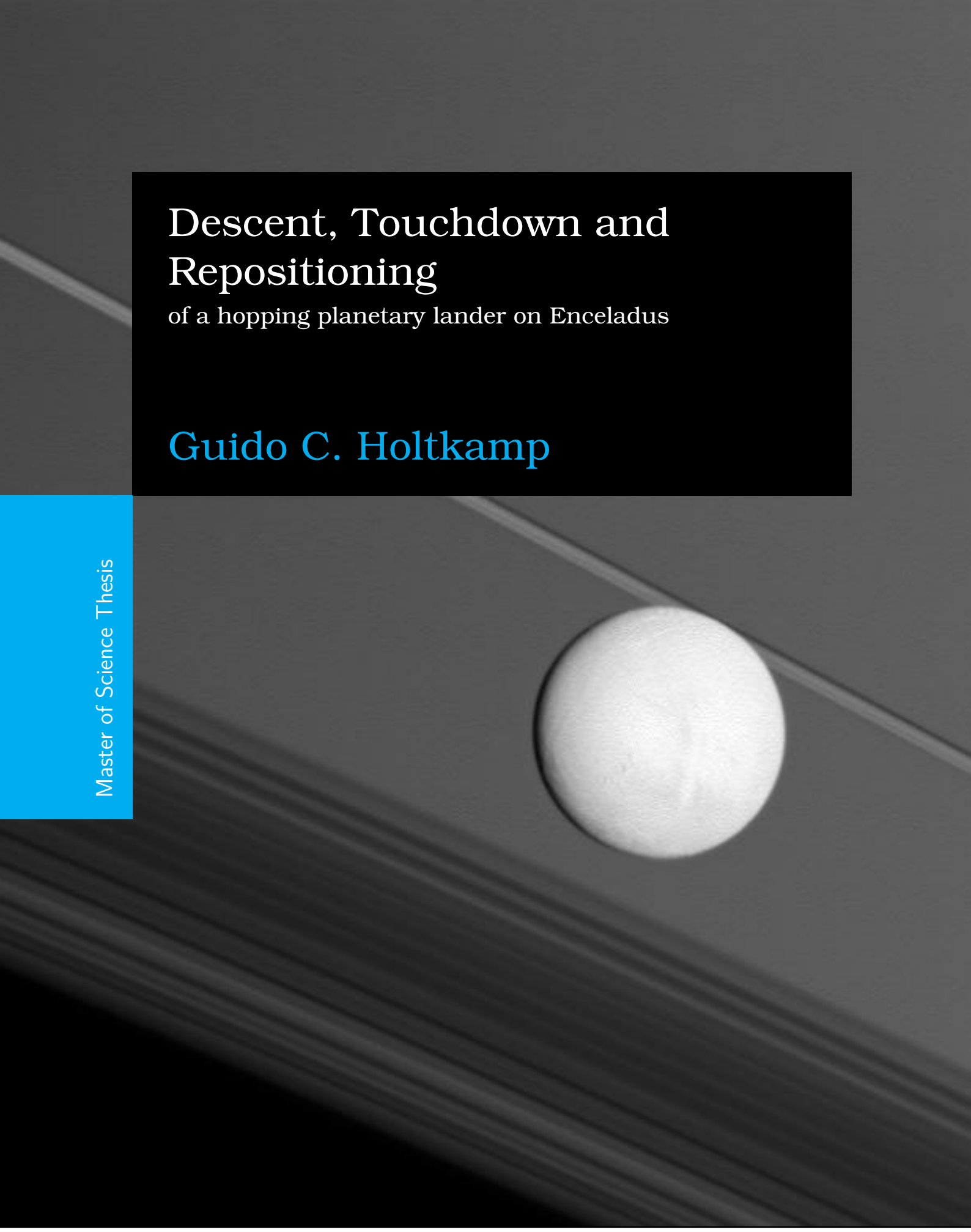
Descent, Touchdown and Repositioning

of a hopping planetary lander on Enceladus

Guido C. Holtkamp

Master of Science Thesis



TU Delft Delft University of Technology

Astrodynamics and Space Missions

Descent, Touchdown and Repositioning

of a hopping planetary lander on Enceladus

Master of Science Thesis

For the degree of Master of Science in Spaceflight at Delft University of Technology

Guido C. Holtkamp

Graduation Date: August 20, 2014

last document update (no changes in content): October 6, 2014

Faculty of Aerospace Engineering (AE) · Delft University of Technology

"Wenn wir den Mast mit dem Magneten nach vorne klappen, dann hängt er doch vor der Lokomotive und zieht sie an, und sie muß immer hinter ihm herfahren. Und in den Kurven legen wir den Mast einfach seitwärts." "Oh!" sagte Jim und bekam kugelrunde Augen vor Staunen, und dann sagte er nach einer Weile: "Donnerwetter!" und schließlich sagte er: "Ja wirklich!"
— *Jim Knopf und die Wilde 13* (Michael Ende)

Cover image: Enceladus in front of Saturn's rings. From photojournal.jpl.nasa.gov, PIA10485.

Preface

Since the spectacular finding of possible liquid water reservoirs on Enceladus by the Cassini spacecraft, scientists are very interested in exploring the Saturnian system in more detail. In particular, *in-situ* measurements over a long period of time would significantly contribute to our understanding of the moon's ecosystem. Both NASA and ESA currently investigate possibilities how such a mission could be realized.

As the final project of the B.Sc. curriculum, a group of students including the author of this report analyzed the feasibility of a planetary lander for surface investigation of Enceladus. It was concluded, that a hopping lander in combination with an orbiter for communication can answer the scientifically important questions, see *Mission to Enceladus for Terrain, Ocean and Plume Exploration* (Ampe et al., 2009). This thesis continues the work and focuses on the design of a comprehensible guidance, navigation and control system for both the descent and the repositioning process.

This thesis work would not have been possible without the help and support of many people.

I would like to express my very great appreciation to my supervisor Dr. E. Mooij, who enabled me to work on this very interesting topic. His expertise, his patience and his drive for perfection added greatly to this report, and it was always possible to knock on his door for questions. I am also grateful for the work and time he invested in co-authoring the paper *Guidance, Navigation and Control for Landing and Repositioning on Saturn's Moon Enceladus*, which is based on this report and will be presented at the AIAA/AAS Astrodynamics Specialist Conference in San Diego, California, in August 2014.

My thanks also go to Dr. P. Visser, Dr. D. Choukroun, ir. Svenja Woicke and Dr. E. Mooij, for taking the time to read my thesis and for being part of the graduation committee.

I greatly appreciate the company and assistance of my fellow students from the department of Astrodynamics and Space Missions. I enjoyed my graduation time, not least because of you.

Special thanks to my friends, who encouraged me and endured my bad moods, if I got stuck in a problem.

I would especially like to thank my family, in particular my parents, who always supported me, and without whom this work, and in fact everything else in my life, would have been impossible.

Abstract

This report deals with the design of a complete guidance, navigation and control system for a planetary lander on Enceladus, for both descent and repositioning. A basic lander model with a main engine, attitude thrusters and a landing gear is presented, for which the equations of motion in different reference frames are derived, including the effects of third body perturbations and non-homogeneous gravity fields. The guidance system incorporates a gravity-turn guidance logic for a flat and a spherical moon model for the initial descent phase, a quadratic guidance logic for both pinpoint landing and repositioning, a velocity nullifying logic for the terminal guidance phase and a ballistic guidance logic for a more fuel-efficient repositioning. The control system consists of a linear quaternion controller combined with a pulse-width-pulse-frequency modulator. The navigation system is build around an Inertial Measurement Unit with an extended Kalman filter to fuse the attitude and position measurements. The on-board software incorporates a basic hazard avoidance system, which uses a simulated LIDAR scan of the target area to generate a hazard map. The retargeting is based on a reachability and fuel consumption analysis. The spacecraft is capable of fully autonomous landings.

This guidance, navigation and control system is combined in a simulation software written in C++ to test the system performance in presence of various error sources. All program elements have been tested with results from literature.

The Monte Carlo simulations of the full GNC system include variances of the lander's state vector and of the GNC configuration parameters. The success rate is 90% for the descent phase, and 98% for the repositioning phase. The mean landing precision is in the order of 1 m, with velocity and pointing errors of 0.2 m/s and 1.4°, respectively. Stricter requirements on the lander's mass distribution, and the implementation of a feedback velocity nullifying guidance scheme can increase the mission success rate even further.

The thesis work shows, that a landing mission to Enceladus is possible with current technology. The Enceladus Lander Simulator is comprehensive software package for descent and repositioning simulations, and its modularity allows extensions in the future.

Table of Contents

List of Figures

List of Tables

Glossary

List of Acronyms

CIRS	Composite Infrared Spectrometer
EKF	Extended Kalman Filter
ESA	European Space Agency
G	Guidance
GC	Guidance and Control
GNC	Guidance, Navigation and Control
GPS	Global Positioning System
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
ISS	Imagine Science Subsystem
LEO	Low Earth Orbit
LIDAR	Light Radio Detecting and Ranging
METOPE	Mission to Enceladus for Terrain, Ocean and Plume Exploration
NASA	National Aeronautics and Space Administration
PD	Proportial-Derivative
PSD	Power Spectral Density
PWPF	Pulse-Width Pulse-Frequency
RK	Runge-Kutta
RKF	Runge-Kutta-Fehlberg

SHM	Surface Hazard Map
SSP	Sub-satellite point
VDFE	Vehicle Dispersion Footprint Ellipse
VIMS	Visible and Infrared Mapping Spectrometer

List of Symbols

Latin Symbols

A	system matrix	-
a	acceleration	$m \cdot s^{-2}$
B	angular momentum	$N \cdot m \cdot s$
b	bias	
C	rotation matrix (with subscripts 1, 2 or 3)	
e	error estimation	
F	force	N
F	system dynamics matrix	
g	gravitational acceleration	$m \cdot s^{-2}$
H	measurement matrix	-
I	inertial tensor	$kg \cdot m^2$
J	linear momentum	$N \cdot s$
K	Kalman gain matrix	-
M	rotation moment (torque)	$N \cdot m$
m	misalignment	-
P	error covariance matrix; point on sphere	
q	attitude quaternion	-
R	measurement noise covariance matrix	
r	position w.r.t. inertial reference frame	m
s	scale factor error	
T	transfer matrix	
u	control input	
V	velocity	$m \cdot s^{-1}$
v	measurement noise	
w	system noise	
x	true state	
z	measurement	
A	assembly (reference frame)	-
B	body-fixed (reference frame)	-

f	force	N
g	gravitational acceleration	$\mathrm{m{\cdot}s^{-2}}$
I	inertial (reference frame)	-
i	orbital inclination	rad
m	mass	kg
N	north pole	-
n	total number of bodies (in a many-body problem)	-
R	rotating (reference frame)	-
r	radius	m
S	surface-fixed (reference frame)	-
s	path length of descent trajectory	m
t,T	time	s
V	vertical (reference frame)	-
v	velocity	$\mathrm{m{\cdot}s^{-1}}$

Greek Symbols

χ	heading angle	rad
δ	latitude	rad
δ_d	downrange	m
γ	flight path angle	rad
μ	gravitational parameter of central body	$\mathrm{m^3{\cdot}s^{-2}}$
ω	angular velocity	$\mathrm{rad{\cdot}s^{-1}}$
Φ	system transition matrix	
ϕ	roll angle	rad
ψ	yaw angle	rad
ρ	great circle segment between two points	rad
σ	standard deviation	
τ	longitude	rad
τ_d	cross range	m
θ	pitch angle; true anomaly	rad
ξ	angle between δ_d and ρ	rad
ζ	angle between meridian of $\mathbf{P}_0$ and ρ	rad

Subscripts

$_{1,2,3}$	rotation about x-, y-, z-axis
$_{\omega}$	rotational rate
$_{as}$	altitude switch (with σ)
$_{a}$	accelerometer
$_{B}$	w.r.t. body reference frame
$_{cb}$	central body
$_{cmd}$	commanded

$_{com}$	center of mass
$_{c}$	circular, Coriolis
$_{d}$	disturbing; downrange or cross range
$_{enc}$	Enceladus
$_{e}$	estimated (with σ)
$_{f}$	final; fuel
$_{I}$	w.r.t. inertial reference frame
$_{i}$	index of body under investigation (in a many-body problem)
$_{j}$	index of perturbing body (in a many-body problem)
$_{k}$	index of main body (in a many-body problem)
$_{k}$	iteration index (Kalman filter)
$_{max}$	maximum
$_{min}$	minimum
$_{moi}$	moment of inertia
$_{m}$	main; measurement; moon, mass (with σ)
$_{p}$	pericenter
$_{q}$	quaternion
$_{rel}$	relative
$_{R}$	w.r.t. rotating reference frame
$_{r}$	radial; with σ: position
$_{sat}$	Saturn
$_{s}$	specific
$_{t2g}$	time-to-go
$_{ta}$	thrust alignment (with σ)
$_{tm}$	thrust magnitude (with σ)
$_{T}$	w.r.t. thrust reference frame
$_{t}$	transfer; target
$_{V}$	w.r.t. vertical reference frame
$_{v}$	velocity (with σ)
$_{x}$	x-axis
$_{y}$	y-axis
$_{z}$	z-axis

Superscripts

$^{+}$	updated (corrected)
$^{-}$	time-propagated (predicted)

Combining Characters

$\hat{\mathbf{x}}$	estimated
$\tilde{\mathbf{x}}$	resultant

Chapter 1

Introduction

Voyager 1 was the first space probe that performed a flyby at Enceladus. The images taken in 1980 revealed a smooth and very reflective surface. The absence of impact craters or other topographic features proved that the moon's surface is very young and undergoes a - probably still active - resurfacing process. This discovery came as a big surprise for the scientific community, as no one expected volcanic activity on a small and icy moon (Squyres et al., 1983). Additional measurements of Voyager 1 confirmed previous Earth-based observations, which indicated that the maximum density of Saturn's E-Ring almost coincidences with the orbit of Enceladus, which in turn suggests that Enceladus constantly replenishes the unstable ring (Terrile and Cook, 1981). Several theories existed to explain the mass ejection of the moon (Terrile and Cook, 1981): due to tidal heating induced by a resonance with Dione, Enceladus might have a liquid subsurface ocean, which reaches the surface through cracks caused by tidal stresses or by meteoroid impacts. In case the moon has a solid interior, the mass ejection can be caused by volatiles pockets inside the crust that occasionally burst due to excess pressure.

About one year after Voyager 1, Voyager 2 passed Enceladus at a lower altitude, enabling the Imaging Science System (ISS) to take pictures with a higher resolution. With the help of these images, it was possible to determine five different topographic regions. The youngest surface areas are only a few hundred million years old, again indicating a still ongoing resurfacing process, which additionally is supported by the discovery of characteristic tectonic faults (Kargel and Pozio, 1996). According to Stone and Miner (1982), radioactive decay cannot provide a sufficient amount of energy for the tectonic activity - tidal heating is a more likely source of energy, but the orbital eccentricity of Enceladus might be too low. Finding the answer to this question was a major scientific goal of the Cassini-Huygens mission.

The double space probe Cassini-Huygens, designed specifically for the investigation of the Saturnian system, was launched in 1997 and performed several close flybys of Enceladus in 2005. Analysis of the magnetometer data indicated the presence of a thin, spatially nonhomogeneous atmosphere concentrated at the lower latitudes. To investigate this anomaly further, the subsequent flybys focused the moon's south polar region. Images taken with the ISS of Cassini showed plumes emanating from this area, ejecting material to altitudes of 400 km and

higher. Further measurements indicated that these geysers have time-dependent characteristics and originate from the so-called *Tiger Stripes*, four parallel, linear ravines, about 500 m deep and 2 km wide (Porco et al., 2006). A large amount of the ejected plume material does not leave Enceladus' sphere of influence and falls back to the surface, which probably is the reason for the very smooth surface of the south polar region. Results from the Composite Infrared Spectrometer (CIRS) also indicated that the surface at the bottom of the Tiger Stripes reaches much higher temperatures than the surface further away from the ravines - up to 145 K, versus a normal temperature of 52 K (Spencer et al., 2006). The mechanisms behind Enceladus' cryovolcanism are currently the subject of scientific investigations, but the verification of one theory requires additional data from instruments and probes, which are not part of the payload of Cassini.

The discoveries of Cassini are even more spectacular due to the possible existence of a habitual zone underneath the surface. Ecosystems as we know them require liquid water, nutrients and source of energy (McKay et al., 2008). It is proven that the latter two conditions are fulfilled, as Enceladus' hot spots receive energy from an undetermined source and eject water, nitrogen compounds and carbon compounds. In case a future mission confirms the existence of a subsurface ocean, Enceladus would be the only known celestial body next to Europa which might posses the key ingredients of life - with the important advantage, that the moderate environmental conditions on Enceladus (especially the lower radiation levels) simplifies in-situ measurements. According to McKay et al. (2008), the habitability of Enceladus can be better estimated, if geological and chemical mechanisms behind the cryovolcanism and their temporal variabilities are known. A geochemical cycle on Enceladus, in which crust material is detached by a subsurface ocean and then expelled via the geysers resurfacing the crust, has a positive effect on the habitability of the subsurface ocean (Parkinson et al., 2008), but the existence of such a cycle is only predicted and must be verified by additional measurements. These measurements should also determine the amount and composition of the E-ring material that Enceladus recollects in its orbit, as the plume material will partially undergo chemical reactions and thus be an additional source of new material (Parkinson et al., 2008).

A lander mission to Enceladus will be able to fulfill the main scientific goals being the investigation of the south-polar plumes and the subsurface ocean, the characterization of the surface and interior, and the analysis of the moon's biologic potential. The thesis work focuses on answering the following question:

> Is it possible with current guidance, navigation and control technology to safely land and reposition a hopping lander on Enceladus?

To answer this question, a flight simulator is developed, that incorporates the complete guidance, navigation and control system of a basic lander model. Each chapter of this report focuses on a single aspect of the *Enceladus Lander Simulator*.

Chapter 2 introduces the Saturnian system and the surface and landscape of Enceladus. This is necessary to derive the requirements for the landing accuracy and to generate a realistic surface hazard map, which will be used as input for the hazard avoidance system.

Chapter 4 discusses all tools necessary to describe the motion of the lander during the mission. This includes the definition of reference frames and coordinate systems, the integration processes, the perturbing forces and the full equations of motion.

Chapter 5 presents all elements of the guidance, navigation and control subsystems, and the principle of autonomous hazard avoidance. Each subsystem is tested and configured separately, and the simulation results are shown at the end of each section.

Chapter 7 gives a concise overview of the Enceladus Lander Simulator setup. It discusses the principles of Monte Carlo simulations, the program structure, the user interface and the method of testing of each program element.

Chapter 8 discusses the simulation results for both the descent and the repositioning phase. The Monte Carlo simulations include variances of the lander's state vector and of the GNC configuration parameters to simulate a imperfect system, and to allow a sensitivity analysis, which is necessary to identify the parameters with the most influence on the mission success.

Chapter 9 summarizes the most important conclusions that can be drawn from this thesis work. Furthermore, it discusses the modifications that can be carried out to improve the Enceladus Lander Simulator, and mentions the areas for possible future work.

Chapter 2

Environment of Enceladus

Section 2-1 deals with the influence of the Saturnian system, such as magnetic and gravitational disturbances and dust particles, the in-orbit spacecraft and the lander during its mission phases.

Data about topography and chemical composition of Enceladus' surface is essential for a adapting the touchdown and repositioning system to the target environment: the landing gear must be able to cope with the elevation, roughness, flexibility and adhesive characteristics of the moon's surface, while the guidance, navigation and control GNC system ensures that the spacecraft is within its flight envelope. This information is collected in Section 2-2.

2-1 The Saturnian System

Enceladus is one of Saturn's 53 (named) moons, and one of the few moons with a sufficient mass to form an ellipsoid shape. Figure 2-1 gives an overview of the Saturnian System. All moons exert gravitational forces on each other to some degree; these so-called *third body perturbations* are discussed in Section 4-7-1.

Enceladus moves through the densest part of Saturn's E-ring, mainly due to the fact that the south-polar geysers are a source for the E-ring particles (see Fig. 2-2). The particle density, however, is very low: Cassini's Cosmic Dust Analyzer (CDA) with a sensor area of 50 cm^2 and a sensitivity for particles larger than 2 μm registered a peak count rate of 4 particles per second (Spahn et al., 2006). Consequently, the drag induced by the E-ring particles will only have an extremely small effect on the motion of the spacecraft and the lander.

Saturn's orbital semi-major axis is 9.537 AU, which results in an average solar flux in the Saturnian system of about 15 Wm^{-2}, which is 10% of the solar flux at Earth. Perturbations due to solar-radiation pressure are for satellites orbiting Earth only a minor disturbance - the effects on a spacecraft orbiting Enceladus are negligible, as demonstrated in Section 4-7-2.

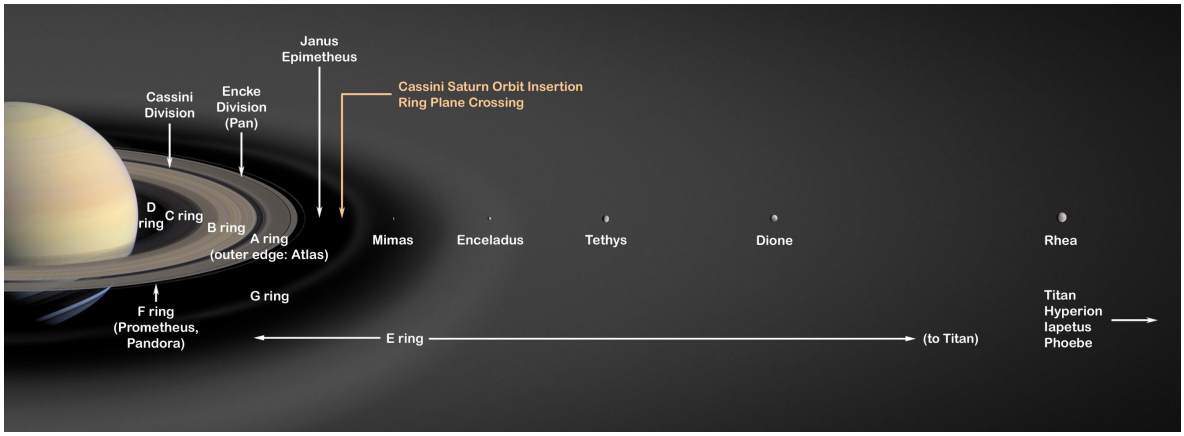


Figure 2-1: Saturn's rings and its larger moons (NASA, 2012b)

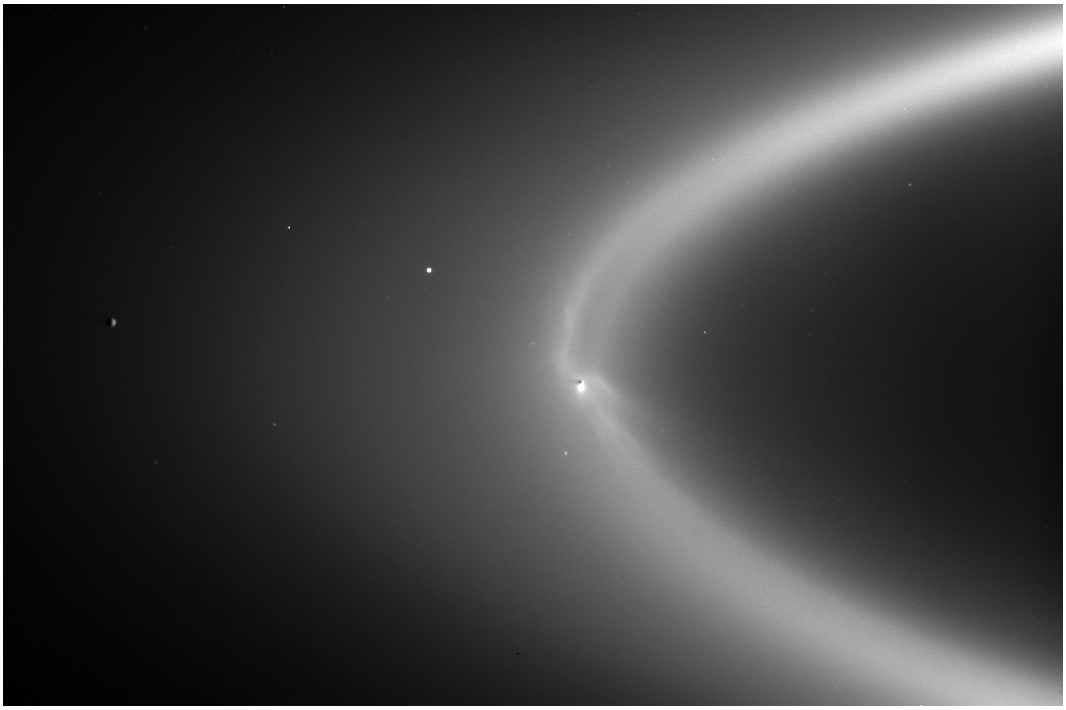


Figure 2-2: Light scattered by the E-ring particles and the plume material ejected by Enceladus (ESA, 2012)

2-2 Surface and Landscape

Enceladus was discovered by William Herschel in 1789 as the sixth largest moon of Saturn. As a consequence of its surface H_2O ice, the moon has a spherical albedo - the ratio of the total reflected light and the incident light - of 0.99, making it the most reflective celestial object in the Solar System. Enceladus follows an almost elliptical orbit through the densest part of Saturn's E-ring, with a semi-major axis of 238,037 km, an orbital eccentricity of 0.0047 and an inclination of 0.009°, which results in an orbital period of 1.37 days (Spencer et al., 2006). As many other moons that move close to their central objects, Enceladus' rotational period is equal to its orbital period due to tidal locking ($\omega = 5.30773 \cdot 10^{-5}$ rad/s). The general characteristics of Enceladus are collected in Table 2-1.

Enceladus shape is an almost perfect ellipsoid; the maximum deviation is below 2 km (Porco et al., 2006). In a standard XYZ Cartesian coordinate system, the ellipsoid equation is given as

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1 \tag{2-1}$$

where x, y and z mark the locations of a surface point and a, b and c are the values of the semi-axes of the ellipsoid. By means of limb measurements with Cassini's ISS, the values of the semi-axes were found to be $a = 256.6 \pm 0.6$ km, $b = 251.4 \pm 0.2$ km and $c = 248.3 \pm 0.2$ km (Thomas et al., 2007). The ellipsoid data often reveal information about the internal structure of a planet or moon, but at the moment it is not possible to determine with absolute certainty whether Enceladus has a differentiated or a homogeneous interior (see Porco et al. (2006) and Thomas et al. (2007)).

Characteristic	Value	Source
distance from Saturn	3.95 R_{sat}	Coustenis et al. (2009)
orbital period	1.370218 days	NASA Fact Sheet[1]
rotational period	synchronous rot.	NASA Fact Sheet
orbital inclination (to equator Sat.)	0.009°	NASA Fact Sheet
orbital eccentricity	0.0045°	Coustenis et al. (2009)
radius	252.1 km	Spencer et al. (2006)
density	1,610 kgm^{-3}	Roatsch et al. (2009)
mass	10.8 x 10^{19} kg	Jacobson et al. (2006)
gravitational parameter μ (=GM)	7.2096 km^3s^{-2}	Jacobson et al. (2006)
gravitational acceleration	0.12 ms^{-2}	Coustenis et al. (2009)
spherical albedo	0.99	NASA Fact Sheet
escape velocity	0.235 ms^{-1}	Coustenis et al. (2009)
surface temperature	114-157 K	Coustenis et al. (2009)

[1] `http://nssdc.gsfc.nasa.gov/planetary/factsheet/saturniansatfact.html` (Oct. 2011)

Table 2-1: General characteristics of Enceladus

Enceladus has several topographic regions (see Fig. 2-3), indicating a complicated geological history. The northern hemisphere of the moon is the oldest terrain, and is heavily cratered.

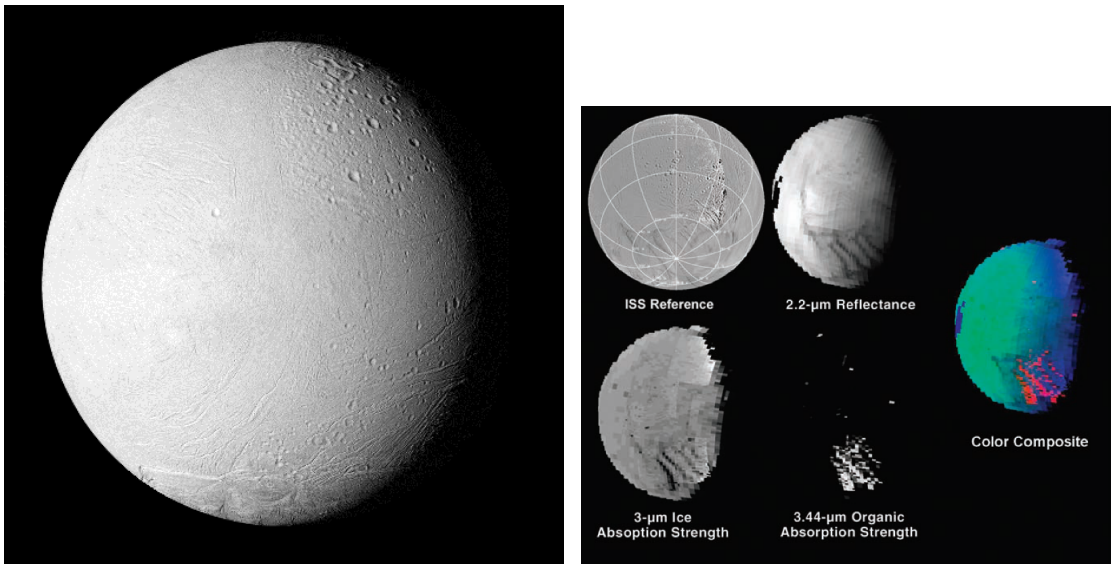


Figure 2-3: Left: High-resolution ISS picture of Enceladus. The different tectonic regions are clearly visible, see text for discussion (NASA, 2012b). **Right:** Absorption maps of Enceladus generated with Cassini VIMS data. There are strong 3-μm ice and 3.44-μm organic absorptions marking the Tiger Stripes (Brown, Clark, et al., 2006)

Some craters at the lower latitudes were flattened over time due to mass flows in that region. This *viscous relaxation* is an important proof for the geological activity of Enceladus and was the focus of many studies before Cassini discovered the geysers in the south polar region (Porco et al., 2006). The remaining surface shows no signs of impact craters and is geological young. These relatively smooth terrains, however, are finely fractured (Porco et al., 2006) as a result of internal activity. Figure 2-5 shows the surface map of Enceladus, including the names of the most prominent features. The topography variations are clearly visible. The landing site is chosen to be near the Tiger Stripe *Baghdad* (see target definition in Section 8-2).

Near-infrared spectrometric investigation of Enceladus by means of Earth-based telescopes showed that Enceladus' surface consists of partially crystalline H_2O ice, which explains the high albedo (Brown, Clark, et al., 2006). Based on long-term observations, there are indications that the reflectance and thus structure of the ice changes over time. With the Visible and Infrared Mapping Spectrometer (VIMS) of Cassini it was possible to investigate the composition and properties of the surface, especially in the geological active area of the south polar region. The H_2O ice grain size near the Tiger Stripes is 0.1 to 0.3 mm and 0.05 to 0.15 mm for the rest of the planet (Brown, Clark, et al., 2006; Jaumann et al., 2008). Consequently, the surface consists of an ice dust layer rather than a massive ice shell. This so-called *regolith* is caused by the accumulation of fine material over time, probably due to dispersed material from impacts and the recollection of E-ring material (Porco et al., 2006). Strong absorption lines indicate that there is a high abundance of CO_2 near and in the Tiger Stripes, which in turn suggests that there is a constant replenishment. The CO_2, however, exists in a pure form only in the colder, northern regions of the moon; the CO_2 in the south-polar region is complexed, probably with the H_2O ice (Brown, Clark, et al., 2006). A south polar atmospheric column with a base area of 2 cm^2 contains about 10^{10} CO molecules (Brown, Clark, et al., 2006). Simple organic material can be found near the Tiger Stripes, see Fig. 2-3.

The existence of ammonia (NH_3) on Enceladus is the basis of several studies about the mechanisms behind the south polar cryovolcanism. The strongest absorption lines of NH_3 almost coincide with the strongest absorption lines of H_2O, which complicates a direct detection. Due to the absence of distinct geological features, there is no direct proof of the existence of NH_3 on the moon, but it is possible to define the upper concentration limits based on the grain size of the H_2O ice. The most likely upper global concentration of NH_3 on the surface is about 2% (Brown, Clark, et al., 2006). Spots of pure NH_3 are also possible as a consequence of the resolution limit of the VIMS.

Note, that this paragraph is largely based on Porco et al. (2006). The south-polar region of Enceladus has unique tectonic characteristics, covering about 70,000 km^2 or 9% of the total surface. Its surface is geologically young and shows a higher reflectance then the surrounding terrain. Images of Cassini's ISS taken during close approaches with a resolution of up to 4 m/pixel showed, that the south-polar region is cluttered with house-sized ice chunks and fine fault lines, but almost completely free of impact craters (see Fig. 2-6). As a consequence, the region is relatively hilly, and the distance between the elevations is between 20 and 100 m. The Tiger Stripes are four large, roughly parallel, linear depressions with an average depth of 500 m, a width of 2 km, a length of 130 km and a spacing of about 35 km. On both sides, they are flanked by ridges with heights of about 100 m and widths of a few kilometers. There are indications that there are ice blocks at the bottom of the Tiger Stripes, which originate from the sides of the fracture. Unlike the rest of Enceladus' surface, the Tiger Stripes are not covered with fine H_2O ice grain. It is assumed that these geological features formed too recently for the development of regolith, or the H_2O ice grains were changed (fully or partially) by the temperature anomalies of the region and thus cannot be directly distinguished with the a spectrographic analysis of the VIMS data. There are no impact craters within several kilometers from the Tiger Stripes, and the few existing impact craters in the south polar region have a diameter of less than 1 km. This suggests a surface age of 500,000 years or younger, compared with an age of 4 billion years of the highly-cratered terrains in the northern hemisphere.

As mentioned before, the surface level of Enceladus does not deviate more than 2 km from the mean ellipsoid. The areas of the south polar region featuring tectonic faults are hundreds of meters higher than the adjacent terrains. Figure 2-4 shows the variations in surface height with respect to the mean ellipsoid for a given latitude.

The south-polar geysers seem to emanate from the Tiger Stripes. These particle jets are ejected at several locations in many directions, all supplying a large plume, which reaches an altitude of 435 km and more. Based on models, the mean vertical velocity of the plume particles is 60 ms^{-1} (Porco et al., 2006), which is considerably lower than the local escape velocity of 235 ms^{-1} (see Table 2-1). Consequently, only a small fraction of the total mass flow can actually escape from the gravitational attraction of Enceladus and feed Saturn's E-ring, estimates are around 1%. The absolute values for the total mass flow and the escape rate are uncertain, estimates vary between 4 - 150 kgs^{-1} and 0.04 - 1 kgs^{-1} (Porco et al., 2006; Hansen et al., 2006). In turn, this mass flow is only a fraction of the calculated mass flow required to replenish the E-ring. Cassini's measurement results also showed temporal variations of the E-ring's physical properties, which indicates that the geyser characteristics change over time.

The surface temperature depends on the location. The north pole is not illuminated by the

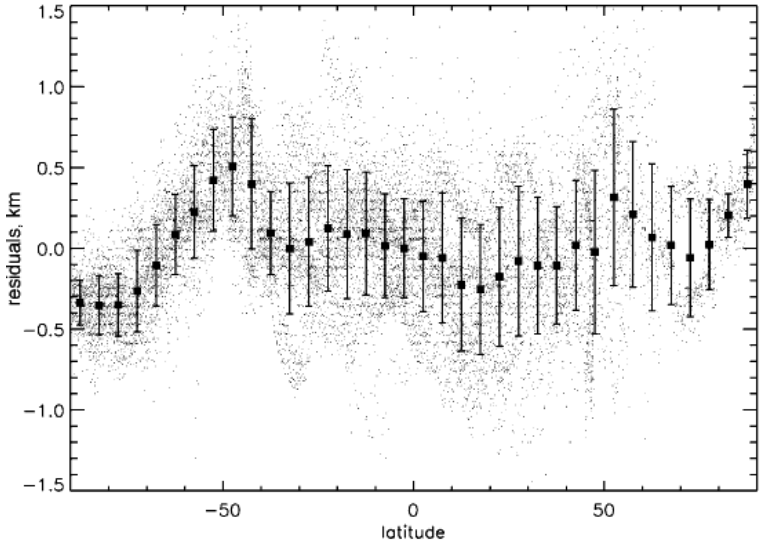


Figure 2-4: Averaged limb deviations from the mean ellipsoid of Enceladus. The dots indicate a measurement result from any longitude value, the error bars show the standard deviation. All values lie within a $\pm$1.5 km deviation (Thomas et al., 2007)

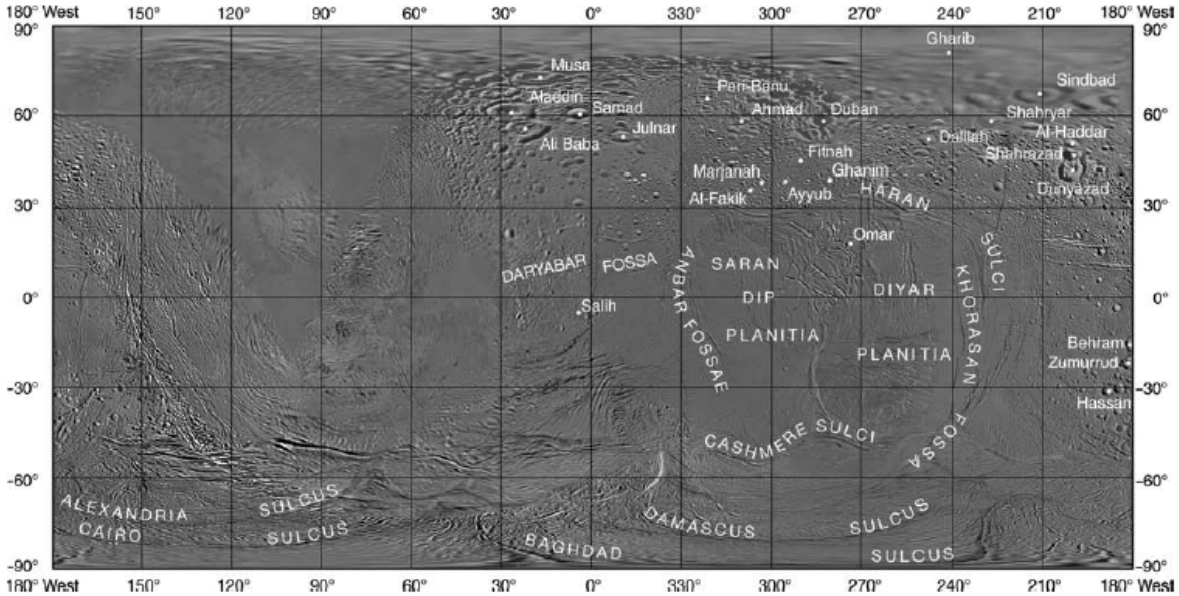


Figure 2-5: ISS map from Enceladus with longitudes and latitudes (Roatsch et al., 2009)

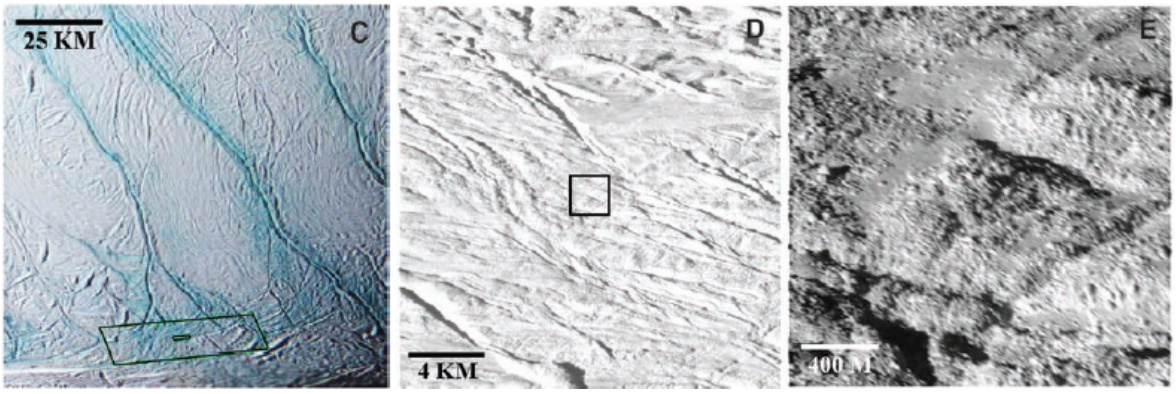


Figure 2-6: Left: False color image (70 m/pixel) of the Tiger Stripes (blue). **Mid:** 37 m/pixel detail of the marked area in the left picture. **Right:** 4 m/pixel detail of the marked area in the left and mid picture. This is the image with the highest spatial resolution so far from the surface of Enceladus. The blur is due to the spacecraft motion. Images taken from Porco et al. (2006)

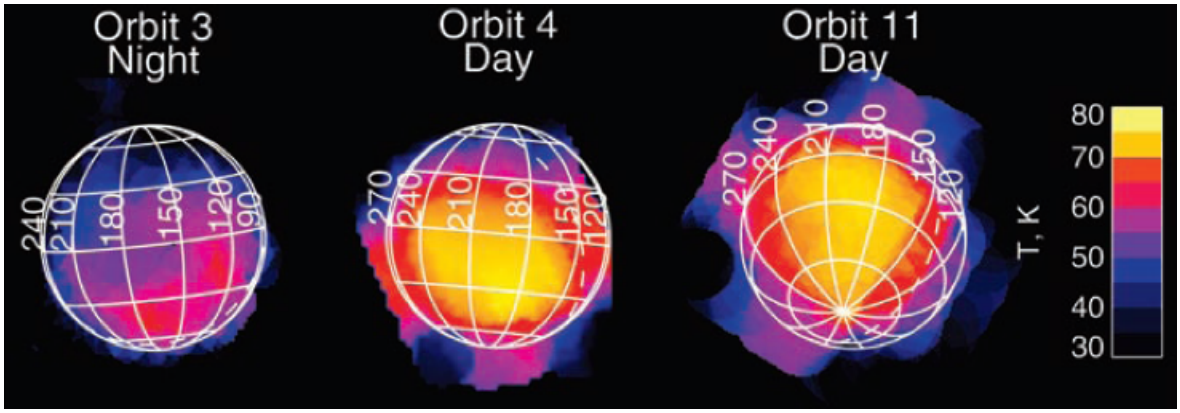


Figure 2-7: Far-IR temperature pictures of Enceladus during three encounters with Cassini (Spencer et al., 2006)

Sun since 1995 and consequently is particularly cold. Calculations suggest an upper limit of about 40 K (Spencer et al., 2006). The field of view size of Cassini's Composite Infrared Spectrometer (CIRS) varies between 6 km and 18 km, which requires the implementation of models to determine the surface temperature for smaller areas. The average temperature increases in southern direction to about 60 K, see Fig. 2-7. According to Spencer et al. (2006), the temperature near the Tiger Stripes are around 145±14 K, with a very low possibility of temperatures below 120 K - even temperatures of 180 K and more can exist in narrow areas. This would be near the 173 K limit of a H_2O/NH_3 melt, which is the basis of some studies about the cryovolcanism of Enceladus.

Chapter 3

Lander Characteristics

This chapter gives an overview of the lander's mass model (Section 3-1), the propulsion system (Section 3-2), and the landing gear (Section 3-3). The data originates from the preliminary subsystem design described in the final METOPE[1] report (Ampe et al., 2009), and as such should only be considered as a guideline and a starting point for further investigations. This chapter closes with a list of simulator configuration parameters related to the lander characteristics discussed in Sections 3-3 to 3-3.

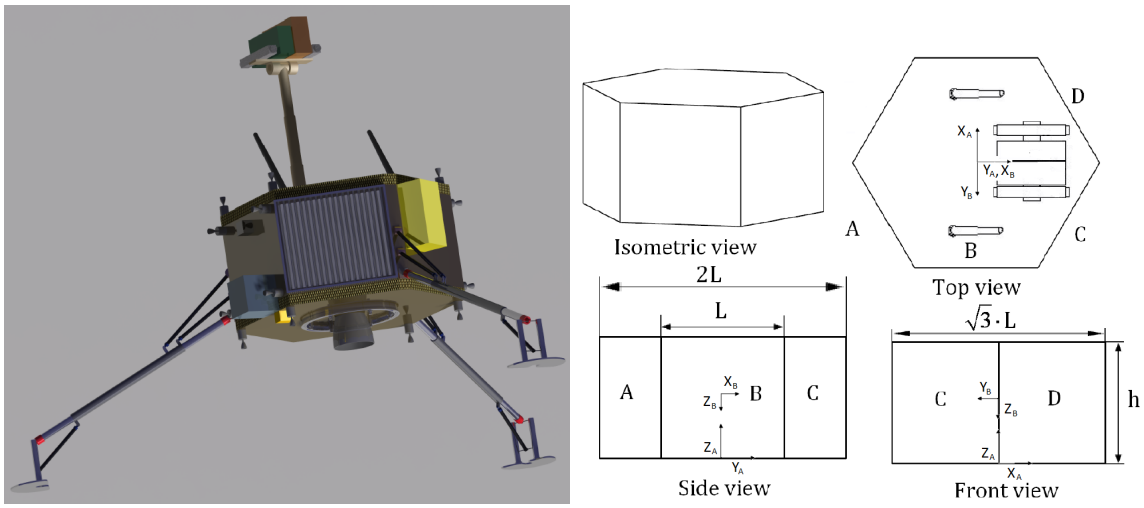
The following sections consist of summaries of the most important data. For a more detailed description of the lander and the design process, the reader is referred to Ampe et al. (2009).

The planetary lander, named *Silenus*, includes only those elements that are attached to the vehicle during a repositioning cycle on Enceladus. Therefore, any orbit injection modules or additional protective elements for the first landing are not considered part of the lander. The mission design includes an orbiter for communication relaying and orbit insertion. The estimated total mission cost the lander segment is 800 million Euros.

3-1 General Characteristics

The body of the planetary lander (see Fig. 3-1a) has a basic hexagonal prism shape, which is beneficial for the attachment of external elements such as the three-legged landing gear, the attitude thrusters and some of the instruments. The concept of an edged spacecraft body has been used in existing missions in low-gravity environments, for example, in the MINERVA robot as part of the Hayabusa mission, and in Philae as part of the Rosetta mission.
A special assembly reference frame, or A-frame, is used to define any location on or within Silenus' structure. The position and the orientation of the A-frame does not change during the design phases, unlike the body reference frame, which originates in the vehicle's center

[1]The preliminary lander design is the result of a group project of 10 students in the final phase of their B.Sc. curriculum at Delft University of Technology. METOPE stands for *Mission to Enceladus for Terrain, Ocean and Plume Exploration*



(a) Isometric view **(b)** Body dimensions

Figure 3-1: Isometric view of Silenus in the deployed configuration and definition of the body dimensions with respect to the assembly frame (Ampe et al., 2009)

of mass. As a first approximation, the body reference frame will originate in the geometrical center of the spacecraft. The origin of the A-frame is fixed in middle of the lander's bottom plane, see Fig. 3-1b. According to the orientation shown in that figure, the body reference frame is shifted along the assembly reference frame's $+Z$-axis by half the lander height h.

The preliminary transfer matrix between these two reference frames may be inferred from the preliminary location of the c.o.m. (see Table 3-1) and the definition of the A-frame as shown in Fig. 3-1b.

Characteristic	Value	Comment
mass at entry	335 kg	incl. 20 kg contingency
mass after touchdown	300 kg	mass for first hop
fuel mass repositioning	20 kg	–
location c.o.m.	0.0, 0.0, 325 mm	x-,y-,z-coord. A-frame
mass M.o.I. (homogen.)	150.0, 150.0, 150.0 kgm^2	I_{xx}, I_{yy}, I_{zz}, B-frame, at entry
diameter	1600 mm	Maximal (2L, Fig. 3-1b)
height	650 mm	h in Fig. 3-1b
face dimensions	800 mm × 650 mm	L × h in Fig. 3-1b

Table 3-1: Idealized general characteristics of Silenus (Ampe et al., 2009)

A position given by the Cartesian coordinates $[x_A, y_A, z_A]$ with respect to the A-frame can be translated into coordinates with respect to the body reference frame by first subtracting the position of the vehicle c.o.m. from the z_A-value, and then reorienting the $X_A-/Y_A-/Z_A-$axis to match the $X_B-/Y_B-/Z_B-$axis (see also Section 4-1-5 for the exact definition of the body

reference frame):

$$\begin{pmatrix} x_B \\ y_B \\ z_B \end{pmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{pmatrix} x_A \\ y_A \\ z_A - 325 \text{ mm} \end{pmatrix} \tag{3-1}$$

Table 3-1 lists the most important characteristics of Silenus for the design of the GNC system. The listed values are idealized and correspond to the nominal lander model.

