landing hazards: slopes, roughness, craters and illumination. Spacecraft should therefore be able to dodge to inherently safe locations.

While it might seem difficult to tackle these problems and challenges, it is important to keep in mind that overcoming great challenges produces great benefits. This philosophy also holds for the problem at hand, for which the following possible benefits are derived:

- Increased mass of delivered payload
- Access to higher elevation surfaces
- Increased landing accuracy
- Increased robustness of EDL systems
- Increased safety and probability of interplanetary mission success
- Increased scientific output
- Reduction in space mission costs
- Increased safety for human spaceflight missions
- Improved reliability of return samples

1.1.4. MARS 2020

The Mars 2020 mission is one of the latest developments in the exploration journey of NASA to Mars. This mission is a so-called flagship-class mission and is part of the Large Strategic Science Missions. This means that the spacecraft is classified as one of the costliest and most capable ever built. This section shortly addresses the key mission characteristics from a scientific perspective. One of the key scientific objective is searching for past evidence of life in regions where in ancient times conditions may have been favorable for microbiological life to have existed. Furthermore, Mars 2020 will prepare for (near) future human space exploration. Because of the success of the Mars Science Laboratory, the same design has been adopted with slight modifications (Fosse et al., 2015).

The timed launch date is currently settled for July/August 2020 to ensure a favorable relative position between Earth and Mars. The mission duration is established at a minimum of one year. Because the mission is mainly MSL based, additional mission costs and risks are suppressed. Even the design of the rover is inherited from the Curiosity rover. Of the total of twenty-three cameras mounted on the spacecraft, seven are purely focusing on the EDL sequence. This gives an indication of the importance NASA gives to the analysis of the EDL phase. Several workshops have taken place to determine the final landing location of the spacecraft. From the 54 initially proposed landing sites, three are currently considered:

- Columbia Hills, Gusev Crater (14.5°S, 175.4°E) → near Spirit
- Jezero Crater (18.855 °N, 77.519°E)
- NE Syrtis (18°N, 77°E)

Important for the final choice of the landing location is the fact that all prior determined scientific objectives can still be met. NASA identified a total of four main scientific goals:

1. Determine if life once existed on Mars.

2. Characterization of Martian climate.

3. Characterization of Martian geology.

4. Preparation for human exploration.

1.1.5. TITAN PRECISION LANDING PARAFOIL

To recapitulate, part of the research focus concerns a parafoil-based precision landing on Titan. The designed algorithm has been verified by simulating Mars landing scenarios and comparing the results with the extensive available literature. This section provides insight as to why Titan is of such interest. Mars and Titan are of course different in essence, in the sense that Mars has a rather thin atmosphere for which conveniently rocket thrusters are utilized in the landing sequence. Yet, the nature of the precision landing guidance problem for both celestial bodies is rather similar. The adopted method, as will be broadly delineated in Chapter 4, is convex optimization guidance. This methodology has been well validated and researched for Mars landing.

Questions remain why Titan is of such interest for NASA in particular and the science community in general. Its geophysical features are such that Titan's liquids and gases are quite similar compared to the ones found on Earth. Furthermore, there is evidence of possible underground oceans. The image from the Cassini Radar Mapper depicted in Figure 1.2 shows some of the northern lakes and seas of the Saturian moon. Interesting is the fact that this image has been constructed based on multiple operational camera resolution modes: 0.3 —1.5 km, 2 —10 km, and 40 —200 km. A technique described as false coloring has been applied to make a clear distinction between the features of land (brown/yellow) and hydrocarbon liquids (blue/black). The proposed landing target area is currently set to be near the Maracaibo Lacus site.

As before-mentioned, the atmosphere of Titan is rather thick, reason being for adapting parafoil landing techniques. A direct consequence of this, is that the EDL sequence is relatively slow (order of hours) and thus wind directions and magnitudes can change considerably during descent. An autonomous Precision Aerial Delivery Systems (PADS) has been chosen to be the payload delivery system. These types of controlled parachutes have several advantages over the traditional parachute. First and foremost, classical canopies have (almost) no control authority. Systems like PADS however have actuators that allow for remote in-flight control capabilities. Nonetheless, these systems have not yet been deployed in space exploration and thus extensive research is required. Earlier research carried out at JPL focused on the EDL sequence till terminal descent. This research is a continuation hereof, with a main focus on the optimization of the terminal descent. During the landing process control is obtained by deflections: symmetric deflections control longitudinal dynamics while asymmetric deflections control lateral dynamics. The complication of wind profile

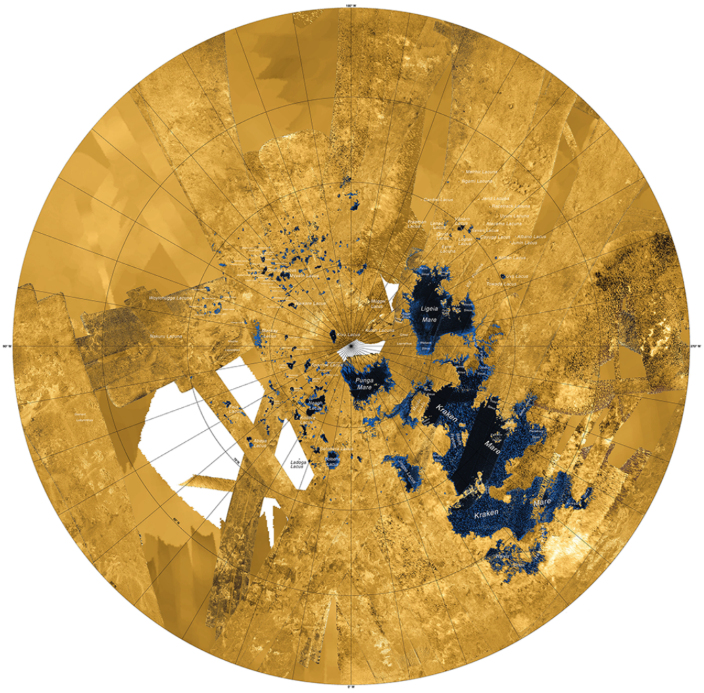


Figure 1.2: Cassini Radar Mapper: North polar lakes and seas on Titan (Hartwig et al., 2016)

magnitudes at high altitudes still holds for terminal descent and thus its incorporation has been an integral part of the analysis. The final landing is generally performed upwind to minimize the touchdown velocity of the payload and reduce roll-over risk. All the modelled wind profiles are based on data obtained from the Cassini-Huygens mission. It is important to mention however that this data is limited and many aspects remain to be unknown. The guidance algorithm should therefore be robust to deal with dispersions.

1.2. Problem Statement

Several key challenges have been presented that come along with precision landing on Titan and Mars. This research will only focus on a selection of these challenges. Former interplanetary guidance algorithms have not fully utilized the spacecraft's divergence capabilities, which resulted in a limited landing accuracy. Accordingly, the main research purpose is to develop a large divergence algorithm for terminal descent that is able to guide the spacecraft to a defined landing location, thereby following optimal paths to achieve a landing precision in the order of meters. The main problem regarding the guidance and navigation systems for precision landing is the fact that limited robustness, uncertainties of navigation sensors, variations in atmospheric properties, wind variations and modelling inaccuracies cause landing accuracy to be rather limited. As will become clear in the chapters ahead, current GNC algorithms do not fully utilize the divergence capability of a space vehicle, which also results in a limited landing accuracy. The research therefore focused on the following aspects:

1. Improved guidance algorithms to increase robustness.
2. Incorporate optimal paths in the design to allow for large divert guidance.
3. Enforce constraints to ensure soft-landing.
4. Insurance of spacecraft autonomy in compliance with real-time EDL flight.

Based on the discussed challenges, the defined problem statement and the established focus aspects of this research, the entrenched research question is stated:

What order of magnitude precision for landing on planetary bodies can be achieved by using advanced guidance, navigation & control systems in combination with state-of-the-art autonomous trajectory optimization algorithms?

Differentiating this question into partial questions breaks down the complexity of the problem and gives rise to a clear research overview. For easy tracking each question has been given a unique **RQ-x** identification:

- **RQ-1:** Is a convex optimized real-time generated guidance algorithm capable of precisely guiding the spacecraft to predetermined landing ground?
- **RQ-2:** What cost function, state constraints and control constraints are needed based on the mission scenarios?
- **RQ-3:** Is a convex optimized real-time generated guidance algorithm capable of retaining to the state and control constraints throughout the entire sequence?
- **RQ-4:** How does the incorporation of a three-degree-of- freedom fuel/power optimal trajectory differ from a six-degree-of-freedom trajectory?
- **RQ-5:** Is the algorithm capable of generating optimal divergence paths in real-time in case the nominal trajectory turns out to be non-feasible?
- **RQ-6:** Are the landing accuracy and the fuel/power consumption mutually related?
- **RQ-7:** How does the algorithm performance differ between a thin atmosphere (Mars) and thick atmosphere (Titan) landing sequence?
- **RQ-8:** Is a landing accuracy on the order of meters theoretically feasible?
- **RQ-9:** What affect do uncertainties in the state variables have on the algorithm, i.e., how robust is the algorithm?

1.3. REPORT OVERVIEW

In the final segment of this chapter the structure of the remainder of research report is established. Chapter 2 defines the current state-of-the-art systems based on a comprehensive literature review. The third chapter focuses on the relevant flight dynamics, providing a total overview of relevant reference frames, state variables, frame transformations and general equations of motion. This includes a delineation of parafoil dynamics for landing on Titan. Chapter 4 forms the core chapter of this study and focuses on relevant Guidance, Navigation and Control (GNC) elements. This chapter starts with a general discussion on strategies for designing GNC systems. The guidance system, forming the core focus of the research, is charged with the task to bring forth steering commands by taking an estimated reference of the prevailing and final state into account. The input state data is provided by the navigation system using the interface information gathered by yet again (theoretical) models and sensors. Throughout the research, however, perfect state knowledge has been assumed due to international regulations on the export of defense technologies. Despite the fact that this research does not specifically focus on the control aspect of the GNC system, it remains an important theme for closed-loop analysis and theoretical utilization of the propulsion system and parafoil pull for guidance purposes. The main aspects are hence highlighted in the final section of the fourth chapter. In Chapter 5 the formal proofs with regard to the convex optimization and guidance method are conferred. This chapter also outlines the difference between applying the golden search method and the successive convexification method for guidance time optimization. The sixth chapter provides a framework for the design of the simulator, which is coupled to the verification of the algorithm. Chapter 7 confers the results of both the nominal Mars and Titan simulations, as well as the assessment of algorithm robustness and sensitivity. In the final chapter conclusive commentaries and recommendations are charted.

2

Heritage

This chapter outlines the heritage systems of landing on planetary bodies, with a particular focus on guided precision landing. The most important systems with regard to pinpoint landing are the EDL systems. Technologies related to these systems may be classified into four main categories: aeroassist and atmospheric entry, descent and targeting, landing and, finally, technologies related to the vehicle systems. All these elements are outlined in Section 2.1 to gain an understanding of the (working mechanism of the) current state of the art and the challenges ahead in terms of autonomous guidance for precision landing. Section 2.2 starts with a of a more broad discussion of the fundamentals of landing dispersions. The discussion ends with several future considerations that increase the precision of landing but are not (directly) related to either the GNC system or the autonomy of the spacecraft. These topics are outlined for enhancing future research.

Being the first Martian and interplanetary mission to have ever performed a guided entry, the Mars Science Laboratory (MSL) mission forms an integral part of this research. Building on this legacy, the Mars 2020 mission will have a near identical design and additionally be more robust. Accordingly, these two missions have been taken as baseline references (i.e., mission and vehicle) for the Mars landing simulation sequences, as has been outlined in Section 2.3. In the same section the heritage research of the Titan precision landing parafoil is outlined, namely, the Precision Aerial Delivery System (PADS). The mission and system requirements for this research are derived and outlined in Section 2.4, marking the conclusion of this chapter.

2.1. Entry, Descent, and Landing Technologies

The EDL system is essential in safely bringing a vehicle from planetary approach conditions to the surface of a Solar System body (or in atmospheric transit phase). For better understanding and insight purposes, this section serves a road map for current heritage technology capabilities and anticipates on the needed development based on NASA Technology Roadmaps. The discussion with regard to this matter is initiated in Section 2.1.1. The breakdown structure shown in Fig. 2.1 depicts the EDL fundamentals and is used for governing the discussion. The breakdown is divided into three levels with the entire segment of the EDL systems on the first (highest) level. The present top priority, highest level goal, of the EDL systems advancement is the enabling of landing heavier payloads travelling at faster velocities safely with high precision. The central elements, further discussed in Sections 2.1.2 -2.1.3, include aeroassist and entry, descent and targeting, landing and vehicle system technologies, respectively.

2.1.1. Heritage and Advancement

Advances in space technology are most of the time driven by mission requirements and are often only adapted if the systems have a high Technology Readiness Level (TRL). But even if only heritage systems are used, the performance is not known to a very accurate extend. This can be attributed to the fact that Titan-like and Mars-like EDL flight condition cannot be completely replicated (yet) or bring forth far beyond reasonable expenses. On top of that, fact of the matter is that the various EDL elements are highly interdependent and altogether determine the behaviour of the system (Adler et al., 2010). As it is not possible to test the EDL end-to-end sequence before launch, qualification strongly depends on computer simulations. It is therefore crucial to collect as much as possible existing EDL flight data and analyze the performance characteristics

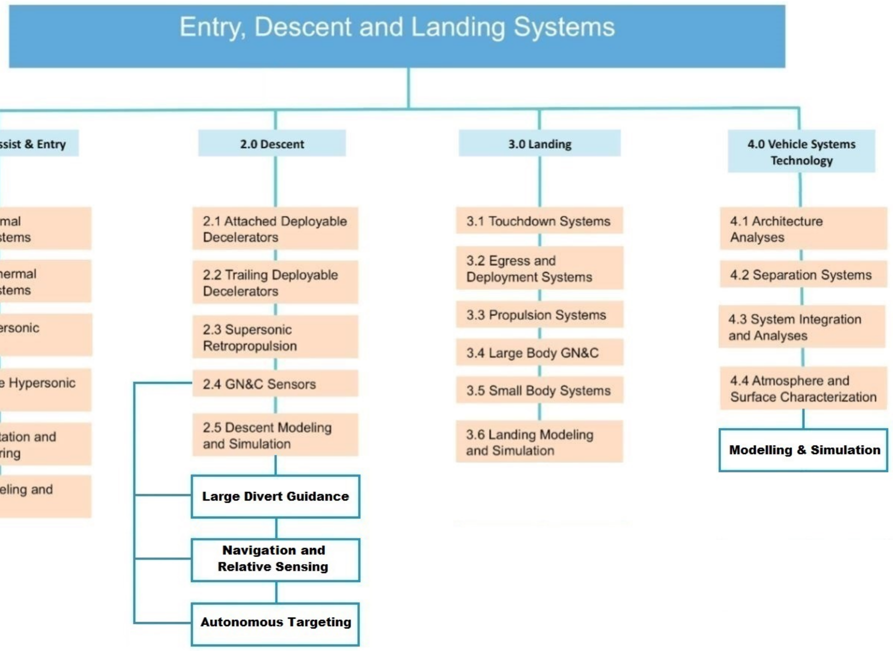

Entry, Descent and Landing Systems
1.0 Aeroassist & Entry
2.0 Descent
3.0 Landing
4.0 Vehicle Systems Technology
1.1 Rigid Thermal Protection Systems
1.2 Flexible Thermal Protection Systems
1.3 Rigid Hypersonic Decelerators
1.4 Deployable Hypersonic Decelerators
1.5 Instrumentation and Health Monitoring
1.6 Entry Modeling and Simulation
2.1 Attached Deployable Decelerators
2.2 Trailing Deployable Decelerators
2.3 Supersonic Retropropulsion
2.4 GN&C Sensors
2.5 Descent Modeling and Simulation
Large Divert Guidance
Navigation and Relative Sensing
Autonomous Targeting
3.1 Touchdown Systems
3.2 Egress and Deployment Systems
3.3 Propulsion Systems
3.4 Large Body GN&C
3.5 Small Body Systems
3.6 Landing Modeling and Simulation
4.1 Architecture Analyses
4.2 Separation Systems
4.3 System Integration and Analyses
4.4 Atmosphere and Surface Characterization
Modelling & Simulation

Figure 2.1: Entry, Descent, and Landing Systems Technology Breakdown Structure (Adler et al., 2010). Blue boxes framework: additional technologies needed for autonomous guided precision landing.

and limits of the systems. That by itself might impose a problem as flight data is at times too scarce for accurate predictions. Conducting sensitivity analyses often aid in better understanding of system behavior.

2.1.2. AEROASSIST AND ATMOSPHERIC ENTRY

The EDL sequence starts with the Aeroassist and Atmospheric Entry (AAE) phase in which the spacecraft is decelerated from hyperbolic arrival orbit as a result of drag in the aerobraking maneuver. Thermal protections systems safeguard the carrying payload from the extreme hypersonic entry environment and decelerators ensure the rapid reduction of entrance velocity. Within the AAE framework, the approach navigation plays an important role in landing ellipse size reduction. Furthermore, upon entry, robust guidance software is crucial for steering the spacecraft to the right destination. At last, navigation processes might get interrupted during hypersonic entry causing errors in process known as blackout phase.

APPROACH NAVIGATION

In terms of precision landing, the approach navigation plays an important role. For unguided ballistic entry, increasing the landing accuracy is mainly attributed to an improvement in the approach navigation (Rapp, 2016). Even though this research focuses on guided powered descent, the approach navigation is very useful in the fact that it can improve initial entry conditions such that the landing ellipse is significantly reduced in size (Putnam and Braun, 2016). The current state of the art is able to produce an accuracy ellipse of 30 km × 3 km, without the consideration of planetary body induced uncertainties (such as strong winds).

ENTRY GUIDANCE

A classical way of classifying guidance methods is distinguishing between predicting capabilities and nominal trajectories (Wingrove, 1963). The former predicts future trajectories with a fast (inner) loop and does not make use of stored trajectories. Taking initial, reference and final state into account, the main task of the guidance system is to generate steering commands for the space vehicle. The necessity of obtaining inputs from the surroundings is provided for by the navigation system and its sensors. The design of such guidance systems must be such that dispersions in the state of the spacecraft and conditions of the atmosphere are dealt with accordingly. An example of an accurate and adaptive guidance algorithm for space vehicles, particularly those with a low lift-to-drag ratio, is the numerical predictor-corrector method (Lu, 2008). This method proved to be successful in the hypersonic guidance of the MSL and hence established a solid base for future

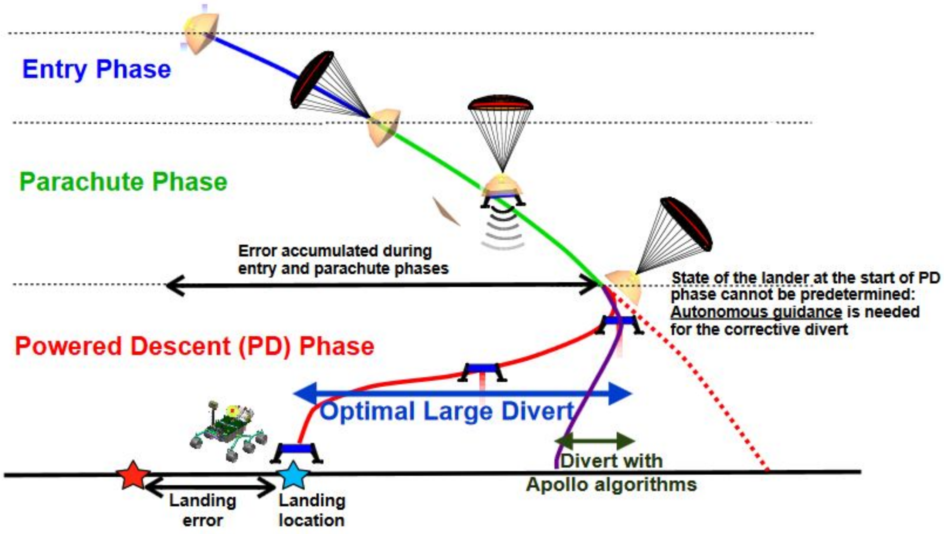

Entry Phase
Parachute Phase
Error accumulated during entry and parachute phases
State of the lander at the start of PD phase cannot be predetermined: Autonomous guidance is needed for the corrective divert
Powered Descent (PD) Phase
Optimal Large Divert
Divert with Apollo algorithms
Landing error
Landing location

Figure 2.2: Optimal large divert algorithms for enabling planetary pinpoint precision landing: G-FOLD (Açıkmeşe and Ploen, 2007).

missions. However, as rightfully stated by Wang et al. (2017), guidance systems need more advancement to become increasingly autonomous, robust and reliable .

HYPERSONIC RADIO BLACKOUT PHASE

The hypersonic phase is highly non-linear in nature and the extreme conditions cause a so-called radio blackout phase, making in-situ measurements difficult. A coat of ionized particles generated by the shock travelling in front of the flow causes an interruption in the Inertial Measurement Unit (IMU) navigation process (Morabito, 2002). This inflicts navigational errors on the spacecraft, causing a reduction in the accuracy of the landing. A proposed solution to this phenomenon is a tightly coupled integrated re-entry navigation system using both IMU and an initially proposed Ground Positioning System (GPS) (Mooij and Chu, 2002). However, one should keep in mind that as of yet both Mars and Titan are GPS-denied environments.

2.1.3. DESCENT AND TARGETING

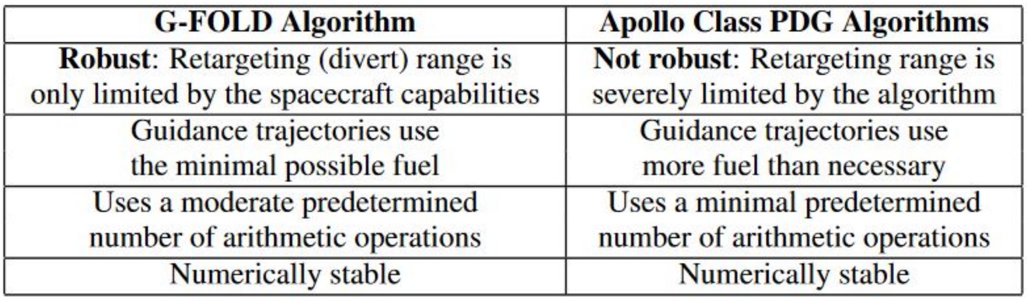
During the entry phase, large errors in both position and velocity accumulate as a consequence of atmospheric ambiguities and the narrow control authorization (Açıkmeşe et al., 2012). The lander corrects for those errors real-time during the descent phase. Targeting ensues during the last portion of the descent in which terrain-relative landing decisions are made. The MSL radar-based terrain sensing system embodies the state of the art of terminal descent technology. Included in this are propulsion systems, relative sensing and autonomous targeting for touchdown preparation. The three latter, being part of the GNC system, are not fully developed yet and serve as a pathway for enabling future precision landing (Munk et al., 2015).

LARGE DIVERT GUIDANCE

The current state-of-the-art powered descent guidance algorithms used in space mission include gravity turn guidance and Apollo era based polynomial guidance. Even the guided entry implemented in MSL was built on algorithms extracted from the Apollo era, allowing minimum diversions only (Wong et al., 2006). These algorithms do not fully utilize the divergence capability of a space vehicle, resulting in a limited landing accuracy. While empirical relations show that these algorithms are efficient in terms of fuel consumption, the minimization of propellant use is not explicitly taken into account. Besides, this optimal fuel usage is only true if the covered lateral divert distances are an approximate of twenty percent of the altitude at which divergence began. For that reason these heritage systems cannot classified among large divert guidance technology. Fig. 2.2 depicts the disparity between Apollo divert algorithms and more advanced optimal large divert systems.

The next main leap in Martian exploration thus lies in implementation of these systems. While the moon of Saturn is different in nature, this is also relevant for landing on Titan. The objective large divert guidance systems have is finding optimal divert paths robustly while falling towards the surface. A useful mathematical technique for such optimal computations is convex optimization guidance. While the objective of minimizing landing error using fuel optimally is inherently nonconvex, it has been proven that an optimum output of

Table 2.1: G-FOLD *vs* Apollo class algorithms (Açıkmeşe et al., 2012)



G-FOLD Algorithm	Apollo Class PDG Algorithms
Robust: Retargeting (divert) range is only limited by the spacecraft capabilities	**Not robust**: Retargeting range is severely limited by the algorithm
Guidance trajectories use the minimal possible fuel	Guidance trajectories use more fuel than necessary
Uses a moderate predetermined number of arithmetic operations	Uses a minimal predetermined number of arithmetic operations
Numerically stable	Numerically stable

the relaxation of the problem also holds for the initial nonconvex problem (Açıkmeşe and Ploen, 2007). The Jet Propulsion Laboratory developed an algorithm, known as Guidance for Fuel Optimal Large Diverts (G-FOLD), for this so-called lossless convexification of nonconvex optimization. The fundamental differences between the more robust G-FOLD and Apollo heritage algorithms are summarized in Table 2.1. To conclude: the main objective of large divert guidance systems is to provide guidance algorithms that on-board are able to solve divergence maneuvers while respecting the limitations of the spacecraft in terms of fuel or power capacity. These form the core topics of the research and are extensively discussed in Chapter 4.

Navigation and Terrain-relative Sensing

Knowing the position and velocity state of the spacecraft in real time during the descent and targeting phase is a critical component of the closed GNC loop, and even more so in case of pinpoint landing. Conventional techniques include determining vertical velocity and altitude using radar and horizontal velocity using Doppler radar (Braun and Manning, 2007). To allow safe precision landing near scientific interesting areas, high-frequency measurements are required while minimizing size, mass and power consumption of the navigation sensors. The importance of this lays in the fact that an improved navigation (sensor) performance increases the reliability, autonomy and accuracy of the GNC scheme (Jiang et al., 2014). The primary challenge in adapting new sensor technology is the detailed physics models needed for full characterization of behavior and spacecraft interaction.

Autonomous Targeting

The ability of the vehicle to make on-board decisions to dodge to a desired target point based on sensor input data is defined as autonomous targeting. Traditional landing approaches are solely based on inertial sensing and had no such systems (Johnson and Montgomery, 2008). The overall objective of autonomous targeting technologies is to enable precise landing at hazardous locations by acquiring knowledge of the area surrounding the target. Common hazards include steep terrains, rough surfaces, craters and areas of no illumination. The latter is mainly ascribed to missions requiring solar power for inducing energy. Furthermore, sites of high elevation are at present defined as hazardous zones. Decreasing inertial errors may be achieved by using a Terrain Absolute Navigation (TAN) system, which enables localization though on-board maps generated by priori sensing data (e.g. satellites). The most common way of bringing this about is through a catalogue based on Digital Elevation Models (DEM). A camera and a crater detection algorithm are used to retrieve information about the position of the vehicle with respect to the inertial frame. The navigation filter uses this data to update the preceding state retrieved from the IMU. As altitude decreases, TAN becomes less and less efficient in improving the errors induced on the IMU measurements. The switch is then made to computing spacecraft position and orientation based on measurements intervals, known as Terrain Relative Navigation (TRN). This system requires two consecutive images of the planetary surface which can also be Laser Imaging Detection And Ranging (LIDAR) or radar images. Analogously, the TRN system requires priori reference data in the form of a DEM. The purpose of the TAN and TRN systems is to aid the inertial navigation system by providing estimates of global position, local position and velocities. A prove of concept of the TRN system has been demonstrated on Earth, and this technology shall also be used on the Mars 2020 mission. As shown in Fig. 2.3, adding the TRN system to the navigation scheme enables larger diverts and backshell separation at higher altitudes (Brugarolas et al., 2016). As suggested by Blackmore (2017), a combination of Earth proven precision guidance and TRN can enable precision landing on Mars, and likewise on Titan.

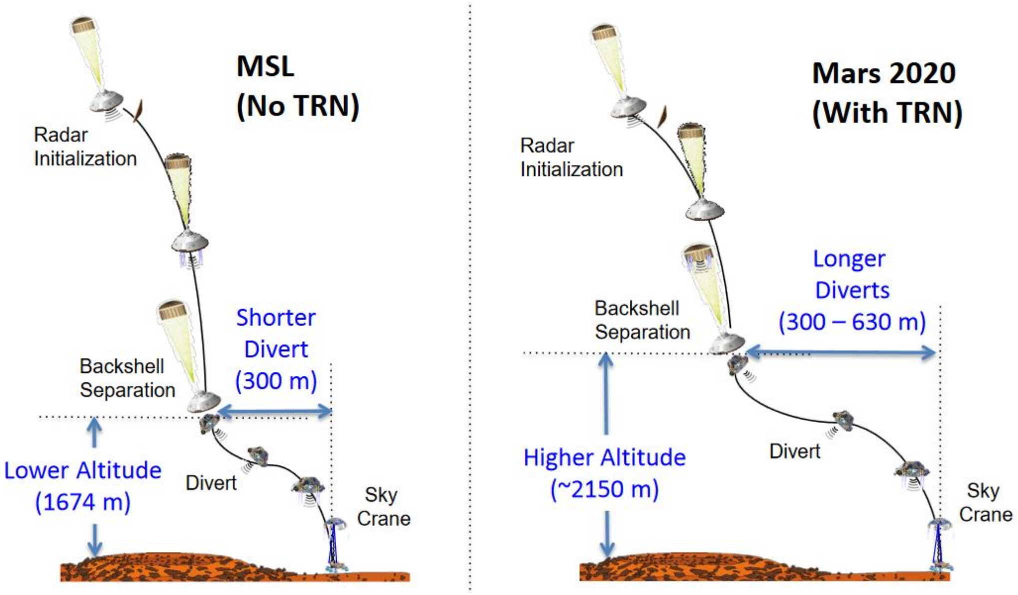


Figure 2.3: Terrain Relative Navigation system enables larger diverts. Image source: NASA/JPL.

2.2. PRECISION LANDING

As was outlined in the background information section in Chapter 1, various successful landings have occurred on Titan and Mars over the last decades. Until MSL, all of these missions mainly used Viking heritage elements and had a ballistic entry with no guidance. This approach has proven to be successful but not sufficient for precision landing. The landing ellipse, the conventional way of presenting planetary landing accuracy, is described in an inertial system. Enabling pinpoint landing does not require re-invention of the wheel but rather some key elements need to be revolutionized. Central in achieving this is the ability to react to trajectory variations caused by uncertainties in the environments, defined as dispersions. Various error sources that contribute to these dispersions, shown in Fig. 2.4, are discussed briefly in Section 2.2.1.

2.2.1. LANDING DISPERSIONS

The inclination of the approach trajectory, entry sense (prograde, retrograde or polar) and location of the landing site are geometric characteristics influencing the landing ellipse dimensions. Other factors that have influence hereupon, illustrated in Fig. 2.4 in terms of relative size based on a 99% likelihood and a 3σ two-dimensional normal distribution, have been been identified by Knocke et al. (2004):

- robustness of real-time guidance algorithms
- navigation uncertainties
- variations in atmospheric properties
- aerodynamic modelling inaccuracies

By tackling the first problem, a spacecraft will be able to react better to the latter ones and hence reduce the landing ellipse size. This research focuses on the sole effect of improved guidance for planetary landing through autonomous trajectory optimization algorithms, of which the discussion is continued in Chapter 4.

2.2.2. FUTURE CONSIDERATIONS

This section highlights the EDL systems and technologies that have significant effect on the landing accuracy but are beyond the scope of this study. To aid future research, a compact list of proposed advances is composed. For a detailed description the reader is referred to Munk et al. (2015). Notice that only some of the future considerations are highlighted and thus the list is not complete.

DECELERATION SYSTEMS

Due to the inherently thin nature of the Martian atmosphere, parachutes have to be rather large and are currently reaching their limit in the amount of mass that can be delivered (Mastropietro et al., 2015). The parachute deceleration system has been used since the 1970s Viking Program. However, future missions

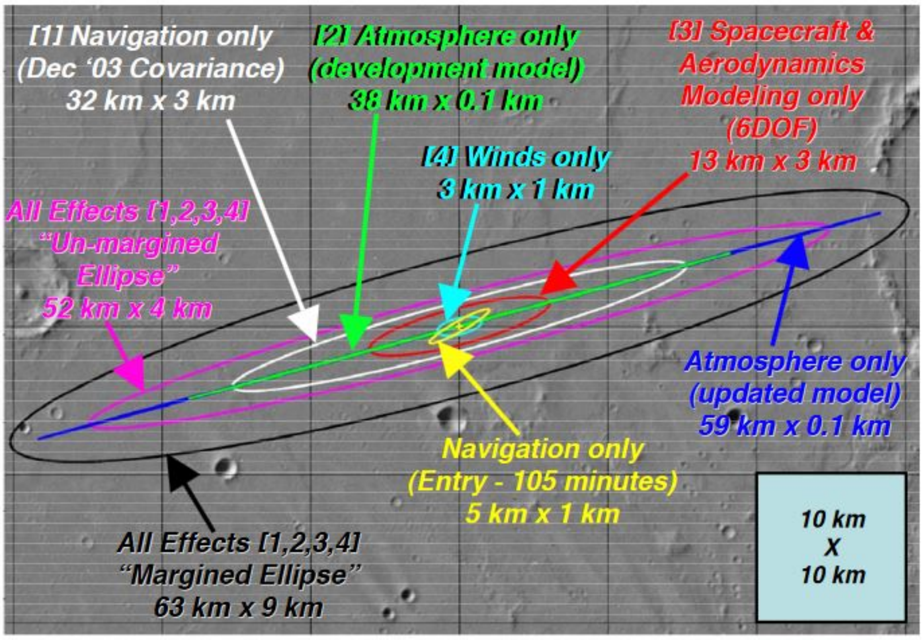


Figure 2.4: Contributions to landing ellipse dispersions (Knocke et al., 2004).

will thus require high-performance innovative landing techniques. An important aspect to consider is that of rigid hypersonic decelerators. By generating lift, the peak heat rate and deceleration are significantly reduced. Also, the abilities in terms of cross-range are improved and landing accuracy is increased (Skolnik et al., 2017).

Propulsion Systems

The utilization of a propulsive system reduces the velocity in the higher atmosphere at a lower expense compared to previously discussed aerodynamic deceleration systems. The class of propulsion systems used for ensuring contact with the surface are of a different nature because of the distinctive operational conditions. State of the art descent thrusters initiate in the supersonic regime, making use of perfect gas jets. This so-called supersonic retropropulsion has been reinvigorated by NASA after decades of stagnation in innovation. Nonetheless, maturation of these systems requires development of sensors and algorithms for the effective control and stabilization of the spacecraft. The most challenging advancement however, is the scaling of systems to larger engines for enabling precision landing of human-class vehicles (Edquist et al., 2014).

Atmospheric Characterization

For all aerodynamic flight phases of planetary landing, from aerocapture/aerobraking to entry and descent, models of the atmosphere are important. Precision landing requires guidance spacecraft to navigate through this strongly varying environment. While advancing guidance and navigation systems will certainly aid in increasing the landing accuracy, the requirement for better models is absolutely necessary. Furthermore, hypersonic aerodynamic uncertainties only accrue with an increase in mass and velocity (Bose et al., 2013). This fact is also true for MSL, of which the flow field structure analysis is especially complicated for the vortex flow (Wei et al., 2015). Characteristic for hypersonic flow is the bow shock wave being detaching from the body and laying very close to the heat shield. The future mission to Titan, Dragonfly, should aid in further characterizing the atmospheric properties of the Saturnian moon.

Sample Return Missions

A powerful type of exploration of the Red Plant are the proposed long term objective Mars sample return mission. This endeavour is important to settle the right preparations for human exploration missions, making analysis of collected rock and dust samples possible on Earth. The major space agencies, amongst them NASA and ESA, have proposed missions with an entry mass of approximately 2000 kg requiring precision landing for placement of the Mars Ascent Vehicle (Buonocore, 2011). Due to the thick nature of Titan's atmosphere a sample return is currently not realistic.

2.3. Reference Missions

This section outlines the reference missions regarding the planetary landing simulations of the conducted research. For the Mars landing cases, delineated in Section 2.3.1, the dual combination of the Mars Science Laboratory (MSL) and Mars 2020 are considered. These two missions have a similar baseline design, with a slight modification in spacecraft mass and some of the on-board instruments. In the second and final segment of this section the Precision Aerial Delivery System heritage for landing on Titan is outlined.

2.3.1. Mars Landing - MSL and Mars 2020

The first and single Martian spacecraft that has performed both a lifting and guided atmospheric entry, and utilized precision landing techniques is the Mars Science Laboratory [1]. Travelling at 5.845 km/s, the spacecraft entered the Martian atmosphere at an entry flight path angle of -15.474 deg, 125 km above MOLA reference ground. A sequence of complex autonomous operations decreased the landing ellipse from hundred(s) of kilometers for the unguided successors to a relatively small target landing ellipse of 7 × 20 km. The top-level requirement for MSL was to land the 899 kg payload within this selected target, at an altitude of 1 km above MOLA reference (Way et al., 2007). The strict landing requirement is attributed to the nature of the Gale crater, of whom the region and dimensions enforced the constraints. Eventually, MSL landed just 2.385 km away from the 4.5965° South and 137.4019° East landing site within the crater. NASA has chosen to build on the legacy of Curiosity as a design baseline for Mars 2020 (Fosse et al., 2015). For that reason scaling methodologies shall be applied to the MSL design, the approach of which has been conferred in Appendix A.

Exo-atmospheric flight

The GNC system for the EDL sequence is activated at TZERO (t = 0s), one minute after separation from the cruise stage. Way et al. (2013) continue by describing how the reaction control systems warm-up for orbital de-spin at a constant rate of 2 rpm and re-orient to the aimed attitude for entry. By means of cruise balance masses, an off-set in the CoM is provided to generate a Lift-to-Drag Ratio (L/D) of 0.24 at Mach 24. This attitude is preserved through a large portion of EI and served two purposes. As specified by Prakash et al. (2008), the landing ellipse error is reduced and the altitude at which the parachute is deployed is increased.

While a higher L/D does increase the flight time and downrange, a too high L/D may lead to the spacecraft entering a skipping flight. Avoiding this is achieved through banking, a process in which the lift vector is tilted out of the vertical plane. This by itself imposes the problem of an increased lateral motion, (at times) unwillingly increasing the cross range of the spacecraft. Other phenomena that affect the footprint dimensions are heating constraints, acceleration, dynamic pressure and the Coriolis effect. Accounting for these path constraints, a robust on-board near real time method for generating footprints for entry vehicles has been developed by Saraf et al. (2004).

Entry Interface

The official start mark of the EDL sequence is the EI, which begins after necessary key preparations procedures have been followed. At this instance in time, the vehicle is located 631.979 km downrange and 7.869 km cross range of the landing mark. As (Way et al., 2013) rightfully points out, an immense segment of the kinetic energy (99.6%) is dissipated through atmospheric friction, putting a lot of strain on the vehicle. In fact, peaks in both heating and deceleration occur during the guidance phase, making it an important portion of the EI. The MSL entry guidance algorithm is divided by Blau (2014) into three main phases:

- pre-bank phase
- range control phase
- heading alignment phase

After sensors detect that acceleration has exceeded 0.5g, the pre-bank phase and the controller commands for banking attitude commence. An acceleration trigger limit is set for the range control. During this phase the bank angle is commanded in such a way that the predicted downrange error is minimized at parachute deployment (Mendeck and Craig, 2011). The error in cross-range is preserved through possible necessary bank reversals by means of what (Prakash et al., 2008) describes as a manageable dead-band limit. This band of input values is also referred to as a neutral zone and occurs within the domain of control systems when the output is zero. They serve a purpose of preventing oscillations and reoccurring activation-deactivation series

[1] While Viking did enter through a combined ballistic-lifting configuration, a precision guidance scheme was not included into the GNC system (Euler et al., 1977).

referred to as hunting (Johnson, 2013). At a relative velocity of 900 m/s the heading alignment is initiated for the minimization of the residual of the cross-range error. The phase ends with preparations for parachute deployment by adjusting the vehicle back to a zero angle of attack profile through balance masses.

At a velocity of approximately Mach 2 and a maximum flight path angle of 5°, parachute deployment is triggered (Pollard et al., 2007). The latter is because the off-axis loads on entry spacecraft has to be limited. Meanwhile, the spacecraft has gone through a "straighten up and fly right" (SURF) operation and the azimuth has been aligned to allow for better radar measurements. Without further explanation why, the parachute trade-off lead to a single 21.5 m baseline design. For the interested reader, Sengupta et al. (2007) provides a detailed overview of MSL parachute decelerator subsystem. The prime functionality of the decelerator is to decrease the downrange velocity from 450 m/s at parachute deployment to roughly 100 m/s at backshell separation. While this reduction in velocity is benevolent, miscues accumulate in the parachute phase as a consequence of winds and atmospheric unpredictabilities. These errors are ought to be reduced by the GNC system for the powered descent phase to allow for precision landing. Preparations for this phase include jettisoning of the backshell, data acquisition of the Martian surface using TDSs, initiation of the powered descent retropropulsion thrusters and PDG system and, finally, jettisoning the parachute.

POWERED DESCENT

The main focus of the research, the powered descent phase as depicted in Fig. 2.5, commences at an approximate altitude range of 1.4 to 1.8 km and a (near vertical) velocity of 100 m/s. The two objectives of the powered descent phase for MSL are bringing the spacecraft to the necessary conditions for initiation of the Sky Crane sequence and divert the spacecraft away from the zone at which the backshell and parachute have been jettisoned (because of potential dangers, e.g., tangling the rover). The end conditions are an altitude of 23 m, a vertical velocity of 0.75 m/s and a 0 m/s horizontal velocity. Prakash et al. (2008) break the powered descent guidance (PDG) logic down into four segments:

- powered approach
- constant velocity profile
- constant deceleration profile
- down throttling

During powered approach, the vertical velocity of the spacecraft is smoothly brought to 20 m/s while the vertical velocity is nulled simultaneously. Subsequently, to accommodate for any errors in altitude, the vehicle is driven into a constant velocity profile at an altitude of 142 m. In case it turns out that the surface lays closer than initially computed, the velocity profile is discarded. During the constant deceleration phase, the spacecraft is further decreased in vertical velocity to the conditional 0.75 m/s. This occurs at a throttling rate of 90% and eventually ends at an altitude of 21 m above ground. At this moment in time half of the total fuel

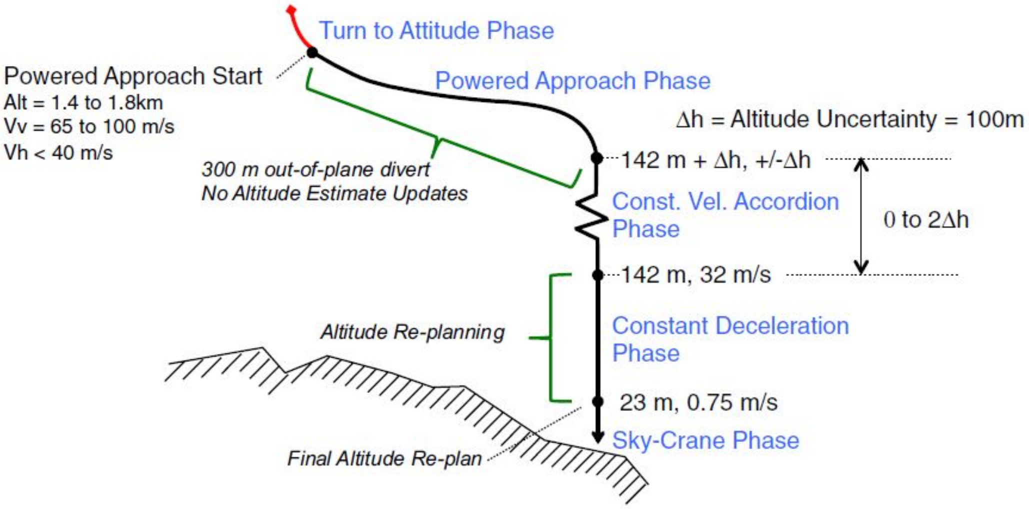


Figure 2.5: Mars Science Laboratory powered descent phase (Martin et al., 2014).

reserves (≈ 400 kg) have been depleted. Finally, the throttle down segment (in the order of 20-25%) is initiated to bring the thrust-to-weight-ratio to one. However, as Dawson et al. (2007) point out, the Mars Landing Engines (MLEs) operate highly inefficient at this setting. Thus four of the eight MLEs are brought to the near shutdown condition of 1% and the remaining four are settled at a 50% throttling setting. The switch to four MLEs introduces disturbances for whom a period of 2.5 seconds is allocated to allow for stabilization. The main engines of the precise propulsion throttling systems are required to produce of maximum thrust level of 25000 N (Guernsey and Weiss, 2013). As stated by Blackmore (2017) and Parissenti et al. (2011), throttling engines are compulsory for Mars (soft) precision landing. For that reason the specifications of the performance requirements for the MSL engines are presented in Table 2.2. In this research the assumption is made that the Mars 2020 class vehicle shall use the same number and type of engines.

Table 2.2: Required *vs* demonstrated performance of the Mars Science Laboratory mono-propellant hydrazine 400 - 3000 N and ≈ 200 s specific impulse (I_{sp}) throttling terminal descent engines (Dawson et al., 2007).

Parameter	**Requirement**	**Demonstrated**
Thrust	400 - 2998 N	31 - 3603 N
Specific impulse	191 - 217 sec	204 - 223 sec
Starts	5	7
Burn time	>220 sec	350 sec
Throughput	155 kg	278 kg
Valve response - 90%	<80 msec	50 msec

LANDING

Even though the employed Sky Crane design is the most advanced system included in the EDL architecture of MSL, its significance on the landing precision is relatively inferior. The discussion on this matter is for that reason kept short. The interested reader is referred to Budney et al. (2010) for a broad analysis on Sky Crane technologies and to Peacocke et al. (2011), for an overview of other terminal landing options. One a side note it is denoted that for human class Mars missions, i.e., order(s) of mass higher, the Sky Crane is not a feasible option. To conclude this section of Martian landing sequences, the starting and final state conditions are conferred in Table 2.3. These conditions, in which position, velocity and mass are included, formed the baseline for the landing simulation sequences and determined the design of the nominal trajectories.

Table 2.3: Mars landing initial and terminal state conditions in accordance with Açıkmeşe and Ploen (2007) and Açıkmeşe et al. (2013).

	Starting state conditions	**Final state conditions**
Position vector	$\mathbf{r}_0 = [2, 0, 1.5]^T$ km	$\mathbf{r}_f = [0, 0, 0]^T$ km
Velocity vector	$\mathbf{v}_0 = [100, 0, -75]^T$ m/s	$\mathbf{v}_f = [0, 0, 0]^T$ m/s
Spacecraft mass	$\mathrm{m}_0 = 1905$ kg	$\mathrm{m}_f > 1505$ kg
On-board fuel	$\mathrm{m}_{\mathrm{fuel}_0} = 400$ kg	$\mathrm{m}_{\mathrm{fuel}_f} > 0$ kg

2.3.2. TITAN LANDING - PRECISION AERIAL DELIVERY SYSTEM

From the perspective of studying prebiotic chemistry, Saturn's moon Titan is presumably one of the richest environments in the Solar System. (Quadrelli et al., 2019) states that studying the science of matter within Titan's atmosphere and beneath its surface is one of the most important planetary science objective. Examining the organic compounds on Titan requires spacecraft to land near regions of fluids and sediments, to be encountered near seas and lakes. Within current technological capabilities landing dispersions extend hundreds of kilometers wide, which preclude landing on large liquid areas with the exception of the beforementioned lakes at the northern high latitudes. Due to the seasons of Titan, space landing missions to these northern lakes are prevented prior the late 2030s. This entails that access to dynamic environments conducive to chemical evolution relies heavily on drift due wind. For these epitomized reasons, precision landing capabilities in an anticipated technology for exploring the habitat of Titan. In this section the mission scenario for the nominal Titan landing simulations is outlined. Starting with Fig. 2.6, which depicts landing sequence of the precision delivery system from entry to landing. Notice that due to tremendous density, this nominal landing sequence takes approximately 2.5 hours, much longer compared to the seven minutes for a

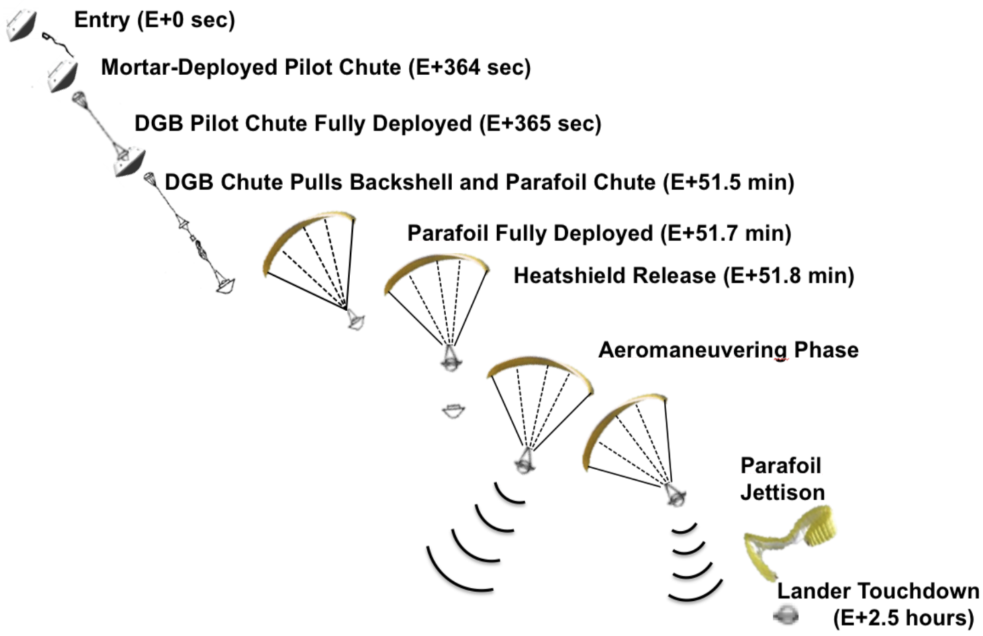

Entry (E+0 sec)
Mortar-Deployed Pilot Chute (E+364 sec)
DGB Pilot Chute Fully Deployed (E+365 sec)
DGB Chute Pulls Backshell and Parafoil Chute (E+51.5 min)
Parafoil Fully Deployed (E+51.7 min)
Heatshield Release (E+51.8 min)
Aeromaneuvering Phase
Parafoil Jettison
Lander Touchdown (E+2.5 hours)

Figure 2.6: Precision Aerial Delivery System (PADS) landing sequence on Titan (Quadrelli et al., 2019).

Mars landing. The benefit of this extended EDL sequence is the fact that less strain is put on the GNC systems in terms of computational speed, as time is less of a driving factor within the autonomy framework.

The research focus with regard to Titan targets the development of a spacecraft with minimum landing error while simultaneously adhering to the sustainable development of space technologies. Likewise to the Martian case, the main focus is the advancement of parafoil guidance and control performance. The greatest source of landing error for a Titan mission is attributed to high altitude winds, and their respective uncertainties due to lack of exploration. For missions requiring low landing accuracy, a high altitude (about 150 km) deployed parachute would suffice. For low delivery error however, an unguided drogue parachute is proposed for the initial phase, followed by a deployed guided parafoil at an altitude of 40 km (Fig. 2.6). This altitude has been chosen cause it has been proven that camera descent technologies can view the surface for estimated positioning purposes, reducing the landing error by over 100 km (Quadrelli et al., 2019). This implies that for adequate state knowledge, terrain relative navigation technologies are of particular need for a Titan mission. Current TRN technology readiness level (TRL) is at six, however, the expectation is that post the landing of the Mars 2020 mission the TRL shall be at nine. To cope with the requirements of TRN (in terms of spacecraft orientation), a pointing constraints within the guidance framework (further elaborated on in Chapter 4). Finally it is denoted that prior research at JPL for the Titan EDL sequence focused on the high altitudes regions, for which optimal path planning results have been obtained. The cases charted in this research focus on the terminal descent phase, particularly at and altitude range ξ_z of 150 - 200 m. The cross range magnitudes (ξ_x and ξ_y) at this altitude vary between -10 and 10 m, depending on wind magnitudes. The downrange velocity ϵ_z approximately varies between -7 and -10 m/s, while the cross range values (ϵ_x and ϵ_y) range between 0 and 10 m/s. For the nominal trajectories, the position vector ξ and velocity vector ϵ are randomly generated within these ranges. The total spacecraft mass (parafoil and payload) equal 201.4 kg. Notice that the total spacecraft mass at the start and end of the landing sequence remains unchanged as for a parafoil there is no mass depletion. The relevant parameters for the Titan landing simulations are summarized in Table 2.4.

Table 2.4: Titan landing initial and terminal state conditions in accordance with Quadrelli et al. (2019).

	Starting state conditions	**Final state conditions**
Position vector	$\mathbf{r}_0 = [\xi_x, \xi_y, \xi_z]^T$ km	$\mathbf{r}_f = [0, 0, 0]^T$ km
Velocity vector	$\mathbf{v}_0 = [\epsilon_x, \epsilon_y, \epsilon_z]^T$ m/s	$\mathbf{v}_f = [0, 0, 0]^T$ m/s
Spacecraft mass	$m_0 = 201.4$ kg	$m_f = 201.4$ kg

2.4. Space Systems Engineering

Vaughan et al. (2018) defines systems engineering as “an interdisciplinary approach governing the total technical effort to transform requirements into a system solution”. The paper, which is partly based on earlier work produced by Gill and Vaughan (2008), describes how systems engineering is essential for technical excellence and successful aerospace operations. The conclusions of the report where summarized into three pointers: “pay attention to details” (G. M. Low), “leave no stone unturned” (W. Braun), and “be aggressive-not passive” (L. B. James). These design philosophies are highly conducive for guaranteeing favorable processes and eventually an outstanding design. Systems engineering forms an integral part of both this research, as well as the ensuing final thesis dissertation. In this section on space systems engineering, the high level system requirements are outlined. These requirements establish the design boundaries and provide insight into the technical problem to be solved.

2.4.1. System Requirements

A system requirement is a requirement stated in engineering terms, thereby either describing capability needs, system characteristics or constraints (Gill, 2016). This section outlines the top-level system requirements imposed by the mission requirements on the GNC system for autonomous precision landing. The general top-level system requirements that hold for the GNC system as a whole have been given a **GNCR-xx** identifier. For the individual contributors (i.e., guidance, navigation and control), the identifiers are defined as **GR-xx**, **NR-xx** and **CR-xx**, respectively. The outlined requirements are based on heritage elements and on needed advances of technology priory discussed in this chapter. Nota bene: these requirements do not differentiate yet between powered and parafoil descent.

- **GNCR-01:** The GNC system shall be able to operate autonomously.
- **GNCR-02:** The GNC system shall be on-board real-time implementable.
- **GNCR-03:** The GNC system shall keep the spacecraft (aerodynamically) stable during EDL.
- **GNCR-04:** The GNC system shall utilize the same thrusters as were used for MSL.
- **GNCR-05:** The GNC system shall be numerically stable.
- **GNCR-06:** The GNC system to be designed shall take TRL restrictions into consideration.
- **GNCR-07:** The GNC system shall comply to the maximum on-board fuel/power capacity.
- **GNCR-08:** The GNC system shall guarantee safe touchdown at specified ground.
- **GNCR-09:** The closed GNC-loop shall be end-to-end verified.

- **GR-01:** The guidance system shall be able to optimally divert from nominal paths.
- **GR-02:** The guidance system shall be able to redirect the spacecraft if the optimum path is not feasible.
- **GR-03:** The guidance system shall include an advanced optimization scheme.
- **GR-04:** The guidance system shall account for translational and rotational dynamics.
- **GR-05:** The guidance system shall cope with arising dispersion accordingly.
- **GR-07:** The guidance system shall be verifiable.

- **NR-01:** The navigation system shall assume perfect state knowledge.
- **CR-01:** The control system shall close the loop robustly.
- **CR-02:** The control system shall be able to cope with the dynamics.
- **CR-03:** The control system shall pose minimum time delay on the GNC system.

Based on the the above requirements, the remainder of the chapters of this report are branched. The next chapter outlines the relevant flight dynamics within the Martian and Titian environment. The main GNC elements are outlined in Chapter 4, while the subsequent chapter focuses on the convex optimization guidance methodology. In the finals chapter the results, conclusions and recommendations are delineated.

3

Flight Dynamics

The science of flight dynamics encompasses the orientation, motion and control of flight vehicles is three dimensional space subjected to forces and moments. For spacecraft conducting a precision landing sequence, these forces and moments are of three types: aerodynamic, gravitational and propulsive. To give measure to the position and orientation of the vehicle, various reference frames are used. These reference frames, more extensively discussed in Section 3.1, are defined as a system of coordinates or a set of axes used to orientate the craft (Mulder et al., 2013). As rightfully stated in Wakker (2015), it is not possible to have a discussion on motion without defining a frame of reference to which the motion is described. Initially, one needs one frame of reference: the inertial reference frame. This suffices due to the fact that Newton's laws of motion are principally valid within this frame. However, at times it is more logical to define a flight vector in one reference frame rather than another for better understanding of the dynamics or it is easier to derive the Equations of Motion (EoM) in a particular frame. In either of these two cases the need for the use of multiple reference frames arises. After defining the necessary frames, the discussion is continued by defining the state variables in Section 3.2 and discussing the transformation between frames in Section 3.3. Methods for conversion between states are delineated in the subsequent section, after which the relevant EoMs are derived. This chapter concludes by outlining the EoMs of a parafoil subject to aerodynamics.

3.1. Reference Frames

In classical mechanics, and (re-entry) flight dynamics explicitly, a reference frame is a system of graduated lines attached to a central body. Likewise referred to as a frame of reference, this system serves to describe the position (and other state physical quantities) of specific points relative to the body over time (Hosch et al., 2006). When describing reference frames, Mooij (1994) makes a distinction between two types of bodies, namely, those that are planet centered and those belonging to the vehicle centered class. In this section, all reference frames needed to express the various relevant scalars and vectors, and derive the necessary equations of motion for planetary precision landing, are categorized in accordance with (Mooij, 1994). The discussion starts with the two planet centered reference frames after which the vehicle centered frames will be discussed. For the latter, the origin of the frames coincides with the Centre of Mass (CoM) of the vehicle. Finally it is noted that all considered frames are Cartesian and right handed and it assumed that the vehicle has one (in longitudinal direction) or more planes of geometrical symmetry.

Planet centered Inertial Reference Frame - Index I

The planet centered inertial reference frame, denoted with index I, is a right-handed orthogonal frame that is also referred to as the Newtonian reference frame in Wakker (2015). This follows from the fact that within this frame a particle either remains at rest or else in a uniform rectilinear motion if no resultant force acts upon it, which is in accordance with Newton's first law of motion [1]. As depicted in Fig. 3.1, the origin of the inertial planet centered reference frame coincides with the CoM of the central body about which the (space)craft is moving. Furthermore, the Martian equatorial plane concurs with the OX_IY_I-plane, while the Z_I-axis is directed north. The direction of the X_I-axis is settled at the zero-longitude meridian at zero time and the right-handed system is completed by the Y_I-axis

[1] Nota bene: in fact this frame is pseudo-inertial because of the motion of Mars around the Sun and Titan around Saturn.

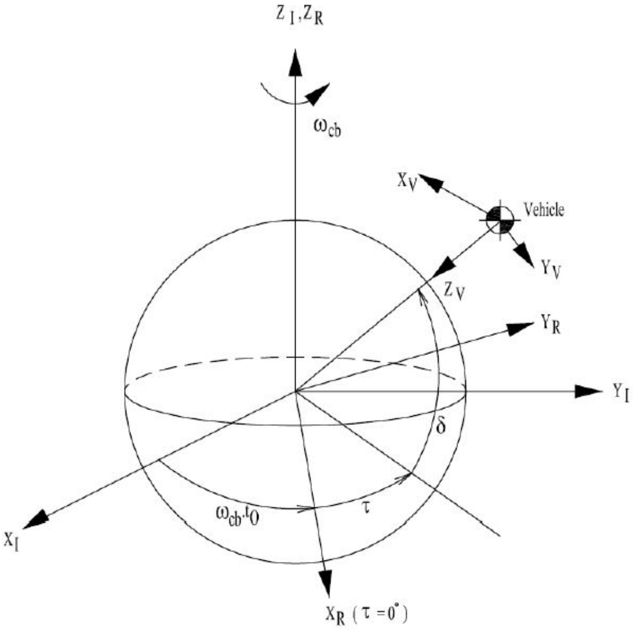


Figure 3.1: Rotating reference frame R and vertical reference frame V with respect to the inertial frame I, defined by the geocentric latitude δ and longitude τ (Mooij, 1994).

PLANET CENTERED ROTATING REFERENCE FRAME - INDEX *R*

Like the inertial reference frame, the rotating frame is fixed to the central body with the Z_R-axis pointing north. As a matter of fact, the two frames coincide at zero time and every full rotation following after that. At zero longitude the X_R-axis intersects the equator and the right-handed system is withal completed by the Y_R-axis. In Fig. 3.1 the configuration of the rotating frame R with respect to the inertial frame I is shown, with ω_{cb} being the rotational rate along the spin-axis of the planetary body [2].

VERTICAL REFERENCE FRAME - INDEX *V*

Along the radial component of the gravitational acceleration, the Z_V-axis points towards the CoM of the planetary body and perpendicular to this lies the X_V-axis in a meridian plane pointing towards the northern hemisphere (Fig. 3.1). The Y_V-axis completes the reference frame and forms the $X_V Y_V$ local horizontal plane. It is important to mention that this description of the vertical frame is only true for a perfect sphere. Since a planet is elliptical in shape, small errors are introduced to the definition.

BODY REFERENCE FRAME - INDEX *B*

Fixed to the vehicle, the X_B and Z_B axes of this frame lie in the plane of symmetry being positive backward and upward, respectively. Notice that is conform the definition used for re-entry vehicle motion analysis, in which the rear of the vehicle enters the atmosphere first (to ensure thermal protection). Again the right-handed system is completed by the Y_B-axis. The orientation of the body frame with respect to the inertial frame is portrayed in Fig. 3.2. The roll rate p, pitch rate q and yaw rate r define the rotation vector $\boldsymbol{\omega}$.

PROPULSION REFERENCE FRAME - INDEX *P*

The propulsion frame is used for analysis of the powered flight portion of the re-entry sequence. Because the X_P-axis is col-linear with the (resulting) thrust force, it plays a central role in the propulsion frame. The axis is positive in the direction of the thrust and the orientation of the propulsion frame relative to the body frame, as depicted in Fig. 3.3, is defined by the azimuth and elevation of the thrust.

[2] This rotational rate is in fact not constant because gravitational influences from other Solar System bodies cause the spin-axis to wobble (i.e. polar motion)

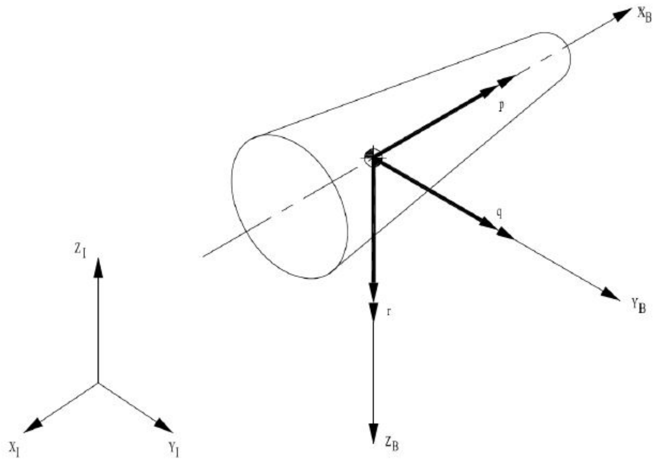


Figure 3.2: The body frame B, with the origin's location in the CoM of the vehicle, relative to the inertial frame I. The vehicle has angular rate $\boldsymbol{\omega} = (p, q, r)^T \rightarrow$ being the roll, pitch and yaw rate, respectively (Mooij, 1994).

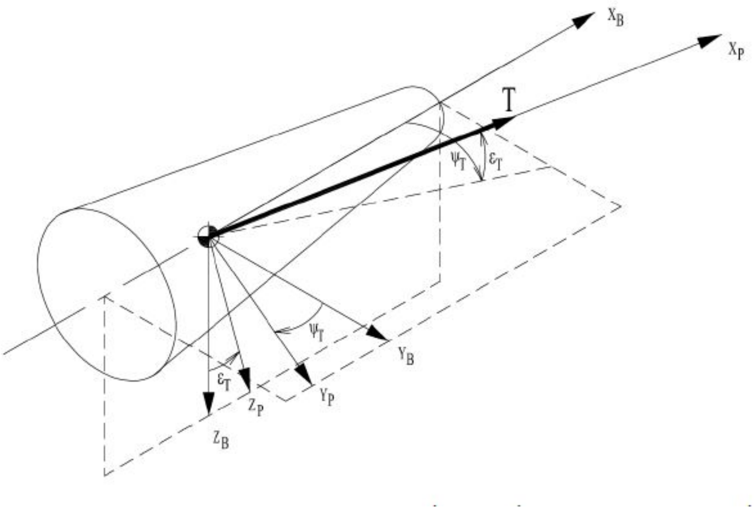


Figure 3.3: The propulsion frame P with respect to the body frame B. Indicated elevation ϵ_T and azimuth ψ_T angles of the thrust force are positive (Mooij, 1994).

Trajectory Reference Frame - Index T

Within the trajectory reference frame a distinction is made between the groundspeed and the airspeed. Along the velocity vector, the X_{TG}-axis is defined relative to the planet centered frame (groundspeed) while the X_{TA}-axis is defined relative to the atmosphere (airspeed). Both frames have the Z-axis point downwards in the vertical frame and the Y-axis completing the right-handed reference frame. Fig. 3.4 illustrates the relation between the TG-frame and the TA-frame relative to the vertical frame.

Aerodynamic Reference Frame - Index A

Alike the trajectory reference frame, a distinction is made between the groundspeed and the airspeed within the aerodynamic frame. Along the velocity vector, the X_{AG}-axis is defined relative to the planet centered frame while the X_{AA}-axis is defined relative to the atmosphere. The X_{AG}- and the X_{AA}-axis are collinear with the X_{TG}- and X_{TA}-axis, respectively. Furthermore, the Z-axes of both frames are collinear with the aerodynamic lift force (groundspeed *vs* airspeed based) but opposite in direction, and the Y-axes finalizes the system. The aerodynamics attitude angles, angle of attack α, slideslip angle β and bank angle σ, relate the body B, aerodynamic A and trajectory T frame. Fig. 3.5 shows that in the event that the vehicle is not banking, the aerodynamic and trajectory reference frames coincide.

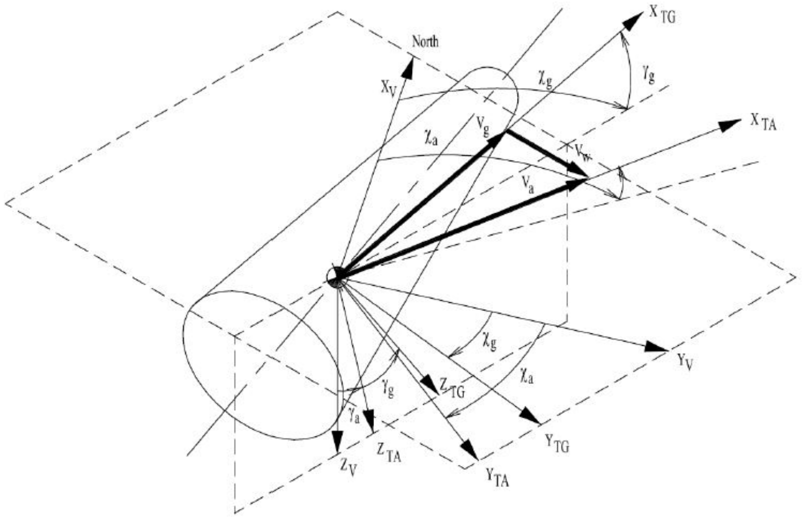


Figure 3.4: The groundspeed based trajectory frame TG and airspeed based trajectory frame TA with respect to the vertical frame V defined by the flight-path angle and heading of the respective speed vector (Mooij, 1994).

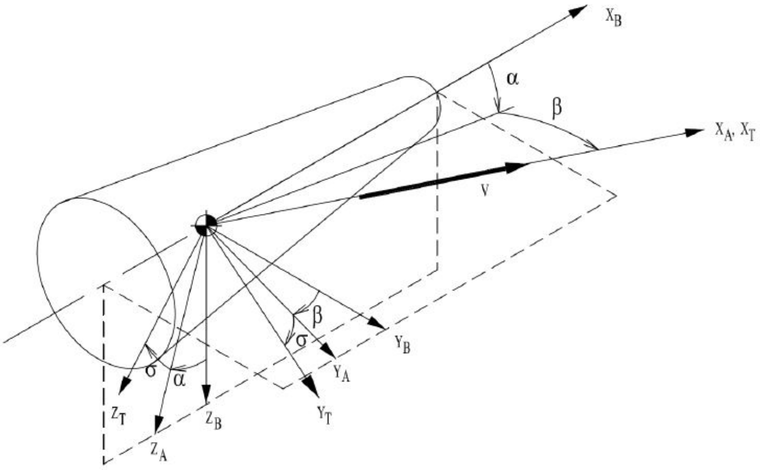


Figure 3.5: The (positive) aerodynamic angles α, β and σ defining the relation between the body frame B, the aerodynamic frame A and trajectory frame T (Mooij, 1994).

WIND REFERENCE FRAME - INDEX W

Being collinear with the wind-velocity vector, the X_W-axis is positive in northern direction for a northern wind. The Z_W-axis points positively downwards for a local horizontal plane wind, and with the Y_W-axis the frame is concluded. In case the winds are of a non-horizontal nature (Fig. 3.6), the W-frame is obtained by performing two rotations over the wind angles starting from the V-frame.

GUIDANCE REFERENCE FRAME - INDEX G

The origin of the adapted guidance frame lies at the nominal landing site with the Z_G-axis pointing upwards. The X_G-axis and the Y_G-axis point in northern and eastern direction, respectively. By assuming the reference frame to be fixed to the target on the planet, as depicted in Fig. 3.7. The guidance frame is only used during the terminal guided portion of EDL and thus the rotation of the planetary body is not included in this defined model (Furfaro et al., 2015). By assuming the frame to have an east-north orientation, the frame becomes inertial in nature (Szmuk and Açıkmeşe, 2018).

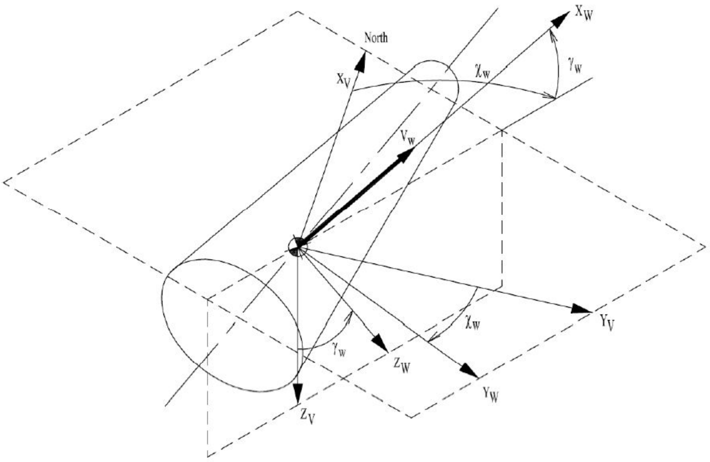


Figure 3.6: The orientation of the wind frame *W* relative to the vertical frame *V* defined by the flight-path angle and heading of the wind vector (Mooij, 1994).