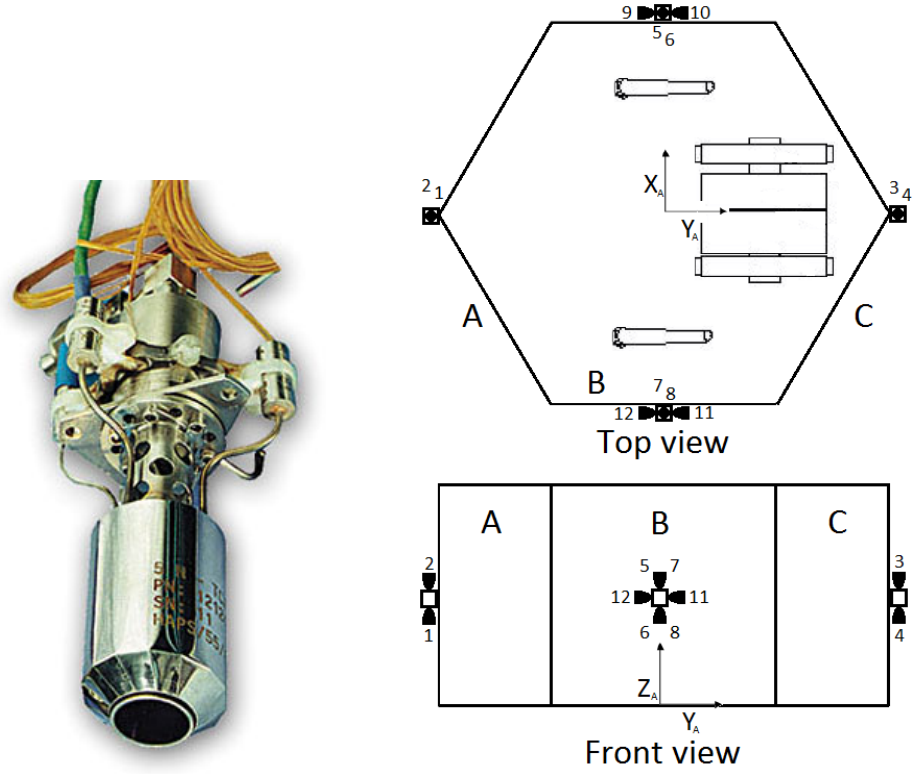
3-2 Propulsion System

The design of the Silenus lander assumes the use of an orbit insertion propulsion module, which will bring the lander in a circular, equatorial orbit around Enceladus at an altitude of 100 km and then separate from the lander. The main thruster of the lander is responsible for the inclination changes and the initiation of the entry and descent, and delivers the main propulsive force necessary for the repositioning on the moon's surface. The design process showed that a thruster comparable to the Marquardt R-4D meets all requirements. This thruster with a weight of 3.76 kg is placed in the middle of the bottom plate of vehicle body (origin of the assembly reference frame) for a good alignment with the center of mass - ideally, the main thruster induces no disturbing moments. The engine uses a mixture of N_2O_4 and MMH as propellant, stored in four spherical tanks, which are positioned symmetrically around the Z_A-axis for stability purposes. The orbital maneuvers and descent process require roughly 54 kg of fuel, another 30 kg are reserved for the repositioning. The main thruster data can be found in Table 3-2 below. In the simulations, it will be assumed that the thruster is freely throttleable between 10% and 100% of the maximum thrust.

Characteristic	Value	Comment
I_{sp}	312 s	vacuum
T_{vac}	490 N	rated thrust
mass flow	158 gs^{-1}	at rated thrust
min. impulse bit	15.6 Ns	-
max. pulses	20,781	-
throttle range	10-100%	assumed
location	0,0,0 mm	x_A-, y_A-, z_A-position (ideally)

Table 3-2: Main thruster characteristics and location (Ampe et al., 2009)

The attitude control of the lander during all mission phases after the separation from the orbit insertion propulsion module is realized with 12 CHT-5 attitude thrusters, see Fig. 3-2a. Manufactured by EADS Astrium, these hydrazine thrusters with a thrust range of 1.85 to 6.0 N have successfully been used on many spacecrafts and seem to be a good choice for lander. Attitude verniers, however, generally have no thrust range and always provide their maximum thrust when activated due to mechanical reasons (see Section 5-2-4). For a better validity, it will be assumed that the verniers always fire with their maximum thrust level. Two thrusters at a time can induce a rotation about one of the principal axes of the B-frame,



(a) CHT-5 thruster **(b)** Locations

Figure 3-2: Location of the opposing CHT-5 attitude thruster pairs in the A-frame. See Table 3-3 for the exact numbers.

so the control of the vehicle can be realized despite the possible failure of one thruster per rotation direction. The thrusters are positioned in opposing pairs in the plane at half the body height. Figure 3-2b indicates with the numbers 1 to 12 the position of each vernier with respect to the A-frame. The lander provides four thruster anchors; the anchors for the rotations about the Y_B- and Z_B-axis are used by four verniers each, the anchors for the rotation about the X_B-axis by two verniers each. The distance between the lander's center mass and the latter anchor set is slightly larger (15%) than the other anchor set, so the pitch will be better controllable then roll and yaw. The characteristics and the exact location of each thruster are listed in Table 3-3. The torque levels produced by this vernier configuration are discussed in the scope of the thruster selection logic, see Section 5-2-3.

The positioning of the verniers in the same plane and their alignment with the $Y_B - Z_B$-axis significantly simplifies the attitude control system design. However, the thruster configuration shown in Fig. 3-2b does not allow for a translational force along the X_A-axis. This is a problem for the accelerometer calibration process, and possibly for the near-target navigation in case the lander cannot be rotated anymore to use the main engine for position correction. In the simulator, it will be assumed that the lander has four additional verniers connected to the two thruster anchors for the pitch control; two pointing in the $+X_B$-direction, and two in the $-X_B$-direction. Figure 3-2b thus only represents the thruster configuration for attitude control. In reality, the thrusters are not always aligned with the body axes, and control

rotations about more than one axis to reduce the system weight and increase the reliability. The determination of the exact location and orientation of each thruster may be a topic for future work.

Characteristic	Value	Comment
I_{sp}	220 s	vacuum
T_{vac}	1.85 - 6.0 N	thrust range, T_{max} used
mass flow	2.8 gs^{-1}	at max. thrust
min. impulse bit	0.1 - 0.3 Ns	-
max. pulses	44,000	12.5 hr accumulated burn time
anchor roll & yaw	$\pm\frac{\sqrt{3}}{2}L$, 0, $\frac{h}{2}$	x_A-, y_A-, z_A-position (ideally)
anchor pitch	0,$\pm L$, $\frac{h}{2}$	see Fig. 3-1b for variables

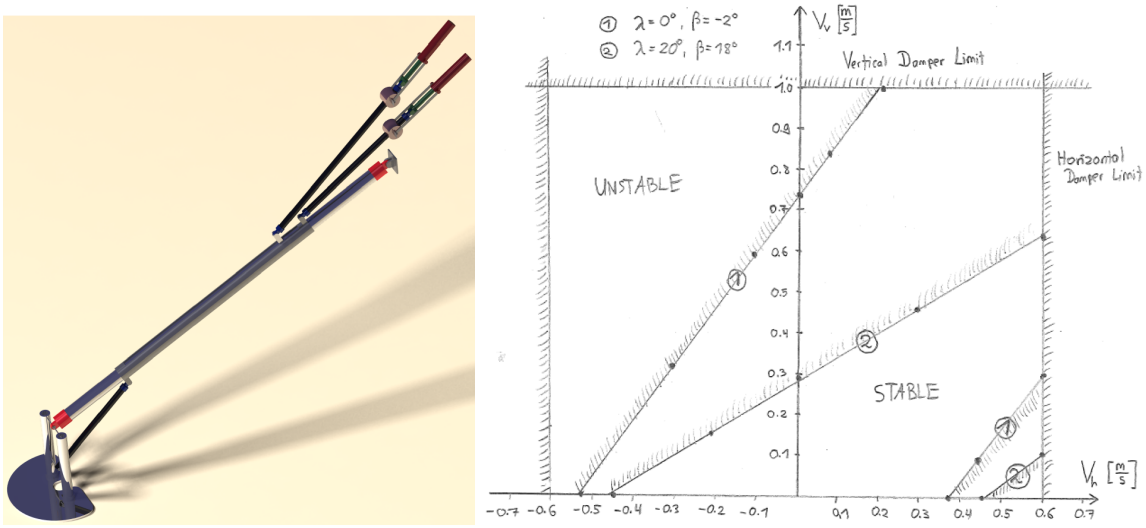
Table 3-3: Attitude control thruster characteristics and location (Ampe et al., 2009)

3-3 Landing Gear

The landing gear must prevent tumbling during the touchdown after the descent or at the end of each repositioning cycle. The topography, the surface character and the very low gravitational attraction of Enceladus complicate a safe touchdown. The design of the Silenus landing gear is based on the three-legged Philae lander. The placement of the main thruster in the middle of the body, however, prevents the use of a coherent system, thus the landing gear consists of three independent legs (see Fig. 3-3a). Calculations have shown that, in theory, the gravitational attraction of Enceladus is sufficiently large to prevent a turnover, only by the horizontal and vertical distance from the c.o.m. of the lander (Ampe et al., 2009). In this way, a safe touchdown is possible without making use of screws and an anchoring system such as used on Philae. Each leg has its own damping system, consisting of two dampers and placed inside the lander for thermal reasons: One damper is directly connected to the main bar of the leg as is mainly responsible for the damping of the velocity normal to the ground. The other damper is connected to a slider, a freely movable element encasing the main bar, which in turn is connected to the foot bar. This shock absorber damps the velocities parallel to the ground. The slider relays the forces parallel to the ground, keep the foot bar in vertical position (which is important for the ground clearance during the touchdown) and allows a space-saving folding of the landing gear. Springs inside the hinges, a restraining system and thermal knives allow a the GNC system to command the deployment of the landing gear before the touchdown.

Characteristic	Value	Comment
Energy dissipation vertical	150 J	max., per stroke
Energy dissipation horizontal	54 J	max., per stroke
Max. stroke length vertical	60 mm	piston movement
Max. stroke length horizontal	70 mm	piston movement
Angle vertical line - leg	65° - 81°	deployed config.

Table 3-4: Landing gear characteristics (Ampe et al., 2009)



(a) Isometric view **(b)** Touchdown velocity limitations

Figure 3-3: The landing gear is designed to allow a safe touchdown for a certain velocity range (Ampe et al., 2009). See text for discussion.

Basic calculations show that the touchdown stability depends on the vertical and horizontal touchdown velocity V_v and V_h as well as on the local surface inclination angle λ and the angle β between the lander's Z-axis and local vertical line (pointing accuracy). A very basic velocity window for two different combinations of λ and β is shown in Fig. 3-3b. Both the horizontal and the vertical dampers have energy dissipation limits (see Table 3-4), which set absolute velocity boundaries, but a too large (or too small) combination of λ/β will still cause a turnover. The vertical and horizontal velocities must be below 0.6 ms^{-1} for all cases. Table 3-4 and Fig. 3-3b summarize the important characteristics of the landing gear.

Currently, the design does not include any means to change the lander's attitude after touchdown. This might have some negative effects on repositioning process in case the surface is inclined away from the target area. Additionally, communication and navigation systems must be able to cope with this condition. Some changes in the design might allow the lander to change its stationary attitude.

3-4 Simulator Configuration: Lander Characteristics

The program parameters related to the lander characteristics discussed in this chapter are listed in Table 3-5. The actual parameter values can be found in Appendix A-1 and A-2. Note, that variable names are only listed once in the table if only their axis designation changes – in those cases, an indication of the number of excluded variable names is added in brackets. The '+2' after `xCOMshift`, for example, refers to `yCOMshift` and `zCOMshift`.

The nominal lander's center of mass is located in the geometrical center of the spacecraft body. It is assumed, that the mass is homogeneously distributed in the shape of a cube, which leads to a diagonal inertia tensor.

Variable Name	Comment
Mass Related Parameters	
`landerMass`	[kg]
`inertiaTensorXX` (+2)	[kgm^2], I_{xx}, see Table 3-1
`inertiaTensorXY` (+2)	[kgm^2], $I_{xy} = I_{yx}$
`xCOMshift` (+2)	[m], Δx_{com}, see Eq. (7-2)
Thruster Data	
`minimumThrustMainEngine`	[N]
`maximumThrustMainEngine`	[N]
`specificImpulseMainEngine`	[s], I_{sp}
`thrustForce`	on/off, test mode
`xPositionMainEngine` (+2)	[m], in B-frame
`eulerOrientationXmain` (+2)	[rad], ZYX-rotation w.r.t. B-frame
`minimumThrustVerniers`	[N]
`maximumThrustVerniers`	[N]
`specificImpulseVerniers`	[s], I_{sp}
`xPositionVernier1` (+11)	[m], in B-frame
`eulerOrientationX1` (+11)	[rad], ZYX-rotation w.r.t. B-frame

Table 3-5: Program parameters related the lander characteristics. Number of similar variables indicated in brackets.

Chapter 4

Spacecraft Motion Simulation

This chapter discusses the concepts and mathematical relations that are necessary to simulate a spacecraft model with six degrees of freedom.

The first step consists of the exact definition of all involved reference frames (Section 4-1). The equations of translational and rotational motion for a lander model with respect to a rotating central body are derived in Section 4-4.

4-1 Reference Frames

The choice of a certain reference frame depends on the use: Newton's laws of motion, for example, are only valid with respect to an inertial frame of reference, but a vertical frame is a more straightforward choice for local navigation purposes. The following subsections describe the concepts of the inertial planetocentric frame, the rotating planetocentric frame, the vertical frame, the body frame and the instrument frame. The transformation matrices between these reference frames can be found in Section 4-3.

4-1-1 Inertial Planetocentric Reference Frame

An inertial reference frame is a reference frame, which itself does not experience an acceleration or rotation with respect to the universe. The origin of a inertial planetocentric frame coincides with the central body's center of mass (Mooij, 1997). Strictly speaking, an inertial planetocentric frame is a *pseudo-inertial* reference frame due to the accelerations the central body experiences during its rotation around its respective main body (*e.g.*, the Sun) and due to the rotation of the Galaxy. For navigational purposes, however, the effects are negligible and the inertial planetocentric reference frame may be considered inertial (Groves, 2008).

The inertial planetocentric reference frame is marked with the index I. The Z_I-axis is pointing north along the body's axis of rotation, and its direction is assumed constant over time. The surface spanned by the X_I- and Y_I-axis coincides with the equatorial plane. The longitude

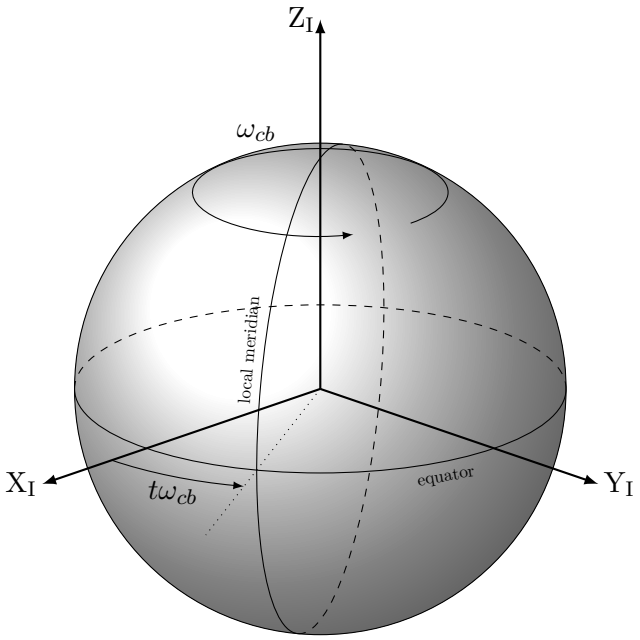


Figure 4-1: Definition of the inertial planetocentric reference frame

of the reference meridian defining the direction of X_I-axis is set to 0° at a given instance of time $t = t_0$. The I-frame does not rotate with the central body, in contrast to the rotating planetocentric reference frame discussed in the next section. A sketch of the I-frame is given in Fig. 4-1. Note, that the Y_I-axis is located 90° advanced of X_I-axis (in positive rotation direction), completing the right handed frame.

4-1-2 Rotating Planetocentric Reference Frame

The rotating planetocentric reference frame, or R-frame, is denoted with the index R and coincides with the inertial planetocentric frame (described in the previous section) at the instance of time $t = t_0$. The Z_R-axis is pointing north along the body's axis of rotation. The longitude of the reference meridian defining the direction of X_R-axis is fixed at 0°; the direction of the Y_R-axis can be found using the right-hand rule. A sketch of the R-frame is given in Fig. 4-2. The angular velocity of the R-frame with respect to the I-frame is the angular velocity ω_{cb} of the central body.

4-1-3 Vertical Reference Frame

The vertical reference frame, denoted with the index V, is a common near- or on-surface navigation frame. The origin of the V-frame is the central point of the vehicle used for navigational purposes (Groves, 2008), often the center of mass or a convenient geometric location on the spacecraft. The Z_V-axis points in the direction of the center of mass of the central body, thus coinciding with the radial component of the gravitational attraction. The X_V-axis lies in a meridian plane, is normal to the Z_V-axis and points north. The Y_V-axis points east and in this way completes the right-hand system. A sketch of the vertical frame

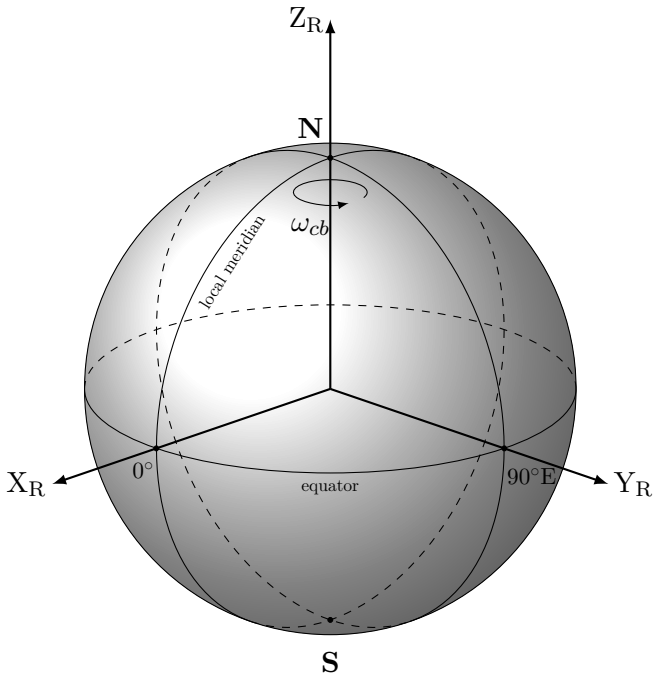


Figure 4-2: Definition of the rotating planetocentric reference frame

is given in Fig. 4-3. In case the central body is a perfect sphere, the $X_V Y_V$-plane is the *local horizontal plane*. The vertical reference frame should not be used with care at the poles of the central body, as the X_V- and Y_V-axis are not defined at these points (Groves, 2008).

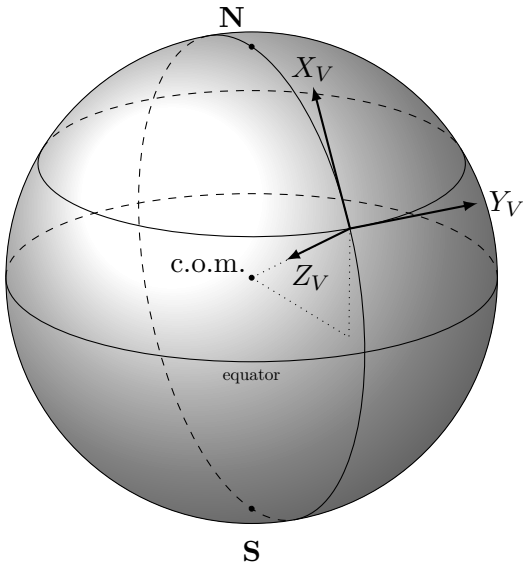


Figure 4-3: Definition of the vertical reference frame

4-1-4 Surface-fixed Reference Frame

The gravity turn guidance and the quadratic guidance laws require a reference frame that – once initialized – does not move with respect to the surface and offers the possibility do divide the current velocity into vertical and horizontal components for an easier analysis of the descent trajectory. The origin of the surface-fixed reference frame, or S-frame, is located on the surface, at the intersection of the spacecraft's position vector with the moon's surface. The X_S-axis points towards the lander, in opposite nadir direction. The Y_S-axis coincides with the projection of the X_B-axis on the local horizontal plane. The Z_S also lies within the horizontal plane and completes the right-handed system. In this way, the X_S-, Y_S- and Z_S-axis represent an upward, forward and sideward motion in case the Z_B-axis (see Section 4-1-5) points in the direction of nadir. A sketch of the S-frame is shown in Fig. 4-4.

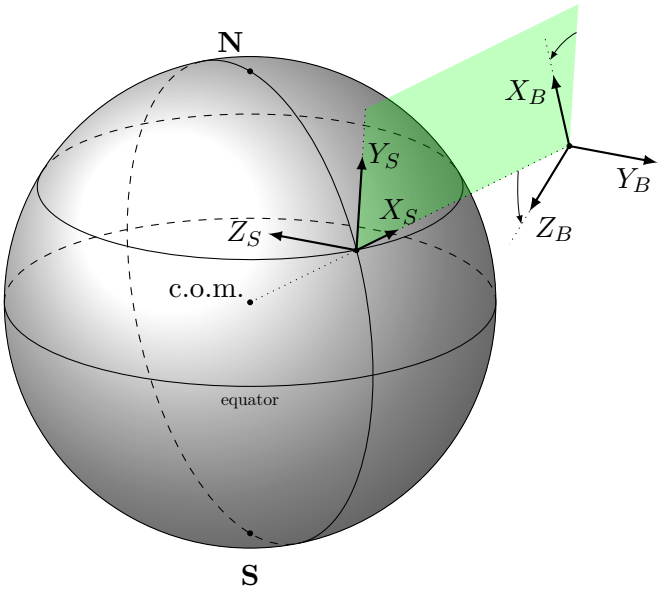


Figure 4-4: Definition of the surface-fixed reference frame. The X_S-,Y_S- and X_B-axis are in are in the same plane (green) at the initialization of the S-frame. The X_S-axis points vertically towards the spacecraft, so that the central body's c.o.m., the origin of the I-frame and the origin of the B-frame are on the same line.

The S-frame is fixed at a given instance of time, which can lead to problems after some time, because the local horizontal plane ignores the moon's spherical shape. The S-frame may be updated during the switch from one guidance law to another, or the it is simply accepted that the surface level has an increasing negative X_B-value when the spacecraft is moving away horizontally from the origin. The easiest way to define the orientation of the S-frame at a given instance of time is to use the current vertical frame and align first Y_V-axis by rotating about the Z_V-axis, and then rotating the new frame about the Y_V'-axis by $\frac{\pi}{2}$ to let the X_V''-axis point in opposite nadir direction. The azimuth angle χ_V follows from expressing the X_B-axis with respect to the V-frame and then using an adapted version of Eq. (4-6c) to determine the azimuth angle of the projected X_B-axis. The exact expression for χ_V is given Section 4-3-2.

Note, that it is not possible to define S-frame in a situation where the lander's velocity vector

is collinear with the nadir vector, but a pure vertical motion can be described nevertheless, as long as the S-frame has been defined at a previous instance of time where at least one horizontal velocity component was not zero.

4-1-5 Body-Fixed Reference Frame

The body-fixed reference frame is denoted with the index B and is used to describe the position and orientation of the planetary lander with respect to another reference frame. The origin of the body-frame frame coincides with the one of the vertical reference frame described in the precious subsection, but the orientation of the axes does not change with respect to the vehicle: the X_B-axis points in the forward direction, the Z_B-axis in the downward direction, and the direction of the Y_Z-axis can be found with the right-hand rule.

This reference frame is also used for aircraft, and accordingly, rotations around the X_B, Y_B and Z_B-axis are called roll, pitch and yaw, respectively (Groves, 2008). A sketch of the body-fixed reference frame is given in Fig. 4-5. The orientation of the body-fixed frame of Silenus was defined in Section 3-1.

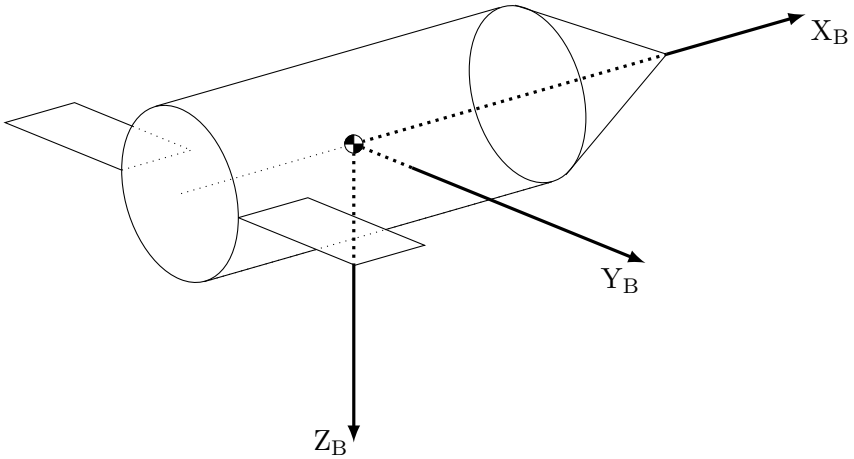


Figure 4-5: Definition of the body-fixed reference frame

4-2 State Variables and Coordinate Systems

State variables are required for the definition of the state of a dynamic system at a given instant of time. The lander's position, velocity and attitude – including their time derivatives – can be described in different coordinate systems.

The Keplerian orbital elements are useful state variables for describing a spacecraft's unperturbed motion in orbit. Expressing the trajectory of a lander during its descent, touchdown and repositioning phases in terms of orbital elements, however, is complicated and provides little physical insight. Cartesian or spherical coordinate systems in combination with Euler attitude angles or attitude quaternions are better suited for this situation, as the following subsections will show. At the end of this section, a state variable set is chosen after weighing the advantages and disadvantages of all systems. The last subsection presents the

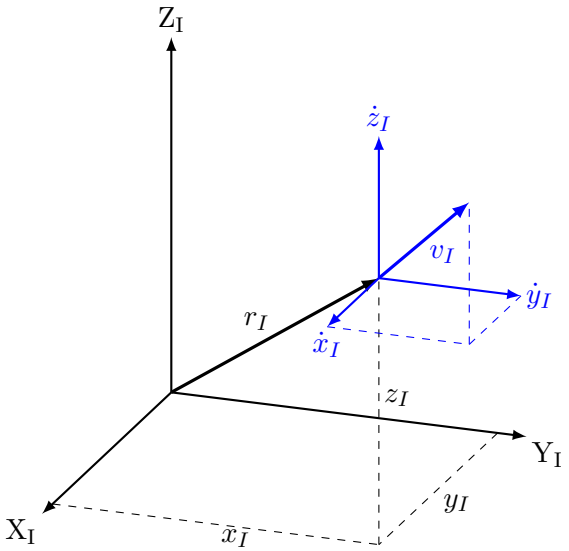


Figure 4-6: Position and velocity of an object in a Cartesian inertial reference frame

relations necessary for transformations between Cartesian and spherical coordinate systems; these equations are important irrespective of the chosen state variable set, as they are required for the derivation of the equations of motion in Section 4-4 and for converting a position given in an orthogonal reference frame (for example, the I-frame defined in the previous section) to position given in terms of longitude, latitude and altitude.

4-2-1 Cartesian Coordinate System

The Cartesian coordinate system is a very common, orthogonal system for the description of position and motion of an object in three-dimensional space. In any of the reference systems mentioned in the previous section, the position $\mathbf{r}$ and the velocity $\mathbf{v}$ of a body is defined by the variables x, y, z and $\dot{x}$, $\dot{y}$, $\dot{z}$, respectively. The index of the used reference frame should always be added. Figure 4-6 shows the Cartesian position and velocity components of an object in the inertial reference frame. In general, a Cartesian state vector describing a non-linear motion is difficult to analyze and involves rapidly changing state variables, which can lead to higher inaccuracies during the numerical integration of the trajectory (see *truncation error*, Section 4-5). The descent process and in particular the repositioning process, however, are of short duration, so the accumulated error will be within limits. An advantage of the Cartesian state vector is its simple concept of grouping position and velocity into three independent components. This allows, for example, for the direct analysis of the lander's vertical and horizontal position and velocity directly before the touchdown.

4-2-2 Spherical Coordinate System

The location of a point in a spherical coordinate system is described by its distance to the origin and two angles. Consequently, the spherical coordinate system is a good choice for measurements with respect to planetocentric reference frame, especially the rotating planetocentric frame. Using the definitions from Mooij (1997), a position is defined by the distance

r, the longitude τ and the latitude δ. The longitude and the latitude are positive in eastern and northern direction, respectively; see Fig. 4-7 for the definition of the angles on a unit sphere.

The velocity of an object with respect to the spherical coordinate system is called *ground speed* and expressed as a vector $\mathbf{V}_G$ with a flight-path angle γ_G and a heading χ_G. Here, γ_G is the angle between the local horizon and $\mathbf{V}_G$ ($-90^\circ \leqslant \gamma_G \leqslant 90^\circ$, positive above local horizon), and χ_G the angle between the local north and $\mathbf{V}_G$ ($-180^\circ \leqslant \chi_G < 180^\circ$, $+90^\circ$ is East). Figure 4-8 shows the definition of the spherical position and velocity parameters with respect to a rotating reference frame. A spherical state vector is easy to interpret, because its components have direct physical meaning. A major disadvantage of this system, however, are the singularities, that occur if $\delta = \pm 90^\circ$ or $\gamma_G = \pm 90^\circ$ (r =0 is impossible for surface landings). In this way, the state of a lander at the south pole or during a vertical flight cannot be described with a spherical coordinate system.

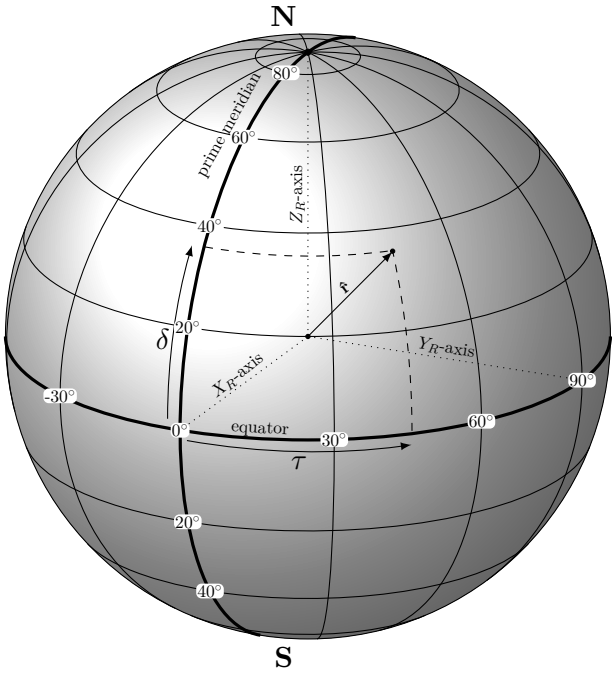


Figure 4-7: Definition of longitude and latitude in a spherical coordinate system on a unit sphere

4-2-3 Euler Attitude Angles

The attitude of an object can be described by a sequence of rotations around the X-, Y- and Z-axis, thus a transformation between two Cartesian coordinate systems. A common application of this procedure is the definition of the orientation of the body reference frame with respect to the inertial or the vertical reference frame. The Euler attitude angles consist of the roll angle ϕ about the X-axis, the pitch angle θ about the Y-axis and the yaw angle ψ about the Z-axis. In aerospace applications, the most common rotation sequence is $\psi \rightarrow \theta \rightarrow \phi$ (Kuipers, 1999). The Euler attitude sequence concept is shown in Fig. 4-9, where orientation

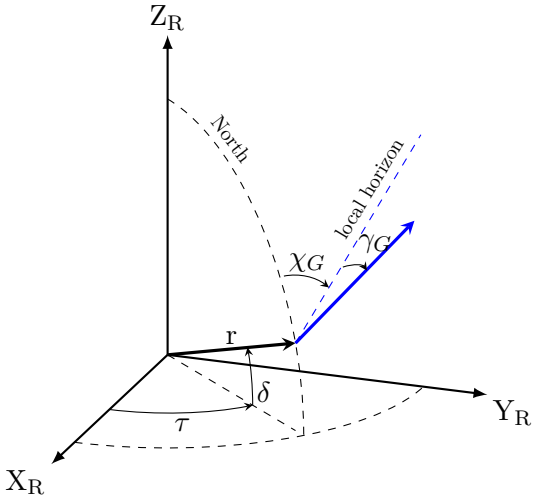


Figure 4-8: Definition of the parameters for position and velocity in the rotating planetocentric frame

of the body reference frame is found using a sequence of rotations with respect to the inertial reference frame.

The angular velocity of the body reference frame with respect to the inertial reference frame is given by the rotation vector $\boldsymbol{\omega} = [p, q, r]^T$, where p, q and r are the roll, pitch and yaw rate about the X_B-,Y_B- and Z_B-axis, respectively. A sketch of the geometrical relation is given in Fig. 4-10.

4-2-4 Attitude Quaternions

Quaternions are hyper-complex number consisting of four components and are often used to describe a rotation in three-dimensional space. Compared with the rotation sequences based on Euler attitude angles, quaternions are computationally more efficient and do not suffer from singularities. On the other hand, they provide no direct insight. A general quaternion $\bar{\mathbf{q}}$ has the form (Kuipers, 1999)

$$\bar{\mathbf{q}} = q_0 + \mathbf{q} = q_0 + \mathbf{i}q_1 + \mathbf{j}q_2 + \mathbf{k}q_3, \tag{4-1}$$

where q_0 and $\mathbf{q} = (q_1, q_2, q_3)^{\mathrm{T}}$ is the quaternion's scalar part and vector part, respectively. An attitude quaternion is always a unit quaternion, so

$$\|\bar{\mathbf{q}}\| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} = 1. \tag{4-2}$$

Equation (4-1) and (4-2) indicate that scalar part is fully determined by the elements of $\bar{\mathbf{q}}$. This redundancy may lead to problems during the numerical integration of the quaternion derivative $\dot{\bar{\mathbf{q}}}$, but the solution is relatively simple and discussed in Section 4-4-3. Furthermore, attitude quaternions are usually represented in vector form without the basis elements $\mathbf{i}$, $\mathbf{j}$ and $\mathbf{k}$, resulting in the form $\bar{\mathbf{q}} = (q_0, q_1, q_2, q_3)^{\mathrm{T}}$. Some authors add the scalar part after the vector part, which leads to different expressions for the equations of rotational motion, if the parts are not continuously treated separately (see, for example, Wie (2008)). In this report, the scalar part of $\bar{\mathbf{q}}$ is always the first element.

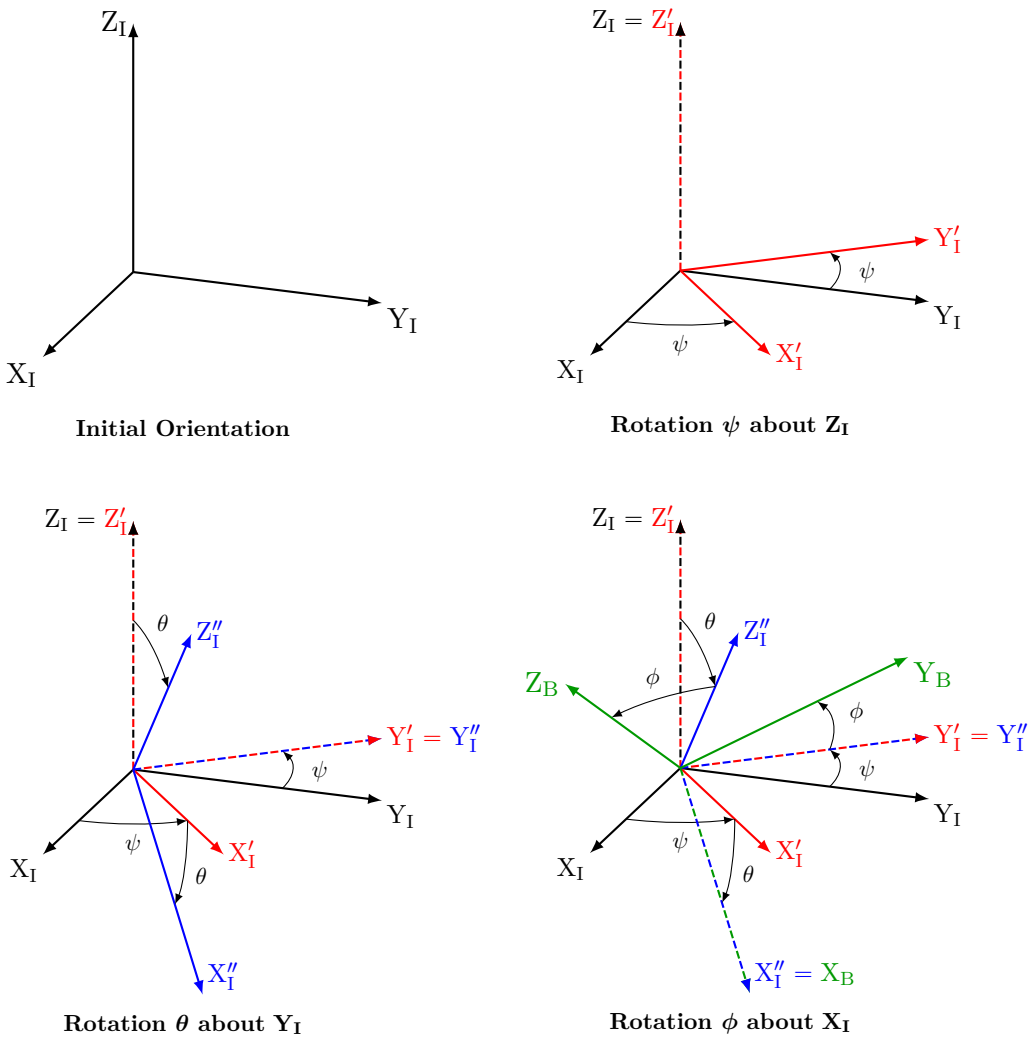


Figure 4-9: Use of the Euler angles to find the orientation of the body frame with respect to the inertial frame

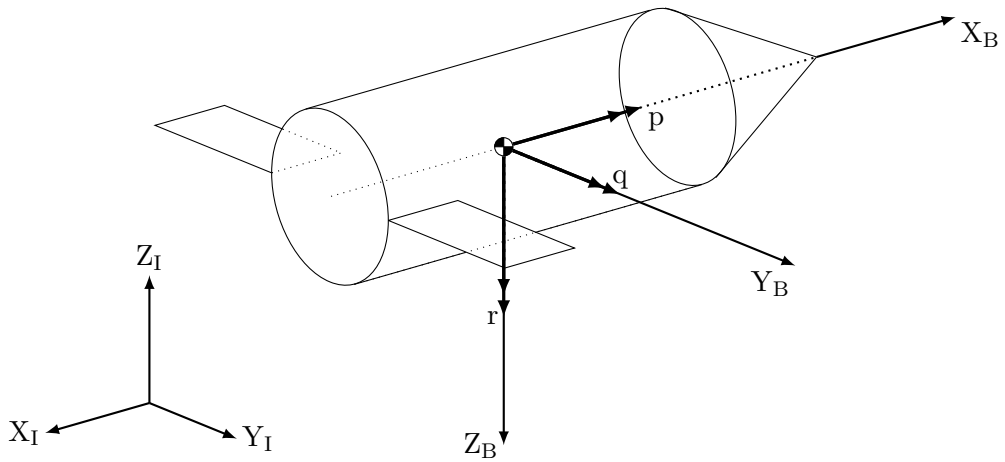


Figure 4-10: The angular velocity of the body frame w.r.t. the inertial planetocentric frame

A frame rotation about a vector $\mathbf{u}$ through an angle θ can be expressed by the quaternion q (Kuipers, 1999; Groves, 2008):

$$\bar{\mathbf{q}} = \left(\cos\frac{\theta}{2}, \mathbf{u}\sin\frac{\theta}{2}\right)^{\mathrm{T}} = \begin{pmatrix} \cos\frac{\theta}{2} \\ \frac{u_1}{\|\mathbf{u}\|}\sin\frac{\theta}{2} \\ \frac{u_2}{\|\mathbf{u}\|}\sin\frac{\theta}{2} \\ \frac{u_3}{\|\mathbf{u}\|}\sin\frac{\theta}{2} \end{pmatrix} \tag{4-3}$$

The rotation system shown in Eq. (4-3) can be used to bring a body from any initial orientation to the desired orientation, given the correct values of $\mathbf{u}$ and θ (Wie, 2008). Vector $\mathbf{u}$ is thus called the *eigenaxis*, and the concept of an eigenaxis rotation can be used, for example, in the design of an attitude controller.

The conversions between Euler angles, quaternions and direction cosine matrices are straightforward, and can be found on pages 167-169 of Kuipers (1999). Quaternions do not follow the commutative law for multiplication. The quaternion algebra is, for example, discussed in Kuipers (1999) and requires some additional routines for a correct software implementation. The flight-software of a spacecraft normally uses quaternions - also the on-board instruments that measure (changes of) position (star trackers, IMU etc.) usually return their results in the form of quaternions; a conversion is more important for post-processing.

4-2-5 Choice of State Variable Set

The Silenus lander will operate near Enceladus' south pole and perform purely vertical motion during the touchdown and take-off processes. Consequently, the spherical state vector should not be chosen due to the singularities that occur in these situations. The Cartesian state vector is the better choice for describing the motion of the spacecraft, because the six position and velocity components are important for the analysis of the touchdown process. In particular, the guidance laws presented in Section 5-1 are defined within an orthogonal coordinate system for one specific axis - a non-Cartesian state vector would require an additional state-variable transformation. The lander's attitude and change of attitude will be expressed in terms of quaternions and angular velocities, respectively. Quaternions are a more complex concept than Euler attitude angles, but they have no singularities, which is required for stable simulation software. Nevertheless, for the data analysis it is necessary to transform the quaternions to Euler attitude angles, because quaternions provide no physical insight. The relations required for these transformation are given in Section 4-2-7.

4-2-6 Coordinate System Transformation

With respect to the rotational planetocentric reference frame, the transformation from the Cartesian position $[x_R, y_R, z_R]$ to the spherical position $[r, \tau, \delta]$ is (Mooij, 1997; Wertz et al., 2009)

$$r = \sqrt{x_R^2 + y_R^2 + z_R^2} \tag{4-4a}$$

$$\tau = \mathrm{atan2}\left(y_R, x_R\right) \tag{4-4b}$$

$$\delta = \arcsin\left(\frac{z_R}{\sqrt{x_R^2 + y_R^2 + z_R^2}}\right) \tag{4-4c}$$

where the `atan2`-function assures that the correct quadrant is determined. The inverse transformation is given by (Mooij, 1997)

$$x_R = r\cos\delta\cos\tau \tag{4-5a}$$
$$y_R = r\cos\delta\sin\tau \tag{4-5b}$$
$$z_R = r\sin\delta \tag{4-5c}$$

The translation of a Cartesian velocity $\mathbf{V}_R = [\dot{x}_R, \dot{y}_R, \dot{z}_R]$ to a spherical velocity $\mathbf{V}_R = [V_G, \gamma_G, \chi_G]$ first requires a reference frame transformation to the vertical reference frame using $\mathbf{V}_V = \mathbf{T}_{V\leftarrow R}\mathbf{V}_R$ (see Section 4-3 for the definition of $\mathbf{T}_{V\leftarrow R}$). The relations then become (modified from Mooij (1997)):

$$V_G = \sqrt{\dot{x}_V^2 + \dot{y}_V^2 + \dot{z}_V^2} \tag{4-6a}$$
$$\gamma_G = -\arcsin\left(\frac{\dot{z}_V}{V_G}\right) \tag{4-6b}$$
$$\chi_G = \text{atan2}\left(\dot{y}_V, \dot{x}_V\right) \tag{4-6c}$$

The inverse operation $\mathbf{V}_R = \mathbf{T}_{R\leftarrow V}\mathbf{V}_V$ then yields the spherical velocity with respect to the rotating reference frame. Similarly, the transformation from the spherical velocity to the Cartesian velocity is given by (Mooij, 1997):

$$\dot{x}_V = V_G\cos\gamma_G\cos\chi_G \tag{4-7a}$$
$$\dot{y}_V = V_G\cos\gamma_G\sin\chi_G \tag{4-7b}$$
$$\dot{z}_V = V_G\sin\gamma_G \tag{4-7c}$$

Again, the inverse operation $\mathbf{V}_R = \mathbf{T}_{R\leftarrow V}\mathbf{V}_V$ yields the velocity with respect to the rotating reference frame.

4-2-7 Attitude System Transformation

For the Euler attitude angles ϕ, θ and ψ as defined in Section 4-2-3 and the rotation sequence $\psi \rightarrow \theta \rightarrow \phi$, the attitude quaternion can be calculated with (Kuipers, 1999)

$$\bar{\mathbf{q}} = q_0 + \mathbf{i}q_1 + \mathbf{j}q_2 + \mathbf{k}q_3 \tag{4-8}$$

where

$$\begin{aligned}
q_0 &= \cos\frac{\psi}{2}\cos\frac{\theta}{2}\cos\frac{\phi}{2} + \sin\frac{\psi}{2}\sin\frac{\theta}{2}\sin\frac{\phi}{2}\\
q_1 &= \cos\frac{\psi}{2}\cos\frac{\theta}{2}\sin\frac{\phi}{2} - \sin\frac{\psi}{2}\sin\frac{\theta}{2}\cos\frac{\phi}{2}\\
q_2 &= \cos\frac{\psi}{2}\sin\frac{\theta}{2}\cos\frac{\phi}{2} + \sin\frac{\psi}{2}\cos\frac{\theta}{2}\sin\frac{\phi}{2}\\
q_3 &= \sin\frac{\psi}{2}\cos\frac{\theta}{2}\cos\frac{\phi}{2} - \cos\frac{\psi}{2}\sin\frac{\theta}{2}\sin\frac{\phi}{2}.
\end{aligned} \tag{4-9}$$

Given a quaternion $\mathbf{q}$ with the elements q_0, q_1, q_2 and q_3, the Euler angles ϕ, θ and ψ follow from the relations (Groves, 2008)

$$\begin{aligned}\phi &= \text{atan2}\left(2\left(q_2q_3 + q_0q_1\right), 2q_0^2 + 2q_3^2 - 1\right) \\ \theta &= -\arcsin\left(2\left(q_1q_3 + q_0q_2\right)\right) \\ \psi &= \text{atan2}\left(2\left(q_1q_2 + q_0q_3\right), 2q_0^2 + 2q_1^2 - 1\right).\end{aligned} \tag{4-10}$$

For the integration of the equations of motion, the attitude quaternion must translated into a transformation matrix. The transformation matrix, or direction cosine matrix, for a quaternion $\mathbf{q}$ of the form shown in Eq. (4-8) is (Kuipers, 1999)

$$\mathbf{T}_q = \begin{bmatrix} 2q_0^2 + 2q_1^2 - 1 & 2\left(q_1q_2 + q_0q_3\right) & 2\left(q_1q_3 - q_0q_2\right) \\ 2\left(q_1q_2 - q_0q_3\right) & 2q_0^2 + 2q_2^2 - 1 & 2\left(q_2q_3 + q_0q_1\right) \\ 2\left(q_1q_3 + q_0q_2\right) & 2\left(q_2q_3 - q_0q_1\right) & 2q_0^2 + 2q_3^2 - 1 \end{bmatrix} \tag{4-11}$$

4-3 Reference Frame Transformations

In Section 4-1, five different reference frames have been defined. Any vector expressed in a given reference frame can be translated into another reference frame by rotating the original frame until it matches the target frame. This is commonly done using a series of rotations about the principle axes of a Cartesian coordinate system. The required rotation matrices are defined in Subsection 4-3-1. The actual transformation matrices between the reference frames of Section 4-1 are given in the subsequent subsections.

4-3-1 Elementary Axis-Rotations

During an Euler-axis rotation, a reference frame is rotated by an angle ϕ, θ or ψ about the X-, Y- or Z-axis of an Cartesian coordinate system (see Fig. 4-9). For the derivation of the transformation matrices it is simpler to write ϕ, θ and ψ as an arbitrary rotation about an angle α. The rotation matrices are then defined as (Mooij, 1997; Wie, 1998)

$$\mathbf{T_1}(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha \\ 0 & -\sin\alpha & \cos\alpha \end{bmatrix} \tag{4-12a}$$

$$\mathbf{T_2}(\alpha) = \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{bmatrix} \tag{4-12b}$$

$$\mathbf{T_3}(\alpha) = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4-12c}$$

where the subscripts 1, 2 and 3 stand for the X-, Y- or Z-axis of the initial reference frame.

A transformation matrix for the translation of a vector from the i-frame to the j-frame is denoted with $\mathbf{T}_{j\leftarrow i}$ in this report. Any transformation matrix between the reference frames

listed in Section 4-1 can be expressed as the result of a rotation sequence using the expressions for $\mathbf{T_1}$, $\mathbf{T_2}$ and $\mathbf{T_3}$. All transformation matrices are by definition orthonormal and thus represent a linear transformation. Any rotation can be negated by rotating over the same angle in reverse direction:

$$\mathbf{T}_{j\leftarrow i} = \mathbf{T}^{-1}_{i\leftarrow j} = \mathbf{T}^{\mathrm{T}}_{i\leftarrow j} \tag{4-13}$$

A rotation sequence is a successive rotation of a reference frame about its current axes, and can be represented by a single transformation matrix as the product of rotation matrices. In this way, the transformation matrix from frame a to b using the successive rotations i, j and k is

$$\mathbf{T}_{b\leftarrow a} = \mathbf{T}_k\mathbf{T}_j\mathbf{T}_i. \tag{4-14}$$

The transformation matrix for the Euler rotation sequence $\psi \rightarrow \theta \rightarrow \phi$ discussed in Section 4-2-3 then becomes

$$\mathbf{T}_{f_2\leftarrow f_1} = \mathbf{T}_1(\phi)\mathbf{T}_2(\theta)\mathbf{T}_3(\psi) \tag{4-15}$$

where f_1 and f_2 are the initial and the final reference frames. The lander's attitude is always expressed as the orientation of the body-fixed reference frame with respect to another reference frame, which may be chosen freely as long as it is defined unambiguously. The Enceladus lander simulator requires the specification of the rotation sequence from the I-frame to the B-frame as user input. The attitude with respect to the other reference frames is determined using the appropriate rotation matrix sequence. This also means, that it is not necessary to derive the exact expressions for the rotations between all reference frames used in this report, because they follow from a sequence of a few basic transformations.

4-3-2 Transformation Matrices

The following sections present the basic transformation matrices between the reference frames given in Section 4-1 using the relations discussed in Section 4-3-1.

Rotating Planetocentric to Inertial Planetocentric Frame

The rotating planetocentric reference frame rotates with the rotational velocity of the central body ω_{cb} around the Z_I-axis and coincides with the inertial planetocentric frame at the instance of time $t = 0$. The transformation matrix from the rotating planetocentric to the inertial planetocentric reference frame after t seconds then becomes (Mooij, 1997):

$$\mathbf{T}_{I\leftarrow R} = \mathbf{C_3}(-\omega_{cb}t) = \begin{bmatrix} \cos\omega_{cb}t & -\sin\omega_{cb}t & 0 \\ \sin\omega_{cb}t & \cos\omega_{cb}t & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4-16}$$

In case of Enceladus, $\omega_{cb} = \frac{2\pi}{1.370218 \text{ days}} = 5.30733 \cdot 10^{-5}$rad/s (see Table 2-1).

Vertical to Rotating Planetocentric Frame

For a vehicle located at a longitude τ and a latitude δ on the surface of the central body (see definitions shown in Fig. 4-7 and 4-8), the transformation matrix from the vehicle to the rotating planetocentric frame is (Mooij, 1997):

$$\mathbf{T}_{R\leftarrow V} = \mathbf{C_3}(-\tau)\mathbf{C_2}(\frac{\pi}{2}+\delta) = \begin{bmatrix} -\cos\tau\sin\delta & -\sin\tau & -\cos\tau\cos\delta \\ -\sin\tau\sin\delta & \cos\tau & -\sin\tau\cos\delta \\ \cos\delta & 0 & -\sin\delta \end{bmatrix} \tag{4-17}$$

Inertial Planetocentric to Body-fixed Frame

The alignment of the body-fixed reference frame with respect to the inertial planetocentric reference frame can be expressed with the Euler attitude angles ϕ_I, θ_I and ψ_I (see Fig. 4-9). Then, the transformation matrix follows from Eq. (4-15) and can be written as (Kuipers, 1999):

$$\begin{aligned} \mathbf{T}_{B\leftarrow I} &= \mathbf{C_1}(\phi_I)\mathbf{C_2}(\theta_I)\mathbf{C_3}(\psi_I) \\ &= \begin{bmatrix} \cos\psi_I\cos\theta_I & \sin\psi_I\cos\theta_I & -\sin\theta_I \\ \begin{pmatrix}\cos\psi_I\sin\theta_I\sin\phi_I \\ -\sin\psi_I\cos\phi_I\end{pmatrix} & \begin{pmatrix}\sin\psi_I\sin\theta_I\sin\phi_I \\ +\cos\psi_I\cos\phi_I\end{pmatrix} & \cos\theta_I\sin\phi_I \\ \begin{pmatrix}\cos\psi_I\sin\theta_I\cos\phi_I \\ +\sin\psi_I\sin\phi_I\end{pmatrix} & \begin{pmatrix}\sin\psi_I\sin\theta_I\cos\phi_I \\ -\cos\psi_I\sin\phi_I\end{pmatrix} & \cos\theta_I\cos\phi_I \end{bmatrix} \end{aligned} \tag{4-18}$$

Vertical to Surface-Fixed Frame

As discussed in Section 4-1-4, the easiest way to define the surface-fixed reference frame is by using a rotation sequence staring at the current vertical reference frame. For the azimuth angle χ_V, the transformation matrix from the vertical to the surface-fixed frame is

$$\mathbf{T}_{S\leftarrow V} = \mathbf{C_2}\left(\frac{\pi}{2}\right)\mathbf{C_3}\left(-\frac{\pi}{2}+\chi_V\right) = \begin{bmatrix} 0 & 0 & -1 \\ \cos\chi_V & \sin\chi_V & 0 \\ \sin\chi_V & -\cos\chi_V & 0 \end{bmatrix} \tag{4-19}$$

The azimuth angle χ_V follows from expressing the X_B-axis with respect to the V-frame,

$$\mathbf{x}_{B,V} = \mathbf{T}_{V\leftarrow B}\left[1,0,0\right]^{\mathrm{T}} \tag{4-20}$$

and then using an adapted version of Eq. (4-6c) to determine the azimuth angle of $X_{B,V}$:

$$\chi_V = \arctan\left(\frac{\mathbf{x}_{B,V_y}}{\mathbf{x}_{B,V_x}}\right) \tag{4-21}$$

Remaining Transformation Matrices

The transformation matrices between two arbitrary reference frames is either one of the matrices given in the previous subsections or is the product of two or more of these matrices.

For example, the transformation matrices between the inertial and the surface-fixed frame are

$$\mathbf{T}_{S\leftarrow I} = \mathbf{T}_{S\leftarrow V}\mathbf{T}_{V\leftarrow R}\mathbf{T}_{R\leftarrow I} \tag{4-22}$$

and, accordingly,

$$\mathbf{T}_{I\leftarrow S} = \mathbf{T}_{I\leftarrow R}\mathbf{T}_{R\leftarrow V}\mathbf{T}_{V\leftarrow S} = \mathbf{T}^{\mathrm{T}}_{S\leftarrow I} \tag{4-23}$$

where the inverse rotations follow from the general rule shown in Eq. (4-13).

4-4 Flight Mechanics

The study of flight mechanics analyzes the effects of forces and moments on the motion of a body. The knowledge about how a spacecraft responds to gravitational accelerations, propulsive forces, attitude control actions etc. allows the design of an effective guidance and control system.

The structure of the lander is assumed to be stiff, and the landing gear is the only moving part. Consequently, during descent, the lander can be represented by a non-elastic, mass-varying vehicle with three translational and three rotational degrees of freedom. The atmospheric density of the moon, even at the south polar region, is completely negligible and allows for the elimination of all aerodynamic influences.

This chapter starts in Section 4-4-1 with the derivation of the external forces and moments acting on the lander, primarily caused by the gravitational acceleration and the propulsive system. The translational and rotational equations of motion are determined in Sections 4-4-2 and 4-4-3, respectively.

The notations and some of the structure of the following derivations are based on Mooij (1997) and Cornelisse et al. (1979). A concise summary of the basic aspects of analytical mechanics can be found in Török (2000).

4-4-1 Forces and Moments acting on the Lander

As discussed in Chapter 2, Enceladus has a negligible atmospheric density. The only external forces and moments acting on the lander during descent, touchdown and repositioning are induced by the gravitational accelerations of all relevant celestial bodies (see Section 4-7-1) and by the on-board thrusters. The effects of these forces are discussed in the following two subsections.

Gravitational acceleration

The gravity field of a planetary body in general is non-homogeneous due to its irregular mass distribution. These gravitational anomalies have a direct influence on an orbiting satellite and should be included in the orbital-mechanics calculations in case the perturbing accelerations are large or accumulate over time.

In case the mass distribution of the main body is symmetric about the polar axis, a spacecraft does not experience a longitudinal gravitational force. The gravitational acceleration vector $\mathbf{g}$ then only has a radial component g_r and a latitudinal component g_δ, so the gravitational force with respect to the spherical reference frame becomes

$$\mathbf{g} = \begin{pmatrix} g_r \\ 0 \\ g_\delta \end{pmatrix}. \tag{4-24}$$

The radial component g_r of $\mathbf{g}$ can be written as (Mooij, 1997; Regan, 1984)

$$g_r = \frac{\mu}{r^2}\left[1 - \frac{3}{2}J_2\left(\frac{R}{r}\right)^2\left(3\sin^2\delta - 1\right) - 2J_3\left(\frac{R}{r}\right)^3\sin\delta\left(5\sin^2\delta - 3\right) - \frac{5}{8}J_4\left(\frac{R}{r}\right)^4\left(35\sin^4\delta - 30\sin^2\delta + 3\right)\right] \tag{4-25}$$

and the latitudinal component g_δ as (Mooij, 1997; Regan, 1984)

$$g_\delta = -3\frac{\mu}{r^2}\left(\frac{R}{r}\right)^2\sin\delta\cos\delta\left[J_2 + \frac{1}{2}J_3\left(\frac{R}{r}\right)\sin^{-1}\delta\left(5\sin^2\delta - 1\right) + \frac{5}{6}J_4\left(\frac{R}{r}\right)^2\left(7\sin^2\delta - 3\right)\right] \tag{4-26}$$

where μ and R are the gravitational parameter and the equatorial radius of the central body, r is the distance to the c.o.m. of central body, J_n are the gravitational moments of the order n, and δ is the latitude of the point under investigation in the rotating spherical coordinate system (see definitions in Section 4-2-2).

Gas giants such as Saturn can be interpreted with high accuracy as rotating fluid bodies in hydrostatic equilibrium (Pater and Lissauer, 2001), which refers to the condition in which the planetary interior in a fluid form would experience no pressures other than its own weight. Several moons in the Saturnian system have a shape that is consistent with the one of relaxed body in hydrostatic equilibrium (Thomas et al., 2007). Enceladus is close to hydrostatic equilibrium (Porco et al., 2006), but there are some irregularities especially in the south polar region, probably leading to a small negative J_3 value. The gravity field of Enceladus, however, is currently not fully known; gravitational moments with degrees higher than 2 are not known with certainty (Russell and Lara, 2009). Enceladus' J_3 value is several orders of magnitude smaller than its J_2 value (2500×10^{-6}, see Table 4-1) and can be assumed to be equal to zero with great accuracy. The estimate for the gravitational parameter of Enceladus, μ_{Enc}, was updated using observations from Cassini and found to be 7.2096 km^3s^{-2} (see Table 2-1). For perfectly *axisymmetric* bodies – which is a reasonable assumption for Enceladus – the gravitational moments J are zero for odd n.

The values of Saturn's first four even numbered gravitational moments and value of J_2 for Enceladus are listed in Table 4-1. As a first estimate, only the perturbations due to the J_2-term are considered, because they are at least one order of magnitude larger than the perturbation induced by the other gravitational moments.

Gravitational Moment	Saturn value [10^{-6}]	Enceladus value [10^{-6}]
J_2	16290.71 $\pm$ 0.27	2500
J_4	-935.83 $\pm$ 2.77	-
J_6	86.14 $\pm$ 9.64	-
J_8	-10.0	-

Table 4-1: Gravitational moments of Saturn and Enceladus (Jacobson et al., 2006; Russell and Lara, 2009)

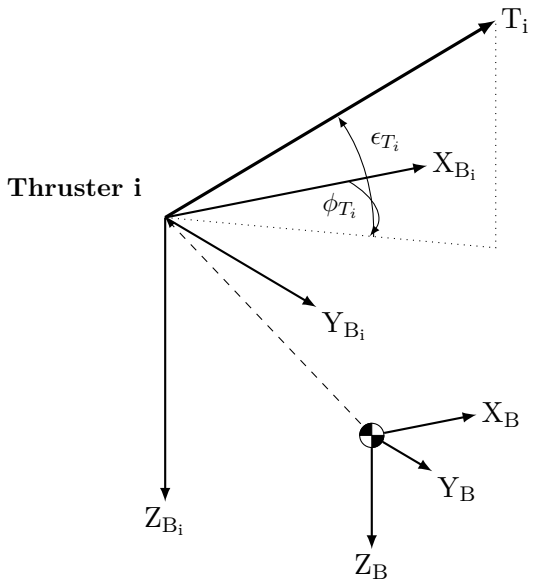


Figure 4-11: Thrust vector of a thruster i with respect to the c.o.m. of the vehicle

It is necessary to express Eqs. (4-24) to (4-26) in Cartesian coordinates to receive the gravitational force with respect to the R-frame, $\mathbf{F_{G,R}}$, but this form will not be used during the further discussions and thus not shown here.

The gravitation of the central body can induce a moment to spacecraft in case gravitational attraction changes over the vehicle body. This effect especially influences large satellites during long missions. However, gravitational moments are negligible for the short descent and repositioning missions phases and generally much smaller than thrust moments.

Propulsive Force

The lander on Enceladus has a main propulsion system and several attitude thrusters (see Section 3-2). The effects of their propulsive forces on the vehicle's translational and rotational motion are discussed in this section.

Let the thruster i be located at a position $\mathbf{r_{T_i}}$ in the vehicle's body reference frame. The direction of its thrust T_i can then be expressed by the elevation angle ϵ_{T_i} and the azimuth angle ψ_{T_i} with respect to a projected, collinear body reference frame with the origin located at $\mathbf{r_{T_i}}$ (see Figure 4-11).

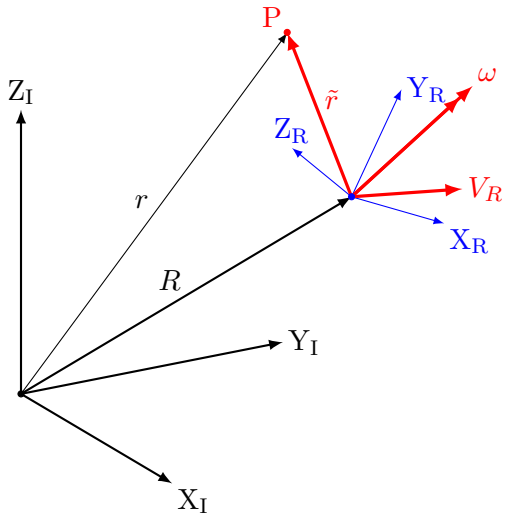


Figure 4-12: Motion of a point expressed in an inertial and a rotating reference frame

The thrust force $\mathbf{F}_{T,B}$ is sum of all T_i:

$$\mathbf{F_{T,B}} = \sum_i \mathbf{T_i} = \begin{pmatrix} T_x \\ T_y \\ T_z \end{pmatrix} \tag{4-27}$$

The components T_x, T_y and T_z follow directly from Fig. 4-11 (Mooij, 1997):

$$\begin{aligned} T_x &= T \cos\phi_T \cos\epsilon_T \\ T_y &= T \sin\phi_T \cos\epsilon_T \\ T_z &= -T \sin\epsilon_T \end{aligned} \tag{4-28}$$

Any thrust vector that does not pass through the c.o.m. of the spacecraft induces a propulsion moment. Following the basic definition of the force moment, the propulsion moment $\mathbf{M}_{T,B_i}$ of a thruster i is the equal to the cross product of the position vector $\mathbf{r}_{T_i}$ and the thrust force vector $\mathbf{F}_{T,B_i}$, all measured with respect to body reference frame. The total propulsion moment for all thrusters n then becomes

$$\mathbf{M}_{T,B} = \sum_{i=1}^{n} \mathbf{M}_{T,B_i} = \sum_{i=1}^{n} \mathbf{r}_{T,i} \times \mathbf{F}_{T,B_i}. \tag{4-29}$$

4-4-2 Equations of Translational Motion

A point p is located at the position $\tilde{\mathbf{r}}$ within a rotating reference frame (see Fig. 4-12). If the position vector of p and the position vector of the origin of the rotating reference frame with respect to the inertial reference frame are designated with $\mathbf{r}$ and $\mathbf{R}$, respectively, then the following relation holds:

$$\mathbf{r} = \mathbf{R} + \tilde{\mathbf{r}} \tag{4-30}$$

The velocity of p with respect to the inertial reference frame can be determined by differentiating Eq. (4-30). This chapter follows the derivative notations of Cornelisse et al. (1979),

where $\frac{\mathrm{d}}{\mathrm{d}t}$ refers to a derivative with respect to the inertial reference frame and $\frac{\delta}{\delta t}$ to a derivative with respect to the rotating reference frame. Keeping in mind that $\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t}$ represents the *absolute* velocity $\mathbf{V}_I$,

$$\begin{aligned}\mathbf{V}_I = \frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t} &= \frac{\mathrm{d}\mathbf{R}}{\mathrm{d}t} + \frac{\mathrm{d}\tilde{\mathbf{r}}}{\mathrm{d}t} \\ &= \frac{\mathrm{d}\mathbf{R}}{\mathrm{d}t} + \frac{\delta\tilde{\mathbf{r}}}{\delta t} + \omega \times \tilde{\mathbf{r}}\end{aligned} \tag{4-31}$$

where ω is the angular velocity of $\tilde{\mathbf{r}}$ with respect to the inertial reference frame. The last two terms on the right-hand side of Eq. (4-31) follow from working out $\frac{\delta}{\delta t}$, which involves the time derivatives of the unit ayes x_R, y_R and z_R in the inertial reference frame. Similarly, the absolute acceleration $\mathbf{a}_I$ of p is

$$\begin{aligned}\mathbf{a}_I = \frac{\mathrm{d}\mathbf{V_I}}{\mathrm{d}t} &= \frac{\mathrm{d}^2\mathbf{R}}{\mathrm{d}t^2} + \frac{\mathrm{d}}{\mathrm{d}t}\left(\boldsymbol{\omega} \times \tilde{\mathbf{r}}\right) + \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\tilde{\mathbf{r}}}{\delta t}\right) \\ &= \frac{\mathrm{d}^2\mathbf{R}}{\mathrm{d}t^2} + \frac{\mathrm{d}\boldsymbol{\omega}}{\mathrm{d}t} \times \tilde{\mathbf{r}} + \boldsymbol{\omega} \times \left(\frac{\delta\tilde{\mathbf{r}}}{\delta t} + \boldsymbol{\omega} \times \tilde{\mathbf{r}}\right) + \boldsymbol{\omega} \times \frac{\delta\tilde{\mathbf{r}}}{\delta t} + \frac{\delta^2\tilde{\mathbf{r}}}{\delta t^2} \\ &= \frac{\mathrm{d}^2\mathbf{R}}{\mathrm{d}t^2} + \frac{\mathrm{d}\boldsymbol{\omega}}{\mathrm{d}t} \times \tilde{\mathbf{r}} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \tilde{\mathbf{r}}) + 2\boldsymbol{\omega} \times \frac{\delta\tilde{\mathbf{r}}}{\delta t} + \frac{\delta^2\tilde{\mathbf{r}}}{\delta t^2}\end{aligned} \tag{4-32}$$

In this equation, only the acceleration of the rotating reference frame with respect the inertial reference frame given by the term $\frac{d^2\mathbf{R}}{dt^2}$ represents an absolute acceleration. The term $2\boldsymbol{\omega} \times \frac{\delta\tilde{\mathbf{r}}}{\delta t}$ represents the so-called *Coriolis* acceleration, and the term $\frac{\delta^2\tilde{\mathbf{r}}}{\delta t^2}$ the *relative* acceleration. The sum of the remaining two term is called the *dragging* acceleration(Mooij, 1997). Equation (4-32) describes the acceleration of a point in a rotating reference frame with respect to an inertial reference frame and as such forms the basis for the derivations of the equations of motion in the following sections.

The sum of all external forces acting on a mass-varying body with respect to the inertial reference frame is (Mooij, 1997)

$$\mathbf{F}_I = \int_m \frac{\mathrm{d}^2\mathbf{r}}{\mathrm{d}t^2}\mathrm{d}m \tag{4-33}$$

where $\mathbf{r}$ represents the position vector of a mass element. The position $\mathbf{r}_{cm}$ of the body's center of mass follows from (Mooij, 1997)

$$\mathbf{r}_{cm}\int_m \mathrm{d}m = \int_m \mathbf{r}\mathrm{d}m \tag{4-34}$$

In this way, the location of a mass element may be written as a combination of $\mathbf{r}_{cm}$ and the location $\tilde{\mathbf{r}}$ of the same mass element with respect to the c.o.m.:

$$\mathbf{r} = \mathbf{r}_{cm} + \tilde{\mathbf{r}} \tag{4-35}$$

The expression for $\mathbf{F}_I$ as given in Eq. (4-33) can now be rewritten using the general Eq. (4-32) for the acceleration of the rotating reference frame with respect the inertial reference frame

given. With $\frac{\mathrm{d}^2\mathbf{R}}{\mathrm{d}^2t} = \frac{\mathrm{d}^2\mathbf{r_{cm}}}{\mathrm{d}^2t}$, $\int_m \tilde{\mathbf{r}}\mathrm{d}m = 0$ and assuming a constant rotation velocity of the body $\left(\frac{\mathrm{d}\boldsymbol{\omega}}{\mathrm{d}t} = 0\right)$, it follows from Eq. (4-32) and (4-33) (Mooij, 1997):

$$\mathbf{F}_I = m\frac{\mathrm{d}^2\mathbf{r_{cm}}}{\mathrm{d}t^2} + 2\boldsymbol{\omega} \times \int_m \frac{\delta\tilde{\mathbf{r}}}{\delta t}\mathrm{d}m + \int_m \frac{\delta^2\tilde{\mathbf{r}}}{\delta t^2}\mathrm{d}m \tag{4-36}$$

Following the naming of the elements of Eq. (4-32), the last two terms of the above equations are brought to the left-hand side and replaced with $\mathbf{F}_C$ and $\mathbf{F}_{rel}$ to indicate the Coriolis force and relative force, respectively:

$$\tilde{\mathbf{F}}_I = \mathbf{F}_I + \mathbf{F}_C + \mathbf{F}_{rel} = m\frac{\mathrm{d}^2\mathbf{r}_{cm}}{\mathrm{d}t^2} \tag{4-37}$$

Equation (4-37) describes the translational motion of a body with varying mass distribution in a non-rotating reference frame. The terms $\mathbf{F}_C$ and $\mathbf{F}_{rel}$ are a consequence of these changes in mass distribution. This equation also visualizes the so-called *Principle of Solidification* (Cornelisse et al., 1979) which describes the translational and rotational motion of a body with varying mass at a given instance of time as the translational and rotational motion of a *rigid* body, with two apparent forces and moments added to the true external forces and moments.

The forces $\mathbf{F}_C$ and $\mathbf{F}_{rel}$ both arise from variations in mass distribution. In absence of an atmosphere, the only mass change follows from the exhaust mass flow. A starting point for the derivation of different expression for $\mathbf{F}_C$ and $\mathbf{F}_{rel}$ is the Reynolds' Transport Theorem. Using the expressions of this chapter and assuming a conventional rocket without a mass inflow, this theorem can be written as (modified from Mooij (1997)):

$$\int_m \frac{\delta\tilde{\mathbf{r}}}{\delta t}\mathrm{d}m = \frac{\delta}{\delta t}\int_m \tilde{\mathbf{r}}\mathrm{d}m + \int_{A_e} \tilde{\mathbf{r}}\left(\rho_e\mathbf{V_e}\cdot\mathbf{n_e}\right)\mathrm{d}A_e \tag{4-38}$$

In this equation, ρ_e is the exhaust mass density, $\mathbf{V}_e$ the exhaust mass flow velocity with respect to the vehicle, A_e the exhaust area, and $\mathbf{n}_e$ a unit vector orthogonal to A_e. From Eq. (4-34), (4-35) and (4-38) it follows that the Coriolis force due to variable mass properties can be written as

$$\mathbf{F}_C = -2\boldsymbol{\omega} \times \int_m \frac{\delta\tilde{\mathbf{r}}}{\delta t}\mathrm{d}m = -2\boldsymbol{\omega} \times (\dot{m}_e\mathbf{r}_e) \tag{4-39}$$

where $\dot{m}_e$ is the exhaust mass flow and $\mathbf{r}_e$ the location of the mass flow center, defined as (Mooij, 1997):

$$\dot{m}_e = \int_{A_e} \left(\rho_e\mathbf{V}_e\cdot\mathbf{n}_e\right)\mathrm{d}A_e \tag{4-40}$$

$$\mathbf{r_e} = \frac{1}{\dot{m}_e}\int_{A_e} \tilde{\mathbf{r}}\left(\rho_e\mathbf{V}_e\cdot\mathbf{n}_e\right)\mathrm{d}A_e \tag{4-41}$$

The Coriolis force due to changing mass properties is small, because direction of the exhaust flow coincides with the lander's Z_B-axis, and the spin velocity of a spacecraft generally is

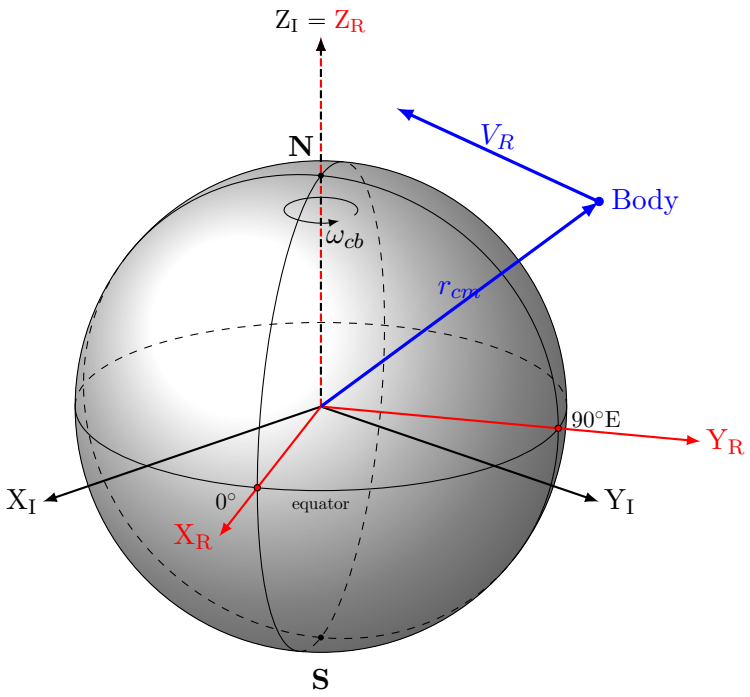


Figure 4-13: Motion of a point in the rotating planetocentric reference frame and the inertial reference frame

relatively low. The exhaust mass flow $\dot{m}_e$ of the lander is 0.158 kgs^{-1} (see Table 3-2), and the location of mass flow center is on the Z_B-axis, about 320 mm below the c.o.m. of the vehicle (see Table 3-1). Assuming a very high spin velocity of 20 degs^{-1} or 0.3491 rads^{-1} about all axes, the Coriolis force according to Eq. (4-39) is $\mathbf{F}_C = [-0.0353,\ 0.0353,\ 0]$ N, which is four orders of magnitude smaller than the actual thrust force of 490 N. This difference is even smaller for the expected lower spin rates. As a consequence, the result of $\boldsymbol{\omega} \times \mathbf{r}_e$ and thus $\mathbf{F}_C$ can be neglected, especially with respect to the much larger $\mathbf{F}_{rel}$.

The alternate expression for the relative force $\mathbf{F}_{rel}$ follows from the time derivative of Eq. (4-38). With the safe assumption that the mean exhaust velocity $\overline{\mathbf{V}}_e$ is much larger the relative velocity of the c.o.m. with respect to the spacecraft frame, the relative fore due to changing mass properties can be written as (Mooij, 1997)

$$\mathbf{F_{rel}} = -\dot{m}_e \overline{\mathbf{V}}_e \tag{4-42}$$

The expression for the translational motion in Eq. (4-37) will now be translated into a formulation for a mass varying body in the *rotating* planetocentric reference frame. This situation is shown in Fig. 4-13.

As mentioned in the definition of the rotating planetocentric reference frame (section 4-1-2), the origins of the Z-axes of the rotating planetocentric reference frame and the inertial planetocentric reference frame always coincide. Equation (4-34) then becomes (Mooij, 1997):

$$\mathbf{r}_{cm,R} = \mathbf{R} + \mathbf{r}_{cm,R} = \mathbf{r}_{cm,R} \tag{4-43}$$

With Eq. (4-37) and (4-43), the acceleration of the body's c.o.m. with respect to the inertial reference frame, expressed in the rotating reference frame, can be determined with the general

expression from Eq. (4-32), where $\boldsymbol{\omega}_R$ is the (constant) rotating reference frame's angular velocity (Mooij, 1997):

$$\mathbf{a}_R = \frac{\mathrm{d}^2\mathbf{r}_{cm}}{\mathrm{d}t^2} = \frac{\mathrm{d}^2\mathbf{R}}{\mathrm{d}t^2} + \frac{\mathrm{d}\boldsymbol{\omega}}{\mathrm{d}t} \times \mathbf{r}_{cm,R} + \boldsymbol{\omega}_R \times (\boldsymbol{\omega}_R \times \mathbf{r}_{cm,R}) + 2\boldsymbol{\omega}_R \times \frac{\delta \mathbf{r}_{cm,R}}{\delta t} + \frac{\delta^2 \mathbf{r}_{cm,R}}{\delta t^2} \quad (4\text{-}44)$$

The first two terms of this equation are equal to zero as a consequence of the previously mentioned definitions, which also allow the replacement of $\frac{\delta}{\delta t}$ and $\frac{\delta^2}{\delta t^2}$ by $\frac{\mathrm{d}}{\mathrm{d}t}$ and $\frac{\mathrm{d}^2}{\mathrm{d}t^2}$, respectively. The translational equation motion for a mass-varying body with respect to the rotating reference frame then finally becomes

$$\mathbf{F}_R = m\frac{\mathrm{d}^2\mathbf{r}_{cm}}{\mathrm{d}t^2} + 2m\boldsymbol{\omega}_R \times \frac{\mathrm{d}\mathbf{r}_{cm,R}}{\mathrm{d}t} + m\boldsymbol{\omega}_R \times (\boldsymbol{\omega}_R \times \mathbf{r}_{cm}) \,. \quad (4\text{-}45)$$

Note that according to the comments on Eq. (4-37), $\mathbf{F}_R$ is the sum of the true external forces *and* the two apparent forces (the relative force and the negligible small Coriolis force) resulting from the mass variations. The velocity $\mathbf{V}_R$ of the vehicle with the c.o.m. at $\mathbf{r}_{cm}$ with respect to the rotating planetocentric reference frame is

$$\mathbf{V}_R = \frac{\mathrm{d}\mathbf{r}_{cm}}{\mathrm{d}t} \quad (4\text{-}46)$$

Thus, the interpretation of Newton's Second Law within the rotating, non-inertial reference frame can be found by rearranging Eq. (4-45) (Török, 2000; Mooij, 1997).

$$m\frac{\mathrm{d}\mathbf{V}_R}{\mathrm{d}t} = \mathbf{F}_R - 2m\boldsymbol{\omega}_R \times \mathbf{V}_R - m\boldsymbol{\omega}_R \times (\boldsymbol{\omega}_R \times \mathbf{r}_{cm}) \quad (4\text{-}47)$$

Equations (4-46) and (4-47) form the basis for the determination of position and velocity of a vehicle within the rotating reference frame. At this point it cannot be assumed that the spacecraft is moving within a non-rotating reference frame, because at the beginning of the descent, the orbital velocity is high (about 115 ms^{-1}) and the distance between the origin of the rotating reference frame and the c.o.m. is large (about 352 km). In this way, the last two terms of Eq. (4-47) are not negligible, despite Enceladus' small angular velocity of 5.30733 $\times 10^{-5}\mathrm{rads}^{-1}$.

The results of Eq. (4-47) will be expressed with respect to a Cartesian coordinate system, because it was decided in Section 4-2-5 to use a Cartesian state vector. The position of the lander thus given by the vector $\mathbf{r}_{cm} = [x_R, y_R, z_R]^T$, and the velocity by the vector $\mathbf{V}_R = [\dot{x}_R, \dot{y}_R, \dot{z}_R]$.

The dynamic equations of translational motion then follow from Eq. (4-47) (Mooij, 1997)

$$\frac{\mathrm{d}\mathbf{V}_R}{\mathrm{d}t} = \begin{pmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{z}_R \end{pmatrix} = \begin{pmatrix} \omega_{cb}^2 x_R + 2\omega_{cb}\dot{y}_R \\ \omega_{cb}^2 y_R - 2\omega_{cb}\dot{x}_R \\ 0 \end{pmatrix} + \frac{1}{m}\left[\mathbf{F}_{G,R} + \mathbf{F}_{T,R}\right], \quad (4\text{-}48)$$

where

$$\mathbf{F}_{G,R} + \mathbf{F}_{T,R} = \mathbf{F}_{G,R} + \mathbf{T}_{R \leftarrow B}\mathbf{F}_{T,B} \quad (4\text{-}49)$$

making use of the transformation matrices of Section 4-3-2 and the force definitions of Section 4-4-1.

It is in the end decided to use the equations of motion with respect to the I-frame (Eq. 4-32) instead the expressions with respect to the R-frame (Eq. 4-48), because the code implementation proved to be more straightforward for the former case. The position, velocity and attitude data with respect to R-frame can be obtained using transformation matrices. In the end, all simulation results will be expressed with respect to the S-frame, so there is in fact no reason to perform the state integration in the R-frame.

4-4-3 Equations of Rotational Motion

The total moment $\mathbf{M}_I$ acting on a solid body with respect to the origin of the inertial reference frame is the sum of all cross products of the forces $\mathbf{F}_i$ and the location $\mathbf{r}_i$ of their according point of attack. This can be written as

$$\mathbf{M}_I = \sum_{i=1}^{N} \mathbf{r}_i \times \mathbf{F}_i = \int_m \mathbf{r} \times \frac{\mathrm{d}^2\mathbf{r}}{\mathrm{d}t^2} \mathrm{d}m \tag{4-50}$$

Similarly, the angular momentum $\mathbf{B}_I$ of the same solid body is equal to the sum of all cross products of $\mathbf{r}_i$ and the according linear momentum $\mathbf{J}_i$ (Mooij, 1997):

$$\mathbf{B}_I = \sum_{i=1}^{N} \mathbf{r}_i \times \mathbf{J}_i = \int_m \mathbf{r} \times \frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t} \mathrm{d}m \tag{4-51}$$

It can be inferred from the two equations before, that $\mathbf{M}_I$ is the time derivative of $\mathbf{B}_I$. The relation for the angular momentum as given in Eq. (4-51) can also be expressed with respect to the vehicle's c.o.m.. Figure 4-14 shows a situation where origin of the body reference frame coincides with the c.o.m. of the vehicle and rotates with the body with an rotational velocity of $\boldsymbol{\omega}$ with respect to the inertial reference frame. Again, the position of a mass element dm is $\mathbf{r}$ in the inertial reference frame and $\tilde{\mathbf{r}}$ in the body reference frame.

Position $\mathbf{r}$ is then replaced by sum $\mathbf{r}_{cm} + \tilde{\mathbf{r}}$ from Eq. (4-35). After executing the time derivate, Eq. (4-51) can be written as (Mooij, 1997)

$$\begin{aligned} \mathbf{B}_I &= \int_m (\mathbf{r}_{cm} + \tilde{\mathbf{r}}) \times \left(\frac{\mathrm{d}\mathbf{r}_{cm}}{\mathrm{d}t} + \boldsymbol{\omega} \times \tilde{\mathbf{r}} \right) \mathrm{d}m \\ &= m\mathbf{r}_{cm} \times \frac{\mathrm{d}\mathbf{r}_{cm}}{\mathrm{d}t} + \int_m \tilde{\mathbf{r}} \times (\boldsymbol{\omega} \times \tilde{\mathbf{r}}) \, \mathrm{d}m \end{aligned} \tag{4-52}$$

where $m\mathbf{r}_{cm} \times \frac{d\mathbf{r}_{cm}}{dt}$ is the angular momentum of the c.o.m. in the inertial reference frame and $\int_m \tilde{\mathbf{r}} \times (\boldsymbol{\omega} \times \tilde{\mathbf{r}}) \, dm$ is the angular momentum of all mass elements with respect to the vehicle's c.o.m.. Consequently (Mooij, 1997),

$$\mathbf{B}_{cm} = \int_m \tilde{\mathbf{r}} \times (\boldsymbol{\omega} \times \tilde{\mathbf{r}}) \, \mathrm{d}m \tag{4-53}$$

According to the conventions introduced in Section 4-2-1, $\tilde{\mathbf{r}}$ and $\boldsymbol{\omega}$ consist of the components $[x, y, z]$ and $[p, q, r]$, respectively. Working out the cross products of Eq. (4-53) (Mooij, 1997),

$$B_x = \int_m \left[p\left(y^2 + z^2\right) - qxy - rxz \right] \mathrm{d}m$$

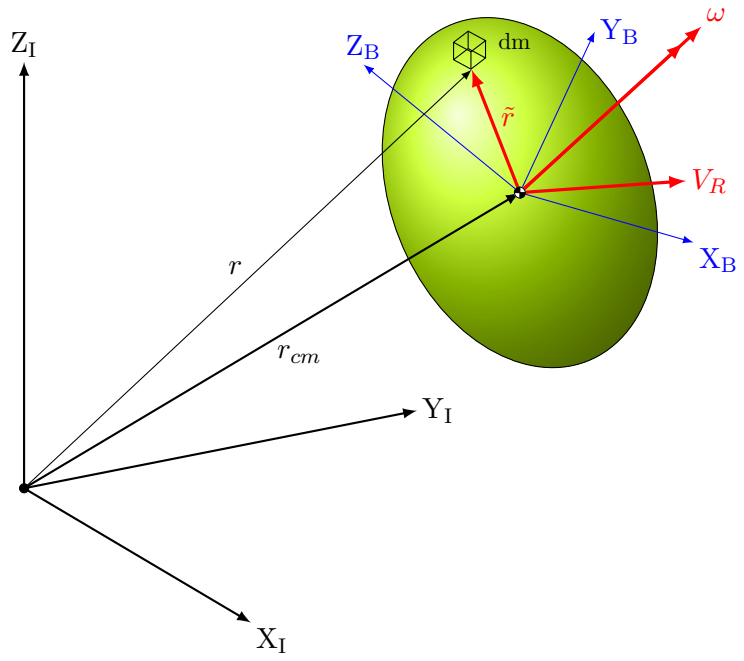


Figure 4-14: The body reference frame of a solid body rotates within the inertial reference frame

$$B_y = \int_m \left[-pyx + q\left(x^2 + z^2\right) - ryz\right] \mathrm{d}m \tag{4-54}$$
$$B_z = \int_m \left[-pzx - qzy + r\left(x^2 + y^2\right)\right] \mathrm{d}m$$

The integral groups represent the so-called *mass moments of inertia* about the principal coordinate axes, and the diagonal *products of inertia*, defined as (Török, 2000, p. 202)

$$\begin{aligned} I_{xx} &= \int_m \left(y^2 + z^2\right) \mathrm{d}m & I_{yy} &= \int_m \left(x^2 + z^2\right) \mathrm{d}m & I_{yz} = I_{zy} &= \int_m yz \, \mathrm{d}m \\ I_{xy} = I_{yx} &= \int_m xy \, \mathrm{d}m & I_{xz} = I_{zx} &= \int_m xz \, \mathrm{d}m & I_{zz} &= \int_m \left(x^2 + y^2\right) \mathrm{d}m \end{aligned} \tag{4-55}$$

The final expression for the angular momentum with respect to the center of mass with varying mass properties then follows from inserting the above relations in Eq. (4-54) and rewriting Eq. (4-53) to Mooij, 1997

$$\mathbf{B}_{cm} = \mathbf{I} \cdot \boldsymbol{\omega} = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix} \cdot \boldsymbol{\omega} \tag{4-56}$$

where $\mathbf{I}$ is the so-called *inertial tensor*.

The derivation of the rotational equations of motion, however, requires an expression for the spacecraft's momentum around its own c.o.m.. A similar situation as shown in Fig. 4-12 will

be used. If the force $\mathbf{F}_I$ is acting on a point located at $\mathbf{r}$ with respect to the inertial reference frame or at $\tilde{\mathbf{r}}$ with respect to the body's c.o.m., the vehicle's moment $\mathbf{M}_{cm}$ about the c.o.m. is (Mooij, 1997)

$$\mathbf{M}_{cm} = \int_m \tilde{\mathbf{r}} \times \frac{\mathrm{d}^2\mathbf{r}}{\mathrm{d}t^2}\mathrm{d}m \tag{4-57}$$

or, using the general expression for $\mathbf{r}$ given in Eq. (4-32),

$$\begin{aligned}\mathbf{M}_{cm} = \int_m \tilde{\mathbf{r}} \times \left(\frac{\mathrm{d}\boldsymbol{\omega}}{\mathrm{d}t} \times \tilde{\mathbf{r}}\right) \mathrm{d}m + \int_m \tilde{\mathbf{r}} \times \left[\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \tilde{\mathbf{r}})\right] \mathrm{d}m \\ + 2\int_m \tilde{\mathbf{r}} \times \left(\tilde{\boldsymbol{r}} \times \frac{\delta\tilde{\mathbf{r}}}{\delta t}\right) \mathrm{d}m + \int_m \tilde{\mathbf{r}} \times \frac{\delta^2\tilde{\mathbf{r}}}{\delta t^2}\mathrm{d}m\end{aligned} \tag{4-58}$$

Equation (4-58) thus describes the moment of a body with varying mass properties around its center of mass. Similar to the naming of the elements of Eq. (4-32), the terms $-2\int\tilde{\mathbf{r}} \times \left(\tilde{\boldsymbol{r}} \times \frac{\delta\tilde{\mathbf{r}}}{\delta t}\right) \mathrm{d}m$ and $-\int\tilde{\mathbf{r}} \times \frac{\delta^2\tilde{\mathbf{r}}}{\delta t^2}\mathrm{d}m$ are called the *Coriolis moment* $\mathbf{M}_C$ and *relative moment* $\mathbf{M}_{rel}$, respectively, as a consequence of the variable mass properties. The sum of the remaining two terms is the apparent moment. Equation (4-58) can be rearranged to (compare with the form of $\tilde{\mathbf{F}}_I$ in Eq. (4-37))

$$\tilde{\mathbf{M}}_{cm} = \mathbf{M}_{cm} + \mathbf{M}_C + \mathbf{M}_{rel} = \int_m \tilde{\mathbf{r}} \times \left(\frac{\mathrm{d}\boldsymbol{\omega}}{\mathrm{d}t} \times \tilde{\mathbf{r}}\right) \mathrm{d}m \;\; + \int_m \tilde{\mathbf{r}} \times \left[\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \tilde{\mathbf{r}})\right] \mathrm{d}m \tag{4-59}$$

where the left-hand side summarizes the external moments due to the gravitation ($\mathbf{M}_{cm}$) and due to the vehicle's thrusters ($\mathbf{M}_C, \mathbf{M}_{rel}$). Similar to the alternate expressions for $\mathbf{F}_C$ and $\mathbf{F}_{rel}$ derived in the previous section, $\mathbf{M}_C$ and $\mathbf{M}_{rel}$ can be rewritten in terms of propulsion parameters. For the Coriolis moment, the term $\tilde{\mathbf{r}}\left(\boldsymbol{\omega} \times \frac{\delta\tilde{\mathbf{r}}}{\delta t}\right)$ can be substituted by an expression that follows from writing out the derivate $\frac{\delta}{\delta t}\left[\tilde{\mathbf{r}} \times (\boldsymbol{\omega} \times \tilde{\mathbf{r}})\right]$ and applying again Reynold's Transport Theorem (Eq. (4-38)). Assuming that the exhaust area A_e is much smaller than the lengthwise dimension of the spacecraft (Mooij, 1997),

$$\mathbf{M}_C = -\frac{\delta\mathbf{I}}{\delta t} \cdot \boldsymbol{\omega} - \dot{m}_e\mathbf{r}_e \times (\boldsymbol{\omega} \times \mathbf{r}_e)\,. \tag{4-60}$$

Similar to this derivation and the derivation of $\mathbf{F}_{rel}$ in Eq. (4-42), the relative moment can be rewritten as (Mooij, 1997)

$$\mathbf{M}_{rel} = -\dot{m}_e\mathbf{r}_e \times \overline{\mathbf{V}}_e. \tag{4-61}$$

The necessary relations for the derivation of the rotational equations of motion have now been established.

As a consequence of the relation $\mathbf{M}_{cm} = \frac{\mathrm{d}\mathbf{M}_{cm}}{\mathrm{d}t}$, it is possible to find a simple, alternative form of the expression for $\mathbf{M}_{cm}$ in case the body may be considered rigid $\left(\frac{\delta\mathbf{I}}{\delta t} = 0\right)$ or approximated using the Principle of Solidification (see discussion of Eq. (4-37)). These so-called *Euler's equations* of a rotating body are written as:

$$\begin{aligned}\mathbf{M}_{cm} &= \frac{\mathrm{d}\mathbf{B}_{cm}}{\mathrm{d}t} = \frac{\delta\mathbf{B}_{cm}}{\delta t} + \boldsymbol{\omega} \times \mathbf{B}_{cm} = \frac{\delta\mathbf{I}}{\delta t} \cdot \boldsymbol{\omega} + \mathbf{I} \cdot \frac{\delta\boldsymbol{\omega}}{\delta t} + \boldsymbol{\omega} \times \mathbf{B}_{cm} \\ &= \mathbf{I} \cdot \dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{I} \cdot \boldsymbol{\omega}\end{aligned} \tag{4-62}$$

The dynamic equations of rotational motion follow from Eq. (4-62) and (4-59) (Mooij, 1997):

$$\dot{\boldsymbol{\omega}} = \mathbf{I}^{-1}\left(\tilde{\mathbf{M}}_{cm} - \boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega}\right) \tag{4-63}$$

$\tilde{\mathbf{M}}_{cm}$ is given by Eq. (4-59), or by the sum of $\mathbf{M}_C$ and $\mathbf{M}_{rel}$ (Eqs. (4-60) and (4-61)), as the gravitational moment is negligible.