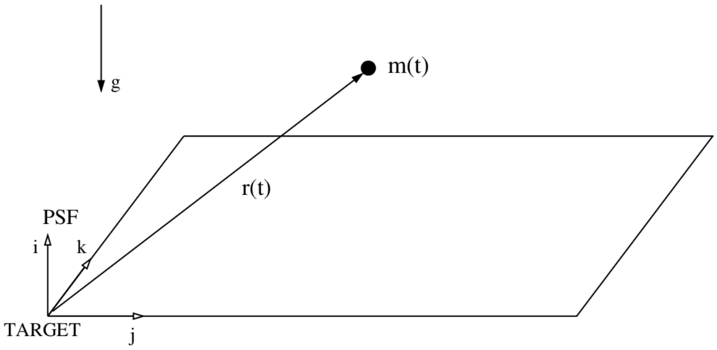


Figure 3.7: Guidance reference frame *G* to the designated landing site for planetary precision landing (Açıkmeşe and Ploen, 2007).

3.2. State Variables Definitions

To determine the mathematical state of a dynamic system, a set of variables is used defined as state variables. This set of variables encloses all imperative and linearly independent elements of the system at zero time along with the forcing functions such that the behavior over time can be determined (Nise, 2011). The variables needed to define the motion of a spacecraft are divided into two subcategories: position and velocity discussed in Section 3.2.1 and attitude and angular rates discussed in Section 3.2.2.

3.2.1. Position and Velocity

Amongst others, position state variables are often expressed in Cartesian or spherical components. Cartesian coordinates ease quantity interpretation of factors as range and altitude relative to spherical coordinates. The terminal descent guidance problem is formulated in an inertial Up-East-North reference frame of which the landing location forms the center, while the spacecraft assumes a body-frame orientation. The governing guidance equations will be outlined in the next chapter. This section outlines the position and velocity variables in an inertial frame. Due to the relative low altitude at which powered descent takes place an Cartesian presentation of state is mainly adapted in this thesis. Furthermore, the wind data are provided in Cartesian coordinates. Common EDL applications present the landing location in terms of a longitude-latitude footprint. This is less relevant for precision landing because of small longitude-latitude difference, however, it it useful to include this representation in the analysis. While a framework for both coordinate systems is provided in this section, it is important to state that Cartesian frames are utilized for the computations. The main reasons for this choice are the generally easier differentiation of the governing equations, the inherent singularity free nature and the independence of origin for the Cartesian frame.

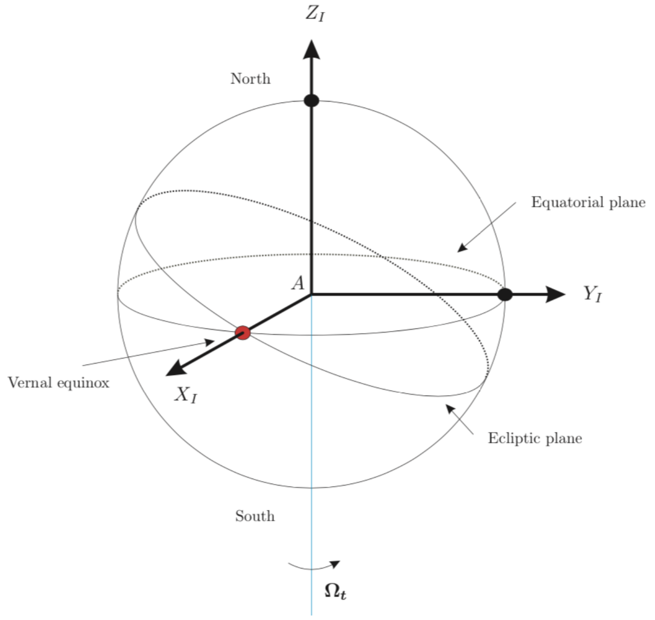


Figure 3.8: Axes X_I, Y_I and Z_I of the Earth-fixed inertial frame. The Z_I is directed to the north, the X_I passes through the equator at the vernal equinox. Completing the right-handed frame, Y_I passes through the equatorial plane. (Mulder et al., 2013).

CARTESIAN

Within the planet centered inertial and rotating reference frames discussed earlier, the position and velocity states are expressed in Cartesian components:

$$\text{Cartesian}\begin{cases} x, y, z & \text{Position} \\ \dot{x}, \dot{y}, \dot{z} & \text{Velocity} \end{cases}$$

SPHERICAL

The spherical state variables for position and velocity will in this thesis only be defined relative to the planet centered inertial frame. These elements, physically represented in Fig. 3.9. include:

$$\text{Position}\begin{cases} r & \text{distance from CoM central body to CoM vehicle} \\ \tau & \text{longitude (-180° } \leq \tau < +180°) \\ \delta & \text{latitude (-90° } \leq \delta \leq 90°) \end{cases}$$

$$\text{Velocity}\begin{cases} V_G & \text{groundspeed} \\ \gamma_G & \text{flight-path angle (-90° } \leq \gamma_G \leq 90°) \\ \chi_G & \text{heading (-180° } \leq \chi_G < 180°) \end{cases}$$

The modulus of the velocity vector, analogous to the groundspeed V_G, makes an angle γ_G (flight-path angle) with the local horizontal plane. Important to notice is that this angle is negative for entry conditions. The heading angle χ_G defines the projection of the velocity vector $\mathbf{V}$ with respect to the local north. The airspeed V_A is related to the groundspeed through the wind speed V_W such that

$$V_A = V_G + V_W \tag{3.1}$$

In Section 3.4 the conversions between states is presented, including a short discussion on the airspeed based velocity. A more detailed derivation regarding the relevant parameters is provided in Section 3.7.

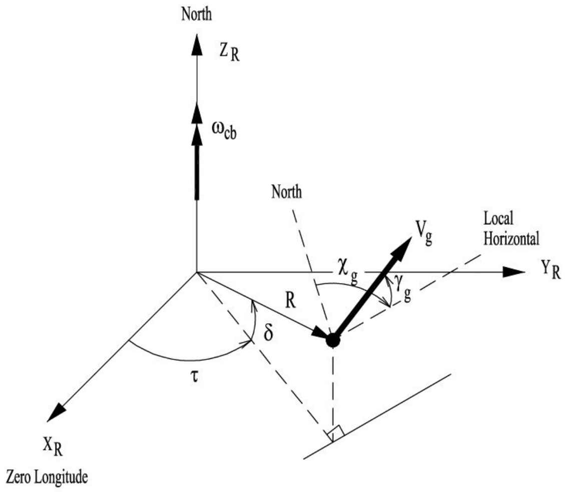


Figure 3.9: The six spherical flight parameters for position (R, τ, δ) and velocity $(V_G, \gamma_G, \chi_G])$. All angles are depicted in their respective positive direction (Mooij, 1994).

3.2.2. ATTITUDE AND ANGULAR RATES

From a mathematical perspective, the attitude of a vehicle is defined as the orientation of a body-fixed frame relative to an external frame. Mathematical attitude dynamics models are used to analyze spacecraft motion characteristics and design GNC systems. Furthermore, these models are used for the verification of designed systems. For the description of spacecraft motion, rigid body dynamics forms the basis and provides first-order analyses (Chu, 2017). For precision landing simulations the assumption of rigid body dynamics is adequate due to the reason that non-rigid effects (e.g. moving on-board liquids or mechanical devices) have minimal effect (Queen et al., 2008).

In this research three types of orientation definitions are considered: Euler angles, aerodynamic angles and quaternions. The angular rates, i.e., the roll rate p, pitch rate q and the yaw rate r, form the rotation vector ω, define the velocity of a body relative to the inertial frame and are expressed along the axes of the body as was shown in Fig. 3.2. Generally, quaternions shall be utilized in the thesis research for parameterization because of their singular free nature, mainly due to the minimal nature of Euler angles. A discussion on Euler and aerodynamic angles has been included, because these representation forms aid in visualization of motion, a matter that the quaternion representation (often) lacks.

CLASSICAL ATTITUDE ANGLES

The classical attitude angles, i.e., the roll angle ϕ, the pitch angle θ and the yaw angle ψ, are a class of Euler angles [3]. These angles, depicted in Fig. 3.10, are characterized by means of an Euler transformation, starting from the inertial frame. The order in which the transformation is carried out is of crucial importance, reason being for discussing this more extensively in Section 3.3. The possibility also exits to define the vehicle's attitude relative to the local horizontal plane but for applications to re-entry problems the presentation with respect to the inertial space suffices.

AERODYNAMIC ANGLES

The angle of attack α, the slideslip angle β and the bank angle σ are part of the class of aerodynamic angles that also form a set of Euler angles. When used in the equations of motion these angles are defined relative to the groundspeed. For a graphical representation of the (positive) aerodynamic angles the reader is referred back to Fig. 3.5.

QUATERNIONS

The four Euler parameters, also known as quaternions, define the orientation of a vehicle relative to the inertial planet centered frame:

$$\mathbf{q} = \left(q_0, q_1, q_2, q_3\right)^T = \begin{pmatrix} q_0 \\ \mathbf{q}_{1:3} \end{pmatrix} \tag{3.2}$$

[3] "The Euler angles are three angles introduced by Leonhard Euler to describe the orientation of a rigid body with respect to a fixed coordinate system" - Source: `https://en.wikipedia.org/wiki/Euler_angles` - *Last accessed on March 29, 2018*

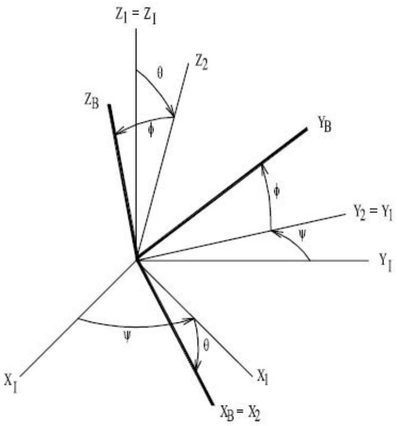


Figure 3.10: Euler angles: roll angle ϕ, the pitch angle θ and the yaw angle ψ about the X_I-, Y_I- and Z_I-axis, respectively (Mooij, 2017).

Each quaternion is a four-dimensional hyper-complex number (one real and three imaginary numbers) and the four parameters are mutually interdependent in accordance with the relation provided in Eq. (3.3). The main advantage of this set of four coordinates is the fact that singularities are ruled out. Because a physical interpretation is difficult for quaternions, a graphical representation is left out.

$$q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1 \tag{3.3}$$

Unfortunately, there is no unique definition to quaternions which leads to different conventions. Therefore the model utilized by Razgus et al. (2017) and by the Jet Propulsion Laboratory (2003) shall be taken as a baseline. Because the representation will be used for attitude description only, normalized unit quaternions suffice. The product of two quaternions then follows from:

$$[\mathbf{q}\otimes] = \begin{bmatrix} q_4\mathbf{I_3} - [\mathbf{q}_{1:3}\times] & \mathbf{q}_{1:3} \\ -\mathbf{q}_{1:3}^T & q_4 \end{bmatrix} \tag{3.4}$$

$$[\mathbf{q}\odot] = \begin{bmatrix} q_4\mathbf{I_3} + [\mathbf{q}_{1:3}\times] & \mathbf{q}_{1:3} \\ -\mathbf{q}_{1:3}^T & q_4 \end{bmatrix} \tag{3.5}$$

For which the following holds:

$$\mathbf{q}_1 \otimes \mathbf{q}_2 = \mathbf{q}_1 \odot \mathbf{q}_2 \tag{3.6}$$

3.3. Transformation of Frames

Axis transformations commonly consist of both translational and rotational motion, leading to a six Degree-of-Freedom (6DoF) analysis. This refers to the freedom of a rigid body (e.g., spacecraft) to move in three-dimensional space. The translation movements include surge, sway and heave in the perpendicular axes (longitudinal, lateral and normal), expressed by vector **T**. The rotations about the three axes are termed roll, pitch and yaw and defined by transformation matrix $\mathbf{C}_{B,A}$ (for a rotation from the A to frame B). Expressing a vector $\mathbf{v}_A$ defined in axis system A relative to another axis system B yields:

$$\mathbf{v_B} = \mathbf{T} + \mathbf{C}_{B,A}\mathbf{v_A} \tag{3.7}$$

Transformations between the right-handed frames of reference discussed in the preceding section are conventionally executed by means of unit axis-rotations, a Directional cosine Matrix (DCM) or unit quaternions. This section focuses on the mathematical description of these transformation operations, delineated in Section 3.3.1. The relevant transformations for precision landing are outlined in Section 3.3.2.

3.3.1. Transformation Operations

Notice that any transformation matrices in this research are generally denoted by **C**. Only quaternion transformation have a different convention. For visualization of the respective frames, the reader is referred back to the previous section.

Unit Axis-rotations

The unit-axis transformation are positive Euler angle rotations in accordance with the right-hand rule, a common mnemonic used in physics for grasping insight regarding orientation convention in $\mathbb{R}^3$ Euclidean space. The transformation is a sequence of rotations of a reference frame around its own X-, Y-, or Z-axis by means of an arbitrary angle α. For the conversion, rotation matrices are utilized described by Eqs. (3.8) - (3.10).

$$\mathbf{C}_1(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha \\ 0 & -\sin\alpha & \cos\alpha \end{bmatrix} \tag{3.8}$$

$$\mathbf{C}_2(\alpha) = \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & & \cos\alpha \end{bmatrix} \tag{3.9}$$

$$\mathbf{C}_3(\alpha) = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.10}$$

Notice that the subscripts 1, 2 and 3 indicate the X- Y- and Z-axis, respectively. The total transformation may be described as a consolidation of the orthonormal unit rotation matrices $\mathbf{C}_1$, $\mathbf{C}_2$ and $\mathbf{C}_3$. The product of these matrices is also of an orthonormal nature and the inverse is the transpose.

Consider two frames of reference denoted as A and B, whose inter-rotation can be decomposed in sequential unit axis transformations. The transformation shown in Eq. (3.11), made up of three successful rotations about the negative X-axis and Z-axis and the positive Y-axis, results in matrix $\mathbf{C}_{B,A}$.

$$\mathbf{C}_{B,A} = \mathbf{C}_2(\alpha_2)\mathbf{C}_3(-\alpha_3)\mathbf{C}_1(-\alpha_1) \tag{3.11}$$

The matrix $\mathbf{C}_{B,A}$ practically means that the transformation is defined from frame A to frame B. The inverse of this matrix is obtained by interchanging the order of rotation and changing the sign of the angles:

$$\mathbf{C}_{A,B} = \mathbf{C}_{B,A}^{-1} = \mathbf{C}_{B,A}^{T} = \mathbf{C}_1(\alpha_1)\mathbf{C}_3(\alpha_3)\mathbf{C}_2(-\alpha_2) \tag{3.12}$$

The Euler angle representation form is a minimum one, in the sense that a minimum number of parameters are involved. Singularities might arise in the process of inversion, which is necessary for derivation of the kinematic equations of motion. These singularities occur when the determinant of a matrix equals zero, a phenomenon known as gimbal lock. While physically this would occur at an angle of ± 90° (vertical maneuvers), it not completely unlikely that such a situation might arise and therefore it will not be ruled out.

Direction cosine Matrix

The cosine of the angles formed between unit vectors of two reference frames form the entries of the DCM (also referred to as rotation matrix or coordinate transformation matrix), which describes the relative attitude. Anew consider two reference frames denoted by A and B, both having a right-handed set of three orthogonal unit vectors [$\mathbf{a}_1$, $\mathbf{a}_2$, $\mathbf{a}_3$] and [$\mathbf{b}_1$, $\mathbf{b}_2$, $\mathbf{b}_3$]. The DCM for the transformation from A to B may be written as:

$$\mathbf{C}_{A,B} = \begin{pmatrix} \mathbf{b_1} \\ \mathbf{b_2} \\ \mathbf{b_3} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{a_1} & \mathbf{a_2} & \mathbf{a_3} \end{pmatrix} \tag{3.13}$$

Due to the orthogonal nature of each of the unit vectors of A and B, the DCM is an orthonormal matrix with:

$$\mathbf{C}\mathbf{C}^T = I = \mathbf{C}^T\mathbf{C} \quad \text{or} \quad \mathbf{C}^{-1} = \mathbf{C}^T \quad \text{with} \quad \det\mathbf{C} = 1 \tag{3.14}$$

The inverse of the DCM matrix from A to B is therefore obtained in a same way as for the Euler angles:

$$\mathbf{C}_{A,B} = \mathbf{C}_{B,A}^{-1} = \mathbf{C}_{B,A}^{T} \tag{3.15}$$

The nine parameter representation model of the DCM uses a total six numbers of parameters more relative to the minimum of parameters (three), making parametrization burdensome in terms of computation. Main advantage however, is the fact that no singularities are introduced (Choukroun, 2014). On a final note, one can obtain the attitude angles about the X- Y- and Z-axis from the DCM through Eq. (3.16), in which atan2 stands for the four-quadrant inverse tangent.

$$\begin{aligned} \theta &= -\arcsin(C_{13}) \\ \varphi &= \text{atan2}(C_{23}, C_{33}) \\ \psi &= \text{atan2}(C_{12}, C_{11}) \end{aligned} \tag{3.16}$$

UNIT QUATERNIONS

Unit quaternions are quaternions represented in their normalized form obtained through:

$$||\mathbf{q}|| = \frac{\mathbf{q}}{|\mathbf{q}|} \tag{3.17}$$

If $\mathbf{q}_{B,A}$ denotes the attitude quaternion from frame A to frame B the operation may be decomposed by using the inertial frame as an intermediary, as has been done in Eq. (3.18). Vector projections onto either frames then follows from Eq. (3.19).

$$\mathbf{q}_{B,A} = \mathbf{q}_{B,I} \otimes \mathbf{q}_{I,A} \tag{3.18}$$

$$\mathbf{v}_B = \mathbf{q}_{B,A} \otimes \mathbf{v}_A \otimes \mathbf{q}_{A,B} \tag{3.19}$$

A favored parametrization for spacecraft attitude are the unit quaternions due to their minimal nature (three) and the fact that they are free from singularities (Lee and Mesbahi, 2017) (Joshi et al., 1995). Disadvantage however, is the difficult physical interpretation. Through substitution of the defined quaternions, DCM elements can be expressed in terms of these quaternions:

$$\mathbf{C}_{A,B} = \mathbf{C}(\mathbf{q}) = \begin{bmatrix} 1-2(q_2^2+q_3^2) & 2(q_1q_2+q_0q_3) & 2(q_1q_3-q_0q_2) \\ 2(q_1q_2-q_0q_3) & 1-2(q_1^2+q_3^2) & 2(q_2q_3+q_0q_1) \\ 2(q_1q_3+q_0q_2) & 2(q_2q_3-q_0q_1) & 1-2(q_1^2+q_2^2) \end{bmatrix} \tag{3.20}$$

3.3.2. TRANSFORMATIONS FOR ENTRY AND DESCENT APPLICATIONS

Frame transformation relevant for deriving the equations of translational and rotational re-entry and descent motions are presented in this section. Consider two successive rotations from A" to A represented by:

$$A" \rightarrow A' \quad \mathbf{C}_{A',A"} \qquad A' \rightarrow A \quad \mathbf{C}_{A,A'} \tag{3.21}$$

$$A" \rightarrow A \quad \mathbf{C}_{A,A"} = \mathbf{C}_{A,A'}\mathbf{C}_{A',A"} \tag{3.22}$$

In Table 3.1, the angle and rotational rate transformations between reference frames for planetary re-entry applications are presented. The relevant quaternion transformation matrix corresponds to the switch from the body frame to the inertial frame, for which the relation is provided by Mooij and Chu (2002):

$$\mathbf{C}_{I,B} = \begin{bmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 2(q_1q_2+q_0q_3) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_2q_3+q_0q_1) & q_0^2-q_1^2-q_2^2+q_3^2 \end{bmatrix} \tag{3.23}$$

3.4. STATE CONVERSIONS

For reasons defined earlier in this chapter, the main representation forms to be used in the computations for the thesis research shall be the Cartesian form for position and velocity, and quaternions for attitude and angular rates. However, as retrieved data is at not always provided in the preferred form and at times it is desired to interpret data visually, conversions between the various states might come in handy.

In this section the mathematical procedure of transforming between various state variables is outlined. Primarily, the transformation from spherical to Cartesian transformation in Section 3.4.1. The section is concluded by describing the Euler reciprocus to quaternion state transformation. Notice that only final forms are presented, for full derivations the reader is referred to Mooij (1994).

3.4.1. SPHERICAL RECIPROCUS TO CARTESIAN

Relating position elements of spherical and Cartesian is relatively simple compared to the more complex velocity relations. For both cases however, derivations are left aside and only final results are presented.

Table 3.1: Transformation between reference frames for planetary re-entry applications

Frames	Transformation	Variables		
		Symbol	Unit	Description
R → **I**	$\mathbf{C}_{I,R} = \mathbf{C}_3(-\omega_{cb}t)$	ω_{cb}	rad/s	rotational rate of the central body
		t	s	time from epoch
V → **R**	$\mathbf{C}_{R,V} = \mathbf{C}_3(-\tau)\mathrm{C}_2(\frac{\pi}{2}+\delta)$	τ	rad	planet centered longitude
		δ	rad	planet centered latitude
W → **V**	$\mathbf{C}_{V,W} = \mathbf{C}_3(-\chi_w)\mathrm{C}_2(-\gamma_w)$	χ_w	rad	heading for wind vector
		γ_w	rad	flight-path angle for wind vector
T → **V**	$\mathbf{C}_{V,W} = \mathbf{C}_3(-\chi)\mathrm{C}_2(-\gamma)$	χ_A , χ_G	rad	heading for air- or groundspeed
		γ_A , γ_G	rad	flight-path angle for air- or groundspeed
V → **G**	$\mathbf{C}_{V,G} = \mathbf{C}_2(\pi)$	-	-	-
A → **T**	$\mathbf{C}_{T,A} = \mathbf{C}_1(\sigma)$	σ_A , σ_G	rad	bank-angle based on air- or groundspeed
B → **A**	$\mathbf{C}_{A,B} = \mathbf{C}_3(\beta)\mathrm{C}_2(-\alpha)$	α_A, α_G	rad	angle-of attack w.r.t. air- or groundspeed
		β_A , β_G	rad	angle of slideslip w.r.t. air- or groundspeed
B → **P**	$\mathbf{C}_{P,B} = \mathbf{C}_2(\epsilon_T)\mathrm{C}_3(\psi_T)$	ϵ_T	rad	elevation angle thrust force
		ψ_T	rad	azimuth angle thrust force
V → **B**	$\mathbf{C}_{V,B} = \mathbf{C}_3(-\psi)\ \mathbf{C}_2(-\theta)\ \mathbf{C}_1(-\phi)$	ϕ	rad	attitude angle about the X-axis
		θ	rad	attitude angle about the Y-axis
		ψ	rad	attitude angle about the Z-axis

POSITION ELEMENTS

The prominent equations relating the spherical with respect to the Cartesian position within the R-frame are:

$$\begin{aligned} r &= \sqrt{x_R^2 + y_R^2 + z_R^2} \\ \tau &= \arctan\left(\frac{y_R}{x_R}\right) \\ \delta &= \arcsin\left(\frac{z_R}{\sqrt{x_R^2 + y_R^2 + z_R^2}}\right) \end{aligned} \tag{3.24}$$

The inverse is also true, namely:

$$\begin{aligned} x_R &= r\cos\delta\cos\tau \\ y_R &= r\cos\delta\sin\tau \\ z_R &= r\sin\delta \end{aligned} \tag{3.25}$$

VELOCITY ELEMENTS

The Cartesian velocity elements are related to spherical velocity components through the following identified relations:

$$\begin{aligned} v_x &= V_G\cos\gamma_G cos\chi_G \\ v_y &= V_G\cos\gamma_G sin\chi_G \\ v_y &= -V_G\sin\gamma_G \end{aligned} \tag{3.26}$$

These equations represent the Cartesian velocity components in the vertical frame. The components in the planet centered rotating frame are easily found by applying the transformation as specified in Table 3.1. From the Cartesian velocity components, neglecting the presence of wind, the groundspeed is derived:

$$V_G = \sqrt{u^2 + v^2 + w^2} \tag{3.27}$$

The two relevant spherical velocity elements are then found by:

$$\chi_G = \arctan\left(\frac{v_y}{v_x}\right) \quad , \quad \gamma_G = \arcsin\left(\frac{v_z}{\sqrt{v_x^2 + v_y^2}}\right) \tag{3.28}$$

The governing wind equations will be based on Cartesian components in the research to maintain a logic overlap with the previously defined external forces. In the final section of this chapter matters concerning wind are outlined. This section is completed by providing the computational scheme for obtaining airspeed based velocity components:

1. Obtain the available wind parameters → Section 3.7.
2. Apply a transformation to these parameters if they are not provided in Cartesian components
3. Compute the airspeed based velocity V_A
4. Compute the airspeed based angle of attack α_A and sideslip angle β_A
5. In case the aerodynamic coefficients are not defined w.r.t. a point coinciding with the CoM of the vehicle, a linear velocity is added to account for the rotation
6. Compute the airspeed based aerodynamic force coefficients C_D and C_L
7. Compute the airspeed based aerodynamic forces
8. Apply a transformation to obtain the groundspeed based aerodynamic force using the airspeed based flight-path angle γ_A and heading χ_A
9. Evaluate the equations of motion
10. Update the state vector

The groundspeed is defined as the sum of the airspeed and the atmospheric velocity relative to the ground:

$$\boldsymbol{V}_{B,G} = \boldsymbol{V}_{B,A} + \boldsymbol{V}_{A,G} \tag{3.29}$$

The wind velocity vector is expressed in the rotating frame as:

$$\boldsymbol{V}_{W,R} = \begin{pmatrix} u_W & v_W & w_W \end{pmatrix}^T \tag{3.30}$$

Combining the expression of Eq. (3.29) and (3.30) yields the decomposition of the groundspeed such that:

$$\boldsymbol{V}_{B,R} = \boldsymbol{V}_{B,W} + \boldsymbol{V}_{W,R} \tag{3.31}$$

Indicating the velocity relative to the atmosphere with subscript A and rewriting the above equation produces the following set of equations:

$$\boldsymbol{V}_{B,W} = \boldsymbol{V}_{B,R} - \boldsymbol{V}_{W,R} = \begin{pmatrix} u_A & v_A & w_A \end{pmatrix}^T \quad \rightarrow \quad V_A = \sqrt{u_A^2 + v_A^2 + w_A^2} \tag{3.32}$$

The airspeed based angle of attack and sideslip angle follow from the body-frame velocity components:

$$\alpha_A = \arctan\left(\frac{w_A}{u_A}\right) \tag{3.33}$$

$$\beta_A = \arctan\left(\frac{v_A}{\sqrt{u_A^2 + w_A^2}}\right) \tag{3.34}$$

By means of these angles the respective force and moment coefficients are derived. The subsequent steps for obtaining the forces and moments in the right frame have been described above. The remainder of the required steps for wind inclusion are described in Section 3.7.

3.4.2. EULER RECIPROCUS TO QUATERNION

The interrelation between classical Euler angles and the four Euler parameters (quaternions) are provided in this section. Again, final transformation forms are presented without the inclusion of derivations.

EULER ANGLES TO QUATERNION

The sequence of the rotation plays an important role in the transformation from Euler angles to quaternions. Beneath the 3-2-1 transformation is considered, while for any other sequence the order is swapped.

$$\begin{aligned}\mathbf{q} &= \begin{pmatrix}\cos(\psi/2)\\0\\0\\\sin(\psi/2)\end{pmatrix}\begin{pmatrix}\cos(\theta/2)\\0\\\sin(\theta/2)\\0\end{pmatrix}\begin{pmatrix}\cos(\phi/2)\\\sin(\phi/2)\\0\\0\end{pmatrix}\\ &= \begin{bmatrix}\cos(\phi/2)\cos(\theta/2)\cos(\psi/2)+\sin(\phi/2)\sin(\theta/2)\sin(\psi/2)\\\sin(\phi/2)\cos(\theta/2)\cos(\psi/2)-\cos(\phi/2)\sin(\theta/2)\sin(\psi/2)\\\cos(\phi/2)\sin(\theta/2)\cos(\psi/2)+\sin(\phi/2)\cos(\theta/2)\sin(\psi/2)\\\cos(\phi/2)\cos(\theta/2)\sin(\psi/2)-\sin(\phi/2)\sin(\theta/2)\cos(\psi/2)\end{bmatrix}\end{aligned} \tag{3.35}$$

EULER ANGLES TO QUATERNION

Finally, the Euler angles may be obtained from the quaternions through the following relation:

$$\begin{pmatrix}\phi\\\theta\\\psi\end{pmatrix}=\begin{bmatrix}\mathrm{atan2}\left(2q_0q_1+q_2q_3),1-2(q_1^2+q_2^2)\right)\\\mathrm{asin}\left(2(q_0q_2-q_1q_3)\right)\\\mathrm{atan2}\left(2q_0q_3+q_1q_2),1-2(q_2^2+q_3^2)\right)\end{bmatrix} \tag{3.36}$$

3.5. EQUATIONS OF MOTION

Relative to the speed of light spacecraft travel at low velocities and relative to massive stars and black holes spacecraft are negligible in weight. For that reason both special and general relativity are not considered in the derivation of the EoM of spacecraft entering a planetary atmosphere. The relevant equations presented in this chapter are based on classical Newtonian mechanics. To account for a 6 DoF environment, the section is subdivided into EoM for translational motion in Section 3.5.2 and EoM for rotational motion in Section 3.5.4. The former describes the motion of the CoM (position and velocity) while the latter describes the motion around the CoM (angular rates). For either cases a non-elastic vehicle and a variable mass m are assumed. Integrators will be used for the propagation of the equations. In that same section simulation methods are described for analysis of the motion.

3.5.1. EXTERNAL FORCES

A spacecraft entering and landing on a planetary body encounters external forces during the sequence. Based on their nature, these forces are categorized as follows:

- Aerodynamics
- Gravity
- Propulsion (thrust)

AERODYNAMIC FORCES

The aerodynamic forces are defined in the airspeed-based aerodynamic AA-frame and consists of drag D, side S and lift L. These forces depend on the respective force coefficients C_D, C_S and C_L, the vehicle reference surface S_{ref} and the dynamic pressure $\bar{q}$:

$$\mathbf{F}_{A,AA} = -\begin{pmatrix}D\\S\\L\end{pmatrix} = -\begin{pmatrix}C_D\bar{q}S_{ref}\\C_S\bar{q}S_{ref}\\C_L\bar{q}S_{ref}\end{pmatrix} \tag{3.37}$$

The dynamic pressure is a quantity that depends on the density of the fluid (Martian atmosphere) and the airspeed-based velocity component V_A, related through Eq. (3.38). The force coefficients depend on the Mach number M, the angle of attack α and sideslip angle β (all airspeed based).

$$\bar{q} = \frac{1}{2}\rho V_A^2 \tag{3.38}$$

The Mach number is related to the airspeed velocity and the speed of sound $a = \sqrt{\gamma RT}$ through Eq. (3.39). In this equation γ is the ratio of specific heat, R the gas constant and T the absolute gas temperature.

$$M = \frac{V_A}{a} = \frac{V_A}{\sqrt{\gamma RT}} \tag{3.39}$$

In the above equations the presence of wind is ignored. Furthermore, when assuming a rotational symmetric spacecraft, the side force S will always be zero. Re-entry guidance system conventionally depend on angle of attack and bank angle control, rendering the slideslip angle useless. However, the inclusion of the wind effect makes the latter assumption invalid and must thus be accounted for. In Section A.3 the assumptions regarding the force coefficients have been outlined.

Gravitational Force

The gravitational force is expressed in the planet centered rotating frame R and depends on the gravitational acceleration $\mathbf{g}$ and the instant mass of the vehicle m:

$$\mathbf{F}_{G,R} = m\mathbf{g} = m \begin{pmatrix} g_x \\ g_y \\ g_z \end{pmatrix} \tag{3.40}$$

When landing on very irregular bodies with surfaces that are rather unknown, accurate knowledge is required of the gravity field. Nonetheless, uniform gravity fields may be assumed for both the Mars and Titan cases simulated in this thesis due to the relative low altitude at which the sequence starts. For real-time precision landing applications, efficient cpu-time is a more important factor, which would entail the choice of a spherical distribution.

Propulsive Force

It is most convenient to express the propulsive force, also defined as thrust T, in the propulsion frame. As such, the thrust vector is collinear with the X-axis of the propulsion frame:

$$\mathbf{F}_{T,P} = \begin{pmatrix} T \\ 0 \\ 0 \end{pmatrix} \quad \text{with} \quad T = I_{sp}\dot{m}g_0 \tag{3.41}$$

Assuming a number of i thrusters, the location $\mathbf{r}_i$, magnitude T_i, elevation angle ϵ_{T_i} and azimuth angle ψ_{T_i} of the sum of the i^{th} thrusters determine the total T magnitude and the moments on the spacecraft. For the orientation of the two respective angles the reader is referred back to Fig. 3.3.

3.5.2. Equations of Translational Motion

Now that the relevant external forces have been defined, the equations of translational motion can be delineated. The equations as defined in this section are to be used for the propagation of the state of the vehicle. This integration procedure shall utilize the conventional inertial frame with position and velocity state variables defined in Cartesian components. Notice that it is required to transform all relevant forces to the inertial frame before propagating the state. The equations of translational motion read:

$$\mathbf{V}_I = \frac{d\mathbf{r}_I}{dt} \tag{3.42}$$

$$\mathbf{a}_I = \frac{\mathbf{F}_I}{m} = \frac{d\mathbf{V}_I}{dt} \tag{3.43}$$

Notice that the acceleration is computed through the application of the second law of Newton. The total inertial force $\mathbf{F}_I$ accounts for a variable mass system due propulsion in the form of the thrust force, such that

$$\mathbf{F}_I = m\frac{d^2\mathbf{r}_{cm}}{dt^2} + 2\boldsymbol{\omega} \times \int_m \frac{\delta \tilde{\mathbf{r}}}{\delta t} dm + \int_m \frac{\delta^2 \tilde{\mathbf{r}}}{\delta t^2} dm \tag{3.44}$$

The various elements carry the following meaning:

- Total external forces defined in the inertial frame: $\boldsymbol{F}_I$
- Acceleration of the CoM relative to the inertial frame: $\frac{d^2\mathbf{r}_{cm}}{dt^2}$
- Coriolis force due change in mass distribution: $\mathbf{F}_C = -2\boldsymbol{\omega} \times \int_m \frac{\delta \tilde{\mathbf{r}}}{\delta t} dm$
- Relative force due change in mass distribution: $\mathbf{F}_{rel} = -\int_m \frac{\delta^2 \tilde{\mathbf{r}}}{\delta t^2} dm$

3.5.3. EXTERNAL MOMENTS

A spacecraft entering and landing on a planetary body also encounters external moments during the sequence. These moments are likewise categorized based on their nature, which is the similar to that of the forces: aerodynamics, gravity and propulsion. Because the gravity force works in the CoM of the spacecraft, no moment is produced about that same center. For the aerodynamic and propulsive forces this is typically not the case. The resulting moments are both defined in the body frame:

$$\mathbf{M}_{A,B} = \begin{pmatrix} \mathscr{L} \\ \mathscr{M} \\ \mathscr{N} \end{pmatrix} = \begin{pmatrix} C_l \bar{q} S_{ref} b_{ref} \\ C_m \bar{q} S_{ref} b_{ref} \\ C_n \bar{q} S_{ref} b_{ref} \end{pmatrix} \quad \text{and} \quad \mathbf{M}_{P,B} = \begin{pmatrix} M_{P_x} \\ M_{P_y} \\ N_{P_z} \end{pmatrix} \tag{3.45}$$

In Section A.3 the assumptions regarding the moment coefficients have been outlined.

3.5.4. EQUATIONS OF ROTATIONAL MOTION

Similar to the translation equations, the equations of rotational motion will be expressed in the inertial frame I. The governing equation follows from:

$$\mathbf{M}_{cm} + \mathbf{M}_C + \mathbf{M}_{rel} = \tilde{\boldsymbol{M}}_{cm} = \int_m \tilde{\boldsymbol{r}} \times (\frac{d\boldsymbol{\omega}}{dt} \times \tilde{\boldsymbol{r}}) dm + \int_m \tilde{\boldsymbol{r}} \times [\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \tilde{\boldsymbol{r}})] dm \tag{3.46}$$

The various elements carry the following meaning:

- Total external moment about CoM of spacecraft: $\boldsymbol{M}_{cm}$
- Coriolis moment due varying mass distributions: $\boldsymbol{M}_C = -2\int_m \tilde{\boldsymbol{r}} \times (\boldsymbol{\omega} \times \frac{\delta \tilde{\boldsymbol{r}}}{\delta t}) dm$
- Relative moment due varying mass distributions: $\boldsymbol{M}_{rel} = -\int_m \tilde{\boldsymbol{r}} \times (\frac{\delta^2 \tilde{\boldsymbol{r}}}{\delta t^2}) dm$
- Pseudo external moment about CoM of spacecraft: $\tilde{\boldsymbol{M}}_{cm}$
- Apparent moment due to angular acceleration relative to I-frame: $\int_m \tilde{\boldsymbol{r}} \times (\frac{d\boldsymbol{\omega}}{dt} \times \tilde{\boldsymbol{r}}) dm$
- Apparent moment due to angular velocity relative to I-frame: $\int_m \tilde{\boldsymbol{r}} \times [\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \tilde{\boldsymbol{r}})] dm$

Notice that derivatives with respect to the body frame (local) are denoted as $\frac{\delta ..}{dt}$. By introducing the pseudo external moment vector the EoM mass-varying body obtain the same form as those for a rigid body. This is known as the solidification principle and allows the derivation of the equations of rotational motion in a more straightforward (Euler) form:

$$\dot{\boldsymbol{\omega}} = \mathbf{I}^{-1}(\tilde{\boldsymbol{M}}_{cm} - \boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega}) \tag{3.47}$$

For this research the Mars 2020 class vehicle shall be taken as a reference. The inertia tensor **I** for this vehicle will be outlined Appendix A, with the assumption of having a double plane of mass symmetry. Describing the attitude in the quaternion inertial frame representation requires knowledge of the time derivatives of the attitude in kinematic quaternion form procured from Wie et al. (1989):

$$\dot{\mathbf{q}}(t) = \begin{pmatrix} \dot{q}_0 \\ \dot{q}_{1:3} \end{pmatrix} = \begin{bmatrix} \frac{1}{2}(q_0\boldsymbol{\omega} + \mathbf{q} \times \boldsymbol{\omega}) \\ -\frac{1}{2}\boldsymbol{\omega}^T \mathbf{q} \end{bmatrix} \tag{3.48}$$

$$\begin{pmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{pmatrix} = \frac{1}{2} \begin{bmatrix} q_0 & -q_3 & q_2 & q_1 \\ q_3 & -q_0 & -q_1 & q_2 \\ -q_2 & q_1 & q_0 & q_3 \\ -q_1 & -q_2 & -q_3 & q_0 \end{bmatrix} \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ 0 \end{pmatrix} \tag{3.49}$$

It is denoted that the moment of inertia is assumed to be constant over time. In reality mass fluctuations due depletion would yield otherwise. However, assessing this effect does not form the core of this research. Furthermore, the sum of the pseudo external moments are defined along the body axes. Assuming a double plane of symmetry for the Mars cases, the equations of rotational motion simplify to:

$$\begin{aligned} \dot{p} &= \frac{M_x}{I_{xx}} + \frac{I_{yy} - I_{zz}}{I_{xx}} qr \\ \dot{q} &= \frac{M_y}{I_{yy}} + \frac{I_{zz} - I_{xx}}{I_{yy}} pr \\ \dot{p} &= \frac{M_z}{I_{zz}} + \frac{I_{xx} - I_{yy}}{I_{zz}} pq \end{aligned} \tag{3.50}$$

3.6. TITAN PARAFOIL DYNAMICS

This section outlines the aerodynamic properties of the parafoil and the associated aerodynamic forces and moments. Various dynamic models are simulated with different levels of complexity for terminal descent. The 3DoF point mass simulations is directly linked to the aerodynamic forces. The more complex 6DoF rigid body dynamics model on top of that considers the moments as well as apparent masses and inertias. The forces and moments derived in this section are based on the high-fidelity model depicted in Fig. 3.11. In the remainder of the section the sum of the aerodynamic forces and moments are derived. This section, and with that this chapter, ends with a discussion on wind dispersions related to Titan's environment.

3.6.1. AERODYNAMIC FORCES AND MOMENTS

The control authority for a parafoil in related to controlling the aerodynamic forces and moments. In this research the model was designed such that the symmetric deflection δ_s controls the longitudinal dynamics, while asymmetric deflection δ_a controls the lateral dynamics. These two parameters are a function of the canopy right and left deflection, δ_r and δ_l, respectively. These two parameters are normalized and constrained to vary between zero and one within the guidance algorithm, such that:

$$\delta_s = \frac{1}{2}(\delta_r + \delta_l) \tag{3.51}$$

$$\delta_a = \delta_r - \delta_l \tag{3.52}$$

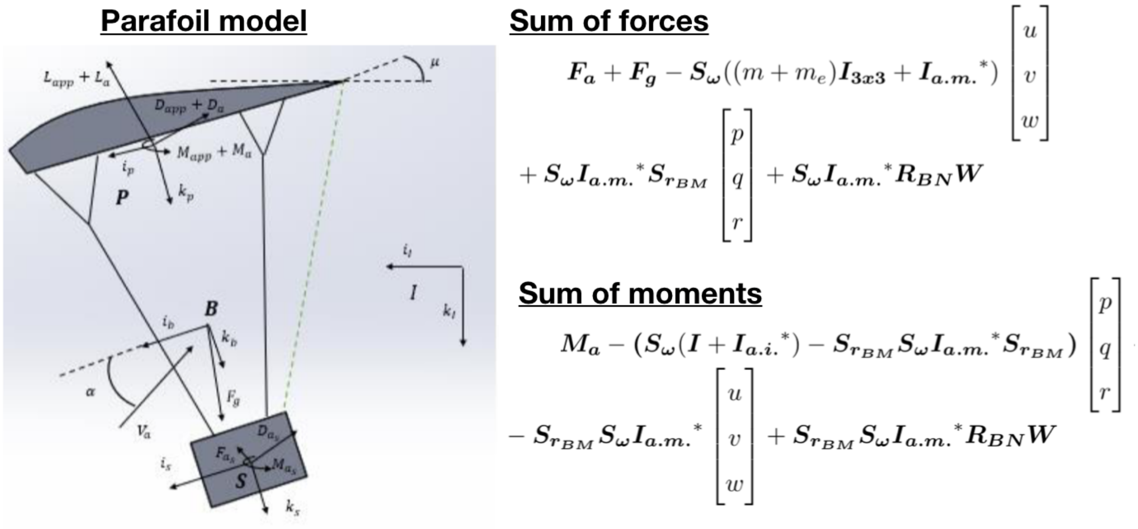
Keeping this in mind, the aerodynamic forces are derived first: lift, drag and side force, respectively:

$$L = C_L \frac{1}{2} \rho V^2 S_{ref} \tag{3.53}$$

$$D = C_D \frac{1}{2} \rho V^2 S_{ref} \tag{3.54}$$

$$S = C_Y \frac{1}{2} \rho V^2 S_{ref} \tag{3.55}$$

In the above equations $\frac{1}{2}\rho V^2$ is defined as the local dynamic pressure and S_{ref} the reference area of either the canopy or the payload. The latter is the case because both the canopy as well as the payload contribute to the total drag value. On top of that, the total drag is eventually increased by another 30% to account for the contribution of the parafoil suspension lines. The lift, drag and side force coefficients for the canopy are computed by means of Eqs. (3.56) - (3.58).



Sum of forces

$$\boldsymbol{F_a} + \boldsymbol{F_g} - \boldsymbol{S_\omega}((m + m_e)\boldsymbol{I_{3x3}} + \boldsymbol{I_{a.m.}}^*) \begin{bmatrix} u \\ v \\ w \end{bmatrix} + \boldsymbol{S_\omega I_{a.m.}}^* \boldsymbol{S_{r_{BM}}} \begin{bmatrix} p \\ q \\ r \end{bmatrix} + \boldsymbol{S_\omega I_{a.m.}}^* \boldsymbol{R_{BN} W}$$

Sum of moments

$$\boldsymbol{M_a} - (\boldsymbol{S_\omega}(\boldsymbol{I} + \boldsymbol{I_{a.i.}}^*) - \boldsymbol{S_{r_{BM}} S_\omega I_{a.m.}}^* \boldsymbol{S_{r_{BM}}}) \begin{bmatrix} p \\ q \\ r \end{bmatrix} - \boldsymbol{S_{r_{BM}} S_\omega I_{a.m.}}^* \begin{bmatrix} u \\ v \\ w \end{bmatrix} + \boldsymbol{S_{r_{BM}} S_\omega I_{a.m.}}^* \boldsymbol{R_{BN} W}$$

Figure 3.11: Side view of the six-degree-of-freedom JPL parafoil model, the sum of forces and the sum moments (Quadrelli et al., 2019).

$$C_L = C_{L_0} + C_{L_\alpha}\alpha + C_{L_{\delta_s}}\delta_s \tag{3.56}$$

$$C_D = C_{D_0} + C_{D_{\alpha^2}}\alpha^2 + C_{D_{\delta_s}}\delta_s \tag{3.57}$$

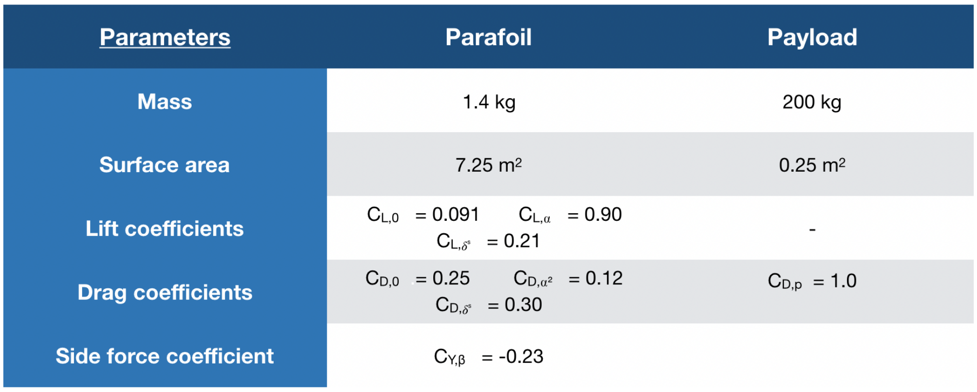
$$C_Y = C_{Y_\beta}\beta \tag{3.58}$$

It becomes clear that the coefficients are a function of the before-mentioned symmetric deflection, angle-of-attack and side-slip angle. The latter two parameters are computed through:

$$\alpha = \frac{w_p}{u_p} \quad , \quad \beta = \frac{v_p}{\sqrt{u_p^2 + w_p^2}} \tag{3.59}$$

The above velocity components are based on the local air velocity, which is also a function of the wind velocities. The discussion regarding this matter is continued in the final section of this chapter. In Table 3.2 the aerodynamic force coefficients and other relevant parameters are presented. Notice that the total drag coefficient of the payload has one contribution, such that $C_D = C_{D_p}$. The reference surface areas are based on the most current mission constraints.

Table 3.2: Aerodynamic properties for modeling the aerodynamic forces, based on Quadrelli et al. (2019)



Parameters	Parafoil	Payload
Mass	1.4 kg	200 kg
Surface area	7.25 m²	0.25 m²
Lift coefficients	$C_{L,0}$ = 0.091 $C_{L,\alpha}$ = 0.90 C_{L,δ^s} = 0.21	-
Drag coefficients	$C_{D,0}$ = 0.25 C_{D,α^2} = 0.12 C_{D,δ^s} = 0.30	$C_{D,p}$ = 1.0
Side force coefficient	$C_{Y,\beta}$ = -0.23	

In a similar fashion, the aerodynamic moments are derived: roll, pitch and yaw, respectively:

$$\mathscr{L} = C_l \frac{1}{2}\rho V^2 S_{ref} b \tag{3.60}$$

$$\mathscr{M} = C_m \frac{1}{2}\rho V^2 S_{ref} c \tag{3.61}$$

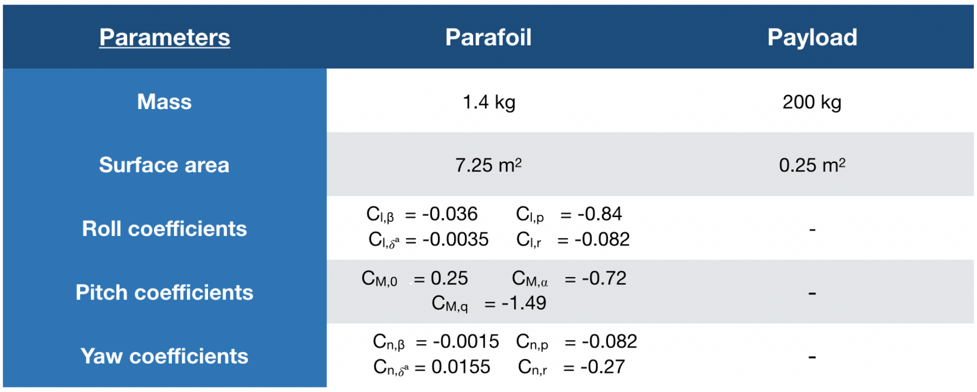
$$\mathscr{N} = C_n \frac{1}{2}\rho V^2 S_{ref} b \tag{3.62}$$

Please note that the relations between the parafoil design parameters are provided in Table 3.3.

Table 3.3: Parafoil design parameters after Quadrelli et al. (2019).

Design Parameter	Equation
Parafoil Wingspan	$b = \sqrt{S \cdot AR}$
Parafoil Chord	$c = \sqrt{S/AR}$
Parafoil Height	$h = 0.14 \cdot c$
Line-length-to-span Ratio	$R/b = 3/4$
Anhedral Angle	$\epsilon = b/2/R$

Table 3.4: Aerodynamic properties for modeling the aerodynamic moments, based on Quadrelli et al. (2019).



Parameters	Parafoil	Payload
Mass	1.4 kg	200 kg
Surface area	7.25 m^2	0.25 m^2
Roll coefficients	$C_{l,\beta}$ = -0.036 $C_{l,p}$ = -0.84 C_{l,δ^a} = -0.0035 $C_{l,r}$ = -0.082	-
Pitch coefficients	$C_{M,0}$ = 0.25 $C_{M,\alpha}$ = -0.72 $C_{M,q}$ = -1.49	-
Yaw coefficients	$C_{n,\beta}$ = -0.0015 $C_{n,p}$ = -0.082 C_{n,δ^a} = 0.0155 $C_{n,r}$ = -0.27	-

The moment coefficients for the canopy are computed as follows

$$C_l = C_{l_\beta}\beta + C_{l_{\delta_a}}\delta_a + C_{l_p}\frac{b}{2V}p + C_{l_r}\frac{b}{2V}r \tag{3.63}$$

$$C_m = C_{m_0} + C_{m_\alpha}\alpha + C_{m_q}\frac{c}{2V}q \tag{3.64}$$

$$C_n = C_{n_\beta}\beta + C_{n_{\delta_a}}\delta_a + C_{n_p}\frac{b}{2V}p + C_{n_r}\frac{b}{2V}r \tag{3.65}$$

In Table 3.4 the aerodynamic moment coefficients and other relevant parameters are presented. The reference surface areas are based on the most current mission constraints. As part of the six-of-degree-model, the moment of inertias need to be computed. For the payload a box is assumed with equal dimensions in all directions, such that the moment of inertias are obtained through

$$I_{x_l} = \frac{m_l}{12}(b_l^2 + c_l^2) \tag{3.66}$$

$$I_{y_l} = \frac{m_l}{12}(a_l^2 + c_l^2) \tag{3.67}$$

$$I_{z_l} = \frac{m_l}{12}(a_l^2 + b_l^2) \tag{3.68}$$

In Eqs. (3.66) - (3.68) a, b and c are the payload dimensions with their values established equaling 0.5 in each direction. The moment of inertias for the parafoil are computed

$$I_{x_p} = \frac{v_{vol}\rho + m_p}{12}(b^2 + h_{mean}^2) \tag{3.69}$$

$$I_{y_q} = \frac{v_{vol}\rho + m_p}{12}(c^2 + h_{mean}^2) \tag{3.70}$$

$$I_{z_q} = \frac{v_{vol}\rho + m_p}{12}(c^2 + b^2) \tag{3.71}$$

The relevant parafoil parameters are summarized in Table 3.5. Notice that v_{vol} is approximated as $0.09bc^2$ and h_{mean} as $\frac{v_{vol}}{bc}$. To project the forces, moments and inertias to the body frame, the rigging angle μ is utilized for the rotational transformation. For these processes the reader is referred to Quadrelli et al. (2019). In the respective paper the reader will also find the exact method of how to derive the apparent forces and moments, of which a brief delineation is also provided in Section 5.2 of this thesis.

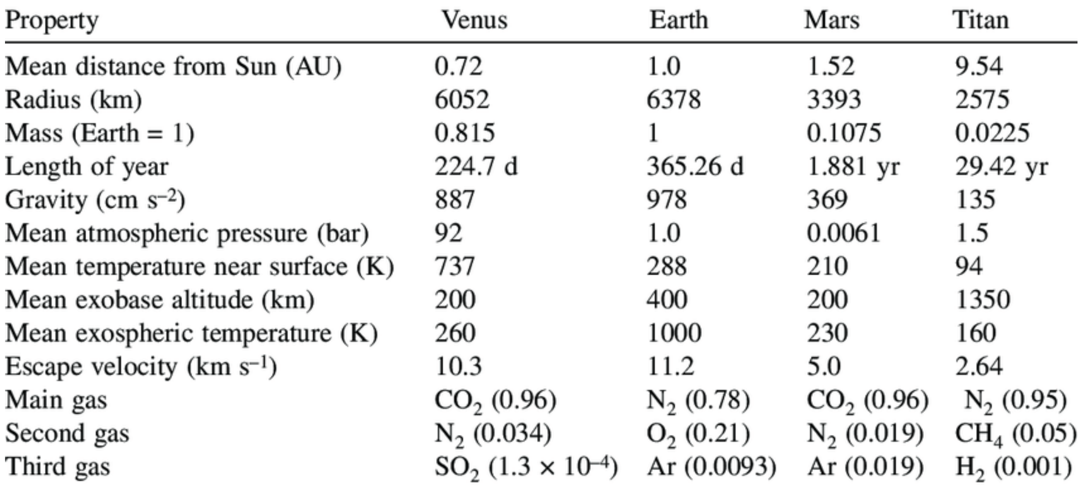
Table 3.5: Canopy dimensions and properties for modeling the inertias, based on Quadrelli et al. (2019). Notice that the current updated canopy area equals 7.25 m^2.

Symbol	Property	Value	Units
AR	Aspect ratio	3.0	-
S	Canopy area	3.14	m^2
t	Canopy thickness	0.075	m
R	Lines length	1.84	m
a	Canopy height	0.164	m
x_c	Center of mass to aerodynamic center distance	0.26	m
σ	Final sigma (contractor's value)	0.45	kg/m^2
z_{rp}	Payload distance	1.5	m
l_{diam}	Lines diameter	0.005	m
l_{ha}	Harness length	-	m
b	Canopy span	-	m
c	Canopy chord	-	m
m_p	Canopy mass	0.002	kg
S_p	Drag surface	-	m^2

3.7. Environmental Properties and Wind Dispersions

The specific environment in which the autonomous GNC systems needs to operate has a major influence on the design approach and the final result. The Martian atmosphere is very thin in nature, consisting of a mixture of 95% CO_2, 3% N_2, CO and H_2. The average surface pressure of 6 mbar equals just 0.6% of the value found on Earth. All these properties cause the Red Planet to have a very distinct climate. Furthermore, Mars is known for its wind gusts which tend to increase with a decrease in altitude. To keep conditions as realistic as possible the choice has been made to model a climate database rather than assume an exponential atmospheric and homogeneous gravity field. The atmospheric properties of Titan are quite different from those on Earth and Mars. Main contributor is the fact that density of the surface is almost five times thicker, while the surface gravity is substantially lower. The exact values are summarized in Table 3.6.

Table 3.6: Atmospheric chemistry of Mars, Venus, and Titan compared to Earth (Krasnopolsky and Lefèvre, 2013).



Property	Venus	Earth	Mars	Titan
Mean distance from Sun (AU)	0.72	1.0	1.52	9.54
Radius (km)	6052	6378	3393	2575
Mass (Earth = 1)	0.815	1	0.1075	0.0225
Length of year	224.7 d	365.26 d	1.881 yr	29.42 yr
Gravity (cm s^{-2})	887	978	369	135
Mean atmospheric pressure (bar)	92	1.0	0.0061	1.5
Mean temperature near surface (K)	737	288	210	94
Mean exobase altitude (km)	200	400	200	1350
Mean exospheric temperature (K)	260	1000	230	160
Escape velocity (km s^{-1})	10.3	11.2	5.0	2.64
Main gas	CO_2 (0.96)	N_2 (0.78)	CO_2 (0.96)	N_2 (0.95)
Second gas	N_2 (0.034)	O_2 (0.21)	N_2 (0.019)	CH_4 (0.05)
Third gas	SO_2 (1.3 × 10^{-4})	Ar (0.0093)	Ar (0.019)	H_2 (0.001)

Perturbations, also defined as dispersions, significantly affect the trajectory of planetary vehicles. The task of the guidance system is to robustly react to these dispersions and get the vehicle back on precision track. Achieving this successfully requires a deep understanding of the phenomena affecting the path in the first place. This asks for a discussion on the influence of wind on the motion of spacecraft travelling through an atmosphere. The reason being that wind can significantly change the trajectory during parachute descent as well as powered descent. A detailed analysis on the influence of wind provided by Mooij (1994) has been taken as a reference base line. Understanding the difference between the ground velocity and speed relative to the atmosphere is key in the discussion. A vehicle flying through a wind gust is altered in local flow conditions tremendously, with the effect only increasing for longer duration bursts. Other wind conditions include steady-state wind, wind shear and turbulence. These phenomena are not considered specifically but will be modelled as (random) guidance errors to study the robustness of the system. The reason for choosing for this method is because the essence lays in the robustness of the GNC system to any arising error. It is therefore of no need to model every disturbance individually and thus the discussion on wind suffices. The governing wind equations are based on Cartesian components to maintain a logic overlap with the previously defined external forces. Consider a wind vector **W** such that:

$$\mathbf{W} = [w_x, w_y, w_z]^T \tag{3.72}$$

The heading and flight-path angle due wind follow from

$$\chi_w = \tan^{-1}\left(\frac{w_y}{w_x}\right) \tag{3.73}$$

$$\gamma_w = \text{asin}^{-1}\left(\frac{w_z}{V_W}\right) \tag{3.74}$$

Notice that $V_W = \sqrt{w_x^2 + w_y^2 + w_z^2}$ denotes the modulus of the wind. The velocity relative to the airspeed $\mathbf{V_A}$ is defined as

$$\mathbf{V_A} = \mathbf{V_G} - \mathbf{W} \tag{3.75}$$

In a spherical coordinate system, the modulus of the airspeed based velocity, heading and flight-path angle are obtained through

$$V_A^2 = V_G^2 + V_W^2 - 2V_G V_W(\cos\gamma_G \cos\gamma_W \cos(\chi_W - \chi_G) + \sin\gamma_G \sin\gamma_W) \tag{3.76}$$

$$\chi_A = \tan^{-1}\left(\frac{\sin\chi_G \cos\gamma_G V_G - \sin\chi_W \cos\gamma_W V_W}{\cos\chi_G \cos\gamma_G V_G - \cos\chi_W \cos\gamma_W V_W}\right) \tag{3.77}$$

$$\gamma_A = \tan^{-1}\left(\frac{\sin\gamma_G V_G - \sin\gamma_W V_W}{V_A}\right) \tag{3.78}$$

And an actual descent rate of

$$\mathbf{V_D} = V_v + w_z \tag{3.79}$$

In which V_v is the airspeed verctor, from which the flight-path angle is derived:

$$\gamma = -\sin^{-1}(\frac{V_d}{V}) \tag{3.80}$$

Regarding the attitude of the vehicle relative to the airspeed, a transformation is needed of the velocity components to the body frame, such that

$$\mathbf{V_B^{BW}} = (u_B, v_B, w_B)^T = \mathbf{C_{B,R}} \mathbf{V_R^{BW}} \tag{3.81}$$

The components in the body frame need to be transformed to the guidance frame, which is inertial in nature in the case of terminal descent. This is entrenched through

$$\mathbf{V_I^{IB}} = (u_I, v_I, w_I)^T = \mathbf{C_{I,B}} \mathbf{V_B^{IB}} \tag{3.82}$$

After obtaining the wind parameters, the airspeed based velocity, angle of attack and sideslip angle are computed. Given the aerodynamic coefficients, the forces are computed. A transformation is applied to obtain the groundspeed based aerodynamic force by utilizing the airspeed based flight-path angle and heading. Since the guidance problem is defined in an inertial frame, with the origin coinciding with the landing site. All forces are defined in the same frame, after which the equations of motion are evaluated. This means that the forces due wind need to be transformed to the inertial frame, which is entrenched using Table 3.1. After updating the state vector within the convex optimization framework, these steps are continuously repeated in a discrete fashion for each temporal node until the sequence is terminated.

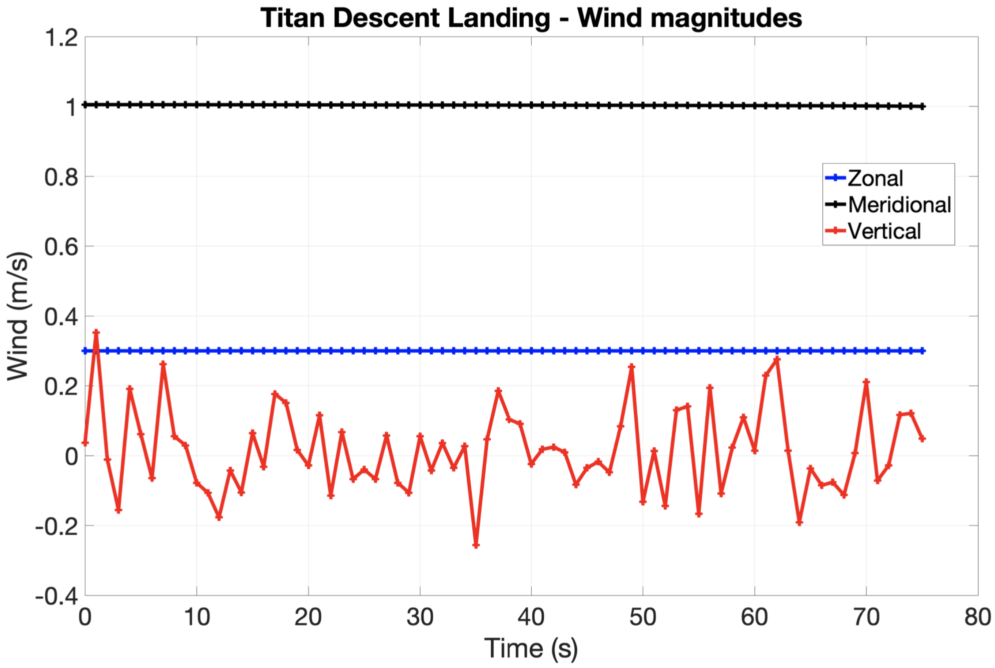


Figure 3.12: Simulated zonal, meridional and vertical wind, magnitudes based on Lorenz et al. (2012).

From the Cassini-Huygens mission it became clear that the thick atmosphere of Titan is also (relatively) strong winds which have a large effect on the parafoil and hence on the landing sequence. The wind model utilized in research is based on Lorenz et al. (2012), as that is one of the few sources for Titan wind properties. In the paper a distinction is made between three wind sources at terminal descent: zonal, meridional and vertical winds. Before computing the magnitudes of these winds, the wind dynamics is expressed. Starting with the latter, data from the vertical winds measured by the Huygens probe dictate a zero mean value with a standard deviation of 0.1 m/s at low altitudes. This model is adapted in the research based on a Gaussian distribution. The zonal and meridional at the altitude of interest were established at a constant value of 0.3 m/s and 1.0 m/s, respectively. Based here-upon the wind magnitudes for a typical parafoil descent sequence has been simulated, the result of which is presented in Fig. 3.12.

4

Guidance, Navigation and Control

The main objective of the Guidance, Navigation, and Control (GNC) system during the descent phase of a space mission is to ensure the reduction of the spacecraft's velocity at the start of the sequence to near zero at the terminal phase. For interplanetary missions an atmospheric entry is involved which slows down the spacecraft from hypersonic to supersonic velocities, after which conventionally a parachute mechanism is deployed. During the atmospheric entry, uncertainties in atmospheric properties introduce navigation dispersions with respect to the target and control authority is rather limited. Eventually this leads to the accumulation of errors in both spacecraft position and velocity. The task of the GNC system, together with the integrated autonomous optimization algorithms, is to correct for these errors in real-time. The robustness of these systems to a large extend determine the order of precision achievable. While performing the manoeuvre, the GNC must ensure the tactical control over landing time and location, while satisfying the constraints of the trajectory in terms of state and control. The conventional design strategy along with an overview of the GNC system is outlined in Section 4.1.

The guidance system, forming the core focus of the research, is charged with the task to bring forth steering commands by taking an estimated reference of the prevailing and final state into account. Deviations relative to a certain reference are calculated based on (theoretical) models and from this corrective actions are determined. Section 4.2 elaborates on the autonomous descent guidance system, distinguishing between 3DoF and 6DoF simulation environments. The input state data is provided by the navigation system, using the interface information gathered by yet again (theoretical) models and sensors. Use of multiple sensors gives rise to the need for filter algorithms to combine and best estimate the data. The matters on navigation are regarded as such that perfect state is assumed. The reason for this assumption is elaborated on in Section 4.3. For closed-loop analysis and theoretical utilization of the propulsion and power systems for guidance purposes, the control aspect of the GNC system is highlighted in the final section of this chapter.

4.1. Design Strategy and System Overview

Comprehending the general design strategy of the GNC system is crucial for understanding what elements need to be optimized to achieve precision landing. Several matters are to be taken into account when considering landing on Mars and Titan, which in essence are completely different planetary bodies. Mars has a very thin atmosphere, while Titan is the only natural satellite in the Solar System with an atmosphere having typically 50% higher surface pressures relative to Earth-based values. Entry and descent uncertainties for Martian and Titanian missions are tremendous due to the fact that the governing EDL properties of these planetary bodies are less explored in comparison to Earth. Likewise, the lack of a robust navigation system such as the Global Positioning System (GPS) causes dispersion propagation, consequently leading to a rather large landing ellipse. Conventionally, a nominal design is considered with no sensitivity for any uncertainties and disturbances or a predictor-corrector scheme is incorporated (Jiang et al., 2014). Within this distinction, the nominal trajectory design, as further expanded on in Section 4.1.1, is used for first-order analysis and gaining better understanding of the working principle of the system. An error integrated design is a type of scheme in which the various disturbance sources are included, on which Section 4.1.2 expands further. Additionally, the problem of trajectory optimization is solved by the introduction of intelligent algorithms capable of global optimization. Chapter 5 delineates the mathematical principles of the convex optimization schemes.

4.1.1. NOMINAL TRAJECTORY DESIGN

For any given mission and space vehicle, an optimal trajectory is calculated to be used as a reference. This so-called nominal trajectory is free from disturbances, considers the planetary environment to a certain complexity level and considers the vehicle properties to a limited accuracy. This trajectory is computed taking the following constraints into account:

- Initial conditions
- Steering (minimum and maximum range for guidance)
- Path (mechanical and thermal loads)
- Terminal location

Designing the GNC system is centered around this nominal trajectory and the more the system can react to uncertainties encountered during EDL, the more robust it is. Nonetheless, one should keep in mind that a too robust system has a negative influence on the maneuverability, while a system that lacks robustness may encounter control problems. The design process is for that reason often a trade-off between the two. It is important to study how the GNC system reacts to uncertainties within the design cycle. For testing the performance in terms of robustness, both 3DoF and 6DoF simulation analyses are applied. Conventionally one simulates various test scenarios in which models of the dynamics, environment and spacecraft are included. The elements of the GNC system are schematically depicted in Fig. 4.1. Notice that this overview is a simplified version, a detailed form with the inclusion of numerous sources of error is presented at a later stage.

The blocks presented in the figure are connected by lines that indicate the flow of data. Analogous with the operations of aircraft, the control of spacecraft entry and descent is branched in three elements: guidance, navigation and control. These elements form the core of this chapter, that by itself forms a core of the research. Before discussing each element extensively in the sections ahead, a description of the task of each block of Fig. 4.1 is provided in the subsequent summation.

- **Mission manager:** contains a nominal reference trajectory which is uploaded to the spacecraft computer pre-flight amongst other on-board models. Based on the estimated state the difference relative to the stored trajectory is computed, which serves as input for the guidance system.
- **Guidance logic:** has the task to guide the CoM of the spacecraft in its motion, e.g., by the initiation of a bank manoeuvre to change lift orientation for guidance. The guidance logic obtains information on the difference between the commanded and measured state, which is defined as the guidance error. The output of the guidance logic forms one of the inputs for the control algorithm.
- **Control algorithm:** this algorithm determines the motion about the CoM after receiving input from the guidance logical and the state estimator. Likewise, the discrepancy between these input channels is defined as the control error. Commands for the actuators are determined based on a predefined concept of control → e.g., PID or LQR control.

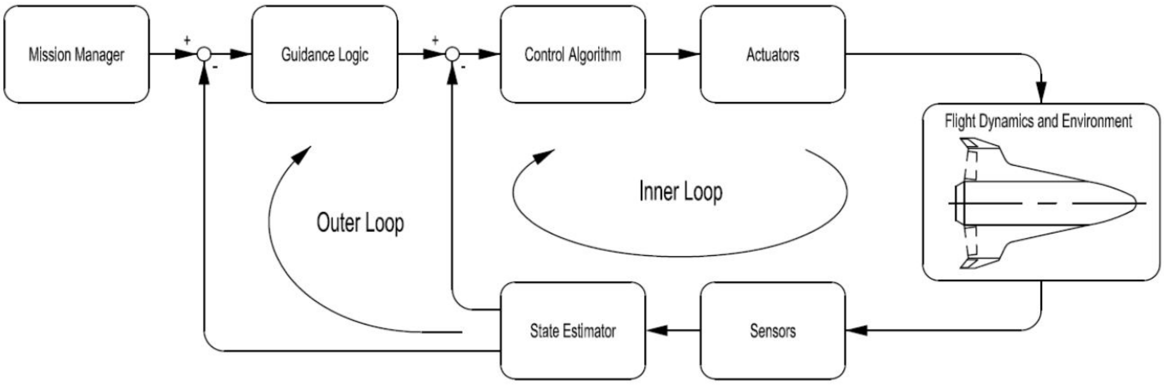


Figure 4.1: Schematic overview of the guidance navigation and control (GNC) system (simplified) (Mooij, 2017).

- **Actuators:** the actuators convert the control commands into actuation of forces and/or moments by means of hardware → e.g., control surfaces (lift/drag), retrorockets or a parafoil.
- **Flight dynamics and environment:** the actual spacecraft in the (Mars/Titan) environment behaving in accordance with the law of physics (dynamics). The state of the spacecraft is produced → e.g., position, velocity, angular rate, rate of heat and mechanical load.
- **Sensors and state estimators:** measure and derive the the state of the system in real-time, respectively. Common sensors used for re-entry and descent include accelerometers, altimeters, gyroscopes, LIDAR, pressure sensors, radar and recently TRN.