It was decided in Section 4-2-5 to use quaternions to describe the attitudes. The attitude of the lander, or the orientation of the body reference frame, with respect to the I-frame is in this way given by the quaternion $\mathbf{q}_{I,B} = (q_0, q_1, q_2, q_3)^T$. The attitude time derivative $\dot{\bar{\mathbf{q}}}_{I,B}$ then can be determined with (Davailus and Newman, 2005)

$$\dot{\bar{\mathbf{q}}}_{I,B} = \frac{1}{2}\begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix}\begin{pmatrix} p \\ q \\ r \end{pmatrix} = \frac{1}{2}\boldsymbol{\Omega}_{\bar{q}}\boldsymbol{\omega} \tag{4-64}$$

or similarly,

$$\dot{\bar{\mathbf{q}}}_{I,B} = \frac{1}{2}\begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix}\begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} = \frac{1}{2}\boldsymbol{\Omega}_{\omega}\mathbf{q} \tag{4-65}$$

The attitude of the lander thus can be described in terms of a quaternion, which changes over time due to the spacecraft's angular rotation vector $\boldsymbol{\omega}$, whose values in turn change with the angular acceleration as given in Eq. (4-63). The Euler attitude representation for any quaternion may be calculated with the relations given in Section 4-2-7; this conversion is necessary for postprocessing. For a the practical software implementation, Eq. (4-64) is selected and modified to

$$\dot{\bar{\mathbf{q}}}_{I,B} = \frac{1}{2}\boldsymbol{\Omega}_{\bar{q}}\boldsymbol{\omega} + K_q\left[1 - \left(q_0^2 + q_1^2 + q_2^2 + q_3^2\right)\right]\bar{\mathbf{q}}, \tag{4-66}$$

which adds an additional term to correct for the (small) numerical errors that occur during the integration process and cause the norm of the attitude quaternion to divert from the nominal value of 1 (see Eq. (4-2)). This problem arises only for the quaternion model, as a quaternion - unlike Euler angles - uses four elements to describe an orientation with only three degrees of freedom. Consequently, the fourth constraint, the norm, must be checked in order to avoid invalid attitude quaternions. It is also possible to normalize the quaternion every integration step, but this might introduce even more errors due to over-corrections. By choosing an appropriate value of K_q, the quaternion norm is slowly driven to 1. This method is also used in the `MATLAB` Aerospace Toolbox, see Mathworks (2014). Tests with Enceladus lander simulator indicated, that $K_q = 1.5$ delivers satisfying results.

A modified version of Eq. (4-64) will be used in the lander's navigation system: in the extended Kalman filter, the derivative of the quaternion's scalar part q_0 is disregarded, because the filter will try to distribute the estimation corrections over all quaternion elements, which can lead to invalid attitude quaternions. The navigation filter attitude time derivatives are discussed in more detail in Section 6-1-5.

4-5 Numerical Integration

Numerical integration techniques are used to solve differential equations that have no or only a very complex analytical solution. The equations of motion derived in Chapter 4-4, for example, require amongst others the input of a thrust force vector and of the vehicle's angular velocity. However, the propulsive force of any activated thruster will always show fluctuations, and the angular velocity is given in discrete values based on the acquisition rate of the attitude sensors. It is not possible to find a closed analytical solution for this problem.

Numerical procedures have two fundamental error sources (Boyce and DiPrima, 2005): The *truncation error* is the actual difference between the true solution and the numerical approximation. Sometimes the truncation error is subdivided into the *local truncation error* and the *global truncation error*, which refer to the error of a single integration step and to the cumulative error of all steps, respectively. The *round-off error* is a consequence of the finite number of digits used during the computation. The truncation error can be reduced by reducing the step-size, but this in turn will increase the round-off error. The determination of the best step-size(s) is part of the solution process.

There several numerical integration methods, but this section will focus in particular on the Runge-Kutta (RK) class of methods. This technique is applicable to many problem and offers a high efficiency and accuracy. Simple methods, such as the *Euler method* or the *Trapezoidal rule*, are disregarded, because their truncation error is orders of magnitude higher than the higher-order RK methods (Boyce and DiPrima, 2005).

The ultimate goal is to find a numerical solution for the initial value problem

$$\mathbf{y}' = \mathbf{f}\left(t,\ \mathbf{y}\right), \qquad \mathbf{y}\left(t_0\right) = \mathbf{y}_0, \qquad t \geqslant t_0 \tag{4-67}$$

where $\mathbf{f}$ is a known function depending on the current function value $\mathbf{y}$ and time t, and $\mathbf{y_0}$ is a given initial value at the starting time t_0. The general Runge-Kutta formula of order v with a integration step-size h is (Iserles, 1996)

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h \sum_{j=1}^{v} b_j \mathbf{f}\left(t_n + c_j h,\ \ \mathbf{y}\left(t_n + c_j h\right)\right) \tag{4-68}$$

where b_j and c_j are the so-called RK weights and nodes, respectively. In principle, $\mathbf{y}_{n+1}$ can be expressed as a linear combination of the lower-order numerical systems, indicated with $\boldsymbol{\xi}_{1,2,\ldots,v}$ (Iserles, 1996):

$$\begin{aligned}
\boldsymbol{\xi}_1 &= \mathbf{y}_n \\
\boldsymbol{\xi}_2 &= \mathbf{y}_n + h a_{2,1} \mathbf{f}\left(t_n,\ \boldsymbol{\xi}_1\right) \\
\boldsymbol{\xi}_3 &= \mathbf{y}_n + h a_{3,1} \mathbf{f}\left(t_n,\ \boldsymbol{\xi}_1\right) + h a_{3,2} \mathbf{f}\left(t_n + c_2 h,\ \boldsymbol{\xi}_2\right) \\
&\vdots \\
\boldsymbol{\xi}_v &= \mathbf{y}_n + h \sum_{i=1}^{v-1} a_{v,i} \mathbf{f}\left(t_n + c_i h,\ \boldsymbol{\xi}_i\right)
\end{aligned} \tag{4-69}$$

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h \sum_{j=1}^{v-1} b_j \mathbf{f}\left(t_n + c_j h,\ \boldsymbol{\xi}_j\right) \tag{4-70}$$

Here, the factors a are the elements of the RK matrix, defined as $A = (a_{j,i})_{j,i=1,2,\ldots,\upsilon}$, with missing elements defined to be zero. The choice of the parameters b, c and A determines the weighting of each calculation stage within one RK step and depend on the application. A common way to display these elements is the RK Tableau, defined as

$$\begin{array}{c|c} \mathbf{c} & A \\ \hline & \mathbf{b^T} \end{array} \tag{4-71}$$

with $\mathbf{b} = [b_1, b_2, \ldots, b_\upsilon]^T$ and $\mathbf{c} = [c_1, c_2, \ldots, c_\upsilon]^T$. The classic fourth-order four-stage Runge-Kutta (RK4) formula can be written as

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ 1 & 0 & 0 & 1 & \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array} \tag{4-72}$$

Applying the above Tableau to Eq. (4-70) gives relatively simple representation (Boyce and DiPrima, 2005)

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h\left(\frac{\mathbf{k}_{n1} + 2\mathbf{k}_{n2} + 2\mathbf{k}_{n3} + \mathbf{k}_{n4}}{6}\right) \tag{4-73}$$

with the parameters

$$\begin{aligned} \mathbf{k}_{n1} &= \mathbf{f}\left(t_n,\ \mathbf{y}_n\right) \\ \mathbf{k}_{n2} &= \mathbf{f}\left(t_n + \frac{1}{2}h,\ \ \mathbf{y}_n + \frac{1}{2}h\mathbf{k}_{n1}\right) \\ \mathbf{k}_{n3} &= \mathbf{f}\left(t_n + \frac{1}{2}h,\ \ \mathbf{y}_n + \frac{1}{2}h\mathbf{k}_{n2}\right) \\ \mathbf{k}_{n4} &= \mathbf{f}\left(t_n + h,\ \mathbf{y}_n + h\mathbf{k}_{n3}\right). \end{aligned} \tag{4-74}$$

It can be shown that for this process the local truncation error is proportional to h^5 and the global truncation error does not exceed the product of a constant times h^4 (Boyce and DiPrima, 2005). In case calculations show that the errors or the computation time become unacceptable, a more complex integration method is required, such as multi-step processes or an adaptive stepsize methods. However, the RK4 technique is either the starting point for the derivation of these methods or can be used to determine the starting values. So-called embedded Runge-Kutta methods estimate the local error κ and compare its value with the predefined tolerance δ. In case κ is below or above a certain threshold, for example $\kappa \leqslant h\delta$, then the step-size is modified accordingly.

A change of h requires a new interpolation of the initial function, which results in a series of new calculations and a longer computation time. A good choice of RK elements, however, can significantly reduce this disadvantage, as shown by Fehlberg (1968). The Runge-Kutta-Fehlberg method (RKF45) is an adaptive step-size integration method, that uses a fifth-order RK method to determine the local error of a fourth-order RK method for an optimal step-size adaption. This adaption increases the accuracy and the efficiency in terms of computation time. Additionally, the step-size history indicates at which instances of time the original

function has higher and lower derivative values. The RKF45 requires the evaluation of the fourth-order RK method (modified from Mathews and Fink (1999))

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h\left(\frac{25}{216}\mathbf{k}_{n1} + \frac{1408}{2565}\mathbf{k}_{n3} + \frac{2197}{4101}\mathbf{k}_{n4} - \frac{1}{5}\mathbf{k}_{n5}\right) \tag{4-75}$$

and the fifth-order RK method (modified from Mathews and Fink (1999))

$$\mathbf{z}_{n+1} = \mathbf{y}_n + h\left(\frac{16}{135}\mathbf{k}_{n1} + \frac{6656}{12825}\mathbf{k}_{n3} + \frac{28561}{56430}\mathbf{k}_{n4} - \frac{9}{50}\mathbf{k}_{n5} + \frac{2}{55}\mathbf{k}_{n6}\right) \tag{4-76}$$

with the six $\mathbf{k}$-elements defined as (modified from Mathews and Fink (1999))

$$\begin{aligned}
\mathbf{k}_{n1} &= \mathbf{f}\left(t_n,\ \mathbf{y}_n\right) \\
\mathbf{k}_{n2} &= \mathbf{f}\left(t_n + \frac{1}{4}h,\ \mathbf{y}_n + \frac{1}{4}\mathbf{k}_{n1}\right) \\
\mathbf{k}_{n3} &= \mathbf{f}\left(t_n + \frac{3}{8}h,\ \mathbf{y}_n + \frac{3}{32} + \frac{9}{32}\mathbf{k}_{n2}\right) \\
\mathbf{k}_{n4} &= \mathbf{f}\left(t_n + \frac{12}{13}h,\ \mathbf{y}_n + \frac{1932}{2197}\mathbf{k}_{n1} - \frac{7200}{2197}\mathbf{k}_{n2} + \frac{7296}{2197}\mathbf{k}_{n3}\right) \\
\mathbf{k}_{n5} &= \mathbf{f}\left(t_n + h,\ \mathbf{y}_n + \frac{439}{216}\mathbf{k}_{n1} - 8\mathbf{k}_{n2} + \frac{3680}{513}\mathbf{k}_{n3} - \frac{845}{4104}\mathbf{k}_{n4}\right) \\
\mathbf{k}_{n6} &= \mathbf{f}\left(t_n + \frac{1}{2}h,\ \mathbf{y}_n - \frac{8}{27}\mathbf{k}_{n1} + 2\mathbf{k}_{n2} - \frac{3544}{2565}\mathbf{k}_{n3} + \frac{1859}{4104}\mathbf{k}_{n4} - \frac{11}{40}\mathbf{k}_{n5}\right).
\end{aligned} \tag{4-77}$$

The optimal step-size h then follows from (Mathews and Fink, 1999)

$$s = \left(\frac{\delta h}{2\|\mathbf{z}_{n+1} - \mathbf{y}_{n+1}\|}\right)^{\frac{1}{4}} \tag{4-78}$$

where δ is the predefined error tolerance. It is possible to increase the overall accuracy by choosing a pair of higher-order RK methods. A complete derivation and discussion of fifth-, sixth-, seventh-, and eighth-order Runge-Kutta formulas with step-size control, including the according RK Tableau's and a performance analysis can be found in Fehlberg (1968). Any integration step $\mathbf{y}_{n+1}$ of a Runge-Kutta and Runge-Kutta-Fehlberg method only uses the data from the preceding integration step, $\mathbf{y}_n$ - they belong to the so-called *one-step methods*. There exist numerical integration methods, that use the data points $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n$ to determine the next step $\mathbf{y}_{n+1}$. The *Adams-Bashforth method*, the *Adams-Moulton method* and the *predicor-corrector method* are examples of such *multi-step methods*. Multi-step methods are in general considerably faster than single-step methods for a given step-size h, but – depending on the application – less accurate (Boyce and DiPrima, 2005). Tests with an orbit integration of a LEO satellite show that multi-step methods are more stable and better preserve the orbit geometry, but require a much lower step-size for a given local error value (Es-hagh, 2005). Es-hagh (2005) recommends to use a RKF method for high-resolution integration, thus integrations with small step-sizes. The velocity and attitude of the Enceladus lander will change rapidly over a relatively short time span, in particular during the touchdown and repositioning process, which involve many correction maneuvers. Consequently, it is better to use an adaptive single-step method to integrate the state of the Silenus lander. The RKF45 method is the first choice due to its simplicity and lower computation time.

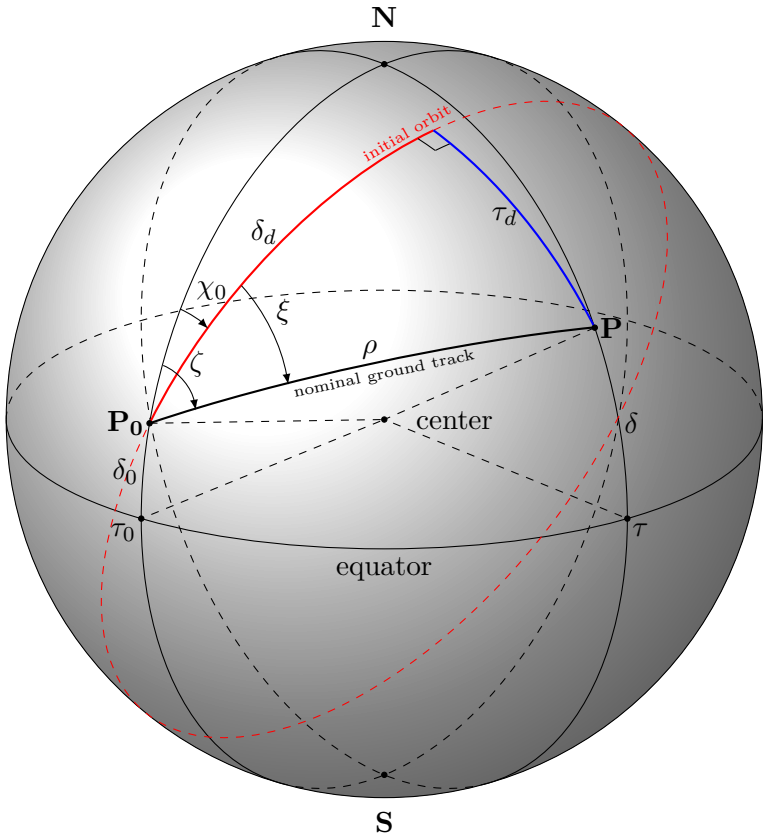


Figure 4-15: Geometry of cross range and downrange problem

4-6 Cross-range and Downrange

The landing accuracy of the vehicle is measured in terms of absolute distance from a predefined target spot. In order to estimate the performance of the guidance system, is useful to extract the current deviation from a reference trajectory, in particular, from a trajectory between an initial point and the target. The *cross range* and the *downrange* are defined as the great circle segments perpendicular and parallel, respectively, to the initial great circle. For atmospheric re-entry control – see for example Cavallo and Ferrara (1996) and Belló-Mora and Baeza-Martin (1995) – this initial great circle is commonly determined by the spacecraft's heading angle χ_0 at the longitude τ_0 and initial latitude δ_0.

The geometry of the cross range and downrange problem is shown in Fig. 4-15. All depicted arcs are segments of so-called *great circles*, which are the largest possible circles on the surface of a sphere. The reference sphere itself is a unit sphere, so the length of the arcs can be expressed in terms radians. The distance in meters between two points on a sphere is simply the product of the arc length in radians and the radius of the sphere. For an extensive discussion on spherical geometry, the reader is referred to Wertz and Larson (1999). The derivation of the equations for the downrange δ_d and the cross range τ_d of a point $\mathbf{P}(\delta, \tau)$, given the initial point $\mathbf{P_0}(\delta_0, \tau_0)$ and the initial heading angle χ_0, involves the derivation of some auxiliary arcs and spherical angles; these are all depicted in Fig. 4-15 and follow to some extent the conventions of Cavallo and Ferrara (1996). Note that the following equations are

only valid for longitude and latitude values in the intervals $[0, 2\pi]$ and $[-\frac{\pi}{2}, \frac{\pi}{2}]$, respectively (see definition of spherical coordinate system, Fig. 4-8). Other range definitions might require additional expressions to ensure that the correct spherical quadrants are used.

The great circle arc ρ connecting $\mathbf{P_0}$ and $\mathbf{P}$ – or, the distance between the points on the surface of the sphere – is the angle between the vectors $\mathbf{p_0}$ and $\mathbf{p}$ pointing to $\mathbf{P_0}(\delta_0, \tau_0)$ and $\mathbf{P}(\delta, \tau)$, respectively. The Cartesian representations of the spherical positions according to the coordinate system transformations given in Eq. (4-5b)) are

$$\mathbf{p_0} = \begin{bmatrix} \cos\delta_0 \cos\tau_0 \\ \cos\delta_0 \sin\tau_0 \\ \sin\delta_0 \end{bmatrix}, \quad \mathbf{p} = \begin{bmatrix} \cos\delta \cos\tau \\ \cos\delta \sin\tau \\ \sin\delta \end{bmatrix}. \tag{4-79}$$

The angle ρ between the unit vectors $\mathbf{p_0}$ and $\mathbf{p}$ follows from the dot product:

$$\begin{aligned} \cos\rho = \mathbf{p_0} \cdot \mathbf{p} &= \cos\delta_0 \cos\tau_0 \cos\delta \cos\tau + \cos\delta_0 \sin\tau_0 \cos\delta \sin\tau + \sin\delta_0 \sin\delta \\ &= \cos\delta_0 \cos\delta \left[\cos\tau_0 \cos\tau + \sin\tau_0 \sin\tau\right] + \sin\delta_0 \sin\delta \\ &= \cos\delta_0 \cos\delta \cos\left(\tau - \tau_0\right) + \sin\delta_0 \sin\delta \end{aligned} \tag{4-80}$$

The angle ζ between the meridian passing through $\mathbf{P_0}$ and ρ can be found with the Law of Cosines for Sides, with ρ and $\frac{\pi}{2} - \delta$ being the known side lengths:

$$\cos\left(\frac{\pi}{2} - \delta\right) = \cos\rho \cos\left(\frac{\pi}{2} - \delta_0\right) + \sin\rho \sin\left(\frac{\pi}{2} - \delta_0\right) \cos\zeta \tag{4-81}$$

This gives, after some rearrangements,

$$\cos\zeta = \frac{\sin\delta - \cos\rho \sin\delta_0}{\sin\rho \cos\delta_0}. \tag{4-82}$$

The angle ξ between the initial great circle and the great circle passing through $\mathbf{P_0}$ and $\mathbf{P}$ then is

$$\xi = \zeta - \chi_0. \tag{4-83}$$

The cross range τ_d, the downrange δ_d and the auxiliary arc ρ form a spherical triangle with a right angle between δ_d and τ_d. Using the relations for right spherical triangles (see, for example, Wertz and Larson (1999, p. 777)), the final expressions for the cross-range and downrange of point $\mathbf{P}$ given $\mathbf{P_0}$ and χ_0 are:

$$\tau_d = \arcsin\left[\sin\rho \sin\xi\right] \tag{4-84}$$

$$\delta_d = \arccos\left[\frac{\cos\rho}{\cos\tau_d}\right] \tag{4-85}$$

Equations (4-84) and (4-85) correspond to the expressions given in (Cavallo and Ferrara, 1996). Depending on the choice of the initial conditions and $\mathbf{P}$, the cross range and downrange can be used for navigational purposes: If $\mathbf{P_0}$ and χ_0 are fixed at a point of time during the flight, δ_d and τ_d represent the lateral and traverse deviations from the initial trajectory; if $\mathbf{P}$ is chosen to be the target point, δ_d and τ_d represent the lateral and traverse distance between the target and the lander at position $\mathbf{P_0}$, flying in the direction χ_0. The latter case will be used as a measure for benchmarking the guidance system. In this situation, the actual distance along the moon's surface between the target point and the lander follows directly from Eq. (4-80).

4-7 Perturbing Forces

Perturbing forces refer to any forces, that disturb the ideal model of a spacecraft orbiting a central body with a point mass. The following sections discuss the effects of additional large point masses (section 4-7-1), and solar radiation pressure (section 4-7-2). The perturbations due to gravity field irregularities were already discussed in Section 4-4-1.

4-7-1 Third Body Perturbation

A spacecraft orbiting Enceladus and even a lander during the descent and repositioning process will to some extend experience the gravitational fields the other celestial bodies as perturbing forces. It is necessary to implement third body perturbations in case the disturbing accelerations result in a notable deviation from the spacecraft's ideal course.

As the planetocentric reference frame is accelerating and thus not an inertial reference frame, the perturbing acceleration must be expressed relative to the main acceleration a_m of the central body Enceladus. The geometry of this problem is depicted in Fig. 4-16, where body k is the central body, body i the orbiting body, and d the disturbing body.

The mass of the lander is negligible with respect to the mass of Enceladus, thus the equations of motion can be written as (Vallado and McClain, 2001)

$$\ddot{\overline{r}}_{ki} = -G\frac{\mu_k}{{r_{ki}}^3}\overline{r}_{ki} + \mu_d\left(\frac{\overline{r}_{kj} - \overline{r}_{ki}}{r_{ij_3}} - \frac{\overline{r}_{kj}}{{r_{kj}}^3}\right), \tag{4-86}$$

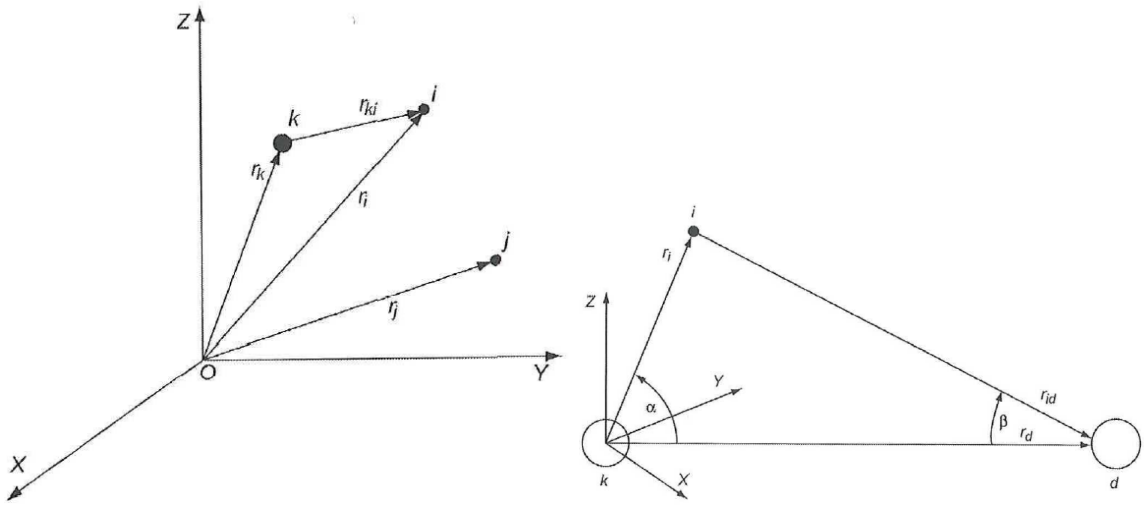
Titan as the second largest moon of the solar system has by far the highest mass - two orders of magnitude higher then moon next in size, Rhea. The largest perturbing force, however, will emanate from Saturn due to its immense mass and the relatively low orbit of Enceladus ($a = 3.95R_{Sat}$). Table 4-2 lists the gravitational parameters, the masses and values of the semi-major axes of the Sun, Jupiter, Uranus, Saturn and Saturn's nine major moons, including Enceladus.

The accelerations that act on a satellite orbiting Enceladus, which in turn orbits Saturn together with other moons, can be described with a so-called many-body problem.

In a system as shown in Fig. 4-16a with a total number of bodies n, the motion of a body i with respect to body k can be derived with (Wakker, 2010)

$$\ddot{\overline{r}}_{ki} = -G\frac{m_i + m_k}{{r_{ki}}^3}\overline{r}_{ki} + G\sum_{j\neq i,k}^{n} m_j\left(\frac{\overline{r}_{kj} - \overline{r}_{ki}}{r_{ij_3}} - \frac{\overline{r}_{kj}}{{r_{kj}}^3}\right), \tag{4-87}$$

where $\overline{r}_i$ is the position vector of body i, $\overline{r}_j$ the position vector of the disturbing body j, G is the gravitational constant and m is the mass of a body. The direction of a vector is indicated by the order of the indexes; for example $\overline{r}_{ki}$ is the vector from body k to body i. Equation (4-86) thus can be used to calculate the motion of body i in relation to a non-inertial reference frame with the origin k due to the gravitational forces between i, j and k. Its first term relates to an unperturbed orbit, the second describes the perturbations introduced by the other bodies.



(a) The many-body problem **(b)** Relative positions

Figure 4-16: The many-body problem and relative positions. **(a)** Geometry of the many-body problem with bodies k, i and j expressed in an inertial Cartesian reference frame with the origin O. **(b)** Positions of bodies i, j and k with respect to each other. Figures from Wakker (2010).

Body	GM [km^3s^{-2}]	M [10^{20} kg]	a [10^3km]	$(a_d/a_m)_{max}$ [-]
Sun	132.712×10^9	1989.1×10^7	-	5.46×10^{-10}
Jupiter	126.686×10^6	1868.6×10^4	778.57×10^3	5.46×10^{-12}
Uranus	5.794×10^6	86.8×10^4	2872.46×10^3	7.65×10^{-15}
Saturn	37.931×10^6	568.5×10^4	1433.53×10^3	3.41×10^{-3}
Titan	8978.14	1345.50	1221.83 ($20.27R_{Sat}$)	5.97×10^{-8}
Rhea	153.94	23.10	527.04 ($8.75R_{Sat}$)	1.28×10^{-8}
Iapetus	120.51	18.10	3561.30 ($59.09R_{Sat}$)	3.24×10^{-11}
Dione	73.11	11.00	377.40 ($6.26R_{Sat}$)	1.66×10^{-8}
Tethys	41.21	6.18	294.66 ($4.89R_{Sat}$)	1.96×10^{-8}
Enceladus	7.21	1.08	238.02 ($3.95R_{Sat}$)	-
Mimas	2.50	0.38	185.52 ($3.08R_{Sat}$)	4.81×10^{-9}

Table 4-2: The gravitational parameters GM, the masses M, the semi-major axes a with respect to the respective central body and the maximum ratio of disturbing acceleration to main acceleration for the Sun, Jupiter, Uranus, Saturn and Saturn's seven largest moons. The values of $(a_d/a_m)_{max}$ were calculated with Eq. (4-88), the values for GM, M and a were taken from NASA (2013)

The maximal relative perturbation can be expressed as a fraction of the main acceleration, using the fact that the maximal perturbation occurs when body i is exactly between the main body and the perturbing body (derived from equations by Wakker (2010)):

$$\left(\frac{a_d}{a_m}\right)_{\max} = \frac{m_d}{m_k}\left(\frac{r_{ki}}{r_{kd}}\right)^2 \left|\left[\left(\frac{1}{1-\frac{r_{ki}}{r_{kd}}}\right)^2 - 1\right]\right| \tag{4-88}$$

Here, the indexes d and m stand for the disturbing body and the main body, respectively. The geometry of the above equation is depicted in Fig. 4-16b. The values of the ratio $(\mathrm{a}_d/\mathrm{a}_m)_{\max}$ for different disturbing bodies, for a spacecraft orbiting Enceladus, are listed in Table 4-2. The following values for the variables in Eq. (4-88) have been used:

- Enceladus is the central body, thus $m_k = 1.08 \times 10^{20}$ kg
- m_d are the masses of the perturbing bodies
- r_{ki} is the distance between the center of gravity of Enceladus and the orbiting spacecraft, thus assuming an orbit of 100 km above the surface, $r_{ki} = R_{\text{enc}} + 100 = 352.1$ km
- r_{kd} is the minimal distance between Enceladus and the disturbing body, thus assuming circular orbits,
  - $r_{kd} = a_d$ for the saturnian moons,
  - $r_{kd} = a_{\text{enc}}$ for Saturn,
  - $r_{kd} = a_{\text{saturn}}$ for the Sun and
  - $r_{kd} = |a_{\text{saturn}} - a_d|$ for Jupiter and Uranus.

As can be seen, the relative perturbation of Saturn is several magnitudes larger than any other perturbation by induced by the other bodies. The disturbance is in fact so large, that it is not possible to bring a spacecraft in a stable polar orbit around Enceladus. Studies such as (Russell and Lara, 2009) indicate that the maximum inclination for a long-term stable orbit (180 days or more) is around 60°. Consequently, a polar orbit is only an option for short term observations and the initiation of the descent process. In case the landing area must be investigated during a longer period of time, the spacecraft must repeatedly correct its orbit, which also has a negative influence on the reliability. This aspect might become important for the design the GNC system.

4-7-2 Solar Radiation Pressure

Solar radiation pressure arises when photons hit a spacecraft. The effective impulse depends on the ratio between the reflected and the absorbed portion of the incoming light, denoted with the reflectivity coefficient ϵ: A perfect black body ($\epsilon = 0$) absorbs all incoming light and experiences only half of the pressure of an ideal reflector ($\epsilon = 1$). Each photon carries a particular amount of energy, so the magnitude of the solar radiation pressure depends on the amount incoming radiation energy per unit of area, measured in terms of the local solar flux Φ. The acceleration that a satellite with a cross-section A perpendicular to the incoming radiation and a mass m will experience is (modified from Montenbruck and Gill (2000))

$$a_{solar} = \frac{\Phi A}{cm}. \tag{4-89}$$

To estimate the order of magnitude of the solar radiation pressure on a object in the vicinity of Enceladus, it is assumed that the effective cross-sectional area of the spacecraft is perpendicular to the incoming solar radiation. Introducing the radiation pressure coefficient C_R,

$$C_R = 1 + \epsilon, \tag{4-90}$$

and applying the formula for surface of a sphere, Eq. (4-89) can be written as

$$a_{\text{solar}} = L_{\odot} C_R \frac{A_{\text{eff}}}{4\pi r^3 cm} \tag{4-91}$$

where $L_{\odot}$ is the solar luminosity (3.827×10^{26} $\text{Jm}^{-2}\text{s}^{-1}$, Pater and Lissauer (2001)) and r the distance between the Sun and the spacecraft. In case a relatively high radiation pressure coefficient of 1.9 is chosen, the perturbing acceleration a_{solar} for the lander with an effective cross section of about 1.04 m^2 (see Table 3-1) is 3.256×10^{-10} ms^{-2} at Saturn. The ratio $\frac{a_{\text{solar}}}{a_m}$ for an orbital height of 100 km then becomes 5.595×10^{-9}. This lower most of the discarded third body perturbations listed in Table 4-2. The solar radiation pressure is thus not included in the Enceladus Lander Simulator.

4-8 Simulator Configuration: Spacecraft Motion

As discussed in the Section 4-2, the basic lander state vector consists of the position $\mathbf{r}$, velocity $\mathbf{v}$, quaternion attitude $\mathbf{q}$ and rotational rate $\boldsymbol{\omega}$. The mass m is also added, because in particular the guidance system needs a current mass estimation to determine the appropriate acceleration level. The Enceladus Lander Simulator processes the estimated state $\mathbf{x}_{est}$ and the true state $\mathbf{x}_{true}$ separately, see Fig. 7-1 for the basic program overview. The simulator configuration parameters corresponding to the state variables, reference frames and perturbing forces are listed in Table 4-3. The mass related parameters can be found in Table 3-5. Just like the previous parameter lists, variable names are only listed once in the table if only their axis designation changes – in those cases, an indication of the number of excluded variable names is added in brackets.

The initial values for $\mathbf{x}_{true}$ for the nominal lander model with respect to the I-frame are configuration parameters in the lander configuration file (Appendix A-1) – $\mathbf{x}_{est}$ is equal to $\mathbf{x}_{true}$ for the this lander model. The attitude is defined in terms of Euler angles for a better insight, but translated internally their quaternion counterparts.

The initial values for $\mathbf{x}_{true}$ and $\mathbf{x}_{est}$, on the other hand, are generated during the Monte Carlo simulations using the nominal values combined with the position, velocity and quaternion standard deviations.

The user can chose between the basic RK4 integrator, or one of the three higher-order RKF techniques. The simulations in Chapter 8 use in general the RKF45 integrator. The RKF56 and RKF78 are part of the `C++` integrators toolbox, but they are not used in the simulations. For more information about the architectural design of the Enceladus Lander Simulator, see Chapter 7.

Variable Name	Comment
Initial State: descent	
`xPosition` (+2)	[m], true position in I-frame
`xVelocity` (+2)	[m/s], true velocity in I-frame
`eulerAnglePhi` (+2)	[°], ZYX-rotation I- to B-frame
`rotationalRateX` (+2)	[rad/s], ω_x
Initial State: repositioning	
`xPositionRepositioning` (+2)	[m], true position in I-frame
`xVelocityRepositioning` (+2)	[m/s], true velocity in I-frame
`eulerAnglePhiRepositioning` (+2)	[°], ZYX-rotation I- to B-frame
`rotationalRateXRepositioning` (+2)	[rad/s], ω_x
Target State: descent	
`xPositionTarget` (+2)	[m], true position in R-frame
`xVelocityTarget` (+2)	[m/s], true velocity in R-frame
`xAccelerationTarget` (+2)	[m/s^2], in R-frame
Reference Frames	
`rotatingReferenceFrame`	on/off, test mode
Integrator	
`integratorID`	RK4, RKF45, -56 or -78
`minimumStepSize`	[s], Δt_{min}
`maximumStepSize`	[s], Δt_{max}
`relativeErrorTolerance`	[-], δ_{rel} for Eq. (4-78)
`absoluteErrorTolerance`	[-], δ_{abs} for Eq. (4-78)
`initialStepSize`	[s], const. for RK4
Gravity Fields	
`centralGravityFieldEnceladus`	on/off
`centralGravityFieldSaturn`	on/off, third body perturbation
`J2GravityFieldSaturn`	on/off
`J2GravityFieldEnceladus`	on/off

Table 4-3: State vector and motion related program parameters

Chapter 5

Guidance and Control

This chapter presents the first part of the lander's flight software, namely the guidance and control subsystems. The discussion of the navigation and hazard avoidance subsystem is extensive and thus shifted to the next chapter.

Section 5-1 introduces concepts of gravity turn guidance, quadratic guidance, velocity nullifying guidance, hybrid-ballistic repositioning guidance and quadratic repositioning guidance. These guidance logics are all implemented in the Enceladus Lander Simulator; their performance is tested separately in the according sections of this chapter, and on a system level, in the actual descent and repositioning simulations in Chapter 8. The guidance phases, including the switch between the guidance logics, are discussed and visualized in Subsection 5-1-6.

The control system, consisting of a linear quaternion controller in combination with a tabulated thruster selection logic and a pulse-width-pulse-frequency modulator, is discussed in Section 5-2.

Each section closes with a summary of the simulator configuration parameters relating to the foregoing discussions.

5-1 Guidance

In recent years, the Moon received increasing scientific and even commercial interest. Several soft-landing missions are currently in the design phase, and the descent guidance is a critical part of each mission. Fortunately, the absence of an atmosphere greatly reduces the complexity of the guidance laws.

The main requirements for soft landing guidance are fuel efficiency, robustness, autonomy and real-time performance (Huang and Wang, 2007). The trajectory with the lowest propellant consumption follows from a global optimization process with fixed initial and terminal vehicle states and a given set of constraints. The robustness of this method, however, suffers in case of unexpected disturbances (Huang and Wang, 2007). In particular the gravity-turn steering – which has minimal losses, because the thrust vector is constantly opposite to the velocity

vector – cannot be used as guidance algorithm without modifications, as the final conditions are fully defined by the initial conditions (Sostaric and Rea, 2005). A sub-optimal method is *explicit guidance* that uses the current state of the vehicle to calculate the optimal steering commands to reach the (fixed) target. Explicit guidance has a higher robustness, especially in environments where not all disturbing accelerations are exactly known (Huang and Wang, 2007). If nominal guidance is complex even for the well-known Moon, then the descent to Enceladus should be largely based on explicit guidance schemes.

The guidance system must ensure that the vehicle reaches its specified target. It uses the current vehicle state to generate steering instruction for the control system, based on trajectory comparison, hazard control, ground commands and other inputs. A planetary lander commonly uses different sets of guidance laws for the various mission phases; this is discussed in Section 5-1-6. The actual guidance laws for the descent and repositioning process on Enceladus will be derived mainly from the lunar landing mission in section. Five main guidance schemes are considered in this chapter: the gravity turn in Section 5-1-1, the quadratic guidance law in Section 5-1-2, the velocity nullifying guidance law for the terminal-landing phase in Section 5-1-3, and the ballistic and quadratic guidance logics for repositioning in Section 5-1-4 and 5-1-5, respectively.

The surface topography of Enceladus imposes high requirements on the guidance system. Section 2-2 explained that the surface near the south polar region is relatively hilly, covered with ice chunks and fine fault lines. The distance between the elevations is between 20 m and 100 m, so the landing accuracy should be better than 10 m to safely land between two ice boulders. It is likely that the hazard avoidance is of particular importance during the approach phase of the descent process, because absolute navigation is very complicated. Once the lander is on the surface, however, it is possible to determine its exact position and orientation. The targets of the repositioning cycles will be selected by the ground control on Earth. In this case, the vehicle must be guided to a safe landing spot as close to the target area as possible (less than 100 m distance), while minimizing the touchdown velocity - the vertical and horizontal velocity at touchdown must be below 0.5 m/s for a landing (see Section 3-3).

The human spaceflight missions *Apollo* and *Constellation* had similar requirements, and they are an important source for information about the guidance logic and hazard avoidance system. The *Altair* lunar lander was the conceptual design of the Constellation landing module. The Altair guidance and in particular the hazard avoidance system was investigated to some extent, and the results are important for the Enceladus mission.

5-1-1 Gravity turn

A spacecraft following a gravity-turn descent trajectory applies thrust solely in the opposite direction of its velocity vector. The implementation in the on-board software is comparatively simple and aims at elimination of the rotational rates about the spacecraft's velocity vector (McInnes, 1996). The gravity turn method minimizes fuel losses and is near optimal in terms of fuel efficiency, but has the disadvantage, that the final conditions are fully defined by the initial, in-orbit conditions (Sostaric and Rea, 2005). A pinpoint landing using a pure gravity turn guidance requires the knowledge of all perturbing forces on the trajectory, and it requires the spacecraft to be at a specific initial state, using perfect control during the descent process.

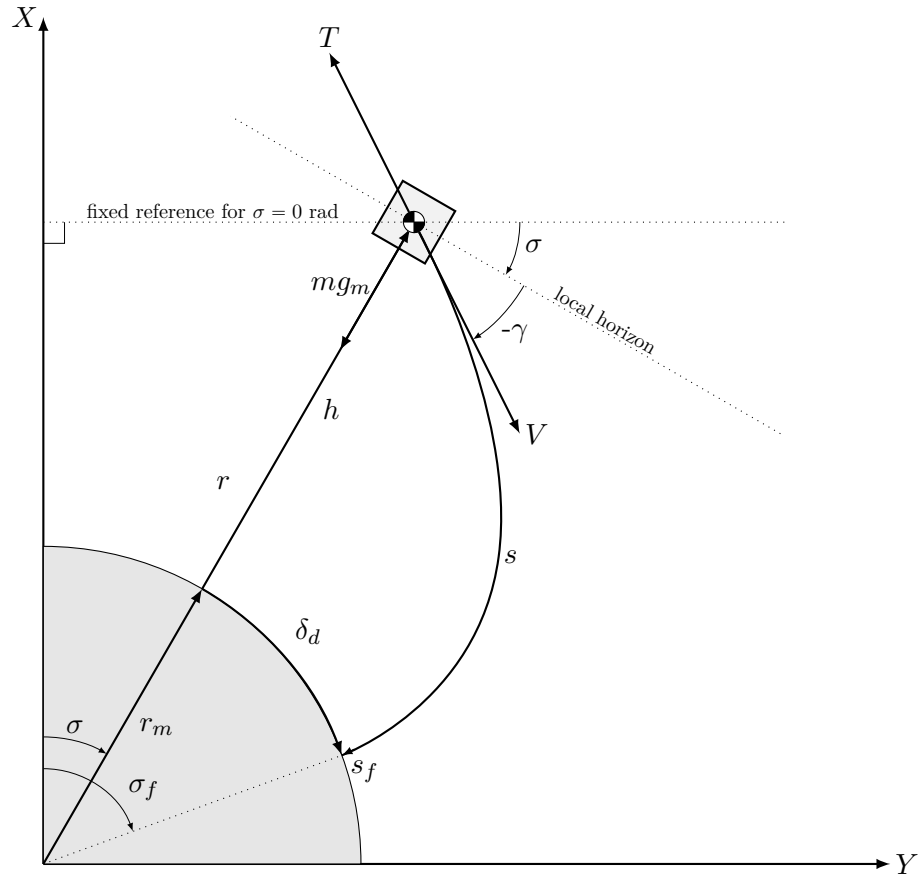


Figure 5-1: Geometry of the gravity turn guidance problem

As these requirements cannot be met to the full extent, the gravity turn guidance may be used for a fuel-efficient deceleration before the start of the terminal descent. This system has successfully been used on the Viking landers (Ingoldby, 1978). For the recent Phoenix lander, a gravity turn was even part of the terminal descent phase due to the low precision requirements (Guo et al., 2012). A different approach uses two feedback guidance loops: one loop based on a gravity turn, that aims at zero velocity at the end of the trajectory, and one loop aiming at minimizing the horizontal components of the landing error (Citron et al., 1964).

Although the concept of a gravity turn is simple, it is not possible to find a closed analytical expression for a suitable guidance logic without making several assumptions to simplify the problem. This limits the accuracy of the method, but it can be shown that the relative errors are in fact small as long as the lander's initial conditions are within certain boundaries. The gravity turn problem is divided into a spherical moon case for small flight path angles and high initial velocities, and a flat moon case for steep flight path angles or low initial velocities. The former case has a higher relevance for the Enceladus Lander, so it is discussed in more detail. The following two sections are based on Citron et al. (1964); other sources are used during the derivation process and for the attempt to increase the method's accuracy. The geometry of the gravity turn guidance problem is shown in Fig. 5-1.

Spherical Moon

The motion of a point mass m with respect to a (pseudo)inertial reference frame at a distance r from the central body's center of mass with a gravitational parameter μ can be described with

$$\ddot{\mathbf{r}} = -\frac{\mu}{r^3}\mathbf{r}. \tag{5-1}$$

The scalar product of Eq. (5-1) and $\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t}$ together with the vector product of the same equation with $\mathbf{r}$ lead to the cylindrical equations of motion for a spacecraft with thrust force T (Wakker, 2010):

$$\ddot{r} - r\dot{\sigma}^2 = -\frac{\mu}{r^2} + \frac{T}{m}\sin\delta \tag{5-2}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(r^2\dot{\sigma}\right) = \frac{T}{m}r\cos\delta \tag{5-3}$$

Here, σ is the polar angle of the spacecraft, and δ is the angle between the thrust vector $\mathbf{T}$ and the normal to $\mathbf{r}$ in the plane of motion, thus a combination of ϵ_T and ϕ_T as shown in Fig. 4-11. The thrust force of a spacecraft following a gravity turn trajectory is always directed in the opposite direction of the velocity vector $\mathbf{V}$. In this case, the absolute value of δ is equal to the absolute value of the flight path angle γ, which is defined as the negative value of the angle between the local horizon and $\mathbf{V}$, see Fig. 5-1. Both $\sin\delta$ and $\cos\delta$ may be expressed in terms of radial, tangential and total velocity:

$$\sin\delta = -\sin\gamma = -\frac{\dot{r}}{V} \tag{5-4}$$

$$\cos\delta = \cos -\gamma = \frac{r\dot{\sigma}}{V} \tag{5-5}$$

As the angular momentum per unit of mass is $\mathbf{r}\times\mathbf{V} = r^2\dot{\sigma}$ and the acceleration in the opposite direction of the current velocity vector is $a = \frac{T}{m}$, Eq. (5-1) becomes (Citron et al., 1964)

$$\ddot{r} - \frac{H^2}{r^3} = -\frac{\mu}{r^2} + a\frac{\dot{r}}{V} \tag{5-6}$$

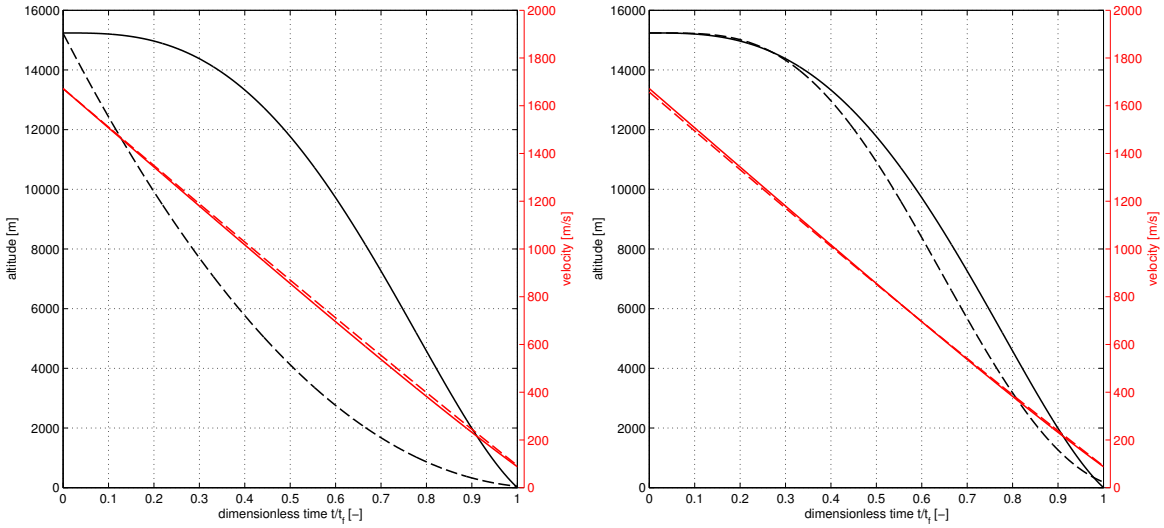
$$\dot{H} = -a\frac{H}{V} \tag{5-7}$$

The body's total energy E is an important auxiliary value for the derivation of the gravity turn guidance equations, because its initial and final value are known beforehand. The change of E as a function of path length s is simple, as a always points in the opposite direction of $\mathbf{V}$:

$$\frac{\mathrm{d}E}{\mathrm{d}s} = \frac{\mathrm{d}t}{\mathrm{d}s}\frac{\mathrm{d}E}{\mathrm{d}t} = \frac{1}{V}\left(-aV\right) = -a \tag{5-8}$$

Integrating Eq. (5-8) leads to (Citron et al., 1964)

$$E = -as + \frac{V_0^2}{2} - \frac{\mu}{r_0} \tag{5-9}$$



(a) $n = 1$ **(b)** $n = 4$

Figure 5-2: Estimated[1](dashed lines) and actual (solid lines) altitude and velocity for the gravity turn guidance for a body with the initial conditions $h_0 = 15240\text{m}$, $V_0 = V_c$ and $\gamma_0 = 0^\circ$ in combination with a constant acceleration and a fixed mass

where the last two terms represent the initial kinetic and potential energy, respectively (see Wakker (2010)). Solving Eq. (5-9) for V^2 after replacing E by the expression for the current kinetic and potential energy, and using the fact that $\frac{h}{r} \ll 1$ (Citron et al., 1964):

$$V^2 = -2as + V_0^2 + 2g_m\left(h_0 - h\right) \tag{5-10}$$

The accuracy of the gravity turn guidance depends on the chosen expression for h in the previous equation, because V^2 is required for the integration of the cylindrical equations of motion. Any expression that differs from the basic Eq. (Citron et al., 1964)

$$h = h_0\left(1 - \frac{s}{s_f}\right) \tag{5-11}$$

ultimately leads to very complex terms in the integrated equations of motion, that cannot be solved for the acceleration command a as a function of time or path length in a closed form. Note that the expression for the final – or total – path length s_f here is simply

$$s_f = \frac{V_0^2 + 2g_m h_0}{2a} \tag{5-12}$$

because the acceleration a for a gravity turn guidance only acts along s.

[1]The modification of Eqs. (5-11) and (5-14) for an order n is:

$$h = h_0\left[1 - \left(\frac{s}{s_f}\right)^n\right]$$

$$V^2 = V_0^2\left[1 - \frac{s}{s_f}\right] - 2g_m h_0 \frac{s}{s_f}\left[1 - \left(\frac{s}{s_f}\right)^{n-1}\right]$$

These equations are only used in Fig. 5-2 with $n = 4$ for demonstration purposes.

The accuracy of Eq. (5-11) depends on the lander's initial conditions h_0, γ_0 and V_0. In particular, steep descent trajectories with γ_0 close to -90° show much better agreements with the approximation for h. The simulation results shown in Fig. 5-2 indicate that in this case replacing $\frac{s}{s_f}$ by a higher-order expression such as $\left(\frac{s}{s_f}\right)^4$ improves the accuracy of the altitude estimation, but even this small modification causes an explosion of terms in the analytical solution. On the other hand, Citron et al. (1964) demonstrates that the relative downrange error is in fact low in case V_0 is larger than the free fall velocity

$$V_{\text{ff}} = g\sqrt{\frac{2h_0}{g}}. \tag{5-13}$$

The gravity-turn equations do not include disturbing accelerations, so the actual accuracy of the this guidance law is probably much lower than for the ideal case due to the comparatively large third-body perturbations from Saturn. This will be investigated during the actual simulation runs and is discussed in Chapter 8. At this point, it seems to be unlikely that the sole implementation of a better expression for h will increase the guidance accuracy. A simple alternative is to deviate from the constant acceleration requirement; this is discussed later in this section.

Substituting the functions for the estimated altitude in Eq. (5-11) and for the final path length in Eq. (5-12) into Eq. (5-10) gives

$$V^2 = V_0^2\left(1 - \frac{s}{s_f}\right) \tag{5-14}$$

This estimated velocity follows the actual value very closely, despite the limitations of Eq. (5-11). The remaining steps for the derivation of the gravity turn guidance are straightforward: The independent variable from the equations of motion (Eq. (5-7)) is changed from time to path length. With the simplifications $\frac{\mathrm{d}r}{\mathrm{d}s} \approx \left(\frac{\mathrm{d}r}{\mathrm{d}s}\right)_0$ and $r \approx r_m$, the rewritten cylindrical equations of motion can be integrated under the conditions that $r = r_0$ and $\frac{\mathrm{d}r}{\mathrm{d}s} = \sin\gamma_0$ when $s = 0$ (Citron et al., 1964):

$$\begin{aligned}\left(\frac{a}{g_m}\right)^2 + \sin\gamma_0\left[\frac{V_0^2}{2h_0 g_m} + 1\right]\frac{a}{g_m} - \\ \frac{\cos^2\gamma_0}{4V_0^2 h_0 g_m}\left[V_0^2 + 2g_m h_0\right]^2\left[1 - \frac{V_0^2}{2r_m g_m}\right] = 0\end{aligned} \tag{5-15}$$

The above equation is unambiguously solvable for the required acceleration $a = \frac{T}{m}$ for any velocity V_0 smaller than the escape velocity V_{esc}. In this idealized case, a is constant along the descent trajectory, which leads to a decreasing thrust T, because m decreases due to the spacecraft's fuel consumption. The precision of the result limited by the initial assumption of $\frac{h}{r_m} \ll 1$, the simplifications for the integration of the equations of motion and the constant value of the local gravitational attraction g_m. The approximate distance traveled across the surface – the downrange δ_d – can be determined with (Citron et al., 1964):

$$\delta_d = r_m\sigma_f = \frac{V_0^2}{2a}\cos\gamma_0\left[\frac{V_0^2 + 2g_m h_0}{V_0^2 + g_m h_0}\right]\left[\frac{r_m}{r_0}\right] \tag{5-16}$$

The geometry of the downrange (and cross range) is discussed in more detail in Section 4-6. The gravity turn guidance assumes a two-dimensional situation, so the cross range is not controlled with Eq. (5-15). A simple guidance logic will track the cross range value and eliminate the errors using the four lateral attitude thrusters pointing in the $\pm X_B$-direction.

The approximate time required for the descent process follows from integrating Eq. (5-14) with respect to t (note that $V = \frac{\mathrm{d}s}{\mathrm{d}t}$) with the boundary condition $s = 0$ at $t = 0$:

$$t_f = \frac{2s_f}{V_0} \tag{5-17}$$

The gravity turn guidance equations in the Enceladus Lander Simulator will now be applied to a real descent problem. The initial conditions and the configuration data are taken from Citron et al. (1964), because comparing the results also ensures a correct implementation of the guidance logic in the software code. This is also the reason why some of the following graphs display non-SI units – the translation is purely a unit conversion at the end of the respective simulation. Note that the configuration and test simulations in this section are executed in `MATLAB`; the full lander simulator in `C++` will then use the results found in this section. This method is also used for the navigation and control system configuration, see Section 5-2-5 and 6-1-5, respectively. The simulation parameters are listed in Table 5-1.

Name	Symbol	Value	Comment
initial velocity	V_0	1672 $\frac{\mathrm{m}}{\mathrm{s}}$	= 5488 $\frac{\mathrm{ft}}{\mathrm{s}}$; V_c at h_0
initial altitude	h_0	15240 m	= 50,000 ft
initial flight path angle	γ_0	0°	—
initial mass	m_0	10,000 kg	from lunar module
gravitational acceleration	g_m	1.5966 $\frac{\mathrm{m}}{\mathrm{s}^2}$	—
radius moon	r_m	1737 km	lunar radius
specific impulse	I_{sp}	260 s	standard value
time step	Δt	0.1 s	—

Table 5-1: Gravity turn guidance simulator parameters (unit test)

The equations of motion for the lander's true state follow from Fig. 5-1 and are in fact a different form of cylindrical equations of motion (5-2) and (5-3):

$$m\dot{V} = -mg_m \sin\gamma - T \tag{5-18}$$

$$mV\dot{\gamma} = -mg_m \cos\gamma \left(1 - \frac{V^2}{g_m r}\right) \tag{5-19}$$

The rate of change of the polar angle and the altitude are then

$$\dot{\sigma} = \frac{V}{r}\cos\gamma \tag{5-20}$$

$$\dot{h} = V \sin\gamma. \tag{5-21}$$

Figure 5-3 shows the velocity, altitude and flight path angle as a function of the of dimensionless time for a constant acceleration value. $\frac{a}{g_m}$ follows from the spacecraft's initial state

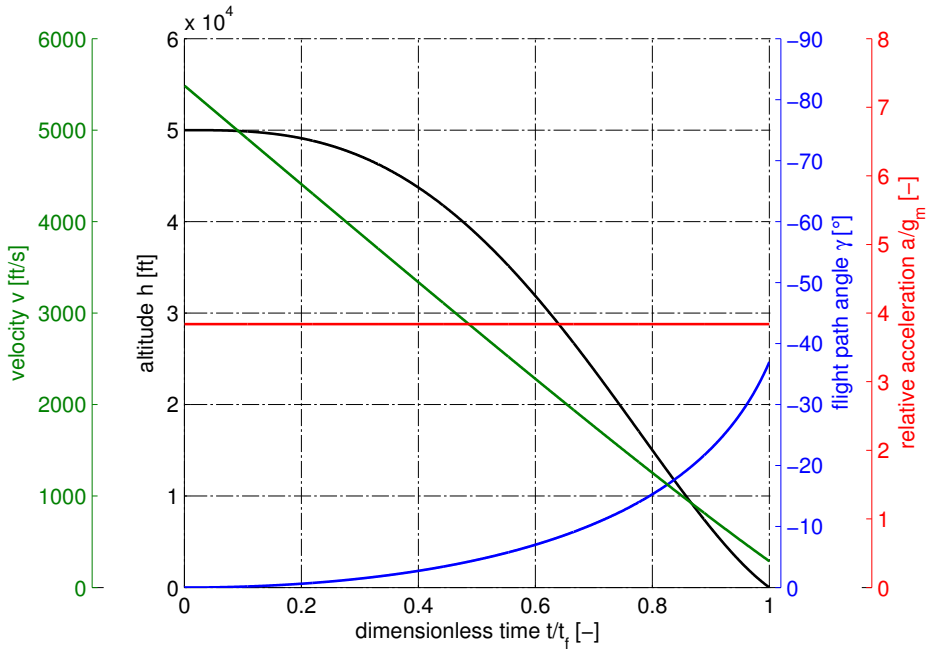


Figure 5-3: Gravity guidance for a lunar landing with constant acceleration

as given in Table 5-1. The thrust force decreases over time to compensate for the mass loss $\dot{m} = -\frac{T}{g_E I_{sp}}$ through the main engine. The velocity at touchdown is 88 fts^{-}1 (27 ms^{-1}), while γ_f is about -40°. These deviations from the expected values of 0 fts^{-1} and -90°, respectively, are the consequence of the simplifications made during the derivation of the gravity turn guidance logic for the spherical moon case. Far better results are achieved by recalculating a each integration step, which in facts leads to an explicit guidance scheme. In this case, a remains approximately constant until $\frac{t}{t_f} = 0.8$, where a slightly decreases – with this technique, the lander achieves a vertical touchdown touchdown at zero velocity, see Fig. 5-4. Figure 5-5 shows the results of the gravity turn simulations of Citron et al. (1964). Their resemblance indicates that the software implementation of Eq. (5-15) was successful.

The same principle was used for figures 5-6 and 5-7 to test the implementation of the downrange estimation from Eq. (5-16). These figures show the estimated downrange and acceleration level as a function of the initial altitude. The remaining initial conditions are the same as used for the previous simulations. Both simulations use constant acceleration levels - the recalculation of the required acceleration as discussed before leads in this case to absolute downrange errors three orders of magnitude larger as shown in Fig. 5-6. The relative error for variable acceleration at $h_0 = 200.000$ ft is 27% in contrast to only 0.1% for constant acceleration.

The large downrange estimation errors for the explicit guidance logic requires a new derivation of the expression for δ_d, because the final path length s_f is not a constant anymore, see Eq. (5-12) and (5-16). This in turn requires a completely new approach to derive a, as the inaccuracies arise from the initial problem simplifications. The constant acceleration case on the other hand may still be used as an initial guidance logic – the downrange estimation is quite good, after all. The quadratic guidance law presented in the next section can correct

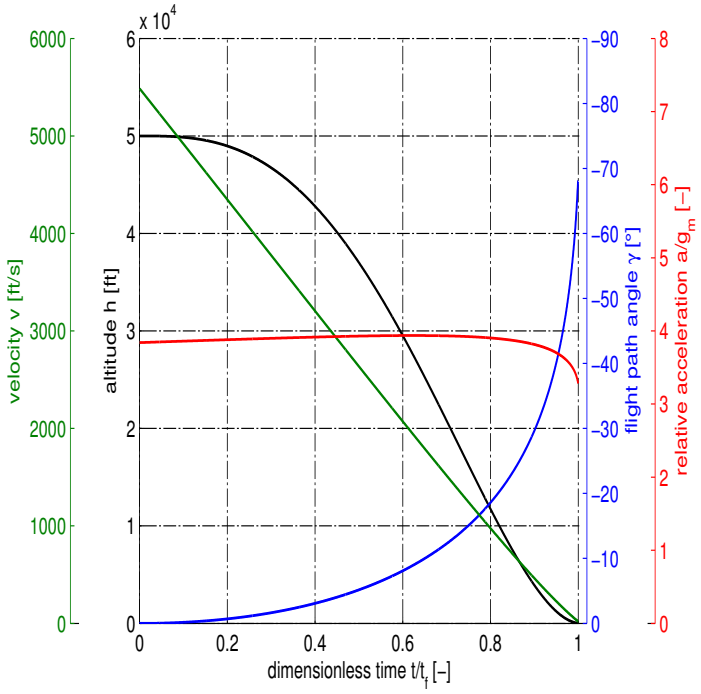


Figure 5-4: Gravity guidance for a lunar landing with variable acceleration. Horizontal scaling modified for comparison with Fig. 5-5.

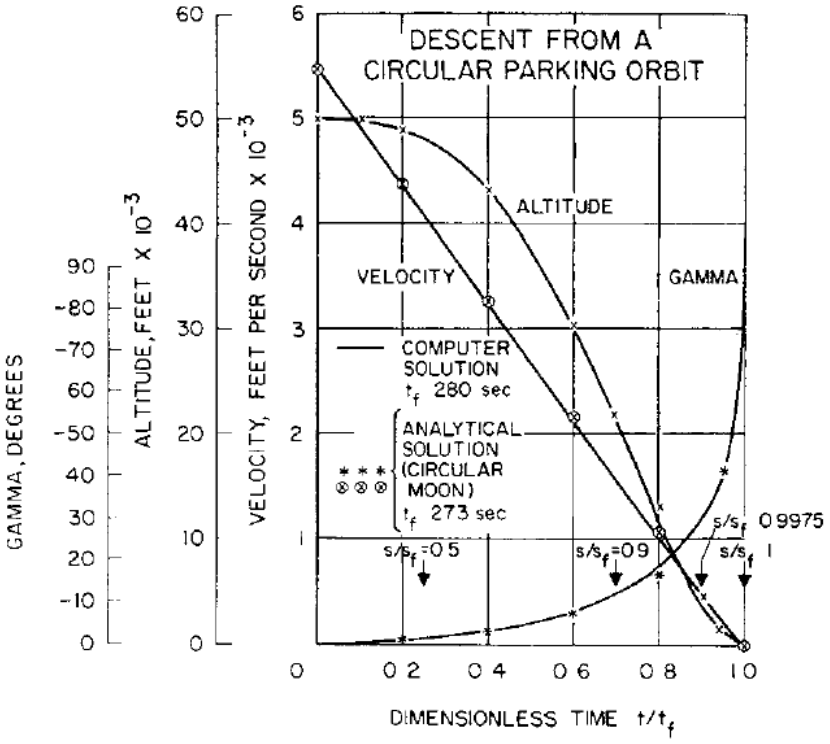


Figure 5-5: Gravity guidance for a lunar landing (Citron et al., 1964)

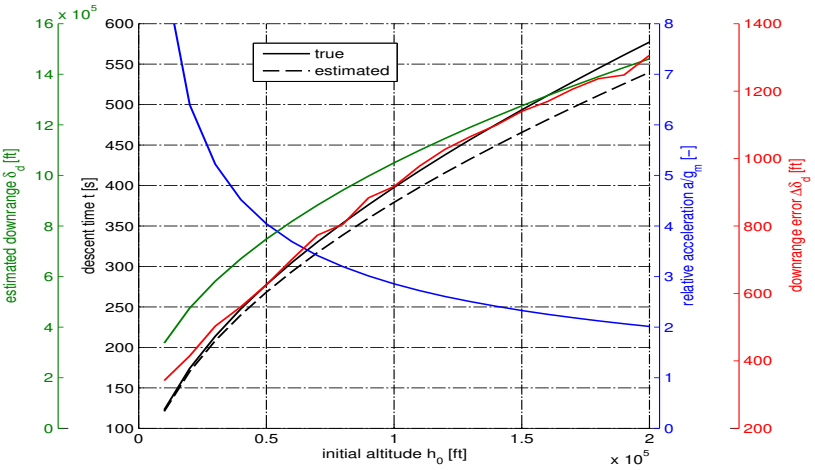


Figure 5-6: Downrange and acceleration as a function. Scaling modified for comparison with Fig. 5-7.

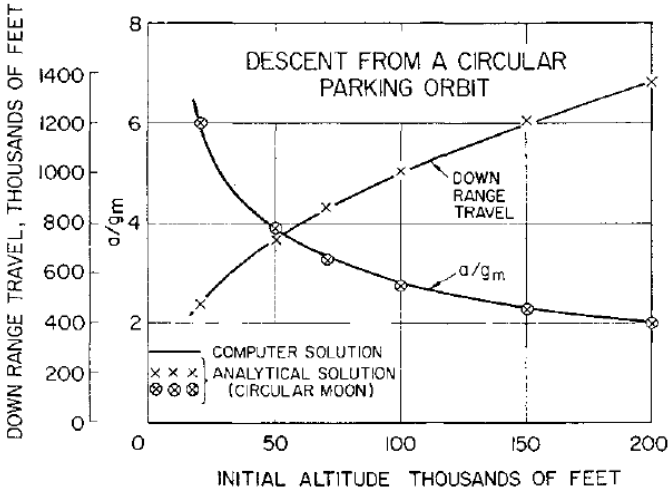


Figure 5-7: Downrange and acceleration as a function of initial altitude (Citron et al., 1964)

the accumulated errors. The question remains whether the use of multiple guidance laws can have a positive effect on the fuel consumption. The simulation results from Enceladus Lander Simulator will provide more information about this, see Chapter 8. Note, that the descent time estimation error increases for higher altitudes: Between h_0 =50,000 ft and h_0 =200,000 ft, the error increases from 3% to 7%.

Flat Moon

In case of a low initial horizontal velocity, the curvature of the surface becomes negligible, which significantly simplifies the mathematical description of the lander's motion. The equations of motion in a surface-fixed reference frame similar to one defined in Section 4-1-4, where the Y-, X- and Z-axis point upward, forward and sideward, respectively, are (Citron et al., 1964):

$$\ddot{y} = -a\frac{\dot{y}}{V} - g_m \tag{5-22}$$

$$\ddot{x} = -a\frac{\dot{x}}{V} \tag{5-23}$$

The derivation process is much simpler compared to the spherical moon model and will not be shown here. The only restriction that will reduce the accuracy of the results within the boundaries of the flat moon model is the assumption of a constant gravity field. The acceleration command follows from the expression (Citron et al., 1964)

$$\left(\frac{a}{g_m}\right)^2 + \sin\gamma_0\left[\frac{V_0^2}{2h_0 g_m}\right]\frac{a}{g_m} - \left[\frac{V_0^2\left(1+\sin^2\gamma_0\right)}{4h_0 g_m} + 1\right] = 0 \tag{5-24}$$

which is unambiguously solvable for the required acceleration $a = \frac{T}{m}$. The deceleration command is, in fact, slightly too large throughout the descent due to the assumption of a constant gravity field. Similar to the guidance logic for the spherical moon model, a may be recalculated each integration step for more accurate results. The equations for the downrange prediction and descent time are (Citron et al., 1964)

$$\delta_d = \frac{V_0^2\cos\gamma_0}{2}\left[\frac{1+\sin\gamma_0}{2a+g_m} + \frac{1-\sin\gamma_0}{2a-g_m}\right] \tag{5-25}$$

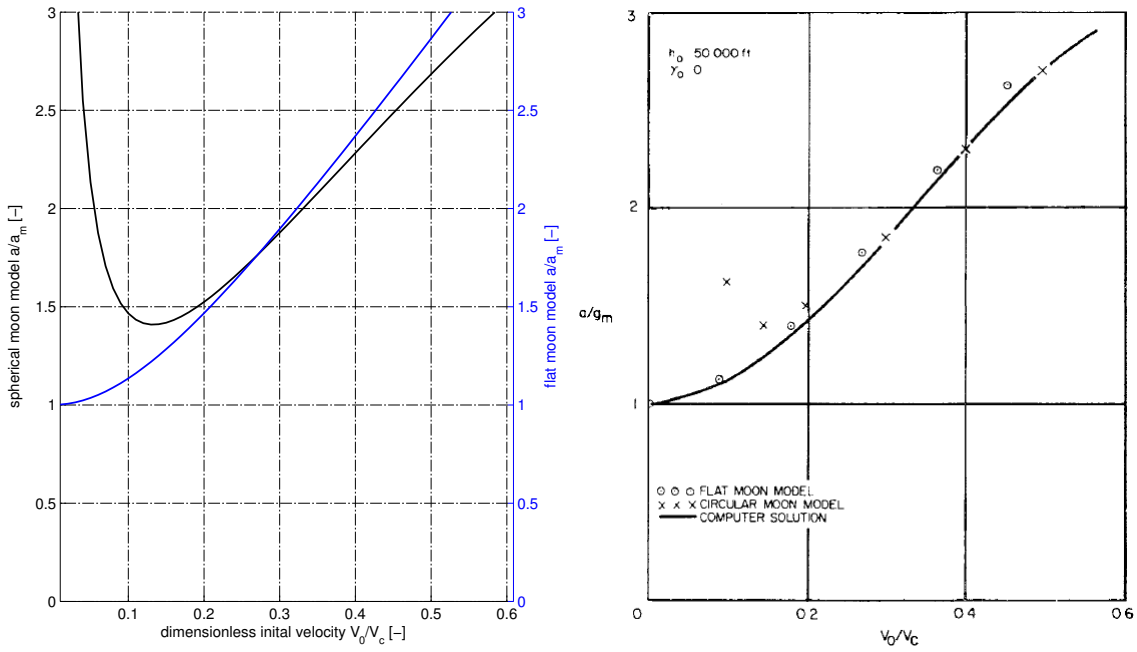
and

$$t_f = \frac{V_0^2}{2}\left[\frac{1+\sin\gamma_0}{a+g_m} + \frac{1-\sin\gamma_0}{a-g_m}\right], \tag{5-26}$$

respectively.

The acceleration commands for initial velocities below $\frac{V_0}{V_c} = 0.6$ are shown in Fig. 5-8a, for both the flat moon and the spherical moon model. The remaining simulation parameters are the same as in the previous section and listed in Table 5-2. With respect to the spherical model, the flat moon model returns lower acceleration commands for $\frac{V_0}{V_c}$ below 0.27 and larger commands for all other cases. The large deviation of $\frac{a}{a_m}$ from the expected value of 1 for initial velocities close to 0 indicates the shortcomings of the spherical model in that velocity region.

The successful software implementation of Eq. (5-24) is tested by comparing Fig. 5-8a with the results from Citron et al. (1964) shown in Fig. 5-8b.



(a) Simulation Results **(b)** Citron et al. (1964)

Figure 5-8: Acceleration command results for the spherical moon and the flat moon model for different initial velocities V_0.

Gravity Turn at Enceladus

The comparatively small radius of Enceladus and the low gravitational acceleration of Enceladus imposes more stringent limitations on the lander's conditions. The simulation parameters are based on the lander characteristics and Enceladus' physical characteristics, and have been discussed in the previous chapters. For convenience, the setup data is collected in Table 5-2. The velocity at touchdown – which should be zero in the idealized case for constant a – and the descent time estimation error increase with larger initial altitudes for the spherical moon model, see Fig. 5-9. This is largely due to the fact that the condition $\frac{h}{r_m} \ll 1$ for Eq. (5-15) is no longer fulfilled. The previous lunar case uses $h_0 = 15$ km for $r_m = 1737$ km. The same ratio $\frac{h_0}{r_m}$ for Enceladus gives a reference value in the order of 3 km. The downrange error increases exponentially, but is still surprising low with about 700 m for $h_0 = 50$ km. The velocity error, on the other hand, for the same initial altitude is 50 ms^{-1}.

One way to limit the impact of the end velocity error for the spherical moon case is to use a lower V_0. Figure 5-11 shows the values of v_f and h_f as a function of the ratio $\frac{V_0}{V_c}$ for a fixed initial altitude $h_0 = 15$ km. At $\frac{V_0}{V_c} = 0.6$, the simulation stopped for the first time by a zero altitude value instead of a negative velocity value. A lower V_0 would thus cause the lander to hover above the surface with a downrange to the target area in order of a few hundred meters. The $\frac{V_0}{V_c}$ region between 0 and 0.6 should be simulated with a flat moon model, which leads to a better approximation of the descent trajectory for low initial velocities.

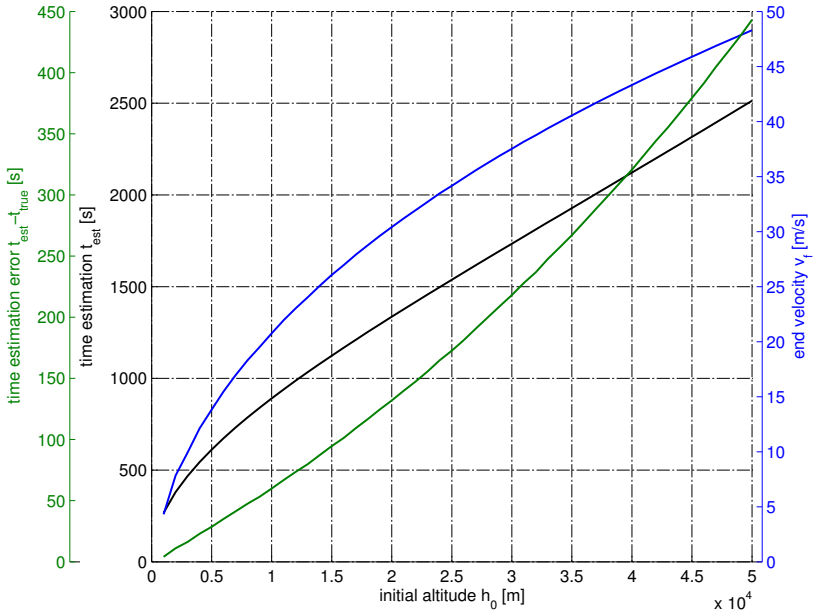


Figure 5-9: Spherical moon model: velocity and descent time error for different h_0

Name	Symbol	Value	Comment
initial velocity	V_0	$0 - V_c$	depends on h_0
initial altitude	h_0	2 – 15 km	—
initial flight path angle	γ_0	0°	—
initial mass	m_0	335 kg	at entry
gravitational acceleration	g_m	0.1011 $\frac{\text{m}}{\text{s}^2}$	at surface
radius moon	r_m	252.1 km	—
specific impulse	I_{sp}	312 s	—
integration time step	Δt	1 s	—

Table 5-2: Gravity-turn guidance simulator for the Enceladus lander

The downrange errors $\Delta\delta_d$ and the end altitude h_f for both the flat moon model and the spherical moon model are shown in Fig. 5-12 as a function of the initial velocity. The lander's initial are the same as in the previous simulations, and the acceleration commands are constant for each value of $\frac{V_0}{V_c}$. The flat moon model achieves lower values for both $\Delta\delta_d$ and h_f for initial velocities up to $\frac{V_0}{V_c} = 0.4$, while the spherical model is clearly the better choice for velocity ratios larger than 0.65. The transition region between 0.4 and 0.65 is relatively large, and it might be feasible to base the downrange prediction and the thrust acceleration on different models. For simplicity, the Enceladus lander gravity turn guidance logic will change from the flat moon model to the spherical moon model at $\frac{V_0}{V_c} = 0.5$. Note, that the results shown in Fig. 5-12 can be improved further by decreasing h_0, as discussed in the previous section.

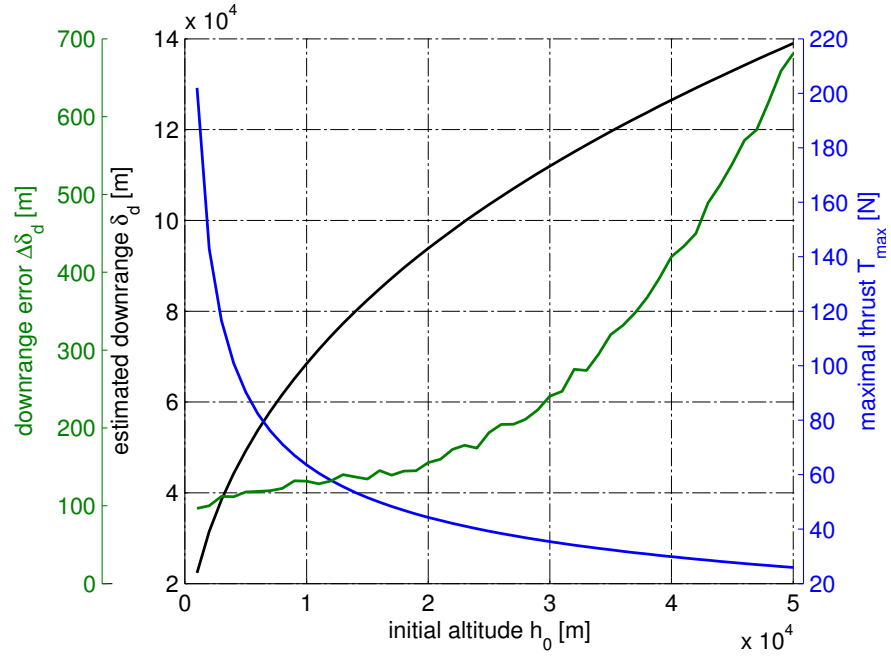


Figure 5-10: Spherical moon model: maximal thrust and estimated downrange for different h_0

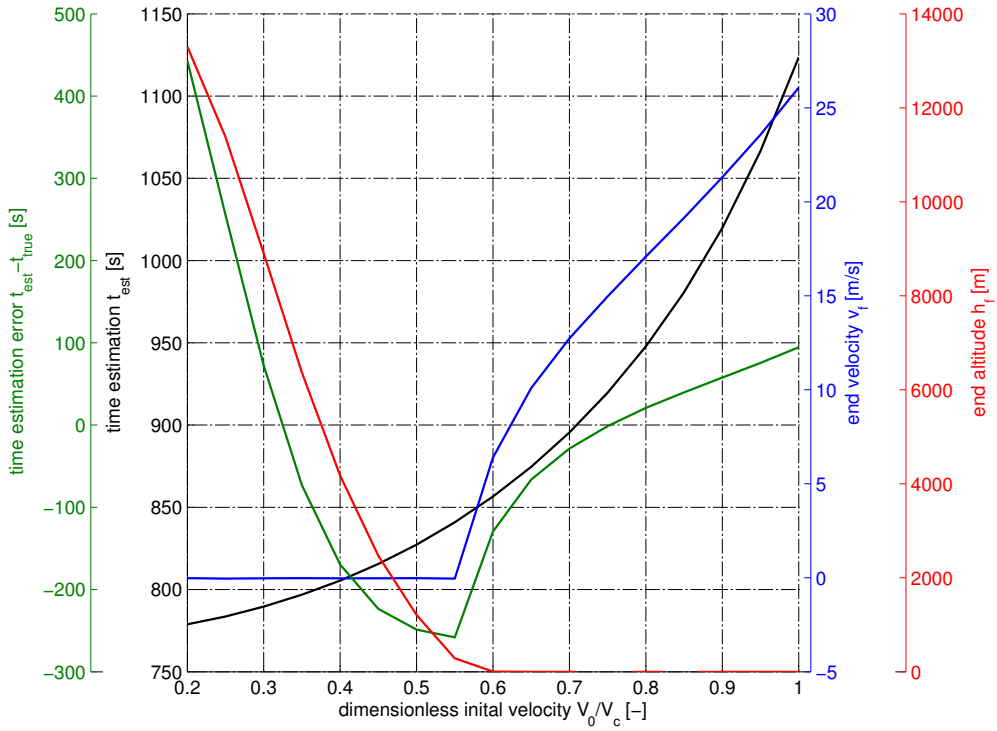


Figure 5-11: Spherical moon model: velocity, altitude and descent time errors for different V_0 at $h_0 = 15,000$ m

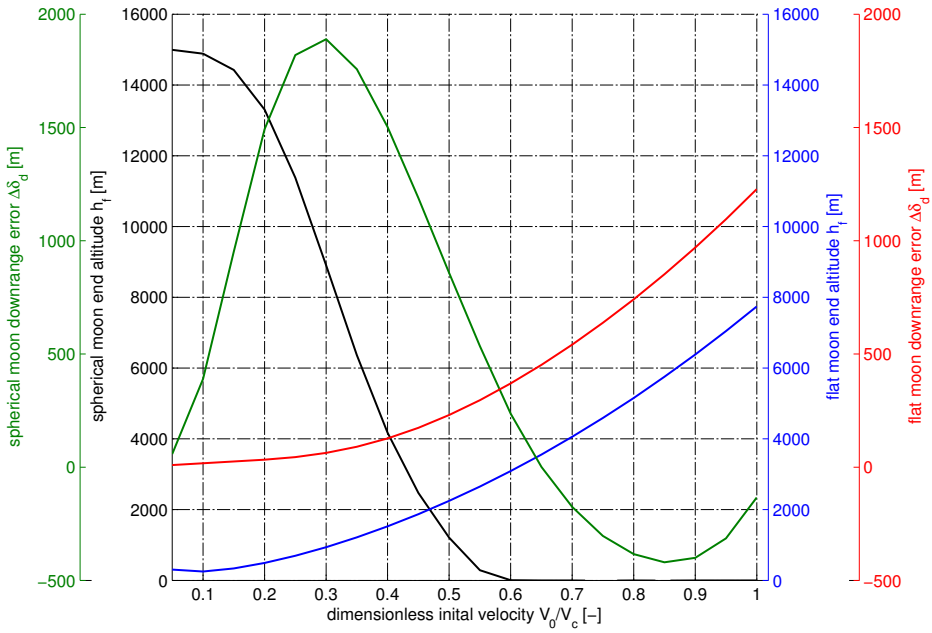


Figure 5-12: Spherical and flat moon model: downrange and altitude errors for different V_0 at $h_0 = 15$ km

5-1-2 Quadratic Guidance

The Apollo and Altair basic guidance law for the braking and approach phase consists of a time-dependent, fourth-order polynomial function that yields the desired trajectory for a given position. The guidance logic follows from solving the acceleration profile two-point boundary-value-problem (TPBVP) between the current position of the lander and its target location (Lee et al., 2010). The target acceleration at a given point also indirectly determines the target attitude. The braking phase and approach phase use the same algorithm, but with different target states. The boundaries of the TPBVP are the initial and target values of the vehicle's position (r_0, r_t), velocity (v_0, v_t) and acceleration (a_0, a_t). A polynomial must be at least quadratic to meet the three target constraints, so the acceleration profile is chosen to have the form (Wong et al., 2002)

$$a(t) = C_0 + C_1 t + C_2 t^2 \tag{5-27}$$

where the coefficients C_0, C_1 and C_2 follow from the six boundary conditions. Two boundary conditions correspond to the velocity and the position, respectively. Thus, integrating Eq. (5-27),

$$v(t) = C_0 t + \frac{1}{2} C_1 t^2 + \frac{1}{3} C_2 t^3 + v_0 \tag{5-28}$$

$$r(t) = \frac{1}{2} C_0 t^2 + \frac{1}{6} C_1 t^3 + \frac{1}{12} C_2 t^4 + v_0 t + r_0. \tag{5-29}$$

The travel time or *time-to-go* from the current to the target state is now defined by t_{go}. Substituting $t = t_{go}$ in the above equations gives (modified from Wong et al. (2002))

$$a(t_{go}) = a_t = C_0 + C_1 t_{go} + C t_{go}^2$$

$$v(t_{go}) = v_t = C_0 t_{go} + \frac{1}{2} C_1 t_{go}^2 + \frac{1}{3} C_2 t_{go}^3 + v_0 \tag{5-30}$$
$$r(t_{go}) = r_t = \frac{1}{2} C_0 t_{go}^2 + \frac{1}{6} C_1 t_{go}^3 + \frac{1}{12} C_2 t_{go}^4 + v_0 t_{go} + r_0$$

from which the coefficients C_0, C_1 and C_2 can be derived (Wong et al., 2002):

$$C_0 = a_t - 6\left(\frac{v_t + v_0}{t_{go}}\right) + 12\left(\frac{r_t - r_0}{t_{go}^2}\right) \tag{5-31}$$

$$C_1 = -6\left(\frac{a_t}{t_{go}}\right) + 6\left(\frac{5v_t + 3v_0}{t_{go}^2}\right) - 48\left(\frac{r_t - r_0}{t_{go}^3}\right) \tag{5-32}$$

$$C_2 = 6\left(\frac{a_t}{t_{go}^2}\right) - 12\left(\frac{2v_t + v_0}{t_{go}^3}\right) + 36\left(\frac{r_t - r_0}{t_{go}^4}\right) \tag{5-33}$$

The only unknown at this point is the value for t_{go}. In principle, t_{go} can be chosen freely, but often there are mission requirements that impose additional constraints. The hazard avoidance system, for example, requires a minimum amount of time until it has identified a safe landing spot. Another possibility is to determine the optimal value for t_{go} for a minimum fuel consumption, using a simple global optimization technique such as the Monte Carlo method or a nested do-loop. Wong et al. (2002) chooses t_{go} such that the acceleration profile for the vertical motion is a linear function of time, thus $C_2 = 0$. In that case,

$$t_{go} = \frac{2v_t + v_0}{a_t} + \sqrt{\left(\frac{2v_t + v_0}{a_t}\right)^2 + \frac{6\left(r_0 - r_t\right)}{a_t}} \tag{5-34}$$

for $a_t \neq 0$, or if $a_t = 0$,

$$t_{go} = \frac{3\left(r_t - r_0\right)}{v_0 + 2v_t}. \tag{5-35}$$

The assumption of a linear vertical acceleration profile allows an easier analysis of the lander's motion under the effect of the gravitational acceleration. The quadratic guidance uses the surface-fixed reference frame as defined in Section 4-1-4 for a simple division of the velocity components: the X_S-, Y_S- and the Z_S-axis describe the upward, forward and sideward motion of the lander, respectively, with respect to origin of the reference frame. Each direction is guided by a separate guidance law. The coefficients C_0, C_1 and C_2 for each direction are determined with the t_{go} from the vertical motion and the current target and state information (see Eq. (5-31)). These calculations are repeated very guidance cycle, for all axes. If needed, the surface-fixed reference frame may be updated simultaneously to reduce the effects of the moon's spherical shape, but it is important to track the target position during all frame updates, because its location follows from a hazard map that was taken with respect the surface-fixed frame at that instance of time. Note that the desired acceleration along the X_S-axis as calculated with Eq. (5-27) includes the gravitational acceleration, which must be subtracted in to generate the actual acceleration command for the lander control system. The gravitational acceleration should not be assumed constant during the flight: the initial guidance system simulations returned a position error in the order of 70 m for an initial altitude of 3000 m in case $g_{\text{enc}} = g_{\text{enc},0}$.

The search for a suitable t_{go} starts with Eq. (5-35), because the desired vertical acceleration is usually zero to avoid large velocity changes shortly before the touchdown. The assumption

of a linear acceleration profile reduces the degree of freedom of Eq. (5-27), so not every combination of initial and target state will lead to a valid result. The time-to-go is negative and thus invalid in case – for example – a descent trajectory ($r_t - r_0 < 0$) with $v_t = 0$ is desired, but the spacecraft is currently moving upward ($v_0 > 0$). Equation (5-34) may be used to find t_{go} in case a_t is specified, but similar restrictions to the initial and final lander state apply. Note, that $t_{go} > 0$ only indicates that it is mathematically possible to reach the target state. To check the actual reachability, the calculated t_{go} must be tested in the way as discussed in the next segment.

A more general approach is to determine a *range* of t_{go} for all a_t within the main engine limitations. The main engine is freely throttleable between 10% and 100% of the rated thrust, see Table 3-2. The lander is not allowed to accelerate downward for safety reasons, so the allowable a_t range is $\frac{490\text{N}}{335\text{kg}}[0, 1] + g_{enc} = [-0.1135\frac{\text{m}}{\text{s}^2}, 1.3492\frac{\text{m}}{\text{s}^2}]$. The t_{go} range follows now from inserting the a_t range into Eq. (5-34) and eliminating all results leading to a negative t_{go}. Figure 5-13 shows the time-to-go as a function of the of previously derived a_t range for a lander at an altitude of 1000 m with a forward and downward velocity of 34 $\frac{\text{m}}{\text{s}}$ ($0.2V_c$) and -30 $\frac{\text{m}}{\text{s}}$, respectively, aiming at a target on the surface 1000 m in front of the initial sub-satellite point. The curve is discontinuous as a consequence of the form of Eq. (5-34). The initial and target states with respect to the surface-fixed reference frame are collected in Table 5-3.

Name	Symbol	Value	Unit
initial position	r_0	[1000, 0, 0]	m
initial velocity	v_0	[-30, 34, 0]	$\frac{\text{m}}{\text{s}}$
target position	r_t	[0, 1000, 0]	m
target velocity	v_t	[0, 0, 0]	$\frac{\text{m}}{\text{s}}$
target acceleration	a_t	[-0.1135, 1.3492]	$\frac{\text{m}}{\text{s}^2}$

Table 5-3: Quadratic guidance simulation parameters (S-frame)

As can be seen in Fig. 5-13, the time-to-go increases exponentially when a_{t_x} is negative and approaches 0. This is a consequence of the initial and target states given in Table 5-3: For a negative a_{t_x}, the term $\frac{2v_t+v_0}{a_t}$ in Eq. (5-34) is positive and approaches infinity for $a_{t_x} \to 0$. On the other hand, for a positive a_{t_x}, this term is negative and cancels out the first term in the square root. The value of t_{go} for $a_{t_x} \to 0$ approaches the result of Eq. (5-35) for the case $t_{go} = 0$.

The next step is to determine the factors C_0, C_1 and C_2 for all axes for each remaining value of t_{go}, which allows the analysis the complete trajectory for each t_{go}. All invalid values for t_{go} are then filtered out in three consecutive steps.

Firstly, all t_{go} are discarded for which the initial acceleration value a_{0_x} is outside the available acceleration range. The vertical motion along the X_S-axis is treated differently than the two horizontal motions due to the additional gravitational acceleration. Based on Eq. (5-27) and (5-31), the following relation must be true for all remaining values:

$$0 \leq C_{0_x} - g_{enc} \leq \frac{T_{rated}}{m_0} + g_{enc} \tag{5-36}$$

The target acceleration for the X_S-axis is as discussed before always within the acceleration range, so if both a_{0_x} and a_{t_x} are valid, all intermediate values a_x are also valid due to the

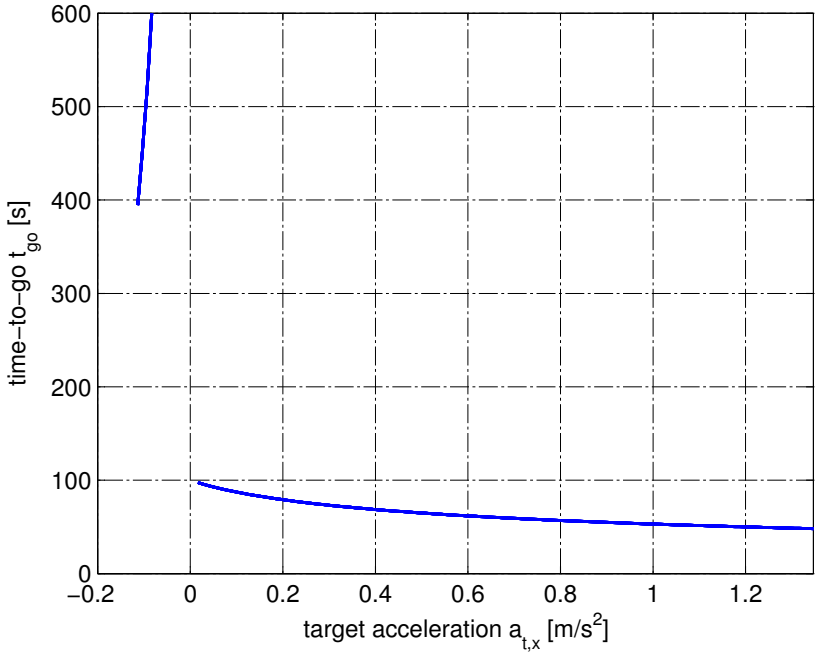


Figure 5-13: Time-to-go as a function of the target acceleration (X_S-axis). t_{go} increases drastically for small negative $a_{t,x}$; the output is limited here to $t_{go} = 600$ s.

linear character of the acceleration profile. Figure 5-14 shows the values of a_{0_x} as a function of a_{t_x} for initial and target states from Table 5-3. No values must be removed, as a_{t_x}-a_{0_x}-pairs are within the boundaries set Eq. (5-36).

Secondly, all t_{go} are removed that lead to a negative altitude at any time during the descent. The local altitude extremes occur at the instances of time t^* for which the first time derivative of the altitude – the vertical velocity – is zero. The candidate points are thus

$$t^*_{1,2} = -\frac{C_{0_x}}{C_{1_x}} \pm \sqrt{\frac{C^2_{0_x}}{C^2_{1_x}} - 2\frac{v_0}{C_{1_x}}}. \tag{5-37}$$

In case the conditions

$$0 \leq t^* \leq t_{go} \ \wedge \ a(t^*) > 0 \tag{5-38}$$

are met, $r(t^*)$ is the local minimum. The lowest value r_{min} of the altitudes r_0, $r(t_*)$ and r_t then is global minimum, and all t_{go} for which $r_{min} < 0$ are disregarded. Figure 5-15 shows the values of r_{min} as a function of $a_{t,x}$ for the same example as used before. The red line marks all $r_{min} - a_{t_x}$-pairs that will be removed in this step. Note, that even the eliminated t_{go}'s – theoretically – guide the lander to the desired target state.

Thirdly, all t_{go} must be removed that, at any time, lead to a required total acceleration larger than the maximum acceleration the main engine can provide. The total acceleration the main engine must deliver is

$$a_{eng}(t) = \sqrt{[a_x(t) - g_{enc}]^2 + a_y(t)^2 + a_z(t)^2}. \tag{5-39}$$

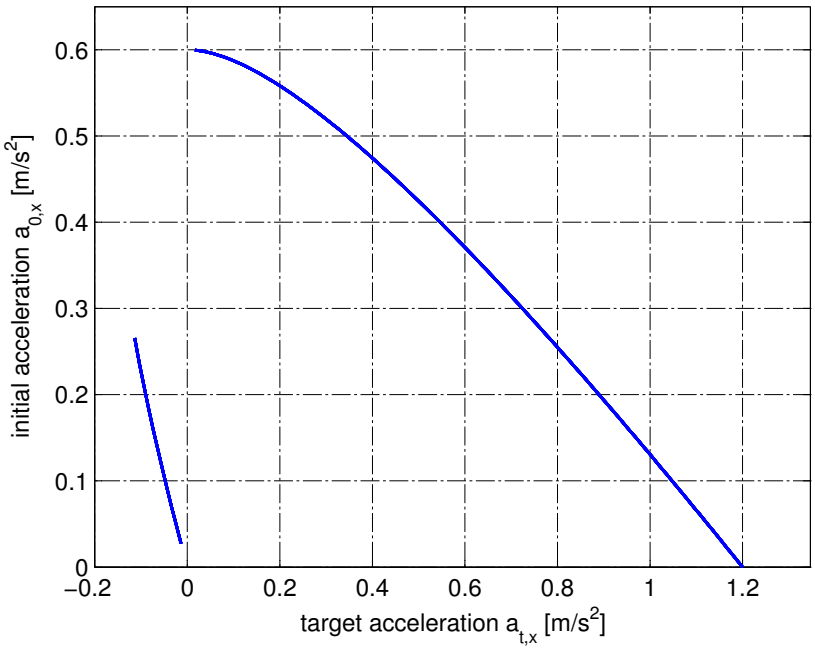


Figure 5-14: Initial acceleration as a function of the target acceleration (X_S-axis). No elements were removed. The discontinuity is a consequence of the discontinuity in Fig. 5-13.

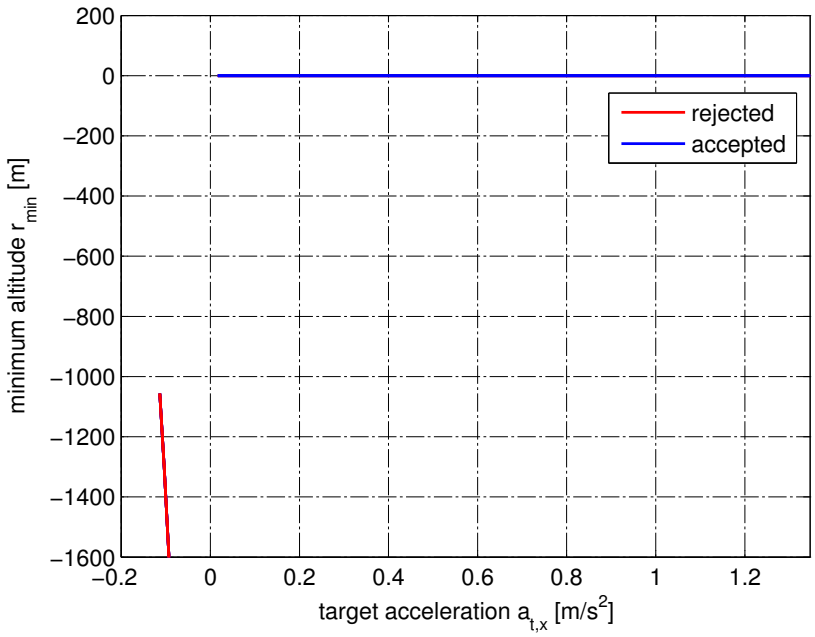


Figure 5-15: Minimum altitude as a function of the target acceleration (X_S-axis). The red line indicates all elements that will be removed. The discontinuity is a consequence of the discontinuity in Fig. 5-13.