As summarized by Mooij (2017), the sensors measure the state, the actuators change it, and together they form the interface with the vehicle. The state estimator, sensors and partial functionality of the mission manager form the navigation system. The two loops, inner and outer, depicted in the figure are defined as the control and guidance loop, respectively. These loops are generally used to determine the (closed-loop) stability of the entire GNC system. The main goal of the navigation filters is to provide best-estimates for the relevant state variables. The relevance of the respective sensors is strongly dependent on the origin of the motion, i.e., entry or powered descent variables. While this research assumes perfect state knowledge, the section on spacecraft navigation (Section 4.3) expands more on this matter for justification purposes.

4.1.2. Error Integrated Design

At various whereabouts in the nominal system that has been presented in the preceding section, errors might get introduced. Typical examples of such errors are the ones induced by imperfections in sensors measuring the state of the spacecraft and deficiencies in the actuators controlling the vehicle. This by all means includes the absence of GPS outside of Earth missions. State estimators, as part of the feedback navigation system discussed in Section 4.3, are commonly used to obtain best estimates for these errors. Unfortunately, state estimators solely are not sufficient for precision landing, especially in the case the system can find no feasible trajectory to the designated target. In a similar fashion, externally induced disturbances due for example wind, which may be treated as errors, have a significant effect on the design trajectory. For a Titan landing sequence, the winds mainly cause tremendous dispersions in the upper atmosphere, as wind speeds are significant relative to the velocity of the lander. In either case, whether internal or external induced errors arise, convex optimization has proven to be a successful solution for dispersion management (Blackmore et al., 2010). The governing principles for this optimization technique are introduced in the preceding section on guidance and further discussed in Chapter 5.

4.1.3. Mission Operations

The mission operations, which for Martian and Titan precision landing should be fully autonomous, assure the rightful operations of the spacecraft. Earth-Mars and Earth-Titan communication links are carried by either electromagnetic waves, radio waves or light. These carriers travel at the speed of light ($c \approx 3 \cdot 10^8$ m/s) while the distance between the Earth and planetary bodies varies over time. It therefore takes several minutes for a carrier signal to reach either Mars or Titan while the spacecraft is performing the EDL sequence, making control based on the Deep Space Network (DSN) impractical (Yu et al., 2017). The Mars landing sequence, dubbed by NASA as the seven minutes of terror, must therefore be performed real-time and fully autonomously. Despite the fact that landing on Titan takes much longer (nearly 2.5 hours), a similar conclusion is entrenched when tilting the discussion to the Earth-Titan system. While the discussion of autonomy is continued in the next chapter, it is established that the mission operations prerequisite:

- measuring of state (position and velocity)
- computing deviations with respect to the nominal trajectory
- resolving corrective guidance orders
- transfer guidance orders to the control system
- converting control outputs to actuator operations

Based on the narrative on the previous page, the remainder of the chapter is structured. In the following section the core of the research is delineated first: the guidance systems and laws. A distinction is made between the various phases of EDL, giving top priority to powered descent guidance. Several navigation and EDL sensor options for determining position, velocity, angular rate and attitude will be discussed in Section 4.3. For closed loop analyses a control system is required and therefore, based on the guidance and navigation schemes, algorithms and control operations are described in Section 4.4.

4.2. Guidance

Analogous to the design strategy for GNC systems, a classical way of classifying guidance methodologies is distinguishing between predicting capabilities and nominal trajectories (Wingrove, 1963). The latter is a stored onboard path reference while the former encompasses methods for predicting possible future paths. Trajectory-independent guidance methods also exits, of which the numerical predictor-corrector guidance is a prime example (Yu et al., 2017). Inherently, the predictive guidance methods are capable of overcoming errors induced through linearization of the governing equations and simplifying assumptions. However, the (powered) descent guidance problem is defined as finding a fuel or position optimal trajectory given a initial and desired final state, also known as the soft landing problem in optimal control literature (Açıkmeşe and Ploen, 2007). This trajectory is bounded by thrust magnitude constraints in case of powered descent and aerodynamic force (maximum lift and drag) constraints in case of parachute descent. Various approaches have been taken to find closed-form solutions for the optimal thrust profile in three-dimensional Euclidean space, all resulting in numerical solutions eventually. These numerical methods are eminently alluring as conditions related to nonconvex guidance, i.e., adjoint and transversality conditions, and the maximum principle, do not need to be enforced. As stated by Açıkmeşe and Ploen (2007), these methods are converted from an infinite-dimensional optimal control problem to a finite dimensional parameter optimization problem, to be subsequently solved using robust nonlinear programming. Yet, for real-time onboard applications, explicit understanding of deterministic convergence characteristics are required, which is generally not the case for a nonlinear algorithm. For reasons defined in Section 4.1.3, Mars and Titan pinpoint landing require real-time optimal trajectory computations which makes it essential to exploit algorithm design methodologies that guarantee convergence to a global optimum. This segment of the chapter exploits a guidance method with deterministic convergence criteria: the convex optimization framework.

A very powerful tool capable of generating optimal paths in real-time based on a given objective function, the dynamics, and the control and state constraints, is the convex optimization method, which is adopted in this research for descent guidance purposes. This section outlines both the 3DoF as well as the 6DoF non-convex optimization guidance fundamentals, which inherently belong to the class of optimal control problems. In Section 4.2.1 the minimum fuel nonconvex optimal control problem is outlined. Following up, in Section 4.2.2, the formulation is extended as a minimum landing error problem, guaranteeing safe touchdown for all mission scenarios. Section 4.2.3 expands on desired additional constraints based upon mission operations. A validated proof of concept, by the name Guidance for Fuel Optimal Large Divert (G-FOLD), is discussed subsequently. Based on the prior established formulations, the problem is firstly extended by including aerodynamic controls (Section 4.2.5) and subsequently extended to a 6DoF formulation using a quaternion attitude representation (Section 4.2.6) . Chapter 5 focuses on the mathematical principles of lossless and successive convexification that allow these algorithms to be adapted onboard in real time. The same chapter confers an outline of the golden search algorithm for finding the optimal time-of-flight.

4.2.1. Nonconvex Optimal Control

During the first two main EDL phases (i.e., entry and parachute descent), commonly spacecraft accumulate large errors in position and velocity (Açıkmeşe et al., 2012). This can be attributed to the fact that atmospheric uncertainties emanate and control authority is rather limited (especially during the parachute phase). Conventional algorithms for powered descent have mainly been inherited from Apollo technologies. These algorithms do not directly take mission constraints or fuel optimization into account, leading to incomplete utilization of vehicle capabilities. This by itself makes precision landing practically impossible and thus a solution has to be found to this problem. The convex programming approach to powered descent guidance for Mars landing seems to be the solution to this matter. This research also exploits the adaptability of the same approach to precision landing on Titan by means of a parafoil. This is further expanded on in Section 4.2.5. The main algorithm is designed such that optimal (state and control) constraints satisfying paths are generated that are physically feasible. This section outlines the general nonconvex optimal control powered descent problem for Mars soft landing. The constitutional non-convexity commences in the thrust domain as it is not possible nor reliable to turn thrusters off once the descent maneuver is initiated.

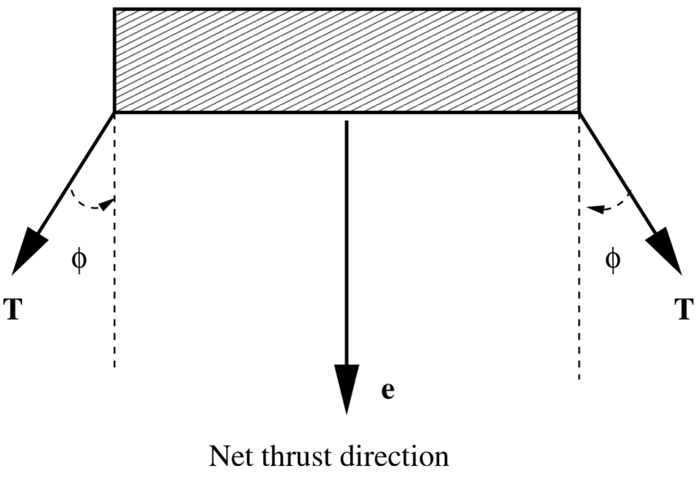


Figure 4.2: Net thrust spacecraft geometry for the powered descent problem (Açıkmeşe and Ploen, 2007).

The spacecraft motion is modelled as a point-mass with a (nonzero) lower bound magnitude in the thrust domain which defines a nonconvex region in the control space. As the descent stage of the sequence occurs at a relatively low attitude (less than 5 km for both Mars as well as Titan landing), a uniform gravitational field is assumed. In the original formulation aerodynamic forces are neglected (Section 4.2.5 goes into more detail regarding this matter). Likewise, the original definition includes translational guidance solely. The instantaneous direction of thrust $\mathbf{e} \in \mathbb{R}^3$ and the cant angle ϕ are depicted in Figure 4.2. The translational dynamics are defined in the surface fixed reference frame that inherently encompasses the planet's angular rate of rotation (Açıkmeşe et al., 2012), such that

$$\ddot{\mathbf{r}}(t) = \mathbf{g} + \frac{\mathbf{T}_c(t)}{m(t)} \tag{4.1}$$

where $\mathbf{r} \in \mathbb{R}^3$ is the target relative position vector, $\mathbf{g} \in \mathbb{R}^3$ the acceleration due planetary gravity, $\mathbf{T_c} \in \mathbb{R}^3$ the net thrust vector and m the spacecraft mass. In modeling the thrust vector during powered descent it is assumed that the rocket engines are in a gimbal locked orientation. The rocket mass depletion function is given by

$$\dot{m}(t) = -\alpha \, \|\mathbf{T}_c(t)\| \tag{4.2}$$

A total of n identical thrusters are considered, i.e., equal levels of thrust $\mathbf{T}$ and specific impulse I_{sp} at each instance of time. Positive α is defined as the constant fuel consumption rate, such that

$$T_c = nT\cos\phi\mathbf{e} \tag{4.3}$$

$$\alpha = \frac{1}{I_{sp}g_0\cos\phi} \tag{4.4}$$

During the entire descent sequence the thrust levels for each rocket engine are bounded

$$0 < T_1 \leq T(t) \leq T_2 \tag{4.5}$$

This implies that given a final time t_f, for $\forall t \in [0, t_f]$

$$\rho_1 \leq \|\mathbf{T}_c\| \leq \rho_2 \tag{4.6}$$

Given is that $\rho_2 > \rho_1 > 0$ and

$$\rho_1 = nT_1\cos\phi \quad , \quad \rho_2 = nT_2\cos\phi \tag{4.7}$$

Notice that Eq. (4.6) defines the nonconvex control space region. Initial and final boundary conditions are established such that

$$m(0) = m_0, \quad \mathbf{r}(0) = \mathbf{r}_0, \quad \dot{\mathbf{r}}(0) = \dot{\mathbf{r}}_0 \tag{4.8}$$

$$\mathbf{r}(t_f) = \mathbf{0}, \quad \dot{\mathbf{r}}(t_f) = \mathbf{0} \tag{4.9}$$

where $\mathbf{r}_0$ and $\dot{\mathbf{r}}_0$ are the initial position and velocity vectors, respectively. Moreover, m_0 the initial mass of the spacecraft, i.e., the wet mass. In the remainder of the nonconvex formulation it is assumed that the initial position along the gravity direction is positive. During the descent maneuver it is required that the trajectory remains above the surface. Defining $\mathbf{r}(t) = [r_1(t), r_2(t), r_3(t)]^T$, this sub-surface constraint is enforced as

$$r_1 \geq 0, \quad \forall t \in [0, t_f] \tag{4.10}$$

Additional state and control constraints might be imposed on the motion depending on mission requirement. Section 4.2.3 elaborates on the respective supplementary constraints needed for Mars and Titan landing. In this section the general form of enforcing state constraints is conferred. For this a position and velocity state vector $\mathbf{x} \in \mathbb{R}^6$ is defined such that

$$\mathbf{x} = \begin{bmatrix} \mathbf{r} \\ \dot{\mathbf{r}} \end{bmatrix} \tag{4.11}$$

Defining $S_j \in \mathbb{R}^{n_i \times 6}$, $c_j \in \mathbb{R}^6$, $a_j \in \mathbb{R}$ and $n_j \leq 6$, the state constraints have the ensuing general form

$$\left\| S_j x(t) - v_j \right\| + c_j^T x(t) + a_j \leq 0, \quad j = 1, \ldots, n_s, \quad \forall t \in [0, t_f] \tag{4.12}$$

Given the above described nonconvex trajectory optimization problem with the given control and state constraints, as has been delineated through Eqs. (4.1) - (4.12), the minimum fuel guidance problem for powered descent landing can be described:

Problem 1: nonconvex minimum fuel guidance problem

Cost Function:

$$\max_{t_f, \mathbf{T}_c}(t_f) = \min_{t_f, \mathbf{T}_c} \int_0^{t_f} \| T_c(t) \| \, \mathrm{d}t \qquad \text{subject to:}$$

Dynamics:

$$\ddot{\mathbf{r}}(t) = \mathbf{g} + \frac{\mathbf{T}_c(t)}{m(t)}$$
$$\dot{m}(t) = -\alpha \, \|\mathbf{T}_c(t)\|$$

Control and State Constraints:

$$0 < \rho_1 \leq \|\mathbf{T}_c\| \leq \rho_2$$
$$r_1 \geq 0$$
$$\left\| S_j x(t) - v_j \right\| + c_j^T x(t) + a_j \leq 0, \quad j = 1, \ldots, n_s$$

Boundary Conditions:

$$m(0) = m_0, \quad \mathbf{r}(0) = \mathbf{r}_0, \quad \dot{\mathbf{r}}(0) = \dot{\mathbf{r}}_0$$
$$\mathbf{r}(t_f) = \mathbf{0}, \quad \dot{\mathbf{r}}(t_f) = \mathbf{0}$$

The above established problem contains a non-convexity in the control domain due to the thrust magnitude inequality. To make the algorithm compatible for real-time applications while maintaining the optimal control result, the problem needs to be put through a process known as lossless convexification of the control magnitude constraint. The mathematical formulation on this matter is continued in Chapter 5. The next section outlines the extension of the above optimal control problem to achieve minimum landing error.

4.2.2. MINIMUM LANDING ERROR

The previous section defined the nonconvex optimal control problem framework, which finds a thrust profile, taking the lander from an initial state to a designated target position while minimizing the consumption of fuel. This section outlines the extension of the problem to the position domain. The extension of the convex optimization approach in this portion of the thesis includes an enhancement ensuring the algorithm can handle cases in which a feasible trajectory to the destination does not exist. The latter is needed for missions that do not have the capacity to carry sufficient fuel to ensure the target is reached for all mission scenarios, hence the backup acts as a fail-safe. The onboard guidance system must ensure a safe landing as close to the target as possible in case such a situation arises (Blackmore et al., 2010). In essence, the nature of the minimum landing error problem, i.e., non-convexity, is similar to Problem 1. However, there are some key differences between the two problems, which are examined in this section. Ahead of examining these differences, the minimum landing error guidance problem for powered descent is described:

Problem 2: nonconvex minimum landing error guidance problem

Cost Function:

$$\min_{t_f, \mathbf{T}_c} \left\| \mathbf{r}(t_f) \right\| \qquad \text{subject to:}$$

Dynamics:

$$\ddot{\mathbf{r}}(t) = \mathbf{g} + \frac{\mathbf{T}_c(t)}{m(t)}$$
$$\dot{m}(t) = -\alpha \left\| \mathbf{T}_c(t) \right\|$$

Control and State Constraints:

$$0 < \rho_1 \leq \|\mathbf{T}_c\| \leq \rho_2$$
$$r_1 \geq 0$$
$$\left\| S_j x(t) - v_j \right\| + c_j^T x(t) + a_j \leq 0, \quad j = 1, \ldots, n_s$$

Boundary Conditions:

$$m(0) = m_0, \qquad \mathbf{r}(0) = \mathbf{r}_0, \quad \dot{\mathbf{r}}(0) = \dot{\mathbf{r}}_0$$
$$m(t_f) \geq m_{\text{dry}}, \quad \mathbf{r}(t_f)^T \mathbf{e}_1 = 0, \quad \dot{\mathbf{r}}(t_f) = \mathbf{0}$$

The cost function defined in Problem 2 differs from the one in Problem 1 as it minimizes the squared distance in Euclidean space rather than final mass (i.e., fuel consumption). Notice that minimizing the distance squared is equivalent to minimizing distance due to the positive norm and monotonic guarantee. Throughout this thesis, $\|\mathbf{x}\|$ defines the norm of vector $\mathbf{x}$. For simplicity purposes, without loss of generality, the target is assumed at zero. Moreover, the surface normal is defined as $\mathbf{e}_1$. In accordance with the outline of the minimum landing error problem $\mathbf{d} \in \mathbb{R}^2$ is established as the surface landing target. Based on the approach of the renewed algorithm, some boundary conditions are altered for the minimum fuel guidance problem:

Modified boundary conditions for Problem 1

Boundary Conditions:

$$m(0) = m_0, \qquad \mathbf{r}(0) = \mathbf{r}_0, \quad \dot{\mathbf{r}}(0) = \dot{\mathbf{r}}_0$$
$$m(t_f) \geq m_{\text{dry}}, \quad \mathbf{e}_1^T \mathbf{r}(t_f) = 0, \quad [\mathbf{e}_2 \quad \mathbf{e}_3]^T \mathbf{r}(t_f) = \mathbf{d}, \quad \dot{\mathbf{r}}(t_f) = \mathbf{0}$$

The key notion of the minimum landing error algorithm is to first perform an optimization that minimizes the distance between the designated target and the reachable landing location; in other words: first solve Problem 2. This returns a specific computed range D such that

$$D = \left\| \mathbf{r}(t_f) \right\| \quad \text{for} \quad \{t_f^*, T_c^*(\cdot)\} \tag{4.13}$$

Utilizing this optimal range, the minimum fuel optimization trajectory is computed as defined in Problem 1, keeping in mind the modified boundary conditions for the renewed problem definition. Notice that

$[\mathbf{e}_2 \quad \mathbf{e}_3]^T \mathbf{r}(t_f) = \mathbf{d}$ now gets defined as $\left\|\mathbf{r}(t_f)\right\| \leq D$ to ensure the algorithm returns the global optimal solution, satisfying the nonconvex thrust constraints. For real-time onboard guidance, the above described algorithm requires a relaxation of the problem through lossless convexification, of which the lemma is detailed in the next chapter. In the next part of the thesis additional state and control constraints are characterized found on mission needs. Summarizing, the prioritized powered descent guidance algorithm for minimum landing error is entrenched as:

Prioritized powered-descent guidance algorithm (after Blackmore et al. (2010))

1. Solve the relaxed minimum-landing-error guidance problem (Problem 2) for $\{\mathrm{t}_f^*, \mathrm{T}_c^*(\cdot)\}$ with corresponding trajectory $\mathbf{r}_c^*(\cdot)$. If no solution exists, return *infeasible*.
2. Solve the relaxed minimum fuel guidance problem to specified range (Problem 1) with $D = \left\|\mathbf{r}(t_f)\right\| \quad \text{for} \quad \{t_f^\dagger, T_c^\dagger(\cdot)\}$.
3. Return $\{t_f^\dagger, T_c^\dagger(\cdot)\}$.

4.2.3. STATE AND CONTROL CONSTRAINTS

The soft landing guidance problem is characterized as a finite horizon optimal control problem. That is, the optimal performance measures for a finite time $t = [0, t_f]$ within system control. Pointedly, the system control is the optimal thrust profile accompanied by a certain Euclidean translational motion $(\mathbf{r}, \dot{\mathbf{r}})$ to minimize either fuel consumption or landing error. The general form for enforcing specific state constraints has been conferred in Section 4.2.1, Eq. (4.12) specifically. This section explicitly delineates state and control constraints for specific mission needs. The complementary state constraints include a bound on the **velocity** and **glide-slope angle**. These constrains are convex in nature and hence fit properly into the framework. The supplementary control constraint, defined as the **thrust point constraint**, has both convex and nonconvex properties, which are examined in this section. This section also reviews how the latter two constrains (glide-slope angle and thrust pointing) are especially important to cope with the requirements of Terrain Relative Navigation (TRN) and Hazard Detection & Avoidance (HDA).

VELOCITY CONSTRAINT

A rather straightforward form of state constraint can be imposed on the velocity magnitude, which might be needed to prevent stall or to keep aerodynamic forces below detrimental values. Consider a certain minimum and maximum velocity magnitude, V_{min} and V_{max}, respectively, then

$$V_{\text{min}} \leq \|\dot{\mathbf{r}}(t)\| \leq V_{\text{max}} \tag{4.14}$$

Imposing a minimum velocity magnitude constraint is especially relevant during parachute or parafoil descent to prevent stall. Imposing the maximum velocity magnitude constraint is mainly motivated from a structural engineering point of view in order to keep mechanical loads below their ceiling threshold. Additionally, the upper velocity bound averts reaching supersonic velocities on planetary bodies with an atmosphere (which Mars and Titan both are), as within such conditions control thrusters may become unreliable.

GLIDE-SLOPE CONSTRAINT

The glide-slope constraint enforces that a specific optimal trajectory to the target is not too shallow nor goes below the surface. Fig. 4.3 depicts the minimum slope angle γ_{gs} that imposes a cone shape on the spacecraft. The glide slope constraint requires the spacecraft to remain within a cone as defined by the minimum slope angle. Notice that for the case of minimum landing error, the apex of the cone concurs with the landing target position, a certain distance $\mathbf{q}$ from the original target. To cope with the requirement of, e.g., terrain-relative navigation, the spacecraft orientation in terms of attitude is restrained. By enforcing the cone shape restricted motion, the lander at higher altitudes for an elongated period, hence surface characteristics can be observed for a certain extended period of time. The feasible positions of the spacecraft glide-slope constraint are defined using a set $\mathbf{X}$ such that

$$\mathbf{X} = \{\mathbf{r} \in \mathbb{R}^3 : \left\|\mathbf{S}(\mathbf{r} - \mathbf{r}(t_f))\right\| - \mathbf{c}^T(\mathbf{r} - \mathbf{r}(t_f)) \leq 0\} \tag{4.15}$$

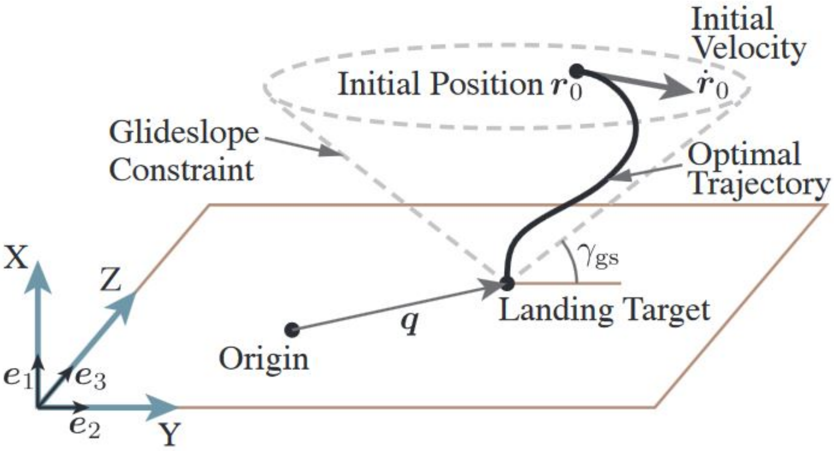


Figure 4.3: Descent guidance glide slope constraint requiring the spacecraft to remain within a cone as defined by a minimum slope angle γ_{gs}, ensuring the spacecraft stays safely from the terrain prior to reaching the target (Carson et al., 2011).

For feasibility purposes **S** and **c** are defined to designate a cone with a vertex at $\mathbf{r}(t_f)$, leading to

$$S \triangleq \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.16}$$

$$c \triangleq \frac{\mathbf{e}_1}{\tan\gamma_{gs}}\gamma_{gs} > 0 \tag{4.17}$$

It is again emphasized that **X** is a convex set within the guidance formulation. The standard definition of the interior (i.e., a subset of the topological space) of **X** is conferred for completeness

$$\text{int}\,\mathbf{X} \triangleq \{x \in \mathbf{X} : \exists\, \epsilon > 0 \quad \text{such that} \quad y \in \mathbf{X} \quad \text{if} \|x - y\| < \epsilon\} \tag{4.18}$$

$$\partial\mathbf{X} \triangleq \{x \in \mathbf{X} : x \not\in \ \text{int}\} \tag{4.19}$$

Notice that the latter equation denotes the boundary of **X**. In the chapter on convex optimization the methodology for extending this constraint to the use of multiple obstacle avoidance is outlined.

THRUST POINTING CONSTRAINT

Generally, sensors for TRN onboard of spacecraft require rather explicit viewing orientations, which imposes the pointing constraint on the attitude orientation of the vehicle. Within the powered descent guidance formulation, the vehicle is modelled as point mass having the thrust vector point alongside the desired direction of force. This framework allows the introduction of the orientation constraint by merely imposing a restriction on the thrust pointing vector. Explicitly implementing attitude dynamics within the formulation would increase the guidance formulation complexity significantly, a discussion left for Section 4.2.6.

The trust pointing constraint belongs to the class of control constraints, whom in essence are more challenging as the sets of feasible controls are nonconvex by definition. Given a definite time-of-flight t_f, a total of three control constraints are defined:

1. Convex upper bound on thrust: $\|\mathbf{T}_c\| \leq \rho_2$
2. Nonconvex lower bound on thrust: $\|\mathbf{T}_c(t)\| \geq \rho_1 > 0$
3. Pointing constraint on thrust: $\hat{\mathbf{n}}^T\mathbf{T}_c(t)/\|\mathbf{T}_c(t)\| \geq \cos\theta$

The first two constrains have already been conferred in Section 4.2.1. Chapter 5 continues on the subject by defining how the non-convexity of the lower thrust bound can be handled without loss of optimality. The remainder of this section focuses on the third control constraint regarding the thrust pointing. The direction vector $\|\hat{\mathbf{n}}\| = 1$, and the angle θ describes the angle of deviation with respect to the normal direction such that

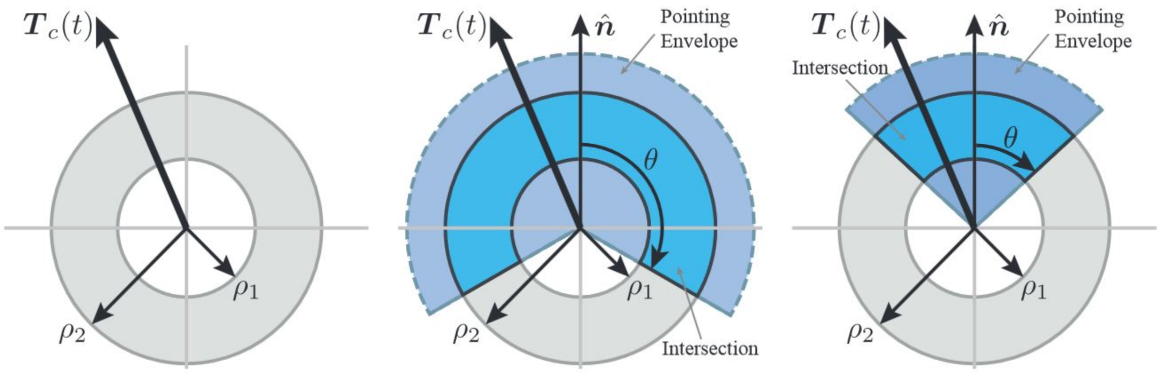


Figure 4.4: Planar view of the thrust bounds (left) and the respective intersection of thrust bounds and thrust pointing constraint for $\theta \in (\pi/2, \pi]$ (middle) and $\theta \in (0, \pi/2]$ (right) (Açıkmeşe et al., 2013).

the constraint is convex if $\theta \leq \pi/2$ and nonconvex when $\theta > \pi/2$. As depicted in Fig. 4.4, the maximum allowable angle of deviation with respect to $\hat{\mathbf{n}}$-direction is $0 \leq \theta \leq \pi$ for $t \in [0, t_f]$. To deal with the non-convexity within the thrust pointing constraint, the convexification process in Chapter 5 extends to the pointing realm. In conclusion of the state and control constraints section it is specified that thrust direction can explicitly be constrained at the start and end of the descent sequence through

$$\mathbf{T}_c(0) = \|\mathbf{T}_c(0)\|\, \hat{n}_0 \tag{4.20}$$

$$\mathbf{T}_c(t_f) = \left\|\mathbf{T}_c(t_f)\right\|\, \hat{n}_{t_f} \tag{4.21}$$

where $\hat{n}_0$ and $\hat{n}_{t_f}$ denote unit vectors construed in accordance with the desired thrust direction.

4.2.4. FUEL OPTIMAL LARGE DIVERT

The Guidance for Fuel Optimal Large Divert (G-FOLD) is a proof of concept algorithm for the powered descent phase of planetary precision landing. The algorithm is designed such that fuel optimal mission constraints satisfying paths are autonomously generated onboard that are physically feasible. The Jet Propulsion Laboratory (JPL) has extensively analyzed this algorithm during the last decade and concluded positively on the essence of it. The GNC system overview for G-FOLD is shown in Fig. 4.5, of which the navigation and control aspects are discussed in Sections 4.3 and 4.4, respectively. The remainder of this section is dedicated

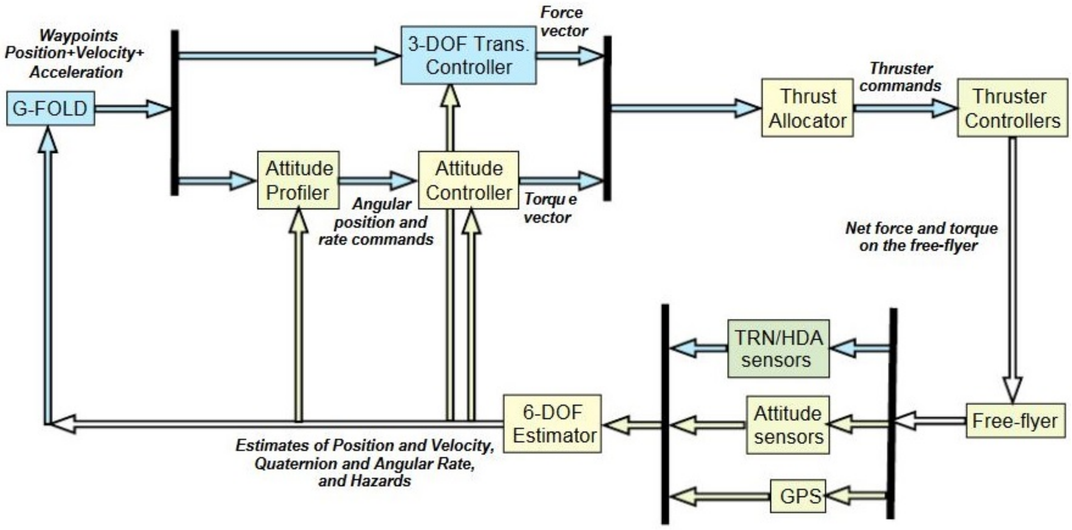


Figure 4.5: G-FOLD guidance, navigation and control scheme depicting the interplay between all components of the flight software. Notice that TRN and HDA refer to Terrain Relative Navigation and Hazard Detection and Avoidance, respectively (Açıkmeşe et al., 2012).

to outlining the G-FOLD algorithm and delineating the Autonomous Ascent and Descent Powered-Flight Testbed (ADAPT). This testbed has been constituted categorically for demonstrating landing technologies of the next-generation landers. The Masten Space Systems G-FOLD test vehicle by the name Xombie is a rocket designed for vertical takeoff and landing in suborbital space (Açıkmeşe, 2017). For an interplanetary mission, the G-FOLD algorithm is initiated after the entry and parachute deployment phases have come to pass, and the spacecraft has been slowed down and descended to a prescribed velocity and altitude, respectively. Atmospheric uncertainties cause dispersions that cannot be predetermined but are very significant in essence. The G-FOLD algorithm is able to tackle this problem autonomously and fuel optimally. This type of optimization solves problems at speeds in the order of micro- to milliseconds, making them a great asset for implication onboard at real time. The nonconvex G-FOLD optimization problem is characterized as:

Problem 3: nonconvex G-FOLD problem definition

Cost Function:

$$\max_{t_f, \mathbf{T}_c} \quad m(t_f) \quad \forall t \in [0, t_f] \quad \text{subject to:}$$

Dynamics:

$$\begin{aligned}
&\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\left(\mathbf{g} + \frac{\mathbf{T}_c(t)}{m(t)}\right) \quad \text{with} \quad A = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad B = \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix} \\
&\dot{m}(t) = -\alpha \, \|\mathbf{T}_c(t)\| \\
&\mathbf{x}(t) \in \mathbf{X}
\end{aligned}$$

Control Constraints:

$$\begin{aligned}
0 < \rho_1 \le \|\mathbf{T}_c\| \le \rho_2 \\
\hat{\mathbf{n}}^T \mathbf{T}_c(t) \ge \|\mathbf{T}_c(t)\| \cos\theta
\end{aligned}$$

Boundary Conditions:

$$\begin{aligned}
m(0) = m_0, \quad \mathbf{r}(0) = \mathbf{r}_0, \quad \dot{\mathbf{r}}(0) = \dot{\mathbf{r}}_0 \\
m(t_f) \ge m_{\text{dry}}, \quad \mathbf{r}(t_f) = \mathbf{0}, \quad \dot{\mathbf{r}}(t_f) = \mathbf{0}
\end{aligned}$$

The dynamic equations in the surface fixed reference frame are defined such that the state vector $\mathbf{x}(t)$, mass vector $m(t)$ and constraint vector $\mathbf{X}$ hold in accordance with

$$\mathbf{x}(t) = (\mathbf{r}(t), \dot{\mathbf{r}}(t)) : \mathbb{R}_+ \to \mathbb{R}^6 \quad \text{and} \quad m(t) : \mathbb{R}_+ \to \mathbb{R}_+ \tag{4.22}$$

$$\mathbf{X} = \left\{ (\mathbf{r}, \dot{\mathbf{r}}) \in \mathbb{R}^6 : \quad \tan\gamma_{gs} \mathbf{e}_1^T \mathbf{r} \ge \sqrt{(\mathbf{e}_2^T \mathbf{r})^2 + (\mathbf{e}_3^T \mathbf{r})^2}, \quad \|\dot{\mathbf{r}}\| \le V_{max} \right\} \tag{4.23}$$

The free-flying closed-loop ADAPT vehicle has demonstrated the ability to onboard calculate divert trajectories in real time during descent. Açıkmeşe (2017) states that the performed test flights manifested divert capabilities reaching 750 m while starting at a 480 m altitude, as Fig. 4.6 depicts. The maneuver was initiated in such a way that the vehicle was moving away from and crosswise with regard to the landing location. In mid-air the aggressive direction reverse was initiated successfully in ~100 ms, an astonishing accomplishment. The vehicle continued its G-FOLD path planned direction, to eventually land precisely on the designated area. Furthermore, it is important to denote that the ADAPT/G-FOLD testbed does not take into account coupled translational and attitude dynamics, i.e., a 6DoF algorithm, as this significantly increases the complexity. Likewise, aerodynamic controls (drag and lift) have been left out of the algorithm. A review on how to incorporate these matters is continued in the sections ahead. The designed closed-loop guidance and control simulator encompassing all the prior specified elements is conferred in Chapter 7.

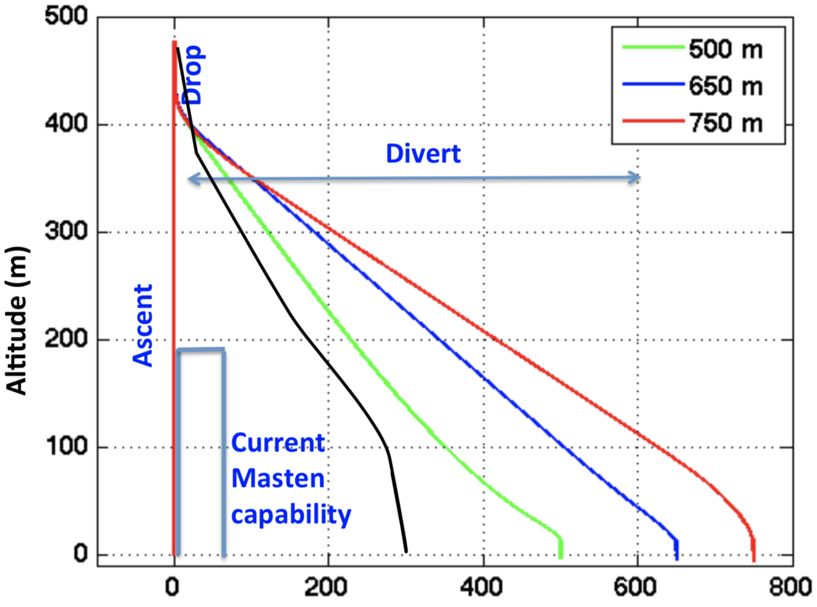


Figure 4.6: Masten Xombie vehicle ADAPT trajectories capabilities starting at an altitude of 450 m for three divergence scenarios: 500, 650 and 750 m (Açıkmeşe, 2017).

4.2.5. Aerodynamic Controls

The convex programming approach is extended to a descent guidance problem in the presences of aerodynamic forces, drag to begin with. This also includes the introduction of a new type of nonconvex control constraint. Due to the control constraints' non-convex nature and the aerodynamics induced nonlinearities, the problem complexity for real-time guidance increases significantly. In a later stage of the thesis, Chapter 5 specifically, the lossless convexification is described that eliminates the minimum thrust constraint and successive convexification is delineated that addresses the remaining nonconvex elements. Aerodynamic forces are particularly dominant for entry and parachute descent guidance in the Mars and Titan environments. For a Titan landing these forces remain eminent during terminal descent due to the fact that a parafoil is guided based upon aerodynamic controls. Nonetheless, during powered descent guidance on Mars aerodynamic phenomena may have significant effect as well, even at lower altitudes. This section shall therefore first outline the fuel optimal powered landing guidance problem with aerodynamic drag based on Szmuk et al. (2016). The algorithm is subsequently formulated as a minimum landing error problem utilizing aerodynamic controls. Notice that both frameworks conferred still assume 3DoF dynamics.

Fuel Optimal Powered Descent Guidance with Aerodynamic Controls

As in the preceding sections, the vehicle is point mass modeled subject to translational dynamics as conferred Section 4.2.1. Additionally, an aerodynamic force $\mathbf{F_A}(t)$ is noted such that

$$\mathbf{F_A}(t) = \frac{1}{2}\rho S_{ref}\mathbf{v}^T\mathbf{v}\begin{pmatrix} C_D \\ C_S \\ C_L \end{pmatrix} \tag{4.24}$$

where the air density ρ is a function of altitude, C_D is the drag coefficient and S_{ref} the reference area. As the powered landing phase occurs at a relatively low altitude within the EDL sequence, density might be assumed constant. Now consider a constant ambient air pressure P_{amb} and rocket nozzle exit area A_{nozzle} leading to an adjustment in the mass depletion function (Eq. (4.2)) to account for back pressure compensation

$$\dot{m}_{bp} = \frac{P_{\text{amb}}A_{\text{nozzle}}}{I_{sp}g_0} \tag{4.25}$$

$$\dot{m}(t) = -\alpha\,\|\mathbf{T}_c(t)\| - \dot{m}_{bp} \tag{4.26}$$

The translational dynamics model encompasses a double integrator driven by a term of acceleration

$$\dot{\mathbf{r}}(t) = \mathbf{v}(t) \tag{4.27}$$
$$\dot{\mathbf{v}}(t) = \mathbf{a}(t) \tag{4.28}$$
$$\mathbf{a}(t) = \frac{1}{m(t)}\left[\mathbf{T}(t) + \mathbf{F_A}(t)\right] + \mathbf{g} \tag{4.29}$$

Within the continuous time domain formulation, the objective of the algorithm is to maximize final mass while adhering to the constraints and conditions. The additional control constraints included in this problem formulation ensure that throttling rate values maintain within a threshold minimum $\dot{T}_{min}$ and maximum $\dot{T}_{max}$. Notice that the acceleration term includes a convex quadratic term which needs to be mathematically detached using a linearization scheme. The nonconvex problem formulation with aerodynamic drag reads:

Problem 4: nonconvex minimum fuel guidance problem with aerodynamic drag

Cost Function:

$$\max_{t_f, \mathbf{T}_c} \quad m(t_f) \quad \forall t \in [0, t_f] \quad \text{subject to:}$$

Dynamics:

$$\begin{aligned}
\dot{m}(t) &= -\alpha \left\|\mathbf{T}_c(t)\right\| - \dot{m}_{bp} \\
\dot{\mathbf{r}}(t) &= \mathbf{v}(t) \\
\dot{\mathbf{v}}(t) &= \mathbf{a}(t) \\
\mathbf{a}(t) &= \frac{1}{m(t)}\left[\mathbf{T}(t) + \mathbf{F_A}(t)\right] + \mathbf{g}
\end{aligned}$$

State Constraints:

$$\begin{aligned}
m_{\text{dry}} &\leq m(t) \\
\left\|\mathbf{r}(t)\right\| \cos\gamma_{gs} &\leq \hat{\mathbf{n}}^T \mathbf{r}(t)
\end{aligned}$$

Control Constraints:

$$\begin{aligned}
0 < \rho_1 \leq \left\|\mathbf{T}_c\right\| &\leq \rho_2 \\
\hat{\mathbf{n}}^T \mathbf{T}_c(t) &\geq \left\|\mathbf{T}_c(t)\right\| \cos\theta \\
\dot{T}_{\text{min}} \leq \frac{d}{dt}\left\|\mathbf{T}(t)\right\| &\leq \dot{T}_{\text{max}}
\end{aligned}$$

Boundary Conditions:

$$\begin{aligned}
&m(0) = m_0, \quad \mathbf{r}(0) = \mathbf{r}_0, \quad \dot{\mathbf{r}}(0) = \dot{\mathbf{r}}_0 \quad \mathbf{T}(0) = T_0 \hat{\mathbf{n}}_0 \\
&m(t_f) \geq m_{\text{dry}}, \quad \mathbf{r}(t_f) = \mathbf{0}, \quad \dot{\mathbf{r}}(t_f) = \mathbf{0} \quad \mathbf{T}(t_f) = \left\|\mathbf{T}(t_f)\right\| \hat{\mathbf{n}}_f
\end{aligned}$$

MINIMUM CONTROL EFFORT - PARAFOIL GUIDANCE

The latter problem of the previous part is formulated as minimum power consumption problem for terminal parafoil descent guidance. This is established by enforcing a cost function that minimizes the control effort. This control effort is constrained to have normalized values between 0 and 1, in accordance with canopy pull regulations. Notice that the total problem described in this section also includes thrust, which will naturally equal zero in case there is no flow of mass. For a parafoil the mass is assumed to be constant (without the consideration of apparent mass). The algorithm is designed such that is can be applied to both cases, based on imposed mission parameters and constraints. The formulation, (Problem 5), conferred in this section thus encompasses full generality, in the sense all possible relevant forces are stated. The pointing constraint remains to be a part of the formulation to cope with the requirements of TRN, but will independent of thrust in case of a parafoil landing. The aerodynamic angles, angle-of-attack α and side-slip angle β are constrained such that they have a threshold minimum and maximum value. The boundary conditions of the sequence are constrained such that the soft-landing is enforced through zero position error (with respect to the target in the guidance frame) and zero touchdown velocity at termination in three-dimensional Euclidean space. The vehicle is point mass modeled subject to translational dynamics as conferred Section 4.2.1. A significant difference with respect to the prior formulation (Problem 4) is that the solution is the fact that the aerodynamics forces are now considered to be control authorities. This implies that the drag coefficient given in Eq. (4.24) is no longer treated as a constant but rather as a function of angle-of-attack α_a and symmetric deflection δ_s. Similarly, the lift force coefficient is considered to be a function of α_a and δ_s. In the next section the moment, angular rate and quaternion dynamics are formulated, after which the 6DoF simulation environment problem definition is conferred.

Problem 5: nonconvex minimum control effort guidance problem

Cost Function:

$$\min_{t_f,\mathbf{T}_c} \|\mathbf{u}(t)\| \qquad \text{subject to:}$$

Dynamics:

$$\begin{aligned}\dot{\mathbf{r}}(t) &= \mathbf{v}(t)\\ \dot{\mathbf{v}}(t) &= \mathbf{a}(t)\\ \mathbf{a}(t) &= \frac{1}{m(t)}\left[\mathbf{T}(t)+\mathbf{F_A}\right]+\mathbf{g}\end{aligned}$$

State Constraints:

$$\begin{aligned}m_{\text{dry}} &\le m(t)\\ \|\mathbf{r}(t)\|\cos\gamma_{gs} &\le \hat{\mathbf{n}}^T\mathbf{r}(t)\end{aligned}$$

Control Constraints:

$$\begin{aligned}0 < \rho_1 \le \|&\mathbf{T}_c\| \le \rho_2\\ \hat{\mathbf{n}}^T\mathbf{F}_c(t) &\ge \|\mathbf{F}_c(t)\|\cos\theta\\ 0 < &u(t) \le 1\\ \alpha_{\text{min}} < &\alpha(t) \le \alpha_{\text{max}}\\ \beta_{\text{min}} < &\beta(t) \le \beta_{\text{max}}\end{aligned}$$

Boundary Conditions:

$$\begin{aligned}m(0) = m_0, \quad \mathbf{r}(0) = \mathbf{r}_0, \quad \dot{\mathbf{r}}(0) = \dot{\mathbf{r}}_0 \quad \mathbf{F}(0) = F_0\hat{\mathbf{n}}_0\\ m(t_f) \ge m_{\text{dry}}, \quad \mathbf{r}(t_f) = \mathbf{0}, \quad \dot{\mathbf{r}}(t_f) = \mathbf{0} \quad \mathbf{F}(t_f) = \left\|\mathbf{F}(t_f)\right\|\hat{\mathbf{n}}_f\end{aligned}$$

4.2.6. Quaternion Attitude Representation

For the powered descent simulator a 6DoF guidance simulator has been developed in this thesis. While the computation time does increase for higher order of degree analyses, the requirement seems inevitable for reaching a precision in the order of meters (Blackmore, 2017). Several available guidance methodologies have been investigated for this pre-landing phase. One of such is the influential E-Guidance method and the modified Apollo guidance version described by Gerth and Mooij (2014). These guidance methodologies treat the problem at hand as a two-point boundary value problem. The paper describes how the method is not satisfyingly robust and thus advises to increase its robustness for increased precision. The methodology construed in this section is numerical in nature and based on the 6 DoF convexification for free final time deduced from Szmuk and Açıkmeşe (2018). Because of the convexification, the difficulty of the problem is decreased significantly while results remain intact. Furthermore, the system is implantable in real-time which is a high-level mission requirement. To guarantee the latter, the Interior Point Method (IPM) is used as a solver. In this part of the research the focus lays on the describing the methodology. Guidance for autonomously landing spacecraft comes along with three challenges for which the following solutions are provided:

1. Non-linear nature of the problem → lossless convexification.
2. Discretization scheme determines the solution's validity → linear time-variant matrices system ensure solution convergence in accordance with the non-linear system.
3. Reference trajectory generation for initialization is difficult → algorithm is capable of initialization for inconsistent trajectories.

The translational dynamics is defined in the surface fixed reference frame as defined in Section 3.1. Modelling of the thrust vector during vector during powered descent will be such that it is assumed that the rocket engines are in a gimbal locked orientation. The mass depletion function for the rocket is given by $\lambda_m = 1/I_{sp}g_0$ leading to the following dynamics:

$$\dot{m} = -\lambda_m \|\mathbf{T}_B(t)\| \tag{4.30}$$

Expressing the translational equations of motion in the inertial guidance frame leads to:

$$\dot{\boldsymbol{r}}_G(t) = \mathbf{V}_G(t) \tag{4.31}$$

$$\dot{\boldsymbol{V}}_G(t) = \frac{1}{m(t)}\mathbf{F}_G(t) + \mathbf{g}_G \tag{4.32}$$

Notice that $\mathbf{F}_G$ accounts for all relevant forces. Orienting the body frame relative to the inertial guidance frame occurs through quaternion parametrization of the direction cosine matrix defined in Eq. (3.20). Because the definition of the transpose of a matrix being equal to the negative of the matrix is needed, the skew-symmetric matrices are defined for some $\boldsymbol{\xi} \in \mathbb{R}^3$ such that:

$$[\boldsymbol{\xi}\times] = \begin{bmatrix} 0 & -\xi_z & \xi_y \\ \xi_z & 0 & -\xi_x \\ -\xi_y & \xi_x & 0 \end{bmatrix} \qquad \Omega(\boldsymbol{\xi}) = \begin{bmatrix} 0 & -\xi_x & -\xi_y & -\xi_z \\ \xi_x & 0 & \xi_z & -\xi_y \\ \xi_y & -\xi_z & 0 & \xi_x \\ \xi_z & -\xi_y & -\xi_x & 0 \end{bmatrix} \tag{4.33}$$

While the attitude dynamics is in accordance with Eq. (3.47) and the angular velocity of the body frame relative to the guidance frame is denoted as $\boldsymbol{\omega}_{B,G}(t) \in \mathbb{R}^3$, the kinematic equation and attitude dynamics in quaternion form are defined as:

$$\dot{\boldsymbol{q}}_{B,G}(t) = \frac{1}{2}\Omega\left(\boldsymbol{\omega}_{B,G}(t)\right)\mathbf{q}_{B,G}(t) \tag{4.34}$$

$$\boldsymbol{J}_B\dot{\boldsymbol{\omega}}_B(t)\mathbf{M}_B(t) - \boldsymbol{\omega}_B(t) \times \boldsymbol{J}_B\boldsymbol{\omega}_B(t) \tag{4.35}$$

In the above equation $\boldsymbol{J}_B$ defines the normalized inertia tensor of the spacecraft. The earlier defined gimbal lock orientation assures that the CoM does not move and thus the engines maintain constant in distance with respect to it. The position of the engines relative to the CoM $\mathbf{r}_{T,B} \in \mathbb{R}^3$ is defined in the body frame and leads to the propulsive moment expression:

$$\mathbf{M}(t) = \mathbf{r}_{T,B} \times \mathbf{T}_B(t) \tag{4.36}$$

Knowing this, the state and control constraints can be defined. The first convex constraints yields that the mass of the vehicle shall remain above the dry mass to make sure fuel depletion does not occur:

$$m_{\text{dry}} \leq m(t) \tag{4.37}$$

The trajectory is restricted to have a downward facing glide cone such that the angle $\gamma_{gs} \in [0°, 90°)$. Defined as convex constraints, the following relations are obtained:

$$\mathbf{e}_1 \cdot \mathbf{r}_G(t) \geq \tan\gamma_{gs} \left\| H_{23}^T \mathbf{r}_G(t) \right\|_2 \tag{4.38}$$

$$H_{23} = [\mathbf{e}_2 \quad \mathbf{e}_3] \tag{4.39}$$

The orientation of the tilt angle of the vehicle relative to guidance frame is denoted as η(t) expressed in formal format as:

$$\cos\eta(t) = \mathbf{e}_1 \cdot \mathbf{C}_{G,B}(t)\mathbf{e}_1 = 1 - 2(q_2^2(t) + q_3^2(t)) \tag{4.40}$$

To avoid excessive tilt maneuvers during the powered descent sequence, a convex constraint is imposed in $\mathbb{R}^4$ quaternion space:

$$\cos\eta_{\max} \leq 1 - 2(q_2^2(t) + q_3^2(t)) \tag{4.41}$$

The final convex constraint makes sure the angular rate of the vehicle is limited to a certain threshold value such that:

$$\|\boldsymbol{\omega}_B(t)\|_2 \leq \omega_{\max} \tag{4.42}$$

Problem 6: general 6DoF nonconvex terminal guidance problem

Cost Function:

$$\max_{t_f, \mathbf{T}_c} \quad m(t_f) \quad \text{or} \quad \min_{t_f, \mathbf{T}_c} \|\mathbf{u}(t)\| \qquad \text{subject to:}$$

Dynamics:

$$\begin{aligned}
\dot{\mathbf{r}}(t) &= \mathbf{v}(t) \\
\dot{\mathbf{v}}(t) &= \frac{1}{m(t)} C_{I/B}(t)\mathbf{T}_B(t) + \mathbf{g}_I \\
\dot{m}(t) &= -\alpha_{\dot{m}} \\
\dot{\boldsymbol{q}}_{B/I}(t) &= \|\mathbf{T}_B(t)\| \frac{1}{2}\Omega(\boldsymbol{\omega}_B(t))\boldsymbol{q}_{B/I}(t) \\
\boldsymbol{J}_B\dot{\boldsymbol{\omega}}_B(t) &= [\mathbf{r}_{T,B}\times]\mathbf{T}_B(t) - [\boldsymbol{\omega}_B(t)\times]\boldsymbol{J}_B\boldsymbol{\omega}_B(t)
\end{aligned}$$

State Constraints:

$$\begin{aligned}
m_{\text{dry}} &\leq m(t) \\
\|\mathbf{r}(t)\| \cos\gamma_{gs} &\leq \hat{\mathbf{n}}^T\mathbf{r}(t) \\
\cos(\theta_{\max} &\leq 1 - 2(q_2^2(t) + q_3^2(t)) \\
\|\boldsymbol{\omega}_B(t)\|_2 &\leq \omega_{\max}
\end{aligned}$$

Control Constraints:

$$\begin{gathered}
0 < \rho_1 \leq \|\mathbf{T}_c\| \leq \rho_2 \\
\hat{\mathbf{n}}^T\mathbf{F}_c(t) \geq \|\mathbf{F}_c(t)\| \cos\theta \\
0 < u(t) \leq 1 \\
\alpha_{\min} < \alpha(t) \leq \alpha_{\max} \\
\beta_{\min} < \beta(t) \leq \beta_{\max}
\end{gathered}$$

Boundary conditions:

As defined in Table 4.1.

Table 4.1: Initial and final conditions of the convexification 6 DoF guidance system

	Initial condition	**Final condition**
Mass	m_{wet}	$\leq m_{\text{dry}}$
Position	$\mathbf{r}_{G,0}$	$\mathbf{0}$
Velocity	$\mathbf{V}_{G,0}$	$\mathbf{V}_{G,f}$
Angular rate	$\boldsymbol{\omega}_{B,0}$	$\mathbf{0}$
Attitude	free	$\mathbf{q}_{B,G,f}$

The boundary conditions imposed on the constraints are free to define in the sense that various combinations are possible. For the Mars landing simulations this research adopted the model by Szmuk and Açıkmeşe (2018) because both the initial and final conditions are in perfect accordance with the mission scenario of MSL, as specified in Section 2.3. The initial conditions and final conditions are summarized in Table 4.1. The problem description for the Mars ($\max_{t_f, \mathbf{T}_c} \quad m(t_f)$) and Titan ($\min_{t_f, \mathbf{T}_c} \|\mathbf{u}(t)\|$) 6DoF landing is conferred below. Notice that depending on the planetary body, the cost is function is defined. The propulsive forces are absent on Titan, which indirectly implies that the mass remains constant over time. On the other hand, lift forces are neglected for the Mars landing sequence. The quaternion and attitude dynamics holds for both sequences. Problem 6 encompasses the nonconvex algorithm description in its full generality.

4.3. Navigation

Accurate estimation of the state of the spacecraft through a navigation scheme forms a fundamental aspect of planetary landing and is therefore indispensable during the EDL sequence. This is especially true for planetary precision landing which requires near precise knowledge of state variables. During approach, navigation systems can maintain reliance on Earth-based activity through the Deep Space Network because there is sufficient time for the uplink of commands through relay telemetry. However, due to light-time delays discussed in Section 4.1.3, this dependency is not an option anymore. The entire operation in which the spacecraft needs to actively guide itself, from atmospheric entry to final landing must thus be fully autonomous (Yu et al., 2017). This section discusses common EDL sensors and their applications and outlines the reasons for assuming perfect state knowledge.

4.3.1. Navigation Sensors

The type of navigation sensors commonly applied for estimating state variables during planetary precision landing are discussed in this section. Computing the trajectories discussed in the guidance section in real-time requires external information from the surrounding environment. The two commonly used sensors for the before-mentioned application are the inertial measurement unit (IMU) and the ground position system (GPS). The combined working effect of convex optimization guidance and GPS navigation was proven by SpaceX to be sufficient for achieving precision landing on Earth of the reusable first stage rockets. However, as declared in Blackmore (2017), any planetary body other than the Earth is a GPS denied environment. Yu et al. (2015) points out that traditional navigation methodologies are not sufficient to fulfill the need of precision landing. Besides the IMU and GPS sensors, several other sensor techniques and instruments exists that aid in navigation. Examples are LIDAR (Light Detection and Ranging) optical sensors for remote sensing and more sophisticated technologies like Terrain Relative Navigation (TRN). The latter, being an integral part of the Mars 2020 mission, has moderately been considered in this research. The guidance system takes into consideration the pointing requirements to allow for sensor measurements. This section highlights several navigation sensors and defines the reasons for assuming perfect state knowledge.

Inertial Measurement Unit

Initially developed for rockets, the inertial measurement unit is a pragmatic device for navigation in general. To measure ambient gravity induced acceleration, accelerometers are used and for angular velocity measurements gyroscopes are incorporated. Furthermore, the IMU itself needs estimations of both position and velocity from an alternative source to start state propagation. Increased error in this initial estimation leads to decreased accuracy of propagation. From a theoretical perspective, measurements through IMUs are perfectly self-sufficient in the sense that no additional information about the surroundings is needed. However, a total of at least nine error sources makes theory different from practical applications. In fact, measuring state with infinite precision is practically impossible. While modelling all error sources is favorable for manufactures, end users prefer more practical models to ease performance analysis. Commonly, navigation design commences with the inclusion of three errors into the model: scale factors s, bias b and misalignments m. Bias, misalignments and scale factors tend to have random walks, while the IMU system is switched on. These random walks are expressed as a white noise normally distributed differential equation and they are time dependent in nature. Randomly generating these errors, for example with a normal distribution, is therefore not an appropriate representation of reality. Furthermore, navigation data of prior (NASA) missions is not available in the public domain due to the International Traffic in Arms Regulations. Due to these reasons this research assumed perfect state knowledge for both the nominal and error-integrated simulations. Consequently, without the proper determination of state, the vehicle would loose orientation relative to the nominal trajectory. For actual flight applications the navigation is thus a crucial entity.

Terminal Descent Sensor

Besides employing the classical IMU for state estimation, MSL required an order of magnitude improvement in terms of sensing: the Terminal Descent Sensor (TDS). The need for developing this system followed mainly from the sophisticated sky-crane landing system. The working concept is rather straight forward and an accuracy in the order of 10^{-1} m/s in terms of velocity is typically achieved. For the range values are obtained within a maximum error spectrum of 2%. To be able to guide and control the spacecraft in the Martian atmosphere knowledge of three dimensional velocity and altitude are required. Similar to the IMU sensors delineated in the preceding part, TDS sensors typical include sources of time-dependent errors such as biases and scale factors. Likewise, navigation data of prior missions is not available. This yet leads to the forced assumption of perfect state knowledge. Nonetheless, there are various different methodologies for testing the robustness of the guidance system, as conferred in chapters ahead.

4.4. Control

It was brought forth several times that the control system is an indispensable part of the closed GNC loop was therefore included in the research. Likewise to the guidance system, the control system must be able to act fully autonomously while persisting the control constraints. As identified by Lee and Mesbahi (2017), planning the motion of a spacecraft bearing both translational and rotational motion in mind is quite a challenge. While quite sophisticated algorithms exits to tackle this problem, such as coupled dual quaternion model predictive control (Lee and Mesbahi, 2017), this research merely focuses on closing the GNC loop. It is important to denote hereby that the control authority to be defined does take into account that the control is executed at an order of magnitude higher frequency with respect to the guidance system to account for rotational dynamics too. To understand the governing principles of control theory, Section 4.4.1 starts with outlining the general state space formulation often utilized for control applications. Based on the proposed guidance schemes for terminal descent, the relevant PID control design strategy is outlined.

4.4.1. State Space Formulation

The state space representation arranges the dynamics of a system in a set of coupled first-order differential equations in the time domain. It is a widely used method for characterizing both single- and multi-variable systems because of the constitutional synergy insight it provides (Whalley, 1991). The representation format allows for efficient information structuring, prediction of responses and stability analyses. The paper also describes how state space formulations maximize accuracy of computational arithmetic and increases programming efficiency significantly. Modern control theory has adapted this format for problem definition and because of its inherent powerful structure it has become the most convenient way to represent equations of motion. While alternate state space forms do exits, this research sticks to the definition by Rowell (2002), such that the general formulation of the EoM in matrix form is notated as:

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu} \tag{4.43}$$

In this so-called dynamics equation, $\mathbf{x}$ is defined as the state vector including all relevant state variables while $\mathbf{u}$ is the control input vector. These vectors have dimensions of $n \times 1$ and $m \times 1$, respectively. The state coefficient matrix $\mathbf{A}$ is of size $n \times x$ and control coefficient matrix $\mathbf{B}$ of a $n \times m$ size. To complete the state space representation, an output equation is needed:

$$\mathbf{y} = \mathbf{Cx} + \mathbf{Du} \tag{4.44}$$

In this equation $\mathbf{y}$, with size $k \times 1$, is defined as the output vector. From that it follows that transmission matrices $\mathbf{C}$ and $\mathbf{D}$ (typically $\mathbf{D} = \mathbf{0}$) have dimensions $k \times n$ and $k \times m$, respectively. Depending on the nature of the problem and the available control authorities, state and control variables can be defined. The dynamics of a system points out a set of minimum state variables to fully characterize the response of that system for a given input set. Often, the state of any system is determined for physical applications and is directly correlated with the independent energy storage in that system (Nise, 2011). This may be formulated in an explicit mathematical description:

> *Prediction of the state of the system and its output for all $t > t_0$ requires a definition of the minimum set of variables $x_i(t)$ for $i = 1, ..., n$, and intelligence regarding the variables at an initial designated time t_0.*

The order of the system is hence determined by the dimension of n, whose magnitude depend on the cohort state variables. It is important to denote that the above derivation is restricted to systems that are linear and time-invariant. The block diagram representation of the total state space model is presented in Fig. 4.7.

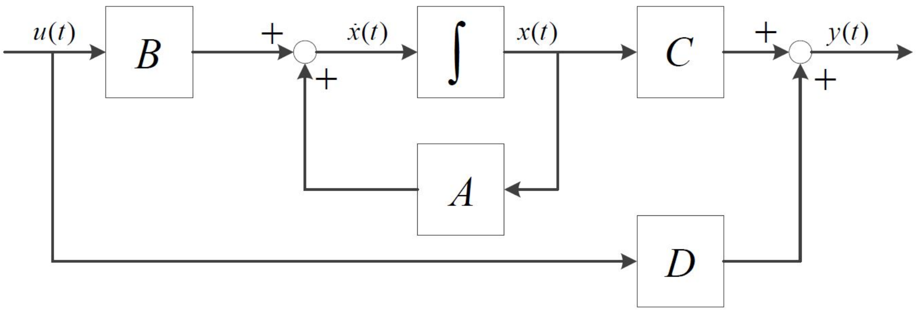


Figure 4.7: State space dynamics in block diagram format (Nise, 2011).

4.4.2. CONTROL DESIGN STRATEGIES

Within the given range in which the physical variables are allowed to vary, it is the task of the control system to provide stability and robust performance. The use of feedback control systems has several advantages. Amongst them, the reduction of phenomena such as noise and disturbance and the increased improvement of tracing command signals (Dahleh et al., 2011). Furthermore, the effects of uncertainties in the dynamics are significantly decreased. Any mathematical model utilized to captivate a motion problem is practically never perfect. Therefore, to maintain closed-loop stability, utilizing a sufficiently robust controller is imperative. The start of this part of the control section focuses on feedback control, after which two often utilized control design strategies are outlined: the PID and LQR control methods.

FEEDBACK CONTROL

To aid the discussion on feedback control, a graphic overview is depicted Fig. 4.8. In this schematized situation the plant represents the system to be controlled which, for the specific application at hand, is the EDL spacecraft. It can be fairly stated that controller is driven by errors and through these inaccuracies inputs are granted to the system. To formulate the control laws, the defined state space format proofs to be convenient.

Through the feedback loop the system becomes a closed loop with the following control law:

$$\mathbf{u} = -\mathbf{K}\mathbf{x} \tag{4.45}$$

Combining Eq. (4.43) with the time liberated feedback gain **K** in Eq. (4.45) yields the following equation:

$$\dot{\mathbf{x}} = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{x} \tag{4.46}$$

Assessment of the eigenvalues and complementary eigenmotion may be obtained through the characteristic equation of Eq. (4.46):

$$\det|\mathbf{A} - \mathbf{B}\mathbf{K} - \lambda\mathbf{I}| = 0 \tag{4.47}$$

Variation of this gain matrix allows for altering the eigenvalues of the system. It is important to denote however that controllability does not depend only on the gain. To be controllable the following mathematical controllability condition, that depends on matrices **A** and **B** must hold (Kuo, 1987):

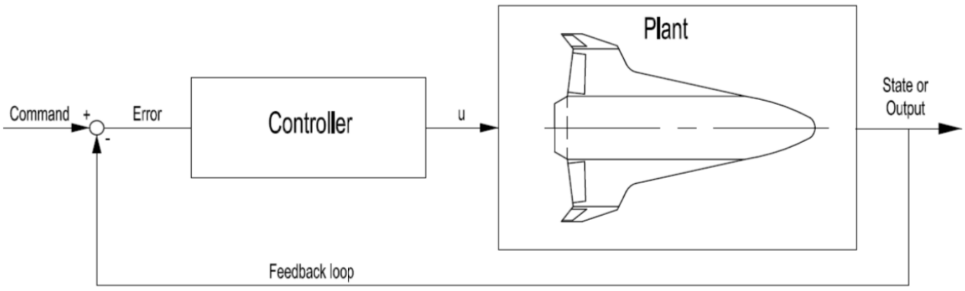


Figure 4.8: Arbitrary plant in a closed-loop feedback control system (Mooij, 2017).

> *For a given state vector* $\mathbf{x}(t)$ *to be controllable at time* $t = t_0$*, there must exist a control input* $\mathbf{u}(t)$ *that is piece-wise continuous in nature and drives the system state to a final state* $t = t_f$ *within a finite time, such that* $t_f > t_0$*. The system is completely state controllable if this conditions hold for every state* $\mathbf{x}(t_0)$*.*

A mathematical observability condition assesses whether the stability of the system is observable (Kuo, 1987):

> *For a given state vector* $\mathbf{x}(t)$ *to be observable there must exist a time* $t_f > t_0$ *that is of finite nature and given any control input* $\mathbf{u}(t)$ *the knowledge of:*
>
> *i) the control input* $\mathbf{u}(t)$ *for* $t_0 \le t < t_f$*,*
> *ii) the four state space matrices* $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ *and* $\mathbf{D}$*,*
> iii) the state output $\mathbf{y}$ for $t_0 \le t < t_f$,
>
> *are adequate to determine* $\mathbf{x}(t_0)$*. The system is completely observable if this conditions hold for every state of the system during a finite time* t_f*.*

To inspect the controllability and observability of a system, it is both necessary and sufficient that:

$$\begin{bmatrix}\mathbf{B} & \mathbf{AB} & \mathbf{A^2B} & \cdots & \mathbf{A^{n-1}B}\end{bmatrix} \tag{4.48}$$

$$\begin{bmatrix}\mathbf{C}^T & \mathbf{A}^T\mathbf{C}^T & (\mathbf{A}^T)^2\mathbf{C}^T & \cdots & (\mathbf{A}^T)^{n-1}\mathbf{C}^T\end{bmatrix} \tag{4.49}$$

have the same dimensions as the vector of state and thus have rank n. Influencing of particular eigenmotions is only possible if these are part, as stated in Mooij (2017), of the controllable and observable state space. In a similar fashion as the state feedback control, the control law for output feedback may be obtained:

$$\mathbf{u} = -\mathbf{K}\mathbf{y} = -\mathbf{K}(\mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}) \tag{4.50}$$

The assumption that $\mathbf{D} = \mathbf{0}$ is widely used, decreasing the difficulty of the application of the control law. Combing this assumption with Eq. (4.43) and the above equation yields:

$$\dot{\mathbf{x}} = (\mathbf{A} - \mathbf{BKC})\mathbf{x} \tag{4.51}$$

Analogously, the characteristic equation for this equation follows from:

$$\det|\mathbf{A} - \mathbf{BKC} - \lambda\mathbf{I}| = 0 \tag{4.52}$$

Through the astute selection of the gain the eigenmotion of the closed-loop system can be managed. In Ogata (2010) proposed methods for behaviour analyses include placement of poles or using Bode plots. A more

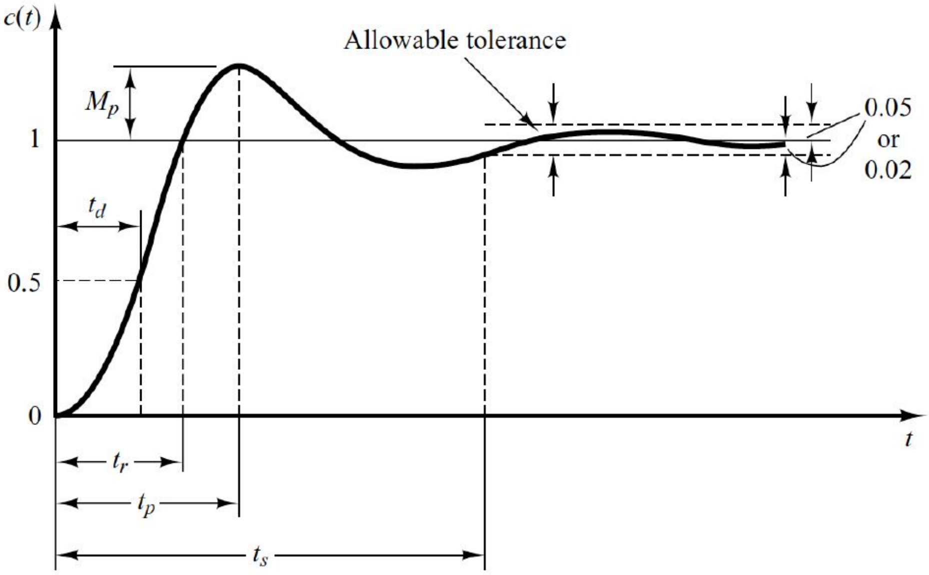


Figure 4.9: Analysis of the steady-state and transient response (Ogata, 2010).

convenient and idiosyncratic methodology of control system analyses is by specifying the characteristics of the transient-response to unit-step inputs as depicted in Fig. 4.9. This method is symbolized by its easy implementation and rapid specification of peculiar symptoms:

1. Delay time t_d → time it takes the response to reach 50% of the final value.
2. Rise time t_r → time it takes the response to rise from 10% to 90% for overdamped systems or 0% to 100% for underdamped systems, both of second order.
3. Peak time t_p → time it takes the response to extend to the overshoot peak.
4. Maximum percent overshoot M_p → unity measured maximum peak overshoot of the response indicative of the systems relative stability.
5. Settling time t_s → time it takes the response to extend and withhold within a specified absolute percentage range, typically 2% or 5%.