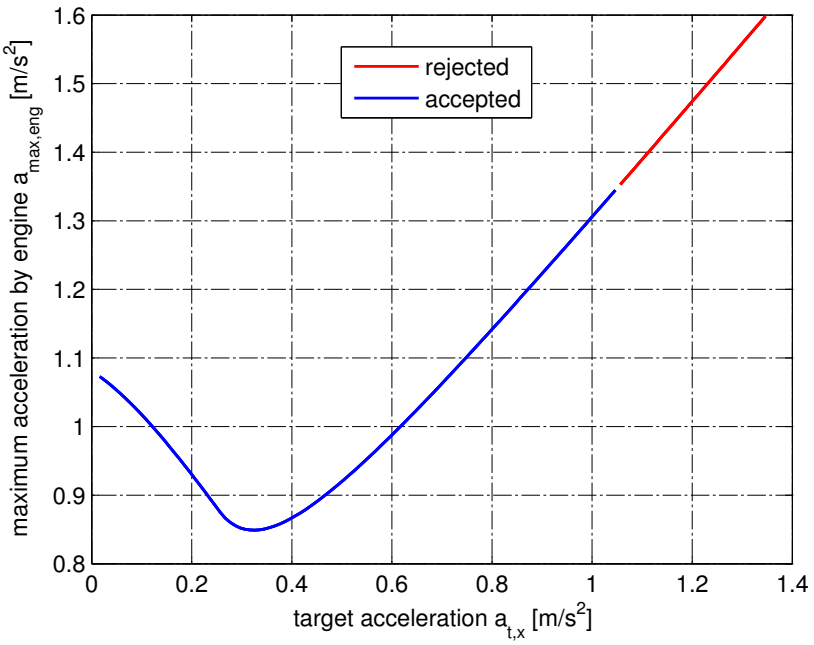


Figure 5-16: Maximal acceleration of the main thruster as a function of the target acceleration (X_S-axis). The red line indicates all elements that will be removed.

The maximum total acceleration $a_{max,eng}$ is hard to find analytically. The alternative is to successively increase t by a step-size Δt until t_{go} is reached, and simply safe the largest value of a_{eng}. As the fuel consumption is directly related to the current thrust level and thus the acceleration integrated over time, finding $a_{max,eng}$ should be combined with calculating the fuel consumption of the entire trajectory. The fuel consumption m_f at time t is

$$m_f = \frac{a_{eng}(t)m(t)}{g_0 I_{sp}} \tag{5-40}$$

where g_0 is the standard acceleration due to gravity (9.80665 $\frac{\mathrm{m}}{\mathrm{s}^2}$). The total fuel consumption for a given t_{go} follows from the numerical integration process. Note, that calculations of $a_{max,eng}$ and $m_{fuel,max}$ must be repeated for each value of t_{go}. Figure 5-16 shows the values of $a_{max,eng}$ as a function of a_{t_x}. The red line indicates all values that exceed the maximum thrust the main engine can provide; these values are removed from the candidate pool.

At this point, all remaining values for t_{go} can be used by the quadratic guidance law for a safe touchdown. The t_{go} with the lowest fuel consumption will be used as input for the guidance system. Figure 5-17 shows the values t_{go} and a_{t_x} as a function of fuel consumption. The minimum fuel consumption is about 7.6 kg including the effect of g_{enc} for a time-to-go of 57.8 s and a vertical target acceleration of 0.76 $\frac{\mathrm{m}}{\mathrm{s}^2}$.

The position, velocity and acceleration as a function of time for the found $t_{go} = 57.8$ s can be found in Fig. 5-18a, 5-18b and 5-18c, respectively. The target state is reached as expected, with a linear vertical acceleration profile, and quadratic horizontal acceleration profiles. The t_{go} for the Y_S- and Z_S-axis might be reduced by a few seconds to avoid a spacecraft rotation close to the surface.

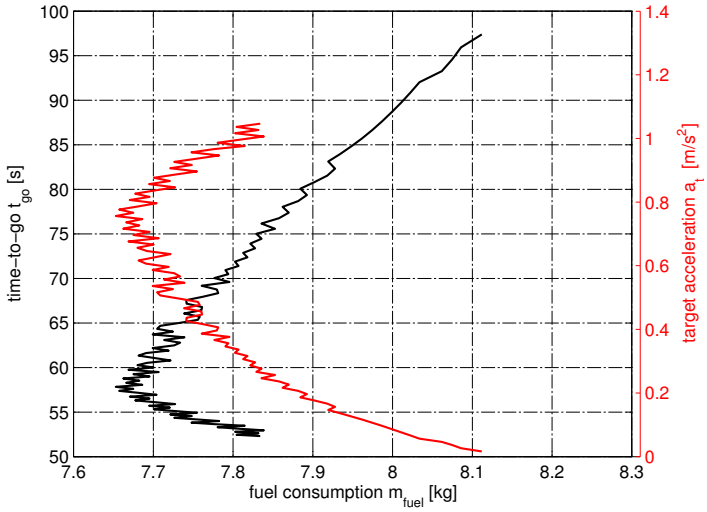
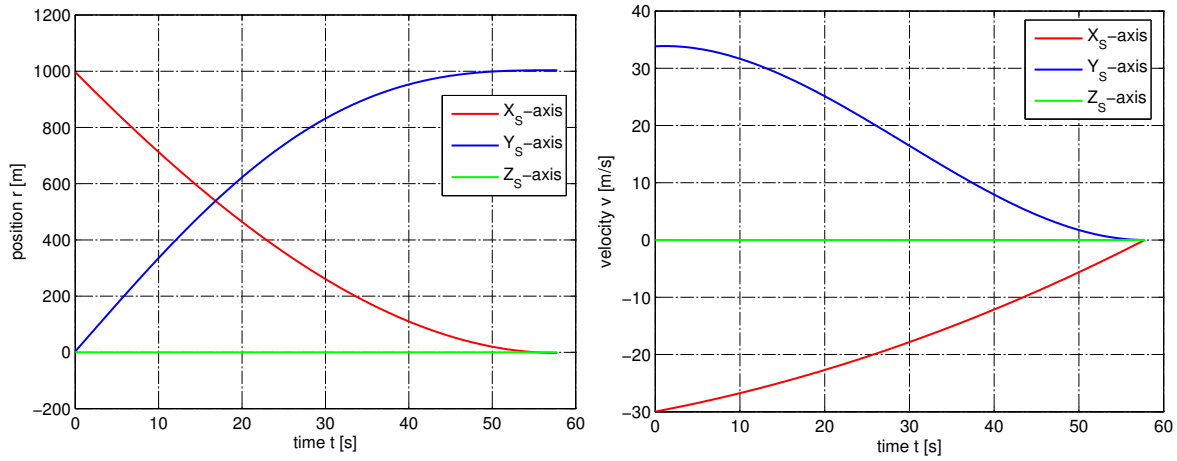


Figure 5-17: Time-to-go and vertical target acceleration as a function of the fuel consumption (S-frame)

The above calculations of t_{go}, C_0, C_1 and C_2 are repeated every guidance cycle, each axis separately (the gravitational acceleration must be taken into account for the vertical axis). The results are then substituted in acceleration profile (Eq. (5-27)). The commanded acceleration then is the difference between the actual acceleration and $a(t)$. Near the target area, t_{go} gets small and causes an explosion of the expressions for C_0, C_1 and C_2. This problem can be avoided by selecting a new target state some seconds before the critical values of t_{go}. The LM used a target below the lunar surface for the approach phase in order to achieve a better visibility for the astronauts. This is also important for the LIDAR-based hazard avoidance system of the Enceladus lander.

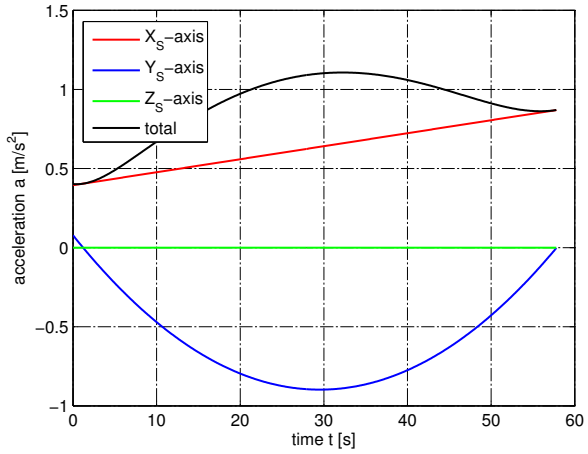
In summary, the guidance system recalculates an acceleration profile between the current position and the target position, taking into account possible derivations from previous trajectory. The quadratic guidance logic was successfully used for the Apollo landings, but also future missions to the Moon (Kos et al., 2010) or Mars (Wong et al., 2002) take this method into consideration due to its robustness along with its relatively high fuel-efficiency.

Just like the moon landers, Silenus will perform a pitch-up maneuver at the end of the braking phase in order to allow the LIDAR to scan the target area and return data to the hazard avoidance system. The exact time of the maneuver is a trade-off between the efficient range of the LIDAR (maximal range is 2 km, see Section 6-1-7) and the time left for maneuvering.



(a) position

(b) velocity



(c) acceleration

Figure 5-18: Quadratic guidance simulation for $t_{go} = 57.8$ s

5-1-3 Velocity Nullifying Guidance

The terminal phase ideally consists only of a vertical descent to the designated landing spot. A non-zero horizontal touchdown velocity can cause an overturning of the vehicle, which results in the failure of the mission. The velocity and roll rate limits depend on the landing gear design - for Apollo 11 lander they were 0.45 ms^{-1} horizontal, 0.2 ms^{-1} vertical, -1.5 degs^{-1} pitch, -6.2 degs^{-1} yaw and -3.7 degs^{-1} roll (Lee, 2011). A rough estimation for the landing gear limits of the Enceladus lander can be found in Section 3-3. During the near-target navigation, the horizontal velocity – and, if necessary, also the vertical velocity – can be controlled with a basic velocity nullifying guidance logic of the form

$$\mathbf{a}_{\text{cmd,S}} = -\frac{1}{\Delta t}\begin{pmatrix} V_{0x,S} - V_{t_x,S} \\ V_{0y,S} \\ V_{0z,S} \end{pmatrix} - \begin{pmatrix} g_{0_{\text{enc}}} \\ 0 \\ 0 \end{pmatrix}, \tag{5-41}$$

where Δt is the time available for the velocity corrections, $\mathbf{V_{0,S}}$ the velocity at the beginning of the nullifying guidance phase, $V_{t_x,S}$ the target velocity along the X_S-axis, and $g_{0_{\text{enc}}}$ the gravitational acceleration at the surface. $V_{t_x,S}$ should be negative to avoid hovering or even climbing. The Enceladus Lander Simulator incorporates two velocity nullifying guidance modes: The first mode controls only the horizontal velocity and can be activated in case the quadratic guidance logic terminates the control of the motion along the Y_S- and Z_X-axis, while the guidance for the vertical motion is still due to a slightly larger t_{go} for that axis (see discussion in previous section). In that case, the top element of $\mathbf{a}_{\text{cmd,S}}$ is reset to zero. The control system will automatically only activate thrusters in the X_BY_B-plane provided that the Z_B-axis approximately points in nadir direction, which should always be the case at the end of the quadratic guidance phase.

The velocity nullifying guidance law for the rotational state is

$$\dot{\boldsymbol{\omega}}_{\text{cmd}} = -\frac{1}{\Delta t}\begin{pmatrix} p_0 \\ q_0 \\ r_0 \end{pmatrix} \tag{5-42}$$

where p_0, q_0 and r_0 are the rotational velocities about the body axes at the beginning of the velocity nullifying guidance phase. The reference moment about the vehicle's center of mass, $\mathbf{M}_{cm}$, then follows from Eq. (4-62) and becomes

$$\mathbf{M}_{cm} = \mathbf{I} \cdot \dot{\boldsymbol{\omega}}_{\text{cmd}} + \boldsymbol{\omega} \times \mathbf{I} \cdot \boldsymbol{\omega} \tag{5-43}$$

which is used instead of the output of the linear quaternion controller from Eq. (5-52).

Note, that Eqs. (5-41) and (5-42) represent the most basic acceleration commands to eliminate the initial velocities and cannot be updated during the time Δt. It is possible to derive more elaborated feedback controllers, see, for example, Astolfi and Rapaport (1998) and Aeyels (1985). The lander's translational and rotational velocity, however, is already small due to the quadratic guidance, so the effort to implement a complex velocity nullifying guidance law would be disproportionate to the benefits. A comparatively simple first step would be to use the existing quaternion controller for velocity nullifying by fixing the attitude command to its value at the beginning of the nullifying phase, but this requires another optimization cycle to find values for the proportional and derivative gains. The design of an explicit velocity

nullifying guidance scheme for both rotational and translational motion is a promising subject of future work on the Enceladus Lander Simulator.

Other options for the control of both the position and translational velocity is the implementation of guidance laws based on Q-guidance or on optimal control. Q-guidance is used in missiles and can be applicable to the terminal guidance phase of planetary lander, see Gregory et al. (2008). Optimal control theory is particularity useful when the horizontal velocity is unexpectedly large. JPL identified this fault scenario as an optimal control problem with a cost function J that aims at minimizing the touchdown velocity over a free or fixed horizontal distance, or downrange. The control variables are the thrust vectors, which are constrained by the engine performance. This optimization problem can be solved using the classic calculus of variations technique. The result is a the optimal control of the thrust vector as a function of time, see for the full derivations Lee (2011) and Guo and Han (2010).

5-1-4 Hybrid Ballistic-Quadratic Repositioning Guidance

In theory, the most fuel efficient repositioning sequence in an atmosphere-less environment will consist of maximum thrust maneuvers at optimal angles at the beginning and the end of trajectory. This technique minimizes the gravitational losses, but it imposes strict requirements on the control and navigation system, because the thrust magnitude and thrust orientation must be as close to the required values as possible for an acceptable landing precision. The Enceladus lander will use a maximum acceleration command in the desired direction to initiate the repositioning phase. Initially, the thruster points downward and is not aligned with the desired elevation angle. This will lead to a misaligned lander velocity vector and thus a wrong touchdown location. The implemented quadratic and velocity nullifying guidance will be used during the last leap phase to reduce the accumulated position and velocity errors. It is possible to add additional guidance laws to reduce the initial velocity errors, but this is beyond the scope of this report.

The velocity change Δv in an ideal situation and in absence of any perturbing forces follows directly from the current spacecraft mass m, and can be calculated with *Tsiolkovsky's equation*

$$\Delta v = g_0 I_{sp} \ln\left(\frac{m_0}{m}\right) \tag{5-44}$$

where g_0 and I_{sp} are the standard acceleration due to gravity and the main engine specific impulse, respectively. The initial and final lander velocity is the same for an idealized purely ballistic trajectory. The total fuel consumption then is double the fuel consumption following from Eq. (5-44) for one impulsive shot:

$$m_f = 2m_0\left(1 - \mathrm{e}^{-\frac{\Delta v}{g_0 I_{sp}}}\right) \tag{5-45}$$

The fuel consumption is higher in reality due to the gravitational losses during the acceleration phase, the perturbing forces, and – most importantly – the initial thrust angle error between the Z_B-axis and desired launch angle. The traveled horizontal distance (downrange along the Y_S-axis) of the ballistic trajectory depends on the initial velocity v_0 and its elevation angle α and can be derived as

$$y = 2\frac{v_0^2}{g_{enc}} \sin\alpha\cos\alpha. \tag{5-46}$$

The maximum range is achieved with α=45°. The hazard avoidance system, however, imposes a minimal altitude requirement h_{req} for the generation of a hazard map. The minimal vertical initial velocity is thus always

$$V_0 \sin\alpha = \sqrt{2 g_{enc} h_{req}}, \tag{5-47}$$

while the minimum range for which α can be set to the optimal angle is $y_{min} = 4h_{req}$. If the desired value for y is below this limit, v_0 follows from Eq. (5-47), and the elevation angle from inserting Eq. (5-47) in Eq. (5-46):

$$\tan\alpha = \frac{4h_{req}}{y}. \tag{5-48}$$

The guidance logic for the first part of the ballistic trajectory in terms of the desired initial velocity and direction can then be summarized as

$$v_0, \alpha = \begin{cases} \frac{\sqrt{2 g_{enc} h_{req}}}{\sin\alpha},\ \tan^{-1}\left(\frac{4h_{req}}{y}\right) & \text{if } y < 4h_{min} \\ \sqrt{y g_{enc}},\ 45^\circ & \text{if } y \geqslant 4h_{min} \end{cases} \tag{5-49}$$

The actual ballistic trajectory starts as soon as the lander has reached v_0 using the maximal available thrust, and the main engine is switched off. The S-frame is defined such that the target is located on the Y_S-axis; any motion along the Z_B-axis is automatically a cross-range error, because the spacecraft ideally moves only in the $X_S Y_S$-plane.

5-1-5 Quadratic Guidance Repositioning

The quadratic guidance logic in combination with the procedure to find an appropriate time-to-go (see Section 5-1-2) is capable to define a reference trajectory for the repositioning between two points on the surface. The lander is initially guided to the desired maximum altitude, and only then to the actual target point. The guidance scheme is generally not fuel optimal, but it is reliable and does not involve extreme maneuvers and high thrust levels. The latter is in fact a critical factor in case the center of mass is not very close the main engine's thrust vector, because the control system is not able to cope with the disturbing moments. This is discussed in more detail in Chapter 8.

The S-frame for the quadratic guidance repositioning is defined in the same way as for the ballistic repositioning discussed in the previous section. In this way, the intermediate target consists only of the desired maximum altitude ($+X_S$-axis), a downrange δ, and a downrange velocity v_δ (both $+Y_S$-axis).

The intermediate target position is chosen to be at the required altitude h_{req} as defined in the previous section, at a downrange equal to half the desired jump distance y_t, with zero cross-range. The intermediate target velocity consists of a downrange velocity equal to the ballistic velocity, which would bring the lander to final target position, and a cross-range and downward velocity both equal to zero. The intermediate target state $\mathbf{x}_{t,1}$ then can be written as

$$\mathbf{x}_{t,1} = \left[h_{req},\ \frac{1}{2} y_t,\ 0,\ 0,\ \frac{1}{2} y_t \sqrt{\frac{g_{enc}}{2h_{req}}},\ 0 \right].$$

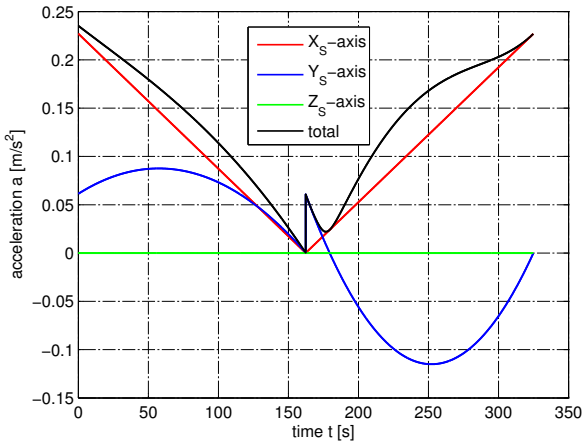
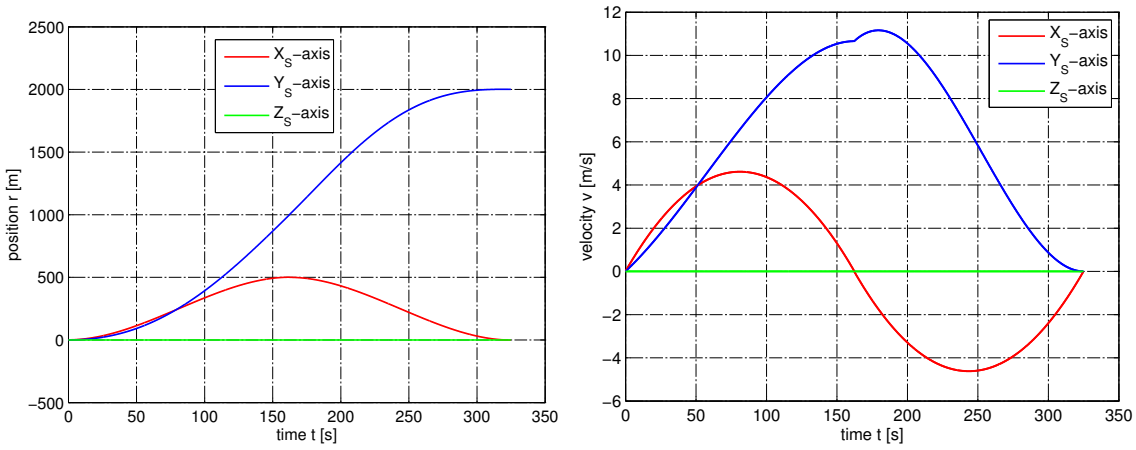


Figure 5-19: Quadratic guidance acceleration commands



(a) position **(b)** velocity

Figure 5-20: Position and velocity as a result of the acceleration shown in Fig. 5-19

This choice is generally not fuel optimal within the framework of the quadratic guidance logic, but some simple grid search optimization applied to the nonzero parameters of $\mathbf{x}_{t,1}$ indicated, that the fuel consumption only changes by a few percent – at the expense of the trajectory smoothness.

Figure 5-19 shows the acceleration commands for a jump distance y_t of 2000 m and a required altitude h_{req} of 500 m. The intermediate and final targets are both reached as illustrated in figures 5-20a and 5-20b. The acceleration curves are not symmetrical, because the initial conditions for the time-to-go search are different at the beginning of each guidance phase. The fuel consumption for this example is about 6.5 kg, while the ideal ballistic repositioning requires 3.3 kg. The purely quadratic guidance repositioning and the combined ballistic and quadratic guidance repositioning are compared in Section 8-4.

5-1-6 Guidance Phases

A guidance system commonly consists of several routines that are activated during different mission phases. This strategy is necessary, because each mission phase has different guidance requirements. The largest part of the descent trajectory should be designed with the focus on fuel-efficiency. However, perturbations and imperfect control will always lead to deviations from the nominal trajectory. These errors can be fatal during the landing phase, so there will always be some kind of trade-off between an optimal minimum fuel trajectory and a robust trajectory. The gravity turn guidance logic cannot be used as an exclusive guidance algorithm without modifications, as the final conditions are fully defined by the (uncertain) initial conditions. The sub-optimal quadratic guidance uses the current state of the vehicle to calculate the optimal steering commands to reach the (fixed) target. This guidance logic has a higher robustness, especially in environments where not all disturbing accelerations are exactly known Huang and Wang, 2007. During the terminal guidance phase, the focus is on the elimination of the remaining translational and rotational velocities – for a safe touchdown, large maneuvers close to the surface should be avoided.

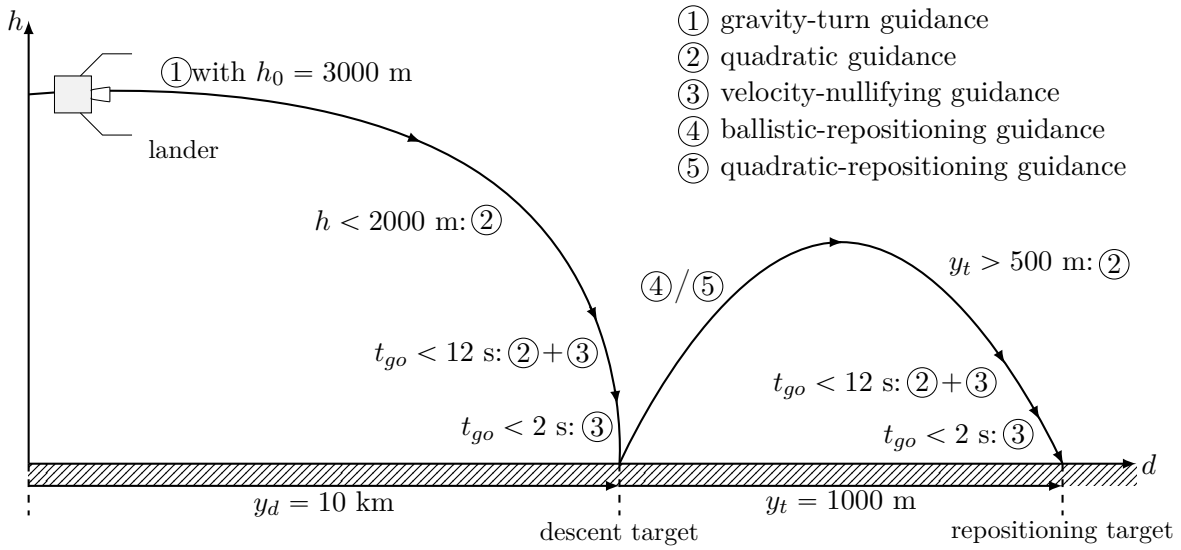


Figure 5-21: Guidance phases

The surface topography of Enceladus imposes strict requirements on the guidance system. As discussed earlier, the surface near the south-polar region is relatively hilly, covered with ice chunks and fine fault lines, so the landing accuracy should be better than 10 m. The landing gear can handle a maximal total landing velocity of 2.0 m/s, with maximal efficiency for vertical touchdowns. The lander's final GNC system incorporates an autonomous hazard avoidance system, which uses a LIDAR to analyze the landing area and to determine whether the surface topography allows for a safe touchdown. The hazard-avoidance system is discussed in Section 6-2 of the next chapter.

Five descent guidance logics have been discussed in this section: the gravity turn, the quadratic guidance law, and the velocity nullifying guidance law for the descent phase, and ballistic and quadratic guidance for repositioning. Figure 5-21 illustrates the active guidance

logics during the descent and repositioning phases, including the switch conditions, which were derived in the previous sections.

5-1-7 Simulator Configuration: Guidance System Parameters

The Enceladus Lander Simulator incorporates all guidance systems discussed in the previous sections. The identification numbers listed in Table 5-4 are used in the program code and in the simulator configuration files to refer to a specific guidance logic.

Logic Number	Description	Comment
0	no guidance	no thruster active
1	gravity turn, spherical model	see Section 5-1-1
2	quadratic guidance	see Section 5-1-2
3	velocity nullifying guidance	see section 5-1-3
4	gravity turn, flat model	see Section 5-1-1
5	combination of 2 and 3	for X_S-axis and $Y_S Z_S$-plane, resp.
6	ballistic repositioning	see Section 5-1-4
7	quadratic repositioning	see Section 5-1-5

Table 5-4: Identification numbers for the currently active guidance logic

The guidance system can be programmed to use one or a combination of different guidance logics in a specific order. The guidance modes available in the Enceladus Lander Simulator are shown in Table 5-5. Unless specified otherwise, all descent simulations in Chapter 8 use mode 4, thus a sequence of gravity turn, quadratic guidance and velocity nullifying guidance. The conditions to switch from one logic to the next one are defined in terms of altitude or time-to-go, and can be specified in the configuration files. The repositioning simulations use either mode 5 or 6 at the beginning, but automatically switch to mode 4 for the terminal guidance phase. Only modes 1, 2 and 3 never switch between guidance logics. The guidance mode identification number is a required input for the simulator configuration file. The simulator automatically changes the currently active guidance logic based on the lander state and the simulator configuration.

Mode Number	Description	Comment
1	gravity turn only	—
2	quadratic guidance only	—
3	velocity nullifying guidance only	test mode
4	gravity turn → quadratic → velocity nullifying	—
5	hybrid ballistic-quadratic repositioning	–
6	quadratic guidance repositioning	–

Table 5-5: Guidance system modes

The guidance-system related simulator parameters can be found in Table 5-6. The variable names are chosen such that their meaning in the context of this chapter should be clear, but the references to particular report sections or equations are listed nonetheless.

Variable Name	Comment
General	
guidanceFrequency	[Hz]
guidanceMode	[-], see Table 5-5
Gravity Turn Guidance	
gravityTurnConstantThrust	on/off, see Fig. 5-4
timeOfSingularUpdate	[s], after sim. start
Quadratic Guidance	
altitudeForSwitchToQuadratic	[m], switch from logic 1 or 4 to 2
timeForSwitchToQuadratic	[s], alternative to previous entry
stepsizeTargetAccSearch	$[\frac{m}{s^2}]$, in a_t-range
fuelConsumptionCalculationStepSizeFactor	[-], for Eq. (5-39) and (5-40)
timeBetweenCfactorsUpdate	[s], for update Eq. (5-30)
timeBetweenT2GoUpdate	[s], for recalculation t_{go}
t2GoSearchOption	[-], Eq. (5-34), (5-35) or free a_t
guidanceOptionT2GoUnsuccessful	see Table 5-4
YZaxisT2GoLead	[s], see discussion Fig. 5-18c
minimalT2GoForCalculationCfactors	[s], for update Eq. (5-30)
t2GoWhenQuadraticGuidanceEnds	[s], then: logic 0 or 3
Velocity Nullifying Guidance	
nullifyingDuration	[s], Δt in Eq. (5-41) and (5-42)
nullifyingActivationT2Go	[s], or link with logic 5
xSaxisTargetVelocity	$[\frac{m}{s}]$, along X_S-axis
Hybrid Ballistic-Quadratic Repos. Guidance	
hopDistance	[m], y in Eq. (5-49)
altitudeRequirement	[m], h_{req} in (5-49)
durationLowThrustTurn	[s]
percentageMaxThrustDuringTurn	[%]
altitudeBallisticToQuadratic	[m], switch to logic 2 (and 5)

Table 5-6: Guidance system related program parameters, both in the lander and the simulator configuration file

5-2 Control

The lander's attitude control system consists of three elements: The quaternion controller (section 5-2-1) interprets the commanded attitude from the guidance system and determines the adequate torque for a reorientation maneuver. The required torque values are then passed to the thruster selection logic (section 5-2-3), which uses Tables and schemes to identify the required thrusters and the available thrust ranges. Finally, this information is passed to the pulse modulator (section 5-2-4), where an optimal series of thrust firings is calculated to meet the moment command about each axis.

5-2-1 Quaternion Controller

The lander's equations of rotational motion are implemented in terms of quaternions and their derivatives. Instead of translating the quaternion attitude back to Euler angles and design a control logic for the angles ψ, θ and ϕ, it is more obvious to use a quaternion controller. Additionally, a single eigenaxis rotation maneuver, as mentioned in Section 4-2-4, has lower maneuver time then the according successive rotation sequence about the body reference frame axes (Wie et al., 1989). A fast reaction to the control command is crucial for successful touchdown and repositioning. A basic proportional-derivative (PD) controller design for quaternions can handle large-angle commands, in contrast to a comparable PD-controller for Euler angles. The former system is thus applicable for a wider range of control commands, so the lander requires less specialized control handles.

The linear controller for the quaternion model will introduce a rest-to-rest eigenaxis rotation maneuver to minimize the deviation from the desired attitude. The basic equation of rotational motion is, see Eq. (4-62),

$$\mathbf{I} \cdot \dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{I} \cdot \omega = \tilde{\mathbf{M}}_{cm} = \mathbf{M}_c + \mathbf{M}_{rel}$$

The total moment $\tilde{\mathbf{M}}_{cm}$ about the body's center of mass is now extended by the *disturbing moment* $\mathbf{M}_d$, which accounts for any moment that is not included in the navigation filter, such as a thrust imbalance. $\mathbf{M}_d$ has a direct influence on the derivation of the control gains, but initially, it will be set to zero and only activated in case the attitude control shows an unexpected long-term behavior. The equation of rotation motion thus becomes

$$\mathbf{I} \cdot \dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{I} \cdot \omega = \mathbf{M}_c + \mathbf{M}_{rel} + \mathbf{M}_d \tag{5-50}$$

The relative moment $\mathbf{M}_{rel}$, or control input, is now chosen such that Eq. (5-50) can be brought to a form that allows a simple dynamical analysis. $\mathbf{M}_{rel}$ should counteract the gyroscopic term of Euler's equation, and introduce a PD-control for the term $\mathbf{I} \cdot \dot{\boldsymbol{\omega}}$:

$$\mathbf{M}_{rel} = \boldsymbol{\omega} \times \mathbf{I} \cdot \boldsymbol{\omega} - \mathbf{M}_c - \mathbf{K} \cdot \mathbf{q}_e - \mathbf{D} \cdot \boldsymbol{\omega}_e \tag{5-51}$$

The PD-control is based on the quaternion error $\mathbf{q}_e$ and the rotational rate error $\boldsymbol{\omega}_e$ (the desired value of $\boldsymbol{\omega}$ is zero, so actually $\boldsymbol{\omega}_e = \boldsymbol{\omega}$). Note, that in this case $\mathbf{q}_e$ consists only of the vector part of the complete error quaternion. The proportional gain matrix $\mathbf{K}$ and the derivative gain matrix $\mathbf{D}$ determine how the target state is approached. Wie (2008) shows that the control logic in Eq. (5-51) is globally asymptotically stable in case the result of $\mathbf{K}^{-1}\mathbf{D}$ is positive definite. This is always the case if $\mathbf{K} = k\mathbf{I}$ and $\mathbf{D} = d\mathbf{I}$ for $k > 0$ and $d > 0$. Equation (5-51) is then rewritten as

$$\mathbf{M}_{rel} = \boldsymbol{\omega} \times \mathbf{I} \cdot \boldsymbol{\omega} - \mathbf{M}_c - k\mathbf{I} \cdot \mathbf{q}_e - d\mathbf{I} \cdot \boldsymbol{\omega}_e \tag{5-52}$$

Combining Eq. (5-50) and (5-51) gives the closed-loop form of the system:

$$\mathbf{I} \cdot \dot{\boldsymbol{\omega}} + k\mathbf{I} \cdot \mathbf{q}_e + d\mathbf{I} \cdot \boldsymbol{\omega}_e = \mathbf{M}_d \tag{5-53}$$

The above equation is in fact a rotation about the eigenaxis, if the error quaternion $\mathbf{q}_e$ is the quaternion rotation from the current $\mathbf{q}$ to the commanded quaternion $\mathbf{q}_c$. If the controller used

for a rest-to-rest maneuver, then $\boldsymbol{\omega}_0 = \mathbf{0}$ and $\boldsymbol{\omega}$ may be interpreted as the rotation velocity $\dot{\theta}$ about the eigenaxis. The rotation about the normalized eigenaxis $\mathbf{e}$ is (see Eq. (4-3))

$$\mathbf{q} = \begin{pmatrix} \cos\frac{\theta}{2} \\ \mathbf{e}_1 \sin\frac{\theta}{2} \\ \mathbf{e}_2 \sin\frac{\theta}{2} \\ \mathbf{e}_3 \sin\frac{\theta}{2} \end{pmatrix} \tag{5-54}$$

which leads to the closed-loop expression for the eigenaxis rotation maneuver (see also Wie (2008)):

$$\ddot{\theta} + d\dot{\theta} + k \sin\frac{\theta}{2} = \left(\mathbf{I}^{-1}\mathbf{M}_d\right) \cdot \mathbf{e} \tag{5-55}$$

The right-hand side of Eq. (5-55) translates the disturbing moment to the according moment about the eigenaxis. As the orientation of $\mathbf{M}_d$ is independent of $\mathbf{e}$, the result of $(\mathbf{I}^{-1}\mathbf{M}_d) \cdot \mathbf{e}$ is time-dependent which leads to varying gains. This problem can be avoided by assuming the worst-case scenario, where $\mathbf{M}_d$ is acting in the same direction as $\mathbf{e}$, with a constant value m_{d_e}. As mentioned before, expression (5-55) is derived under the assumption of a rest-to-rest maneuver. The lander's guidance system constantly produces new attitude commands, so the initial rotational velocity for each control cycle is not zero. $\boldsymbol{\omega}$ will be small for safety and navigational reasons anyway, but the chosen control logic sets the actual limits of the maximal allowable rotational velocity.

Translating Eq. (5-55) into the Laplace domain, assuming a constant $\mathbf{M}_{d_0}$ about the eigenaxis and a small rotation angle θ so that $\texttt{sin}\ \theta \approx \theta$, leads to the equation

$$s^2\theta(s) + ds\theta(s) + \frac{k}{2}\theta(s) = \frac{\left(\mathbf{I}^{-1}\mathbf{M}_{d_0}\right) \cdot \mathbf{e}}{s} \tag{5-56}$$

For the case where $\mathbf{M}_{d_0}$ is not a null vector, the proportional gain depends on the maximal acceptable steady-state error of θ. Applying the final value theorem[2] on Eq. (5-56) (see Chu (2012)),

$$\lim_{t\to+\infty} \theta(t) = \theta_{ss} = \lim_{s\to 0} \theta(s) \cdot s = \lim_{s\to 0} \frac{\left(\mathbf{I}^{-1}\mathbf{M}_{d_0}\right) \cdot \mathbf{e}}{s^2 + ds + \frac{k}{2}} = \frac{2\left(\mathbf{I}^{-1}\mathbf{M}_{d_0}\right) \cdot \mathbf{e}}{k} \tag{5-57}$$

which leads to the proportional gain factor k

$$k = \frac{2\left(\mathbf{I}^{-1}\mathbf{M}_{d_0}\right) \cdot \mathbf{e}}{\theta_{ss}} \tag{5-58}$$

where θ_{ss} is the desired steady-state value of the Euler rotation angle θ. In case $\mathbf{M}_{d_0}$ is a null vector, the closed-loop expression in Eq. (5-55) can be compared with the expression for a damped harmonic oscillation (Wie, 2008)

$$\ddot{\theta} + d\dot{\theta} + k\frac{\theta}{2} = \ddot{\theta} + 2\zeta\omega_n\dot{\theta} + \omega_n^2\theta = 0 \tag{5-59}$$

[2]The final value theorem states, that in case $\lim_{n\to\infty} f(t)$ has a finite value, then it is equal to $\lim_{n\to 0} s \cdot F(s)$ in Laplace space.

where ζ is the damping ratio and ω_n the natural frequency. ζ is set to the optimal damping value of $\frac{\sqrt{2}}{2}$ (Chu, 2012). The standard relation between the settling time t_{set}, or maneuver time in this case, and ω_n and ζ is (Wie et al., 1989)

$$t_{set} = \frac{4}{\zeta \omega_n} \tag{5-60}$$

so comparing terms in Eq. (5-59) and using the expression for t_{set}, the proportional gain factor for the undisturbed case is

$$k = 2\omega_n^2 = 2\left(\frac{4}{\zeta t_{set}}\right)^2 = 2\left(\frac{8}{t_{set}\sqrt{2}}\right)^2 = \frac{64}{t_{set}^2} \tag{5-61}$$

The derivative gain factor d for both cases follows again from comparing terms in Eq. (5-59), which finally leads to the expression

$$d = 2\zeta\sqrt{\frac{1}{2}k} = 2\frac{\sqrt{2}}{2}\sqrt{\frac{1}{2}k} = \sqrt{k} \tag{5-62}$$

For a desired steady-state error of 0.001 in terms of quaternion elements — or, equivalently, a settle time t_{set} of 6.2 s—, the resulting PD gains k and d for the given constant disturbing and the given inertia tensor are 1.6587 and 1.2879, respectively.

The only missing element in the expression for the control torque (Eq. (5-51)) is an expression for the quaternion error $\mathbf{q}_e$. The quaternion error is the quaternion rotation between the desired attitude $\mathbf{q}_c$ and the current attitude $\mathbf{q}$. In quaternion form, this can be written as

$$\mathbf{q}_e = \mathbf{q}_c^{-1} \otimes \mathbf{q} \tag{5-63}$$

where $\otimes$ indicates the quaternion multiplication, and $\mathbf{q}_c^{-1}$ is the inverse of $\mathbf{q}_c$. The quaternion inverse for a unit quaternion only changes the sign of the vector part. In matrix form, Eq. (5-63) becomes

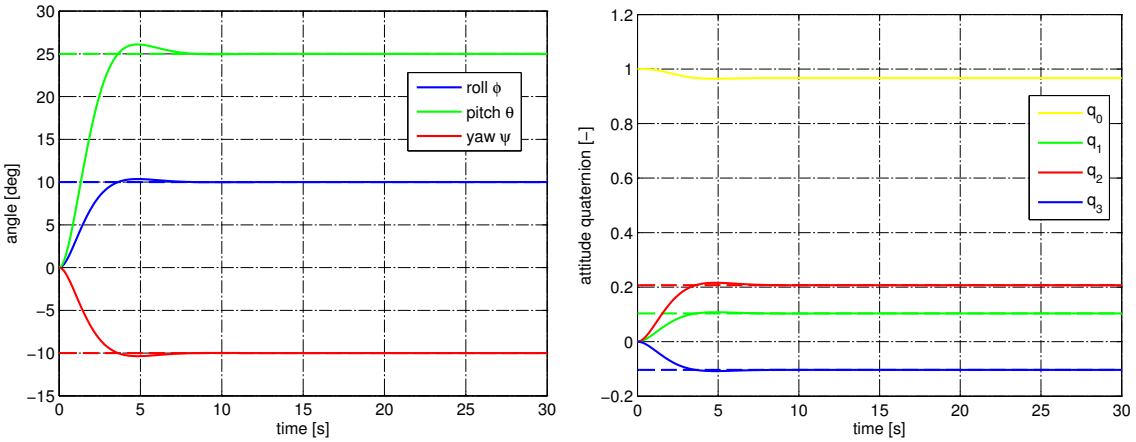
$$\begin{bmatrix} q_{e_0} \\ q_{e_1} \\ q_{e_2} \\ q_{e_3} \end{bmatrix} = \begin{bmatrix} q_{c_0} & q_{c_1} & q_{c_2} & q_{c_3} \\ -q_{c_1} & q_{c_0} & q_{c_3} & -q_{c_2} \\ -q_{c_2} & -q_{c_3} & q_{c_0} & q_{c_1} \\ -q_{c_3} & q_{c_2} & -q_{c_1} & q_{c_0} \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{5-64}$$

Note that for Eq. (5-50), (5-51) and (5-52), the error quaternion only consists of the vector elements of $[q_{e_0}, q_{e_1}, q_{e_2}, q_{e_3}]$, so $\mathbf{q}_e = [q_{e_1}, q_{e_2}, q_{e_3}]^{\mathrm{T}}$.

Figure 5-22 shows the response of the lander model to a commanded attitude change of $[\phi_c, \theta_c, \psi_c] = [50^\circ, 25^\circ, -10^\circ]$, in absence of any disturbances, for $k = k_0 = 1.6587$ and $h = \sqrt{k_0} = 1.2879$. The inertia tensor I is diagonal with $I_{xx} = I_{yy} = I_{zz} = 150\ \mathrm{kgm}^2$ (see Table 3-1), and the control is proportional and unlimited, so the control torque $\mathbf{M}_{rel}$ is directly inserted in the equations of rotational motion. By close inspection of Fig. 5-22b, the settling time t_{set} is with about 8 s above the expected 6.2 s. This is a consequence of the approximation $\texttt{sin}\ \theta \approx \theta$ in Eq. (5-56), which is not accurate for large rotation angles. Reducing the attitude command by a factor 10 or more leads to a convergence below t_{set}.

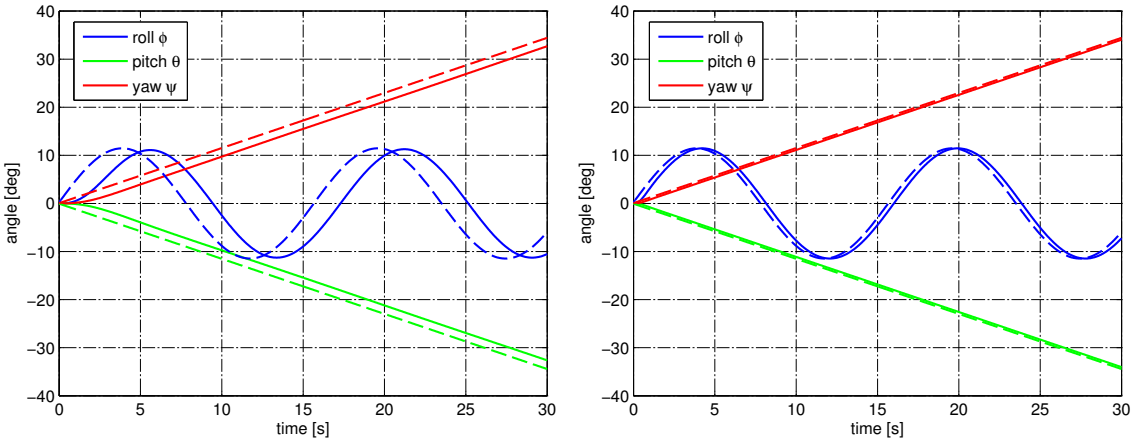
Figure 5-23a shows the response of the same lander model to a varying attitude command $[\phi_c, \theta_c, \psi_c] = [0.2\sin(0.4t),\ -0.02t\ , 0.02t]$ rad. With the original proportional gain k_0, the

spacecraft's true attitude lags behind the commanded attitude. This problem can be solved by increasing k, see Fig. 5-23b, where $k = 20 \cdot k_0$. A disadvantage is obviously the higher required torque levels: The maximal torque for the first case is 106 Nm and 2130 Nm for the second (both about the $+X_B$-axis).



(a) Euler angles **(b)** Quaternions

Figure 5-22: Constant attitude command (dashed lines) and true attitude (solid lines)



(a) k = 1.6587 **(b)** k = 20 · 1.6587=33.174

Figure 5-23: Attitude command (dashed lines) and true attitude (solid lines) for different proportional gains. d is equal to $\sqrt{k}$ in both cases.

5-2-2 Attitude Command Generation

The guidance system returns an acceleration command with respect to the I-frame. The lander's main engine points in the direction of the $+Z_B$-axis, so the B-frame – or, the lander itself – must always be oriented such that the acceleration command vector and the $-Z_B$-axis are collinear and point in the same direction. Close to the target, the control system includes

the attitude verniers in the translational control, but at that instance of time, the flight-path angle is already -90°.

The lander attitude as part of the state vector is expressed in terms of a quaternion rotation from the I-frame to the B-frame, or $\mathbf{q}_{B\leftarrow I}$. According to the theory discussed in Section 4-2-4, it is possible to bring a body from any initial orientation to the desired orientation, given the correct rotation axis, or eigenaxis, $\mathbf{e}$, and the rotation angle θ. The eigenaxis for a rotation to align the two known vectors $\mathbf{a}_{\text{cmd,I}}$ and $\mathbf{x}_{\text{-Z}_B\text{,I}}$ follows from the normalized cross product

$$\begin{aligned}\mathbf{e} &= |\mathbf{a}_{\text{acc,I}} \times \mathbf{x}_{\text{-Z}_B\text{,I}}| \\ &= \left| \mathbf{a}_{\text{acc,I}} \times \mathbf{T}_{I\leftarrow B} \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix} \right|. \end{aligned} \tag{5-65}$$

The general formula for the cross product of two vectors $\mathbf{a}$ and $\mathbf{b}$ is (Török, 2000)

$$\mathbf{a} \times \mathbf{b} = |a|\,|b| \sin\theta \mathbf{n} \tag{5-66}$$

which leads to the corresponding rotation angle

$$\theta = \sin^{-1}\left(\frac{\mathbf{a}_{\text{acc,I}} \times \mathbf{x}_{\text{-Z}_B\text{,I}}}{\|\mathbf{a}_{\text{acc,I}}\|\,\|\mathbf{x}_{\text{-Z}_B\text{,I}}\|} \right). \tag{5-67}$$

Equation (5-65) and (5-67) now fully define the quaternion rotation error $\mathbf{q}_{\text{err}}$ (see also Eq. (4-3)):

$$\mathbf{q}_{\text{err}} = \begin{pmatrix} \cos\frac{\theta}{2} \\ \mathbf{e}_1 \sin\frac{\theta}{2} \\ \mathbf{e}_2 \sin\frac{\theta}{2} \\ \mathbf{e}_3 \sin\frac{\theta}{2} \end{pmatrix} \tag{5-68}$$

The commanded attitude $\mathbf{q}_c$ then becomes

$$\mathbf{q}_c = \mathbf{q}_{\text{err}} \otimes \mathbf{q}_{B\leftarrow I}. \tag{5-69}$$

One problem in the above derivation of $\mathbf{q}_c$ is the direct application of θ on $\mathbf{e}$: The definition of the cross product in Eq. (5-66) involves the unit direction vector $\mathbf{n}$. This vector is parallel to $\mathbf{e}$, but may point in the opposite direction, leading to a wrong rotation angle if the system is rotated about $\mathbf{e}$. Both Eqs. (5-65) and (5-66) are unambiguous on their own, so there exists a combined mathematical expression. The most practical solution at this point is to determine whether the commanded attitude $\mathbf{q}_c$ given $\mathbf{q}_{B\leftarrow I}$ in fact leads to an orientation where $\mathbf{a}_{\text{cmd,I}}$ and $\mathbf{x}_{\text{-Z}_B\text{,I}}$ are collinear and point in the same direction (the unit direction vectors must be equal). If this is not the case, $\mathbf{e}$ should point in the opposite direction.

This attitude command does not control the orientation of the lander about the Z_B-axis. A real lander probably has an additional requirement for the pointing direction of the X_B-axis due to the LIDAR's field-of-view. For the Enceladus Lander Simulator, the orientation of the Y_B-/Z_B-axis is only important as soon as the attitude verniers are used for translational control – which is not the case with the current velocity nullifying guidance setup. Simulations

in MATLAB with the basic control system indicated that it is possible to use the same principle as shown in the previous equations to align the Y_B-axis with the Y_S-axis. This results in two different $\mathbf{q}_c$, which do not conflict – i.e., address the same thrusters – as long as the rotation axes are perpendicular to each other. This is only true during the terminal-descent phase. A reorientation during the approach phase will require a more advanced system that trades opposing thrust commands based on the current lander state.

5-2-3 Thruster Selection Process

The Enceladus lander has four attitude thrusters for each axis. To eliminate the effects on the translational motion, they are activated in pairs, firing in opposite directions at opposite locations on the lander. This idealized design was chosen to limit the complexity of the lander simulator. However, it offers, in fact, already a certain degree of redundancy: in case one thruster fails, attitude control about that axis is still possible, but with small effects on the translational motion. As will be explained in the next section in more detail, verniers are either activated or deactivated. Consequently, they always produce a fixed torque level. The lander's 12 attitude thrusters can produce two different torque levels about each axis. The moment arms and thrust levels are the same for all verniers, so the torque produced by two verniers is twice the torque produced by one.

The thruster selection logic can either be based on a linear programming technique or on a phase-plane system.
The linear programming approach is particularly useful if the spacecraft has a large number of verniers that are not aligned with the body axes. The result is always the optimal thruster combination for a commanded attitude change. The attitude control system of the Enceladus lander produces a moment command $\mathbf{M}_{rel}$, which can be translated into a thrust about each body axis. Each thrust level corresponds to one or two out of four available thrusters - or none at all. Linear programming is not required for this relatively simple problem.
The second selection logic type defines regions in the phase-plane plot of θ_e and ω_e, each corresponding to a predefined thruster combination. The Space Shuttle's reaction control system includes 38 thrusters (Wie, 2008), which are activated by means of a phase-plane logic for each axis, such as shown in Fig. 5-24. This phase-plane defines nine regions, some of which directly initiate thruster firings, and others use additional information to determine the appropriate actions. The phase-plane logic was used in Apollo and Space Shuttle mission (Wie, 2008), and can have a high level of detail due to the various control logics for each thruster combination. The Enceladus lander uses a modification of the phase-plane logic. It is not based on the error angle and the error rotational rate, but on the required torque $\mathbf{M}_{\text{ref}}$ from the quaternion controller. The selection logic first identifies the thruster or thruster combination that is responsible for rotation direction that is demanded by $\mathbf{M}_{\text{ref}}$. Then, based on the magnitude of each element of $\mathbf{M}_{\text{ref}}$, it is determined whether the torque about each axis should be generated by one or two verniers. The selection logic will always designate either one or two verniers out of the pool of four verniers for each axis. The thrusters are not activated by the thruster-selection logic. Instead, the theoretical torque levels are passed to the pulse modulator (section 5-2-4), which determines how the identified thrusters should be fired to approximate $\mathbf{M}_{\text{ref}}$.

The exact location of each vernier is shown in Fig. 3-2b. From this figure, and the vernier characteristics given in Table 3-3, it possible to allocate a rotational direction and a potential

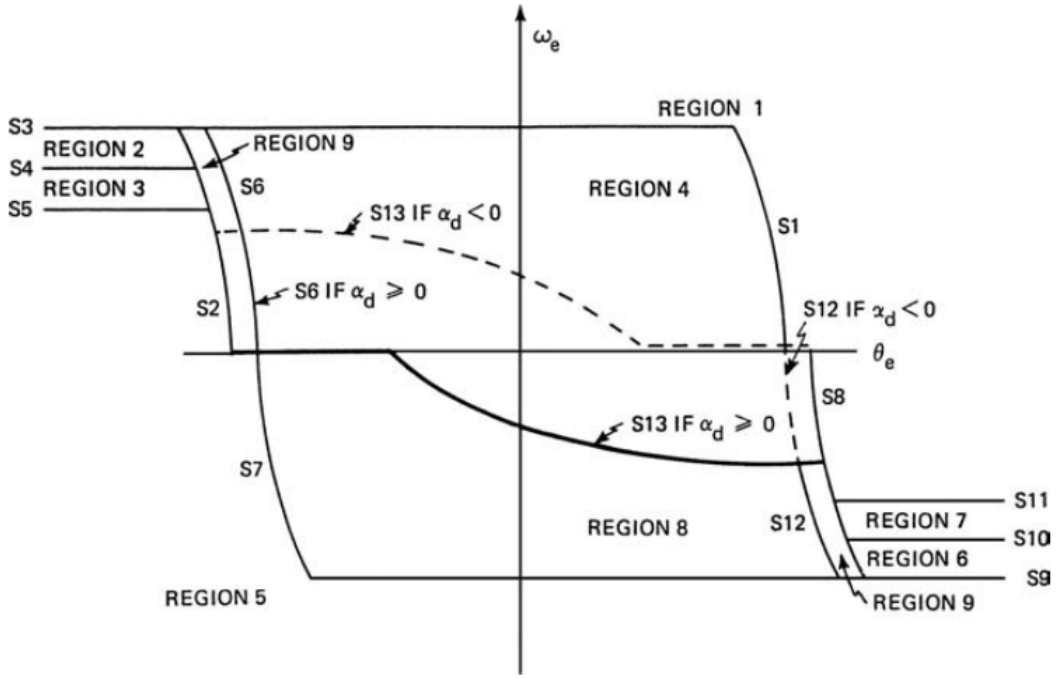


Figure 5-24: Thruster selection phase-plane logic of the Space Shuttle (Wie, 2008)

torque to each attitude thruster. Table 5-7 summarizes this information. As mentioned in Section 3-2, the moment arm for verniers responsible for the pitch control is larger than the ones for the roll and yaw control (L and $\frac{\sqrt{3}}{2}L$, respectively). This leads to a slightly higher potential torque about the X_B-axis. The torque level for each axis is now either the torque produced by one thruster, or the torque produced by two thrusters - which is double the value of the other case.

Rotation Axis	Vernier Number	Torque Level 1	Torque Level 2
$+X_B$	1, 3	4.80 Nm	9.60 Nm
$-X_B$	2, 4	4.80 Nm	9.60 Nm
$+Y_B$	5, 7	4.16 Nm	8.36 Nm
$-Y_B$	6, 8	4.16 Nm	8.36 Nm
$+Z_B$	9, 11	4.16 Nm	8.36 Nm
$-Z_B$	10, 12	4.16 Nm	8.36 Nm

Table 5-7: Attitude thruster torque levels. Step 1 of the thruster selection process. See Table 3-3 for naming and positioning

The identification of the correct thruster combination and their theoretical torque levels conclude the first step of the thruster selection process. The torque level passed to the pulse modulator depends on the values of $[M_{\text{ref}_1}, M_{\text{ref}_2}, M_{\text{ref}_3}]$. For both positive and negative rotations, the logic returns the lower torque level 1 for absolute values below 4.80/4.16 Nm, and the higher torque level 2 for all other values. The complete phase logic for the Enceladus lander is shown in tabulated form in Table 5-8. The torque-level selection is the second

and last step of the thruster selection process. The vernier identification numbers and the theoretical torque levels can now be passed to the pulse modulator for the determination of the firing sequence. The logic may be expanded and refined based on the attitude thruster characteristics. In case the simulations show that more powerful verniers are required, the values in Table 5-7 can be modified accordingly. If proportional thrusters seem to have more advantages, both Tables change, and a phase-plane plot logic in the form of Fig. 5-24 might become necessary, because the selection logic might activate the thrusters directly for certain thrust ranges.

Name	$\mathbf{M}_{\text{ref}}$ boundaries lower	upper	Torque Passed
Region 1	$\geq$ 0	$\leq$ Level 1	Level 1
Region 2	$>$ Level 1	$+\infty$	Level 2
Region 3	$\geq -$Level 1	$<$ 0	Level 1
Region 4	$-\infty$	$<-$Level 1	Level 2

Table 5-8: Torque levels passed to the pulse modulator for a given $\mathbf{M}_{\text{ref}}$. Step 2 of the thruster selection process.

5-2-4 Pulse-Width Pulse-Frequency Modulator

The thrust level of attitude verniers is usually not freely controllable, because the design of proportional valves is complex, and the thruster itself is relatively small. Furthermore, low thrust levels cause dirt particles to stick to the valve opening and block their complete closure (Wie, 2008). This results in leakage and in disturbing forces, which can eventually endanger the entire mission. The valves are thus either open or closed, so the thruster produces its rated thrust or no thrust at all. Pulse modulation is a technique to approximate the commanded thrust level by a series of thrust pulses. The vernier valves have a high operational speed that allows a wide range of pulse widths. Modification of the pulse frequency can lead to a lower number of required thrust pulses, which will reduce the wear-out of the verniers. The pulse-width pulse-frequency (PWPF) modulator can influence both characteristics, and thus optimize for a specific objective, such as fuel consumption or number of thruster firings. The basic PWPF modulator shown in Fig. 5-25 is a combination of the schemes given in Wie (2008) and McCelland (1994). The PWPF modulator consists of a first-order lag filter in front of a Schmitt trigger, both inside a feedback loop. U_{on} and U_{off} are the cut-in and cut-out values. The Schmitt trigger returns the predefined trigger output U_{out} in case the input value is larger than U_{on}, and output $-U_{\text{out}}$ in case the input value is smaller then $-U_{\text{on}}$. The trigger incorporates hysteresis, which means, that a previous U_{out} or $-U_{\text{out}}$ is deactivated by the according U_{off} or U_{off}, respectively. The filter gain K_m, the filter time constant T_m and the dead-band size $h = U_{\text{on}} - U_{\text{off}}$ are the modulator parameters. The values of these parameters can follow from an optimization process, a trial-and-error approach or from tabulated standard values. Note that the hysteresis region should be between U_{on} and $-U_{\text{on}}$ — in that case, the trigger would be unable to return a zero output, and the PWPF modulator becomes a bang-bang controller. This condition can be written as

$$0 < h < 2U_{\text{on}} \tag{5-70}$$

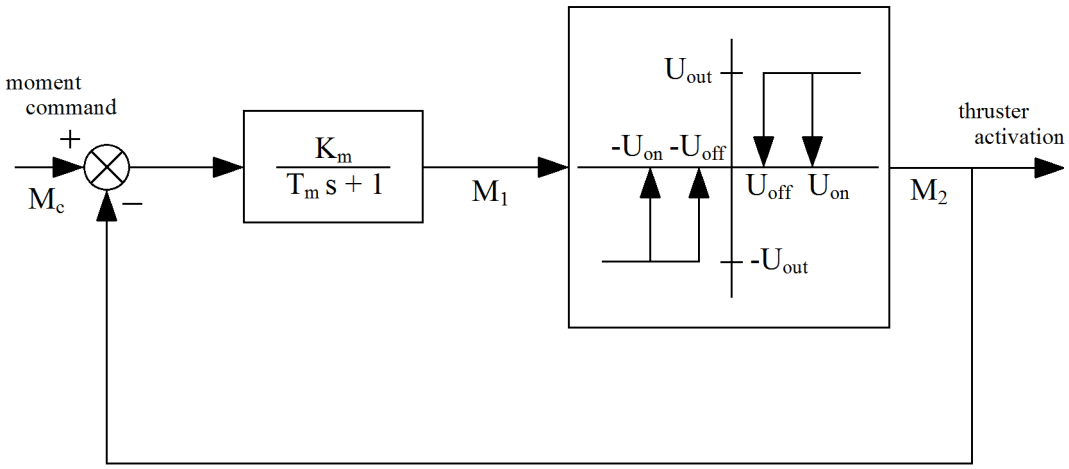


Figure 5-25: Pulse-Width Pulse-Frequency Modulator

The output of the filter element in Laplace space is, by inspection of Fig. 5-25,

$$M_1(s) = [M_c(s) - M_2(s)] \frac{K_m}{T_m s + 1} \tag{5-71}$$

which can be written as

$$[M_c(s) - M_2(s)] K_m = M_1(s) + sM_1(s)T_m \tag{5-72}$$

Equation (5-71) can be translated back to the time domain by applying the inverse Laplace transform. For the two time steps k and $k+1$, the transformation $\mathcal{L}^{-1}\{sM_1(s)T\}$ is the time derivative of $M_1(t)T_m$. As T_m is constant, the discrete time-step version of Eq. (5-72) becomes (McCelland, 1994):

$$[M_c(k+1) - M_2(k)] K_m = M_1(k) + \frac{M_1(k+1) - M_1(k)}{\Delta t} T_m \tag{5-73}$$

where Δt is the PWPF sampling time. The vernier's minimum impulse bit for the maximal thrust of 6 N is 0.3 Ns (see Table 3-3), so the idealized minimum activation time is 0.05 s. If Δt is below this value, the modulator might try to set the thrust to 0, which is not possible for 0.05 s after the activation. Consequently, Δt is set to 0.05 s in the Enceladus lander simulator. The PWPF modulator output, however, changes for different values of Δt, because a larger sample time leads to a delayed response: the filter output is determined every Δt, so the actual crossing of U_{on} or U_{off} might be recorded late. The modulator output converges for small values of Δt. The filter output at the time instance $k+1$ follows from solving Eq. (5-73) for $M_1(k+1)$(McCelland, 1994):

$$M_1(k+1) = [M_c(k+1) - M_2(k)] K_m \frac{\Delta t}{T_m} + M_1(k) \left[1 - \frac{\Delta t}{T_m}\right] \tag{5-74}$$

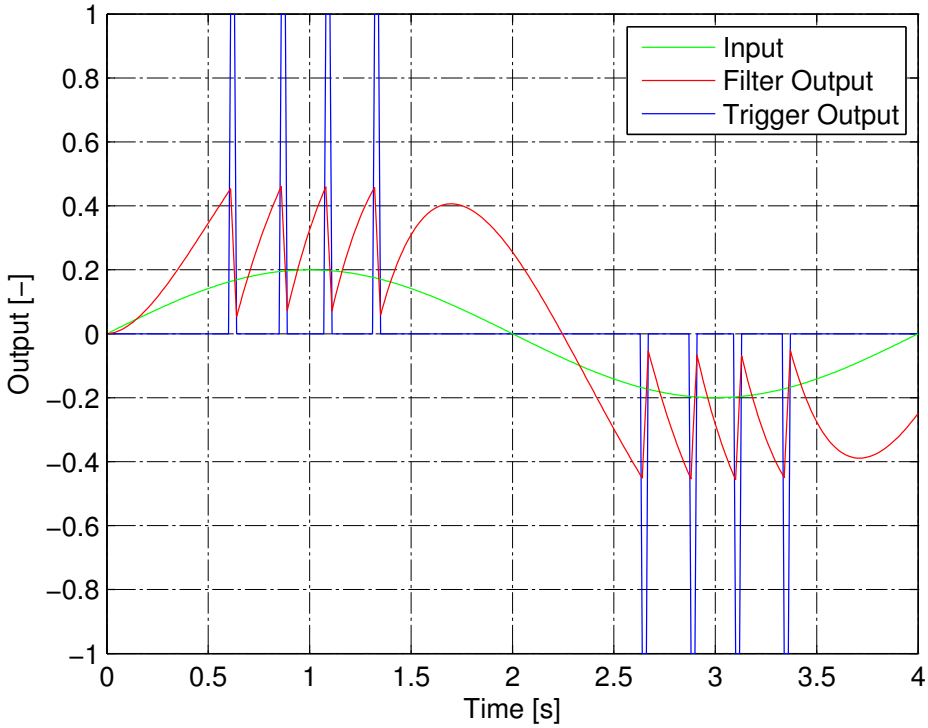


Figure 5-26: PWPF output for a sine wave as input

The current value $M_1(k+1)$ is fed in the Schmitt trigger, whose output $M_2(k+1)$ is conditional and can be written as:

$$M_2(k+1) = \begin{cases} U_{\text{out}} & \text{if } M_1(k+1) \geq U_{\text{on}} \\ 0 & \text{if } |M_1(k+1)| < U_{\text{on}} \wedge \\ & \quad \big[\big(M_2(k) = U_{\text{on}} \wedge M_2(k+1) < U_{\text{off}}\big) \\ & \quad \vee \big(M_2(k) = -U_{\text{on}} \wedge M_2(k+1) > -U_{\text{off}}\big)\big] \\ -U_{\text{out}} & \text{if } M_1(k+1) \leq -U_{\text{on}} \end{cases} \tag{5-75}$$

Equation (5-74) and (5-75) form the basis of the code implementation in both `MATLAB` and `C++`. The former program is used for the PWPF-modulator configuration, the results are then inserted in the Enceladus Lander Simulator. The correct implementation has been checked by comparing the results for specific inputs with the plots shown in McCelland (1994) and Krøvel (2005). A representative PWPF modulator output for sine-wave input with a 0.2 amplitude is shown in Fig. 5-26. The filter output rises until it reaches $U_{\text{on}} = 0.45$, which triggers the output $U_{\text{out}} = 1$. Due to the feedback loop, this in turn leads to a decreasing filter output until it falls below $U_{\text{off}} = 0.15$, setting the trigger output to zero. The PWPF modulator does not respond to input values with an absolute value below $\frac{U_{\text{on}}}{K_m}$, as the filter output does not reach U_{on}. On the other hand, the modulator will reach saturation — thus, a constant output of $\pm U_{\text{out}}$ — when the input is above $U_{\text{out}} + \frac{U_{\text{off}}}{K_m}$ (Krøvel, 2005). The saturation level can be used in the thruster-selection logic as an indication to use a higher thrust level. The lower and upper input boundaries influence the pointing accuracy of the lander. A large K_m and small $\pm U_{\text{on}}$ will have a positive effect on the attitude error.

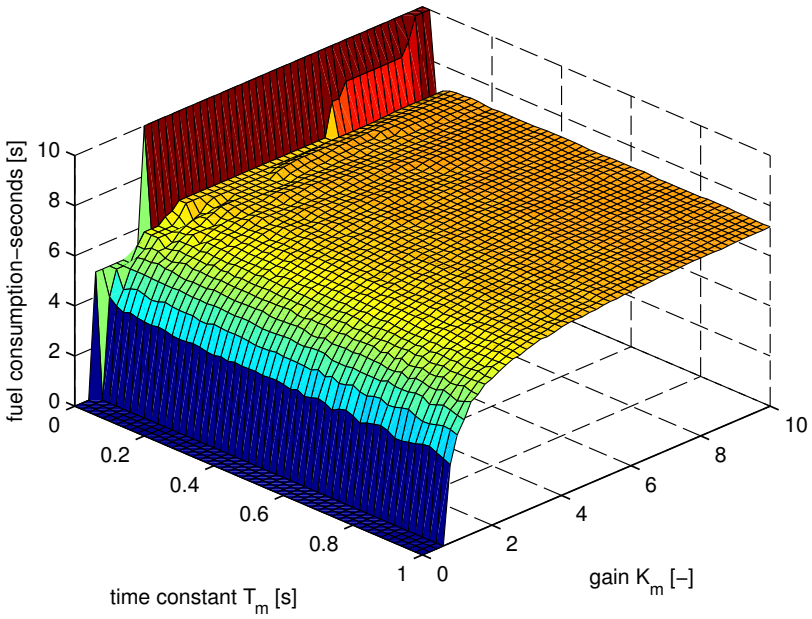


Figure 5-27: Fuel consumption for different values of K_m and T_m

Name	Symbol	Optimization Range	Nominal Value
cut-in value	U_{on}	0 to 1	0.45
hysteresis	h	0 to 2	0.3
filter gain	K_m	0 to 10	4.5
time constant	T_m	0.02 to 1 s	0.15 s
command input	M_c	—	0.75
trigger output	U_{out}	—	1
sampling time	Δt	—	0.01 s
simulation time	T_f	—	10 s

Table 5-9: PWPF modulator parameters for the basic optimization process

The filter gains K_m and T_m and the Schmitt trigger factors U_{on} and U_{off} will be determined using two optimization processes that aim at minimizing the fuel consumption and the number of required thruster pulses. This principle is based on Krøvel (2005), who executes the optimization process on a static, dynamic and full system model. In this section, the investigation is limited to the PWPF modulator as shown in Fig. 5-25 for a step input and without a system model. The full control system, consisting of the PD controller, the thruster selection and the PWPF modulator, will be tested and configured at the end of this chapter (section 5-2-5).

The PWPF modulator parameters for the basic optimization process are given in Table 5-9. The nominal values indicate the values used when the corresponding parameters are not an optimization variable. Note that the time constant T_m cannot be zero due to the fraction $\frac{\Delta t}{T_m}$ in Eq. (5-74). The values in Table 5-9 are equal to those suggested in Krøvel (2005), so

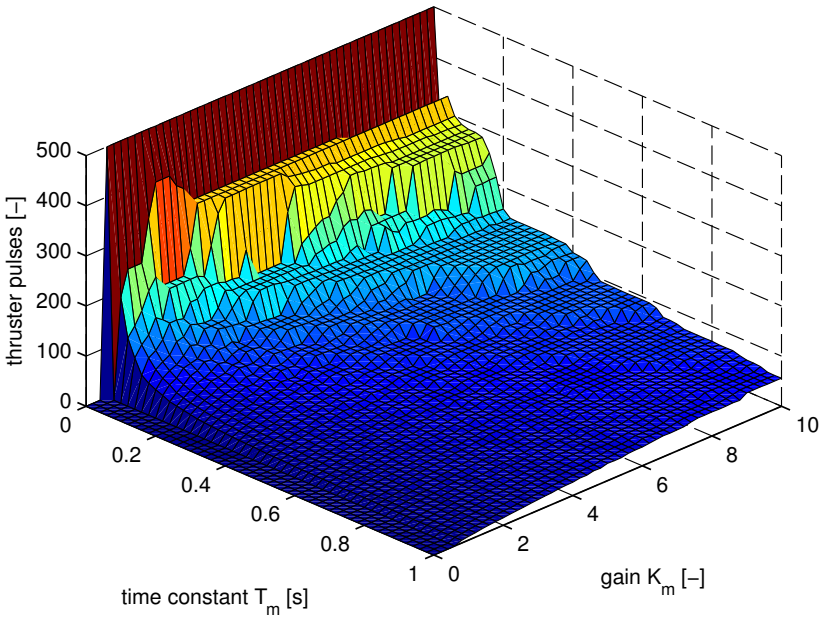


Figure 5-28: Thrust pulses about X_B-axis for different values of K_m and T_m

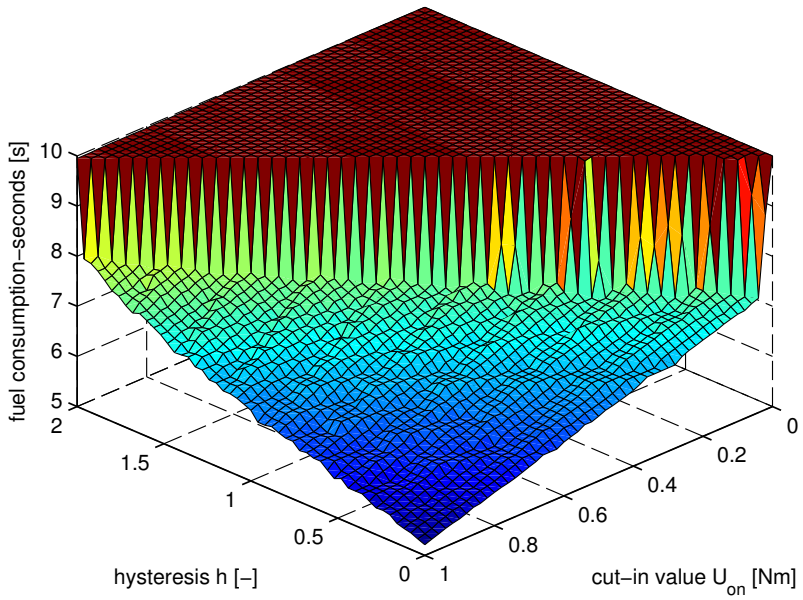


Figure 5-29: Fuel consumption for different values of U_{on} and h

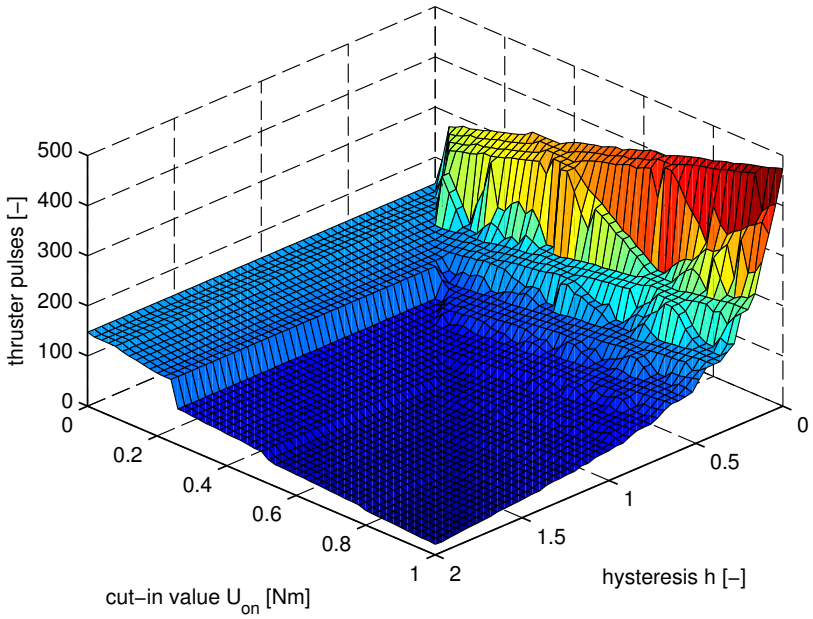


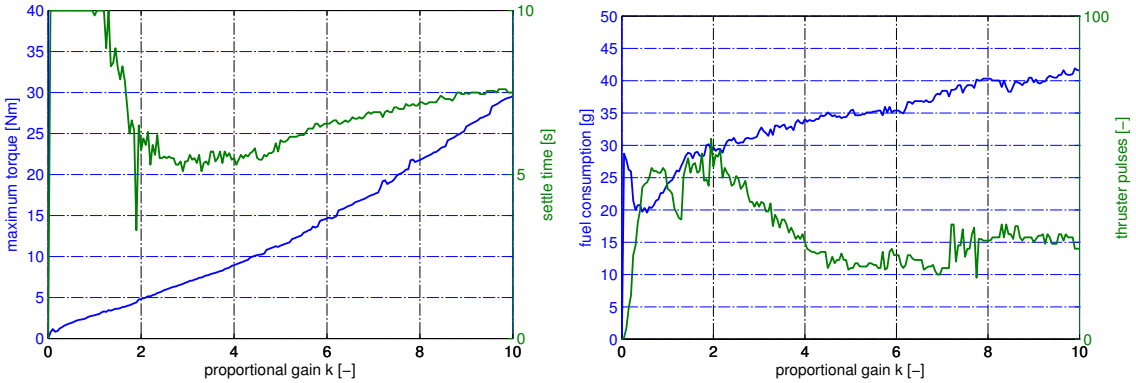
Figure 5-30: Thrust pulses about X_B-axis for different values of U_{on} and h

the simulation results can be used as an additional unit test for the PWPF modulator. U_{off} follows from the previously mentioned relation $h = U_{\text{on}} - U_{\text{off}}$. h is chosen instead of U_{off} due to the relation in Eq. (5-70).

Figure 5-27 and 5-28 show the fuel consumption and number of thruster pulses for a K_m-range of [0,10] and a T_m range of [0.02,1] s. The fuel consumption is measured in terms of consumption-seconds, so for an actual mass indication, this number must be multiplied by the thruster mass flow. The fuel consumption increases drastically for values of $T_m < 0.05$ s, and, on the other hand, decreases for filter gains below 3. The pulse number can be inferred from changes in the Schmitt trigger output, because a jump from 0 to $\pm U_{\text{out}}$ or back indicates a valve activation inside the corresponding vernier. The pulse number is thus the number of changes divided by two, as a pulse consists of an activation and a deactivation. The number of thruster pulses can be decreased by choosing a large value for T_m in combination with a small value for K_m.

Figure 5-29 and 5-30 show the fuel consumption and number of thruster pulses, respectively, for a U_{on}-range of [0,1] and a h range of [0,2]. The fuel consumption is lower for larger U_{on}, which on the other hand increases the thruster pulses. A similar contrary behavior can be seen for the hysteresis range: A large h decreases the thruster pulses, but increases the fuel consumption.

Note that in both Fig. 5-29 and 5-30, all results for which $h > 2U_{\text{on}}$ are in fact invalid according to Eq. (5-70). In those cases, the output cannot become zero, so the fuel consumption is always equal to the simulation time. Krøvel (2005) treats these parts of the plot differently, which leads to a mirroring of the fuel consumption results along the line $h = 2U_{\text{on}}$ in Fig. 5-29. Other than that, all optimization results in this section are very similar the results shown in (Krøvel, 2005).



(a) maximal reference torque and settle time **(b)** fuel consumption and thruster pulses

Figure 5-31: Control system performance for $d = \sqrt{k}$ and different values of k

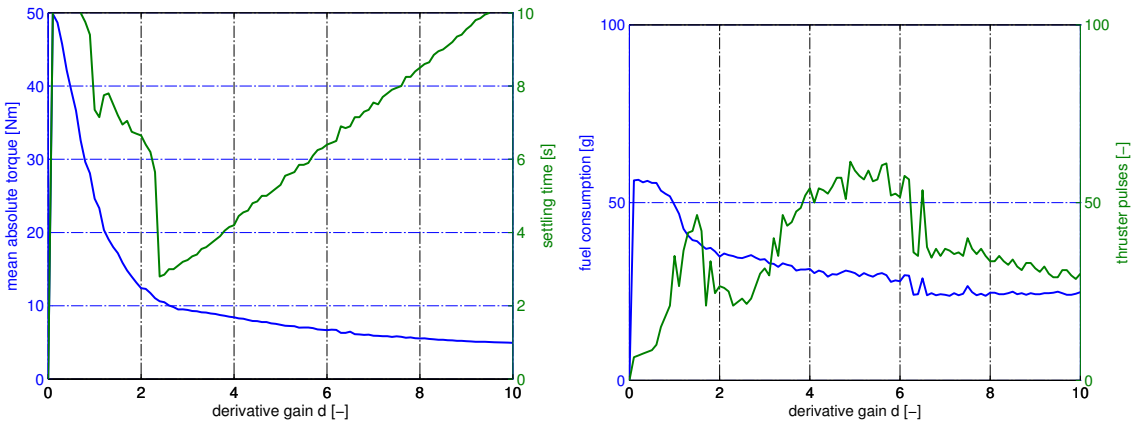
5-2-5 Control System Configuration

The system model of the Enceladus lander has been incorporated in several `MATLAB` files for unit tests before, so it is relatively simple to add the PD-controller from Section 5-2-1, the thruster selection logic from Section 5-2-3 and the equations for the PWPF modulator. The full lander simulator in `C++` will then use the values found in the optimization runs.

The first step in the spacecrafts' control cycle is the determination of a reference or command torque with the PD-controller. As discussed in Section 5-2-1, the idealized PD-controller requires only the specification of the steady-state error θ_{ss} or the settling time t_{set}. The PD-controller, however, is part of an imperfect system without proportional control and unlimited thrust. Furthermore, a dynamic excitation can lead to a delayed system response, which might require different PD-controller configuration (see Fig. 5-23. It is thus decided to first investigate the system response to a step input for different values of the proportional gain k and a fixed derivative gain value of $d = \sqrt{k}$ (see Eq. (5-62)). When a suitable k is found, the process is repeated with a fixed k and different values of d. The optimization range in both cases is [0,10].

Initially, the lander is aligned with the Enceladus inertial reference with rotational rates equal to zero. The attitude command in terms of Euler angles was at the beginning the same step input as shown in Fig. 5-22, thus $[\phi_c, \theta_c, \psi_c] = [50^\circ, 25^\circ, -10^\circ]$. But simulations indicated that a large attitude command of 50° allows no gain optimization, because the thrusters are active all the time, and the spacecraft is unable to reach a stable final state within the simulation time of 10 s (about 15 s are required for this). The following calculations will thus focus on a 10° realignment command about the X_B-axis. The available torque levels are 4.8 Nm and 9.6 Nm, see Table 5-8. The PWPF modulator parameters are fixed at their respective nominal values of $U_{\text{on}} = 0.45$, $h = 0.3$, $K_m = 4.5$, $T_m = 0.15$ and $\Delta t = 0.01$ s (see Table 5-9). Note, that in this case U_{on} and U_{off} are factors which must be multiplied with the current torque level (U_{out}). The fuel consumption is now measured in grams instead of consumption-seconds, because the mass flow of each vernier is known to be 2.8 gs^{-1}, see Table 3-3.

Figure 5-31 shows the settling time, total fuel consumption and number of thruster pulses for the range k =[0,10]. The maximum torque shown in Fig. 5-31a is the largest torque command



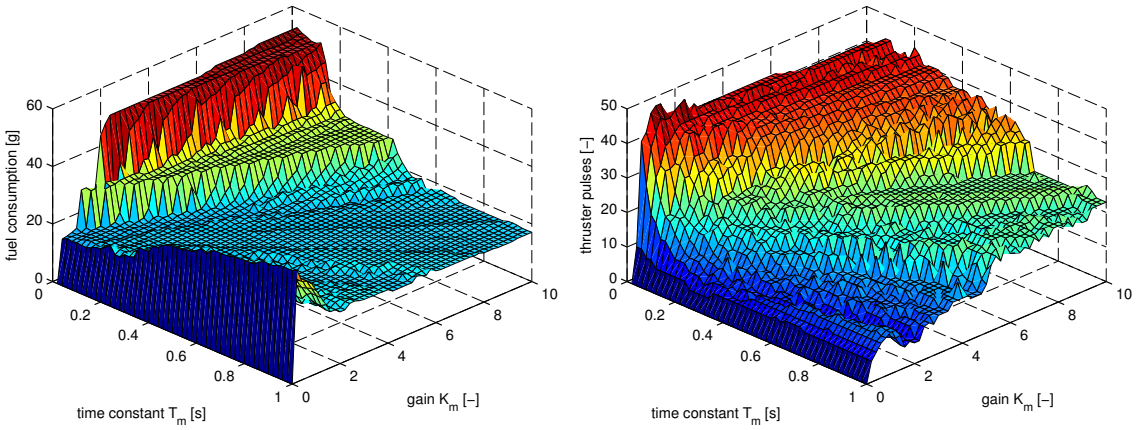
(a) mean reference torque and settling time **(b)** fuel consumption and thruster pulses

Figure 5-32: Control system performance for $k = 5$ and different values of d

for a given k, determined by the PD-controller (Eq. (5-52)). The settling time is the instance of time, where the difference between the commanded attitude and the true attitude exceeds 0.5°, starting at the simulation end and working *backwards* in time. This approach was chosen, because for larger proportional gains, the attitude error oscillates with a decreasing amplitude about zero until it finally falls below the 0.5°-threshold. k-values below 1, in general, do not lead to convergence within the simulation time of 10 s. A large k results in a large torque command, which in turn leads to more constant and high thrust output — thus, a lower number of thrust pulses and with larger fuel consumption. The fuel consumption approaches the maximal value of 56 g (two thrusters with a consumption of 2.8 gs^{-1} each are active over full simulation run of 10 s). The settling time is minimal for proportional gains between 4 and 6. In that interval, the fuel-consumption and the thruster-activity curves are in a depression. A proportional gain of 5 seems to be a reasonable trade-off between the three parameters. $k = 5$ is chosen as a new proportional gain value for the PWPF modulator optimization.

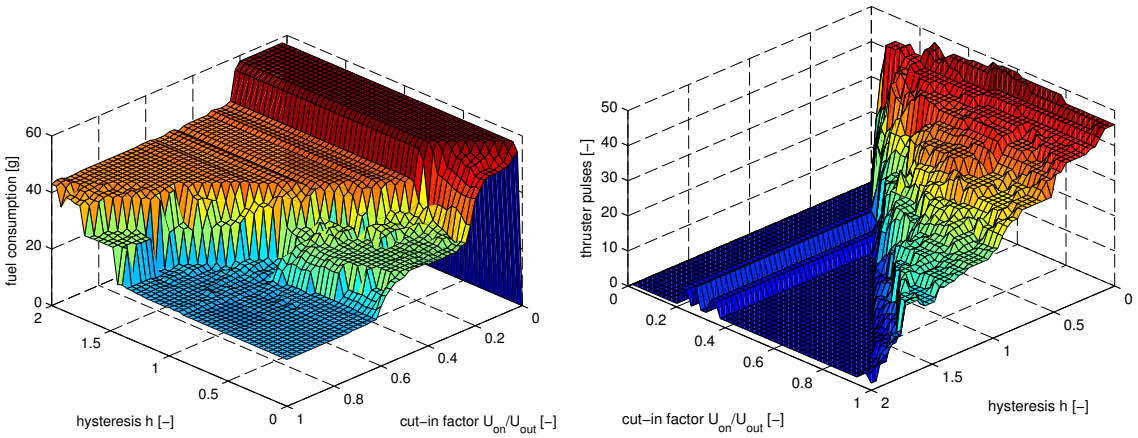
Similar to Fig. 5-31, Fig. 5-32 shows the settling time, total fuel consumption and number of thrust pulses for $k = 5$ and the range $d = [0, 10]$. The maximum torque value in this case is independent of the derivative gain value, so the mean absolute torque as a function of d is given instead. Increasing d leads to a decreasing overshoot. Values of d larger than 3 completely eliminate the overshoot and slower approach to the reference command — this has a negative effect on the settling time, but it decreases the mean absolute torque levels and thus the fuel consumption. The number of thrust pulses does not show a general trend for an increasing d. Small changes in d leads to different reference torques and in consequence to a different PWPF modulator behavior. The minimization of the settling time is the primary objective, but a derivative gain slightly larger than 3 is at the same time reasonable trade-off between fuel consumption and thruster firings. The derivative gain is chosen to be 3.5, which is larger than the value used in Fig. 5-31, $d = \sqrt{5} = 2.24$. All following simulations use a PD-controller with $k = 5$ and $d = 3.5$.

Figure 5-33 and 5-34 show the results of filter- and trigger-optimization process. The basis of the calculations is the same as in Section 5-2-4, but the input is now the output of the torque selection logic, which in turn depends on the PD-controller output. The simulation parameters are the same as shown in Table 5-9. The filter output, however, is limited to



(a) fuel consumption **(b)** thruster pulses

Figure 5-33: Filter optimization results for a the control system with $k = 5$ and $d = 3.5$



(a) fuel consumption **(b)** thruster pulses

Figure 5-34: Trigger optimization results for a the control system with $k = 5$ and $d = 3.5$

$1.5U_{\text{on}}$ (see a similar approach in McCelland (1994)). A limit is necessary to prevent the filter from accumulating large error values, in particular for larger k with the large reference torque levels. This would lead to a thruster action opposite to the currently required thrust direction, because the filter tries to reduce the eliminate errors from the past.

The filter optimization results for the system are similar to those used in the unit tests shown in Fig. 5-27 and 5-28: K_m and T_m should be small for lower number of thrust pulses. The fuel consumption increases again with decreasing T_m, but the trend for K_m is now reversed: a larger K_m leads to lower fuel consumption, in particular for the interval $[0, 2]$. A large K_m is also advantageous for the a smaller trigger dead-band range, because the PWPF modulator does not respond to input values with an absolute value below $\frac{U_{\text{on}}}{K_m}$ (see Section 5-2-4). Considering this, the combination $T_m = 0.45$ s and $K_m = 8$ seems to be a good choice for the Enceladus Lander Simulator.

The trigger optimization results shown in Fig. 5-34 show only minor differences when compared to the unit test results from Section 5-2-4: combinations of h and $\frac{U_{\text{on}}}{U_{\text{out}}}$ close to the

critical limit $h = 2\frac{U_{\text{on}}}{U_{\text{out}}}$ and beyond lead – again – to a massive increase in fuel consumption, but the consumption for $h < 2U_{\text{on}}$ is now relatively stable. The thruster pulses, on the other hand, follow the same trend as shown in Fig. 5-30: a lower $\frac{U_{\text{on}}}{U_{\text{out}}}$ and a higher h result in a lower number of thruster pulses. For the Enceladus Lander Simulator, the combination $\frac{U_{\text{on}}}{U_{\text{out}}} = 0.6$ and $h = 0.5$ will be used.

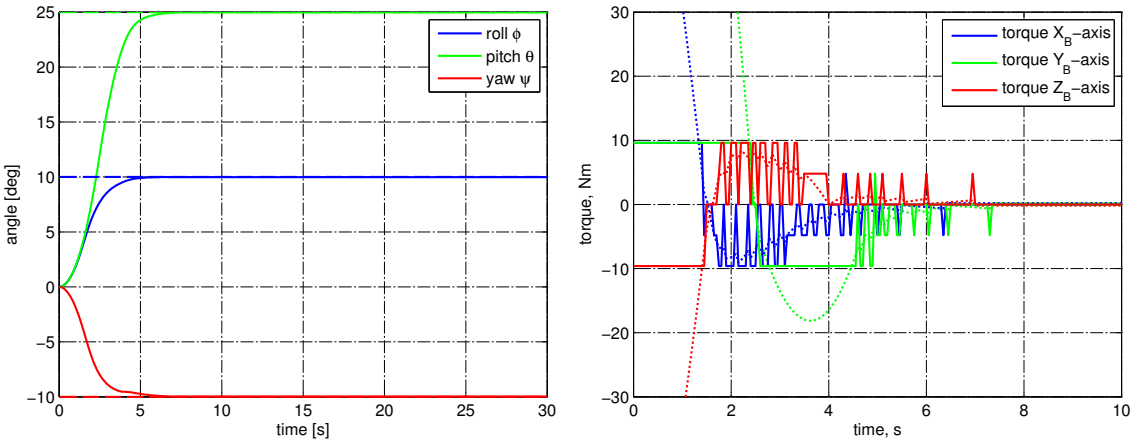
The different behavior of the full system compared to the PWPF modulator only is caused by the additional system elements, in particular the thrust thruster selection logic, the lander system model, and the large reference torques with the respect to the low available torque levels. The thruster selection logic reduces in many cases the fuel consumption in the region $h > 2\frac{U_{\text{on}}}{U_{\text{out}}}$ below the maximum value of 56 g (two thrusters active during the full 10 s simulation time), because it deactivates one thruster despite the fact that the modulator is unable to return zero output. Furthermore, the body model adds inertia to the list of system parameters. The spacecraft inertia causes a continuous rotation, when a single thrust pulse is commanded. A low cut-in value leads to a series of opposing thruster firings that aim at correcting the drift caused by the previous firings. This behavior is not present in the PWPF model itself. Finally, a large attitude command, such as the 10° used in previous calculations, causes the controller to activate a pair of thrusters and leave it on over a long period of time, reducing the total number of thruster pulses.

Name	Symbol	Value	Comment
proportional gain	k	1.5	reduced from 5
derivative gain	d	3.5	$> \sqrt{k}$
cut-in value	U_{on}	$0.6{\cdot}U_{\text{out}}$	—
cut-out value	U_{off}	$0.1{\cdot}U_{\text{out}}$	—
hysteresis	h	$0.5{\cdot}U_{\text{out}}$	$= U_{\text{on}} - U_{\text{off}}$
filter gain	K_m	8.0	—
filter limit	$\text{M}_{1_{max}}$	$1.5{\cdot}U_{\text{on}}$	must be $> U_{\text{on}}$
time constant	T_m	0.45 s	—
sampling time	Δt	0.05 s	= min. vernier on-time

Table 5-10: Control system parameters for the Enceladus Lander Simulator

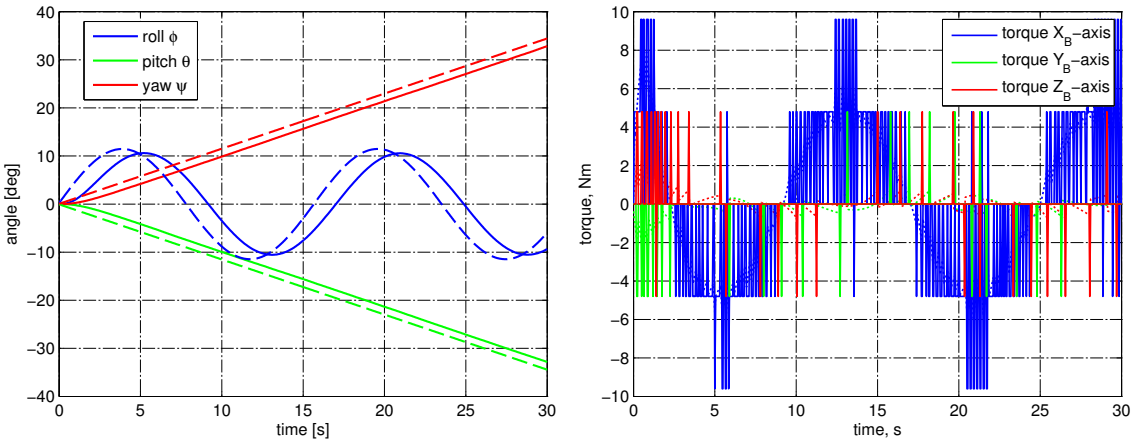
The control-system parameters used in the Enceladus Lander Simulator are collected in Table 5-10. The proportional gain of 5 used in this section had to be reduced to 1.5 in the simulator setup, because the reference moment calculated from the guidance-system output has large and quickly changing values – in particular, in combination with a large initial state estimation error. In some cases, this leads to uncontrollable rotational velocities and thus to a mission failure, see Section 8-3-2. The following simulations demonstrating the effects of the selected control-system parameters are thus not fully representative for the actual lander behavior. It will react a bit slower to the current attitude commands for the aforementioned reasons.

Figure 5-35 and 5-36 show the response of the command system to a step input and a varying input, using the original configuration parameters from Table 5-10. To better compare the results, the input commands are the same as used for the PD-controller unit, see Fig. 5-22



(a) attitude **(b)** thruster history (detail)

Figure 5-35: Simulation for the commanded attitude from Fig. 5-22 and the PWPF modulator parameters from Table 5-10. Dashed lines indicate reference values, solid lines true values.



(a) attitude **(b)** thruster history

Figure 5-36: Simulation for the commanded attitude from Fig. 5-23 and the PWPF modulator parameters from Table 5-10. Dashed lines indicate reference values, solid lines the true values.

and 5-23 — with the only exception of the reference roll angle, which was reduced from 50° to 10° during the PWPF optimization process.

The curves in Fig. 5-35a representing the spacecraft's true attitude over time (solid lines) overshoot their respective reference command (dotted lines). This is a consequence of the proportional gain being larger than the value for optimal damping; see discussion in Section 5-2-1. The system was optimized primarily for a low settling time, so $k = 5$ leads to a lower t_{set} at the expense of a higher fuel consumption. The fuel consumption, however, is still within acceptable limits, see the discussion on Fig. 5-31. The large step inputs cause a large PD-controller output (dotted line in Fig. 5-35b). This in turn leads to long thrust pulses at the maximum available torque level.

The lander's true attitude lags behind the dynamic reference command, which might have been expected from the previous PD-controller unit tests. The error is, however, lower than for the case where $t_{\text{set}} = 1.6587$ s, see Fig. 5-23a. The thruster activity is high for the periodic reference command about the X_B-axis. The torque level is always minimal for pitch and yaw rotations, because the reference commands are low enough for the PWPF to trigger a thrust pulse, before the error exceeds the threshold for the higher torque level. In fact, this behavior should be expected for most of the trajectory, as the guidance system always returns a reference acceleration command based on the current and the target attitude.

5-2-6 Simulator Configuration: Control System Parameters

The Enceladus Lander Simulator incorporates a linear quaternion controller, a PWPF controller and a thruster manager as discussed in the foregoing sections. The thruster manager has a very simple basic setup and has no configurable parameters, but is by far the largest program file within the control system due to the large number of thrusters and the routines to check for thrust limits and to distinguish between positive and negative moments or forces.

All control-system related simulator parameters are listed in Table 5-11 together with the references to the according variable names or to the equations in this chapter.

Variable Name	Comment
`PWPFactive`	on/off, if off: perfect $\mathbf{M}_{cmd}$
`proportionalGainQuaternionController`	[-], k in Eq. (5-52)
`derivativeGainQuaternionController`	[-], d in Eq. (5-52)
`PWPFfilterGain`	[-], K_m in Eq. (5-73)
`PWPFtimeConstant`	[s], T_m in Eq. (5-73)
`PWPFsamplingTime`	[s], δt in Eq. (5-73)
`PWPFcutInFactor`	[-], multiplied with U_{out}
`PWPFcutOutFactor`	[-], multiplied with U_{out}
`PFPWfilterLimitFactor`	[-], multiplied with U_{out}

Table 5-11: Control system related program parameters

Chapter 6

Navigation and Hazard Avoidance

This chapter presents the second part of the lander's flight software, namely the navigation and hazard avoidance subsystems.

Section 6-1 introduces the concept of sensor fusion and state estimation with an extended Kalman filter, and of observability of a dynamic system. This information is then applied to a basic sliding body example to analyze the effects of different instrument choices on the state estimation results. The next step is setup of the Enceladus lander navigation system, which includes a filter tuning process and discussion of the available sensors.