PROPORTIONAL INTEGRAL DERIVATIVE CONTROL

A well-known and the most utilized feedback control tool is the Proportional Integral Derivative (PID) controller. The working principle of this controller is to first calculate an error between the measured and desired value and subsequently adjust the control input **u** as such that the miscue is minimized. In case either the system at hand or the flight environment are not known to an accurate extend, the PID controller provides a good starting point. Disadvantages are that optimal control and stability are not guaranteed. The combined complete PID control law $u(t)$ is written as a function of the error signal $e(t)$ and the respective PID gains:

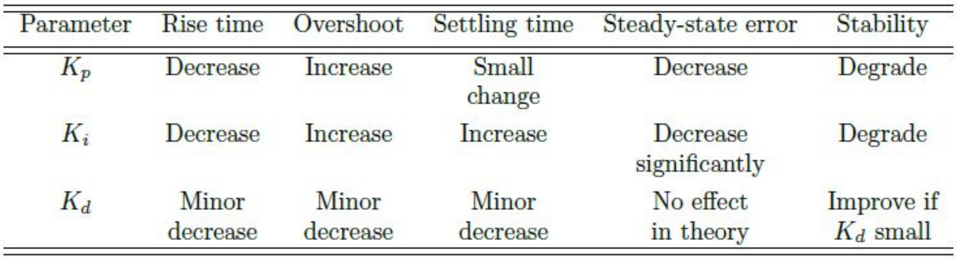
$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt} \tag{4.53}$$

To understand the underlying working principle of the single-input single-output system the various gain contributions are outlined:

1. Proportional gain K_p → in essence an amplifier in direct proportion to the error signal. Choosing the gain too high leads to instability while a low gain value slows down the response.
2. Integral gain K_i → change of the output $u(t)$ in proportion to accumulated past errors. Commonly effective in reduction of steady-state residual errors but easily gives rise to overshoot. The latter problem might be tackled by introducing a limit integrator that is only triggered for certain bandwidths of error.
3. Derivative gain K_d → proportional to the error change rate often used for overshoot magnitude reduction. The change rate and the transient response of the controller is however slowed down.

Based on the various properties, gain tuning is often necessary to find an optimal combination of the gains. The influence of the three control gains on the response are captivated in Table 4.2. This table provides a good starting points for control tuning. For more sophisticated analyses for gain tuning are given in for example Kuo (1987), D'Souza (1988) and Ogata (2010). The built-in function of Matlab - Simulink allows for automatic optimal control that meets specified design requirements. The control system toolbox provides the advantages that optima are even found for plants that traditional methodologies are not able to handle quite as well. The system is even capable of generating response plots similar to the one depicted in Fig. 4.9, earlier discussed Bode plots and allows easier transfer of analyses in the time or frequency domain. Ultimately, this allows for verification of design parameters as delay, rise and settling time, overshoot and margins of gains. This marks the conclusion of the guidance, navigation and control chapter. The next chapter outlines the mathematical principles of the convex optimization method.

Table 4.2: Operational conditions and critical requirements of the PID control system (Mooij, 2017).



Parameter	Rise time	Overshoot	Settling time	Steady-state error	Stability
K_p	Decrease	Increase	Small change	Decrease	Degrade
K_i	Decrease	Increase	Increase	Decrease significantly	Degrade
K_d	Minor decrease	Minor decrease	Minor decrease	No effect in theory	Improve if K_d small

5

Convex Optimization

In this chapter the mathematical principles of the convex optimization are conferred, distinguishing between lossless and successive optimization guidance. The start of the chapter marks the general formulation of the convex optimization problem. Subsequently, the necessary linearization process is charted, along with a stability assessment for control purposes. The original planetary landing lemmas are formulated for the Mars powered descent guidance problem and are conferred in Sections 5.2 and 5.3, respectively. Coupled to these two methodologies is the problem of solving for optimal time-of-flight. In the section on lossless convexification the golden search algorithm is outlined, a powerful method for solving the matter of optimal fight time. The successive counterpart is another powerful approach which solves for time within the optimization framework, making t a convex optimization variable. Besides the original formulation, this chapter outlines the expansion of the algorithm for the application of convex optimization to precision landing on Titan. This entails an aerodynamic parafoil descent guidance problem in which the control authority is obtained through the pull of the canopy. For details regarding the parafoil dynamics the reader is referred back to Section 3.6. The current chapter is concluded with the expansion of the algorithm to allow for obstacle avoidance during the landing sequence. This is achieved by applying a multi glide-slope constraint, relaxing the angle as the space vehicle approaches the terminal location.

5.1. General Formulation

In this section the general formulation of the convex optimization problem is outlined. Included in the discussion is the reaching of optimal control for dynamical systems through the Pontryagin Maximum Principle. The general formulation ends with a comparison between convex and nonconvex problems. A general optimization problem mathematically defined with a vector x as the optimization variable reads:

$$\begin{aligned} &\text{minimize} f_0(x) \\ &\text{subject to} f_i(x) \leq b_i, \quad i = 1, ..., m \end{aligned} \tag{5.1}$$

In the above formulation f_0 is the objective function and f_i the constraint functions subject to constants b_i. For convex optimization problems both the objective as well as the constraint functions are convex such that the following inequality is satisfied for all $x, y \in \mathbb{R}^n$ and all $\alpha, \beta \in \mathbb{R}$ such that $\alpha + \beta = 1$, $\alpha \geq 0$, $\beta \geq 0$:

$$f_i(\alpha x + \beta y) \leq \alpha f_i(x) + \beta f_i(y) \tag{5.2}$$

Two well-known subclasses of convex optimization are the least-squares method and the linear programming method. These methods are not regarded in this research because the former takes no constraints into account while the latter assumes all constraint functions as linear. This is not useful for the problem at hand that requires inequality constraints to be satisfied to ensure planetary precision landing. Solving the convex optimization problem is not done through a predefined analytical formula as it does not exists. However, effective methods like the Interior-point approach are quite practical for producing full solutions. This method is generally capable of solving a convex problem within a feasible iteration rate (Boyd and Vandenberghe, 2004) and its functioning foundation is discussed subsequently.

Interior-point Method

Consider the following convex optimization problem with includes inequality constraints such that:

$$\begin{aligned} \text{minimize} \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \leq 0, \quad i = 1, \ldots, m \\ & Ax = b \end{aligned} \tag{5.3}$$

It is a given that the functions $f_0,\ldots,f_m$: $\mathbb{R}^n \rightarrow \mathbb{R}$ are convex in nature and continuously differentiable twice. More so, $A \in \mathbb{R}^{p\times n}$, rank A $= p < n$ and an optimal solution $x^\star$ is assumed to exist with the optimal value defined as $f_0(x^\star) = p^\star$. In mathematical optimization, Karush-Kuhn-Tucker necessary conditions are introduced to ensure dual optimal solutions $\lambda^\star \in \mathbb{R}^m$ and $\nu^\star \in \mathbb{R}^p$:

$$\begin{gathered} Ax^\star = b, \quad f_i(x^\star) \leq 0, \quad i = 1, \ldots, m \\ \lambda^\star \geq 0 \\ \nabla f_0(x^\star) + \sum_{i=1}^{m} \lambda_i^\star \nabla f_i(x^\star) + A^T \nu^\star = 0 \\ \lambda_i^\star f_i(x^\star) = 0, \quad i = 1, \ldots, m \end{gathered} \tag{5.4}$$

The several options that exist within the Constrained Nonlinear Optimization Algorithms of Matlab to solve this problem will be utilized in the research. The benefit of this tool is the possibility to include limitless constraints to solve the guidance problem autonomously and with high fidelity (Szmuk and Açıkmeşe, 2018).

Pontryagin Maximum Principle

Many textbooks alternate between employing either the maximum or minimum principle of Pontryagin. This research adopts the maximum principle model in agreement with Açıkmeşe and Blackmore (2011). For the Pontryagin Maximum Principle the Hamiltonian H and adjoint equations $\dot{\boldsymbol{\lambda}}$ are defined:

$$H = -L[\mathbf{x}(t), \mathbf{u}(t), t] + \boldsymbol{\lambda}(t)^T f[\mathbf{x}(t), \mathbf{u}(t), t] \tag{5.5}$$

$$\dot{\boldsymbol{\lambda}}^T = -\frac{\partial H}{\partial \mathbf{x}} = \frac{\partial L}{\partial \mathbf{x}} - \boldsymbol{\lambda}^T \frac{\partial f}{\partial \mathbf{x}} \tag{5.6}$$

Optimal control $\mathbf{u}^\star$ must be appointed such that the Hamiltonian is maximized for $\mathbf{u} \in U$:

$$\mathbf{u}^\star = \arg[H(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{u}, t)] \tag{5.7}$$

Convex *vs* Nonconvex Functions

To gain an understanding regarding the difference between convex and nonconvex functions, the matter is briefly featured. As depicted in Fig. 5.1, a convex function includes one optimal solution. This solution is promptly the global solution of the problem and else the non-existence of a global solution proofs itself. For nonconvex problems on the other hand, several local optimal points exists and addressing which of these is the global optimum is computationally burdensome. This proofs that the efficiency of convex optimization is much higher. Furthermore, from a mathematical perspective, convex functions are easier to interpret. Finally, it was stated before that lossless convexification of a nonconvex optimization problem does not decrease solution quality. To relax the problem this research adapts the lossless convexification process.

5.2. Lossless Convexification

Pointedly, the optimization problem, initially adapted from Açıkmeşe and Ploen (2007), is formulated as a second-order cone programming problem (SOCP). This class of convex programming has the ability to optimally solve problems in polynomial time, implying that a converged global optimum is found within a deterministic upper bound on the number of iterations in real-time. In Section 5.2.1 the lossless convexification lemma is outlined for the Mars powered descent guidance based on a convex programming approach. Subsequently, the algorithm description is refined based on enhancements and a discretization process of the optimal control problem. In Sections 5.2.4 and 5.2.5, the algorithm is further expanded based on a minimum landing error definition after (Blackmore et al., 2010) and (Açıkmeşe et al., 2013), and parafoil aerodynamic incorporated guidance, respectively. The former also outlines the Golden Search Method for solving for optimal time-of-flight. This also marks the bridge to the next section on successive convexification.

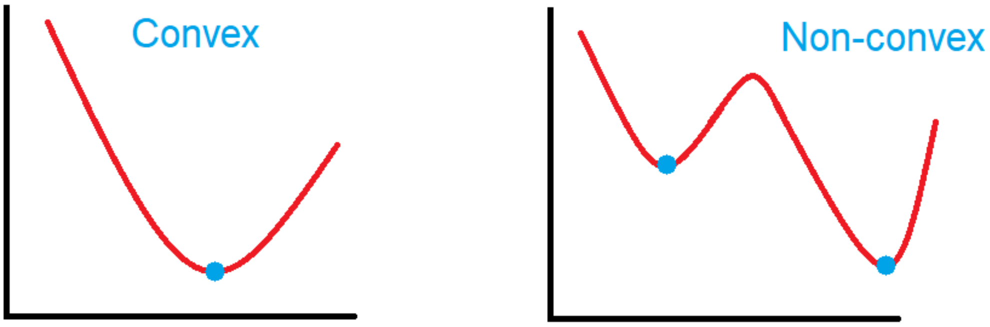


Figure 5.1: Graphical example of a convex function (left) and nonconvex function (right).

5.2.1. CONVEX PROGRAMMING

Recall the original guidance problem definition from Section 4.2, specifically Problem 1. The cost function was defined such that the control strain T_c was minimized over time as conferred in Eq. (5.8).

$$\max_{t_f,\mathbf{T}_c}(t_f) = \min_{t_f,\mathbf{T}_c} \int_0^{t_f} \|T_c(t)\| \,\mathrm{d}t \tag{5.8}$$

To deal with the non-convex nature of the problem emanating from the lower bound of the thrust control constraint a slack variable Γ is introduced, which replaces $\|T_c(t)\|$ in the cost function. Additionally, the mass flow dynamics is now defined as

$$\dot{m}(t) = -\alpha \,\|\Gamma(t)\| \tag{5.9}$$

The control authority inequality is redefined accordingly

$$0 < \rho_1 \le \|\Gamma(t)\| \le \rho_2 \tag{5.10}$$

After Açıkmeşe and Ploen (2007), the Hamiltonian of this redefined problem formulation is characterized as

$$H(x, m, T_c, \Gamma, \lambda_0, \lambda) = \lambda_0\Gamma + \lambda_1^T x_2 + \frac{\lambda_2^T T_c}{m} + \lambda_2^T g - \alpha\lambda_3\Gamma \tag{5.11}$$

Notice that x characterizes the state vector such that $x = [x_1^T, x_2^T]$, containing both position and velocity elements. Furthermore the costate vector $\lambda = [\lambda_1, \lambda_2, \lambda_3]^T \in \mathbb{R}^7$. Based on the earlier conferred Pontryagin's Maximum Principle, there must exist $\lambda_0 \le 0$ and a continuous function λ in vector form, such that the following three conditions must hold:

1. The adjoint function $[\lambda_0, \lambda(t)^T]^T \neq 0 \forall\, t \in [0, t_f^\star]$ and

$$\dot{\lambda} = \frac{\partial H}{\partial y}(y^\star, v^\star, \lambda_0, \lambda) \tag{5.12}$$

2. The Pointwise Maximum Principle

$$H(y^\star(t), v^\star(t), \lambda_0, \lambda) \ge H(y^\star(t), v, \lambda_0, \lambda) \quad \forall\, v \in \Psi \tag{5.13}$$

3. Transversality condition with $\Psi \equiv [{y^\star}^T, {v^\star}^T, \lambda_0, \lambda^T)]^T$ such that

$$[H(\Psi(0)), -\lambda(0)^T, -H(\Psi(t_f^\star)), \lambda(t_f^\star)^T]^T \quad \text{is orthogonal to vector} \quad [0, y^\star(0)^T, t_f^\star, y^\star(t_f^\star)^T] \tag{5.14}$$

From the first condition the following conclusions are derived:

$$\dot{\lambda}_1 = 0 \tag{5.15}$$

$$\dot{\lambda}_2 = \lambda_1 \tag{5.16}$$

$$\dot{\lambda}_3 = \frac{1}{m^2}\lambda_2^T T_c^\star \tag{5.17}$$

Given the boundary conditions entrenched in Problem 1, the following conditions are defined

$$\lambda_3(t_f^\star) = 0 \tag{5.18}$$

$$H(\Psi(t_f^\star)) = 0 \tag{5.19}$$

Since $\lambda_2(t) \neq 0$, $m^\star(t) > 0$ and the Hamiltonian H depends on T_c (in accordance with Eq. (5.11)), the pointwise principle implies that

$$\left\| T_c^\star(t) \right\| = \Gamma^\star(t) \quad \forall t \in [0, t_f^\star] \tag{5.20}$$

$$\rho_1 \leq \left\| T_c^\star(t) \right\| \leq \rho_2 \quad \forall t \in [0, t_f^\star] \tag{5.21}$$

The former implies that the optimal solution for the convexified problem corresponds to the optimal solution for the original nonconvex case. For the formal proof of the latter the reader is referred to Appendix B, Section B.1 specifically. Fig. 5.2 depicts the convexification process within the control space. The shaded region in the upper figure depicts the nonconvex constraint set, which through the slack variable is convexified to the cone presented in the lower figure.

The remainder of this section is dedicated to the proof of the existence of an optimal solution within the convex domain. Based on Section 5.1, consider the following general optimal control problem

$$\begin{gathered} \min J(y(\cdot), u(\cdot), t_f) = \int_0^{t_f} g(y(t), u(t))dt \\ \text{subject to} \, \dot{y}(t) = f(y(t), v(t)) \end{gathered} \tag{5.22}$$

Functions f and g are of continuous nature on $\mathbb{Y} \times \mathbb{V}$. For each t and $y \in \mathbb{R}, \mathbb{Q}^+(t, y)$ must be convex, where

$$\mathbb{Q}^+(t) = \left\{ (z_1, z_2) : z_1 \geq g(y, v), z_2 = f(y, v), v \in \mathbb{U} \right\} \tag{5.23}$$

$$\begin{gathered} \mathbb{Q}^+(x, m) = (z_1, z_2, z_3, z_4), \quad z_1 \quad \text{and} \quad z_4 \in \mathbb{R}, \quad z_2 \quad \text{and} \quad z_3 \in \mathbb{R}^3 \\ \geq \Gamma, (z_2, z_3, z_4) = (x_2, g + T_c/m, -\alpha\Gamma), \rho_1 \leq \Gamma \leq \rho_2, \| T_c \| \leq \Gamma \end{gathered} \tag{5.24}$$

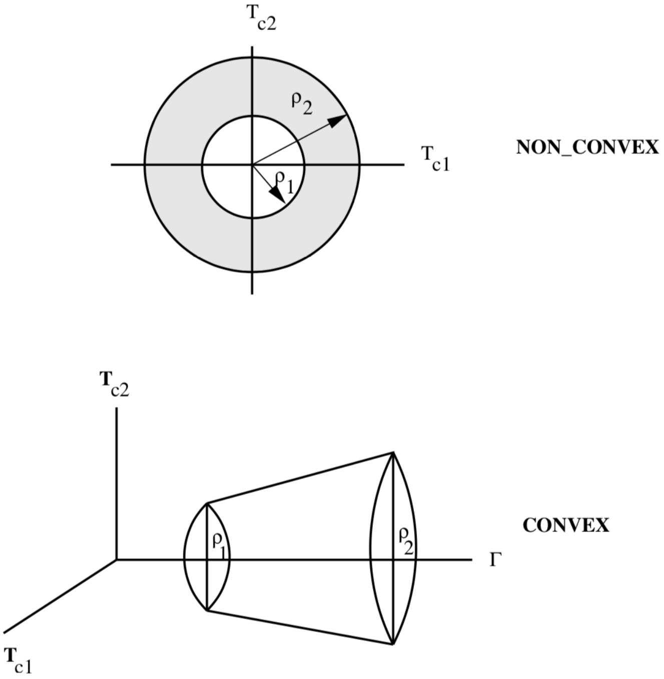


Figure 5.2: Two-dimensional and three-dimensional representation of the convexified control space (Açıkmeşe and Ploen, 2007).

Combining $\mathbb{Q}^+(t)$ and $\mathbb{Q}^+(x,m)$ yields $\mathbb{Q}^+(t,x,m)$. This implies that $(z_1,z_2,z_3,z_4) \in \mathbb{Q}^+(t,x,m)$ is only true if the following conditions are met

$$\begin{aligned} z_1 &\geq -z4/\alpha \\ z_4 &= -\alpha\Gamma \\ z_3 &= g + T_c/m \\ z_2 &= x_2 \\ -z_4 &\geq \alpha \|T_c\| \\ \alpha \|(z_2-g)m\| &\leq -z_4 \\ \|T_c\| &\leq \Gamma \\ \rho_1 &\leq \Gamma \leq \rho_2 \end{aligned} \tag{5.25}$$

From the above equalities and inequalities the following conclusions are drawn:

- All equalities are linear
- All inequalities are linear or second-order cone constraints

This implies that $\mathbb{Q}^+(t,x,m)$ is a convex set. This also implies that the powered descent guidance problem has an optimal and feasible solution, which concludes the formal proof of the theorem.

5.2.2. ENHANCEMENTS

For the construction of the numerical algorithm that solves the powered descent problem, a change of variable methodology is introduced for the purpose of efficiency. Notice that this entails that the optimal problem is still formulated in the continuous domain. The discretization of the algorithm will follow later in this section on lossless convexification. The method described in this section is based on Açıkmeşe and Ploen (2007) and Açıkmeşe et al. (2008). First two change of variables are defined such that

$$\sigma \equiv \frac{\Gamma}{m} \quad \text{and} \quad u \equiv \frac{T_c}{m} \tag{5.26}$$

Consequently, the dynamic equations of Problem 1 are re-written to the following from

$$\ddot{r}(t) = u(t) + g \tag{5.27}$$

$$\frac{\dot{m}(t)}{m(t)} = -\alpha\sigma(t) \tag{5.28}$$

Re-arranging this differential equation yields

$$m(t) = m_0 e^{-\alpha \int_0^{t_f} \sigma(\tau) d\tau} \tag{5.29}$$

The mass flow rate $\alpha > 0$ and thus minimizing fuel consumption is equivalent to minimizing $\int_0^{t_f} \sigma(\tau) d\tau$. The constraints on the control space in terms of the new variables becomes

$$\|u(t)\| \leq \sigma(t) \quad \forall t \in [0, t_f] \tag{5.30}$$

$$\frac{\rho_1}{m(t)} \leq \sigma(t) \leq \frac{\rho_2}{m(t)} \quad \forall t \in [0, t_f] \tag{5.31}$$

What becomes evident from the above equations that the inequalities are within the convex control space. However, the mass variable m causes these inequalities to have nonlinear nature, making them nonconvex in nature. To resolve this matter the inequalities are to convexified. This process is initiates by defining

$$z \equiv \ln(m) \tag{5.32}$$

Such that mass differential equation conferred earlier becomes

$$\dot{z}(t) = -\alpha\sigma(t) \tag{5.33}$$

And the inequality now yields

$$\rho_1 e^{-z(t)} \leq \sigma(t) \leq \rho_2 e^{-z(t)} \quad \forall t \in [0, t_f] \tag{5.34}$$

Within this formulation the first segment of the inequality is convex in nature, while the second segment is not. By utilizing a Taylor series expansion of e^{-z} this matter is resolved, for which the proof is provided in Appendix B.2. The first part is approximated by means of a second-order cone, using three expansion terms, such that

$$\rho_1 e^{-z_0}[1-(z-z_0)+\frac{(z-z_0)^2}{2}] \leq \sigma \tag{5.35}$$

In which $z_0(t)$ is a given computed through $z_0(t) = \ln(m_{\text{wet}} - \alpha\rho_2 t)$. The latter part approximated by means of a linear method, using two expansion terms, such that

$$\sigma \leq \rho_2 e^{-z_0}[1-(z-z_0)] \tag{5.36}$$

By defining $\mu_1 \equiv \rho_1 e^{-z0}$ and $\mu_2 \equiv \rho_2 e^{-z0}$, the total convexified inequality is $\forall\, t \in [0, t_f]$ conferred as

$$\mu_1(t)[1-[z(t)-z_0(t)+\frac{[z(t)-z_0(t)]^2}{2}] \leq \sigma(t) \leq \mu_2(t)1-[z(t)-z_0(t)] \tag{5.37}$$

Additional constraints are posed on the mass of the spacecraft to ensure the physical limits are not violated throughout the landing sequence. These are enforced through

$$\ln(m_{\text{wet}} - \alpha\rho_2 t) \leq z(t) \leq \ln(m_{\text{wet}} - \alpha\rho_1 t) \tag{5.38}$$

Summarized, the convexified optimal control powered descent guidance problem is provided as

Problem 7: Convexified powered descent guidance problem

Cost Function:

$$\min_{t_f, u(\cdot), \sigma(\cdot)} \int_0^{t_f} \sigma(t)dt \qquad \text{subject to:}$$

Dynamics:

$$\dot{\mathbf{r}}(t) = u(t) + g$$
$$\dot{z}(t) = -\alpha\sigma(t)$$

State Constraints:

$$z_0(t) \leq z(t) \leq \ln(m_{\text{wet}} - \alpha\rho_1 t)$$

Control Constraints:

$$\|u(t)\| \leq \sigma(t)$$
$$\mu_1(t)[1-[z(t)-z_0(t)+\frac{[z(t)-z_0(t)]^2}{2}] \leq \sigma(t) \leq \mu_2(t)1-[z(t)-z_0(t)]$$

Boundary Conditions:

$$z_0(t) = \ln(m_{\text{wet}} - \alpha\rho_2 t), \quad \mu_1(t) = \rho_1 e^{-z_0(t)}, \quad \mu_2(t) = \rho_2 e^{-z_0(t)}$$
$$z(0) = \ln(m_{\text{wet}}), \quad \mathbf{r}(0) = \mathbf{r}_0, \quad \dot{\mathbf{r}}(0) = \dot{\mathbf{r}}_0, \quad \mathbf{r}(t_f) = \dot{\mathbf{r}}(t_f) = \mathbf{0}$$

5.2.3. Discretization

The next and final step within the lossless convexification process is the discretization of the optimal control problem. A more comprehensive outline has been included in Appendix B.3. The discretization process transfers the problem from the infinite dimensional to the finite dimensional time domain with equal intervals between the so-called temporal nodes, given as

$$t_k = k\Delta t, \quad k = 0, \ldots, N \tag{5.39}$$

Some conservative physical bounds can de derived directly from the dynamics. When defining the time bound as

$$t_{\min} \leq t_f \leq t_{\max} \tag{5.40}$$

The limitations of fuel and thrust levels impose the following bounds

$$t_{\min} = \frac{(m_{\text{wet}} - m_{\text{fuel}})\, \|\dot{r}(0)\|}{\rho_2} \quad \text{and} \quad t_{\max} = \frac{m_{\text{fuel}}}{\alpha \rho_1} \tag{5.41}$$

For the discretized scheme this implies bounds on N such that

$$N_{\min} \leq N \leq N_{\max} \tag{5.42}$$

which leads to the rounded integer bounds computed through

$$N_{\min} = \text{round}(\frac{t_{\min}}{\Delta t} + 1) \quad \text{and} \quad N_{\max} = \text{round}(\frac{t_{\max}}{\Delta t} + 1) \tag{5.43}$$

The discretization has a piece-wise linear control input such that

$$u(t) = u_k + (u_{k+1} - u_k)\tau \tag{5.44}$$

$$\sigma(t) = \sigma_k + (\sigma_{k+1} - \sigma_k)\tau \tag{5.45}$$

in which τ is computed through

$$\tau = \frac{t - t_k}{\Delta t} \qquad \forall t \in [t_k, t_{k+1}) \tag{5.46}$$

This discretization scheme results in a finite dimension second-order cone program within a specified time t_f. The total problem definition is presented below

Problem 8: Discretized and convexified powered descent guidance problem

Cost Function:

$$\min_{u_0,\ldots,u_N,\sigma_0,\ldots\sigma_N} -z_N \quad \forall k = 0,\ldots,N \qquad \text{subject to:}$$

Dynamics:

$$\begin{aligned} r_{k+1} &= r_k + \frac{\Delta t}{2}(\dot{r}_k + \dot{r}_{k+1}) + \frac{\Delta t^2}{12}(u_{k+1} - u_k) \\ \dot{r}_{k+1} &= \dot{r}_k + \frac{\Delta t}{2}(u_k + u_{k+1}) + g\Delta t \\ z_{k+1} &= z_k - \frac{\alpha \Delta t}{2}(\sigma_k + \sigma_{k+1}) \end{aligned}$$

State Constraints:

$$z_{0,k} \leq z_k \leq \ln(m_{\text{wet}} - \alpha \rho_1 t)$$

Control Constraints:

$$\begin{aligned} \|u_k\| &\leq \sigma_k \\ \mu_{1,k}[1 - [z_k - z_{0,k} + \frac{[z_k - z_{0,k}]^2}{2}] &\leq \sigma_k \leq \mu_{2,k} 1 - [z_k - z_{0,k}] \end{aligned}$$

Boundary Conditions:

$$z_0(t) = \ln(m_{\text{wet}} - \alpha \rho_2 t), \quad \mu_1(t) = \rho_1 e^{-z_0(t)}, \quad \mu_2(t) = \rho_2 e^{-z_0(t)}$$

$$z(0) = \ln(m_{\text{wet}}), \quad \mathbf{r}(0) = \mathbf{r}_0, \quad \dot{\mathbf{r}}(0) = \dot{\mathbf{r}}_0, \quad \mathbf{r}(t_f) = \dot{\mathbf{r}}(t_f) = \mathbf{0}$$

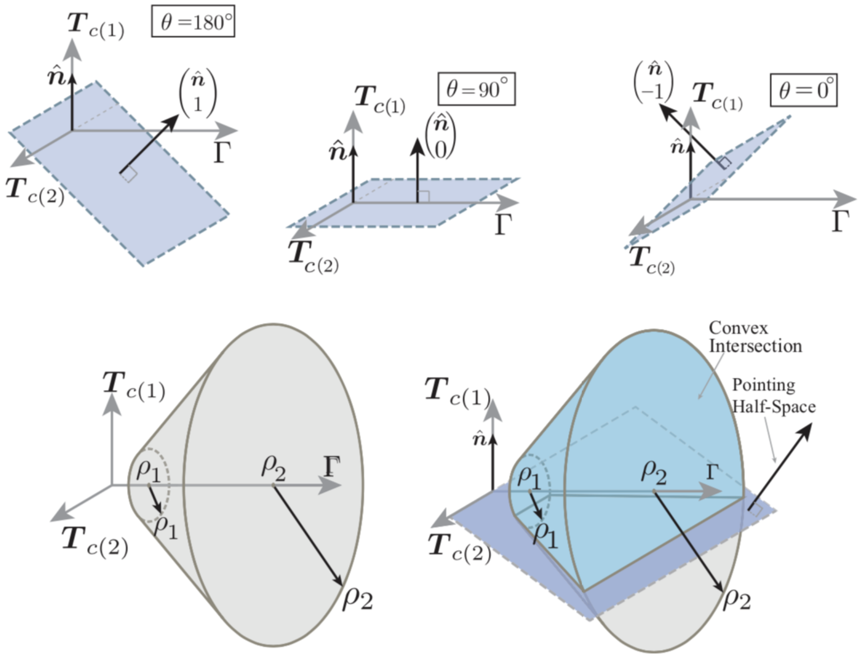


Figure 5.3: Half-space orientation of the relaxed pointing constraint (Açıkmeşe et al., 2013).

Pointing Constraint Relaxation

The soft landing problem belongs to the class of nonlinear optimal control, which in is formally extended to hold for the constraint of pointing thrust. Based on the prior established relaxed problem the following convex constraints are defined

1. Thrust upper bound: $\|\mathbf{T}_c(t)\| \leq \Gamma(t)$
2. Slack variable bounds: $\rho_1 \leq \Gamma(t) \leq \rho_2$
3. Pointing constraint: $\hat{\mathbf{n}}\mathbf{T}_c(t) \geq \Gamma(t)\cos\theta$

Within the $\mathbf{T}_c - \Gamma$ space, the relaxed pointing constraint forms a half-space of values of thrust, denoted by $(\hat{\mathbf{n}}, -\cos\theta)$. In Fig. 5.3 the half-space constraint is depicted for several θ values. It also depicts that the relaxed set of the pointing constraint is convex in nature. In the next section the algorithm is formulated as a minimum landing error problem, with a coupled algorithm to solve for optimal time.

5.2.4. Minimum Landing Error and Optimal Flight-time

In this section the convexified algorithm for guaranteeing a minimum landing error within the landing scenario is established. The discussion in this segment of the section is associated to the line search method for finding the optimal time-of-flight, defined as the golden search technique. For the nonconvex counterpart of the powered descent problem definition the reader is referred to Section 4.2.2. The main idea behind the minimum landing error algorithm is to first that the distance between the predefined target and the physically feasible landing site is minimized. For this purpose the following cost function is defined

$$\min_{t_f, \mathbf{T}_c(\cdot), \Gamma(\cdot)} \left\|\mathbf{r}(t_f)\right\| \tag{5.47}$$

It is denoted that the final result of the minimum landing error is a certain distance to target D. This algorithm hence first guarantees the feasibility of the computed trajectory. Subsequently, based on the computed trajectory the minimum fuel guidance problem, discussed in the previous section, is solved. Instead of restricting the terminal position at zero, the constrained is relaxed to be

$$\left\|\mathbf{r}(t_f)\right\| \leq D \tag{5.48}$$

Solving the minimum landing error problem requires three important steps. The first two are the change of variables process and the time discretization, discussed in Sections 5.2.2 and 5.39, respectively. Without repeating the derivation outlined in these previous sections, the entire algorithm outline is provided

Problem 9: Discretized and convexified minimum landing error problem

Cost Function:

$$\min_{t_f, \mathbf{T}_c(\cdot), \Gamma(\cdot)} \left\| \mathbf{r}(t_f) \right\| \qquad \text{subject to:}$$

Dynamics:

$$\begin{aligned} r_{k+1} &= r_k + \frac{\Delta t}{2}(\dot{r}_k + \dot{r}_{k+1}) + \frac{\Delta t^2}{12}(u_{k+1} - u_k) \\ \dot{r}_{k+1} &= \dot{r}_k + \frac{\Delta t}{2}(u_k + u_{k+1}) + g\Delta t \\ z_{k+1} &= z_k - \frac{\alpha \Delta t}{2}(\sigma_k + \sigma_{k+1}) \end{aligned}$$

State Constraints:

$$z_{0,k} \leq z_k \leq \ln(m_{\text{wet}} - \alpha \rho_1 t)$$

Control Constraints:

$$\begin{aligned} \| u_k \| \leq & \sigma_k \\ \mu_{1,k}[1 - [z_k - z_{0,k} + \frac{[z_k - z_{0,k}]^2}{2}] \leq & \sigma_k \leq \mu_{2,k} 1 - [z_k - z_{0,k}] \end{aligned}$$

Boundary Conditions:

$$\begin{gathered} z_0(t) = \ln(m_{\text{wet}} - \alpha \rho_2 t), \quad \mu_1(t) = \rho_1 e^{-z_0(t)}, \quad \mu_2(t) = \rho_2 e^{-z_0(t)} \\ z(0) = \ln(m_{\text{wet}}), \quad \mathbf{r}(0) = \mathbf{r}_0, \quad \dot{\mathbf{r}}(0) = \dot{\mathbf{r}}_0, \quad \mathbf{r}(t_f) = \dot{\mathbf{r}}(t_f) = \mathbf{0} \end{gathered}$$

GOLDEN SEARCH METHOD

The third and last step required is a line search method to obtain the optimal time-of-flight t_f, a technique on which the remainder of this section focuses. The technique makes sure that the interval N_{min} and N_{max} is shortened within each iteration by a ratio known as the golden ratio. The powerful technique is possible as it is known beforehand within which domain the optimal solution can be found. The golden search method, also referred to as the bisection method, is rather straightforward. As depicted in Fig. 5.4, two points are picked within a certain pre-defined interval $[a, b]$. For the landing problem this interval corresponds to the time-of-flight domain. For these points the algorithm is solved, which in general terms leads to

$$f(M^-) \quad \text{and} \quad f(M^+) \tag{5.49}$$

As can be deduced from the figure, $f(M^-)$ and $f(M^+)$ yield

$$f(M^-) = f(\frac{a+b}{2} - \frac{\epsilon}{2}) \quad \text{and} \quad f(M^+) = f(\frac{a+b}{2} + \frac{\epsilon}{2}) \tag{5.50}$$

Notice that this means that the algorithm needs to be computed twice and the outcome of interest, which in this case is the fuel mass, is compared with one another. Two possible scenario's emerge:

1. $f(M^-) > f(M^+)$
2. $f(M^-) < f(M^+)$

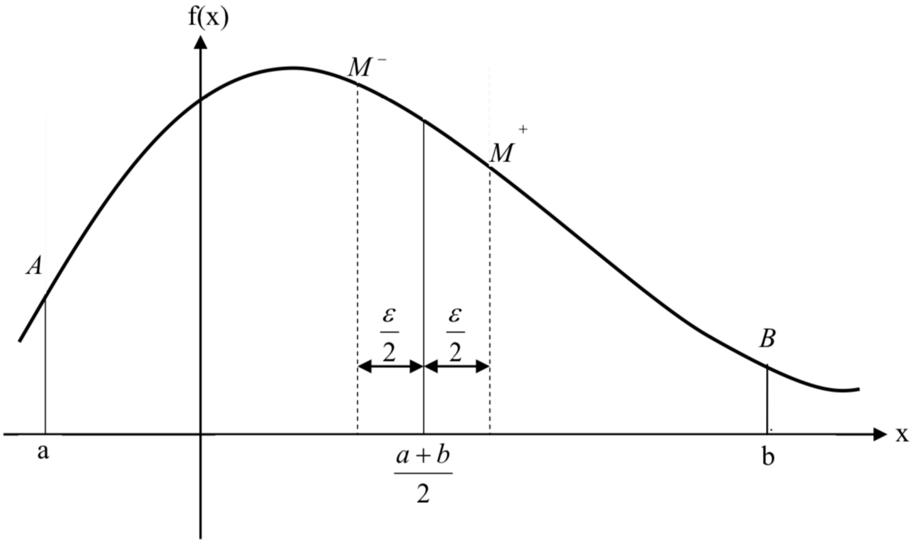


Figure 5.4: Golden search method with equal intervals [1]

Scenario 1 leads to the conclusion that the optimal solution lays on interval $[M^-, b]$, while scenario 2 leads to a solution on interval $[a, M^-]$. So far in the discussion the test points M^- and M^+ are chosen arbitrarily. These need to be chosen wisely as they determine the computation efficiency of the algorithm. A proven way to save time of computation is basing these points on the special properties of the golden ratio ϕ, which according to the definition equals:

$$\phi = \frac{1+\sqrt{5}}{2} \tag{5.51}$$

More specifically, the computation time is decreased through the relation

$$\frac{1}{\phi^2} = 1 - \frac{1}{\phi} \tag{5.52}$$

Based here-upon the starting points M^- and M^+ are defined such that

$$M^- = b - \frac{b-a}{\phi} \quad \text{and} \quad M^+ = a + \frac{b-a}{\phi} \tag{5.53}$$

This computation is repeated till convergence within a certain pre-defined threshold value is obtained.

5.2.5. PARAFOIL CONVEX GUIDANCE

With the formal proof and mathematical principles being outlined in the previous sections, this segment of the chapter focuses on the extension of the algorithm towards the compatibility for landing a parafoil. Distinct from powered descent guidance, the control authority is now provided through the electric powered pull of the landing canopy. The associated dynamics has been captivated in Section 3.6. The algorithm has been formulated as a minimum control effort problem such that the control varies between

$$u_{\text{min}} \leq u(t) \leq u_{\text{max}} \tag{5.54}$$

Since the control effort $u(t)$ is a normalized entity over time, it is known beforehand that the maximum value $u_{\text{max}} = 1$. Nonetheless, one cannot say per definition that the minimum value $u_{\text{min}} = 0$, as this would require a scenario with no presence of active aerodynamic forces at all. Given the mission scenario, it is excluded to define this beforehand. This is especially true as the parafoil will require high angle-of-attack profile to counter the gravitational pull of planetary body and the emerging wind gusts. Therefore, similar to the powered descent cases delineated earlier in this chapter, the parafoil guidance problem nonconvex in nature. Distinct from the powered descent case, the cost function to be minimized is now defined as

$$\min_t \|\mathbf{u}(t)\| \tag{5.55}$$

[1] After http://mathforcollege.com/nm/mws/gen/09opt/mws_gen_opt_txt_goldensearch.pdf

Likewise, introducing a slack variable resolves the issue of nonconvex optimal control, such that

$$u_{\min} \leq \Gamma_u \leq u_{\max} \tag{5.56}$$

Even tough the mass of a parafoil is constant over time, apparent forces cause disturbances within this assumption, making mass yet a function of time. Before continuing the discussion on the guidance problem, the definition and mathematical principles of apparent masses and inertias are outlined. Notice that no full derivations are provided. For that the reader is referred to the extensive paper on parafoil apparent physics by Lissaman and Brown (1993). Any body moving within a fluid causes that particular fluid to be set into motion which generates a field of pressure about the body. This field is what is known as apparent mass pressure, a ratio between the mass of the vehicle and the shifted mass of air. For an aircraft this phenomena is often neglected, however, for a parafoil it heavily influences the ram-air dynamics characteristics. This ratio is approximated through

$$M_r = \frac{m}{\rho S^{2/3}} \tag{5.57}$$

in which S is the vehicle surface reference area and ρ is the density computed using the following relation obtained from Quadrelli et al. (2019), known as the Yelle model

$$\rho = 5.43 e^{\frac{-0.512}{1000} h} \tag{5.58}$$

Notice that the density is a function of altitude, which is a function of time. A ratio M_r in the order of 0.8 is typical for Earth-like environments, while a parafoil within the Titan environment typically has values of 7. In Figs. 5.5 and 5.6 a geometric visualization is provided of an inflated canopy with apparent masses and inertias. After (Lissaman and Brown, 1993), the representation is defined by a platform of rectangular shape with span b and chord c. The airfoil is assumed to have thickness t, and the arch of the canopy is defined by dimension a. It is noted that the actual shape of the parafoil can be modelled using sophisticated using Computational Fluid Dynamics methods. This is however beyond the scope of both the paper and this thesis. The reason for that is the fact that the adapted method already provides useful accuracy for apparent mass modelling. Keep in mind that the computed mass and inertia are not the real dynamic fluid properties but a representation of the transported energy due acceleration, causing additional work which increases the velocity Kowaleczko (2014). The tensors of the apparent mass and inertia are defined as

$$\mathbf{M}_f = \begin{bmatrix} A & 0 & 0 \\ 0 & B & 0 \\ 0 & 0 & C \end{bmatrix} \quad \text{and} \quad \mathbf{I}_f = \begin{bmatrix} I_A & 0 & 0 \\ 0 & I_B & 0 \\ 0 & 0 & I_C \end{bmatrix} \tag{5.59}$$

The individual components of these governing equations are computed by means of the two set of equations given in Eq. (5.60) and (5.61). In these equations AR is the aspect ratio, b the wingspan of the parafoil, c the parafoil chord, h the parafoil height and t its thickness. Furthermore, the equations are related through

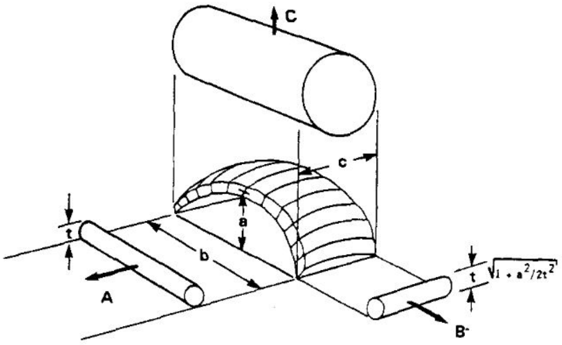


Figure 5.5: Depiction of the apparent mass due to a parafoil in an aerodynamic environment (Lissaman and Brown, 1993).

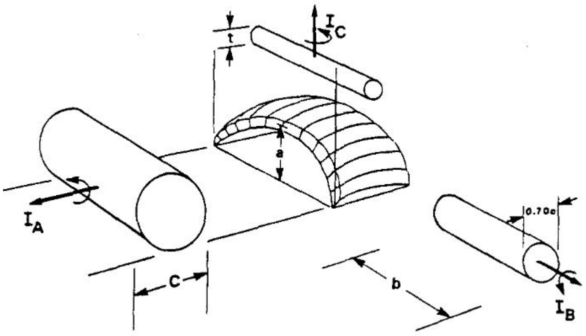


Figure 5.6: Depiction of the apparent inertia due to a parafoil in an aerodynamic environment (Lissaman and Brown, 1993).

$AR = b/c$, $t_\star = t/c$ and $a_\star = a/b$, such that the following relations are established

$$\begin{aligned}
A &= 0.667\rho(1+\frac{8}{3}a_\star^2)t^2 b \\
B &= 0.267\rho(1+2\frac{a_\star^2}{t_\star^2}AR(1-t_\star^2)t^2 c \\
C &= 0.785\rho\sqrt{1+2+a_\star(1-t_\star)}\frac{AR}{1+AR}c^2 b
\end{aligned} \tag{5.60}$$

$$\begin{aligned}
I_A &= 0.055\rho\frac{AR}{1+AR}c^2 b^3 \\
I_B &= 0.0308\rho\frac{AR}{1+AR}[1+\frac{\pi}{6}(1+AR)ARa_\star^2 t_\star^2]c^4 b \\
I_C &= 0.0555\rho(1+8a_\star^2)t^2 b^3
\end{aligned} \tag{5.61}$$

For a detailed outline of the coefficients and factors to model the influence of apparent mass and inertia is provided by Kowaleczko (2014). This includes a contribution analysis, with the conclusion that the apparent mass in z-direction is the greatest contributor. Specifically, the same order of magnitude as the entire canopy. This means the contribution is still significantly lower in order of magnitude compared to the payload, nonetheless, significant enough to be considered. Furthermore, the apparent inertia is largest in y-direction for short suspension lines (about 5 m), and for long suspension lines (about 10 m) in x-direction. The contribution of in the apparent inertia is likewise in the same order of magnitude as the canopy.

Now that the matter on varying mass has been outlined, the discussion on the nonconvex lower bound of the pull control is continued. Just like with the lower bound on the thrust magnitude due to the fact that the propulsion system cannot be turned off once ignited, the parafoil cannot be guaranteed to have a zero control magnitude. It is important to denote however that this physically is possible. To resolve to issue and make sure the computation encompasses a full landing scenario, mass is considered to be time dependent and thus the differential equation in Eq. (5.32) likewise holds. Through a process of changing variables and the introduction of the slack variable, the minimum control effort is formulated. From the time history of the control effort, the total required power can be derived, a discussion beyond the scope of this thesis.

5.2.6. Advantages and Disadvantages

To conclude the chapter on lossless convexification, some the advantages and disadvantages coupled to this method are conferred. The discussion in this section also marks the bridge to the next section on successive convexification. The main **advantages** of the lossless convexification method are:

- No loss of generality (i.e., guarantee of an optimal solution). This is especially the case when formulating the descent algorithm as a minimum landing error problem.
- The soft landing is established, meaning that zero position error and zero touchdown velocity are both achieved in case a feasible trajectory exists. If not, the minimum landing error guidance algorithm ensures landing as close as possible to the predefined target.
- The lossless convexification guidance algorithm has been verified and validated extensively. The same working principle of the algorithm is applied during the booster recovery of the Falcon 9 (precision landing sequence). Furthermore, as described by Açıkmeşe et al. (2014), the algorithm has been applied during the MSL mission as part of the fly-away algorithm during terminal descent. Conclusively, as described in Chapter 4, the G-FOLD algorithm has been tested extensively by JPL and Masten Aerospace.
- The algorithm is executable in real-time when programmed in high level languages (e.g. C++), although this is beyond the scope of this research, where a Matlab implementation will be used. Table 5.1 depicts the validated CPU time for the earlier conferred G-FOLD algorithm. The algorithm is theoretically capable to run 8-10 SOCP iterations for each trajectory in a matter of 10 ms in C.

While the above mentioned points are all valid, there some minor downsides to the lossless convexification method. These **disadvantages** are summarized below:

- The golden search method is needed to solve for the optimal time-of-flight problem. The disadvantage of the golden search method is that its properties do not guarantee theoretical convergence towards an optimal solution.

Table 5.1: Preparing for interplanetary precision landing, on-board application feasibility after (Açıkmeşe, 2017).

		Monte-Carlo Analysis	Flight Implementation
Method	**NLP-based**	**Generic IPM for SOCP in M**	**Custom IPM for SOCP in C**
CPU time (ms) on a laptop	20000	3000	10
Reliability	<80%	>99%	>99%

- The non-linear nature of the equations of motion make it difficult to captivate the full dynamics within the algorithm. The applied change of variables is possible when considering 3DoF dynamics, but for the highly nonlinear 6DoF dynamics another methodology needs to be applied.

To ensure that finding the optimal time problem is resolved, as well as the handling of the non-linear 6DoF dynamics, the newest methodology coupled to this algorithm has been exploited. While it works on the same principle as the lossless convexification method, the successive method approaches the problem in a slightly different manner. This method guarantees convergence towards an optimal solution within several iterations. For that reason the method has also been adapted in this research and it the main mathematical principles are explored in the next section.

5.3. SUCCESSIVE CONVEXIFICATION

The discussion in the previous section on generating a reference trajectory for the powered and parafoil descent phase assumed a 3 DoF analysis. In this section the successive convexification method it outlined, which in an extension of the previous discussion. The discussion is initiated with an outline on the linearization and discretization process. Following up, the trust regions and virtual control parameter are discussed. Additionally, the final version of the rigid body 6DoF terminal guidance formulation in discrete time is presented, which marks the end of the section on successive convexification.

5.3.1. LINEARIZATION AND DISCRETIZATION

This section initiates with outlining general linearization theory after which this is directly applied to the focus system of this research, namely, the six-degree-of freedom successive convexification powered descent guidance problem. Such linearizations, as shown in the general formulation, are often conducted using Taylor series approximations. Through the linearization operation it becomes possible to investigate the characteristic open-loop behaviour and natural stability of the spacecraft.

MATHEMATICAL FORMULATION OF LINEARIZATION

From a pure mathematical perspective, linearization is defined as linearly approximating a given function at a given point. These mathematical techniques are applied to assess the local stability of nonlinear differential equations or dynamical systems that are discrete. The most convenient way of doing this is by utilizing the complex conjugate pairs eigenvalues of the Jacobian matrix at a certain point of equilibrium. Positive eigenvalues are indicative of unstable equilibrium while negative eigenvalues are, as can be guessed, stable. Eigenvalues mixed in sign gives rise to the possibility for a saddle point to exist. For the approximations of nonlinear systems, i.e., linearization, Taylor series are the widely used mathematical techniques and those are also adapted in this research. Taylor series for a given infinitely differentiable function $f(x)$ near a given number a is denoted as:

$$\sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n \tag{5.62}$$

Commonly, the above equation is not often utilized because it is more convenient to approximate a given function using a finite number of series. If a function is considered with a state of dimensions n and the linearization is about $\mathbf{x} = \mathbf{x}_0$, the linearized function reads:

$$y = f(\mathbf{x}_0) + f_{x_1}(\mathbf{x}_0)\Delta x_1 + f_{x_2}(\mathbf{x}_0)\Delta x_2 + \ldots + f_{x_n}(\mathbf{x}_0)\Delta x_n \tag{5.63}$$

$$= f(\mathbf{x}_0) + \sum_{i=1}^{n} f_{x_i}(\mathbf{x}_0)\Delta x_i \tag{5.64}$$

The result of the linearization using first order Taylor series expansion forms a set of coupled differential equations in state space form:

$$\Delta\dot{\mathbf{x}} = \mathbf{A}\Delta\mathbf{x} + \mathbf{B}\Delta\mathbf{u} \tag{5.65}$$

In this equation the matrices **A** and **B** define the system and input, respectively. These can be derived straight from the nonlinear EoM:

$$\mathbf{A} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \vdots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \cdots & \frac{\partial f_1}{\partial u_n} \\ \vdots & \vdots & \vdots \\ \frac{\partial f_n}{\partial u_1} & \cdots & \frac{\partial f_n}{\partial u_n} \end{bmatrix} \tag{5.66}$$

Notice that the functions f_n represent the various governing equations of motion that are applied by the guidance system to assure planetary precision landing through ruling dynamics. In the subsequent section the linearization is applied to the 6 DoF EoMs for powered descent convex optimization guidance.

Descent Guidance Linearization

In this section linearization theory is applied to the convex problem that has been defined in Section 4.2. The discussion is initiated with defining the state vector $\mathbf{x}(t) \in \mathbb{R}^{14}$ and control vector $\mathbf{u}(t) \in \mathbb{R}^{3}$:

$$\mathbf{x}(t) = \begin{bmatrix} m(t) & \mathbf{r}_G^T(t) & \mathbf{V}_G^T(t) & \mathbf{q}_{B,G}^T(t) & \boldsymbol{\omega}_B^T(t) \end{bmatrix}^T \tag{5.67}$$

$$\mathbf{u}(t) = \mathbf{T}_B(t) \tag{5.68}$$

Recall the dynamics and kinematics defined in Eqs. (4.2), (4.31), (4.32) and (4.34), now expressed as a nonlinear vector function $f : \mathbb{R}^{14} \times \mathbb{R}^{3} \rightarrow \mathbb{R}^{14}$ based on state and control vector:

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)) = \begin{bmatrix} \dot{m}(t) & \dot{\mathbf{r}}_G^T(t) & \dot{\mathbf{V}}_G^T(t) & \dot{\mathbf{q}}_{B,G}^T(t) & \dot{\boldsymbol{\omega}}_B^T(t) \end{bmatrix}^T \tag{5.69}$$

The next-step is to define the above expression in terms of normalized time $\tau \in [0,1]$ applying a chain rule:

$$\frac{d}{dt}\mathbf{x}(t) = \frac{d\tau}{dt}\frac{d}{d\tau}\mathbf{x}(t) \tag{5.70}$$

To ease the notation, a dilation coefficient $\sigma = \frac{dt}{d\tau}$ is introduced such that the vector function may be written in terms of normalized time and this dilation coefficient:

$$\mathbf{x}'(t) = \frac{d}{d\tau}\mathbf{x}(\tau) = \sigma f(\mathbf{x}(\tau), \mathbf{u}(\tau)) \tag{5.71}$$

It is the right-hand side of this equation that is approximated using Taylor series of first order. For a certain reference trajectory, the state, control and dilation coefficient are defined as $\hat{\mathbf{x}}(\tau)$, $\hat{\mathbf{u}}(\tau)$ and $\hat{\sigma}$), respectively. At a later stage of the discussion these quantities are construed. The complete linearized convex formulation normalized with respect to time reads:

$$\mathbf{x}'(t) = \mathbf{A}(\tau)\mathbf{x}(\tau) + \mathbf{B}(\tau)\mathbf{u}(\tau) + \sum(\tau)\sigma + \mathbf{z}(\tau) \tag{5.72}$$

$$\mathbf{A}(\tau) = \hat{\sigma} \cdot \frac{\partial}{\partial \mathbf{x}} f(\mathbf{x}, \mathbf{u})\Big|_{\hat{\mathbf{x}}(\tau), \hat{\mathbf{u}}(\tau)} \tag{5.73}$$

$$\mathbf{B}(\tau) = \hat{\sigma} \cdot \frac{\partial}{\partial \mathbf{u}} f(\mathbf{x}, \mathbf{u})\Big|_{\hat{\mathbf{x}}(\tau), \hat{\mathbf{u}}(\tau)} \tag{5.74}$$

$$\sum(\tau) = f\left(\hat{\mathbf{x}}(\tau), \hat{\mathbf{u}}(\tau)\right) \tag{5.75}$$

$$\mathbf{z}(\tau) = -\mathbf{A}(\tau)\hat{\mathbf{x}}(\tau) - \mathbf{B}(\tau)\hat{\mathbf{u}}(\tau) \tag{5.76}$$

After linearizing the governing EoM, the derivation is proceeded by linearizing the lower bound constraint of the trust as that is the only remaining nonconvex source. For this a new function g is defined such that $g : \mathbb{R}^3 \rightarrow \mathbb{R}$:

$$g(\mathbf{u}(\tau)) = T_{\min} - ||\mathbf{u}(\tau)||_2 \leq 0 \tag{5.77}$$

Applying a first order Taylor series to this time normalized constraint yields:

$$T_{\min} \leq \mathbf{B}_g(\tau)\mathbf{u}(\tau) \tag{5.78}$$

$$\mathbf{B}_g(\tau) = \frac{\hat{\mathbf{u}}^T(\tau)}{||\hat{\mathbf{u}}(\tau)||_2} \tag{5.79}$$

Discretization

The rest of this section discusses the discretization, again with respect to normalized time, of the optimization problem that is needed parameterize the continuous time elements into finite dimensions. The following two sets are defined:

$$K = \{0, 1, ..., K-2, K-1\} \tag{5.80}$$

$$\bar{K} = \{0, 1, ..., K-3, K-2\} \tag{5.81}$$

For the time an index k is introduced such that for all $k \in K$ the normalized time follows from:

$$\tau_k = \left(\frac{k}{K-1}\right) \tag{5.82}$$

A hold of first-order is assumed over all time $\tau \in [\tau_k, \tau_{k+1}]$ elements for the preservation of feasibility, such that for all $k \in \bar{K}$ the following control law holds:

$$\mathbf{u}(\tau) = \left(\frac{\tau_{k+1} - \tau}{\tau_{k+1} - \tau_k}\right)\mathbf{u}_k + \left(\frac{\tau - \tau_k}{\tau_{k+1} - \tau_k}\right)\mathbf{u}_{k+1} \tag{5.83}$$

$$\mathbf{u}(\tau) = \alpha_k(\tau)\mathbf{u}_k + \beta_k(\tau)\mathbf{u}_k \tag{5.84}$$

For the propagation of the state $\mathbf{x}_k = \mathbf{x}(\tau_k)$ to $\mathbf{x}_{k+1} = \mathbf{x}(\tau_{k+1})$ a state transition matrix is introduced that is governed through the following differential equation for all $k \in \bar{K}$:

$$\frac{d}{d\tau}\mathbf{\Phi}_A(\tau, \tau_k) = \mathbf{A}(\tau)\mathbf{\Phi}_A(\tau, \tau_k) \quad \text{with} \quad \mathbf{\Phi}_A(\tau_k, \tau_k) = I \tag{5.85}$$

The complete set of governing discrete in time dynamic equations relating $\mathbf{x}_k$ to $\mathbf{x}_{k+1}$ for all $k \in \bar{K}$ is finally presented:

$$\mathbf{x}_{k+1} = \bar{\mathbf{A}}_k\mathbf{x}_k + \bar{\mathbf{B}}_k\mathbf{u}_k + \bar{\mathbf{C}}_k\mathbf{u}_{k+1} + \bar{\sum}_k \sigma + \bar{\mathbf{z}}_k \tag{5.86}$$

$$\bar{\mathbf{A}}_k = \Phi_A(\tau_{k+1}, \tau_k) \tag{5.87}$$

$$\bar{\mathbf{B}}_k = \int_{\tau_k}^{\tau_{k+1}} \Phi_A(\tau_{k+1}, \xi)\mathbf{B}(\xi)\alpha_k(\xi)d\xi \tag{5.88}$$

$$\bar{\mathbf{C}}_k = \int_{\tau_k}^{\tau_{k+1}} \Phi_A(\tau_{k+1}, \xi)\mathbf{B}(\xi)\beta_k(\xi)d\xi \tag{5.89}$$

$$\bar{\sum}_k = \int_{\tau_k}^{\tau_{k+1}} \Phi_A(\tau_{k+1}, \xi)\sum(\xi)d\xi \tag{5.90}$$

$$\bar{z}_k = \int_{\tau_k}^{\tau_{k+1}} \Phi_A(\tau_{k+1}, \xi)z(\xi)d\xi \tag{5.91}$$

With is discrete format of governing dynamic equations the discussion on linearization is enclosed.

5.3.2. Control Regions and Virtual Control

The main working principle of the successive convexification method is to solve a sequence of relaxed convex problems in an iterative manner, with i denoting the i^{th} iteration. For this methodology to work properly, the control regions of must maintain feasible and bounded throughout the landing sequence. This entails that the cost function is to be augmented to mitigate the risk of unbound solutions. To start with, the following parameters are defined

$$\delta\mathbf{x}_k^i = \mathbf{x}_k^i - \mathbf{x}_k^{i-1}, \quad \forall k \in \mathbb{K} \tag{5.92}$$

$$\delta\mathbf{u}_k^i = \mathbf{u}_k^i - \mathbf{u}_k^{i-1}, \quad \forall k \in \mathbb{K} \tag{5.93}$$

$$\delta\sigma^i = \sigma^i - \sigma^{i-1} \tag{5.94}$$

Subsequently, defining $\mathbf{\Delta}^i \in \mathbb{R}_+^K$ and $\Delta_\sigma^i \in \mathbb{R}_+$, the constraints are entrenched

$$\delta\mathbf{x}_k^i \cdot \delta\mathbf{x}_k^i + \delta\mathbf{u}_k^i \cdot \delta\mathbf{u}_k^i \leq \mathbf{e}_k \cdot \mathbf{\Delta}^i, \quad \forall k \in \mathbb{K} \tag{5.95}$$

$$\delta\sigma^i \cdot \delta\sigma^i \leq \Delta_\sigma^i \tag{5.96}$$

To deal properly with the additional constraints, the cost function is augmented through the addition of weight terms $w^i_\Delta \in \mathbb{R}_{++}$ and $w_{\Delta_\sigma} \in \mathbb{R}_{++}$, such that

$$c^i_\Delta = w^i_\Delta \left\| \boldsymbol{\Delta}^i \right\|_2 + w_{\Delta_\sigma} \left\| \Delta^i_\sigma \right\|_1 \tag{5.97}$$

Problem 10: Successive convexification problem definition in discrete time:

Cost Function:

$$\min_{\sigma^i, \mathbf{u}^i_k} \quad \sigma^i + w_v \left\| \bar{\mathbf{v}}^i \right\|_1 + w^i_\Delta \left\| \bar{\boldsymbol{\Delta}}^i \right\|_2 + w_{\Delta_\sigma} \| \Delta_\sigma \|_1 \quad \text{subject to:}$$

Dynamics:

$$\mathbf{x}^i_{k+1} = \bar{\mathbf{A}}^i_k \bar{\mathbf{x}}^i_k + \bar{\mathbf{B}}^i_k \bar{\mathbf{u}}^i_k + \bar{\mathbf{C}}^i_k \bar{\mathbf{u}}^i_{k+1} + \bar{\sum}^i_k \sigma^i + \bar{\mathbf{z}}^i_k + \bar{\mathbf{v}}^i_k$$

State Constraints:

$$\begin{aligned}
m_{\text{dry}} &\leq m^i_k \\
\tan\gamma_{gs} \left\| H^T_{23} \mathbf{r}^i_{G,k}(t) \right\|_2 &\leq \mathbf{e}_1 \cdot \mathbf{r}^i_{G,k} \\
\cos\eta_{max} &\leq 1 - 2 \left\| H_q \mathbf{q}^i_{B,G,k} \right\|^2_2 \\
H_q &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
\left\| \boldsymbol{\omega}^i_{B,k} \right\|_2 &\leq \omega_{\max}
\end{aligned}$$

Control Constraints:

$$\begin{aligned}
F_{\min} &\leq \mathbf{B}_g(\tau_k) \mathbf{u}^i_k \\
\left\| \mathbf{u}^i_k \right\|_2 &\leq F_{\max} \\
\cos\delta_{max} \left\| \mathbf{u}^i_k \right\|_2 &\leq \mathbf{e}_1 \cdot \mathbf{u}_k
\end{aligned}$$

Boundary Conditions:

$$\begin{aligned}
& m^i_0 = m_{\text{wet}} \\
& \mathbf{r}^i_{G,0} = \mathbf{r}_{G,i} \quad \text{and} \quad \mathbf{r}^i_{G,K} = \mathbf{0} \\
& \mathbf{V}^i_{G,0} = \mathbf{V}_{G,i} \quad \text{and} \quad \mathbf{V}^i_{G,K} = \mathbf{V}_{G,f} \\
& \mathbf{q}^i_{B,G,K} = \mathbf{q}_{B,G,f} \\
& \boldsymbol{\omega}^i_{B,0} = \boldsymbol{\omega}_{B,i} \quad \text{and} \quad \boldsymbol{\omega}^i_{B,K} = \mathbf{0} \\
& \mathbf{e}_2 \cdot \mathbf{u}^i_K = \mathbf{e}_3 \cdot \mathbf{u}^i_K = 0
\end{aligned}$$

Trust Regions:

$$\begin{aligned}
\delta\mathbf{x}^i_k \cdot \delta\mathbf{x}^i_k + \delta\mathbf{u}^i_k \cdot \delta\mathbf{u}^i_k &\leq \Delta^i_k \\
\left\| \delta\sigma^i \right\|_1 &\leq \Delta^i_\sigma
\end{aligned}$$

During the process of convergence it can be encountered that the linearization is not feasible, known as artificial infeasibility. To cope with the requirements of on-board applications, a virtual control parameter $v^i_k \in \mathbb{R}^{14}$ is introduced. This parameter is integrated into dynamics and its assigned a weigh term of its own. In case of heavy infeasibility, the control parameter acts to prevent the algorithm from losing convergence. The weight of parameter is adjusted in accordance with the solution quality of each iteration till a certain desired threshold is achieved. The general rigid-body six-degree-of-freedom terminal guidance formulation is presented in Problem 10. This includes all relevant forces, which are established in accordance with the nature of the descent. Notice that formulation is successively convexified and conferred in discrete time.

5.4. Multi-Glideslope for Obstacle Avoidance Guidance

In the last portion of this chapter the algorithm is extended for obstacle avoidance compatibility based on multi-glideslope constraints. Specifically, when approaching the destination target, roughness of terrain might raise the need for avoiding path obstacles detrimental to the mission. Examples of such are mountains and hills for which a subsurface flight constraint or single glideslope constraint does not suffice, or would not yield the most optimal solution. Determining the properties of this terrain, such as terrain altitude and slope, would be based on sensor input data (e.g. TRN and HDA) or pre-loaded on-board maps. The discussion regarding the navigation aspect of this matter is beyond the scope of this research. This section focuses on the mathematical principles of multiple glideslope constraint guidance, such that the mission is subdivided into several segments, and subsequently optimized individually while ensuring mission safety. This is especially necessary when landing in areas with high variation in elevation. Another mission scenario would be for the spacecraft to have to land in a crater or in between mountains. The glide-slope constraint enforces that a specific optimal trajectory to the target is not too shallow nor goes below the surface. The glide-slope constraint requires the spacecraft to remain within a cone as defined by the minimum slope angle. To cope with the requirement of, e.g., TRN, the spacecraft orientation in terms of attitude is restrained. By enforcing the cone shape restricted motion, the lander at higher altitudes for an elongated period, hence surface characteristics can be observed for an certain extended period of time. As earlier conferred in Chapter 4, **S** and **c** are defined to designate a cone with a vertex at $\mathbf{r}(t_f)$, leading to

$$\mathbf{S} \triangleq \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{c} \triangleq \frac{\mathbf{e}_1}{\tan\gamma_{gs}} \gamma_{gs} > 0 \tag{5.98}$$

In this research two mission scenarios are considered in which the activation of this guidance system would be necessary. Firstly, as depicted in the sample simulation of Fig. 5.7, a scenario might arise in which several hills raise the need for segmenting the landing sequence into multiple phases, thereby enforcing different

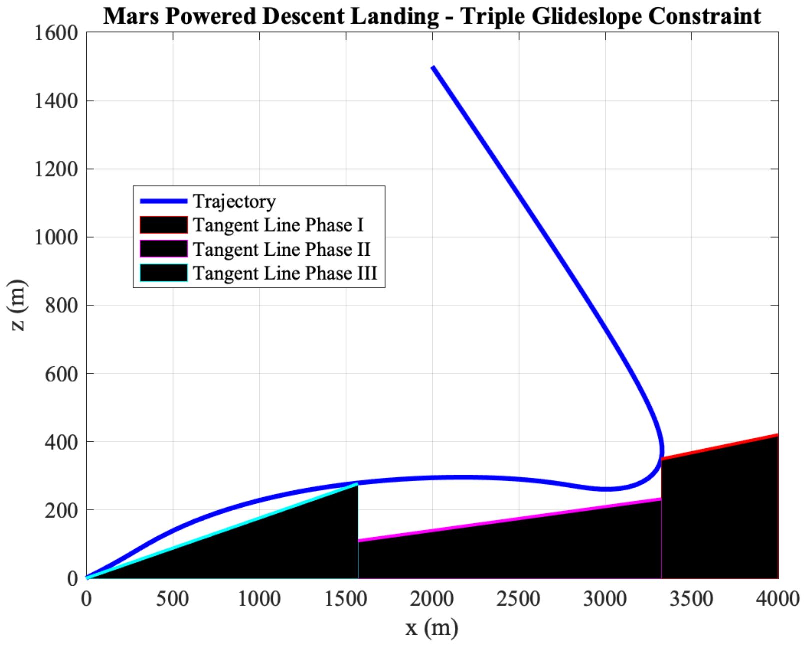


Figure 5.7: Sample simulation of the multiple glideslope constraint for obstacle avoidance during planetary descent. Notice that the trajectory in the figure is made to touch the extreme points exactly for visualization purposes, which in real-time would not be the case.

tangent slopes based on hill altitude and slope. This guarantees mission safety while still adhering to the mission constraints, optimal path planning and soft landing, and is enforced by

$$\begin{array}{l} \text{for} \quad i = 1:N \\ \qquad \gamma_{gs_n} \quad [0, \gamma_{gs_{\max}}] \quad \in \mathbb{R} \\ \qquad r_1 \leq \dfrac{r_3}{\tan(\gamma_n)} \\ \text{end} \end{array} \tag{5.99}$$

Notice that this loop can be segmented into multiple phases and repeated if necessary. In the example on the previous page, a total of three phases have been adopted. The trajectory in the figure is made to touch the extreme points exactly for visualization purposes, which in real-time would not be the case. In the results section of this thesis the difference between the nominal optimal path and the augmented paths are outlined.

A second possible scenario might arise when a spacecraft needs to land at the rim of a large mountain or in between a valley. A single and regular glideslope constraint would be too steep in such case, constraining the trajectory excessively or yielding the optimal path infeasible. For such cases the path is yet again augmented into differential steps. Distinct from the multiple multi-glide constraint, the obstacle avoidance guidance is enforced in a slightly different manner. In the outline conferred below the principles are based on a double phase scenario. For a mission profile requiring more phases, the algorithm can be adjusted accordingly. The constraint on the first half of the temporal nodes yield

$$\begin{array}{l} \text{for} \quad i = 1:N/2 \\ \qquad \alpha_{gs_n} \quad [0, \alpha_{gs_{\max}}] \quad \in \mathbb{R} \\ \qquad \beta_{gs_n} \quad [0, \beta_{gs_{\max}}] \quad \in \mathbb{R} \\ \qquad r_3 \geq \alpha_{gs_n} r_1 + \beta_{gs_n} \\ \text{end} \end{array} \tag{5.100}$$

While for the second phase of the landing sequence (from $N/2 : N$) the constraint is enforced in a same fashion as outlined in Eq. (5.99). A sample simulation is provided in Fig. 5.8. Likewise, in the results chapter (Section 7.4) of the thesis, a full analysis is provided compared to the nominal trajectory.

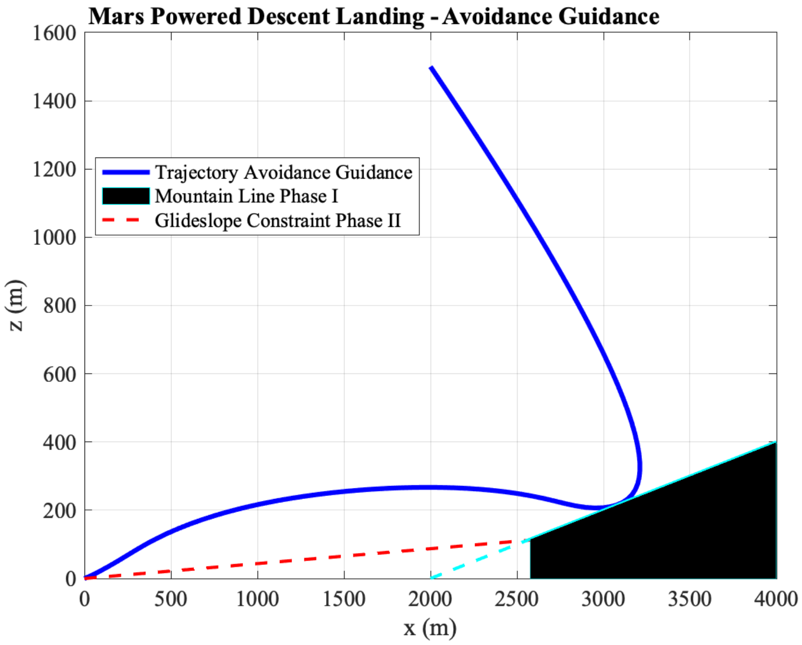


Figure 5.8: Sample simulation of the mountain avoidance guidance algorithm applied during planetary descent. Notice that the trajectory in the figure is made to touch the extreme points exactly for visualization purposes, which in real-time would not be the case.

6

Simulator Design and Verification

Based on the concepts, theories and mathematical principles outlined in the preceding chapters, this part of the thesis outlines the simulator design. In particular, this chapter starts (in Section 6.1) with the delineation on the generated architecture and declares the order in which each individual operation is to be executed. The design of the guidance, navigation and control systems is centered around the powered or parafoil descent phase nominal trajectory. That is the stage in which utilization of the full spacecraft capability would make it plausible for the vehicle to correct for accumulated dispersions and re-guide the spacecraft to precision. Notice that hardware is not part of the designed system nor the research, hence the outline of an experimental simulator. To ensure the proper working of this algorithm simulator, a step-wise verification process has been coupled to the research. Section 6.2 delineates the conducted unit tests and established the correct functionality of the nominal algorithm. This shall also enhance future research at NASA/JPL and TU Delft on the matter of precision landing on planetary bodies.

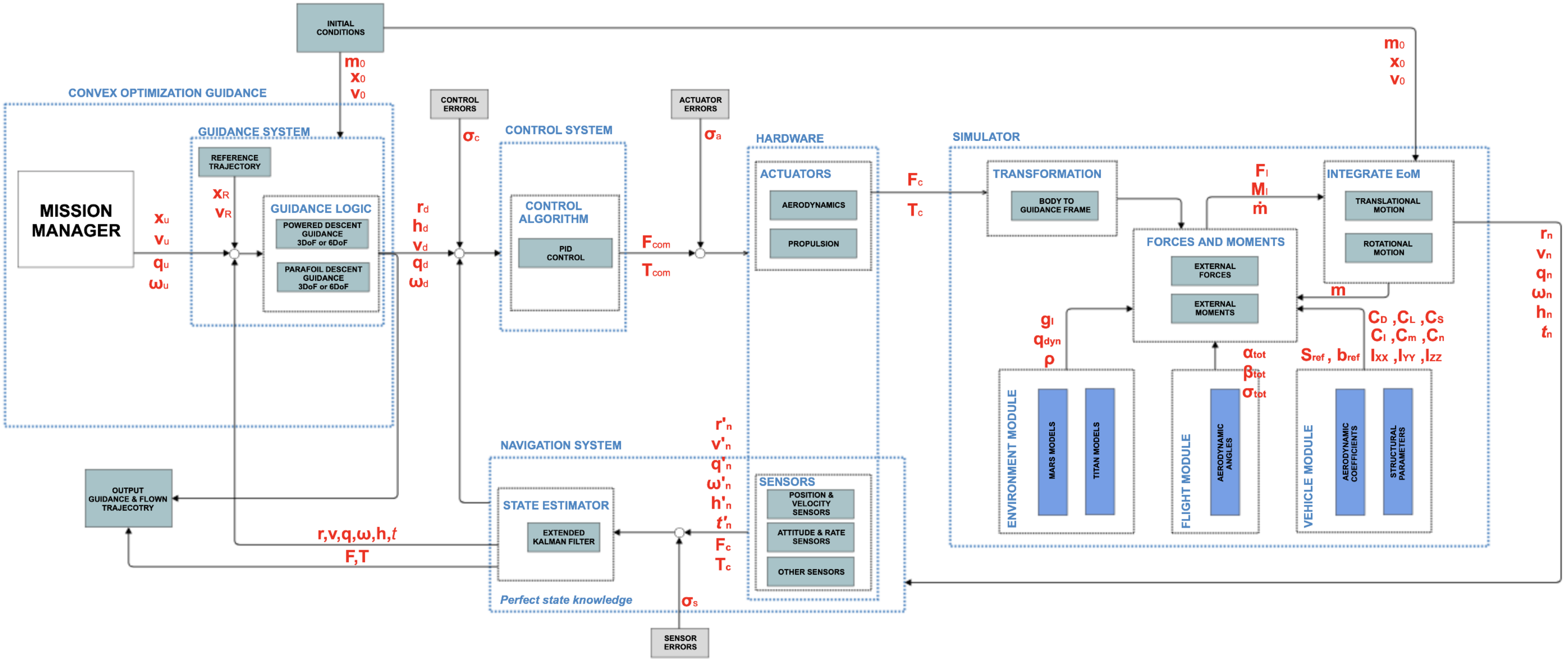
6.1. Top-Level Architecture

To initiate the section on the top-level architecture of the simulator, a descriptive functionality overview is provided. For the outline of the respective elements of the architecture, the reader is referred back to Section 4.1. MATLAB is the software in which the top level architecture is written, for both the Mars simulations as well as for Titan. The main reasons for choosing this platform is the extensive of built-in convex optimization solvers (CVX) and the proof of concepts of prior work that have conducted simulations using CVX. Furthermore, as this is a research thesis, real-time implementation of the algorithm is not necessary for proof of concept. The concept of precision landing is established as a potential solution through Matlab simulations, with a recommendation to transfer to for example C++ for on-board implementation. The code can be converted to C++ when fast computations are needed, which would be the case for actual flight software. The methods for numerical propagation of the states are outlined in Appendix C, while the detailed outline of the algorithm files is provided in Appendix D. Please notice that these files are only available at Delft University of Technology and NASA's Jet Propulsion Laboratory / California Institute of Technology.

To follow up with the discussion in this section, Fig. 6.1 depicts the top-level architectural of the simulator to be designed. The governing model is centered around the guidance of the powered descent phase while the other systems (i.e., navigation and control) are integrated for closed-loop purposes. Notice, however, that the navigation segment of the simulator assumes perfect state knowledge (for reasons conferred in Section 4.3). To concretize the integration strategy and appraise the working behavior of each of the individual GNC model based blocks, the following strategy is followed in the process:

1. Design powered descent guidance (PDG) algorithm (nominal trajectory)
2. Perform point-mass PDG simulation
3. Design powered descent control system
4. Perform point-mass powered descent GNC simulation
5. Extend algorithm to parafoil descent guidance
6. Extend algorithm to obstacle avoidance guidance
7. Perform powered and parafoil descent GNC sensitivity analysis

Figure 6.1 Top-level architectural design of the simulator for precision landing on Mars and Titan. Input and output data flows are highlighted in red.



The verification to be discussed in the next section and the sensitivity analysis to be discussed in the subsequent chapter are directly coupled to this design outlined on the previous page. The descriptive outline of the functionalities are conferred below.

- **Guidance:** Forming the core of the research and differentiated into two segments: powered (Mars) and parafoil (Titan) guidance. The nominal trajectories are taking as reference path, while an active guidance system makes sure the vehicle adheres to this optimal path by providing corrective manoeuvres to the introduced error. Section 4.2 outlines the guidance laws in detail. A sample rate of 100 Hz is used for simulating the trajectories.
- **Control:** A PD/PID controller is utilized to maintain the vehicle in-path, running at a control rate of 20 Hz. Control errors (for example, due wind) are induced to assess on-board re-optimization capabilities. For the outline of the control laws, be referred to Section 4.4.
- **Actuators:** As described in more detail in Chapters 4 and 5, the Mars landing sequence utilizes thrusters during the landing segment, while the Titan landing makes use of the pull of the parafoil canopy to maneuver within the moon's thick atmosphere. The actuator errors (such as misalignment of thrusters, which is a realistic scenario), are introduced to test the robustness of the algorithm.
- **Flight Dynamics & Environment:** The dynamics are governed by the aerodynamic and thruster (i.e., propulsive) parameters. For a comprehensive of the governing forces, moments and equations, the reader is referred to Chapter 3. In Appendix C the mathematical principles of integrators are delineated.
- **Navigation:** For reasons entrenched in Section 4.3, this research assumes perfect state knowledge.
- **Mission Manager:** The mission manager runs the landing simulation at a rate of 100 Hz while ensuring the algorithm adheres to the governing laws.

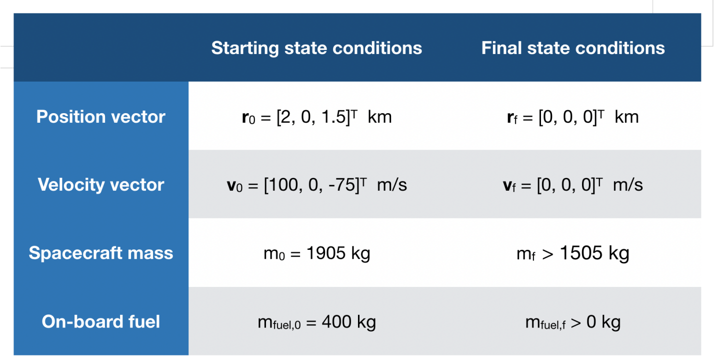
The algorithm parameters are defined as the established boundary conditions within the problem formulation. This include obstacle avoidance guidance, utilized to dodge hills and/or land near mountain rims. The models of the two landing sequences considered are the MSL/Mars 2020 like vehicle for the Mars landing and the parafoil model for Titan. The properties of both have been provided in the preceding chapters. As the descent phase takes place at relatively low altitudes of the EDL, the gravity and density model are kept simple. Changes of these two parameters are marginal at low altitudes. As part of the environmental model, wind gusts play a significant role on both planetary bodies and their effect has been analyzed in this research. All the above described elements together eventually define optimal trajectories for the space vehicle to follow, thereby establishing precision landing data of various types. This data it utilized for generating graphs that visualize the landing profile and provide in-depth understanding of the motion. In the next section the algorithm coupled to this top-level architecture is verified based on the extensive available literature.

6.2. VERIFICATION

In this final section of the chapter a verification framework is entrenched as the proof that the designed product complies with prior established requirements. The designed algorithm is directly compared to powered descent guidance simulations of industry's best practise in terms of precision landing. Landing on Mars has been extensively researched and hence considerable literature resources are available for comparison. The verification is based on point-mass simulations for lossless convexification. The verification process is segmented into six system tests which also marks the outline of this section, and are charted beneath:

1. **Verification I:** Powered descent guidance for Mars landing utilizing the convex programming approach based on Açıkmeşe and Ploen (2007) → Section 6.2.1.
2. **Verification II:** Golden search method for finding the optimal time-of-flight based on Açıkmeşe et al. (2008) → Section 6.2.2.
3. **Verification III:** Minimum landing error guidance for Mars Landing based on Blackmore et al. (2010) → Section 6.2.3.
4. **Verification IV:** Non-convex thrust bound and constraint of pointing for powered descent based on Carson et al. (2011) → Section 6.2.4.
5. **Verification V:** Soft landing optimal control based on Açıkmeşe et al. (2012) and Açıkmeşe et al. (2013) → Section 6.2.5.
6. **Verification VI:** Six-degree-of-freedom rigid body Mars powered descent guidance after Szmuk and Açıkmeşe (2018) → Section 6.2.6.

Table 6.1: Mars landing initial and terminal state conditions in accordance with Açıkmeşe and Ploen (2007) and Açıkmeşe et al. (2013).



	Starting state conditions	Final state conditions
Position vector	$\mathbf{r}_0 = [2, 0, 1.5]^T$ km	$\mathbf{r}_f = [0, 0, 0]^T$ km
Velocity vector	$\mathbf{v}_0 = [100, 0, -75]^T$ m/s	$\mathbf{v}_f = [0, 0, 0]^T$ m/s
Spacecraft mass	$m_0 = 1905$ kg	$m_f > 1505$ kg
On-board fuel	$m_{fuel,0} = 400$ kg	$m_{fuel,f} > 0$ kg

The baseline starting conditions are presented in Table 6.1, which are altered slightly based on the specific unit verification to the maintain the one-on-one verification. Naturally, these adjustment are conferred in their respective sections. The section concludes by conferring results for six-degree-of-freedom simulations. The latter simulation properly show the effect of inclusion of the glide-slope constraint within the simulation.

6.2.1. Verification I: Powered Descent Convex Programming

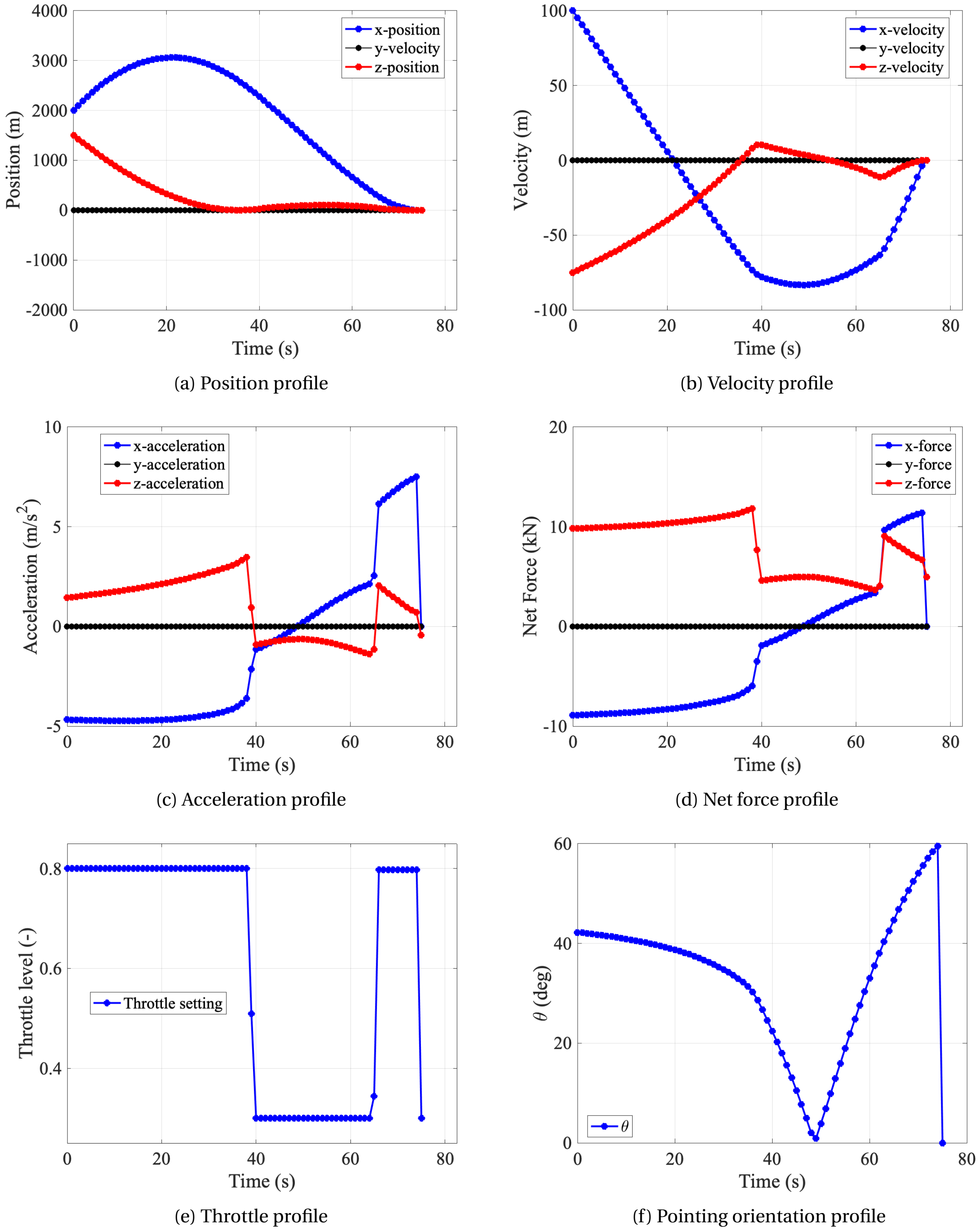
The verification results of the convex programming powered descent point-mass simulation are presented in this section. The comparison inn this section focuses on the landing sequence with solely a no-subsurface flight constraint (i.e., no other constraints). For the results of the plain sequence (no constraints at all) and the enforced glide-slope constraint in correspondence with Açıkmeşe and Ploen (2007), the reader is referred to Appendix E. The starting and terminal state conditions (position, velocity and mass as established in Table 6.1) are in accordance with relevant literature to allow for one-on-one verification. Furthermore, the maximum thrust $\bar{T} = 3.1$ kN, while the throttle level setting various between $T_{min} = 0.3\bar{T}$ and $T_{min} = 0.8\bar{T}$. At last a landing constraint $\hat{n}_f = [1 \quad 0 \quad 0]$ is imposed on the sequence such that the vehicle lands in the direction opposite to the gravity vector. The results for the Mars in-plane maneuver are presented in Fig. 6.2.

From the time history of the position and velocity states it is deduced that the implemented algorithm produces identical results. Overall the algorithm plots resemble the reference plots very closely. It becomes evident from the position plot that when not enforcing the glide-slope constraint, the trajectory may reach zero-altitude early on in the sequence. This phenomena causes a large portion of the flight segment ($t \in (25, 50)$) to be subterranean, which is not desired during powered descent. In Figs. E.1, E.2 and E.3 the examples are conferred of a plain maneuver (no constraints) and a maneuver with the glide-slope constraint enforced, respectively. Section 6.2.3 also delineates the latter matter in more detail.

In accordance with denoted Lemmas in Chapter5 and Appendix B, the throttle history marks a max-min-max profile, corresponding with profiles often encountered in optimal control literature. With regard to the acceleration and net force time history there is one significant difference in the final value of the gravity direction acceleration. This difference might be attributed to the fact that the respective literature seems to enforce a final burn in that direction. This hypothesis has been tested by varying the final thrust direction magnitude, which indeed generates different final throttle profiles. Another difference cause could be the applied discretization scheme. The same type of discrepancy is observed for the final throttle setting. This might also be the direct consequence of the small difference between the fuel consumption of the two algorithms, 387.9 kg vs 389.4 kg as respectively denoted in Table 6.2. Nonetheless, both plots are overall rather similar. This also becomes evident from the thrust pointing in the final figure. It is evident from the overall plots that the control and state constraints are satisfied throughout the entire sequence.

Table 6.2: Outline of the optimal flight-time t_f and fuel consumption m_{fuel} results of the literature source Açıkmeşe and Ploen (2007) and the designed algorithm. Notice that the left part of each table segment refers to the respective values obtained by Açıkmeşe and Ploen (2007), while the right part denotes the figure and result of this thesis.

Parameters	**Paper**	**Fig. E.1**	**Paper**	**Fig. 6.2**	**Paper**	**Fig. E.2**	**Paper**	**Fig. 6.3**
Optimal time (s)	72	73	75	75	81	81	69	71
Fuel consumption (kg)	387.9	389.4	390.4	391.8	399.5	399.9	293.6	293.4



(a) Position profile

(b) Velocity profile

(c) Acceleration profile

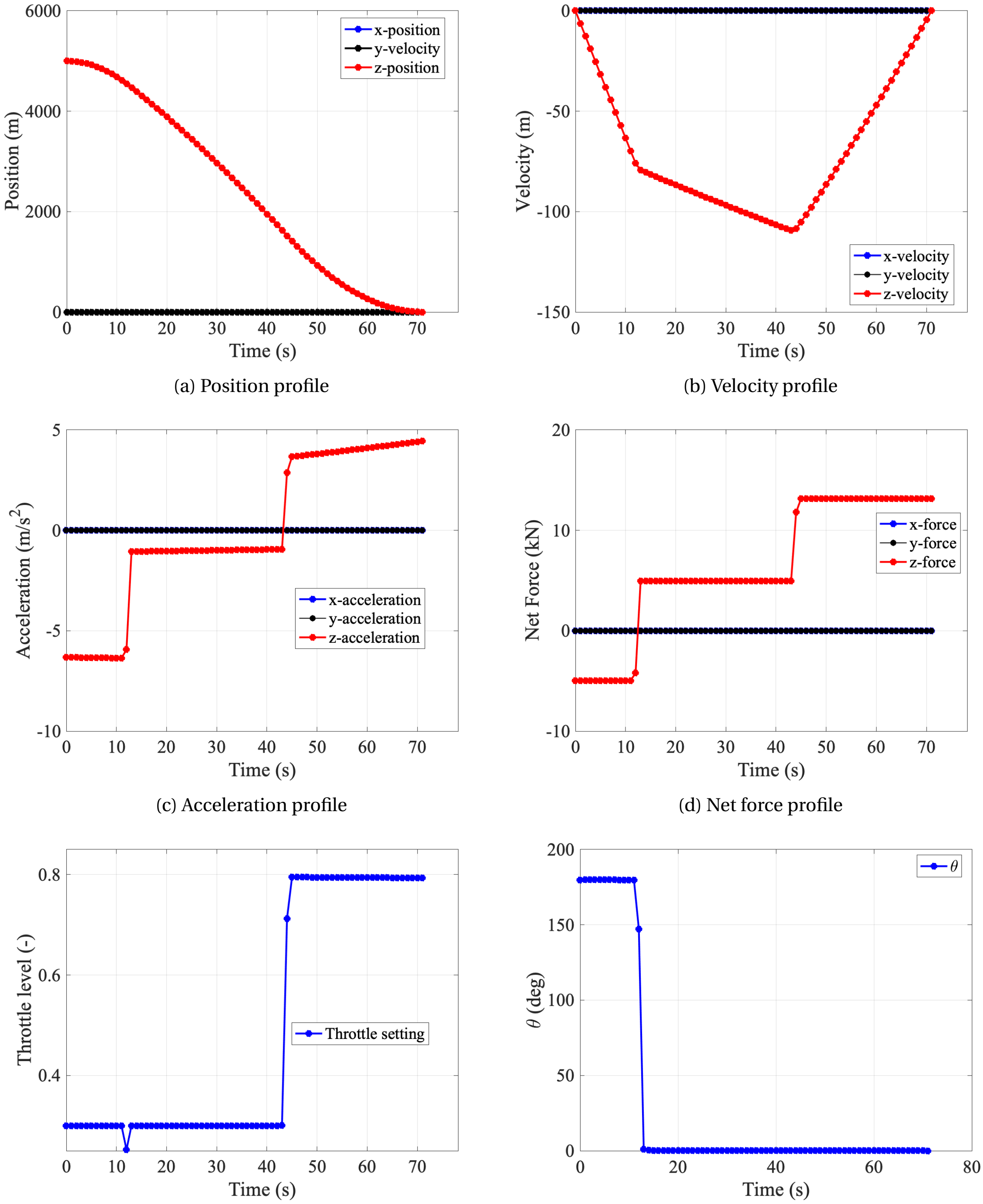
(d) Net force profile

(e) Throttle profile

(f) Pointing orientation profile

Figure 6.2: Mars in-plane maneuver with a subsurface flight condition and the glide-slope constraint not enforced,. Time history of the position, velocity, acceleration, net force, throttle setting and pointing orientation. Figures in correspondence with Fig. 6 of Açıkmeşe and Ploen (2007).

To demonstrate the working principle of the thrust point constraint, the example described in the last portion of this section is used. This corresponds to Fig. 6.3, with altered starting conditions of $r_0 = [5000, 0, 0]^T$ and $\dot{r}_0 = [0, 0, 0]^T$ with respect to the nominal conditions established in Table 6.2. Initially, the vector points in the opposite direction of the gravity direction, which later on in the sequence is reversed. Likewise, the algorithm produces near identical results with regard to the paper. Small discrepancies are again observed



(a) Position profile

(b) Velocity profile

(c) Acceleration profile

(d) Net force profile

(e) Throttle profile

(f) Pointing orientation profile

Figure 6.3: Mars in-plane maneuver for visualization purposes of the thrust point constraint. Time history of the position, velocity, acceleration, net force, throttle setting and pointing orientation. Figures in correspondence with Fig. 10 of Açıkmeşe and Ploen (2007).

which are given similar reasons as the ones conferred for Fig. 6.2. Additionally, it is observed that the location of the temporal nodes differs from the paper at various locations. This further emphasizes the idea that the discretization scheme is the main cause of the differences. **Conclusively, the system test of the convex programming algorithm is verified.**

6.2.2. VERIFICATION II: GOLDEN SEARCH METHOD

The methodology for finding the optimal time-of-flight during the powered descent sequence, the golden search method, has been outlined in Chapter 5. In this section the verification of this methodology is provided. In Fig. 6.4 several fuel consumption values have been plotted against a time-of-flight range (based on conservative minimum and maximum values). The figure, which is typical in nature for a t_f vs fuel usage plot, is motivated through the realization that the minimum fuel is a function of uni-modal nature that has a global minimum. The fuel consumption figure corresponds to the starting values entrenched in Table 6.2 while running the guidance for fuel optimal large diverts algorithm. Notice that the physical limit denotes the maximum on-board fuel of 400 kg, which rules out a significant portion of the possible flight time values.

The optimal flight time corresponds to a rounded-off value of 73 seconds, one second longer than the value presented in the paper for the same scenario. The reasons for this slight difference are attributed to the discrepancies discussed in the previous section. The throttle profile is marginally different, which naturally causes a different mass consumption profile and hence a different optimal time of flight. Discretization of the algorithm plays a significant role here-in. As mentioned in Açıkmeşe et al. (2008), time discretization even causes rise to the issue of inter-sample subsurface flight. This means within the instances of time (i.e., the temporal nodes), there is no guarantee on a theoretical base that the algorithm adheres to the imposed constraint. This is due to the fact that the constraints (especially the glide-slope constraint to be outlined in the next section) is imposed on a number of finite instances of time. This issue can resolved by imposing additional constraints, for which the reader is referred to the paper for further information. Regarding the optimal time of flight, within the given context, it is verified that the algorithm produces realistic results. **Conclusively, the system test of the golden search algorithm is verified.**

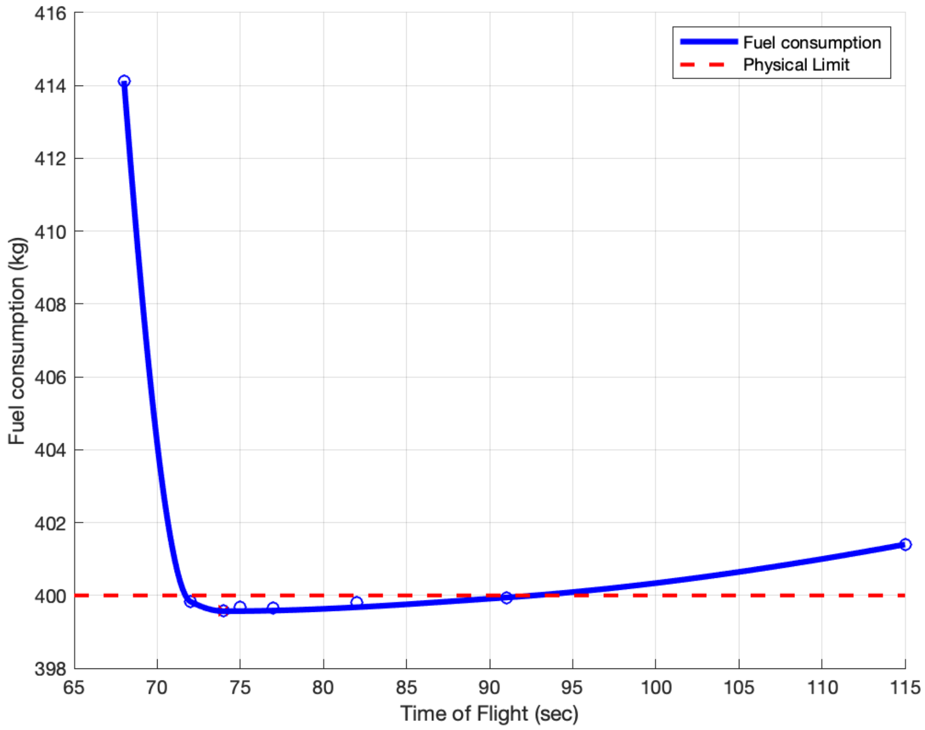
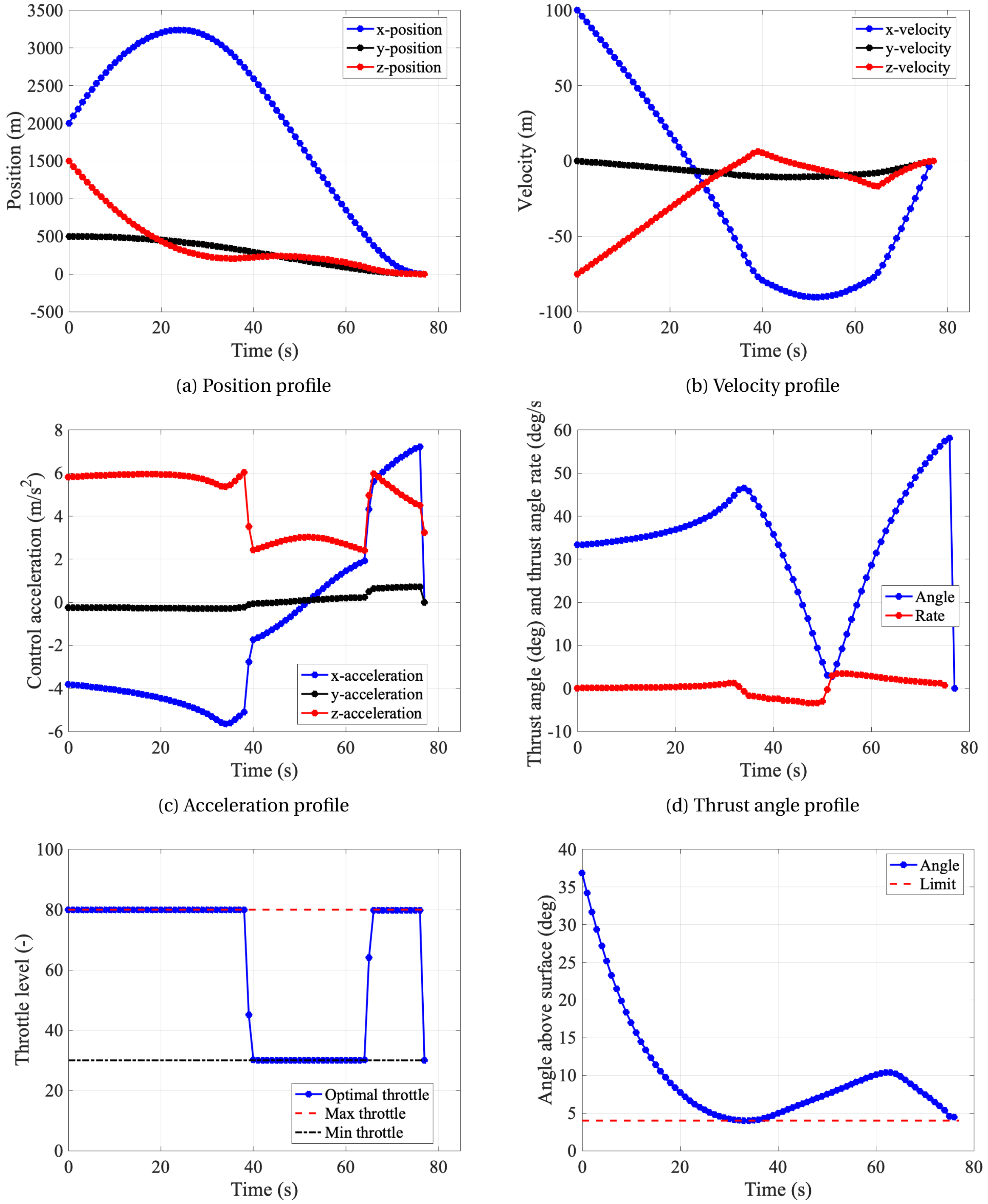


Figure 6.4: Golden search method applied for finding optimal time-of-flight t_f.

6.2.3. VERIFICATION III: MINIMUM LANDING ERROR

The second simulation in terms of Mars landing assumed an out-of-plane maneuver (three-dimensional Euclidean space) with the enforcement of glide-slope constraint. Furthermore, the algorithm is extended such that it is compatible for providing a minimum landing error guidance sequence. The conditions are slightly altered with respect to the conditions entrenched in Table 6.2. The propulsion system is limited in terms of minimum and maximum thrust such that $\rho_1 = 4972N$ and $\rho_2 = 13260N$, with a depletion rate of $\alpha = 4.53 \times 10^{-4}$ s/m. To prevent the trajectory from reaching the surface early in the descent, a glide-slope constraint is enforced at an angle of 4°. The initial position and velocity are given by $\mathbf{r}_0 = [1500, 500, 2000]^T$ m and $\dot{\mathbf{r}}_0 = [-75, 0, 100]^T$ m/s. In Figs. 6.5 and 6.6 the results of the simulations are provided. Corresponding to the results in Blackmore et al. (2010), the algorithm was terminated at $t_f = 79s$ with a fuel consumption



(a) Position profile

(b) Velocity profile

(c) Acceleration profile

(d) Thrust angle profile

(e) Throttle profile

(f) Angle above surface profile

Figure 6.5: Mars out-of-plane maneuver with the glide slope constraint enforced. Time history of the position, velocity, acceleration, thrust angle, throttle setting and angle above surface. Figures in correspondence with Fig. 2 of Blackmore et al. (2010).

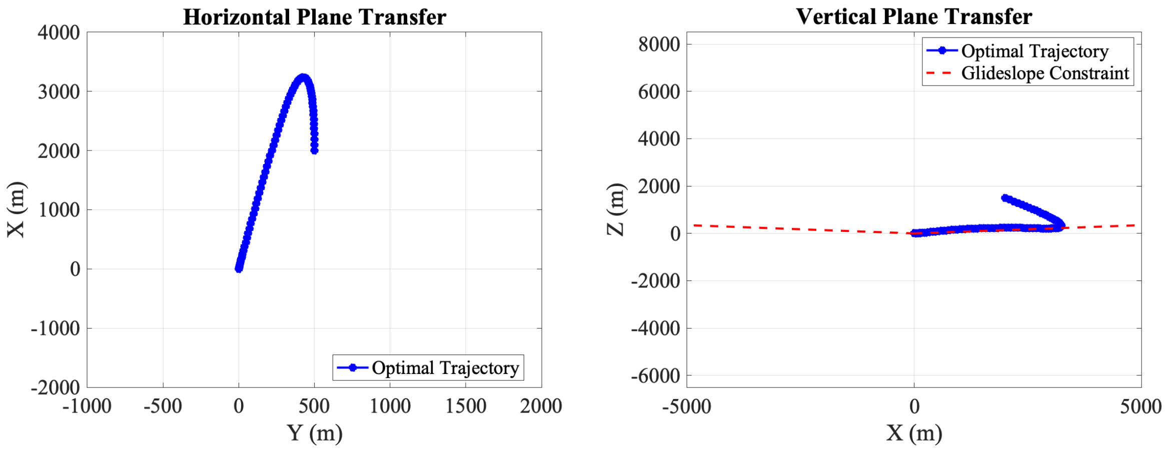


Figure 6.6: Mars out-of-plane maneuver with the glide slope constraint enforced. Horizontal and vertical plane transfer. Figures in correspondence with Fig. 2 of Blackmore et al. (2010).

of 399.4 kg. This establishes the value of the convex optimization minimum landing error method, as an optimal solution was found even when the spacecraft's physical boundaries where nearing its edge in terms of distance from target and initial velocity. One can observe from the figures that the altitude trajectory abstains from tending to zero until sequence termination in the soft-landing process, which verifies the proper working on the glide-slope constraint. This is especially well visible in Fig. 6.6. Likewise, the throttle profile is of a max-min-max nature, after the lemma of optimal control problems. Furthermore, the soft landing on the spacecraft is entrenched, as both position and velocity elements terminate at zero. In Appendix E an example is provided for which the paper claims there is no absolute soft landing. What is remarkable however is that designed algorithm of this research still managed to guide the spacecraft to precision landing. **Conclusively, the system test of the minimum position error guidance algorithm is verified.** The next section further enhances the non-convex pointing constraint and the thrust bound.

6.2.4. VERIFICATION IV: NON-CONVEX POINTING CONSTRAINT AND THRUST BOUND

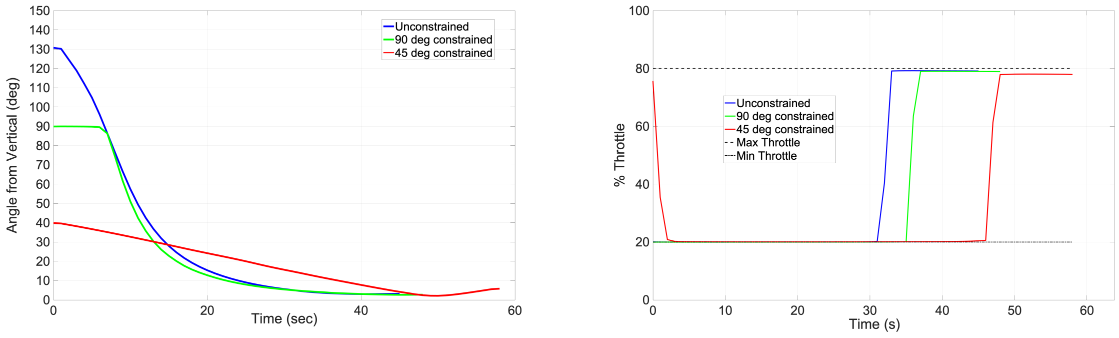
The pointing constraint formulation as part of the powered descent algorithm ensures that the spacecraft orientation is restricted within a desired pointing cone. This constraint is needed for example for terrain-relative sensors that need a specified viewing field during descent and landing. The constraint, and the lower thrust bound coupled to it, are both non-convex in nature. The convexified algorithm is verified in this section. The descent is initiated assuming the following properties:

- Initial mass $m_0 = 2000$ kg
- Fuel mass $m_f = 300$ kg
- Minimum operational throttle $\rho_1 = 0.2T_{\max}$
- Maximum operational throttle $\rho_2 = 0.8T_{\max}$
- Maximum design thrust $T_{\max} = 24000$ N

The simulations are performed out-of-plane, such that the position vector and velocity vector have non-zero elements in all three Euclidean space. Defined in the guidance frame, the following initial conditions hold $\mathbf{r}_0 = [2400, 450, -330]^T$ and $\dot{\mathbf{r}}_0 = [-10, -40, 10]^T$ A total of three simulations have been run with a variation in pointing constraint. The first to be unconstrained, the second at angle of 90° and the last at 45°, the result of which have been captivated in Table 6.3 and Fig. 6.7. As can be observed from the left figure, the profiles

Table 6.3: Outline of the fuel consumption m_{fuel} and optimal flight-time t_f results of the literature source Carson et al. (2011) and the designed algorithm. Notice that the left part of each table segment refers to the results of the paper, while the right part denotes the results of the designed algorithm of this thesis.

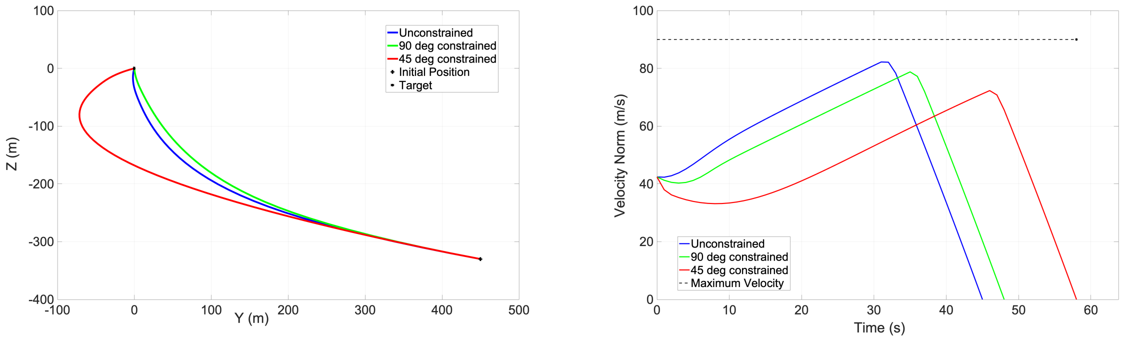
Parameters	Fuel Paper (kg)	Fuel Algorithm (kg)	Flight Time Paper (s)	Flight Time Algorithm (s)
Unconstrained	200.1	199.3	44.63	46
90 ° Constraint	201.8	201.8	46.96	49
45 ° Constraint	222.3	222.3	57.29	59



(a) Thrust pointing profile

(b) Throttle setting

Figure 6.7: Mars out-of-plane maneuver with the pointing constraint enforced. Thrust pointing attitude and throttle profile. Figures in correspondence with Figs. 5 and 6 of Carson et al. (2011).



(a) Surface trajectory

(b) Velocity norm

Figure 6.8: Mars out-of-plane maneuver with the glide slope constraint enforced. Surface trajectory and velocity norm. Left-hand figure in correspondence with Fig. 6c of Açıkmeşe et al. (2013).

of thrust pointing remain well within the established restrictions. The associated throttle magnitude profile is included in the depiction. The figures are well in correspondence with the original formulation of Carson et al. (2011). The only observed difference is the initial throttle value, which for the paper is about 38% and for the designed algorithm lays around 75%. This is likewise attributed to the formulation of the discretization scheme. This idea is further enforced through the provided table. For this research the choice has been made to make the flight-time a integer variable, hence the rounded off numbers. The paper presents different values, which consequently leads to a different fuel consumption of the unconstrained simulation. Remarkably, the fuel consumption for the remainder two simulations are exactly the same. At last it is stated that the tighter the vertical angle constraint, the higher the fuel usage and the required flight time. **With that observation it is concluded that the system test of the thrust pointing and thrust bound algorithm is verified.**

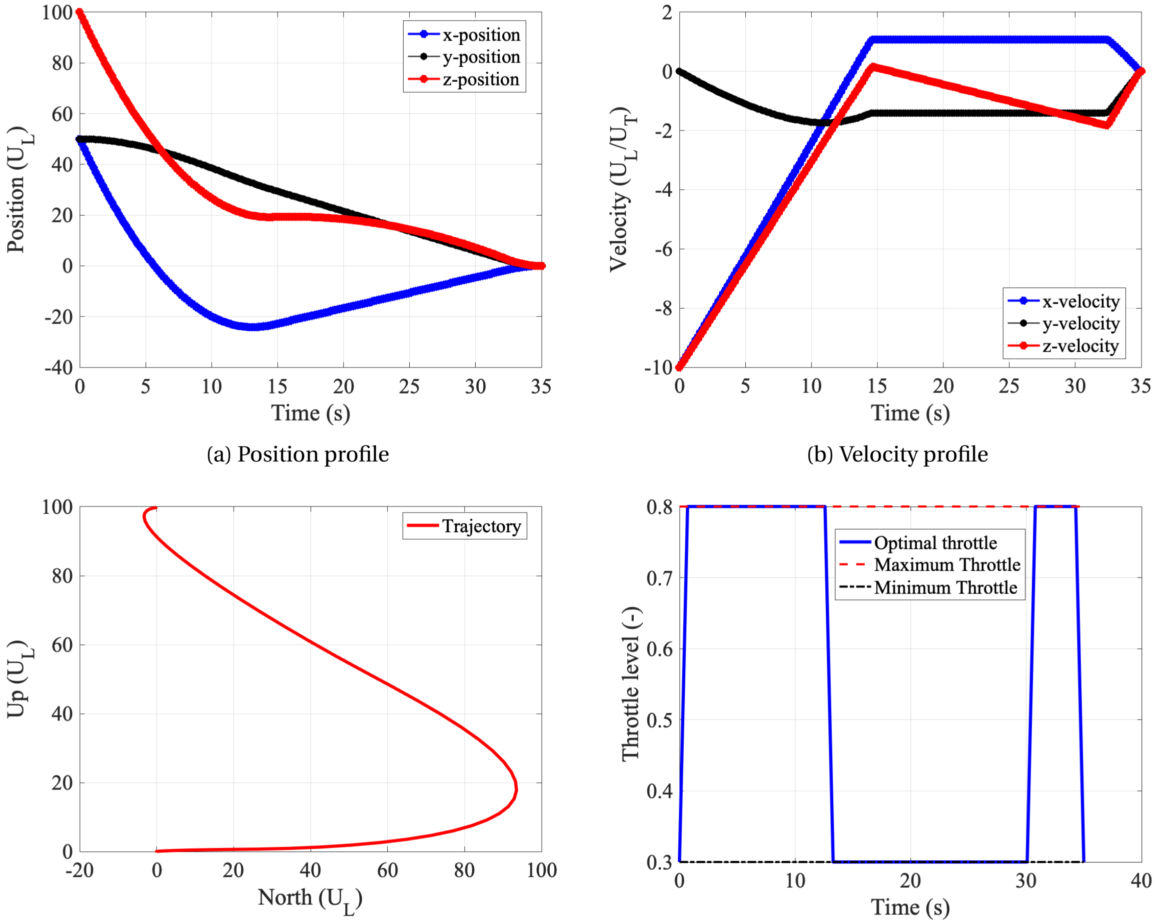
6.2.5. Verification V: Soft Landing Optimal Control

In this final section on unit verifications, the algorithm is run to solve the soft landing optimal control problem. The targeted landing site is defined at $\mathbf{d} = \mathbf{0}$. This unit verifications marks the entire planetary soft landing sequence with all its constraints and is based on Açıkmeşe et al. (2013). Similar to the unit verification conducted in previous section, it becomes evident from Fig. 6.8 that the tighter the pointing constraint, the more overshoot of the surface trajectory. The three trajectories correspond to the earlier conferred throttle profile. It is denoted that for these simulations a glide-slope angle γ_{gs} of 30 ° was enforced. The velocity as depicted in the right-hand figure was given a maximum value of 90 m/s. It becomes evident from the velocity norm plot that the algorithm manages to maintain below this normalized velocity throughout the sequence. From the surface trajectories it becomes evident that the 45° case overshoots the y-axis to maintain satisfaction of the pointing constraint, while the 90° and the unconstrained cases take a more direct path. This is in exact accordance with the results obtained by Açıkmeşe et al. (2013). **With that observation it is concluded**

that the system test of the soft landing optimal control algorithm is verified. In the next and final part of this chapter the emphasis is put on establishing the working principle of the glide-slope constraint through a 6DoF successive convex optimization out-of-plane sequence.

6.2.6. VERIFICATION VI: SIX-DEGREE-OF-FREEDOM MARS POWERED DESCENT GUIDANCE

The results for the six-degree-of-freedom Mars powered descent simulation are presented beneath. It is denoted that the simulations have been carried out assuming normalized unit values, rather that actual dimensions. This has been done to keep the discussion in accordance with Szmuk and Açıkmeşe (2018). Nonetheless, rather than reproducing the exact results of the paper, the designed algorithm that has been verified in the previous sections has been extended to become compatible with successive optimization guidance. The position profile in Fig. 6.9a clearly shows how the mission constraints are satisfied by making sure that the final position in all three directions match the landing location in the inertial guidance frame. Interestingly, the x−position shows signs of a large diversion from the path, indicating the algorithm has difficulty maintaining a more straight path. A similar observation is deduced from Fig. 6.9c. The up-north trajectory shows a large deviation in northern direction, before the algorithm re-guides the spacecraft to the designated target. For the position profile in Fig. 6.9b these deviations are not explicitly present. It is concludes that the imposed glide-slope constraint obviously affects the position profile, but that this does not necessarily mean the integrated state variables, i.e., velocity and acceleration, are affected in a similar fashion. Furthermore, in accordance with the results obtained by Szmuk and Açıkmeşe (2018), the optimal throttle profile in Fig. 6.9d deviates from the optimal profile $max-min-max$ often encountered in control theory literature. Despite this difference, the profile is clearly shows a sign of an optimal profile while adhering to the imposed



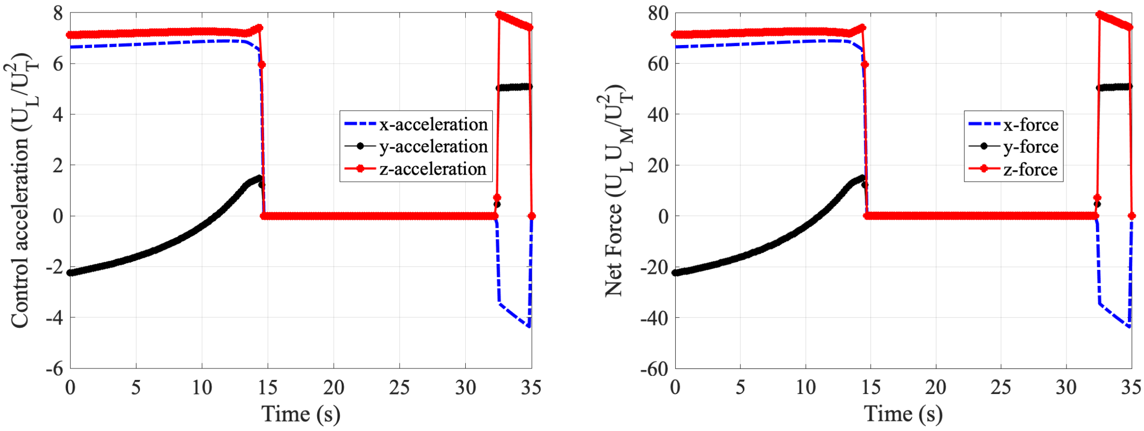
(a) Position profile

(b) Velocity profile

(c) Up-north trajectory

(d) Throttle time history

Figure 6.9: Successive convexification simulation results: position, velocity, trajectory and throttle setting.



(a) Acceleration profile

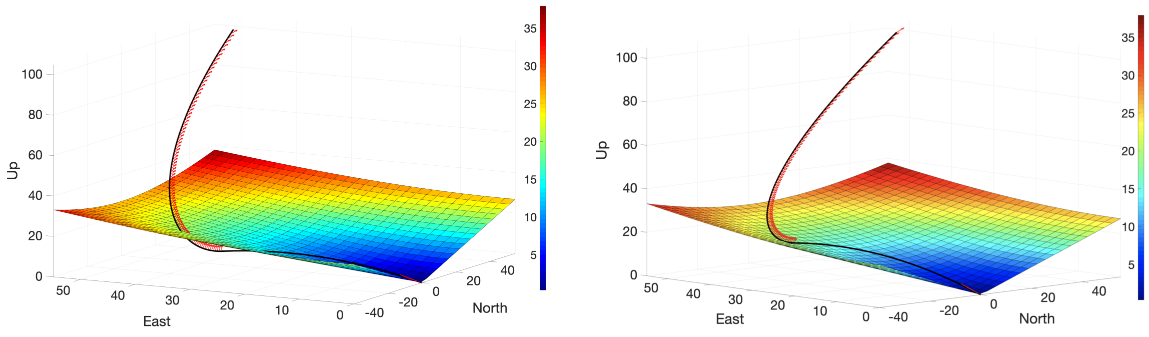
(b) Net force profile

Figure 6.10: Successive convexification simulation results: acceleration and net force time history.

normalized thrust magnitude bounds. Despite the fact that trajectory has been initiated with a reference trajectory which is dynamically infeasible, the algorithm managed to correct for this within 9 iterations. The imposed virtual control parameter plays a significant role here-in. Nonetheless, it should be mentioned that this algorithm is very well suitable for on-board implementation, the successive method does not provide full theoretical guarantee of convergence to an optimal solution in finite time. The applied method is however more robust compared to the golden search method discussed in Section 6.2.2. This is attributed to the fact that the time element is modelled as convex variable, thereby being an integral part of the framework. Further research is needed to establish whether the obtained time is indeed the theoretical optimal value, a matter beyond the scope of this thesis.

Additionally, the time history profiles of the acceleration and net force are provided in Figs. 6.10a and 6.10b, respectively. The acceleration profile in $z-$direction is in accordance with the behavior of the trajectory. That is, a coast burn is initiated at the start of the sequence. Subsequently, the spacecraft transfers into a gliding phase, with no acceleration in z-direction between 15s and 33s. Finally, to correct for this phase, a terminal burn is initiated. Accordingly, the net force profile shows similar behavior.

Notice that for visualization purposes two simulations have been run, one in which the glide-slope constraint is not enforced (Fig. 6.11a) and one in which the constraint is enforced (Fig. 6.11b). The inertia and moment properties of the MSL/Mars 2020 like vehicle are included in Appendix A. From the plots it becomes evident how the trajectory is affected by the constraint in three-dimensional Euclidean space. **The proper working of the glide-slope constraint is verified and the verification chapter is concluded.** A detailed analysis about varying the constraint in a step-wise fashion per mission segment will be discussed in Section 7.4.



(a) Glide-slope constraint not enforced

(b) Glide-slope constraint enforced.

Figure 6.11: Mars 3D simulation visualization in normalized three-dimensional Euclidean space.

7 Results

In this chapter of this research report the relevant simulation results are conferred, starting with the results of the active guidance, navigation and control system for precision landing on Mars. The nominal flight conditions for Mars EDL have been entrenched in the mission heritage chapter and are in line with realistic values for MSL and Mars 2020. For the active GNC system errors are induced to this nominal flight-path to assess the ability of the guidance system to maintain track to the designated landing site. In Section 7.1 the comparison is established between the nominal flight trajectory and the active guidance system. Subsequently, in the same section, conclusions are drawn regarding the proper working of the precision landing system. As part of the robustness assessment, the algorithm has been extended to become compatible with landing on Titan, incorporating parafoil dynamics within the high-density environment. The results are presented in Section 7.2, distinguishing between 3DoF and 6DoF high-fidelity dynamics. Likewise, the algorithm has also been extended to be capable of obstacle avoidance during powered descent. This includes avoidance of hills during descent and landing near rims of mountains or in valleys. These matters are outlined in 7.4. To maintain a clear overview of the simulations, a comprehensive simulation plan has been generated:

- **Guidance, navigation and control simulations.** Initially, the nominal trajectories are conferred after which the results for the active cases are presented. This allows for a proper distinction analyses between the two. It is denoted that initial state conditions and constrains are kept similar. A total of two cases are investigated: (1) plain powered descent sequence and (2) a descent trajectory with an enforced glide-slope constraint.

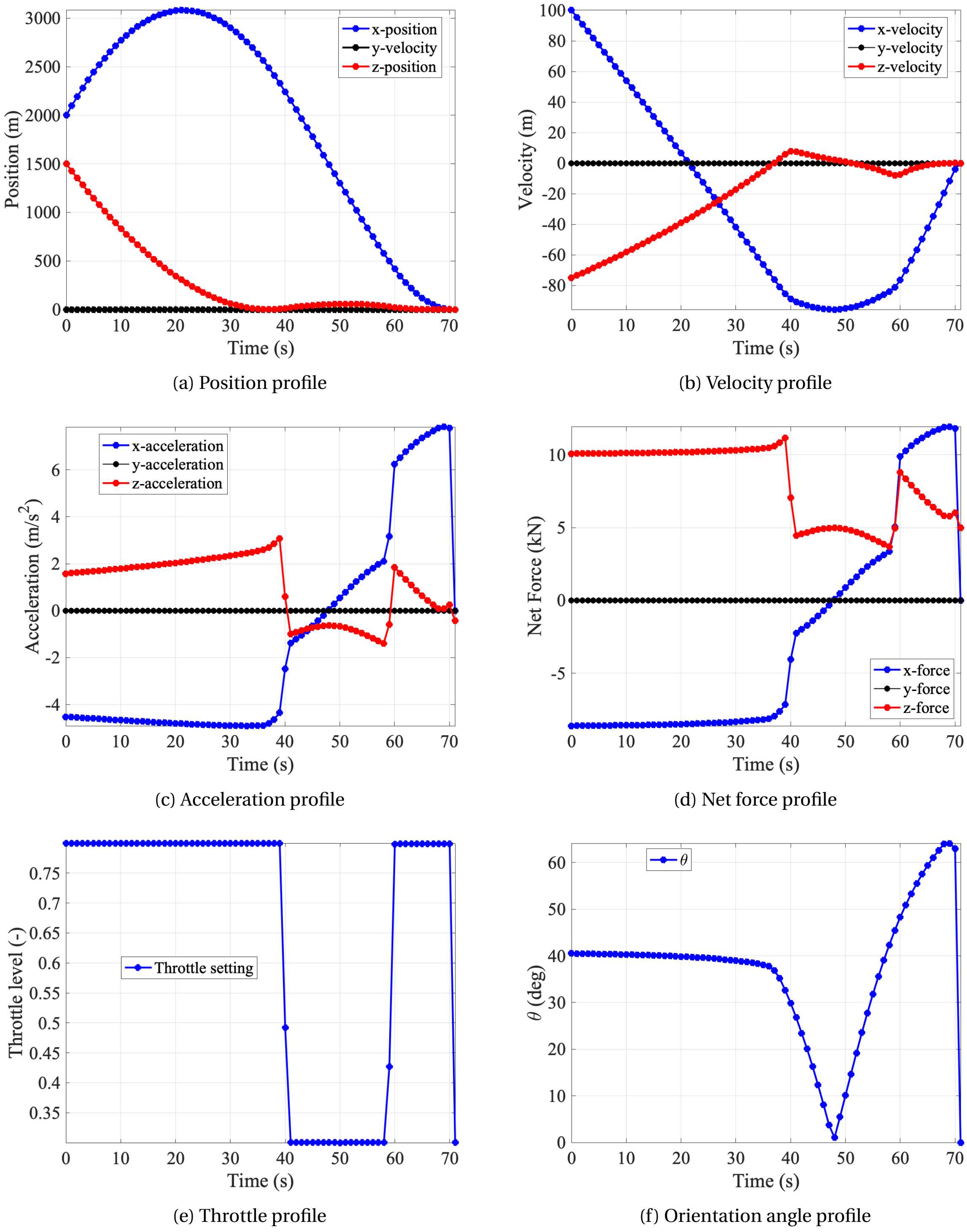
- **Titan precision landing simulations.** As part of the research conducted at NASA/JPL/Caltech, the algorithm has been extended to become compatible with landing in the Saturn's moon environment. Simultaneously, this allows for a robustness assessment of the algorithm. A total of three simulation cases are delineated. The first two assume three degree-of-freedom point-mass dynamics and are differentiated between (1) plain conditions and (2) wind conditions with enforced glide-slope constraint and pointing constraint. The latter (3) encompasses high-fidelity 6DoF dynamics with the conditions similar to (2). The analyses include a sensitivity analysis for parafoil descent guidance.

- **Obstacle avoidance guidance.** The last part of the simulations, as part of the robustness assessment of the designed algorithm, is the extension towards compatibility of obstacle avoidance. Included in the outlined simulations are two cases: (1) hill avoidance guidance and (2) planetary landing near mountain rims and in valleys.

It is denoted that all simulations have been carried using CVX in Matlab. For an extensive guide on the algorithm files, the reader is referred to Appendix D. However, it is again emphasized that these files are not part of the public domain and only available for use at NASA's Jet Propulsion Laboratory and to a limited extent at Delft University of Technology. The next chapter concludes the dissertation by presenting some conclusive remarks regarding the research, as well as recommendations to enhance future research and implementation of guidance algorithms for precision landing on Titan. This chapter concludes with powered descent sensitivity analyses, pushing the boundaries of the algorithm to evaluate its working in varying conditions.

7.1. Nominal and Closed-Loop Guidance, Navigation and Control

The nominal starting conditions for the Mars landing sequence have been established in the chapter on heritage systems based on MSL and Mars 2020 class vehicles, specifically in Tab. 2.3. In this section the nominal simulated precision landing trajectories are presented. These simulations come in two types: plain unconstrained simulation (Fig. 7.1) and glide-slope constrained simulation (Fig. 7.2).



(a) Position profile

(b) Velocity profile

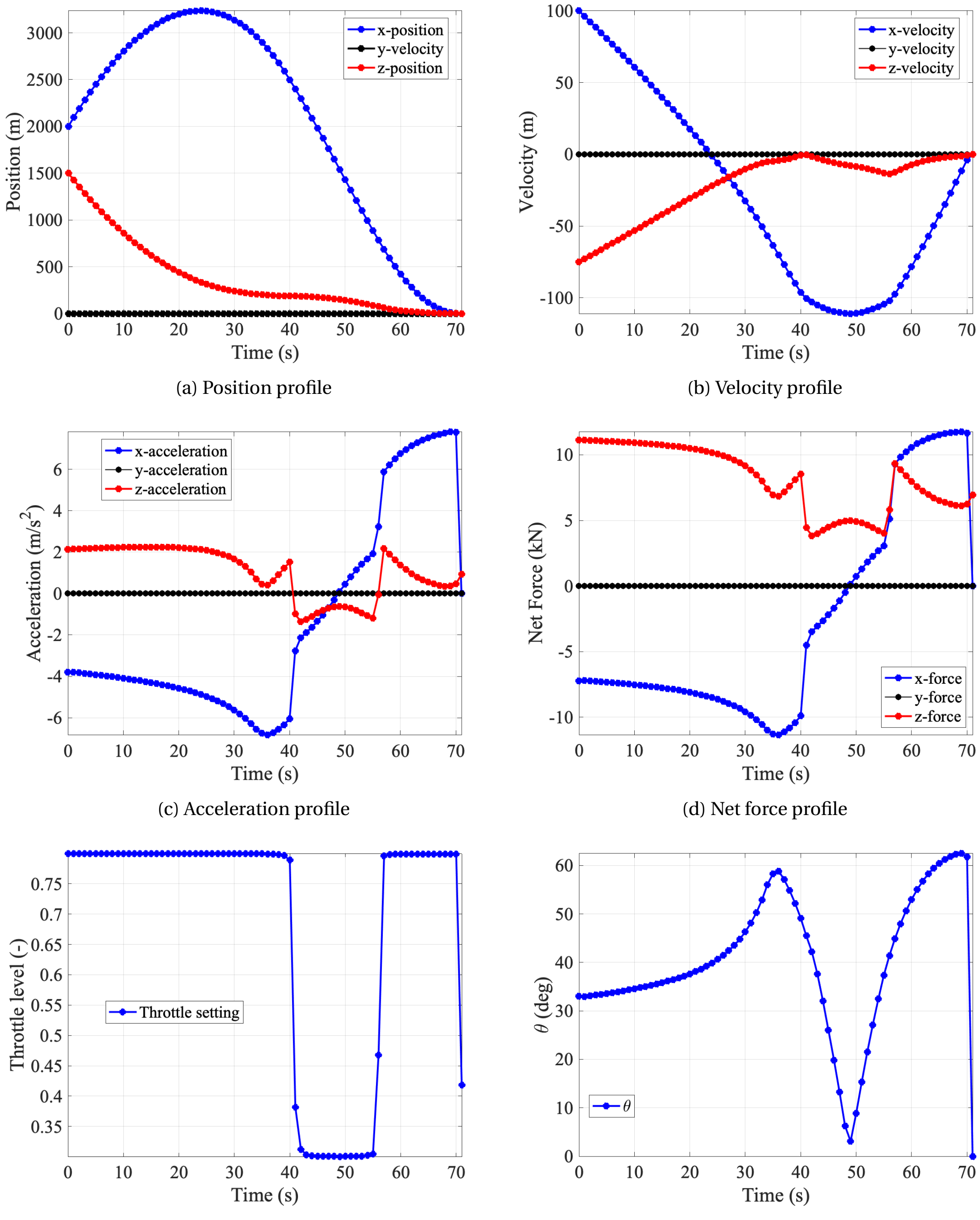
(c) Acceleration profile

(d) Net force profile

(e) Throttle profile

(f) Orientation angle profile

Figure 7.1: Nominal unconstrained Mars powered descent landing simulation. Time history of the position, velocity, acceleration, thrust angle, throttle setting and orientation angle.



(a) Position profile

(b) Velocity profile

(c) Acceleration profile

(d) Net force profile

(e) Throttle profile

(f) Orientation angle profile

Figure 7.2: Nominal glide-slope constrained Mars powered descent landing simulation. Time history of the position, velocity, acceleration, thrust angle, throttle setting and orientation angle.

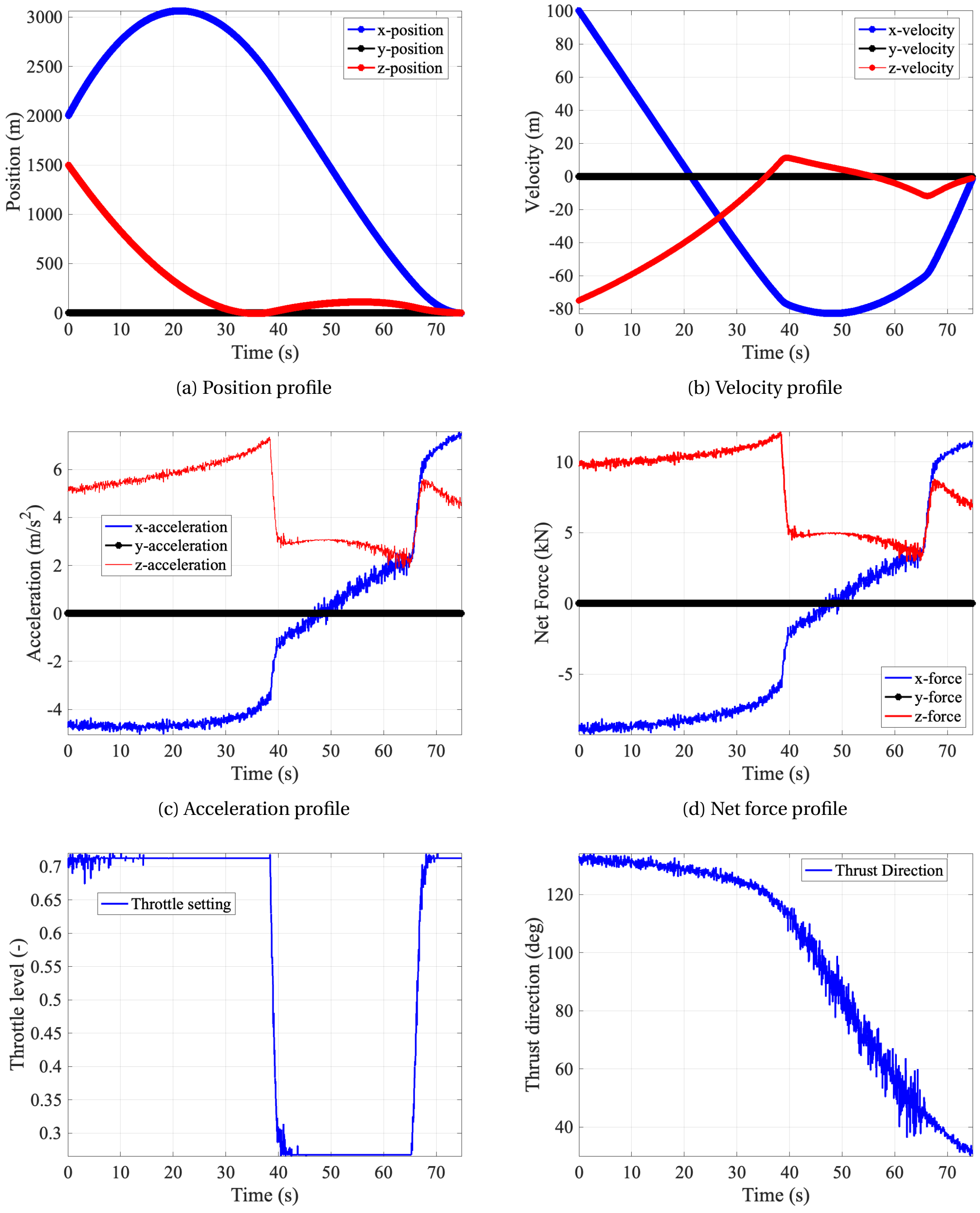
From the two simulations it becomes directly evident that the state and control constraints are satisfied throughout the entire simulated landing sequence. Furthermore, the soft landing optimal control problem is solved, which establishes the proper working of the precision landing technology. In Fig. 7.1 the time history of the up position (z-direction) clearly shows a subsurface flight after 30 seconds of flight. This is an obviously highly undesired phenomenon. Notice, this entails the algorithm adheres to the subsurface flight condition $r_z \geq 1$, as depicted in Fig. 7.1a. Through the enforcement of the glide-slope constraint this issue is resolved, as becomes evident from the position depiction in Fig. 7.2a. The velocity profiles in Fig. 7.1b and 7.2b are quite similar. Nonetheless, the magnitudes of the two plots differ significantly. This is attributed to the fact that the imposed constraint enforces more limits on the algorithm, causing the sequence to deviate more from the plain case. The acceleration (Fig. 7.1c and 7.2c), net force (Fig. 7.1d and 7.2d) and throttle profiles (Fig. 7.1e and 7.2e) differ mainly in discretization properties. The dynamics however for both cases is rather similar. The thrust angle profiles of Fig. 7.1f and 7.2f differ significantly, which is a direct consequence of the fact that the θ-angle is a function of the altitude (i.e., the glide-slope constraint affects this significantly). Conclusively, the nominal trajectories follow a desired optimal path, remain within the defined physical limits, adhere to the flight dynamics and, finally, establish planetary precision landing. Nonetheless, these nominal trajectories simulate trajectories assuming ideal conditions. Planetary landing sequences take place in conditions which are from ideal, especially when landing on Mars.

In Fig. 7.3 and 7.4 more realistic simulations are presented, i.e., closed-loop guidance and control within error inducing environments. In the next paragraph more information is provided regarding the simulations of the closed-loop GNC system. In the presented closed-loop guidance and control simulations several sources of errors have been introduced to assess the proper working of the designed GNC algorithm. This can be observed from the random nature of the acceleration and throttle plots. Notice that this section outlines a single run example, which is needed to outline the behavior. The Monte Carlo analysis associated to the closed-loop GNC will be discussed in Section 7.1.1. The figures are in correspondence with the earlier presented nominal cases to allow for one-on-one difference analyses. Therefore Fig. 7.3 outlines the results for the plain case (no constraints), while Fig. 7.4 includes the enforced glide-slope constraint. The only difference between the nominal and closed-loop plots is the last figure on the right-hand side. For the closed-loop guidance and control cases the thrust angle rather than the pointing angle has been included, to show the random variation of the thrust variable. The loop is based on Fig. 6.1 and includes all facets of planetary landing. More specifically, the following sources of errors have been introduced for a single case:

- Induced thrust misalignments through random variations of the convexified minimum and maximum thrust magnitude. The misalignment is given a $\Delta\rho = 5\%$ order with a standard deviation $\sigma_{\Delta\rho} = 1\%$.
- Thrust noise which alters the spacecraft acceleration. Notice that this differs from the thrust misalignment. The magnitude variation is given a $\Delta T = 100$ N order, with a standard deviation $\sigma_{\Delta T} = 25$ N.
- Navigation variations in terms of position, velocity and on-board fuel mass. Notice that this research assumes perfect state knowledge for reasons defined in Section 4.3. Introducing artificial navigation errors is merely included to assess guidance robustness. Standard deviations from the normal are considered such that $\sigma_r = 1.0$, $\sigma_v = 0.5$, and $\sigma_m = 0.5$ for the navigation loop.

To close the full GNC loop a PID controller is coupled to the guidance algorithm. Section 7.1.1 outlines some Monte Carlo analysis batches, while the control design overview is presented in Section 7.1.2. For specifics regarding the design the reader is referred to Section 6.1 and for outline of the controller type the reader is referred to Section 4.4. At last, state propagation is provided through a fourth-order Runge-Kutta integrator, of which the specifics are included in Appendix C.

From the two sets of figures it becomes clear that despite the introduction of several sources of error, the algorithm still manages to guide the spacecraft to the designated target through a precision landing sequence. The closed-loop GNC simulations produces similar results compared to the counter nominal cases for both the position profiles in Fig. 7.3a and 7.4a, and the velocity profiles in Fig. 7.3b and 7.4b. This is expected, as the algorithm should guide the spacecraft to the nominal path despite induced errors. The convex optimization method is robust method for guaranteeing soft landing. The control and state constraints are likewise adhered to. What is remarkable is the fact that despite the introduction of random errors which affect the acceleration, net force, throttle and thrust angle profiles, the algorithm manages to fly the spacecraft in a path very similar to the nominal case. This observation holds for both the plain case and the glide-slope constrained case. In Fig. 7.5 the in-plane trajectories for both landing sequences are provided. The mass depletion dynamics figures, i.e., the variation of mass due fuel depletion, are given for both cases in the same figure. From the trajectory plots it becomes more evident what important role the glide-slope constraint plays. The mass depletion dynamics figures are rather similar and both remain within the desired physical limit. These observations conclude the section on closed-loop guidance and control. In the next section the results for precision landing on Titan are provided.



(a) Position profile

(b) Velocity profile

(c) Acceleration profile

(d) Net force profile

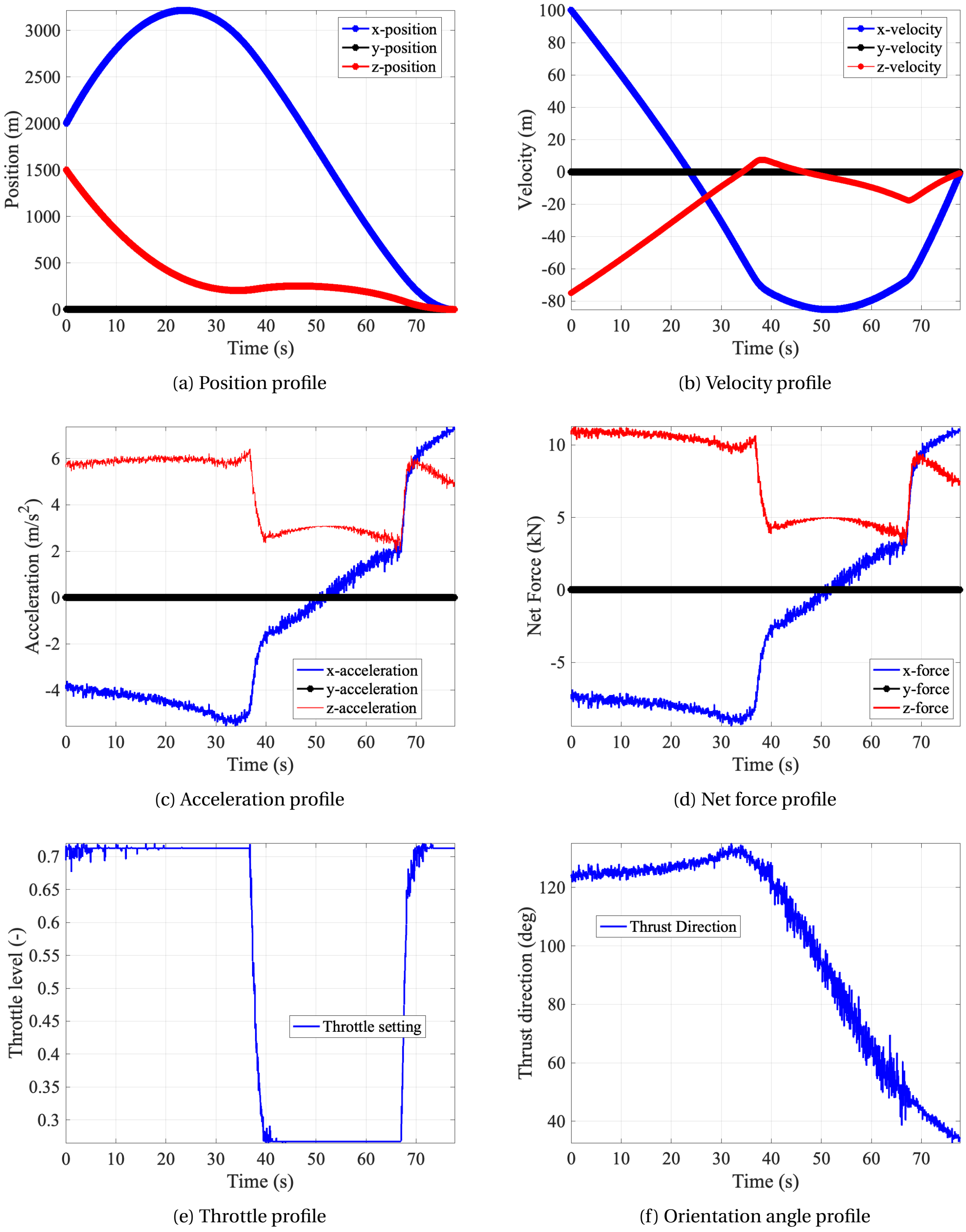
(e) Throttle profile

(f) Orientation angle profile

Figure 7.3: Closed-loop GNC unconstrained Mars powered descent landing simulation. Time history of the position, velocity, acceleration, thrust angle, throttle setting and thrust angle.

7.1.1. Monte Carlo Analysis

To provide insight into the robustness of the algorithm, this section outlines a Monte Carlo analysis of six test cases. For all cases a sample number $n = 500$ has been used. Also, a normal random type of distribution has been outlined using the Matlab 'normrnd(μ, σ)' command to generate the simulated errors. For the exact values of the order of change and the respective standard deviations, the reader is referred to the discussion in the previous section. In the previous sections of this chapter it has been conferred that the algorithm



(a) Position profile

(b) Velocity profile

(c) Acceleration profile

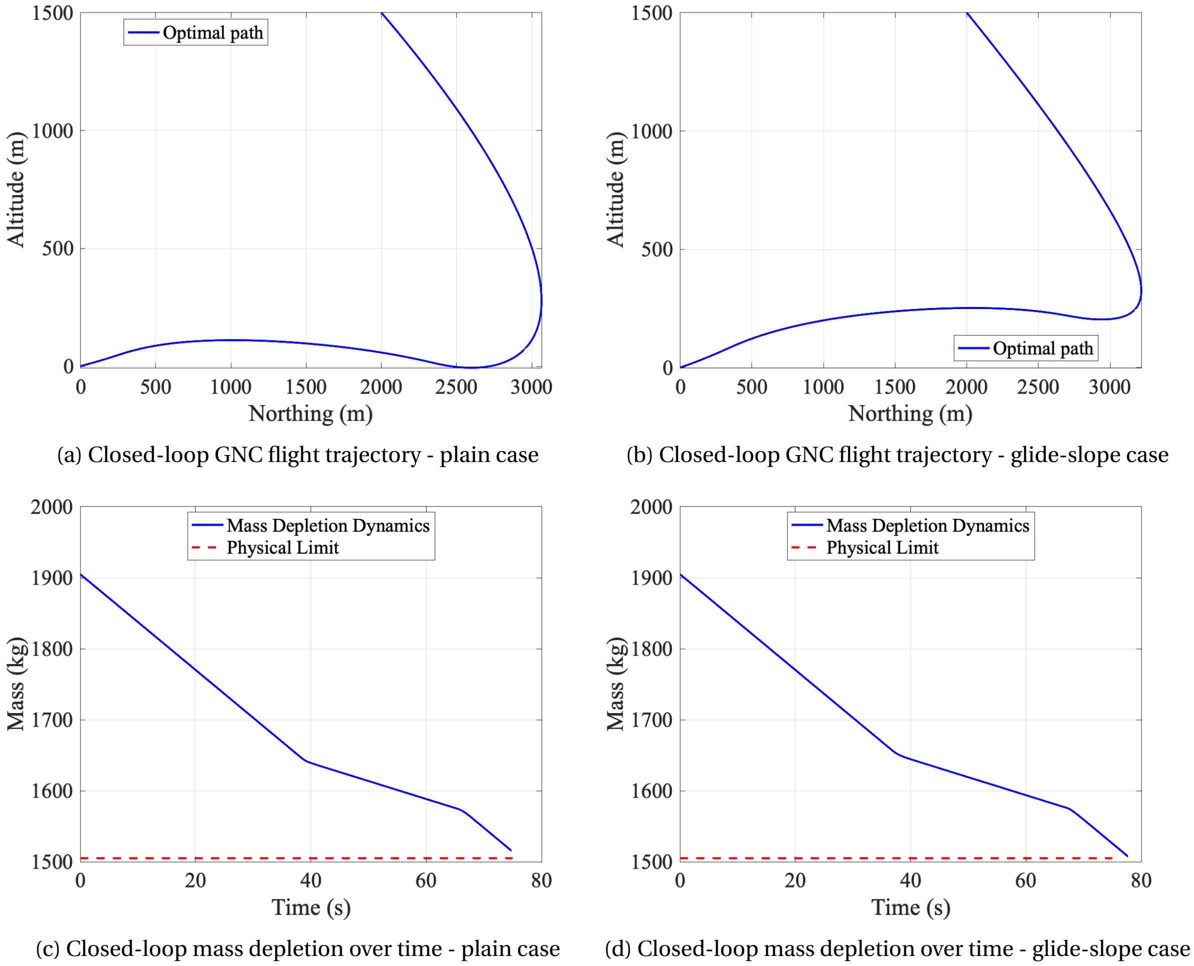
(d) Net force profile

(e) Throttle profile

(f) Orientation angle profile

Figure 7.4: Closed-loop GNC glide-slope constrained Mars powered descent landing simulation. Time history of the position, velocity, acceleration, thrust angle, throttle setting and thrust angle.

manages to stay well within the mission constraint limits, and is also able to robustly tackle induced errors (as became evident in the single test case). Nonetheless, an aspect that needs to analyzed is the behavior of fuel mass consumption due to the induced errors, which is what this Monte Carlo analysis focuses on (theoretical outline provided in Appendix C.3). The following six alterations have been considered:



(a) Closed-loop GNC flight trajectory - plain case

(b) Closed-loop GNC flight trajectory - glide-slope case

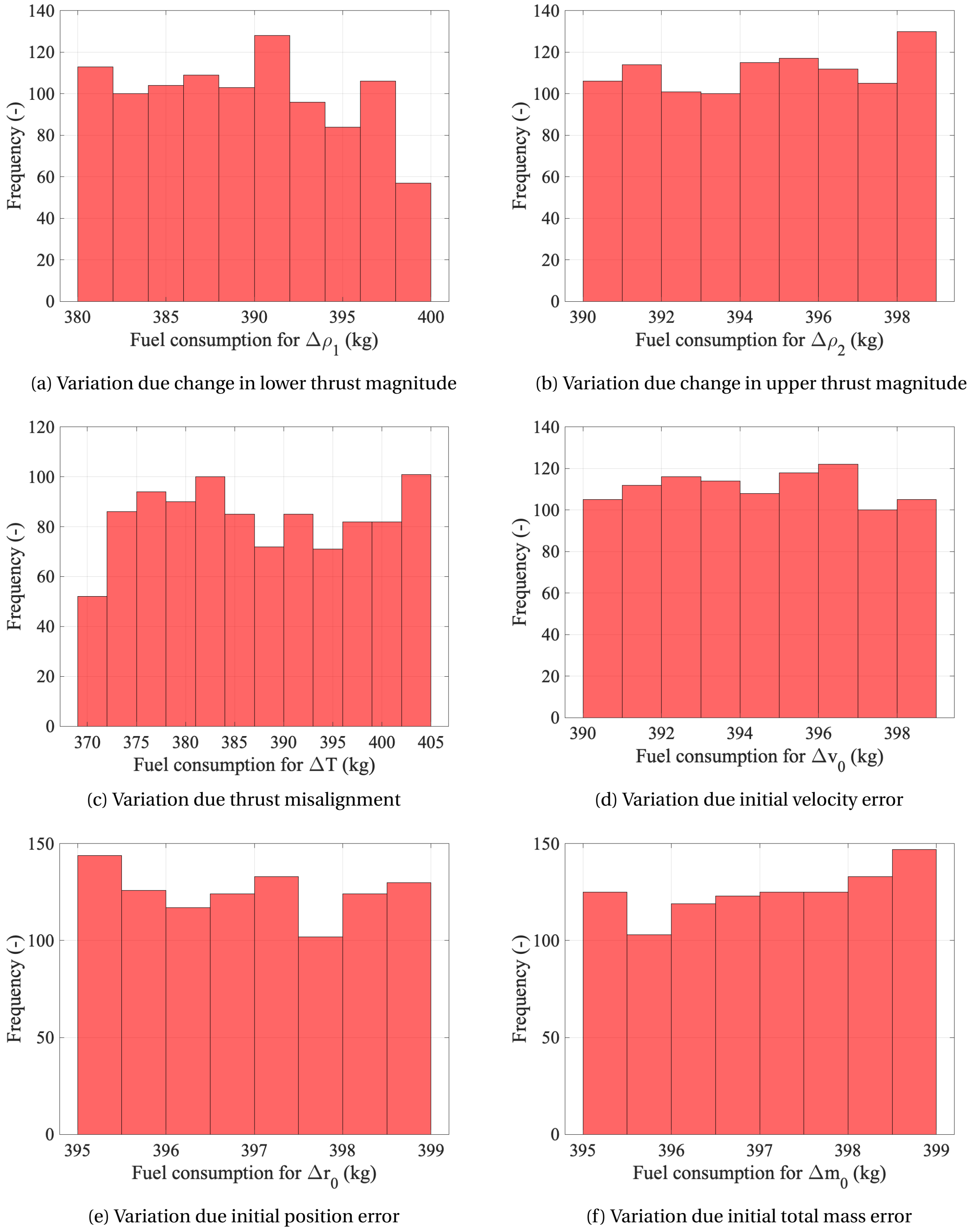
(c) Closed-loop mass depletion over time - plain case

(d) Closed-loop mass depletion over time - glide-slope case

Figure 7.5: Closed-loop GNC trajectory plot plain (top-left) and glide-slope constrained (top-right). Associated mass depletion over times figures included beneath.

1. Variation of lower thrust limit → Fig. 7.6a.
2. Variation of upper thrust limit → Fig. 7.6b.
3. Thrust misalignment → Fig. 7.6c.
4. Error in initial position → Fig. 7.6e.
5. Error in initial velocity → Fig. 7.6d.
6. Error in initial mass → Fig. 7.6f.

The results of the run test cases are presented in Fig. 7.6 as histograms containing 500 samples. It is recalled that the total available fuel mass was established to be 400 kg. Overall, the histograms have a uniform distribution, indicating consistency in data. In Fig. 7.6a the histogram for altering the lower thrust magnitude is presented. Interestingly, the variations are quite extreme for this case. The algorithm manages to maintain within the desired limit for 440 samples. Nonetheless, for about 60 samples the maximum fuel bound is violated. This means that the trajectory switches to minimizing the landing error. This is mainly attributed to the fact that a too high lower bound on the thrust significantly increases fuel consumption, leading to this undesired phenomenon. It is therefore advised for actual flight cases to maintain thrust bounds within reasonable limits, both upper and lower. A similar conclusion is drawn when inducing thrust misalignments, as depicted in Fig. 7.6c. For all the other test cases the algorithm adheres to the on-board fuel limit. Changes in velocity have a more significant affect on fuel consumption than changes in position and velocity. This is shown in Figs. 7.6e, 7.6d and 7.6f, respectively. Nonetheless, the variations are less extreme compared to the thrust bound cases. This concludes the algorithm robustness assessment based on Monte Carlo analyses, with a main conclusion that despite the undesired deviation in for a marginal number of samples, the algorithm obtains a very high success rate when inducing errors of random nature.



(a) Variation due change in lower thrust magnitude

(b) Variation due change in upper thrust magnitude

(c) Variation due thrust misalignment

(d) Variation due initial velocity error

(e) Variation due initial position error

(f) Variation due initial total mass error

Figure 7.6: Monte Carlo histogram distributions for various test cases.

7.1.2. CONTROL SYSTEM DESIGN

In this section an outlines is provided regarding the control system that has been used to close the guidance and control loop. It is again emphasized that this research mainly focused on the guidance aspect of GNC for planetary precision landing. The loop has been closed using a PID controller, which is considered to be a sophisticated control method. Yet, there are of course other more robust methods, such as the H-infinity methodology. A matter beyond the scope of this research, as a PID controller suffices for the proposed analyses. On a side note it is stated that despite the basic principles of PID control, it is often used for actual flight implementation as it provides robust results with ease of design. For this research a PID controller has been used with gain values: $K_P = 0.875$, $K_I = 0.75$, and $K_D = 0.125$. These values have been established based on a trial-and-error fine tuning of the controls: position, acceleration and velocity, respectively. These controls are multiplied by the actual time response mass of the vehicle. The step response plot in Fig. 7.7 has been generated in Matlab using the step command.

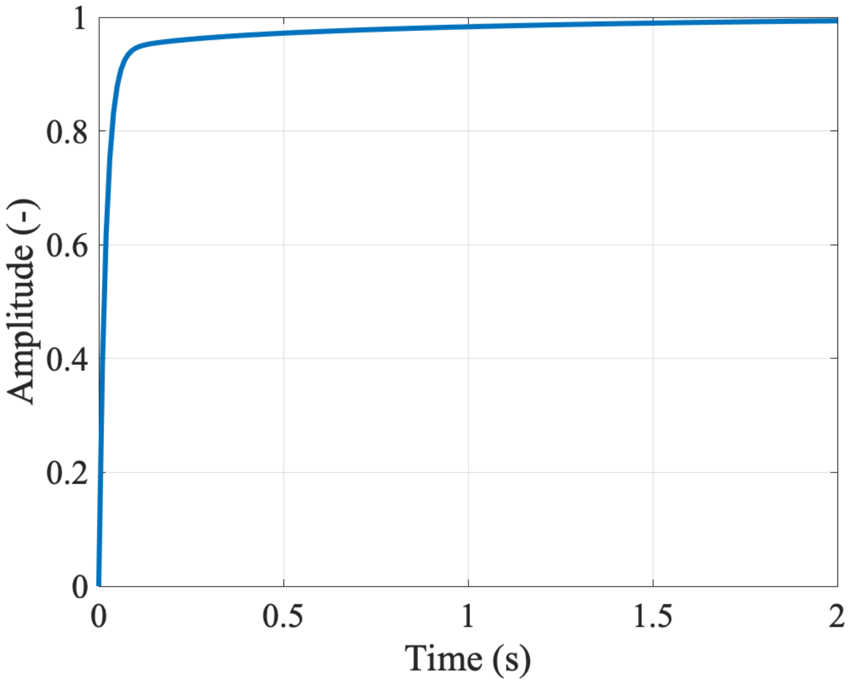


Figure 7.7: Bode diagram (magnitude and phase) of the PID controller.

From the figure it becomes evident that the step response has no overshoot. Please notice that only the first two seconds of the response have been presented, as the response after remained constant. Also, there is no presence of a steady-state error. The rise time appears to be very fast, which is desired. Based on this plot it is concluded that the closed-loop system is stable. For the problem at hand, this analysis suffices to analyze the precision landing sequence. It is advised that for actual flight implementations, a more detailed control analysis is carried out. This should include bode plots, gain tuning and root locus plots, amongst others.

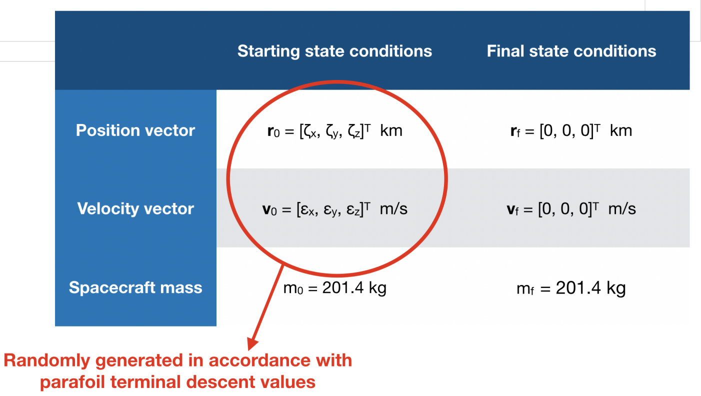
7.2. Precision Landing on Titan

In this section the designed algorithm is extended to become suitable for landing a powered parafoil on Titan. The conferred results are based on both three-degree-of-freedom point-mass simulations for lossless convexification and six-degree-of-freedom successive convexification simulations. The conditions for landing on Titan are repeated in Tab. 7.1. The exact order of magnitudes of the initial parafoil descent states have been provided in Section 2.3.2. In Section 7.2.1 the results of two types of point-mass simulation are presented: an in-plane manoeuvre with the glide-slope constraint not enforced and a 3D manoeuvre with the glide-slope constraint enforced. For both cases the starting and terminal state conditions (position, velocity and mass) are in accordance with relevant literature to allow for one-on-one verification. These starting conditions are presented in Tab. 2.3. Section 7.2.2 on the other hand outlines the rigid-body simulations based on six-degree-of-freedom dynamics. In both sections conclusive remarks are conferred regarding the robustness of the algorithm and the establishment of soft landing on Titan.

7.2.1. Point-mass simulations: three-degree-of-freedom dynamics

The results for the Titan 3D manoeuvre (no glide-slope constraint nor wind) are presented in Figs. 7.8 and 7.9. From the figures it becomes evident that the algorithm adheres to the soft-landing constraint (both in 2D and 3D space). It also becomes clear that the surface is reached quite fast, as a consequence of leaving out the

Table 7.1: Titan landing initial and terminal state conditions (randomly generated) in accordance with Quadrelli et al. (2019)



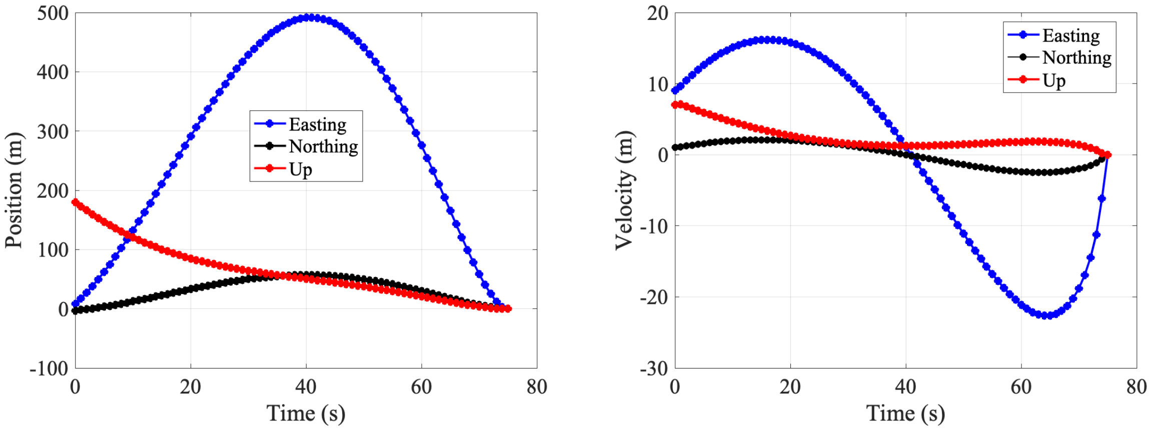
	Starting state conditions	Final state conditions
Position vector	$\mathbf{r}_0 = [\zeta_x, \zeta_y, \zeta_z]^T$ km	$\mathbf{r}_f = [0, 0, 0]^T$ km
Velocity vector	$\mathbf{v}_0 = [\varepsilon_x, \varepsilon_y, \varepsilon_z]^T$ m/s	$\mathbf{v}_f = [0, 0, 0]^T$ m/s
Spacecraft mass	$m_0 = 201.4$ kg	$m_f = 201.4$ kg

Randomly generated in accordance with
parafoil terminal descent values

slope constraint. Notice that these simulations assume ideal conditions (no Titanian winds). Interestingly, the easting direction path deviates largely, both in terms of position and variation of velocity. This is observed in Fig. 7.8 and attributed to the lack of control authority in that direction. Also, the algorithm contains a constraint to land against the wind. There reason for including this constraint is to make sure that the parafoil does not get into stall conditions, thereby endangering mission safety. Therefore, during terminal descent the direction is changed aggressively. Likewise for powered descent, the convex optimization algorithm is capable of doing this. In Fig. 7.9 this phenomenon can be observed. Also included in this figure are a 3D plot of several random state simulations to assess the algorithm's capabilities to handle dispersions of varying nature. Conclusively, the algorithm for landing on Titan assuming ideal conditions works properly. The next paragraph outlines the landing manoeuvre assuming wind and a glide-slope constraint.

The results of the 3D manoeuvre with the glide-slope constraint enforced and wind inclusion are presented in Figs 7.10 - 7.13. The figures contain the time histories of the position and velocity and the trajectories in 2D and 3D Euclidean space. Furthermore, the control history of the pull is included and the aerodynamic accelerations in all three directions. At last, the aerodynamic angles of relevance, the angle-of-attack and side-slip angle are plotted against time. In the next paragraph the simulation results are discussed.

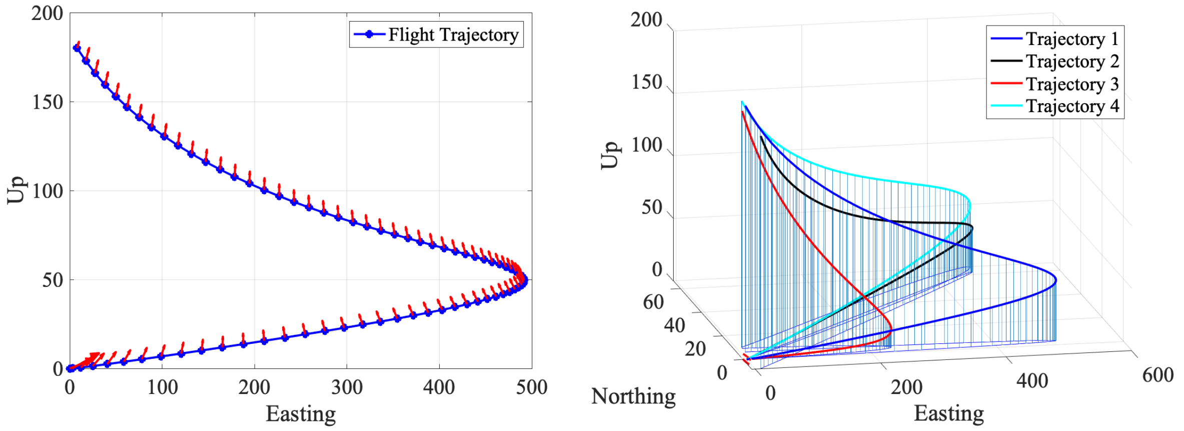
From the control history it is derived that the guidance algorithm tells the spacecraft to glide in the first part of the sequence (zero-control effort) after which the control effort is maximized. The aerodynamic angles maintain within acceptable limits and the final position and velocity vector is at the desired zero-magnitude level (soft-landing), as becomes evident from Figs. 7.12b. The inclusion of the glide-slope constraint becomes clear in both the 2D and 3D plot. The sequence remains well above ground till the termination of the



(a) Position profile

(b) Velocity profile

Figure 7.8: Titan point mass simulation (3DoF) with the glide-slope constraint not enforced and no wind. Position and velocity time histories of the optimal in-plane trajectory.



(a) Trajectory up-east

(b) 3D trajectories

Figure 7.9: Titan point mass simulation (3DoF) with the glide-slope constraint not enforced and no wind. Optimal in-plane and out-of-plane trajectories.

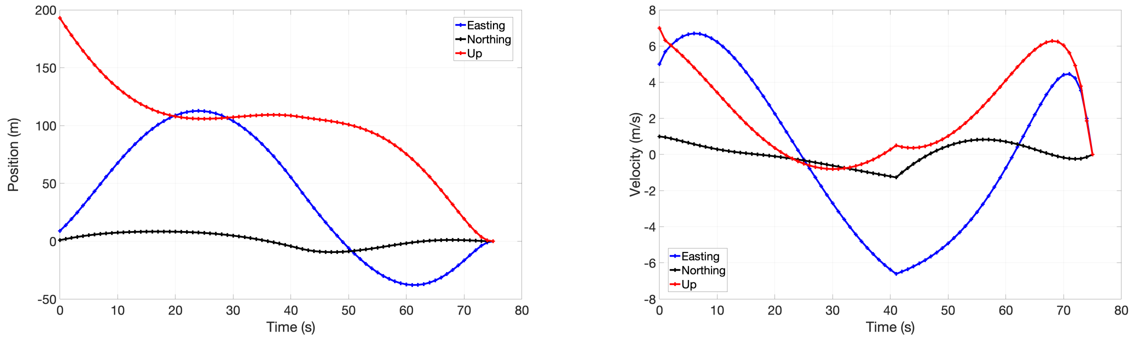
sequence. It also becomes clear that the inclusion of the glide-slope constraint improves the landing constraint against the wind. This is observed from the easting position of the position profile, which changes sign after approximately 50 s to switch direction. Note, the wind magnitudes used in this simulation have been conferred in Section 3.7. The 2D and 3D trajectory plots in Fig. 7.11 further entrench the truth of this observation. Like for the plain case, the control authority in easting direction is seen to be rather limited. As expected, the lift acceleration is the largest contributor of the motion, followed by the drag. The sequence requires a high angle-of-attack profile throughout almost the entire sequence. This can be compared to a lifting body during planetary entry. The side-slip angle profile shows low values of the angle, which is in accordance with earlier simulations carried out at the Jet Propulsion Laboratory. Conclusively, the algorithm is extended to work successfully for point mass simulations with 3DoF dynamics. In the next section the rigid body (6DoF) simulations are presented.

7.2.2. RIGID-BODY SIMULATIONS: SIX-DEGREE-OF-FREEDOM DYNAMICS

The results for the six-degrees-of-freedom Titan parafoil descent simulation are presented in this section. In this section a single example is given based on the following starting conditions:

$$\mathbf{r_0} = [9, -7, 183]^T \text{m} \quad \text{and} \quad \mathbf{v_0} = [9, 0, -8]^T \text{m/s} \tag{7.1}$$

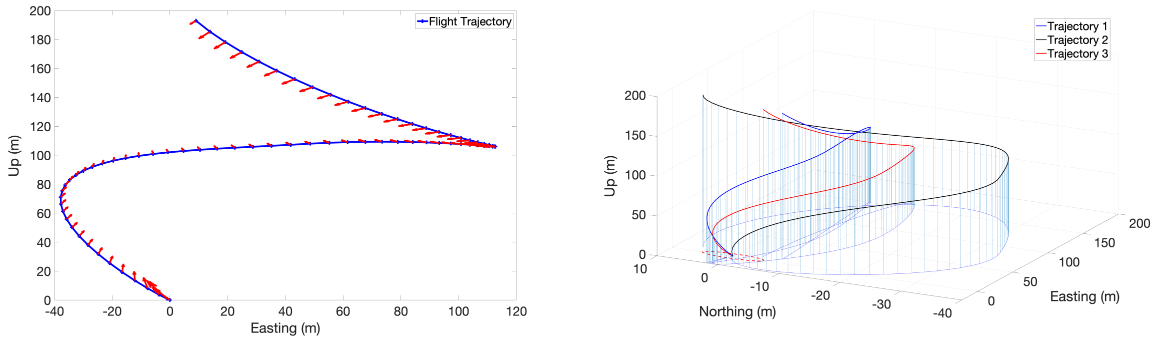
The next section will outline a detailed sensitivity analysis through the variation of governing variables. From Figs. 7.14 it can be observed that the positioning and velocity terminate at zero magnitude, hence the soft-



(a) Position profile

(b) Velocity profile

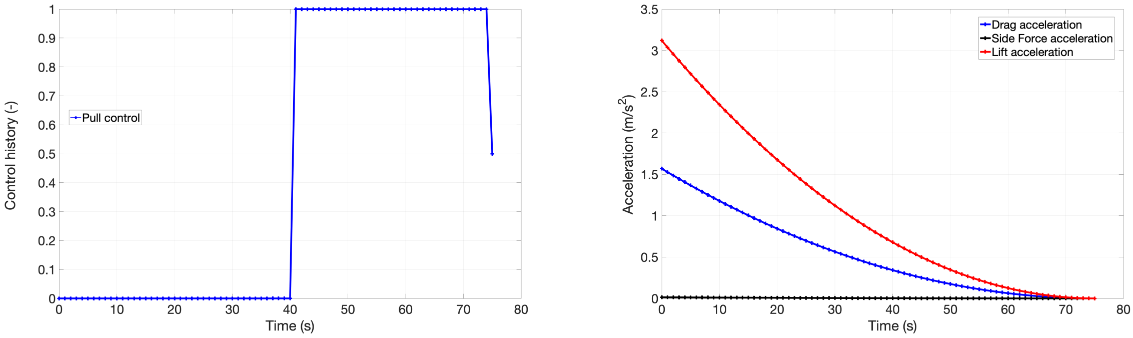
Figure 7.10: Titan point mass simulation (3DoF) with the glide-slope constraint enforced and wind included. Position and velocity time histories of the optimal in-plane trajectory.



(a) Trajectory up-east

(b) 3D trajectories

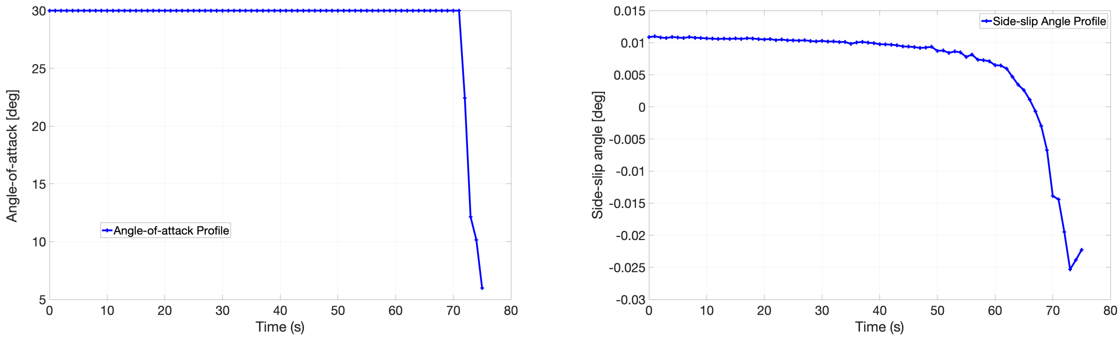
Figure 7.11: Titan point mass simulation (3DoF) with the glide-slope constraint enforced and wind included. Optimal in-plane and out-of-plane trajectories.



(a) Control profile

(b) Aerodynamic accelerations

Figure 7.12: Titan point mass simulation (3DoF) with the glide-slope constraint enforced and wind included. Time histories of the control normalized pull and aerodynamic accelerations.

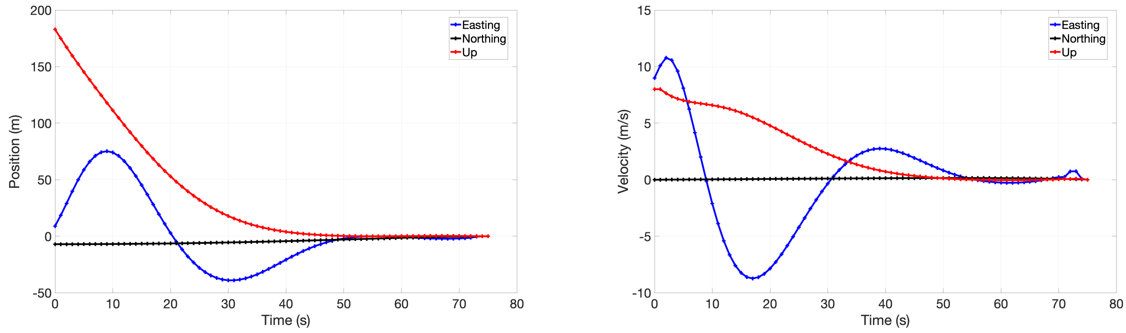


(a) Angle-of-attack profile

(b) Sideslip angle profile

Figure 7.13: Titan point mass simulation (3DoF) with the glide-slope constraint enforced and wind included. Time histories of the aerodynamic angles: angle-of-attack and side-slip angle.

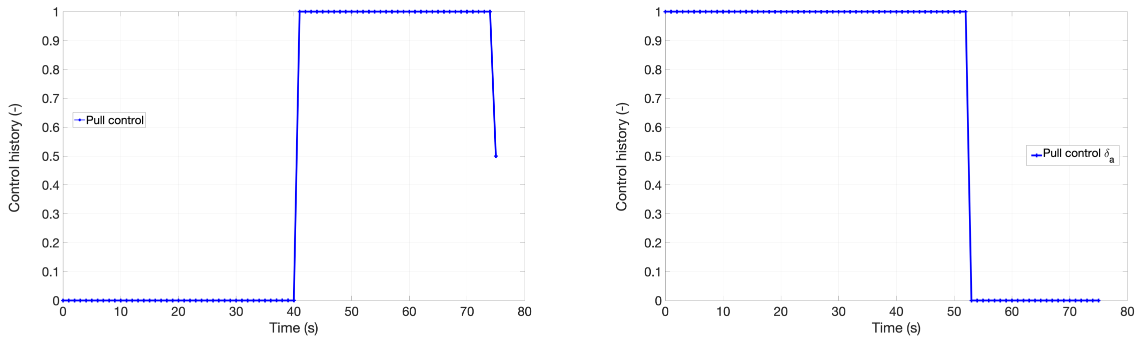
landing simulation is successful. Both control authorities (symmetric and asymmetric deflections) remain with the desired normalized values between 0 and 1, as becomes evident from Fig. 7.15. Interesting remark is that the control authorities seem to take turns in controlling the parafoil to the target. From the attitude rate plot (Fig. 7.16) it becomes evident that control authority in y-direction is anew rather limited. This is in accordance with the established equations, as the deflections have no control over the force in y-direction (side force) nor the moment in y-direction (pitching moment). This issue could be tackled through inclusion of a propulsion system. The quaternion attitude representation is restricted to have one unit quaternion of one, which is the case as deduced from Fig. 7.16b. Furthermore, Figs. 7.17a and 7.17b, show that the



(a) Position profile

(b) Velocity profile

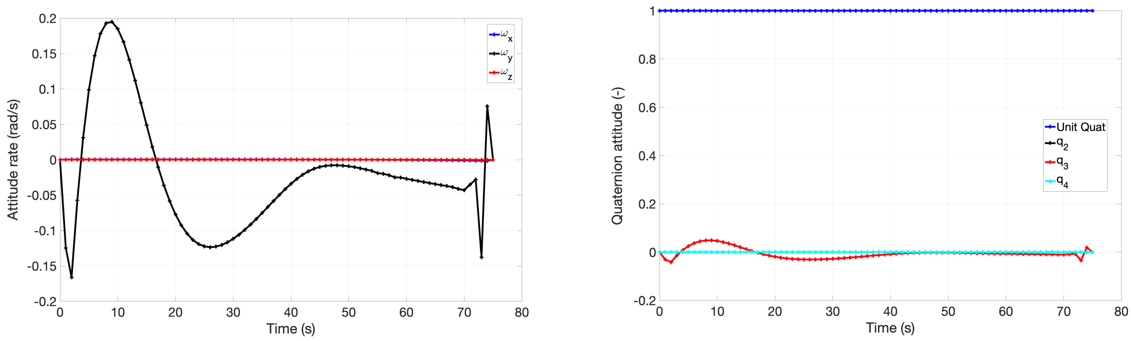
Figure 7.14: Titan rigid body simulation (6DoF) with the glide-slope constraint enforced and wind. Time history of the position and velocity states.



(a) Symmetric control profile

(b) Asymmetric control profile

Figure 7.15: Titan rigid body simulation (6DoF) with the glide-slope constraint enforced and wind. Time history of the control deflections.

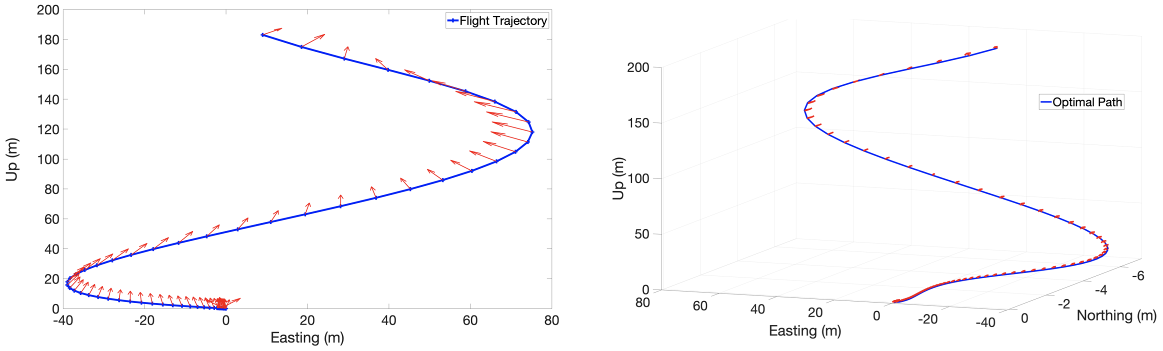


(a) Attitude rate profile

(b) Quaternion attitude profile

Figure 7.16: Titan rigid body simulation (6DoF) with the glide-slope constraint enforced and wind. Time history of angular rates and quaternion angles.

sequence starts off by gliding in opposite direction to the wind, after which a switch is made to use the wind to benefit. At termination, the spacecraft is turned back in opposite direction of the wind, which is an imposed constraint on the system. This is included to allow the parafoil to use the wind to slow down. From this single working example it is concluded that the algorithm is very well suitable for landing a parafoil on Titan, thereby successfully ensuring precision landing. Nonetheless, to assess the robustness of the algorithm, a sensitivity analysis has been carried out, of which the results are conferred in the subsequent section.



(a) Trajectory up-east

(b) 3D trajectory

Figure 7.17: Titan rigid body simulation (6DoF) with the glide-slope constraint enforced and wind included. Optimal in-plane and out-of-plane trajectories.

7.3. SENSITIVITY ANALYSIS PARAFOIL DESCENT

This section outlines a discussion on the sensitivity analyses that has been coupled to the design and simulation approach of the parafoil descent cases. This sensitivity is necessary because the parafoil design parameters and initial flight conditions have been engineered, both no flight reference exists. Furthermore, an incorporation of a sensitivity analysis provides great insight into how several conditions may effect the flight trajectory, the control effort and the attitude. Through this process the robustness of the parafoil guidance algorithm is typically evaluated. This insight is of vital importance for the understanding of precision landing trajectories and how design parameters influence the accuracy. Based on the simulation results it is even possible to point out possible risk factors that influence the landing precision or the flight trajectory in general. The parameters that are included in the sensitivity study have been captivated in Tab. 7.2. Notice that these values have been altered for a total of 6 cases: adding 5%, 10% and 20%, and reducing 5%, 10% and 20% to the nominal case. While the percentages are relative, absolute values have been used in the simulations.

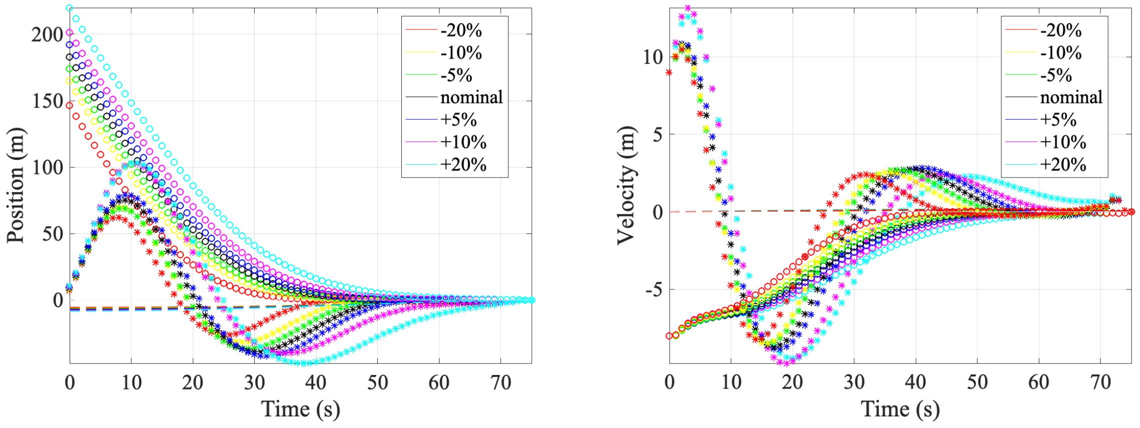
Table 7.2: Parameters of sensitivity and their respective percentages.

Parameter	Description	Sensitivity percentage
$\mathbf{r}_0$	initial position	± 20%
$\mathbf{v}_0$	initial velocity	± 20%
C_L	total lift coefficient	± 20%
C_D	total drag coefficient	± 20%
$\mathbf{r}_0$ and $\mathbf{v}_0$	coupled position and velocity	± 20%
C_L and C_D	coupled lift and drag coefficients	± 20%

The results are now conferred for each scenario, presenting time histories of the position, velocity, throttle setting and the trajectory. These entities have been chosen as they provide the most valuable simulation information. In Fig. 7.18 - 7.33 all the simulation results are presented. To maintain a clear discussion, this section outlines a short discussion and some conclusions per varied variable:

- **Initial position sensitivity:** presented in Figs. 7.18 - 7.20. From the position profile the conclusion is drawn that despite the imposed variation, the algorithm behaves in a similar fashion. The trajectories presented in Fig. 7.19a show very similar behavior. On the other hand, the total control history varies extremely with variation in initial position. This effect is strongest for a 20%. While the control still adheres to the imposed bound between 0 and 1, the fluctuations are large. The switch logic might get too excessive for on-board control. The algorithm no longer follows a distinct max-min-max profile. Rather, it seems as if noise has been introduced despite the fact that this is not the case. The quaternion and attitude profiles show stable variations just like the position and velocity profiles.
- **Initial velocity sensitivity:** presented in Figs. 7.21 - 7.23. The observations are very much alike for the previous case. Additionally, it is denoted that the control fluctuation to some extend is extremer and present for multiple flight cases. Nonetheless, the control bounds stay within the imposed limit and all trajectories show signs of stability.

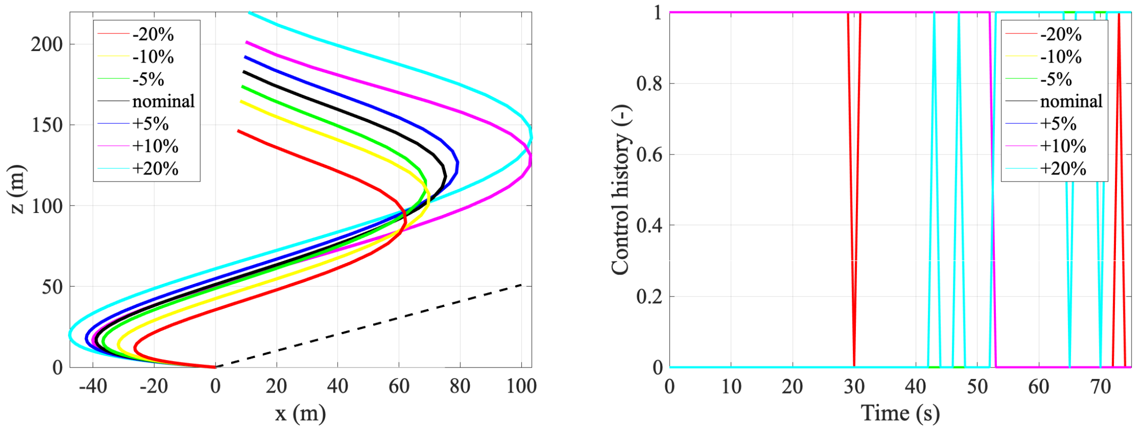
- **Lift coefficient sensitivity:** presented in Figs. 7.24 - 7.26. The lift coefficients have a very strong effect on both position and velocity. This is of course expected, as the coefficients govern the aerodynamics. Interestingly, the trajectories are very similar however, with a single major outlier for the 20% case. Control profiles show signs of extreme fluctuation, probably due to the heavy change in attitude, as can be observed in Figs. 7.26b and 7.26a.
- **Drag coefficient sensitivity:** presented in Figs. 7.27 - 7.29. The drag coefficients show a similar yet stronger effect compared to the lift cases. Interestingly, the trajectory profile 7.28a is rather similar to lift counterpart. Control fluctuations are likewise extreme, but within the imposed limits. This is again a consequence of the less stable attitude for the extremer cases.
- **Coupled initial position and velocity sensitivity:** presented in Fig. 7.30 and 7.31. Coupling these two parameters leads to limited variation. Results are quite robust with overall very similar results. The conclusion is drawn that the algorithm is very well capable of guiding the parafoil to the target for a wide range of starting state conditions in terms of position and velocity.
- **Coupled lift and drag coefficient sensitivity:** presented in Fig. 7.32 and 7.33. As expected, for this case variations are extreme for position, velocity, attitude rate and quaternion. The parafoil has been designed assuming engineered aerodynamic parameters. Heavy alterations in these governing parameters leads to very different profiles. For actual flight implementation the most optimal parameters would of course be considered.



(a) Position profile

(b) Velocity profile

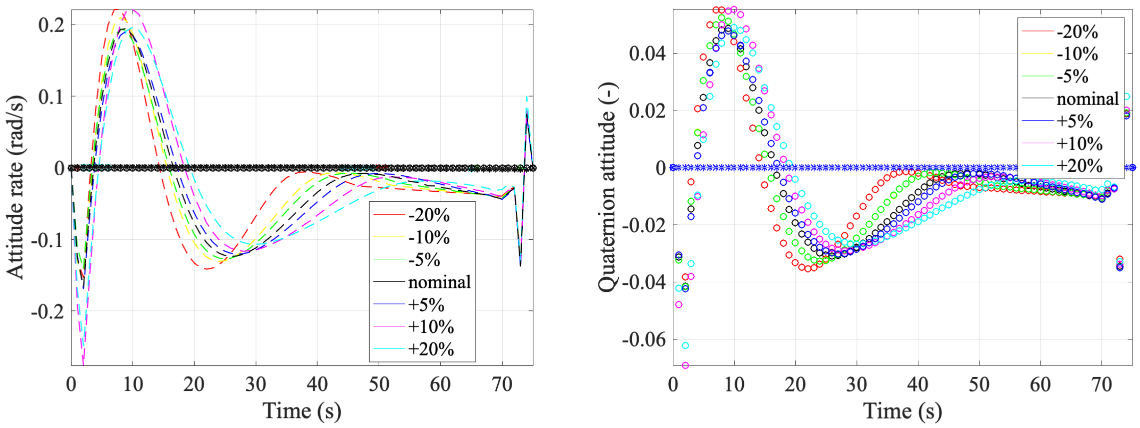
Figure 7.18: Parafoil descent position and velocity time histories due to initial position sensitivity of ± 20%.



(a) Trajectory

(b) Control profile

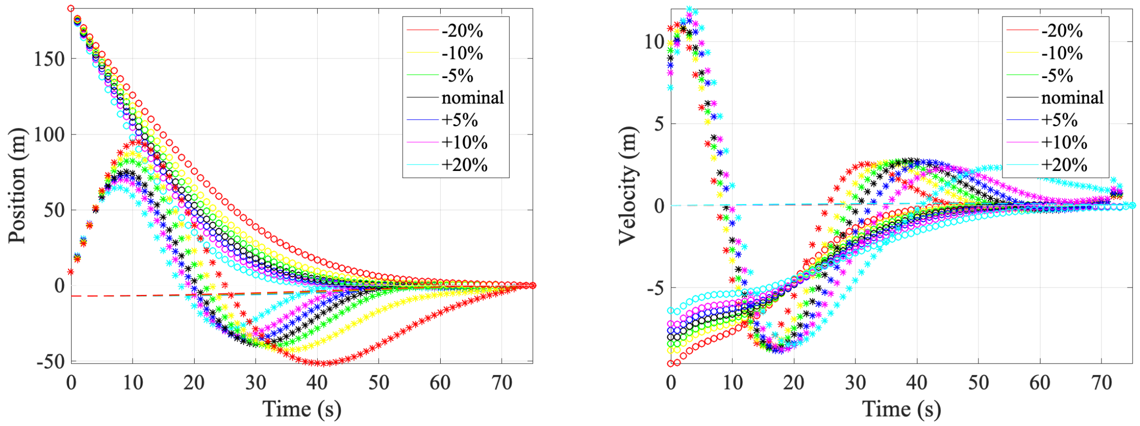
Figure 7.19: Parafoil descent trajectory and total control history due to initial position sensitivity of ± 20%.



(a) Attitude rate profile

(b) Quaternion attitude profile

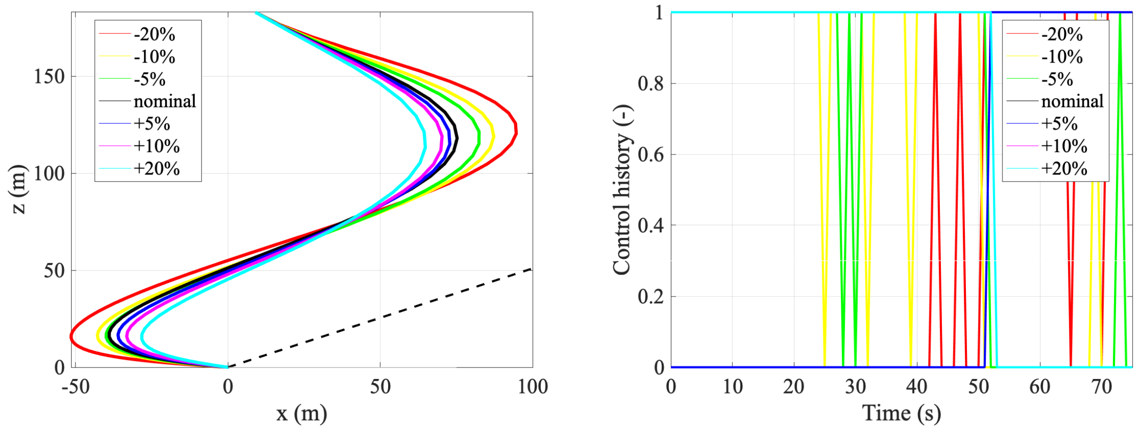
Figure 7.20: Parafoil attitude rate and quaternion attitude time histories due to initial position sensitivity of ± 20%.



(a) Position profile

(b) Velocity profile

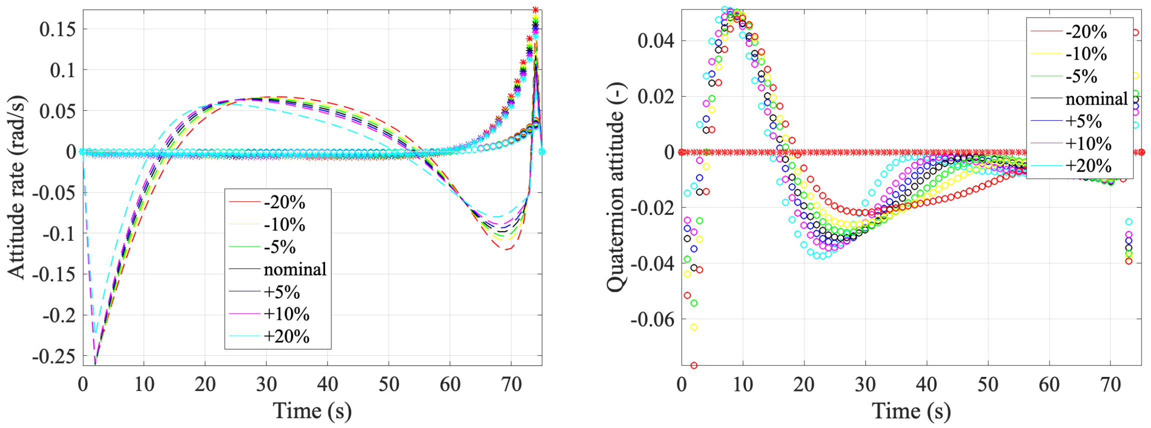
Figure 7.21: Parafoil descent position and velocity time histories due to initial velocity sensitivity of ± 20%.



(a) Trajectory

(b) Control profile

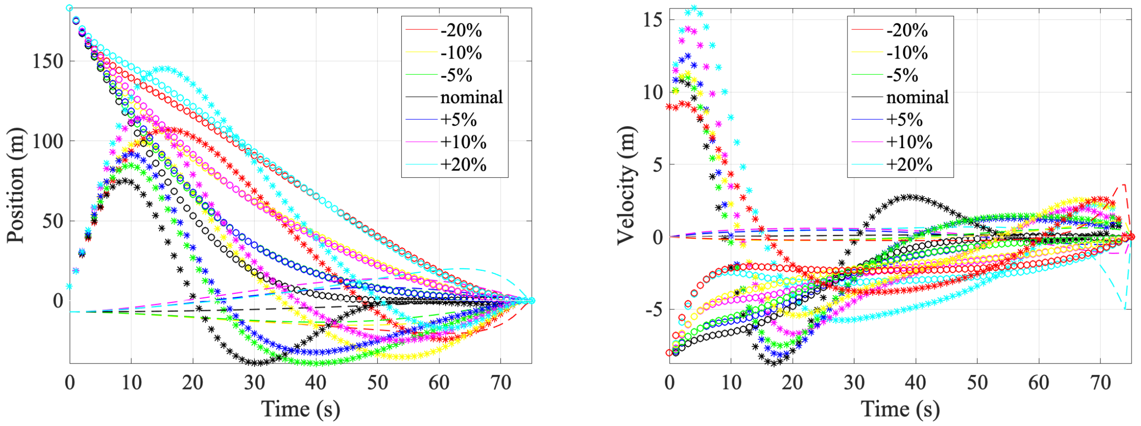
Figure 7.22: Parafoil descent trajectory and total control history due to initial velocity sensitivity of ± 20%.



(a) Attitude rate profile

(b) Quaternion attitude profile

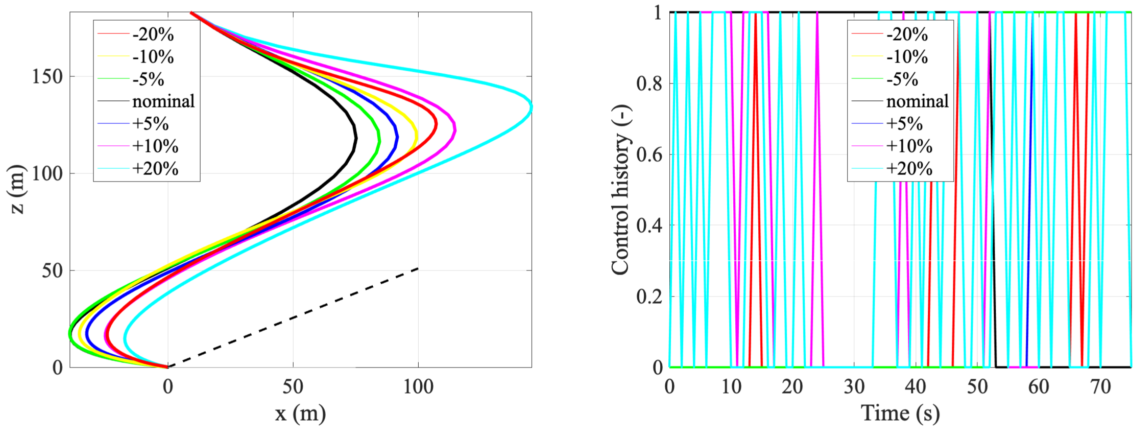
Figure 7.23: Parafoil attitude rate and quaternion attitude time histories due to initial velocity sensitivity of ± 20%.



(a) Position profile

(b) Velocity profile

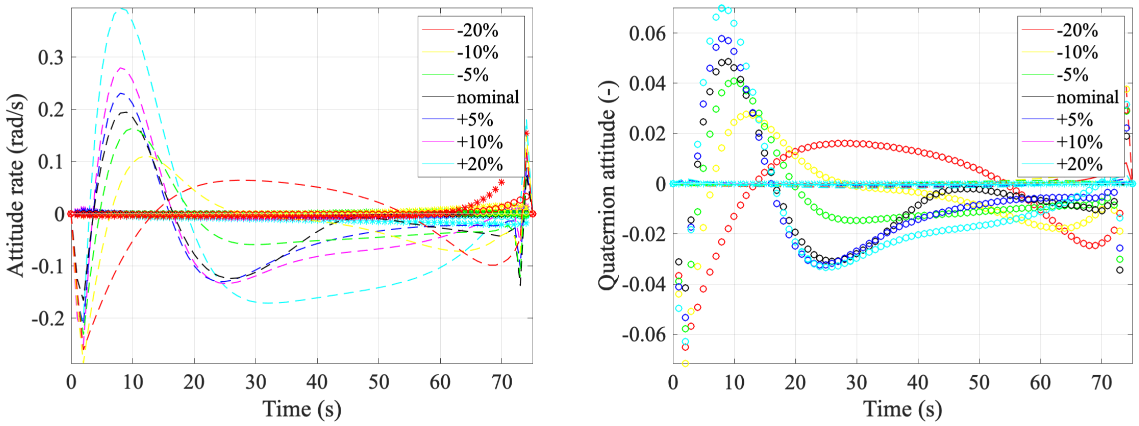
Figure 7.24: Parafoil descent position and velocity time histories due to lift coefficients sensitivity of ± 20%.



(a) Trajectory

(b) Control profile

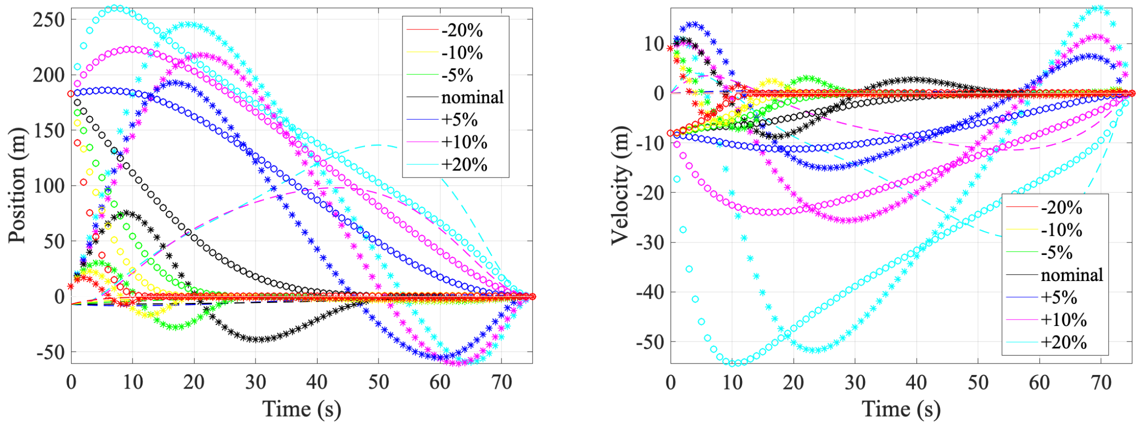
Figure 7.25: Parafoil descent trajectory and total control history due to lift coefficients sensitivity of ± 20%.



(a) Attitude rate profile

(b) Quaternion attitude profile

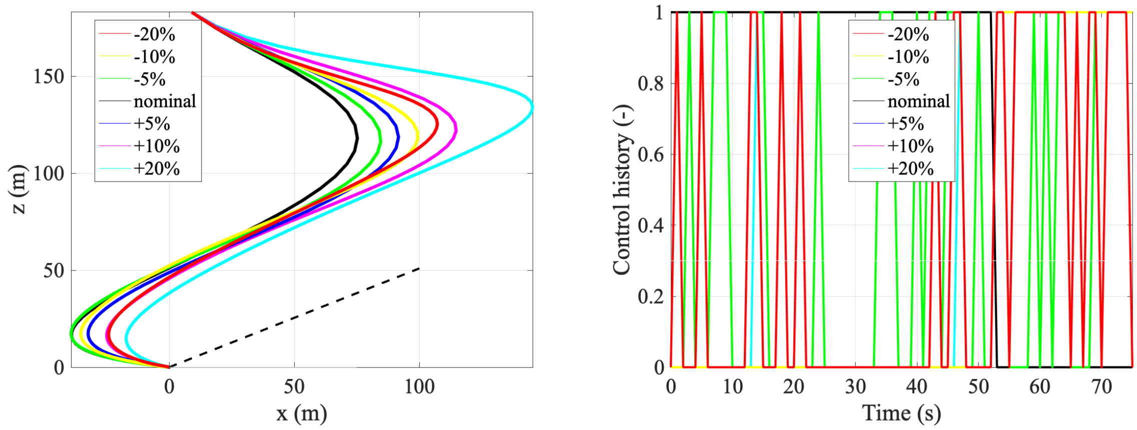
Figure 7.26: Parafoil attitude rate and quaternion attitude time histories to lift coefficients sensitivity of ± 20%.



(a) Position profile

(b) Velocity profile

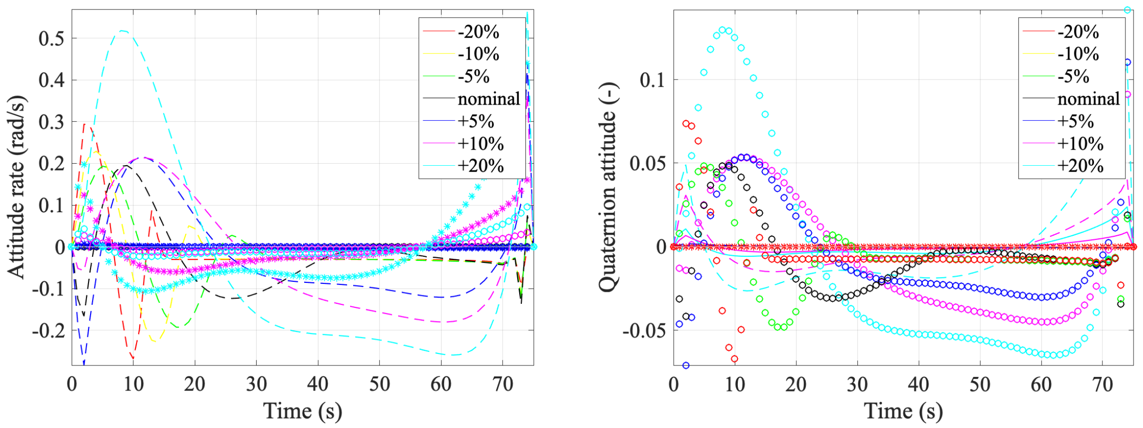
Figure 7.27: Parafoil descent position and velocity time histories due to drag coefficients sensitivity of ± 20%.



(a) Trajectory

(b) Control profile

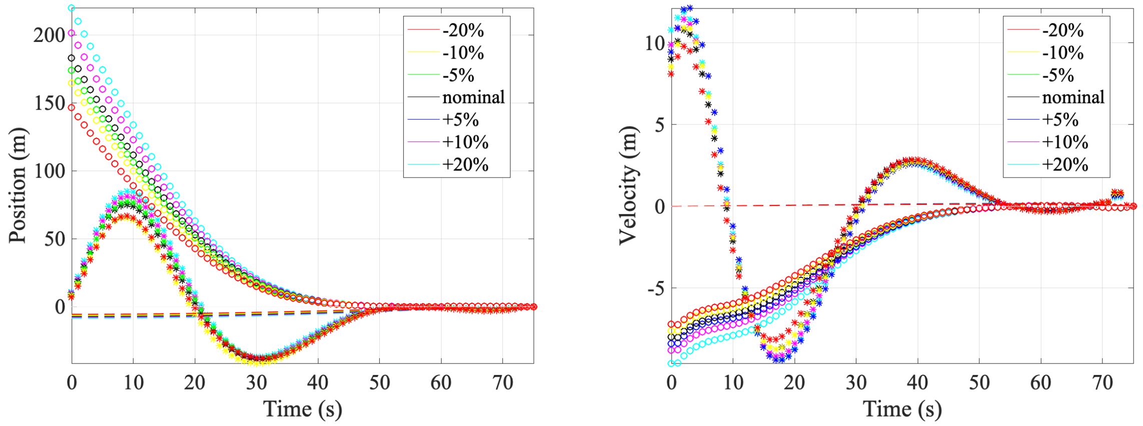
Figure 7.28: Parafoil descent trajectory and total control history due to drag coefficients sensitivity of ± 20%.



(a) Attitude rate profile

(b) Quaternion attitude profile

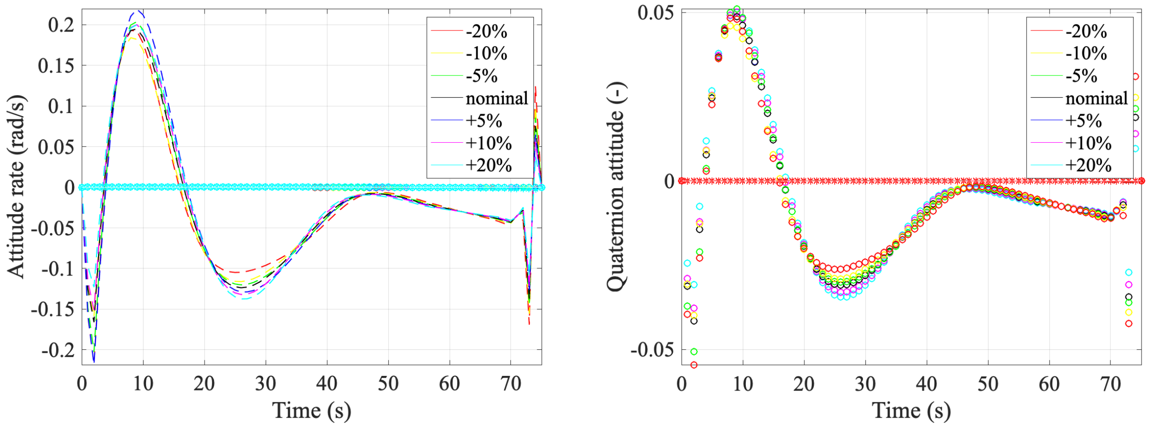
Figure 7.29: Parafoil attitude rate and quaternion attitude time histories to drag coefficients sensitivity of ± 20%.



(a) Position profile

(b) Velocity profile

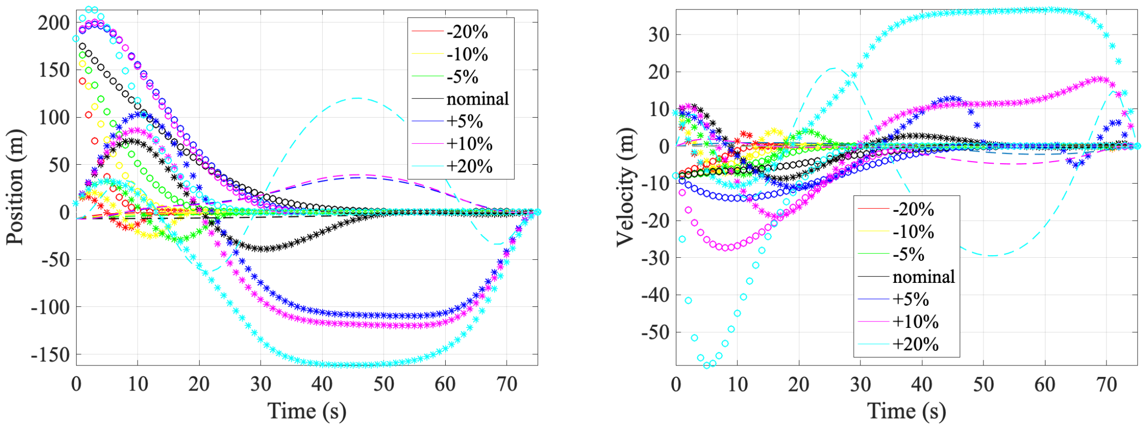
Figure 7.30: Parafoil descent position and velocity time histories due to coupled initial position and velocity sensitivity of ± 20%.



(a) Attitude rate profile

(b) Quaternion attitude profile

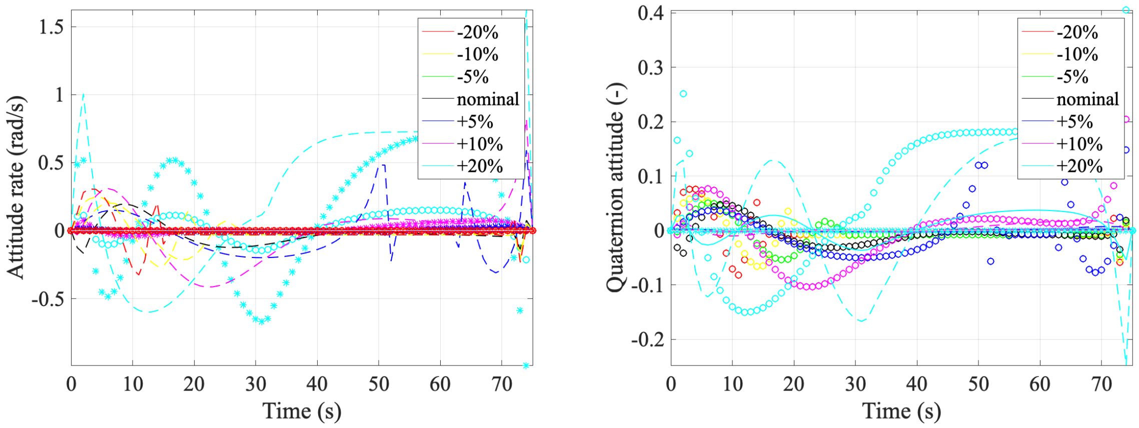
Figure 7.31: Parafoil descent trajectory and total control history due to coupled initial position sensitivity of ± 20%.



(a) Position profile

(b) Velocity profile

Figure 7.32: Parafoil descent position and velocity time histories due to coupled lift and drag coefficients sensitivity of ± 20%.



(a) Attitude rate profile

(b) Quaternion attitude profile

Figure 7.33: Parafoil descent trajectory and total control history due to coupled lift and drag coefficients sensitivity of ± 20%.

7.4. Obstacle Avoidance Guidance

In this portion of this chapter the algorithm is extended for obstacle avoidance compatibility based on multi-glideslope constraints. In this research two mission scenarios are considered in which the activation of this guidance system would be necessary. Specifically, when approaching the destination target, roughness of terrain might raise the need for avoiding path obstacles detrimental to the mission. As stated in Chapter 5 hills for which a subsurface flight constraint or single glideslope constraint does not suffice, or would not yield the most optimal solution. This scenario is further elaborated on in Section 7.4.1. The second mission scenario occurs when the spacecraft has to land at the rim of a mountain or in a valley/crater. This scenario is further discussed in the last part of this section.

7.4.1. Hill Avoidance

This section focuses on outline of the multiple glideslope constraint guidance simulation. The mission segment is divided into several parts and each portion is provided its own glide-slope constraint. Subsequently each segment is optimized individually while ensuring mission safety. This is especially necessary when landing in areas with high variation in elevation. In Fig. 7.34 the difference between a single and multiple glideslope constraints are depicted in a single figure. In the simulation a total of three phases have been adopted. The trajectory in the figure is made to touch the extreme points exactly for visualization purposes, which in real-time would not be the case. Contrary to the expectation, the multi glide-slope constraint actually relaxes the burden on the guidance algorithm. As can be observed from the figure, a single constraint pushes

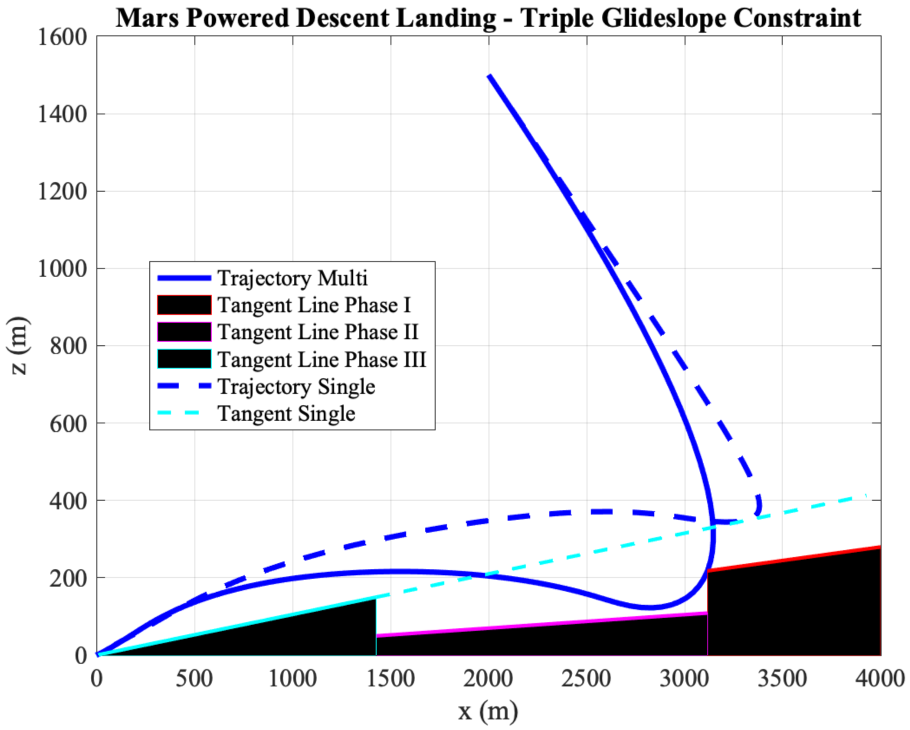


Figure 7.34: Simulation of the hill avoidance guidance algorithm applied during planetary descent. Notice that the trajectory in the figure is made to touch the extreme points exactly for visualization purposes, which in real-time would not be the case.

the algorithm to maintain unnecessary heights throughout a large portion of the simulation. Meanwhile, the segmentation causes the optimal path to follow a more relaxed path. It is therefore highly recommended for actual planetary landing algorithms to include a hill avoidance algorithm that in real-time calculates the required slopes and altitudes (for example based on TRN). This would relax the strain on the guidance system and ensure mission safety while still adhering to the ultimate mission goal: planetary precision landing. Another interesting observation is the fact that despite the relaxation of the slope in the mid-phase and final phase of the landing sequence, the algorithm still decides to travel the spacecraft up, glide and then push for an after burn. The latter has been concluded after careful analysis of the control history associated to this trajectory, which very closely resembles the control history plot of Fig. 7.4. Conclusively it can be stated that the hill avoidance guidance algorithm is a valuable addition to the system.

7.4.2. PLANETARY LANDING AT MOUNTAIN RIMS AND VALLEYS

A second possible scenario might arise when a spacecraft needs to land at the rim of a large mountain or in between a valley. A single and regular glideslope constraint would be too steep in such case, constraining the trajectory excessively or yielding the optimal path infeasible. For such cases the path is yet again augmented into differential steps. Distinct from the multiple multi-glide constraint, the obstacle avoidance guidance is enforced in a slightly different manner. Rather than originating from the target landing site, this glide-slope may enforce anywhere. In Fig. 7.35 the simulation for landing near the rim of a mountain is presented. From the plot it becomes directly evident what significant role the obstacle avoidance algorithm plays during planetary precision landing. As can be seen in the figure, a single glide-slope constraint would have caused the trajectory to crash into the mountain, which of course would be detrimental. Differentiating the constraint in two phases, one with a mountain line and one with a glide-slope, allows for optimal path planning while ensuring mission safety. Anew it is observed that despite the relaxed constraint in the terminal phase, the trajectory of the convex optimization method still enforces the spacecraft to fly upwards, glide and then land. This is attributed to the dynamics of the planetary precision landing sequence. The latter is proven by removing the imposed constraint, and simulating a sample. Without the imposed constraint the algorithm merely on the dynamics shows similar behavior, which leads to the conferred conclusion.

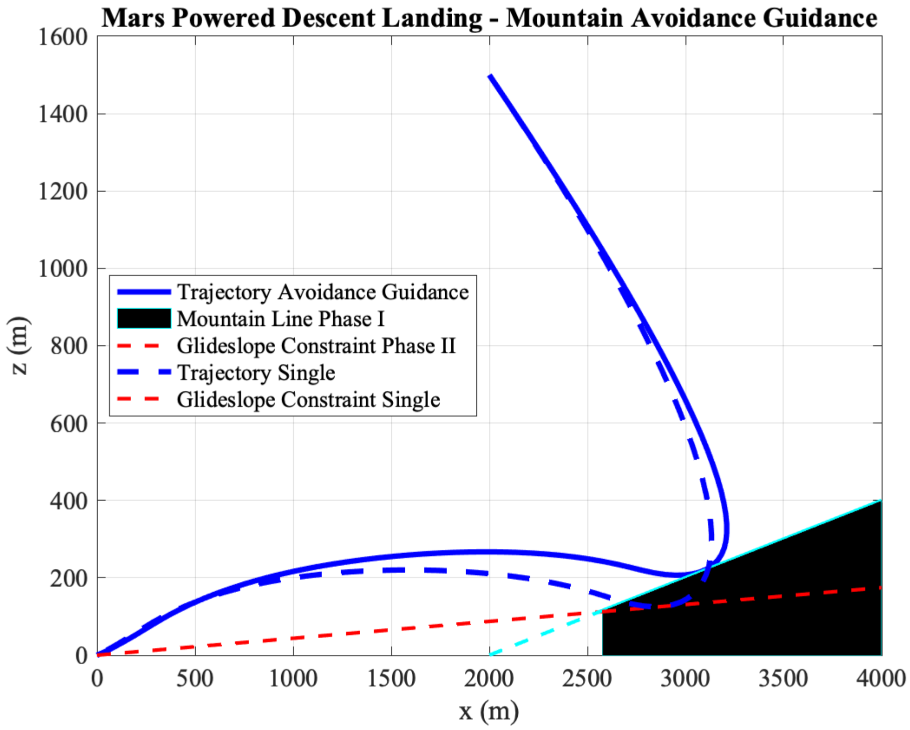


Figure 7.35: Simulation of guidance algorithm applied during planetary descent to land at mountain rims or in valleys. Notice that the trajectory in the figure is made to touch the extreme points exactly for visualization purposes, which in real-time would not be the case.

7.5. Sensitivity Analysis Powered Descent

This chapter concludes with a discussion on the sensitivity analyses that has been coupled to the design and simulation approach of the research. This sensitivity is necessary because some of the parameters have either been approximated or guessed based on engineering sense and are thus not known to the extend of desired accuracy. Furthermore, an incorporation of a sensitivity analysis provides great insight into how several conditions may effect the flight trajectory or quite the opposite, not effect the flight path at all. Through this process the robustness of the results is typically evaluated. This insight is of vital importance for the understanding of precision landing trajectories and how design parameters influence the accuracy. Based on the simulation results it is even possible to point out possible risk factors that influence the landing precision or the flight trajectory in general. The parameters that are included in the sensitivity study have been captivated in Tab. 7.3. An additional sensitivity study is added in which each time it is assumed that a single of the navigation sensor seemingly shows failure and is therefore unable to provide the spacecraft with state (i.e., position, velocity and angular rate) estimations.

Table 7.3: Parameters of sensitivity and their respective percentages.

Parameter	Description	Sensitivity percentage
r_0	initial position	± 20%
v_0	initial velocity	± 15%
m	mass (fuel and dry)	± 10%
$T_{\max}$	maximum thrust magnitude	± 10%
γ_{gs}	glide-slope angle	+ 100%

Now that the parameters of sensitivity are established, the sensitivity analysis procedure is outlined. For each parameter, two sensitivity examinations are established: one of parameter increase and one of parameter decrease, both based on the respective sensitivity percentages. A step-size 5% is chosen for sensitivity percentages equal to 15%, and 20%. For the sensitivity percentage of 10%, cases of 1%, 5% and 10% are analyzed. Notice that for the glide-slope constraint only positive values are considered, up to a value of 100% above the nominal angle of $\gamma_{gs} = 4°$. While one parameter is adjusted it is made sure that all other variables are maintained constant to be able to draw rationale conclusions. It goes without saying that only the variables that influence the trajectory are considered. The results are now conferred for each scenario, presenting time histories of the position, velocity, throttle setting and the trajectory. These entities have been chosen as they provide the most valuable simulation information. In Fig. 7.36 - 7.44 all the simulation results are presented. The final paragraph of this section presents some conclusive remarks regarding the sensitivity.

A variation of five governing variables has been conducted to assess the proper working of the algorithm. The results have been presented in the preceding figures. For an analysis of coupled variables, please be referred to Section 7.3. To maintain a clear discussion, this section outlines a short discussion and some conclusions per varied variable:

- **Initial position sensitivity:** presented in Figs. 7.36 and 7.37. From the position profile the conclusion is drawn that despite the increase in distance from the target, both in easting and upwards direction, the algorithm maintains its ability to guide the spacecraft to terminal conditions. Coupled to this the algorithm also manages to terminate at zero velocity for all cases. Nonetheless, as becomes evident from the throttle profile, for an increase in initial position of 20%, the algorithm no longer follows a distinct max-min-max profile. Rather, it seems as if noise has been introduced despite the fact that this is not the case. The conclusion is drawn that the distance is actually too large for the algorithm given the available fuel mass. A further increase in position would have yielded an infeasible trajectory.
- **Initial velocity sensitivity:** presented in Figs. 7.36 and 7.37. Interestingly, altering the velocities has more effect on the position time history than the altering the positions themselves. This is attributed to the coupled dynamics, in which position is a propagated variable that is dependent on time and velocity. The remark regarding the throttle profile discussed in the preceding point becomes even more evident for the initial velocity sensitivity case. The non-linear dynamics immediately cause the throttle to vary highly at increasing velocities. It is denoted that this is not a desired phenomenon, as altering the throttle setting to often might lead to engine failure.
- **Mass sensitivity:** presented in Figs. 7.40 and 7.41. Similar conclusions as delineated above are drawn with regard to the variation in mass. As expected the mass variation has tremendous effect on the velocity, as velocity is directly coupled to the depletion dynamics. As the fuel and dry mass are not related linearly, a simple one-on-one scaling does not working for the precision landing problem.
- **Maximum thrust sensitivity:** presented in Figs. 7.42 and 7.43. Indirectly this encompasses all variables associated with the thrust magnitude and orientation. Lowering the thrust magnitude likewise more burdensome for the algorithm, as the same mass needs to be carried with less force.
- **Glide-slope sensitivity:** presented in Fig. 7.44 and 7.45. The trajectory plot shows that each simulation adheres to the enforced cone constraint. The algorithm is very well capable of handling this constraint. However, at at high-valued angles the play-field of the spacecraft may get restricted too much. This is evident from the 100% increase, where the control history alters highly. Nonetheless, the powered descent guidance algorithm maintains its capability to establish the soft landing at the designated target.

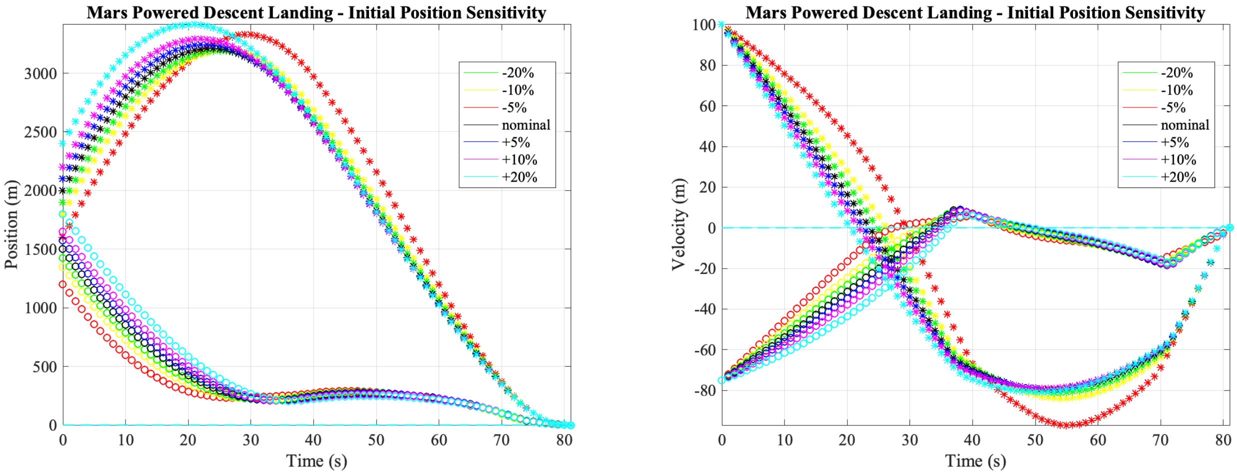


Figure 7.36: Powered descent position and velocity time histories due to initial position sensitivity of ± 20%.

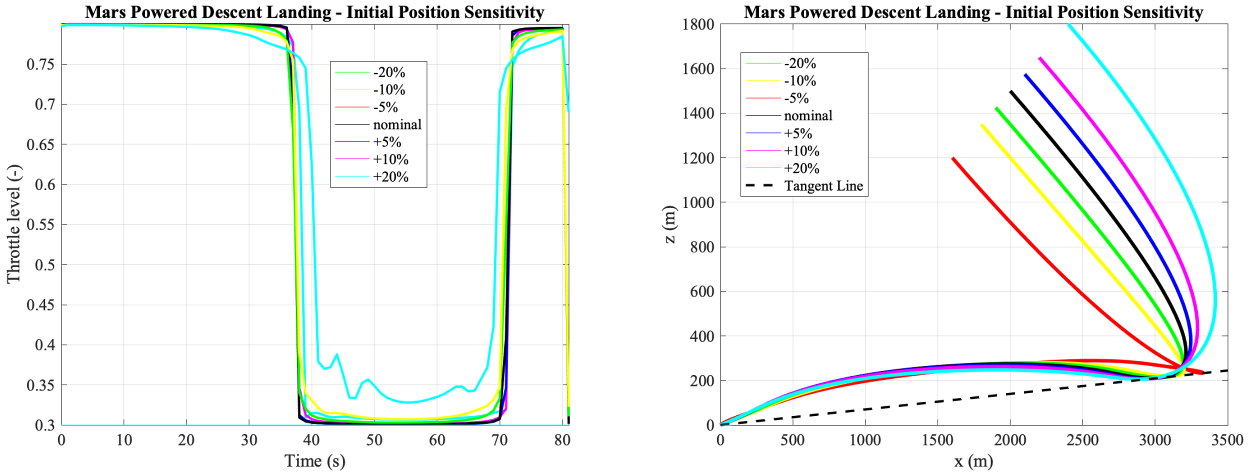


Figure 7.37: Powered descent control time history and emerging trajectories, both due to initial position sensitivity of ± 20%.

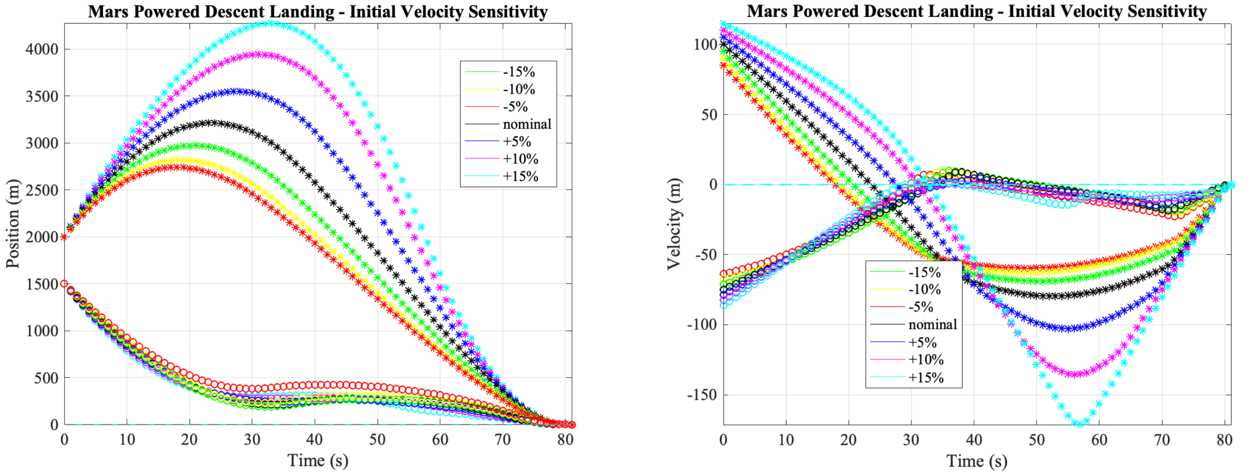


Figure 7.38: Powered descent position and velocity time histories due to initial velocity sensitivity of ± 15%.

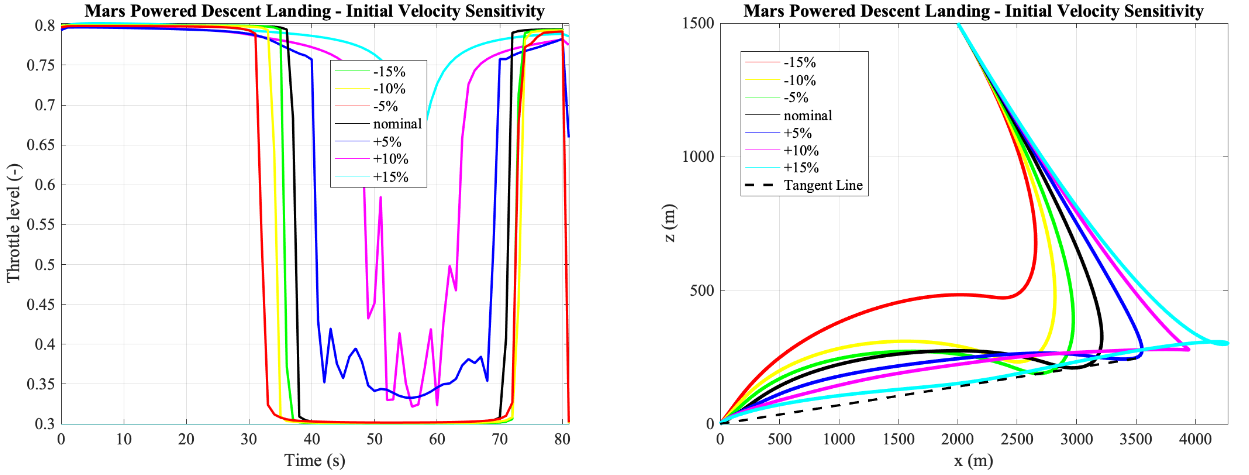


Figure 7.39: Powered descent control time history and emerging trajectories, both due to initial velocity sensitivity of ± 15%.

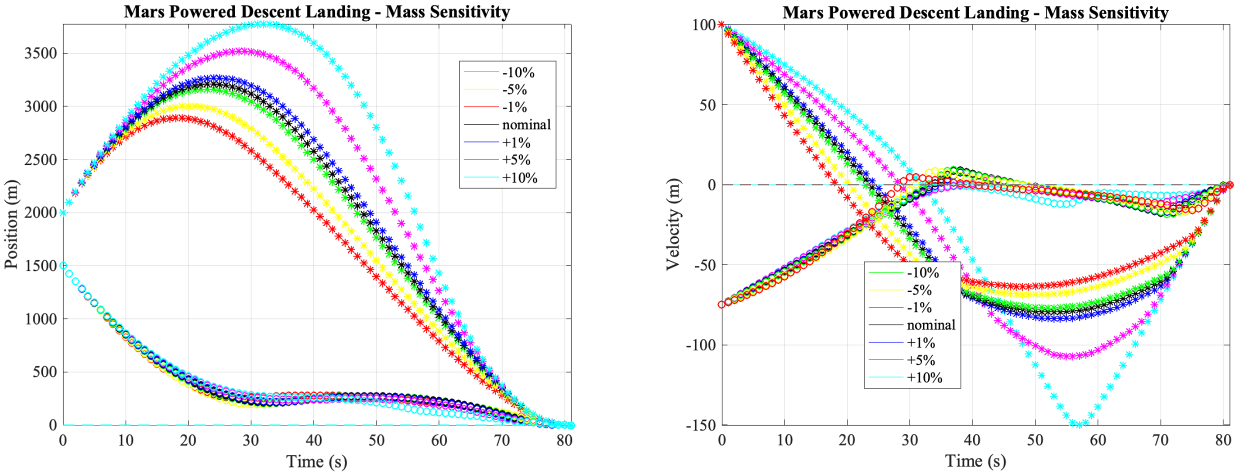


Figure 7.40: Powered descent position and velocity time histories due to initial mass sensitivity of ± 10%.

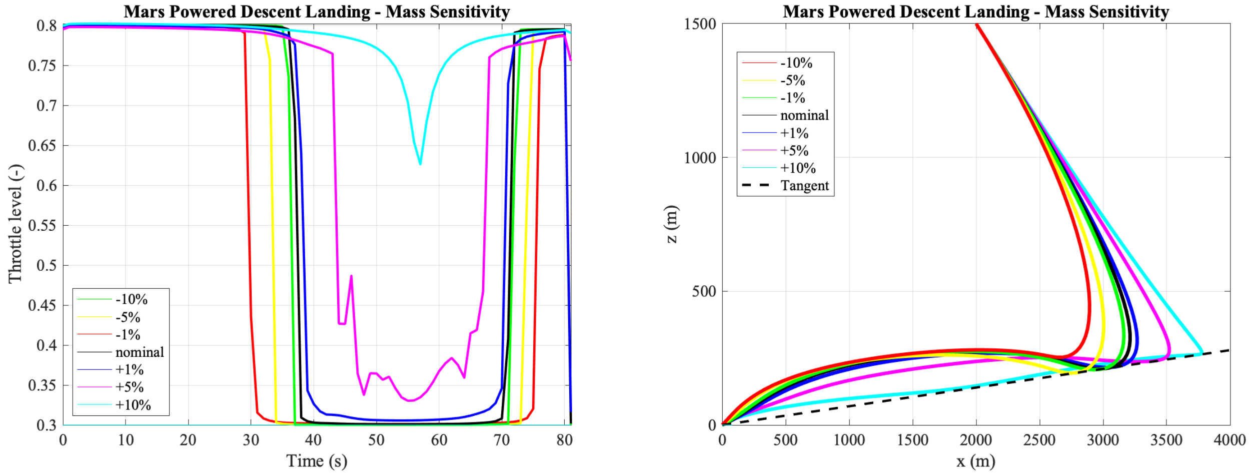


Figure 7.41: Powered descent control time history and emerging trajectories, both due to initial mass sensitivity of ± 10%.

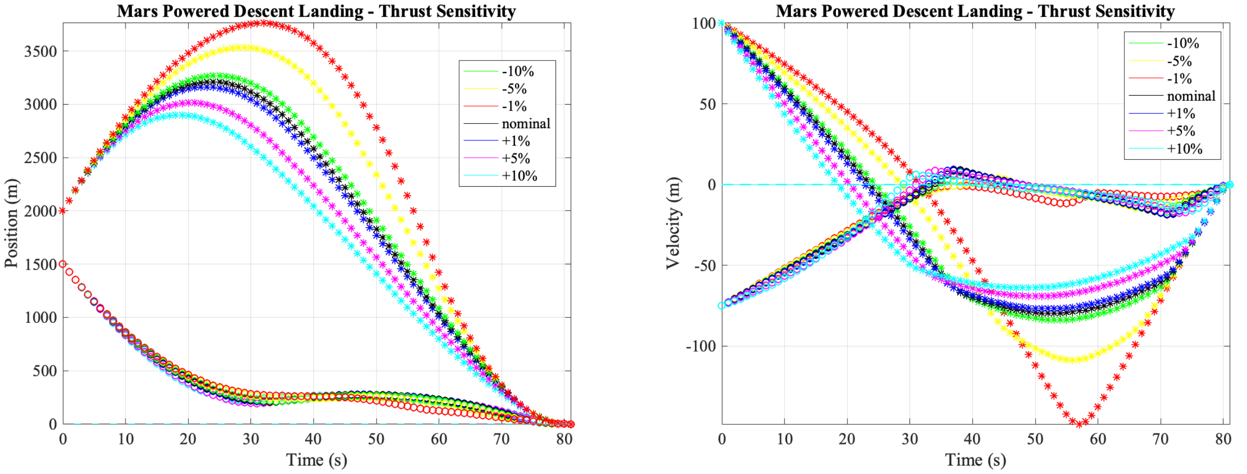


Figure 7.42: Powered descent position and velocity time histories due to maximum thrust sensitivity of ± 10%.

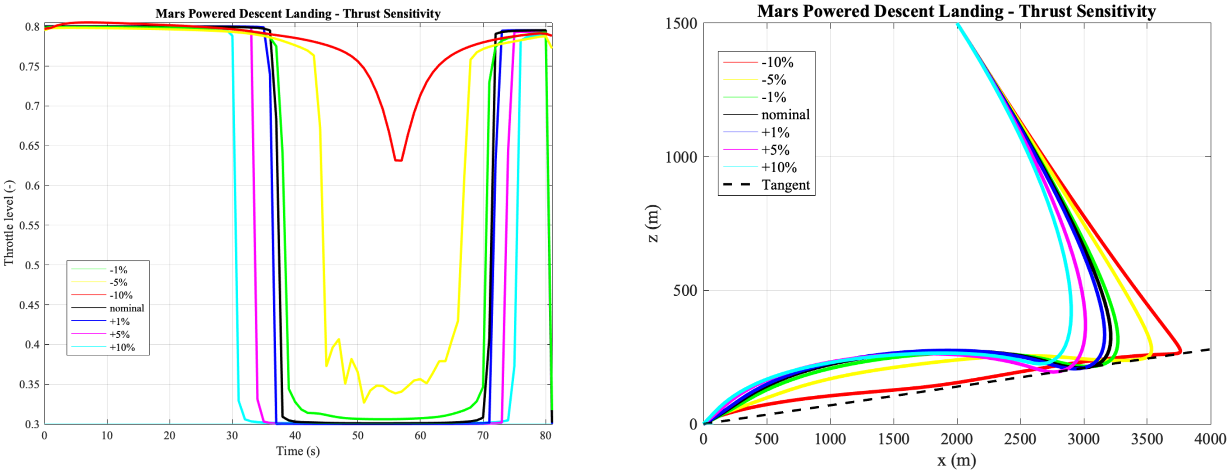


Figure 7.43: Powered descent control time history and emerging trajectories, both due to maximum thrust sensitivity of ± 10%.

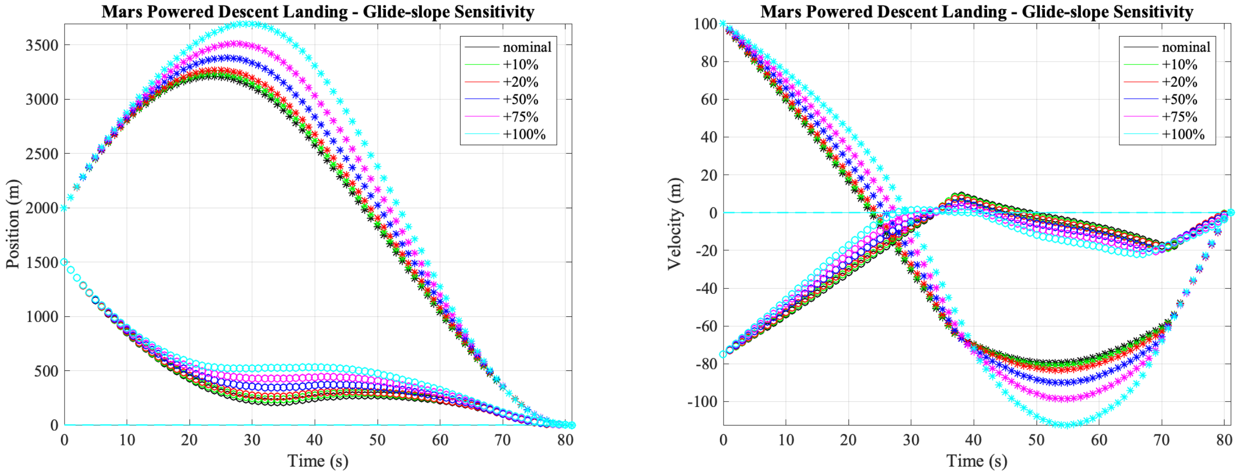


Figure 7.44: Powered descent position and velocity time histories due to glide-slope sensitivity of + 100%.

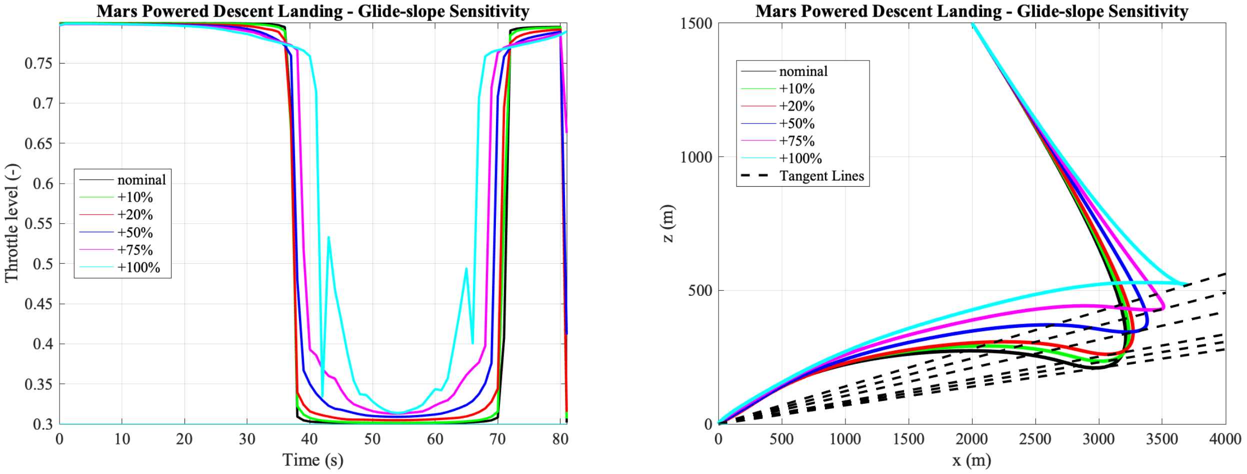


Figure 7.45: Powered descent control time history and emerging trajectories, both due to glide-slope sensitivity of + 100%.

8

Conclusions & Recommendations

Precision landing is an anticipated technology for future interplanetary missions. Achieving this requires advanced autonomous GNC technologies to allow spacecraft to land autonomously and optimally, as on-board resources are limited. This research, which has been conducted at JPL's Department of Robotics (Modeling & Simulation of Planetary Vehicles) and Delft University of Technology, has therefore focused on autonomous guidance and control for planetary precision landing. Based on the defined problem statement and the established focus aspects of this research, the entrenched research question was stated to be as followed:

> What order of magnitude precision for landing on planetary bodies can be achieved by using advanced guidance, navigation & control systems in combination with state-of-the-art autonomous trajectory optimization algorithms?

Specifically, this research focused on the planetary bodies Mars and Titan, for which the lossless and successive convex optimization methods have been adapted. From the carried out analyses the conclusion is drawn that absolute precision landing through this robust method is very much possible. It is emphasized however that despite robust guidance and control methodologies, proliferation of other anticipated technologies are needed to eventually establish planetary precision landing. Powered descent guidance for Mars landing sequences is an extensively research topic and the research has been based on the comprehensive availability of literature. The algorithm has been extended to become compatible with two more mission scenarios. As part of the NASA Space Exploration Technology Directorate, a parafoil has been proposed for landing on Titan due to its cost effectiveness, ease of deployment, relatively low mass compared to the prospective payload and capabilities of precise autonomous delivery. The first extension of the algorithm therefore focused on making the convex optimization method compatible for landing a parafoil on Titan. At last, the algorithm was extended for obstacle avoidance compatibility based on multi-glideslope constraints. Specifically, when approaching the destination target, roughness of terrain might raise the need for avoiding path obstacles detrimental to the mission. A comprehensive number of conclusions and recommendations have been derived, which are delineated after providing brief answers to each research sub-questions.

- **RQ-1:** Is a convex optimized real-time generated guidance algorithm capable of precisely guiding the spacecraft to predetermined landing ground?
  Despite the random nature associated with entry, descent and landing, the convex optimization algorithm has proven to robustly tackle dispersions and guide the spacecraft to the designated landing site. The extension to compatibility for landing on Titan, the added obstacle avoidance design and the conducted sensitivity analysis further entrench this statement.

- **RQ-2:** What cost function, state constraints and control constraints are needed based on the mission scenarios?
  For all mission scenarios imposing the glide-slope constraint is needed. Other state constraints are lower and upper velocity bounds for avoidance of stall and structural damage, respectively. Furthermore, the control constraint naturally follows from the spacecraft properties, thus either thrust bounds or pull control bounds. The cost function is entrenched based on the desired final outcome: minimizing fuel consumption, limiting position error of minimizing control effort. All these cases have been researched in this thesis, and for all cases the convex optimization method adapted well.

- **RQ-3:** Is a convex optimized real-time generated guidance algorithm capable of retaining to the state and control constraints throughout the entire sequence?
For all cases considered in this research, the state and control constraints were adhered to. No cases were encountered in which the constraints were violated. At times, constraints were relaxed deliberately to assess the working principle of the guidance system. The convex optimization framework is a very powerful tool for incorporating state and control states into the guidance scheme.

- **RQ-4:** How does the incorporation of a three-degree-of- freedom fuel/power optimal trajectory differ from a six-degree-of-freedom trajectory?
From the verification and results chapter the conclusion is drawn that the three-degree-of-freedom is more easily implemented. Nonetheless, it has become evident that implementation of a 6DoF guidance scheme is vital, especially for landing on Titan. In the parafoil sensitivity analysis there has been shown that alterations in variables might cause extreme fluctuations in attitude (i.e., attitude rate and quaternion attitude), a matter to be controlled for actual flight. For Titan this is more present due to the relatively more significant wind, as well as the inherent design of a parafoil spacecraft.

- **RQ-5:** Is the algorithm capable of generating optimal divergence paths in real-time in case the nominal trajectory turns out to be non-feasible?
An extensive minimum landing error analysis has been carried out, from which the conclusion is drawn that the algorithm is indeed capable of minimizing distance to target for extreme scenarios that give rise to such an guidance scheme. It is denoted that the designed algorithm serves as a proof of concept, as Matlab is not fast enough for flight implementation. For on-board guidance C++ would be one of the logical choices.

- **RQ-6:** Are the landing accuracy and the fuel/power consumption mutually related?
In the robustness assessment and sensitivity analyses carried out in this research, it was found that the fuel/power consumption is highly related to imposed errors. All Monte Carlo generated histograms showed a wide sensitivity for variation in variables. Additionally, it is denoted that from a theoretical perspective the accuracy and consumption are of course mutually related. This is explained as follows: in case infinite fuel would be available, a fuel theoretical guarantee could be provided of reach the target within infinite time. This would turn the problem into an infinite horizon optimal control problem.

- **RQ-7:** How does the algorithm performance differ between a thin atmosphere (Mars) and thick atmosphere (Titan) landing sequence?
In terms of establishing soft landing, the algorithms do not differ (i.e., in their eventual goal). However, the spacecraft for landing on either planetary body differs significantly from the other. This means that no definitive conclusion can be drawn with respect to the performance of the convex optimization method within thin and thick atmospheres. On the other hand, it is now known that the method is suitable for landing on Earth (SpaceX booster recovery), Mars and Titan. This give rise to the probable conclusion that the adapted method theoretically works robustly in a wide range of type of atmospheres.

- **RQ-8:** Is a landing accuracy on the order of meters theoretically feasible?
Based on the carried analyses the conclusion is drawn that a landing accuracy is certainly feasible. It is however denoted that advanced guidance schemes play a significant role here-in, but are not sufficient. Besides tackling the limitations of the guidance system itself, the coupled systems require equal advancement. The recommendation section further elaborates on this matter.

- **RQ-9:** What affect do uncertainties in the state variables have on the algorithm, i.e., how robust is the algorithm?
Two sensitivity analyses have been carried out. One for parafoil terminal descent associated to Titan, and a similar one for powered descent to landing on Mars. For both cases the conclusion was drawn that the algorithm is very robust in tackling uncertainties. Furthermore, three sources of error were introduced into the closed GNC loop: thrust magnitude errors, thrust misalignments and navigation errors. The algorithm managed to tackle these errors while adhering to all constraints and solving the imposed cost function.

Conclusions

Conventionally, large dispersions are caused during the initial EDL phase, which are corrected for during terminal descent. This research has adapted the convex optimization method for simulating terminal descent sequences, with a distinct focus on guidance formulated as a finite horizon optimal control problem. The simulations have been verified by directly comparing the results to industry's current best practise (for Mars landing). All simulations have been carried using CVX in Matlab. The obtained results are found to be (near) identical and hence the designed algorithm has been verified. The algorithm has subsequently been extended to make it compatible for a Titan parafoil landing simulations and obstacle avoidance. To maintain a clear discussion on the drawn conclusions, the are delineated in an enumerate fashion:

1. From the time history of the position and velocity states it is deduced that the implemented algorithm produces identical results. Overall the algorithm plots resemble the reference plots very closely. It becomes evident from the position plot that when not enforcing the glide-slope constraint, the trajectory may reach near zero-altitude early on in the sequence. This phenomenon causes a large portion of the flight segment to be subterranean, which is not desired during powered descent.

2. The throttle history marks a max-min-max profile, corresponding with profiles often encountered in optimal control literature. With regard to the acceleration and net force time history there is one significant difference in the final value of the gravity direction acceleration. This difference might be attributed to the fact that the respective literature seems to enforce a final burn in that direction. Another difference cause could be the applied discretization scheme. The same type of discrepancy is observed for the final throttle setting. This might also be the direct consequence of the small difference between the fuel consumption of the two algorithms for the plain case, 387.9 kg vs 389.4 kg. Additionally, it is observed that the location of the temporal nodes differs from the Açıkmeşe and Ploen (2007) at various locations. This further emphasizes the idea that the discretization scheme is the main cause of the differences. Nonetheless, both plots are overall rather similar. This also becomes evident from the thrust pointing figures. It is evident from the overall plots that the control and state constraints are satisfied throughout the entire sequence.

3. The optimal flight time found through the golden search method corresponds to a rounded-off value of 73 seconds, one second longer than the value presented in the Açıkmeşe and Ploen (2007) for the same scenario. The reasons for this slight difference are attributed to the discrepancies discussed in the previous item. The throttle profile is marginally different, which naturally causes a different mass consumption profile and hence a different optimal time of flight. Discretization of the algorithm plays a significant role here-in. It is denoted that the golden search method does not theoretically guarantee the obtaining of time of flight in real time, hence the adaption of the more robust successive method.

4. For the minimum error landing algorithm, corresponding to the results in Blackmore et al. (2010), the algorithm was terminated at $t_f = 79s$ with a fuel consumption of 399.4 kg. This establishes the value of the convex optimization minimum landing error method, as an optimal solution was found even when the spacecraft's physical boundaries where nearing its edge in terms of distance from target and initial velocity. One can observe from the figures that the altitude trajectory abstains from tending to zero until sequence termination in the soft-landing process, which verifies the proper working on the glide-slope constraint. Likewise, the throttle profile is of a max-min-max nature, after the lemma of optimal control problems. Furthermore, the soft landing on the spacecraft is entrenched, as both position and velocity elements terminate at zero. An example is provided for which the paper claims there is no absolute soft landing. What is remarkable however is that designed algorithm of this research still managed to guide the spacecraft to precision landing.

5. It becomes evident from the velocity norm plot that the algorithm manages to maintain below this normalized velocity throughout the sequence. From the surface trajectories it becomes evident that the 45° case overshoots the y-axis to maintain satisfaction of the pointing constraint, while the 90° and the unconstrained cases take a more direct path. This is in exact accordance with the results obtained by Açıkmeşe et al. (2013).

6. Based on Szmuk and Açıkmeşe (2018) the six-degree-of-freedom Mars powered descent simulation are entrenched. From the two plots (including the 3D surface plot cone) it becomes evident how well the trajectory adheres to the glide-slope the glide-slope constraint in three-dimensional Euclidean space.

7. For the Titan landing cases, both the 3DoF and 6DoF simulations show promising results for on-board implementation. The soft-landing is guaranteed while adhering to imposed mission constraints. The control time history remains within desired limits, as well as the aerodynamic angles. The terminal landing is enforced such that the payload lands in parallel direction to the gravity vector. The glide-slope constraint has been properly enforced to ensure the spacecraft does not reach the surface fast, which would cause an undesired sub-surface flight. Likewise, the pointing constraint has been enforced successfully to cope with the requirements of Terrain Relative Navigation.

8. From the figures of the obstacle avoidance simulations it becomes directly evident what significant role the algorithm plays during planetary precision landing. A single glide-slope constraint would have caused the trajectory to crash into the mountain, which of course would be detrimental. Differentiating the constraint in two phases, one with a mountain line and one with a glide-slope, allows for optimal path planning while ensuring mission safety. Anew it is observed that despite the relaxed constraint in the terminal phase, the trajectory of the convex optimization method still enforces the spacecraft to fly upwards, glide and then land. This is attributed to the dynamics of the planetary precision landing sequence.

Recommendations

The next steps in future research are the implementation of the 9DoF high-fidelity model within NASA/JPL sophisticated DARTS (Dynamics Algorithms for Real-Time Simulation) tool to test for real-time hardware design and visualize terminal descent within the software. Actual testing in C++ is also advised to guarantee fast convergence. Including thrusters and on-board fuel (to a certain allowable mass) is also advised for Titan terminal descent to increase control authority. This is especially true for motion in (force) and around (moment) Easting direction, as direct control of side force and pitch moment are not possible within the parafoil dynamics framework. To avoid stall and destruction of the parafoil through mechanical loads, minimum and maximum velocity magnitudes should be imposed in actual flight, respectively. Like with the Mars case, Titan simulations should be run to test the robustness of the algorithm in case a trajectory turns out to not be feasible. This is defined as a minimum landing-error problem, which basically minimizes position error in case absolute soft-landing (zero position error) is theoretically not possible. The latter can have many reasons, but the most significant is distance from target. These considerations, amongst others, will eventually lead to the realization of precision landing on Mars and Titan. For the system level recommendations, like for the conclusions, a comprehensive list is provided:

1. **Guidance system recommendations:**
   The guidance system, though being very robust, has been limited by the chosen models and the chosen dynamics. First of all, a sophisticated analysis should be carried out regarding the (atmospheric) properties of both Titan and Mars. For the parafoil, a more detailed dynamics analysis can be carried out by incorporating 9DoF multi-body dynamics. Aerodynamic effects should be considered for the Mars case. Optimal time-of-flight should be found more robustly, while theoretical guarantees for this matter should be investigated.

   The depletion of mass due thrust causes the moment of inertia to change over time. This effect has not directly been taken into account into this research, and thus requires further analysis.

   The discretization scheme heavily influences the control profile, amongst others. This became evident in the verification chapter. Various schemes should be designed and tested. On top of that, it has been denoted that in between the temporal nodes there is no theoretical guarantee of adhering to imposed constraints. The discretization scheme plays a significant role here-in and should carefully be designed.

   State propagation has been established using the Runge-Kunga 4 integrator. Different methods should be tested to assess the response of the GNC system.

   The research adapted a method of uploading a reference trajectory to the computer and running the sequence, which is then controlled through a PID controller. The method of on-board adaptive correction should be explored, as it might provide more robust results.

2. **Navigation system recommendations:**
   This research assumed perfect state knowledge for both the nominal and error-integrated simulations. Consequently, without the proper determination of state, the vehicle would loose orientation relative

to the nominal trajectory. For actual flight applications the navigation is thus a crucial entity. This includes the integration of sensor inputs through navigation filters.

Besides the inertial measurement unit and the Terrain Relative Sensors, a GPS like constellation should be designed around the planetary body at which the landing will take place. This could be in the form of micro-satellites to aid during EDL. Lack of robust navigation systems such as GPS cause large dispersions, a matter to be solved.

3. **Control system recommendations:**
   The GNC loop has been closed using a PID controller. Alternative methods, such as H-infinity control should be explored. A detailed sensitivity analysis should be coupled to this to conclude whether more sophisticated controllers are beneficial for precision landing matters. Furthermore, it is advised that for actual flight implementations, a more detailed control analysis is carried out. This should include bode plots, gain tuning and root locus plots, amongst others.

   A problem often encountered in EDL control is that advanced control system have difficulty following up with the guidance sample rate, especially for attitude control. While this is a limit imposed by robustness of the flight computer, it should be considered in the design phase. On-board applications are limited of course by the maximum descent mass. Nonetheless, control systems of proper robustness exist for incorporation into the planetary descent problem and should therefore be explored.

   The matter of actuator saturation has not been considered in this research. For both the Mars and Titan landing cases this phenomenon could occur and should therefore be modelled. Also, singularities in for example the transfer functions might occur. Coupled to the incorporation of navigation, measurement noise might arise. These sources for errors should all be counted for in the design of flight systems.

Appendices

A

Design Parameters MSL & Mars 2020

NASA has chosen to build on the legacy of Curiosity as a design baseline for Mars 2020 (Fosse et al., 2015). For that reason scaling methodologies shall be applied to the MSL design, of whom the before-mentioned properties are available. Prior to launch, the mass properties of MSL were determined on a spin balance table (at the Kennedy Space Center in the Payload Hazardous Servicing Facility). Given the empirical nature of the results, a sensitivity analysis will be coupled to the procedure. The sensitivity percentages depend on the accuracy of the measurements, the reliability of the source and the type of scaling (i.e., linear or quadratic). Important to mention is that the managers of the Mars 2020 mission have stated in Gerstenmaier (2017) that the mass growth of the rover is one of the open risks of the project. Mass particularly affects inertia, location of the CoG and thrust needed for maneuvering, hence careful considerations are a prerequisite. Furthermore scaling also influences aerodynamic properties. These matters are outlined in this Appendix.

A.1. Mass and Dimensions Scaling

On 24^{th} October 2011, the MSL spacecraft went through a final mass measurement with a predefined accuracy requirement of 0.5%. After accounting for the various elements, the total spacecraft mass was found to be 3838.7 kg ± 16.7 kg, which meant an accuracy of 0.4% was achieved and thus the requirement was met. The rounded individual mass contributions of the major flight systems as determined by NASA are presented in Tab. A.1. Confer to the design concept presented in Wilson (2013), initially the Mars 2020 would have had a mass of 950 kg. However, due to scientific demands imposed on the design team, the current payload mass of the mission equals 1050 kg.

Table A.1: Rounded and percentile individual mass contributions of the major Mars Science Laboratory flight systems. Data source: NASA/JPL

Mission stage	Substages	Total mass (kg)	Mass percentile (%)
Cruise		**539**	**14.0**
EDL		**2401**	**62.5**
	Backshell	891	23.2
	Powered descent stage	730	19.0
	Hydrazine fuel	400	10.4
	Heat shield	380	9.9
Payload (Curiosity rover)		**899**	**23.4**
Total		**3839**	100

A.2. Principal Moments of Inertia Scaling

Distinctive to the above, the principal inertia scaling shall be of a quadratic essence. First, the Moments of Inertia (MoIs) about the principle axes (i.e., I_{xx}, I_{yy} and I_{zz}) for MSL shall be established. The inertia values of the entry capsule have been obtained from Schoenenberger et al. (2014) while all others values presented in Tab. A.2 are retrieved from Fields (2012), Petrick et al. (2014) and analytical models (irrelevant data left blank). On a confidence interval of 99.87%, the MoI of the dry spacecraft shall be known within 5%. Such

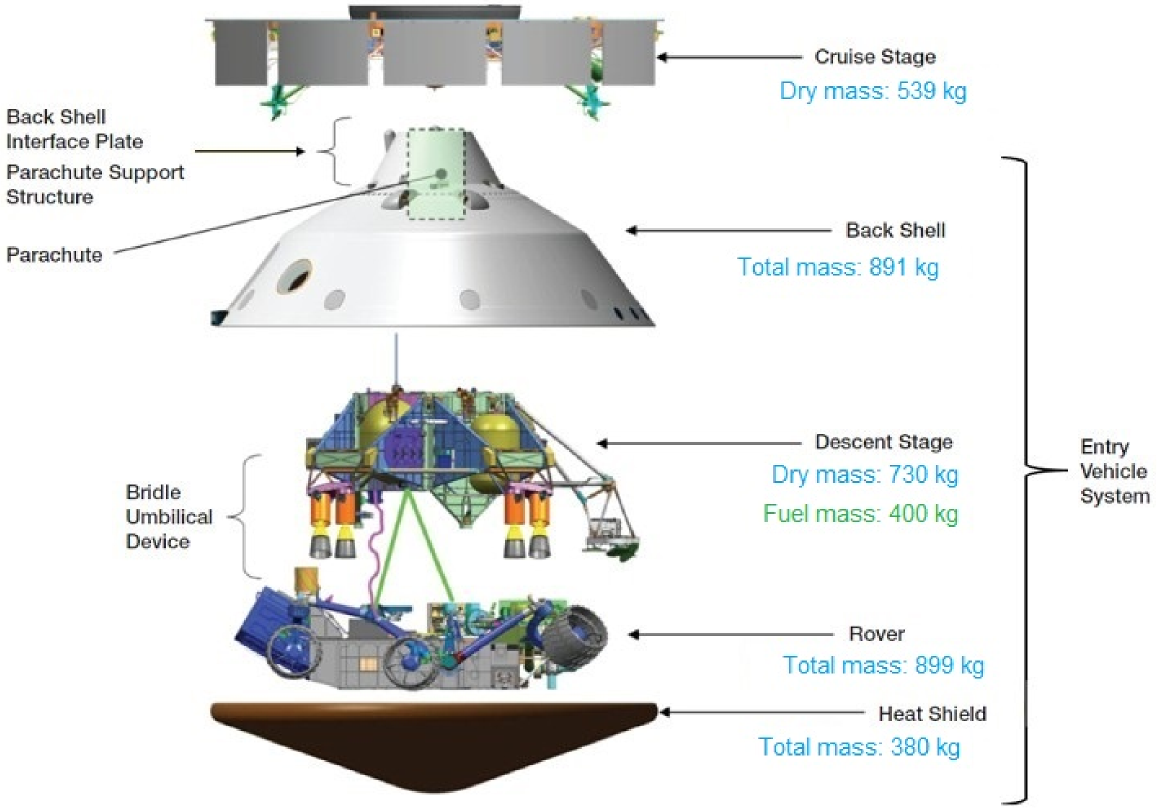


Figure A.1: Mars Science Laboratory major flight systems (Prakash et al., 2008) (Fields, 2012) (adjusted).

was the requirement established in Fields (2012), where there is also stated that the I$_{zz}$ values were directly measured. Notice that the I$_{xx}$ and I$_{yy}$ values of the powered descent stage have been established based on normalized ratios obtained from measurements of a 100 mm subscale module of the MSL capsule.

Table A.2: Mars Science Laboratory principal Moments of Inertia: predicted and measured I$_{zz}$ values from Fields (2012) and estimated entry capsule values from Schoenenberger et al. (2014).

System	**I$_{xx}$ (kg-m^2)**	**I$_{yy}$ (kg-m^2)**	**I$_{zz}$ (kg-m^2)**		**I$_{zz}$ Percentile difference (%)**
	Predicted	**Predicted**	**Predicted**	**Measured**	
Entry capsule	4800	3800	2900	-	-
Rover	-	-	768.9	772.2	0.4
Descent stage (dry)	1045.9	748.2	713.3	723.6	1.4
Backshell	-	-	1039.9	1039.6	0.0
Heatshield	-	-	1318.1	1365.0	3.4
Cruise stage	-	-	1115.1	1163.1	4.1
Maximum difference					**4.1%**

The coordinate systems are chosen as such that the products of inertia are approximately zero, hence leaving only the principal moments of inertia. This assumption can be made in case the spacecraft has two axes of mass symmetry. While in reality an axis is never perfectly symmetrical in terms of mass, this assumption maintains it validity due to negligible offsets. Bedford and Fowler (2008) provide the following relation:

$$\mathbf{I} = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{A.1}$$

By applying the parallel-axis theorem, an object's inertia matrix **I'** can be used to determine the inertia matrix of **I** based on differential mass elements and distances from the original CoG:

$$\begin{aligned} I_{xx} &= I_{x'x'} + (d_y^2 + d_z^2)m \\ I_{yy} &= I_{y'y'} + (d_x^2 + d_z^2)m \\ I_{zz} &= I_{z'z'} + (d_x^2 + d_y^2)m \end{aligned} \tag{A.2}$$

While distance will be assumed to scale linearly for the Mars 2020 class vehicle, inertia shall have a quadratic effect as delineated in the equations. Because the high mass density of fuel tanks have a large distance from the CoG, the effect will be greatest. In the above discussion for powered descent, only dry spacecraft mass has been considered. Inertia values for the fueled spacecraft to be computed in the subsequent research will be based on scaling of mass (previous section), scaling of distance and the parallel-axis theorem. By applying this method, inertia assessment at various instances of the powered descent phase will be possible. Finally it is denoted that, given the 5% MSL measurement accuracy requirement settled in Fields (2012), the quadratic nature of the scaling and the intermediate reliability of the source (i.e., partly analytical models), a sensitivity study of ± 10% shall be coupled to the principal moment of inertia scaling.

A.3. AERODYNAMICS ASSESSMENT

For modelling of the aerodynamic forces and moments, knowledge of the force and moment coefficients is required. Because the shape of the Mars 2020 vehicle reassembles MSL, the assumption is made that these non-dimensional coefficients are similar. However, to maintain the scientific tendency of the research, a sensitivity analysis will be coupled to this assumption. The sensitivity values are based on the great analysis conducted by Way et al. (2013) on the aerodynamics of the MSL. The aerodynamic force coefficients are primarily defined in terms of an axial and normal component through Eq. (A.3), where a_x and a_z stand for the acceleration information provided by accelerometer.

$$\begin{aligned} C_A &= -\frac{ma_x}{\bar{q}S_{ref}} \\ C_N &= -\frac{ma_z}{\bar{q}S_{ref}} \end{aligned} \tag{A.3}$$

The dynamic pressure is determined from the climate database density values at radar measured altitudes. The accelerations are obtained from the IMU in accordance with the following relation extracted from Karlgaard et al. (2013):

$$\begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} = \begin{pmatrix} \tilde{a}_x \\ \tilde{a}_y \\ \tilde{a}_z \end{pmatrix} - \begin{bmatrix} -(\omega_y^2+\omega_z^2) & (\omega_x\omega_y - \dot{\omega}_z) & (\omega_x\omega_z + \dot{\omega}_y) \\ (\omega_x\omega_y - \dot{\omega}_z) & -(\omega_x^2+\omega_z^2) & (\omega_y\omega_z - \dot{\omega}_x) \\ (\omega_x\omega_z - \dot{\omega}_y) & (\omega_y\omega_z + \dot{\omega}_x) & -(\omega_x^2+\omega_y^2) \end{bmatrix} \begin{pmatrix} x_m \\ y_m \\ z_m \end{pmatrix} \tag{A.4}$$

In this equation $\tilde{a}_x$, $\tilde{a}_y$ and $\tilde{a}_z$ are the sensed IMU accelerations and x_m, y_m and z_m the position of the sensors relative to the CoM of the vehicle in the B-frame. The measured angular velocities and their derivatives are also provided by the IMU system. The lift and drag coefficients are derived using orientation geometry:

$$\begin{aligned} C_L &= C_N \cos\alpha - C_A \sin\alpha \\ C_D &= C_N \sin\alpha + C_A \cos\alpha \end{aligned} \tag{A.5}$$

In a similar fashion the moment coefficients are determined by:

$$\begin{aligned} C_l &= \frac{M_{\text{aero},i}}{\bar{q}S_{ref}b_{ref}} \\ C_m &= \frac{M_{\text{aero},j}}{\bar{q}S_{ref}b_{ref}} \\ C_n &= \frac{M_{\text{aero},k}}{\bar{q}S_{ref}b_{ref}} \end{aligned} \tag{A.6}$$

Angular rate information provided by the IMU is used to determine the static moments M_{aero}. As stated in Way et al. (2013), these moments are subject to significant errors. Yet another reason to couple a substantial sensitivity to the analysis. The data conferred in this appendix has been used as a baseline reference because detailed aerodynamic properties and flown trajectories are not available (yet) for Mars 2020.

B

Proof of Theorems

In this appendix several proof of theorem is provided regarding parts of the convex optimization method. In Section B.1 the formal proof is outlined that guarantees that the designed algorithm adheres to the threshold bounds of the thrusters. In the subsequent section, the Taylor Series expansion outlined in Section 5.2.2 is proven to maintain within the control magnitude limits throughout the landing sequence. Section B.3 outlines the comprehensive time discretization process, which also marks the end of this appendix.

B.1. Thrust Limits

In Section 5.2 part of the directly relevant proof has been conferred with regard to the lossless convexification method. This section focuses on the proof of the latter part of Eq. 5.21. This implies that the thrust limit must remain within the desired threshold values throughout the entire landing sequence. For this purpose the Hamiltonian is defined by Açıkmeşe and Ploen (2007) such that

$$H(\Psi(t)) = R_{12}(t)\Gamma^{\star}(t) + R_0(t) \quad \text{with} \quad R_{12}(t) = R_1(t) + \|R_2(t)\| \tag{B.1}$$

For completeness, the R-values are provided through

$$R_0(t) = \lambda_1(t)^T x_2(t) + \lambda_2(t)^T g \tag{B.2}$$

$$R_1(t) = \lambda_0 - \alpha\lambda_3(t) \tag{B.3}$$

$$R_2(t) = \frac{\lambda_2(t)}{m(t)} \tag{B.4}$$

In accordance with the pointwise principle, the following conditions are derived

$$\Gamma^{\star}(t) = \begin{cases} \rho_1, & \text{if} \quad R_{12}(t) < 0) \\ \rho_2, & \text{if} \quad R_{12}(t) > 0) \end{cases} \tag{B.5}$$

The condition to be proven must show that $R_{12}(t) \neq$ for $[0, t_f^{\star}]$. From $T_c^{\star} = R_2\Gamma^{\star} / \|R_2\|$ it is derived that

$$\dot{R}_1 = -\frac{\alpha}{m}\|R_2\|\Gamma^{\star} \tag{B.6}$$

From the fact that $\lambda_2 = mR_2$, it is derived that

$$\frac{d\|R_2\|}{dt} = \frac{1}{m}\frac{d\|\lambda_2\|}{dt} + \frac{\alpha}{m}\|R_2\|\Gamma^{\star} \tag{B.7}$$

From which follows that

$$m\dot{R_{12}} = \frac{d}{dt}\|\lambda_2\| = -\frac{1}{2\|\lambda_2\|}\lambda_2^T\lambda_1 = \frac{1}{2\|\lambda_2\|}(\|\lambda_1\|^2 t - \lambda_1^T a) \tag{B.8}$$

In the above equation vectors λ_1 and a are constant. As mass m is larger than zero, $\dot{R}_{12} \neq 0$ as for all t if $\lambda_1 \neq 0$. To complete the theorem, it has to proven that also for the case that $\lambda_1 = 0$ the statement $\dot{R}_{12} \neq 0$ is still valid.

Assume some scalars β_1 and β_2 which are both positive, for which holds that

$$0 = r_0 + \frac{g t_f^2}{2} + \dot{r}_0 t_f + \beta_1 \hat{n} \tag{B.9}$$

$$0 = \dot{r}_0 + g t_f + \beta_2 \hat{n} \tag{B.10}$$

In which $\hat{n}$ equals $\lambda_2 / \|\lambda_2\|$. Multiplying both inequalities with $-g/\|g\|$ leads to

$$0 = r_{0_1} + \frac{\hat{g} t_f^2}{2} + \dot{r}_{0_1} t_f \tag{B.11}$$

$$0 = \dot{r}_{0_1} + \hat{g} t_f \tag{B.12}$$

Here r_{0_1} and $\dot{r}_{0_1}$ are the initial position and velocity components, respectively. Notice that these are defined opposite to the axis of gravity direction. Combining both yields

$$r_{0_1} = -\dot{r}_{0_1}^2 / 2\hat{g} \leq 0 \tag{B.13}$$

$$\dot{r}_{0_1} > 0 \tag{B.14}$$

In Section 5.2 it has been established that $r_{0_1} > 0$ as the subsurface glide constraint. This would contradict the above established inequality and hence $R_{12}(t) \neq 0$ for all λ_1 and t, which formally completes the proof.

B.2. Taylor Series

In Section 5.2.2 outlined the Taylor series expansion about the established equation e^{-z}, which is continuously differentiable. This section focuses on the proof of this methodology. By applying the Mean Value Theorem one obtains

$$e^{-z} = e^{-z_0} \left\{ 1 - [z(t) - z_0(t)] + \tfrac{[z(t)-z_0(t)]^2}{2} \right\} - e^{-\hat{z}} \tfrac{[z(t)-z_0(t)]^3}{6} \tag{B.15}$$

In which $\hat{z} \in [z_0, z]$, which implies that

$$e^{-z_0} \left\{ 1 - [z(t) - z_0(t)]] + \tfrac{[z(t)-z_0(t)]^2}{2} \right\} \geq e^{-z} \tag{B.16}$$

This shows that the lower bound in Eq. 5.37 is larger than the one in Eq. 5.34. For the upper bound similar conclusions can be drawn, starting with

$$e^{-z} = e^{-z_0} \left\{ 1 - [z(t) - z_0(t)] \right\} + e^{-\hat{z}} \frac{[z(t) - z_0(t)]^2}{2} \tag{B.17}$$

Likewise $\hat{z} \in [z_0, z]$ this implioes that

$$e^{-z_0} \left\{ 1 - [z(t) - z_0(t)]] \right\} \geq e^{-z} \tag{B.18}$$

The latter proves that the upper bound in Eq. 5.34 is larger than the upper bound in Eq. 5.37. This marks the completion of the formal Taylor Series expansion proof.

B.3. Discretization of Time

This section outlines the mathematical principles behind the conversion of the infinite-dimensional problem to a finite one through time discretization. Recall that for time intervals Δt the temporal nodes are given by Eq. 5.39. A vector of parameters is defined such that

$$\eta = \begin{bmatrix} \mathbf{p}_0 \\ \vdots \\ \mathbf{p}_M \end{bmatrix} \quad \text{where} \quad \mathbf{p}_j \in \mathbb{R}^4 \tag{B.19}$$

The control variables $\mathbf{u}$ and σ are described in terms of Eq. B.19 and basis functions $\phi(\cdot)$, such that

$$\begin{bmatrix} \mathbf{u}(t) \\ \sigma(t) \end{bmatrix} = \sum_{j=0}^{M} \mathbf{p}_j \phi_j(t) \quad t \in [0, t_f] \tag{B.20}$$

The dynamics of the powered descent problem, which are differential equations in nature, may be expressed in terms of those coefficients, yielding

$$\mathbf{y}_k \equiv \begin{bmatrix} \mathbf{r}(t_k) \\ \dot{\mathbf{r}}(t_k) \\ z(t_k) \end{bmatrix} = \Phi_k \begin{bmatrix} \mathbf{r}_0 \\ \dot{\mathbf{r}}_0 \\ \ln m_{\text{wet}} \end{bmatrix} + \Lambda \begin{bmatrix} \mathbf{g} \\ 0 \end{bmatrix} + \Psi\eta \quad k = 1, \ldots, N \tag{B.21}$$

$$\begin{bmatrix} \mathbf{u}_k \\ \sigma_k \end{bmatrix} \equiv \begin{bmatrix} \mathbf{u}(t_k) \\ \sigma(t_k) \end{bmatrix} = \Upsilon\eta \quad k = 1, \ldots, N \tag{B.22}$$

In these equations the matrices $\Phi_k, \Psi_k, \Lambda_k$ and Υ_k are time indexed functions determined based on the the choice of basis functions. After Blackmore et al. (2010), in the research a linear piecewise nature of these functions has been adapted. Here-from follows that $M = N$, such that

$$\phi_j(t) = \begin{cases} \frac{t_j - t}{\Delta t} & \text{when} \quad t \in [t_{j-1}, t_j) \\ \frac{t - t_j}{\Delta t} & \text{when} \quad t \in [t_j, t_{j+1}) \\ 0 & \text{otherwise} \end{cases} \tag{B.23}$$

Define some matrices such that

$$E = [I_{3\times3} \quad 0_{3\times4}] \tag{B.24}$$

$$F = [0_{1\times6} \quad 1] \tag{B.25}$$

$$E_u = [I_{3\times3} \quad 0_{3\times1}] \tag{B.26}$$

$$E_v = [0_{3\times3} \quad I_{3\times3} \quad 0_{3\times1}] \tag{B.27}$$

Mathematical outline of the time discretized minimum landing error guidance problem

Cost Function:

$$\min_{N,\eta} \left\| E\mathbf{y}_N \right\|^2 \qquad \text{subject to:}$$

$$\left\| E_{\mathbf{u}} \Upsilon_k \eta \right\| \le \mathbf{e}_4^T \Upsilon_k \eta, \quad k = 0, \ldots, N$$

$$\rho_1 e^{-z_0(t_k)} \left[1 - (F\mathbf{y}_k - z_0(t_k)) + \frac{(F\mathbf{y}_k - z_0(t_k))^2}{2} \right] \le \mathbf{e}_4^T \Upsilon_k \eta \le \rho_2 e^{-z_0(t_k)} \left[1 - (F\mathbf{y}_k - z_0(t_k)) \right]$$

$$E\mathbf{y}_k \in \mathbf{X}, \quad k = 1, \ldots, N$$

$$F\mathbf{y}_N \ge \ln m_{\text{dry}}$$

$$\mathbf{y}_N^T \mathbf{e}_1 = 0$$

$$E_v \mathbf{y}_N^T = 0$$

$$\mathbf{y}_k = \Phi_k \begin{bmatrix} \mathbf{r} \\ \dot{\mathbf{r}} \\ \ln m_{\text{dry}} \end{bmatrix} + \Lambda_k \begin{bmatrix} \mathbf{g} \\ 0 \end{bmatrix} + \Psi_k \eta \quad k = 1, \ldots, N$$

The above problem can solved highly efficiently with a guarantee for global convergence towards the optimal solution. Within this framework N replaces time t as the time-of-flight parameter. The methodology for finding the optimal time-of-flight has been established in Section 5.2.4.

C

Numerical Methods

To obtain numerical solutions for the governing equations presented in the chapters on flight dynamics, guidance, navigation & control and autonomous optimization, numerical analysis methods are required. Numerical methods are a combined discipline of mathematics and computational computer sciences, utilized for the creation, implementation, propagation and analysis of continuous variables problems. Shortly, one can state that numerical analysis concerns all aspects from theoretical development to the efficient practical implementation of a problem. As more detailed models of assorted science problems are developed, more sophisticated numerical analysis methodologies become increasingly needed (Chasnov, 2012). Systems that undergo some sort of change are commonly described by differential equations. For the problem at hand, i.e., planetary precision landing, it has been made clear numerical methods are needed to propagate within the GNC loop. This appendix focuses on techniques used for integration of the predefined governing equations.

C.1. Governing Principles

This section focuses on the important mathematical principles that are connected to the numerical methods. Some key driving factors are established from which the efficiency and characteristics of a numerical integrator can be derived. The mathematical definition of a first order ordinary differential equation is given:

$$y' = f\left(x, y(x)\right) \tag{C.1}$$

To enhance the discussion the assumption is made that $y = y(x)$ is differentiable and x is an independent variable. However, as the above problem may raise an infinite number of solutions, an additional condition is established. From this initial condition a unique solution may be derived:

$$y(x_0) = y_0 \tag{C.2}$$

It is also possible to take the above system to a higher order of dimensions, thereby assuming a set n of ordinary first order differential equations:

$$\begin{aligned} y_1' &= f_1(x, y_1(x), ..., y_n(x)) \\ y_2' &= f_2(x, y_1(x), ..., y_n(x)) \\ &\vdots \\ y_n' &= f_n(x, y_1(x), ..., y_n(x)) \end{aligned} \tag{C.3}$$

Assuming an unique initial condition exists for each of these differential equations, the system is written in matrix and vector form:

$$\mathbf{y}' = f(x, \mathbf{y}) \tag{C.4}$$

$$\mathbf{y}(x_0) = \mathbf{y}_0 \tag{C.5}$$

In the vector space system of Eqs. (C.4) and (C.5) the following holds:

$$
\begin{aligned}
\mathbf{y}' &= (y_1', \dots, y_n')^T \\
f(\mathbf{x}, \mathbf{y}) &= \begin{pmatrix} f_1(x, y_1, \cdots, y_n \\ \vdots \\ f_n(x, y_1, \cdots, y_n \end{pmatrix} \\
\mathbf{y}_0 &= (y_{1_0}, \dots, y_{n_0})^T
\end{aligned}
\tag{C.6}
$$

In case the first order differential equation of Eq. (C.1) is considered to be of a general form with order m, the differential equation reads:

$$
y^{(m)}(x) = f(x, y(x), y'(x), \dots, y^{(m-1)(x)}) \tag{C.7}
$$

Auxiliary functions can be introduced such that the m^{th} order differential equation can be transferred to a system of first order differential equations. These auxiliary functions are:

$$
\begin{aligned}
z_1(x) &= y(x) \\
z_2(a) &= y'(x) \\
&\vdots \\
z_m(x) &= y^{(m-1)}(x)
\end{aligned}
\tag{C.8}
$$

The complete system of first order differential equations then reads:

$$
\mathbf{z}' = \begin{pmatrix} z_1' \\ \vdots \\ z_{m-1}' \\ z_m' \end{pmatrix} = \begin{pmatrix} z_2 \\ \vdots \\ z_m \\ f(x, z_1, z_2, \cdots, z_m) \end{pmatrix} \tag{C.9}
$$

The equations of translational motion and rotational motion derived for entry, descent and landing form various systems of n ordinary differential equations. Both the three as well as the six degree of freedom equations have been outlined. The former requires the solving of a system of six differential equations while the latter requires twelve coupled equations at a minimum. In the next section the numerical integrators for solving these type of equations are outlined.

C.2. NUMERICAL INTEGRATORS

Numerical integration is the process in which approximate solutions are found for definite integrals using techniques to be discussed in this section. Uberhuber (1997) makes a distinction between univariate and multiple variable integrals through the terms of definition quadrature and cubature, respectively. The integrators are used for state propagation purposes (Pocha, 1987). In this section the mathematical procedures related to some types of integration methods are outlined. Even though the Euler integration is not robust enough for the problem at hand, it is still outlined because of the insight it provides regarding integration methodologies. The initial value problem Euler integrator is discussed in Section C.2.1. A more sophisticated form for solving initial value problems is the Runge-Kutta 4 (RK4) method to be delineated in Section C.2.2.

C.2.1. EULER INTEGRATOR

A well-known but in terms of accuracy and stability limited numerical integration method is the one of Euler. Its advantages are the simple implementation and the fact that it provides a useful starting point for understanding integration methods. The essentials of the working principle are based on finding a slope at a certain point and with a step-size h then calculate a new slope to alter the direction to that desired new slope (Evans, 2009). The equation that describes the general form for integrators and gives and governs the approximation of the next value is defined:

$$
\mathbf{x}(t_0 + h) \approx \mathbf{x}_0 + h\Phi = \eta(t_0 + h) \tag{C.10}
$$

In this equation, $\mathbf{x}$ is the physical truth value, $\mathbf{x}_0$ the initial value, η the numerical approximation and h the step-size. It is the increment function Φ that defines the integration and for the Euler integrator this function is given in Eq. (C.11).

$$
\Phi_{Euler} = \dot{\mathbf{x}}_{\mathbf{0}} = f(t_0, \mathbf{x}_0) \tag{C.11}
$$

Since the integration is performed over time, the function **f** is a time-derivative of the vector **x** and thus a function of t_0 and/or $\mathbf{x}_0$. Notice that this method has only been described as it provides insight into the propagation matter. This research adapted a more robust method, further described in the next section.

C.2.2. RUNGE-KUTTA 4 INTEGRATOR

A quite celebrated numerical method for finding solutions to differential equations is the Runge-Kutta (RK4) method. It has the advantage that the step length may be adjusted at any time in the computation course. Also, if implemented correctly, the results are (in a relative sense) quite accurate (Milne, 1950). In a similar fashion, the RK4 method is based on Eq. (5.1), but the increment function is different from that of Euler. For the RK4 method the increment function is given in Eq. (C.12). Four different approximations are used to describe the changes of the parameter(s) during the integration step. These approximations are then averaged. Notice that this is still a single-step technique as no information comes from x_{n-1}.

$$\Phi_{RK4} = \frac{1}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4) \tag{C.12}$$

In this equation $\mathbf{k_i}$ are time-derivative vectors at various points within the integration step. These vectors are calculated using the Equations given in C.13.

$$\begin{aligned}
\mathbf{k_1} &= f(t_0, \mathbf{x}_0) \\
\mathbf{k_2} &= f(t_0 + h/2, \mathbf{x}_0 + h\mathbf{k_1}/2) \\
\mathbf{k_3} &= f(t_0 + h/2, \mathbf{x}_0 + h\mathbf{k_2}/2) \\
\mathbf{k_4} &= f(t_0 + h, \mathbf{x}_0 + h\mathbf{k_3})
\end{aligned} \tag{C.13}$$

Notice that the derivatives at t_0, $t_0 + h/2$ and $t_0 + h$ are used in this integration method.

C.3. STOCHASTIC SIMULATIONS

Stochastic simulations belong to a class of numerical methods in which the behavior of both variables as well as the system as a whole is assessed in case stochastic (random) probabilities are introduced (Gilmore, 2006). A broad spectrum of engineering and science problems comprise phenomena that are random in space and time (Field, 2008). Planetary precision landing is a perfect example of a physical process that involves various uncertainties. Because the nature of the uncertainties is random, stochastic simulations create a framework of random values from which outputs are computed and recorded. New random variables are introduced during each step until a sufficient amount of data is accumulated. Based on the total data output a most probable conclusion is drawn regarding the likelihood and variety range of the estimates. A stochastic process is a mathematical model utilized for conducting the simulation. The Monte Carlo simulation is a very powerful tool to generate these sample estimates. Across the numerous papers read on planetary precision landing, the Monte Carlo is by far the most utilized method. This methodology has therefore been adapted into the research.

C.3.1. MONTE CARLO TECHNIQUE

The Monte Carlo technique is considered to be a type of stochastic sampling technique. The principle is rather simple: parameters are obtained in a random way for which the outcome of the function is evaluated. This is repeated for a number of samples after which the absolute minimum from these samples is located. Commonly, the normal distribution is used for the dissemination of the random values. In probability theory this continuous probability distribution is also known as the Gaussian distribution with probability density:

$$f(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{C.14}$$

In this equation μ is the mean of the distribution, σ the standard deviation and σ^2 the variance. It is assumed that the three-sigma rule holds such that all values lay within an empirical treat of $3\sigma = 99.7\%$ from the mean. To ensure the usefulness of this method and dub the central limit theory applicable, the number of samples needs to be sufficiently high. General advantages of the Monte Carlo method are that the full spectrum is covered with relatively small numbers of samples and the CPU time is limited. Furthermore, it is easy to implement, check and interpret. Main disadvantages are the fact that the method is not robust and that the resolution is limited. However, the advantages are of greater value with respect to the disadvantages. Furthermore, verification through the grid search technique should highlights conceivable flaws.

D

Algorithm User Guide

This document encompasses a brief descriptive outline regarding the developed modeling and simulation algorithm for precision landing on planetary bodies. **Please notice that these algorithms are only available for use at NASA's Jet Propulsion Laboratory and to a limited extent at Delft University of Technology**. The convex optimization guidance method has been adapted during the research conducted at the Jet Propulsion Laboratory. The absolute soft landing guidance problem is characterized as a finite horizon optimal control problem. That is, the optimal performance measures for a finite time $t = [0,\ t_f]$ within system control. All simulations have been carried out in Matlab, specifically by using the CVX modeling system for convex optimization. Part D.1 describes the process for downloading and installing CVX to make Matlab compatible for running the simulations. To allow for ease of user interface, Part D.2 outlines the general structure of the contained files. A clear distinction is made between landing on Mars and landing on Titan, of which the contained files have been delineated in Sections D.3 and D.4, respectively. These bodies are of course different in essence, in the sense that Mars has a rather thin atmosphere for which conveniently rocket thrusters are utilized in the terminal landing sequence. For Titan on the other hand, a Precision Aerial Delivery Systems (PADS) parafoil is proposed due its conveniency within the Saturian moon's thick atmosphere. Yet, the nature of the (convex optimization) precision landing guidance problem for both celestial bodies is rather similar.

D.1. Matlab CVX - Setup Process

This section outlines the setup process for making Matlab compatible with CVX disciplined convex programming. It is important to mention that CVX is not a native Matlab toolbox that can be installed from the Mathworks environment, hence additional yet straightforward steps are required for compatibility purposes. The process includes a download and several installation steps. For general information regarding CVX Matlab software for disciplined convex programming, please be referred to `http://cvxr.com/cvx/`. At the bottom of the page there is a link to a PDF file that contains an extensive CVX Users' Guide.

Download

At the time of writing the most current software version release is Matlab CVX 3.0 Beta, which can be downloaded from `http://cvxr.com/cvx/beta/`. The software is compatible with Linux, Mac and Windows 64 bit, and Windows 32 bit. Download CVX for the desired operating system and, if needed, unzip the downloaded folder. Make sure that within the main downloaded folder 'cvx' there is a .m file named: cvx_setup.

Installation

After completion of the download process, the next step is to (permanently) setup CVX within Matlab. Move to the Matlab root directory on your device (where Matlab has been installed) and open the 'toolbox' folder. On a Windows device the source file is often found in path 'C: \Program Files\MATLAB'. On a Mac device move to Finder, open Applications, right-click on Matlab and click 'Show Package Contents'. After localizing toolbox, copy the downloaded 'cvx' folder (entirely) into the 'toolbox' folder. Now open the .m file: cvx_setup from the following path: '..\Matlab\toolbox\cvx' and run it from the opened directory (do not copy it elsewhere). Once opened, click the Matlab run button. Installation hereafter should take several seconds, depending on the computation speed of the device. After successful installation Matlab returns the following prompt as an completion response:

```
---------------------------------------------------------------------------
Setting CVX paths...done.
Saving update path...done.
Searching for solvers...8 shims found.
2 solvers initialized (* = default):
 *  SDPT3    4.0       {cvx}/sdpt3
    SeDuMi   1.34      {cvx}/sedumi
3 solvers not found:
    ECOS     https://github.com/ifa-ethz/ecos
    GLPK     http://glpkmex.sourceforge.net/
    SCS      https://github.com/cvxgrp/scs
2 solvers require a CVX Professional license:
    Gurobi             {cvx}/gurobi/maci64
    Mosek    unknown    {cvx}/mosek/maci64
1 solver skipped due to other errors:

        This solver shim is not compatible with CVX 3.0. Please contact
            the authors for an update.
Saving updated preferences...done.
Testing with a simple model...done!
---------------------------------------------------------------------------
To change the default solver, type "cvx_solver <solver_name>".
To save this change for future sessions, type "cvx_save_prefs".
Please consult the users' guide for more information.
---------------------------------------------------------------------------
WARNING: An existing copy of "square.m" was found in your MATLAB path:
   /Applications/MATLAB_R2018b.app/toolbox/signal/signal/square.m
Models using SQUARE() in CVX expressions will not be affected; but outside
of CVX, this version will be used, and it likely has a different meaning.
To avoid any confusion, just use X.^2 instead of SQUARE(X) in CVX.
---------------------------------------------------------------------------
```

D.2. FILES OVERVIEW

This section provides a higher-level overview of the files to allow interested users to navigate more easily through the folders. **Please be referred to Figure D.1 to follow up with the subsequent outline.** The main folder is named **Simulations**, after which a first split is made, based on the distinct planetary body, i.e., either **Mars Landing** or **Titan Landing**. The next division is based on model fidelity. For three-degrees-of-freedom (3DoF) simulations a process by the name of lossless convexification is adopted, which is a class of convex optimization utilized to ensure absolute soft landing (zero position error and zero touchdown velocity). Comprehensive literature and the main Lemma that proofs transfer of the inherently nonconvex guidance problem to the convex domain without loss in generality can be found at https://doi.org/10.1109/TCST.2012.2237346. For the successive convex optimization guidance literature, which is basically an extension of the lossless convex guidance problem that allows for the advanced application of six-degree-of-freedom (6DoF) high-fidelity dynamics and likewise allows time to be defined as an optimization variable, please be referred to https://doi.org/10.2514/6.2018-0617. As depicted in the figure, both the Mars and Titan landing simulations include a distinct **Lossless 3DoF** and a **Successive 6DoF** folder. It is important to mention that within each of these folders, a folder named **Reference** can be found which includes research papers (PDF-files) which have been utilized for that respective simulation. For example, '..\Simulations\TitanLanding\Lossless\Reference' particularly includes literature on the Titan environment, 3DoF parafoil dynamics, lossless convex optimization, etc. For a comprehensive overview of how all of these references have been fused together in the research, the reader is referred to the report.

In the final part of this section the lowest level of Figure D.1 is described. The Guidance for Fuel Optimal Large Divert (**G-FOLD**) algorithm is found in the **Lossless 3DoF** folder, while the full powered descent algorithm is included in the **Successive 6DoF** folder. Both these Mars landing simulations are to be described in more detail in Section D.3. In a similar fashion, for Titan landing simulations there is a Guidance for Control Optimal Large Divert (**G-COLD**) algorithm within the **Lossless 3DoF** folder. The full parafoil descent algorithm is to be found in the **Successive 6DoF** folder. The two latter matters, regarding simulating and modeling Titan precision landing, are broadly described in Section D.4. The section on Titan precision landing also marks the end of this user manual.

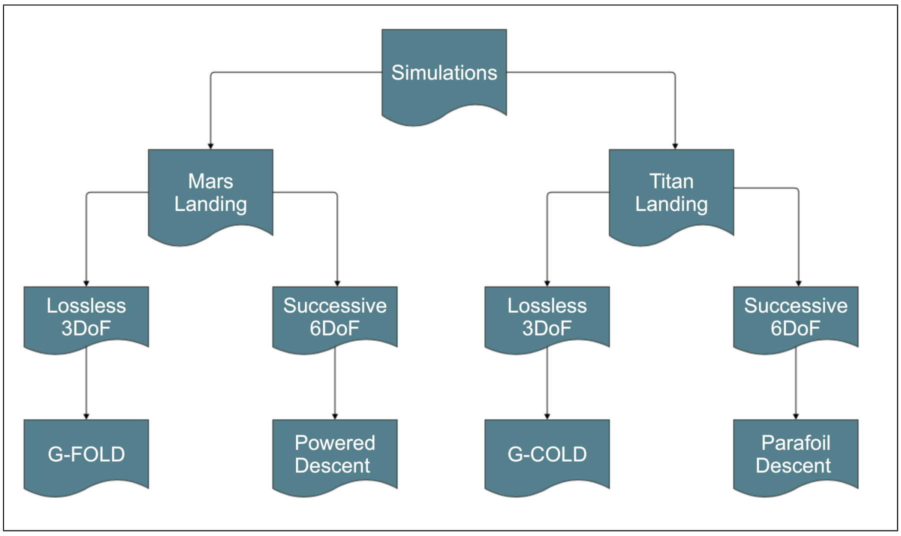


Figure D.1: Overview of the folders contained within the main folder **Simulations**.

D.3. Mars Landing Simulations

This section outlines the Matlab files contained within the **Mars Landing** folder. All Mars simulations have been performed as an exercise of algorithm verification. Extensive literature for Mars landing is available and several flight software has even been validated (such as G-FOLD). Please note that in all the Matlab files comments are included for ease of tracking regarding what is being computed and what variables are relevant. The section starts with the outline of the G-FOLD algorithm .m files, which is based on lossless convex optimization and comes in two types. The second and likewise final part outlines the main successive convex optimization powered descent algorithm.

1. **Guidance for Fuel Optimal Large Diverts**
   As before-mentioned, the lossless powered descent guidance algorithm adapted in the research is referred to as G-FOLD. This algorithm can be defined in two fashions. Fuel optimal guidance practically ensures zero touchdown velocity and zero position error while adhering to the on-board fuel limit constraint. This algorithm has been described beneath first. Building further upon this algorithm, minimum landing error guidance simulates cases in which a trajectory with zero error magnitude turns out to be infeasible. In the latter case the guidance algorithm has to ensure safe landing while minimizing position error and flying a fuel optimal path.

   - **Fuel Optimal Guidance**
     The fuel optimal guidance folder by itself contains three folders: **Papers**, **Simulations** and **Figures**. The **Papers** folder basically contains the relevant literature for this particular algorithm. The latter folder contains the simulations results (i.e., **Figures**). This section focuses on the **Simulations** folder, which contains the .m simulation files. However, in order to follow up with discussion, please open `https://doi.org/10.2514/1.27553`, as the algorithm is exactly reproducing the results in this paper (which moreover can also be found in the **Papers** folder).

     The **Simulations** folder within the **Fuel Optimal** folder contains the following elements:

     1. **Constants**
        The two files contained within are **PDMarsConstants.m** and **randomposvel.m**

* Notice that folder names are marked with color 🟦

* Notice that the .m file names are marked with color 🟦

PDMarsConstants.m: contains the needed constants for Mars Powered Descent Guidance. This includes fundamental physical constants, spaceflight constants, Mars bulk parameters and thruster data of the descent stage vehicle. Comments within the file point out the specific meaning of variables as well as their units.

randomposvel.m: is a random state (position and velocity) generation file. The generated states are within a realistic magnitude range (specified in the file for each state element). This file serves the purposes for testing the robustness of the algorithm by randomly varying starting states to be able to observe how the guidance system reacts to it.

2. **Final thrust, subsurface and glide-slope**
   Each of these three folders contains two .m files, **figureX.m** and **GFOLD.m**. The final thrust folder additionally also includes **plot3D.m**

   figureX.m: this file contains the code that generates the figures for visualization. The X is a number which refers to which figure of the earlier defined paper is being reproduced. To clarify with an example: the final thrust folder containts figure5.m, which reproduces figure 5 of the paper.

   GFOLD.m: is the actual working guidance algorithm that utilizes the earlier defined PDMarsConstants.m. Notice that it does not utilize the randomposvel.m, as starting states for this case are fixed to reproduce the paper figures. The algorithm starts with defining the remaining constants (vehicle mass, start and final state conditions, gravity vector and time constants). Optimal time is assumed to be a given in the current case. The remainder of the file includes the optimal CVX guidance computation sample code for G-FOLD:

```
% Sample of the powered descent guidance algorithm
cvx_solver SeDuMi                  % Self-dual minimization to solve SOCP
cvx_begin                          % Initialize convex optimization

variables (), maximize()

subject to
% Initial conditions
eq(r(:,1) ..., eq(v(:,1) ..., eq(z(1),  ...

% Final conditions
eq(r(:,N) ... , eq(v(:,N) ..., eq(u(:,N) ...

%% Dynamics
for i=1:N-1
    eq(v(:,i+1) ..., eq(r(:,i+1) ..., eq(z(i+1) ...
end
%% Numerical algorithm
for i=1:N
    %% Control constraint
    norm(u(:,i))...

    %% Taylor series expansion
    z0 = ..., z1 = ..., mu1 = ...

    % Second-order cone and linear inequality
    sigma(i)...
end
cvx_end                             % End convex optimization
```

The above code is a sample of the G-FOLD code which has been conferred for explanation purposes. The cvx_solver is defined generally defined first, depending on the application. The default solver is SDPT3, used for solving standard semi-definite programs (SDP). Generally, self-dual minimization (SeDuMi) is faster, through the implementation of a self-dual embedding technique that performs optimization over self-dual homogeneous cones. This however comes at the expanse of reliability. As none of the solvers are perfect, the solver type has been varied throughout the research to find out which solver is preferred for which application. Following up, cvx_begin tells CVX to start the optimization process. The variables should be defined, which is this case include position (r), velocity (v), control parameter (u), mass logarithm (z) and slack variable (sigma). The cost function in the above example is defined as maximizing z, which implies the maximization of mass (and hence the minimization of fuel consumption). The cost function is subject to certain equality conditions (initial and final), dynamics and a defined numerical algorithm. Please be referred to the enhancement paper on convex optimization guidance for precision landing: `https://doi.org/10.2514/6.2008-6426`. This document perfectly outlines the derivation of the dynamics, explains why the mass logarithm has been adapted as an optimization variable and defines the numerical convexification algorithm that naturally follows from that. The termination of the algorithm is established through cvx_end.

Based on mission requirements, additional state and control constraints can be enforced within the convex optimization framework. Example of such are the glide-slope constraint (avoids sub-surface flights through and glide cone at an angle γ) and the pointing constraint (copes with the requirements for TRN by restricting pointing to a certain maximum angle θ.

plot3D.m: this file is contained in the final thrust folder and plots the motion in 3D Euclidean space. Notice that this is not a verification algorithm but rather for visualization purposes only. The guidance algorithm working principle is still operating under the same code.

3. **Golden Search**
   Additional to **figureX.m** and **GFOLD.m**, the Golden Search folder includes **conditions.m**, **GoldSearch.m** and **minfuel.m**.

   conditions.m: this .m files adds certain simulations parameters additional to **PDMarsConstants.m**, which are relevant for this particular simulation.

   GoldSearch.m: this applies a Golden Search algorithm (line search) to find the optimal time of flight. This is briefly explained in the paper. Notice that the Golden Search algorithm has no theoretical guarantee of finding t_{opt} in real-time. This is one of the reasons for applying successive convexification in a later stage. The **GoldSearch.m** applies consecutive iterations of minimum fuel optimizations (using **minfuel.m**) through applying the golden ratio for convergence to an acceptable threshold value.

   minfuel.m: this .m file is a function containing the G-FOLD algorithm in function form (to allow for ease of iterations). The input values are all defined throughout the other files. The output is the fuel usage, which is compared relative to earlier computed fuel values. The **GoldSearch.m** uses this information to determine the direction in which to convergence.

- **Minimum Landing Error Guidance**
  The minimum landing error guidance algorithm, also referred to as soft landing problem, is an algorithm that minimizes position error relative to the target in case zero position error turns out to be infeasible based on the dynamics and fuel limit. In essence it is an extension of the G-FOLD algorithm, just defined in a slightly different fashion. Likewise, the folder itself contains: **Papers**, **Simulations** and **Figures**. The **Simulations** folder contains several of the files which already have been conferred above: **PDGMarsConstants.m**, **conditions.m**, **GoldSearch.m** and **minfuel.m**. Additionally, a few files are added to simulate the guidance problem, which are: **guidance.m**, **minfuelPE.m**, **plotFuel.m** and **plotMotion.m**.

guidance.m: this is the main algorithm that basically runs all the other files. First, **GoldSearch.m** is run to obtain optimal time, followed by minimum landing error guidance (**minfuelPE.m**). The guidance algorithm terminates by running (**minfuel.m**).

minfuelPE.m: This function computes a minimum fuel trajectory based on a given optimal time and returns the minimum position error within the fuel limit. This can either be absolute zero or near it (depending on trajectory feasibility). This scenario is realistic based on in EDL trajectories. The minimization or r from target is subsequently used in **minfuelPE.m**. To compute the minimum fuel trajectory within the given feasible range.

plotFuel.m: plots the total fuel consumption for a wide ranges of time to proof that the optimal time found is indeed corresponding to minimum fuel.

plotMotion.m: plots the time history of the motion. This includes horizontal plane transfer, vertical plane transfer, angle above surface, throttle, position, velocity, acceleration, thrust angle and rate.

2. **Successive Convexification Powered Descent Guidance**

Within the successive convex guidance framework, time becomes an optimization variable. Furthermore, six-degree-of-freedom is captured and each state is propagated in a discrete step-wise fashion. To be able to do so, a first guess is needed regarding all convex variables. This is established through **iterationzero.m**, a first iteration of the solution. Notice that is solution can be dynamical inconsistent and feasible (guaranteed convergence is the power of the successive method). The **guidance.m** is the main file that runs the finite horizon optimal control algorithm. In order to follow up with discussion, please open `https://doi.org/10.2514/6.2018-0617`. The file **motionplots.m** is the equivalent of the earlier delineated **plotMotion.m**. The file **parameters.m** defines the simulations parameters. Finally, to cope with the requirements of 6DoF dynamics, an omega tensor file (**omega.m**) and a file computing the direction cosine matrix (**CBI.m**) are included.

iterationzero.m: this file runs the initial guess guidance algorithm. The control effort multiplied by a virtual control parameter is the cost function to be minimized. This virtual control parameter is needed to make the initial guess a feasible trajectory. The framework is very similar to the earlier conferred **GFOLD.m** setup.

guidance.m: the main file runs the first iteration (**iterationzero.m**), from which it extracts the relevant parameters needed for the subsequent iterations. To enforce convergence in a controlled fashion, the change in time, position, velocity and control is restricted between each iteration. Also, a certain maximum number of iterations is defined in case the threshold desired tolerance is not reached before. On a side note, the latter has not been encountered during the research. The convergence requirements are established as follows:

```
for i = 1:N
    %% Convex parameter bounds (enforce convergence)
    le(tf - tfk, 1);         % Maximum delta time
    le(norms(r-rk), 1);      % Maximum delta position norm
    le(norms(v-vk), 1);      % Maximum delta velocity norm
    le(norms(u-uk), 1);      % Maximum delta control norm

    %% Iteration conditions
    if (k == kmax) || (abs((cvx_optval - cvxk))<=vct)
        break
    end
end
```

parameters.m: is the file in which the simulation parameters are defined. This includes vehicle mass data, thruster data, flight conditions, state conditions, gravity vector, flight moment data and initialization parameters.

omega.m: this function takes the angular velocity ω-vector as input and returns the ω-tensor, also defined as the skew-symmetric matrix. This computation is needed in propagation of quaternion dynamics.

CBI.m: this function takes the quaternion vector as input and returns the direction cosine matrix. This computation is required in the process of propagating the torque dynamics.

D.4. TITAN LANDING SIMULATIONS

Precision landing on Titan formed the core of the research, as employing convex optimization guidance for the parafoil terminal descent phase has not been extensively researched before. This section outlines the Matlab files contained within the **Titan Landing** folder. The aero-maneuvering dynamics and control formulation for parafoil precision landing on Titan is mainly based on `https://doi.org/10.1109/AERO.2019.8742230`. The remainder of the research utilized during the research is contained within the **Papers** folder. Anew note that in all the Matlab files comments are included for ease of tracking regarding what is being computed and what variables are relevant. The section starts with the outline of the G-COLD algorithm .m files, which is based on lossless convex optimization. The final part outlines the main successive convex optimization parafoil descent algorithm.

1. **Guidance for Control Optimal Large Diverts**

   As before-mentioned, the lossless parafoil descent guidance algorithm adapted in the research is referred to as G-COLD. This is defined as such due to the fact that for a parafoil the control effort rather than the fuel usage is minimized. This algorithm has been simulated in two fashions. There is the **Ideal** case, in which it is assumed that there is no wind present in the terminal descent. The **Wind** inclusion algorithm assumes the contrary. The files contained for each of these cases are elaborated on beneath.

   - **Ideal Conditions**
     Within this folder there are several folders that can be found that include simulation plots and the .fig files of these respective figures. Please be referred to the earlier mentioned report for an outline of these simulation results. For the ideal case, there are two main .m files contained that run the simulation: **PDconstants.m** and **PDalgorithm3D.m**. These files contain the respective constants and main algorithm, respectively.

     PDconstants.m: includes the relevant constants needed for the simulation, such as Titan gravity. Notice that some Martian data is also included to allow for algorithm generalization in a later stage. In this way the algorithm can be run for either planetary body without having to worry about defining each planetary bulk variable at each time.

     PDalgorithm3D.m: runs the main guidance simulation based on a reduced 3DoF model. Initial position and velocity state conditions are generated in a random fashion, based on realistic magnitudes for Titan terminal descent. The cost function within this framework is the minimization of the control effort. Finally, this file also generates a 3D-motion plot and the time history of trajectory, position and velocity.

   - **Wind Inclusion**
     Contains a more sophisticated 3DoF model of the parafoil landing sequence. Anew, there is **figures** and **plots** folder, containing .fig and .png simulation files, respectively. Besides the .m files **PDconstants.m** and **PDalgorithm3D.m** that this folder also contains, there are additional ones. These are:
     parameters.m, **angleofattack.m angleofsideslip.m**, **CDc.m**, **CL.m**, **CY.m**, **density_titan.m** and **wind_titan.m**. Each file is briefly described to allow for ease of user interface:

parameters.m: includes the additional relevant constants needed for the simulation. These are the aerodynamic elements, discretization scheme. This file runs **randomstate.m**, which are randomly generated conditions that have been saved to maintain consistent starting conditions throughout the simulation to allow for more compelling comparisons.

angleofattack.m: this function takes the spacecraft velocity and wind velocity as input values, from which it computes the effective velocity. Subsequently it calculates the angle-of-attack α.

angleofsideslipe.m: this function takes the spacecraft velocity and wind velocity as input values, from which it computes the effective velocity. Subsequently it calculates the side-slip angle β.

CDc.m: this function takes the angle-of-attack and individual drag coefficient contributions, from which it computes the drag coefficient. Please notice, this does not include the drag coefficient contribution due δ_s, because this is a convex optimization variable defined within the main framework.

CDc.m: this function takes the angle-of-attack and individual lift coefficients contributions, from which it computes the lift coefficient. Please notice, this does not include the lift coefficient contribution due δ_s, because this is a convex optimization variable defined within the main framework.

CY.m: this function takes the side-slip angle and individual side-force coefficients contributions, from which it computes the total side-force coefficient.

density_titan.m this function takes the flight altitude and returns the local air density on Titan. Be referred to `https://doi.org/10.1016/j.pss.2012.05.015` for information regarding computing environmental properties of the Saturnian moon.

wind_titan.m this function takes the flight altitude and returns the wind velocity in three directions: zonal, meridional and zonal. Likewise, be referred to `https://doi.org/10.1016/j.pss.2012.05.015` for information regarding computing environmental properties of the Saturnian moon.

2. **Successive Convexification**
   In the final section of this algorithm outline, the successive parafoil descent guidance algorithm is presented. Within the successive guidance framework, time is an optimization variable. Six-degree-of-freedom is captured and each state is propagated in a discrete step-wise fashion. This requires a first guess regarding all convex variables. This is established through **iteration.m**, a first iteration of the solution. Like for Mars case, the **guidance.m** is the main file that runs the finite horizon optimal control algorithm. Also, to cope with the requirements of 6DoF dynamics, an omega tensor file (**omega.m**) and a file computing the direction cosine matrix (**CBI.m**) are included. The file **motionplots.m** plots the desired time histories of the motion. The file **parameters.m** defines the simulations parameters. There are several files from the G-COLD case which are anew applied. Additionally, there are more aerodynamic computation files included to cope with the requirements for 6DoF dynamics. These are: **aeroparameters.m**, **aerodynamics.m**, **flightpath.m**, **Rbn.m**, **Rpw.m** and **quat2eul.m**:

   aeroparameters.m: includes the additional relevant constants needed for the simulation. These are the parafoil properties, (apparent) mass and inertia computations. This file also includes the needed distances between spacecraft part and the center of mass to be able to compute the parafoil moments.

   aerodynamic.m: this file runs an initial guess on the aerodynamic forces and moments for both parafoil and payload. It largely utilizes the aero-parameters priory established.

flightpath.m: this function takes the spacecraft velocity and wind velocity as input values, from which it computes the effective velocity. Subsequently it calculates the flight-path angle γ.

Rbn.m: this function computes the transformation matrix from the navigation to the body frame.

Rpw.m: this function computes the transformation matrix from the wind to the parafoil frame.

Rpw.m: this function computes the transformation matrix from quaternions to Euler angles.

Within the convex optimization frame, the aerodynamic force naturally plays a prominent role in parafoil dynamics. The dynamics is propagated in a discrete fashion. This includes a dynamic pressure q computation, local air velocity v_u, the aero-control coefficients, and the wing control parameters δ_s and δ_s. Furthermore, the value for the payload drag coefficient is included. At last, the total drag value is multiplied by 1.3 to account for suspension line drag contribution. By mentioning this, the user manual of the algorithm is concluded.

E

Additional Verification Results

As continuation of Chapter 6, this appendix provides several additional verification results. These results further establish the proper working of the designed algorithm and are presented for the enhancement of future research on the matter of planetary precision landing. For a detailed outline of the algorithm itself, the reader is referred to Section 6.1, Section 6.2 and Appendix D. For each figure conferred in this appendix, a short description is provided regarding the results. This includes a short descriptive outline of the main similarities and differences with respect to industry's best practise on the subject matter. The caption of each figure refers to the respective paper of which the results have been reproduced.

- **Figure E.1**: This figure marks the run of the plain convex optimization algorithm, thus without any constraints. From the position profile it is observed that the spacecraft gets below the surface, which obviously is not a desired phenomenon. This further establishes the need for the glideslope constraint. The results are near identical to the ones presented in Açıkmeşe and Ploen (2007).
- **Figure E.2**: Different from the previous figure, this figure contains a glide-slope constraint at angle of 4°. This becomes evident from the position figure, as the trajectory remains well above the surface throughout the entire simulated flight. Likewise, results are near identical to the ones presented in Açıkmeşe and Ploen (2007).
- **Figure E.3**: In correspondence with Fig. E.2, this figure depicts the tangent line due the glide-slope constraint in two-dimension Euclidean space. This figure corresponds exactly to the result obtained from the algorithm of Açıkmeşe and Ploen (2007).
- **Figure E.4 and Fig. E.5:** These figures correspond to the results of Blackmore et al. (2010). However, there is a distinct difference as the algorithm of the paper reports the trajectory to be infeasible, ensuring a safe landing at a distance of 404 m from the target. The designed algorithm however managed to guarantee soft landing, thus exactly at the predefined landing location. The optimal time-of-flight of 78 s and fuel usage of 400 kg however are in correspondence. Notice that for this simulation the velocity was altered to equal to $\dot{\mathbf{r}}_0 = [-75, 40, 100]^T$, such that there is an initial velocity vector with non-zero elements in all three directions of Euclidean space.

For convenience purposes Tab. 6.2 is anew provided without cross-reference nor caption

Parameters	**Fig. 5**	**Fig. E.1**	**Fig. 6**	**Fig. 6.2**	**Fig. 7**	**Fig. E.2**	**Fig. 10**	**Fig. 6.3**
Optimal time (s)	72	73	75	75	81	81	69	71
Fuel consumption (kg)	387.9	389.4	390.4	391.8	399.5	399.9	293.6	293.4

Conclusively it is again emphasized that based on the five unit verifications outlined in Section 6.2, the designed algorithm has been extensively verified. Based on this verification the algorithm has been extended to become compatible for landing on Titan using a parafoil. Furthermore, the algorithm has been extended to include obstacle avoidance guidance to ensure mission safety in multiple mission scenarios.

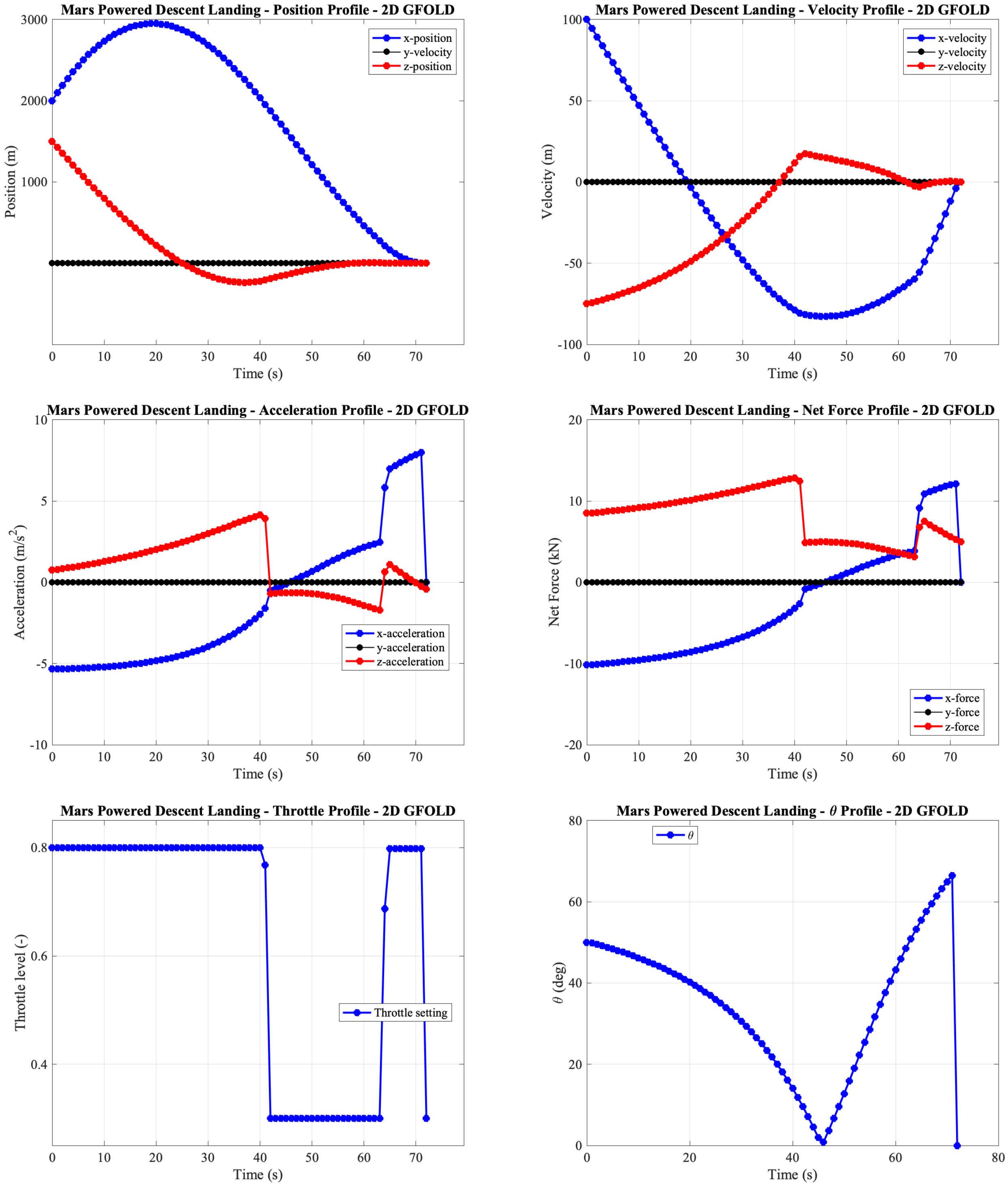


Figure E.1: Mars in-plane manoeuvre with no constraints enforced.. Time history of the position, velocity states, acceleration, net force, throttle setting and pointing orientation. Figures is correspondence with Fig. 5 of Açıkmeşe and Ploen (2007).

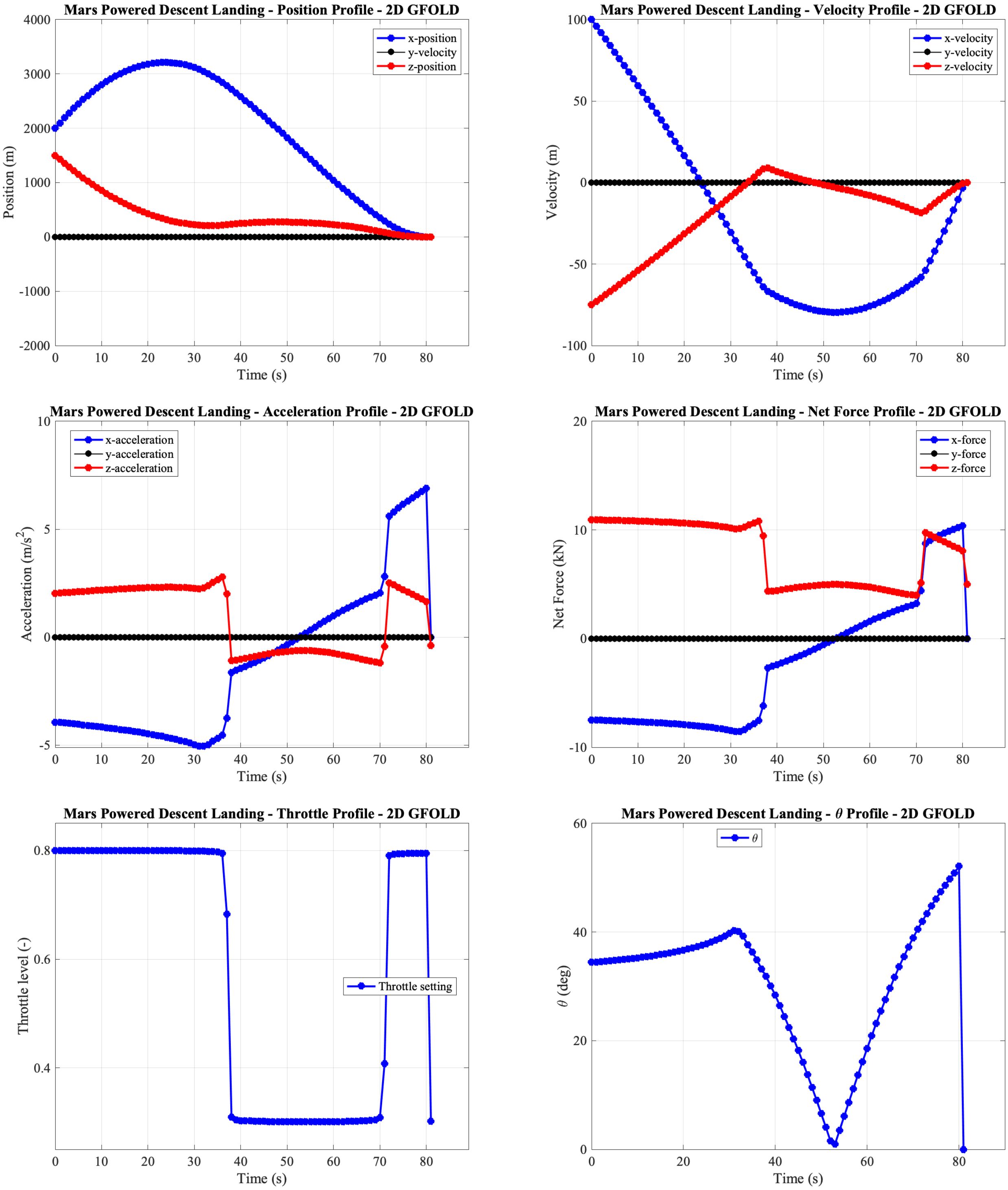


Figure E.2: Mars in-plane manoeuvre with the glide-slope constraint enforced. Time history of the position, velocity states, acceleration, net force, throttle setting and pointing orientation. Figures is correspondence with Fig. 7 of Açıkmeşe and Ploen (2007).

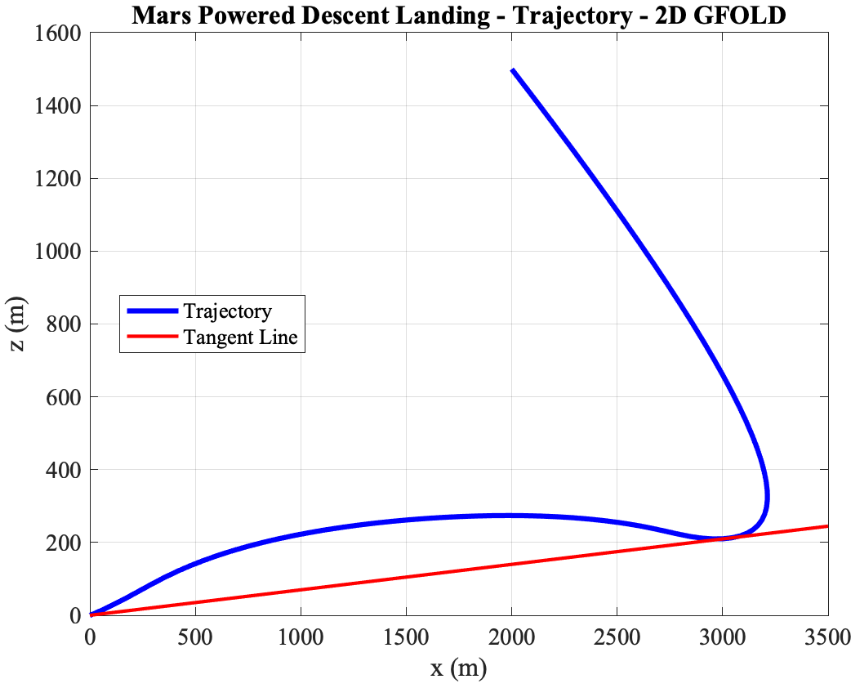


Figure E.3: Visualization of the glide-slope constraint, after Fig. 8 of Açıkmeşe and Ploen (2007).

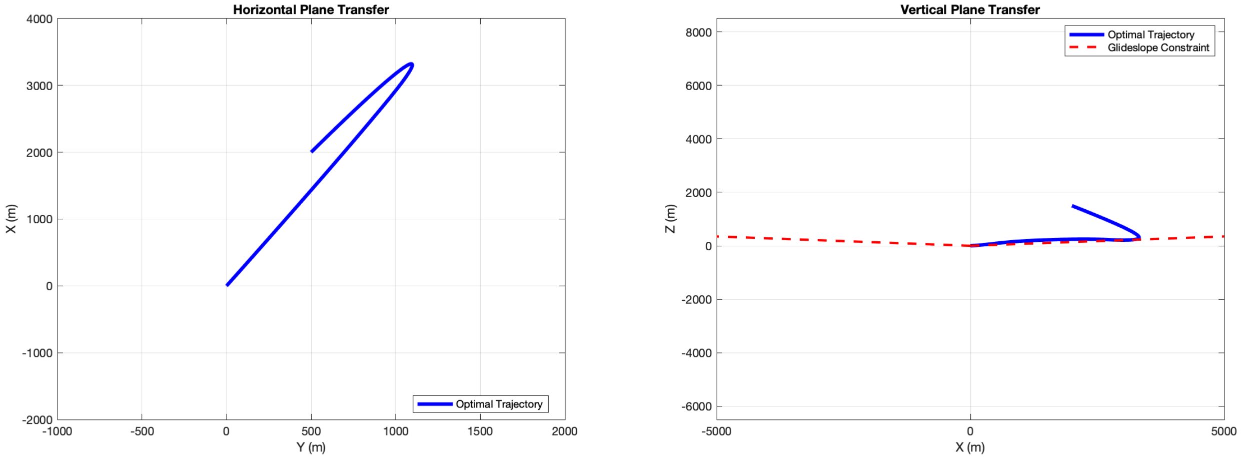


Figure E.4: Mars out-of-plane manoeuvre with the glide slope constraint enforced. Time history of the position, velocity states, acceleration, thrust angle, throttle setting and angle above surface. Figures is correspondence with Fig. 3 of Blackmore et al. (2010).

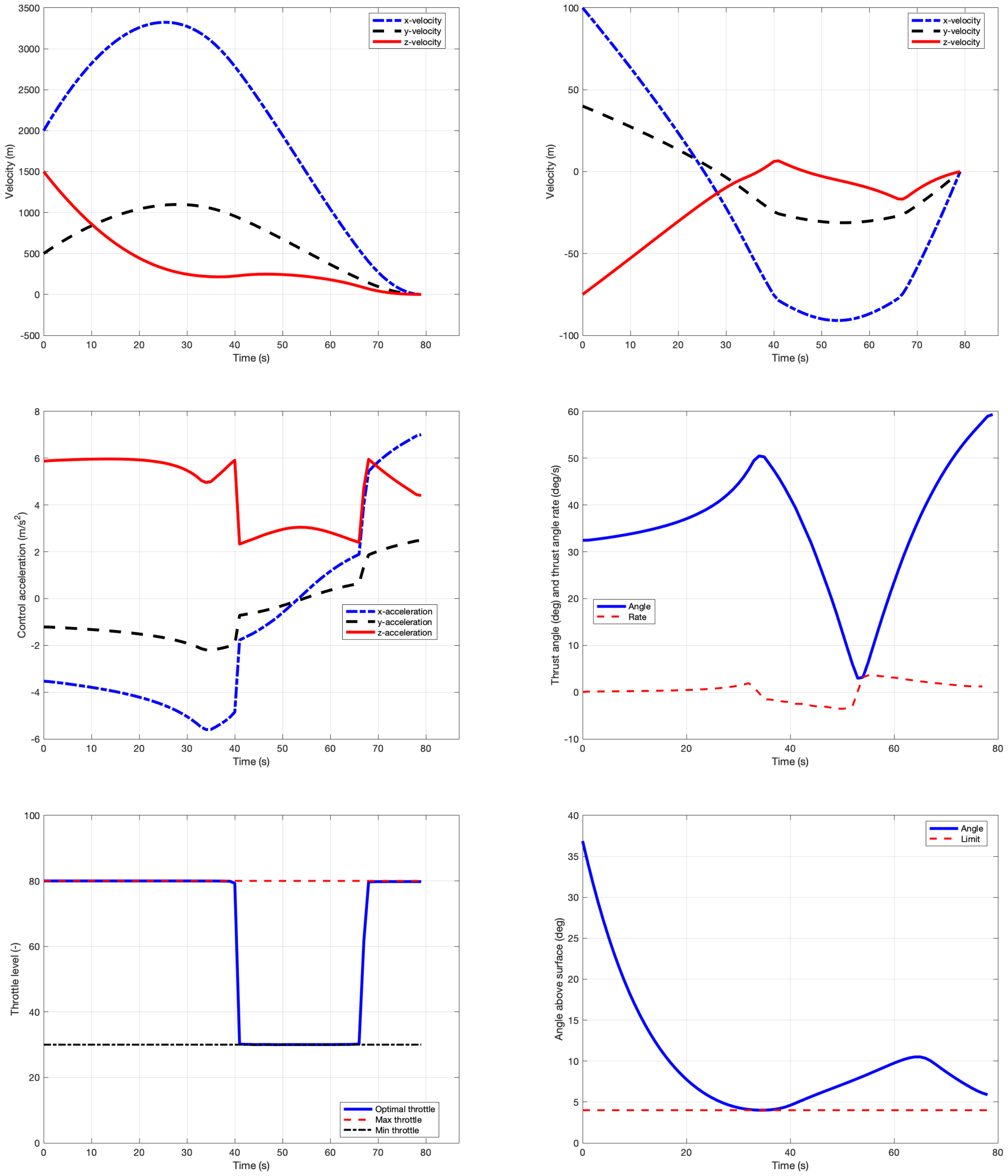


Figure E.5: Mars out-of-plane manoeuvre with the glide slope constraint enforced. Horizontal and vertical plane transfer. Figures is correspondence with Fig. 3 of Blackmore et al. (2010).

Bibliography

Açıkmeşe, B. "Convexification of Motion Planning and Control Problems". *University of Washington, Seattle: Air Force Office of Scientific Research Program Meeting*, September 2017.

Açıkmeşe, B. and Blackmore, L. "Lossless Convexification for a Class of Optimal Control Problems with Nonconvex Control Constraints". *Automatica*, Vol. 47, No. 2, pp. 341-347, February 2011.

Açıkmeşe, B. and Ploen, S. R. "Convex Programming Approach to Powered Descent Guidance for Mars Landing". *AIAA Journal of Guidance, Control and Dynamics*, Vol. 30, No. 5, pp. 1353–1366, September-October 2007.

Açıkmeşe, B., Blackmore, L., Scharf, D. P., and Wolf, A. "Enhancements on the Convex Programming Based Powered Descent Guidance Algorithm for Mars Landing". *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Honolulu, Hawaii, August 2008.

Açıkmeşe, B., Casoliva, J., Carson, J. M., and Blackmore, L. "G-FOLD: A Real-Time Implementable Fuel Optimal Large Divert Guidance Algorithm for Planetary Pinpoint Landing". *LPI Contribution, Concepts and Approaches for Mars Exploration*, Vol. 1679, p. 4193, Houston, Texas, June 2012.

Açıkmeşe, B., Carson, J. M., and Blackmore, L. "Lossless Convexification of Nonconvex Control Bound and Pointing Constraints of the Soft Landing Optimal Control Problem". *IEEE Transactions on Control Systems Technology*, Vol. 21, No. 6, November 2013.

Açıkmeşe, B., Sell, S., Martin, M. S., and Biesiadecki, J. "Mars Science Laboratory Flyaway Guidance, Navigation, and Control System Design". *Journal of Spacecraft and Rockets*, Vol. 51, No. 4, July 2014.

Adler, M., Wright, M., Campbell, C., Engelund, W., and Rivellini, T. "Entry, Descent, and Landing Roadmap - Technology Area 9". National Aeronautics and Space Administration, November 2010.

Atkinson, D., Kazeminejad, B., Lebreton, J., Witasse, O., Pérez-Ayúcar, M., and L.Matsond, D. "The Huygens Probe Descent Trajectory Working Group: Organizational Framework, Goals, and Implementation". *Planetary and Space Science*, Vol. 55, No. 13, pp. 1877-1885, November 2007.

Bedford, A. and Fowler, W. *Engineering Mechanics Dynamics*. Pearson Education South Asia Pte Ltd, 2008.

Blackmore, L. "Autonomous Precision Landing of Space Rockets". *Frontiers of Engineering: Reports on Leading-Edge Engineering from the 2016 Symposium*, pp. 33-42, Washington, DC, 2017.

Blackmore, L., Açıkmeşe, B., and Scharf, D. P. "Minimum-Landing-Error Powered-Descent Guidance for Mars Landing Using Convex Optimization". *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 4, pp. 1161-1171, July–August 2010.

Blau, P. "Preliminary MSL Entry, Descent & Landing Reconstruction". Spaceflight 101 - Space News and Beyond, August 2014.

Bose, D., Brown, J. L., Prabhu, D. K., Gnoffo, P., Johnston, C. O., and Hollis, B. "Uncertainty Assessment of Hypersonic Aerothermodynamics Prediction Capability". *Journal of Spacecraft and Rockets*, Vol. 50, No. 1, pp. 12-18, January-February 2013.

Boyd, S. and Vandenberghe, L. *Convex Optimization*. Cambridge University Press, 2004.

Braun, R. D. and Manning, R. M. "Mars Exploration Entry, Descent and Landing Challenges". *Journal of Spacecraft and Rockets*, Vol. 44, No. 2, pp. 310-323, March-April 2007.

Brugarolas, P., Casoliva, J., Johnson, A., Mohan, S., Chen, A., Stehura, A., Way, D., and Dutta, S. "On-board Terrain Relative Safe Target Selection For the Mars 2020 Mission". *13th International Planetary Probe Workshop*, Laurel, Maryland, June 2016.

Budney, C., Braun, B., May, L., Marais, D. D., Braun, R. D., Elkins-Tanton, L. T., and Lognonne, P. "Mars 2018 Sky Crane Capabilities Study". *The Planetary Science Decadal Survey Mars Panel*, March 2010.

Buonocore, M. "Guided Entry: a Necessary Step from Exomars to Precision Landing on Mars". *4th European Conference for Aerospace Science*, May 2011.

Carson, J. M., Açıkmeşe, B., and Blackmore, L. "Lossless Convexification of Powered-Descent Guidance with Non-Convex Thrust Bound and Pointing Constraints". *IEEE American Control Conference*, San Francisco, California, June-July 2011.

Chasnov, J. R. "Introduction to Numerical Methods". Hong Kong University of Science and Technology, 2012.

Choukroun, D. "Spacecraft Attitude Determination Lecture Notes". Delft University of Technology - Faculty of Aerospace Engineering, 2014.

Chu, Q. P. "Spacecraft Attitude Dynamics and Control - Course Notes". Delft University of Technology - Faculty of Aerospace Engineering, February 2017.

Dahleh, M., Dahleh, M. A., and Verghese, G. "Robust Stability in SISO Systems". Massachusetts Institute of Technology, 2011.

Dawson, M., Brewster, G., Conrad, C., Kilwine, M., Chenevert, B., and Morgan, O. "Monopropellant Hydrazine 700 lbf Throttling Terminal Descent Engine for Mars Science Laboratory". *43rd AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, Cincinnati, Ohio, July 2007.

D'Souza, A. F. *Design of Control Systems*. Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1988.

Edquist, K. T., Korzun, A. M., Dyakonov, A. A., Studak, J. W., Kipp, D. M., and Dupzyk, I. C. "Development of Supersonic Retro-Propulsion for Future Mars Entry, Descent, and Landing Systems". *Journal of Spacecraft and Rockets*, Vol. 51, No. 3, pp. 650-663, May-June 2014.

Euler, E. A., Adams, G. L., and Hopper, F. W. "Design and Reconstruction of the Viking Lander Descent Trajectories". *Astrodynamics Specialist Conference*, Jackson Hole, Wyoming, September 1977.

Evans, L. "Honors Differential Equations - Limitations of Euler's Method For Numerical Integration". Massachusetts Institute of Technology, 2009.

Field, R. V. "Stochastic Models: Theory and Simulation". Sandia National Laboratories, Albuquerque, New Mexico and Livermore, California, March 2008.

Fields, K. "Mass Property Measurements of the Mars Science Laboratory Rover". *27th Space Simulation Conference*, Annapolis, Maryland, November 2012.

Fosse, E., Harmon, C., Lefland, M., Castillo, R., and Devereaux, A. "Inheriting Curiosity: Leveraging MBSE to Build Mars 2020". *AIAA Space Conference and Exposition*, Pasadena, California, August-September 2015.

Furfaro, R., Wibben, D., Gaudet, B., and Simo, J. "Terminal Multiple Surface Sliding Guidance for Planetary Landing: Development, Tuning and Optimization via Reinforcement Learning". *Journal of the Astronautical Sciences*, Vol. 62, No. 1, pp. 73–99, July 2015.

Gerstenmaier, W. H. "NASA's Mars 2020 Project Final Report". National Aeronautics and Space Administration - Office of Inspector General, Janaury 2017.

Gerth, I. and Mooij, E. "Guidance for Autonomous Precision Landing on Atmosphereless Bodies". *AIAA Guidance, Navigation, and Control Conference*, National Harbor, Maryland, January 2014.

Gill, E. "Space Systems Engineering Lecture Notes". Delft University of Technology - Faculty of Aerospace Engineering, April 2016.

Gill, P. S. and Vaughan, W. W. "Technical Excellence: A Requirement for Good Engineering". *47th AIAA Aerospace Sciences Meeting*, Reno, Nevada, January 2008.

Gilmore, S. "An Introduction to Stochastic Simulation". University of Edinburgh, June 2006.

Guernsey, C. S. and Weiss, J. M. “Lessons Learned from the Development of the MSL Descent Stage Propulsion System”. *23rd AAS/AIAA Spaceflight Mechanics Meeting*, Kauai, Hawaii, February 2013.

Hartwig, J., Colozza, A., Lorenz, R., Oleson, S., Landis, G., Schmitz, P., Paul, M., and Walsh, J. “Exploring the depths of Kraken Mare – Power, thermal analysis, and ballast control for the Saturn Titan submarine”. *Cryogenic*, Vol 74, pp. 31-46, Marc 2016.

Holmberg, N. A., Faust, R. P., and Holt, H. M. “Viking ’75 Spacecraft Design and Test Summary Volume I - Lander Design”. NASA Scientific and Technical Information, pp. 63-64, November 1980.

Hosch, W. L., Shukla, G., and Young, G. “Reference Frame Physics”. *Encyclopædia Britannica*, April 2006.

Ingoldby, R. N. “Guidance and Control System Design of the Viking Planetary Lander”. *Journal of Guidance, Control, and Dynamics*, Vol. 1, No. 3, pp. 189-196, May-June 1978.

Jet Propulsion Laboratory. “Quaternions White Paper”. NASA Navigation and Ancillary Information Facility, November 2003.

Jiang, X., Wang, Y., and Li, S. “Next-generation Mars EDL GNC: Challenges and Solutions”. *IEEE Chinese Guidance, Navigation and Control Conference*, Yantai, China, August 2014.

Johnson, A. E. and Montgomery, J. F. “Overview of Terrain Relative Navigation Approaches for Precise Lunar Landing”. *IEEE Aerospace Conference*, Big Sky, Montana, March 2008.

Johnson, C. D. *Process Control Instrumentation Technology*. Pearson New International Edition, 2013.

Joshi, S., Kelkar, A. G., and Wen, J. T. Y. “Robust Attitude Stabilization of Spacecraft using Nonlinear Quaternion Feedback”. *IEEE Transactions on Automatic Control*, Vol. 40, No. 10, pp. 1800–1803, October 1995.

Karlgaard, C. D., Kutty, P., Schoenenberger, M., Schidner, J., and Munk, M. “Mars Entry Atmospheric Data System Trajectory Reconstruction Algorithms and Flight Results”. *51st AIAA Aerospace Sciences Meeting*, Grapevine (Dallas/Ft. Worth Region), Texas, January 2013.

Knocke, P. C., Wawrzyniak, G. G., Kennedy, B. M., Desai, P. N., Parker, T. J., Golombek, M. P., Duxbur, T. C., and Kass, D. M. “Mars Exploration Rovers Landing Dispersion Analysis”. *Astrodynamics Specialist Conference and Exhibit*, Providence, Rhode Island, August 2004.

Kowaleczko, G. “Apparent masses and inertia moments of the parafoil”. *Journal of Theoretical and Applied Mechanics*, Vol. 52, pp. 605-616, January 2014.

Krasnopolsky, V. and Lefèvre, F. “ Chemistry of the Atmospheres of Mars, Venus, and Titan.”. *Comparative Climatology of Terrestrial Planets*, pages 231–275, January 2013.

Kuo, B. C. *Automatic Control Systems*. Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1987.

Lee, U. and Mesbahi, M. “Constrained Autonomous Precision Landing via Dual Quaternions and Model Predictive Control”. *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 2, pp. 292-308, September 2017.

Lissaman, P. and Brown, G. “Apparent mass effects on parafoil dynamics”. *Aerospace Design Conference*, February 1993.

Lissauer, J. J. and de Pater, I. *Fundamental Planetary Sciences - Physics, Chemistry and Habitability*. Cambridge University Press, September 2013.

Lorenz, R., Newman, C., Tokano, T., Mitchell, J., Charnay, B., Lebonnois, S., and Achterberg, R. “Formulation of a wind specification for Titan late polar summer exploration”. *Planetary and Space Science*, Vol 70, No.1, pp. 73-83, June 2012.

Lu, P. “Predictor-Corrector Entry Guidance for Low-Lifting Vehicles”. *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 4, pp. 1067-1075, July-August 2008.

Martin, M. S., Mendeck, G. F., Brugarolas, P. B., Singh, G., Serricchio, F., Lee, S. W., Wong, E. C., and Essmiller, J. C. "In-flight Experience of the Mars Science Laboratory Guidance, Navigation, and Control system for Entry, Descent, and Landing". *9th International ESA Conference on Guidance, Navigation & Control Systems*, Porto, Portugal, June 2014.

Mastropietro, A. J., Kempenaar, J., Redmond, M., Pauken, M., and Ancarrow, W. "First Test Flight Thermal Performance of the Low Density Supersonic Decelerator (LDSD) Supersonic Flight Dynamics Test (SFDT) Vehicle". *45th International Conference on Environmental Systems*, Bellevue, Washington, July 2015.

Mendeck, G. F. and Craig, L. "Entry Guidance for the 2011 Mars Science Laboratory Mission". *AIAA Atmospheric Flight Mechanics Conference, Guidance, Navigation, and Control*, Portland, Oregon, August 2011.

Milne, W. E. "Note on the Runge-Kutta Method ". *Journal of Research of the National Bureau of Standards*, Vol. 44, May 1950.

Mooij, E. "The Motion of a Vehicle in a Planetary Atmosphere". Delft University of Technology - Faculty of Aerospace Engineering, January 1994.

Mooij, E. "Re-entry Systems Lecture Notes". Delft University of Technology - Faculty of Aerospace Engineering, September 2017.

Mooij, E. and Chu, Q. P. "Tightly-coupled IMU/GPS Re-entry Navigation System". *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Monterey, California, August 2002.

Morabito, D. D. "The Spacecraft Communications Blackout Problem Encountered during Passage or Entry of Planetary Atmospheres". *Interplanetary Network Progress Report*, August 2002.

Mulder, J. A., van Staveren, W. H. J. J., van der Vaart, J. C., de Weerdt, E., de Visser, C. C., in 't Veld, A. C., and Mooij, E. "Flight Dynamics Lecture Notes". Delft University of Technology - Faculty of Aerospace Engineering, March 2013.

Munk, M., Prince, J., Chandler, F., Campbell, C., Cheatwood, F. M., Moholt, M., Steltzner, A., Venkatapathy, E., and Wright, M. "Entry, Descent, and Landing Systems - TA 9: NASA Technology Roadmaps". National Aeronautics and Space Administration, July 2015.

National Research Council. *An Assessment of Balance in NASA's Science Programs*. The National Academies Press, Washington, DC, 2006. ISBN 978-0-309-10221-6. doi: 10.17226/11644. URL https://www.nap.edu/catalog/11644/an-assessment-of-balance-in-nasas-science-programs.

Nise, N. S. *Control Systems Engineering*. California State Polytechnic University *et* John Wiley & Sons Inc. , No. 6, April 2011.

Ogata, K. *Modern Control Engineering*, Fifth Edition. Prentice Hall, 2010.

Parissenti, G., Pessana, M., Gaia, E., Zaccagnino, E., Santilli, F., Pavarin, D., Bettella, A., Ronningen, J. E., Valle, T., Put, P. V., Tijsterman, R., Popola, R., Pistek, A., Sotto, E. D., Grandou, D., Stasse, V., Luca, L. T. D., Galfetti, L., and Maggi, F. "Throttleable Hybrid Engine for Planetary soft Landing ". *4th European Conference for Aerospace Sciences*, St. Petersburg, Russia, July 2011.

Peacocke, L., Perkinson, M. C., Reed, J., Lutz, T., Wolf, M., and Boltz, J. "Terminal Descent and Landing System Architectures for a Mars Precision Lander". *8th International Planetary Probe Workshop*, Portsmouth, Virginia, June 2011.

Petrick, D., Davis, B., Peregino, P., and Hufnal, P. "An Experimental Approach to Determine the Flight Dynamics of NASA's Mars Science Lab Capsule". Army Research Laboratory, Aberdeen Proving Ground, Maryland, January 2014.

Pocha, J. J. *An Introduction to Mission Design for Geostationary Satellites*. Kluwer Academic Publishers, 1987.

Pollard, B., Berkun, A., Tope, M., Andricos, C., Okonek, J., and Lou, Y. "Ka-band Radar Terminal Descent Sensor". Physical Sciences Tech Brief, Engineering Solutions for Design & Manufacturing, October 2007.

Prakash, R., Burkhart, P. D., Chen, A., Way, D. W., Guernsey, C., Kipp, D., Lorenzoni, L., Mendeck, G., Powell, R., Rivellini, T., Martin, M. S., Sell, S., and Steltzner, A. "Mars Science Laboratory Entry, Descent, and Landing System Overview". *IEEE Aerospace Conference*, Big Sky, Montana, May 2008.

Putnam, Z. R. and Braun, R. D. "Advances in Guidance, Navigation, and Control for Planetary Entry, Descent, and Landing Systems". *Advances in the Astronautical Sciences Guidance and Control Conference*, Breckenridge, Colorado, February 2016.

Quadrelli, M. B., Schutte, A., Rimani, J., and Ermolli, L. "Aero Maneuvering Dynamics and Control for Precision Landing on Titan". *IEEE Aerospace Conference*, Big Sky, Montana, April 2019.

Queen, E. M., Prince, J. L., and Desai, P. N. "Multi-Body Modeling and Simulation for the Mars Phoenix Lander Entry, Descent and Landing". *AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Guidance, Navigation, and Control*, Honolulu, Hawaii, August 2008.

Rapp, D. *Human Missions to Mars: Enabling Technologies for Exploring the Red Planet.* Springer International Publishing Switzerland, No. 2, 2016.

Razgus, B., Mooij, E., and Choukroun, D. "Relative Navigation in Asteroid Missions Using Dual Quaternion Filtering". *Journal of Guidance Control and Dynamics*, Vol. 40, No. 9, pp. 2151-2166, August 2017.

Rowell, D. "State-Space Representation of LTI Systems". Massachusetts Institute of Technology, October 2002.

Saraf, A., Leavitt, J. A., Mease, K. D., and Ferch, M. "Landing Footprint Computation for Entry Vehicles". *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Providence, Rhode Island, August 2004.

Schoenenberger, M., Kutty, P., Queen, E., and Karlgaard, C. "The Aerodynamics of Axisymmetric Blunt Bodies Flying at Angle of Attack". *IEEE Aerospace Conference*, Big Sky, Montana, March 2014.

Sengupta, A., Steltmer, A., Witkowski, A., Rowan, J., and Cruz, J. "Overview of the Mars Science Laboratory Parachute Decelerator Subsystem". *IEEE Aerospace Conference*, Big Sky, Montana, March 2007.

Skolnik, N., Li, L., Rossman, G., Kamezawa, H., Longo, A., Braun, R. D., and Sforzo, B. "Design of a Novel Hypersonic Inflatable Aerodynamic Decelerator for Mars Entry, Descent, and Landing". *AIAA Atmospheric Flight Mechanics Conference*, Grapevine, Texas, January 2017.

Soffen, G. A. and Snyder, C. W. "The First Viking Mission to Mars". *American Association for the Advancement of Science*, Vol. 193, No. 4255, pp. 759-766, August 1976.

Sostaric, R. "The Challenge of Mars Entry, Descent & Landing)". NASA Johnson Space Center, April 2010.

Szmuk, M. and Açıkmeşe, B. "Successive Convexification for 6-DoF Mars Rocket Powered Landing with Free-Final-Time". *AIAA Guidance, Navigation, and Control Conference*, Kissimmee, Florida, February 2018.

Szmuk, M., Açıkmeşe, B., Berning, A., and G.Huntington. "Successive Convexification for Fuel-Optimal Powered Landing with Aerodynamic Drag and Non-Convex Constraints". *AIAA Guidance, Navigation, and Control Conference*, San Diego, California, January 2016.

Thomson, B. J. and el Baz, F. "Future Mars Rovers: The Next Places to Direct Our Curiosity". Earth & Space Science News - Planetary Sciences, September 2014.

Uberhuber, C. W. *Numerical Computation: Methods, Software, and Analysis.* Springer, Berlin, Heidelberg, New York, Tokyo, 1997.

Vaughan, W. W., NASA Emeritus, and Marshall, G. C. "Systems Engineering: A Key for Success in Aerospace Operations". *Aeronautics and Aerospace Open Access Journal*, Vol. 2, No. 5, February 2018.

Wakker, K. F. "Fundamentals of Astrodynamics". Delft University of Technology - Faculty of Aerospace Engineering, January 2015.

Wang, T., Zhang, H. B., Zeng, L., and Tang, G. "A Robust Predictor–Corrector Entry Guidance". *Aerospace Science and Technology*, Vol. 66, pp. 103-111, July 2017.

Way, D. W., Davis, J. L., and Shidner, J. D. "Assessment of the Mars Science Laboratory Entry, Descent, and Landing Simulation". *23rd AAS/AIAA Space Flight Mechanics Meeting*, Vol. 148, pp. 563-581, Kauai, Hawaii, February 2013.

Way, D. W., Powell, R. W., Chen, A., Steltzner, A., Martin, M. S., Burkhart, P. D., and Mendeck, G. F. "Mars Science Laboratory: Entry, Descent, and Landing System Performance". *IEEE Aerospace Conference*, Big Sky, Montana, March 2007.

Wei, T., Xiaofeng, Y., Yewei, G., and Yanxia, D. "Hypersonic Flow Field Prediction and Aerodynamics Analysis for MSL Entry Capsule". *International Journal of Aerospace and Mechanical Engineering*, Vol. 9, No. 6, June 2015.

Whalley, R. "State-space Representations for Two-dimensional Control System Models". *Applied Mathematical Modelling*, Vol. 15, No. 2, 1991, pp. 90-97, February 1991.

Wie, B., Weiss, H., and Arapostathis, A. "A Quaternion Feedback Regulator for Spacecraft Eigenaxis Rotations". *Journal of Guidance Control and Dynamics*, Vol. 12, No. 3, pp. 375-380, May 1989.

Wilson, M. "Mars 2020 Mission Concept". Draft Environmental Impact Statement Scoping Public Meetings, October 2013.

Wingrove, R. C. "Survey of Atmosphere Re-entry Guidance and Control Methods". *AIAA Journal*, Vol. 1, No. 9, pp. 2019-2029, September 1963.

Wong, E., Singh, G., and Mascieralli, J. "Autonomous Guidance and Control Design for Hazard Avoidance and Safe Landing on Mars". *Journal of Spacecraft and Rockets*, Vol. 43, No. 2, pp. 378-384, March-April 2006.

Yu, Z., Xu, R., and Cui, P. "A Multi Sensor Based Integrated Navigation for Pin-Point Landing on Mars". *AIAA Guidance, Navigation, and Control Conference*, Kissimmee, Florida, July 2015.

Yu, Z. S., Cui, P., and Crassidis, J. L. "Design and Optimization of Navigation and Guidance Techniques for Mars Pinpoint Landing: Review and Prospect". *Progress in Aerospace Sciences*, Vol. 94, pp. 82-94, September 2017.

Zubrin, R. "Colonising the Red Planet: Humans to Mars in Our Time". *Architectural Design*, Vol. 86, No. 6, November 2014.