Section 6-2 deals with the lander's hazard avoidance system, and shows, how a reachability and fuel consumption map can be derived from the scan of a true surface hazard map, and how these maps used in the retargeting process.

Each section closes with a summary of the simulator configuration parameters relating to the foregoing discussion.

6-1 Navigation

Navigation is the process of determining the state of an object at a given instance of time. The position and velocity of an interplanetary spacecraft can be inferred from a transmitted signal, so navigation is to some extent part of the communication system (Brown, 2002). Traditionally, the orbit determination is a ground-based process: position data are collected from several ground stations over a period of time, which gives a very good position estimate at a certain instance of time in the past. From there, the equations of motion are used for orbit propagation (Wertz et al., 2009).

Precise attitude determination and near-target positioning, on the other hand, require specially designed instruments, which will be presented in the following sections. Real-time navigation is often necessary for thruster firings or a correct targeting (Wertz et al., 2009). This is usually done in the on-board *inertial navigation system* (INS), which is designed to

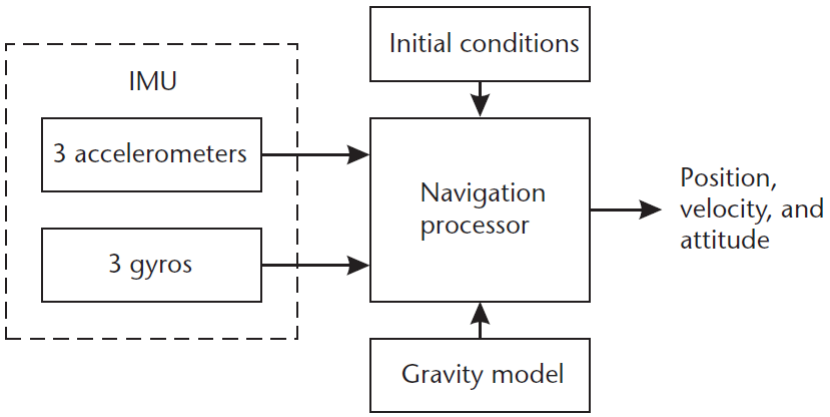


Figure 6-1: Basic inertial navigation system elements (Groves, 2008)

.

track the changes of the spacecraft's inertial state over time. The inertial state for any three-dimensional object has six degrees of freedom (three translations and three rotations). The INS uses real-time translational and rotational acceleration data from the instrumentation to determine the changes in position, velocity and attitude with respect to the initial conditions. Inertial navigation is a dead-reckoning process, which means, that the current position estimation is determined by adding the measured velocity to the previous position, taking into account the plane changes that follow from the propagated attitude. This system obviously suffers from cumulative errors for longer observation times. Inertial navigation is commonly combined with external, lower-frequency measurements to correct for these errors. Silenus incorporates star sensors for attitude corrections during all mission phases. The position corrections follow from relative position data between the orbiter and the lander during the entry phase, and from absolute distance measurements during the approach phase.

Figure 6-1 shows the basic elements of a general INS. The navigation processor includes one or more filters for the handling of the sensor measurement data. This is necessary, because every instrument has a different accuracy, measurement rate and reliability - sometimes they return unacceptable results or no results at all.

The initial conditions for the velocity and position (see Fig. 6-1) for interplanetary spacecraft are determined in ground-based processes, as mentioned before. There are many techniques available for orbit determination, including two-way ranging, Doppler positioning, direction finding and marker beacons (Groves, 2008). Doppler measurements of the downlink carrier frequency are commonly used to determine the spacecraft's velocity (Brown, 2002). The position follows from the time difference between the transmission and receiving of the signal at different ground stations. The motion of all planetary bodies, including Enceladus, is studied extensively, and the rotation of any ground station about the Earth axis and about the Sun is in principle known at all time. This allows for the elimination of the relative motion between the antennas on Earth and on the spacecraft. The algorithms for processing the observation data, including models to minimize the errors due to the Earth atmosphere, are well-established (Wertz and Larson, 1999). The main space navigation instrument of a mission to the Saturnian system will be a space-tracking network, such as NASA's Deep Space Network (DSN) or ESA's European Space Tracking network (ESTRACK). The former system was successfully used for the Cassini-Huygens mission. This navigation principle is,

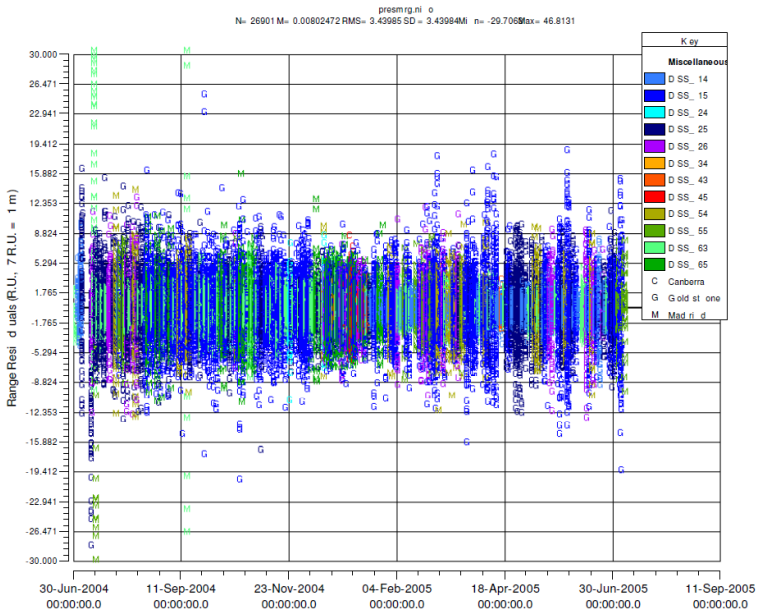


Figure 6-2: Normalized two-way Doppler range measurements for Cassini, using the DSN ground stations in Madrid, Goldstone and Canberra (Antreasian et al., 2006)

.

however, no option for the other mission element - the planetary lander - due to its limited signal strength and long signal travel time. Furthermore, the position error of a few kilometers (see Cassini orbit determination performance, for example, Antreasian et al. (2006)) is in this case much too high.

The exact position of the Mars Rovers is inferred on-ground by comparing the images from the panoramic cameras with satellite images from the Martian surface. The inertial navigation in that case is based on odometry (i.e., tracking the wheel activity), tilt sensors, gyros and accelerometers. A comparable technique will be used for the Enceladus mission. An orbiter can provide a very precise relative position estimate for the lander. Even without knowing the exact position of the spacecraft with respect to Enceladus, the orbiter can always define a position with respect to the lander. This is a very important tool to set, for example, a target point for repositioning.

The position, velocity and attitude measurements from Silenus' instruments must be sufficiently accurate to allow for a safe descent, touchdown and repositioning. The guidance system selects a landing spot based on a goodness map, which also accounts for measurement errors. Consequently, less accurate state data will lead to a lower number of safe landing areas. The landing gear can only handle velocities below 0.6 ms^{-1} in both vertical and horizontal direction (see Section 3-3), and the pointing accuracy directly reduces the maximal allowable surface elevation. The required accuracy for distance measurements depends on the accuracy of the velocity measurements. Assuming that the velocity of the lander can be determined with an accuracy of 0.3 ms^{-1}, the maximum allowable positioning error is 40 cm. In the worst case, the instrument would then measure a vertical velocity of 0 ms^{-1} and a height of 0 m, when the lander is in fact moving downwards with 0.3 ms^{-1} from an altitude of 40 cm. The impact velocity then does not exceed the landing gear limits. The design of the landing

gear assumes a maximal pointing error of 2° (see Section 3-3). Silenus' sensor configuration must be able to meet these requirements.

The software implementation of the navigational filter is complex, as it must be able to handle both internal and external measurements for the estimation of the six translational and the six rotational states, taking into account the concepts of filter calibration and observability. It was decided to first generate a simple model with one degree of freedom, and then extend this to the full lander model with six degrees of freedom.

6-1-1 Sensor Fusion and State Estimation

There are several filter techniques for the estimation of a system based on imperfect measurements. The GNC system of a hopper on Enceladus requires real-time position and attitude information, thus the calculations must be performed very quickly with an acceptable accuracy level. As indicated during the selection process of the navigational sensors, the overall accuracy of the position, velocity and attitude determinations should be better than 0.4 m, 0.3 ms^{-1} and 2°, respectively. The extended Kalman filter (EKF) seems to be the best choice, as it is a common, reliable and relatively fast system. The EKF is a first-order correction of the linear Kalman filter, which is the optimal filter for the estimation of a linear system. This correction consists of a linearization of the system about the last estimation, and thus neglects higher-order terms. The EKF is a widely-used and effective filter for most spacecraft attitude estimation processes (Crassidis et al., 2007); for example, this filter was successfully used in the HAYABUSA navigation system, with position inputs from optical navigation cameras, LIDAR and laser range finder (Kubota et al., 2006). The shortcomings of the EKF arise from the linearization process, which will lead to a divergence over a longer period of time or to inaccurate results in case the system is highly nonlinear. Recent developments, such as the unscented Kalman filter or the backwards-smoothing EKF, aim at the minimization of these errors. Sigma-point filter and particle filters are new alternatives to the EKF and incorporate higher-order approximations of the nonlinear system (Crassidis et al., 2007). These more sophisticated filter techniques are complex, and it is assumed that the performance of the simpler EKF is sufficient for the Enceladus lander. This is because the duration of the descent and repositioning process is short and the dynamic behavior of lander is known and not highly non-linear.

Each instrument of the lander works at a different frequency. Inertial navigation data from the IMU is available at a high frequency between 100 and 1000 Hz, and are as such an input for the on-board state propagation to the instance of time, where an external measurement from the star sensors, LIDAR, or other sensor, is available. The navigation filter uses these external measurements to correct the state estimation and update the error estimations. It is obvious, that the entire process is discrete due to the discontinuous measurements. The following section will, therefore, focus on the *discrete* version of the Extended Kalman Filter. Note, that in theory it is possible to translate parts of the navigational filter into a continuous-time system, for example, by applying a curve fitting process to the IMU measurements and then propagate the relevant elements analytically, or by integrating the time derivatives of the these elements. The latter was applied to a simple test case with a falling body in atmosphere, and it was concluded, that a hybrid EKF offers no real advantages, because the overall system remains discrete, and the discrete filter essentially becomes continuous for a small sampling time.

6-1-2 The Extended Kalman Filter

The basic Kalman filter consists of a set of *linear* equations for the estimation of the state of a system based on imperfect measurements. Invented 1960 by Rudolf E. Kalman, the Kalman filter underwent several modifications and nowadays is the standard real-time estimation technique for various navigational systems. This navigation filter design is only valid for linear measurement and system models, which is an reasonable assumption for many ground-based systems. The navigation systems of spacecrafts, however, are *nonlinear* and they depend on a large number of factors. The Extended Kalman Filter is a first-order correction of the basic Kalman filter and linearizes the system and measurement matrices each computation cycle for better estimation quality. This section discusses the elements and the implementation of a discrete Extended Kalman Filter.

Until stated otherwise, the following discussion is based on Welch and Bishop (2006). The change in true state $\mathbf{x} \in \mathbb{R}^n$ of a nonlinear system at time instance k can be represented by the nonlinear differential equation

$$\dot{\boldsymbol{x}}_k = f\left(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}\right) \tag{6-1}$$

where f is the state transition function, $\mathbf{u} \in \mathbb{R}^l$ is an optional control input and $\mathbf{w} \in \mathbb{R}^n$ is the system noise vector. The control input comprises any influences, that do not depend on the current system state, for example the thrust forces or external disturbing accelerations. The measurement vector $\mathbf{z} \in \mathbb{R}^m$ is modeled as a nonlinear function of the current $\mathbf{x}$ and the measurement noise $\mathbf{v} \in \mathbb{R}^m$:

$$\mathbf{z}_k = h\left(\mathbf{x}_k, \mathbf{v}_k\right) \tag{6-2}$$

For the EKF it is assumed that the system noise and the measurement noise are independent of each other, white and have a normal distribution with covariances equal to the square of the of the elements in $\mathbf{w}$ and in $\mathbf{v}$, respectively. The discrete measurement noise covariance matrix is then given by (Groves, 2008)

$$\mathbf{R}_k = E\left(\mathbf{v}_k \mathbf{v}_k^T\right), \tag{6-3}$$

where E is a function that returns the *expected values* of the inserted matrix. Due to the previous assumption, that $\mathbf{v}$ is white noise - which is a random fluctuation of a measurement value with a constant power spectral density, that can change over time, depending on the modeled system and its environment - E sets all elements, that are not on the matrix main diagonal, to zero. Similarly, the discrete process noise covariance becomes

$$\mathbf{Q}_k = E\left[\mathbf{w}_{k-1} \mathbf{w}_{k-1}^T\right]. \tag{6-4}$$

While $\mathbf{R}_k$ essentially remains each step apart from the random fluctuations of its diagonal values, the determination of $\mathbf{Q}_k$ is more complex, because a noise on one system variable has a direct influences on the noise on the other variables. This usually results in a non-diagonal system noise covariance matrix. The correct determination of $\mathbf{Q}_k$ will be discussed later, as this requires additional definitions.
The difference between the true state $\mathbf{x}$ and the time-propagated state estimate $\hat{\mathbf{x}}^-$ (thus the result of the intermediate step *after* the propagation and *before* the update of $\mathbf{x}$), and

the difference between the true state $\mathbf{x}$ and the updated state estimate $\hat{\mathbf{x}}^+$ define the time-propagated and the updated error estimates, respectively:

$$\mathbf{e}_k^- \equiv \mathbf{x}_k - \hat{\mathbf{x}}_k^- \tag{6-5}$$
$$\mathbf{e}_k^+ \equiv \mathbf{x}_k - \hat{\mathbf{x}}_k^+ \tag{6-6}$$

The according time-propagated and updated error covariance are expected values of the matrix spanned by the error estimate vectors times their transposes:

$$\mathbf{P}_k^- = E\left[\mathbf{e}_k^- \mathbf{e}_k^{-T}\right] \tag{6-7}$$
$$\mathbf{P}_k^+ = E\left[\mathbf{e}_k^+ \mathbf{e}_k^{+T}\right] \tag{6-8}$$

To find the correct expressions for $\mathbf{P}_k^-$ and $\mathbf{P}_k^+$, and in fact, also the discrete process noise covariance $\mathbf{Q}_k$ discussed before, it is necessary to derive the so-called *fundamental matrix* $\mathbf{\Phi}_k$, which propagates the state vector of the previous time step $k-1$ to the current step k, , using a linearization of the state transition function in Eq. (6-1). The linearization L of a general function $y = f(x)$ about the point $x = a$ is (Stewart, 2003, p. 262)

$$L\left[f(x)\right] = f(a) + f'(a)(x - a) \tag{6-9}$$

which at the same time is equal to the fist two terms ($n = 1$) of the Taylor series representation of f (Stewart, 2003, p. 761)

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)(a)}}{n!}(x - a)^n \tag{6-10}$$

The general expression of Eqs. (6-9) and (6-10) can now be applied to the state-space equations (6-1) and (6-2). Assuming that $\mathbf{w}$ and $\mathbf{v}$ have a zero mean and thus can be set to zero as a first approximation, the estimated linearized state vector $\hat{\mathbf{x}}_k$ and the estimated linearized measurement vector $\hat{\mathbf{z}}_k$ then become

$$L\left[\hat{\mathbf{x}}_k(t)\right] = \hat{\mathbf{x}}_k(t_k) + \mathbf{F}_{k-1}\hat{\mathbf{x}}_k(t_k)\,(t - t_k) \tag{6-11}$$
$$L\left[\hat{\mathbf{z}}_k(t)\right] = \hat{\mathbf{z}}_k(t_k) + \mathbf{H}_k\hat{\mathbf{x}}_k(t_k)\,(t - t_k)\,, \tag{6-12}$$

where $\mathbf{F}_{k-1}$ and $\mathbf{H}_k$ are the n×n dynamics matrix and the m×n measurement matrix, respectively, and can be calculated with

$$\mathbf{F}_{k-1} = \left.\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_{k-1}^+,\mathbf{w}=\mathbf{0}} \tag{6-13}$$
$$\mathbf{H}_k = \left.\frac{\partial h(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_k^-,\mathbf{w}=\mathbf{0}} \tag{6-14}$$

Equations (6-13) to (6-14) are so-called Jacobian matrices. Let $\mathbf{f} = (f_1, f_2, ..., f_m)$ be a general function which translates a vector with the coordinates $\mathbf{x} = (x_1, x_2, ..., x_n)$ from $\mathbb{R}^n$ to $\mathbb{R}^m$. For a vector $\mathbf{a}$, the Jacobian matrix $\mathbf{J}(\mathbf{a})$ then is defined as

$$\mathbf{J}(\mathbf{a}) = \frac{\partial \mathbf{f}(\mathbf{a})}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial f_1(\mathbf{a})}{\partial x_1} & \cdots & \frac{\partial f_1(\mathbf{a})}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m(\mathbf{a})}{\partial x_1} & \cdots & \frac{\partial f_m(\mathbf{a})}{\partial x_n} \end{bmatrix}. \tag{6-15}$$

Based on (6-11), the linearized fundamental matrix for Eq. (6-11) is

$$\mathbf{\Phi}_k(t) = \mathbf{I} + \mathbf{F}_{k-1}t \tag{6-16}$$

For many problems it is sufficient to only use the first two terms of the Taylor series to approximate $\mathbf{\Phi}$ (see, for example, Zarchan and Musoff (2005)). However, for larger sampling times or an observability analysis (which is discussed later in this chapter), it might be necessary to add more terms or use more terms. The following discussion is based on Zarchan and Musoff (2005), unless stated otherwise. The exact expression for the fundamental matrix is

$$\mathbf{\Phi}_k(t) = \mathrm{e}^{\mathbf{F}t} = \mathbf{I} + \mathbf{F}t + \frac{\mathbf{F}^2t^2}{2!} + \frac{\mathbf{F}^3t^3}{3!} + \dots \tag{6-17}$$

where $\mathbf{I}$ is a n×n identity matrix. The time t in the context of a discrete EKF is equal to the sample time T_s of the measurement $\mathbf{z}_k$. The discrete process noise matrix $\mathbf{Q}_k$ then follows from the expression

$$\mathbf{Q}_k = \int_0^{T_s} \mathbf{\Phi}_k(\tau)\mathbf{Q}\mathbf{\Phi}_k^T(\tau)\,\mathrm{d}\tau \tag{6-18}$$

where $\mathbf{Q}$ is the continuous process noise matrix from Eq. (6-4). The integration is usually straightforward - especially, if some elements on main diagonal of $\mathbf{Q}$ are zero. This is a common situation, because according to Zarchan and Musoff (2005) process noise should always be added to the highest derivative of correlating state elements *only* (for example, only to the velocity, if the position is a state variable, too). The process noise is propagated in the navigation filter and will in the end also influence the other depending states.

At this point, all relevant information for the application of the so-called *Riccati* equations has been derived. The Riccati equations use the time-propagated error covariance $\mathbf{P}_k^-$ to determine a suitable gain $\mathbf{K}_k$ for interpreting the incoming measurements, and then update the error covariance to $\mathbf{P}_k^+$. The updated state estimate $\hat{\mathbf{x}}_k^+$ follows from the time-propagated state estimate $\hat{\mathbf{x}}_k^-$ and a weighted measurement difference between the actual and the true measurement:

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k\left(\mathbf{z}_k - h\left(\hat{\mathbf{x}}_k^-\right)\right) \tag{6-19}$$

The Kalman gain matrix $\mathbf{K}_k$ is thus multiplied with the measurement innovation in order to update the propagated state estimate. The goal now is chose the form of $\mathbf{K}_k$ which minimizes the updated error covariance $\mathbf{P}_k^+$ (Eq. (6-8)). It can be shown that optimal Kalman gain matrix is given as (Groves, 2008)

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}_k^T\left(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k\right)^{-1} \tag{6-20}$$

Using the above relations, Eqs. (6-7) and (6-8) can be rewritten as (Groves, 2008; Welch and Bishop, 2006)

$$\mathbf{P}_k^- = \mathbf{\Phi}_k\mathbf{P}_{k-1}^+\mathbf{\Phi}_k^T + \mathbf{Q}_{k-1} \tag{6-21}$$

$$\mathbf{P}_k^+ = \left(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\right)\mathbf{P}_k^-. \tag{6-22}$$

Equations (6-20), (6-21) and (6-22) form the aforementioned Riccati equations and are the core of each Kalman filter. The intermediate mathematical conversions and derivations are described in more detail in Terejanu (2004) and Ribeiro (2004).

Given the initial estimates $\hat{\mathbf{x}}_{k-1}^{+} = \hat{\mathbf{x}}_{0}^{+}$ and $\mathbf{P}_{k-1}^{+} = \mathbf{P}_{0}^{+}$, the EKF process starts with the propagation of the state estimation $\hat{\mathbf{x}}_k$ using

$$\hat{\mathbf{x}}_k = g\left(\hat{\mathbf{x}}_{k-1}^{+}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}\right) \tag{6-23}$$

which is usually a numerical integration of Eq. (6-1). The state prediction $\hat{\mathbf{x}}_k$ is then used to determine the measurement matrix $\mathbf{H}_k$ in Eq. (6-14). The fundamental matrix $\mathbf{\Phi}_k$ is calculated with the previous corrected state estimation and follows from Eq. (6-16) or (6-17), which also allows the determination of the discrete process noise matrix $\mathbf{Q}_k$ in Eq. (6-18). A common initial configuration for the continuous process noise $\mathbf{Q}$ required at this step is the squared estimated difference between the initial true state and the initial estimated state, divided by the observation time T_f (Zarchan and Musoff, 2005):

$$\mathbf{Q} = \operatorname{diag}\left(\frac{\left[x_{1_0} - \hat{x}_{1_0}^{+}\right]^2}{T_f}, \frac{\left[x_{2_0} - \hat{x}_{2_0}^{+}\right]^2}{T_f}, \ldots, \frac{\left[x_{n_0} - \hat{x}_{n_0}^{+}\right]^2}{T_f}\right) \tag{6-24}$$

The prediction phase is concluded by the propagation of the error covariance from $\mathbf{P}_{k-1}^{-}$ to $\mathbf{P}_{k}^{-}$. The update phase starts with the determination of the Kalman gain $\mathbf{K}_k$ using Eq. (6-20), which finally allows for the update of both the predicted state estimation and the predicted error covariance to $\hat{\mathbf{x}}_k^{+}$ and $\mathbf{P}_k^{+}$, respectively.

The stability and effectiveness of the Kalman filter depends on the choice of the system and measurement error matrices $\mathbf{Q}_k$ and $\mathbf{R}_k$, and on the initial value of the error covariance matrix $\mathbf{P}_0^{+}$. The choice of $\mathbf{P}_0^{+}$ and $\mathbf{R}_k$ directly influences the Kalman gain, see Eq. (6-20), and thus the weighting of the recent measurements. According to Groves (2008), the choice of the ratio $\mathbf{P}/\mathbf{R}$ is the decisive factor in a Kalman filter: an underestimation of $\mathbf{P}/\mathbf{R}$ results in an slow conversion process, an overestimation to instability (see Fig. 6-3). In practice, the values for $\mathbf{P}_0^{+}$ and $\mathbf{Q}_k$ follow from the instrument specifications and tests. A common strategy is to determine the smallest possible value of $\mathbf{R}$ for a stable Kalman filter, given fixed values for $\mathbf{P}_0^{+}$ and $\mathbf{Q}_k$ (Groves, 2008).

The navigation filter converges correctly (see, for example, Zarchan and Musoff (2005)), if the residual $\hat{\mathbf{x}}_k - \mathbf{x}_k$ is at least 67% of the observation time within the theoretical limits given by square-root of the according values on the main diagonal of $\mathbf{P}_k^{+}$.

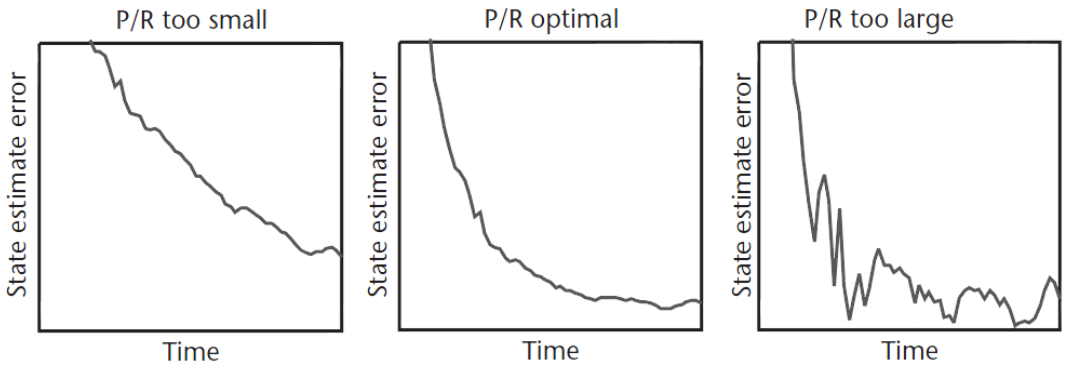


Figure 6-3: Tuning of the Kalman filter based on the ratio $\mathbf{P}/\mathbf{R}$ (Groves, 2008)

6-1-3 Observability

A necessary condition for a successful navigation filter is the *observability* of the dynamic system. A system is said to be observable, if one can derive the internal states from a given input and a measured output. In case one or more states are completely unrelated to the output, it is impossible to deduce those variables, and the system is not observable. Note that observability is only a necessary, not a sufficient condition for reliable navigation system: both the initial state and the control vector influence the estimation process. Consequently, it is very likely that the convergence of the navigation filter requires excitations of several state elements, for example by thruster firings. The observability of a dynamical system can be determined by investigating the state-space model, such as described in Eqs. (6-1) and (6-2). For convenience, these equations are repeated here. The state space model for the state vector $\mathbf{x}_k$ and the output vector $\mathbf{z}_k$ is

$$\mathbf{x}_k = g\left(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_k\right) \tag{6-25a}$$

$$\mathbf{z}_k = h\left(\mathbf{x}_k, \mathbf{v}_k\right) \tag{6-25b}$$

As shown in Eq. (6-17), the system transition matrix for this dynamic system at the discrete time index k is

$$\mathbf{\Phi}_k(t) = \mathbf{I} + \mathbf{F}t + \frac{\mathbf{F}^2 t^2}{2!} + \frac{\mathbf{F}^3 t^3}{3!} + \ldots. \tag{6-26}$$

Note that it is usually not sufficient to only consider the first two terms for an observability analysis, because this might eliminate relationships between state variables, which is a crucial element for derivation of the internal states. The so-called *observability Gramian* at index k then is defined as (Guermah et al., 2008)

$$\mathbf{W}\left(0,k\right) = \sum_{j=0}^{k-1} \mathbf{\Phi}_j^T \mathbf{H}_j^T \mathbf{H}_j \mathbf{\Phi}. \tag{6-27}$$

The system in Eq. (6-25a) is observable if, and only if, there is a time index k for which the *rank* of the observability Gramian is equal to the number of state variables n (see theorem 5, Guermah et al. (2008)). The rank in this case is defined as the number of linear independent column vectors in $\mathbf{W}\left(0,k\right)$ as defined in Eq. (6-27). Since the Gramian here is always a

n×n quadratic matrix, the maximal - or *full* - rank is n. Only a matrix with a full rank is invertible (see Körner (2012)), so the necessary condition for an observable system is a non-zero determinant of $\mathbf{W}(0,k)$. The diagonal values of the Gramian give a qualitative indication about the observability of all state variables.

6-1-4 Navigation Filter Implementation: Sliding Body Model

The initial model consists of a body that moves frictionless along one axis. The body carries an accelerometer and is observed by an external sensor, which can provide position or velocity measurements. The internal sensor suffers from a bias, a scale error and white noise, whereas the external sensor only suffers from the latter. The body is accelerated by an external force along its direction of movement. The situation is depicted in Fig. 6-4.

The goal of this model is to estimate the position x and velocity $\dot{x}$ of the sliding body, given an imprecise, high-frequency on-board accelerometer and a limited number of external measurements. In this simple case, the accelerometer model is

$$a_m = (1 + s_a) f_s + b_a + v_a \tag{6-28}$$

where s_a is the scale factor error, b_a the measurement bias, v_a a random noise error and f_s the specific force (or non-gravitational acceleration) acting on the sliding body. v_a for accelerometers is commonly expressed in terms of root power spectral density (PSD), which means, that a basic averaged error is divided by the root of the sensor's measurement frequency, leading to the customary unit of $\mu g/\sqrt{Hz}$ (Groves, 2008), see Section 6-1-7 about the IMU for information about accelerometer error models. The values of the two non-random error sources are important for a good position and velocity estimation, so they are added to the state vector, which then becomes

$$\mathbf{x} = \begin{pmatrix} x \\ \dot{x} \\ b_a \\ s_a \end{pmatrix}. \tag{6-29}$$

The external sensor will initially take position measurements only. As indicated before, the external measurements errors are limited to random noises, so the measurement equation only depends on the true position. Assuming that the errors are time-independent, the state-space representation of the sliding body model is

$$\dot{\boldsymbol{x}} = f(\mathbf{x}) + \mathbf{w} = \begin{pmatrix} \dot{x} \\ f_a \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{pmatrix} \tag{6-30}$$

$$\boldsymbol{z} = h(\mathbf{x}) + v_p = x + v_p \tag{6-31}$$

where w_1, w_2, w_3 and w_4 are the random process noises on the four state-vector elements. In theory, both w_1 and w_2 should be zero, as as they correspond to the position and velocity, which are integrals of the remaining two state variables - as discussed in Section 6-1-2, process noise should only be added to the highest derivatives of correlating state elements. However,

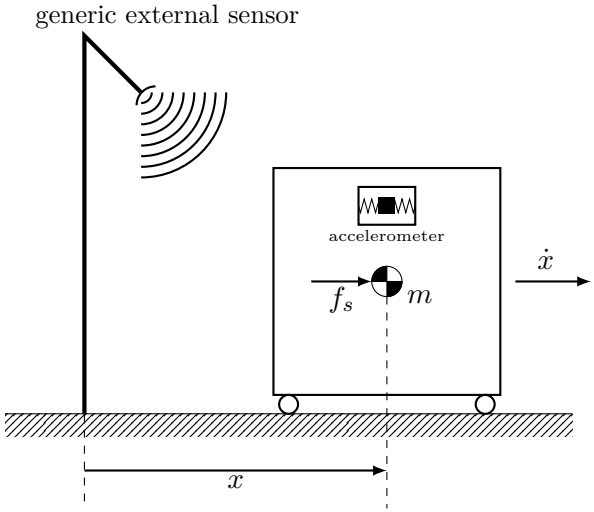


Figure 6-4: Sliding body example to test the early stage of the navigational filter

the simulations have shown, that the effect of w_3 and w_4 is on the first state vector element is very small, which leads to a an insensitivity to external measurements after some time. It is therefore decided to assume that all elements of $\mathbf{w}$ are nonzero. Equations (6-30) and (6-31) lead to the following system-dynamics and measurement matrix:

$$\mathbf{F}_{k-1} = \left.\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\boldsymbol{x}}_{k-1}^{+}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \mathrm{f}_{23} & \mathrm{f}_{24} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{-a_m+\hat{b}_a}{(1+\hat{s}_a)^2} & \frac{-1}{1+\hat{s}_a} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{6-32}$$

$$\mathbf{H}_k = \left.\frac{\partial h(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\boldsymbol{x}}_{k}^{-}} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \tag{6-33}$$

The continuous process noise matrix $\mathbf{Q}$ is a diagonal matrix with the diagonal entries equal to process noise vector $\mathbf{w}$. The *discrete* process noise matrix $\mathbf{Q}_k$ follows from integrating the quadratic form of the system transition matrix $\boldsymbol{\Phi}_k$ and $\mathbf{Q}$ over the sampling time T_s. Note that the sampling time is the measurement frequency of the external sensor only; the internal accelerometer has a higher measurement frequency independent from $\mathbf{z}$. For simplicity, $\boldsymbol{\Phi}_k$ in the derivation of $\mathbf{Q}_k$ is determined using only the first two terms of the Taylor-series expansion (see Eq. (6-16)), which gives

$$\mathbf{Q}_k = \int_0^{T_s} \boldsymbol{\Phi}_k(\tau)\mathbf{Q}\boldsymbol{\Phi}_k^T(\tau)\,\mathrm{d}\tau \tag{6-34}$$

$$= \int_0^{T_s} \begin{bmatrix} 1 & \tau & 0 & 0 \\ 0 & 1 & \mathrm{f}_{23}\tau & \mathrm{f}_{24}\tau \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} w_1 & 0 & 0 & 0 \\ 0 & w_2 & 0 & 0 \\ 0 & 0 & w_3 & 0 \\ 0 & 0 & 0 & w_4 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ \tau & 1 & 0 & 0 \\ 0 & f_{23}\tau & 1 & 0 \\ 0 & f_{24}\tau & 0 & 1 \end{bmatrix} \mathrm{d}\tau \tag{6-35}$$

$$= \begin{bmatrix} w_3\frac{T_s^3}{3} + w_1 T_s & w_2\frac{T_s^2}{2} & 0 & 0 \\ w_2\frac{T_s^2}{2} & w_3 f_{23}^2\frac{T_s^3}{3} + w_4 f_{24}^2\frac{T_s^3}{3} + w_2 T_s & w_3 f_{23}\frac{T_s^2}{2} & w_4 f_{24}\frac{T_s^2}{2} \\ 0 & w_3 f_{23}\frac{T_s^2}{2} & w_3 T_s & 0 \\ 0 & w_4 f_{24}\frac{T_s^2}{2} & 0 & w_4 T_s \end{bmatrix}. \tag{6-36}$$

This completes the derivation of all relevant elements of the navigation filter for the one-instrument-case. Table 6-1 gives an overview of the exact values used during the simulations. The integration time step Δt refers to the time step used inside the EKF to propagate $\hat{\mathbf{x}}_{k-1}^+$ to $\hat{\mathbf{x}}_k^-$ between two accelerometer measurements. The accelerometer measurement is updated every accelerometer sample time T_a. The state correction $\hat{\mathbf{x}}_k^+$ is performed as soon as an external measurement is available, i.e. every position sample time T_s. The true state $\mathbf{x}_k$ is propagated with a basic Runge-Kutta 4th order integrator before the actual navigation filter is initiated. The external measurements $\mathbf{z}_k$ and the internal accelerometer measurements a_m follow from the original true state acting on the sliding body, the latter is modified by the true values b_a and s_a.

Figures 6-5 and 6-6 show the error in the estimation of s_a and b_a as a function of the simulation time for a constant and for a varying true acceleration, respectively. The error is always the difference between the true and the current estimated value. The theoretical limits, indicated by dotted red lines, follow from the square roots of the values on the main diagonal of $\mathbf{P}_k^+$. Note that the estimations for both the position and the velocity in all cases are within the theoretical limits and thus not shown here. As can be seen, the filter cannot track the bias and scale error if the acceleration is constant. This might be expected, because of the found combination of s_a and b_a in fact yields the correct value for the specific force as given in Eq. (6-28). However, as soon as the acceleration is changing, such as shown in Fig. 6-6, the derived value for f_s is inaccurate. This leads to deviation between $\mathbf{x}_k$ and $\hat{\mathbf{x}}_k^+$, which is corrected by the external position measurements. Consequently, one error source must be removed from the state vector in case the acceleration is constant. The theoretical estimation limits of b_a and s_a are in the order of 0.12 ms^{-2} and 0 08, which is not sufficient for many applications. The main reason for low observability of these states just based on position measurements. The observability Gramian for the one-instrument-case is generated during the simulation run according to Eq. (6-27), and the values on its main diagonal are $[2000, 0.2, 1.06\times10^{-5}, 2.5\times10^{-6}]$. The rank of the observability Gramian is 3, which indicates, that the system is, in principle, not observable.
The observability of b_a and s_a can be improved by adding external measurements of another state element, in this case, velocity measurements. Equation (6-31) and (6-33) then become

$$\mathbf{z} = h(\mathbf{x}) + \mathbf{v} = \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} v_p \\ v_v \end{bmatrix} \tag{6-37}$$

$$\mathbf{H}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{6-38}$$

As can be seen in Fig. 6-7, the errors are slowly diverging from the theoretical limits in case the true acceleration is constant; the reason for this is same as for the one-instrument-case. A periodic acceleration, on the other hand, leads to faster and more precise estimates for b_a and s_a compared with the previous case. The theoretical estimation limits are now 0.0008 ms^{-2}

Description	Variable	Value
true initial state	$\mathbf{x}_0$	$\begin{bmatrix} -10\ \text{m} \\ 2\ \text{ms}^{-1} \\ 0.5\ \text{ms}^{-2} \\ 1\ \text{ms}^{-2} \end{bmatrix}$
estimated initial state	$\hat{\mathbf{x}}_0^+$	$\begin{bmatrix} 0\ \text{m} \\ 0\ \text{ms}^{-1} \\ 0\ \text{ms}^{-2} \\ 0\ \text{ms}^{-2} \end{bmatrix}$
process noise - pos. measurement only	$\mathbf{w}_1$	$\begin{bmatrix} 0.5\ \text{m} \\ 0.02\ \text{m} \\ 0.0013\ \text{ms}^{-2} \\ 0.005\ \text{ms}^{-2} \end{bmatrix}$
process noise - pos. and vel. measurement	$\mathbf{w}_2$	$\begin{bmatrix} 0.5\ \text{m} \\ 0\ \text{m} \\ 0\ \text{ms}^{-2} \\ 0\ \text{ms}^{-2} \end{bmatrix}$
measurement noise	$\mathbf{v}$	$\begin{bmatrix} 0.01\ \text{m}^2 \\ 0.01\ \text{m}^2\text{s}^{-2} \end{bmatrix}$
estimated initial covariance	$\mathbf{P}_0^+$	$\begin{bmatrix} 100\ \text{m}^2 & 0 & 0 & 0 \\ 0 & 4\ \text{m}^2\text{s}^{-2} & 0 & 0 \\ 0 & 0 & 0.25\ \text{m}^2\text{s}^{-4} & 0 \\ 0 & 0 & 0 & 1\ \text{m}^2\text{s}^{-4} \end{bmatrix}$
accelerometer noise	v_a	$0.001\ \text{ms}^{-2}$
simulation length	T_f	200 s
position sensor sample time	T_s	0.1 s
accelerometer sample time	T_a	0.01 s
integration time step	Δt	0.001 s
true acc. bias	b_a	$1\ \text{ms}^{-2}$
true acc. scale error	s_a	0.5

Table 6-1: Simulation configuration for the sliding body example

and 0.0006, respectively. The values on the main diagonal of the observability Gramian are $[2000, 2000, 0.385, 0.09]$, which is a clear improvement with respect to the one-instrument-case. The rank of the observability Gramian is now 4, so the system should be observable.

The sliding body problem shows, that is important to determine the observability of the lander model for a specific instrument configuration – and even if the observability Gramian has a full rank, the navigation filter requires filter tuning and excitations on all states for good state estimation. The design of the lander navigation system in the next section will account for the observability requirements, and the filter tuning even has its own section (Section 6-1-6).

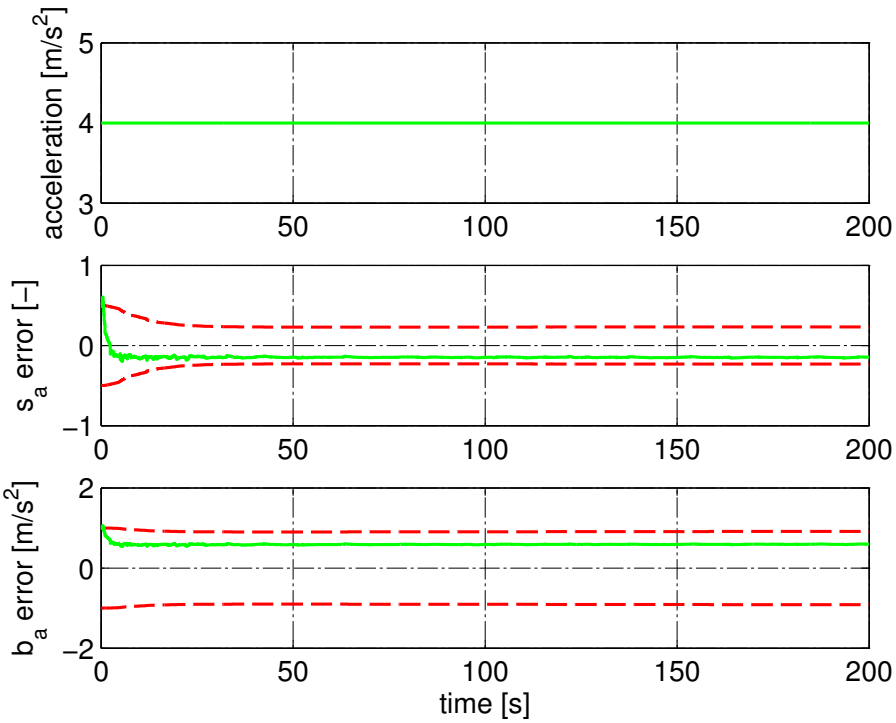


Figure 6-5: Error estimations for a constant true acceleration, using position measurements only. Dashed line indicates the theoretical limit.

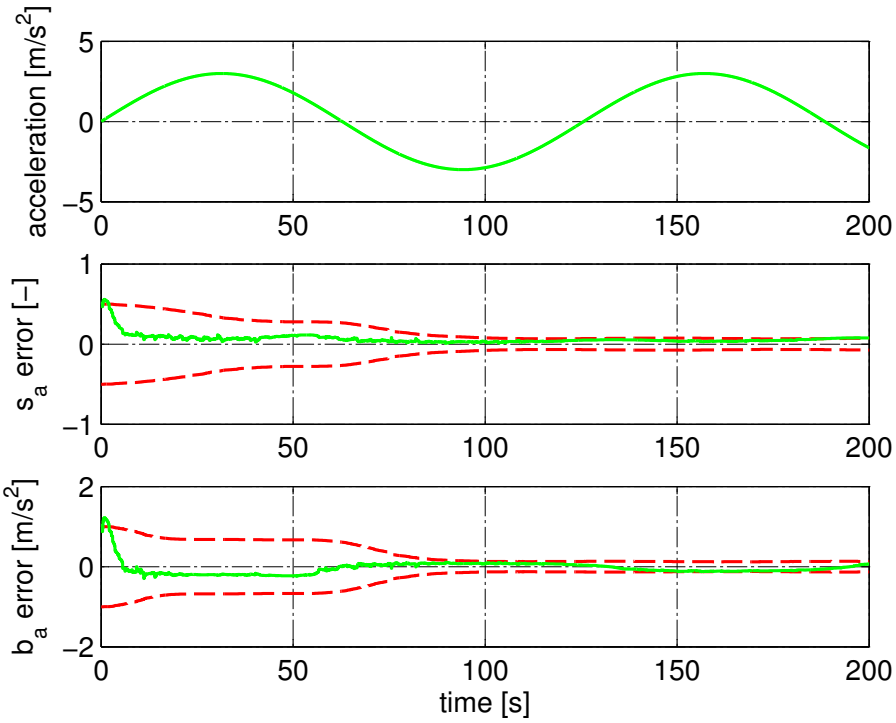


Figure 6-6: Error estimations for a varying true acceleration, using position measurements only. Dashed line indicates the theoretical limit.

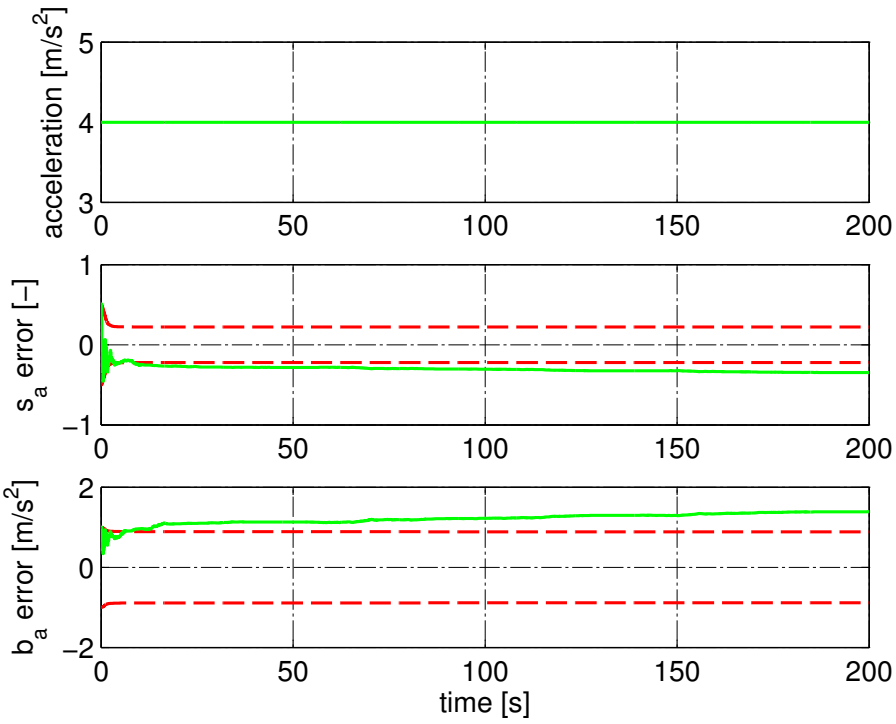


Figure 6-7: Error estimations for a constant true acceleration, using position and velocity measurements. Dashed line indicates the theoretical limit.

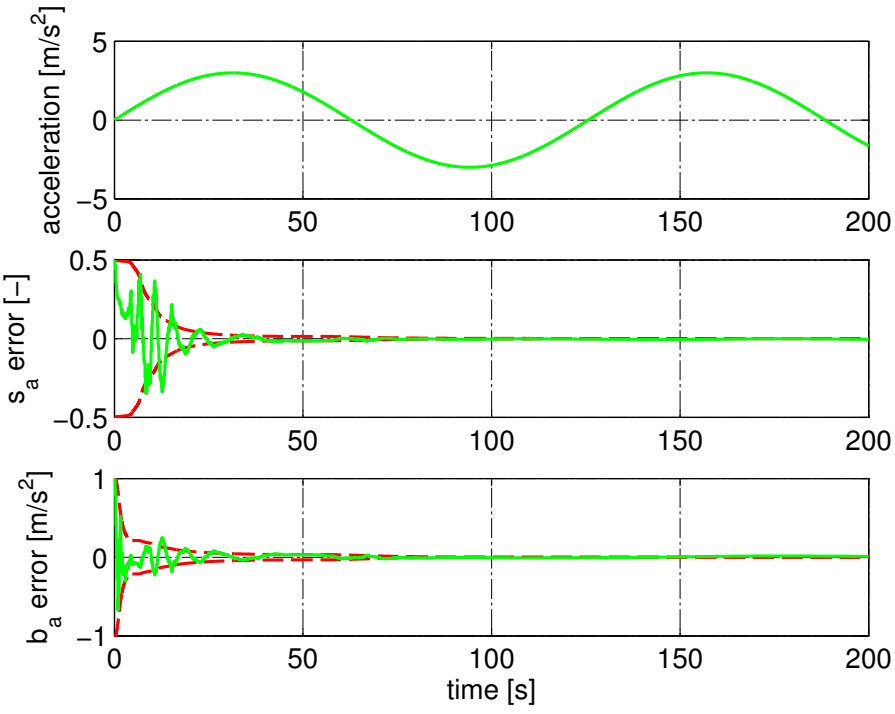


Figure 6-8: Error estimations for a varying true acceleration, using position and velocity measurements. Dashed line indicates the theoretical limit.

6-1-5 Navigation Filter Implementation: Enceladus Lander

The full lander model has six degrees of freedom and is constantly influenced by its environment and by its own thrust commands.

As discussed before, the lander navigation system is based on an IMU, aided by external attitude and position measurements. The absolute attitude information is generated with the on-board star sensors. The orbit determination far away from Earth, on the other hand, is complicated, because it is not possible to use global navigation systems such as GPS. It is possible to determine the position of an interplanetary spacecraft with Earth-based antenna networks, but the position error will be in the order of kilometers and arrives with a time delay due to the large distance between the spacecraft and ground station. The semi-major axis of Saturn is about $1{,}433.53\times10^6$ km (see Table 4-2), so the one-way light time is about

$$\Delta t_{Earth-Saturn} = \frac{(1,433.53 - 149.60)\cdot 10^6 \text{ km}}{299,792.5 \text{ km}\cdot \text{s}^{-1}} = 4283\text{s} = 71.4 \text{ min.} \tag{6-39}$$

The orbiter will be tracked during its entire mission life. The on-board position estimation is then regularly updated with delayed absolute position measurements. It is impossible to accurately land the lander at an exact, predefined location on Enceladus due to the limitations of the orbit determination process. However, it *is* possible to let the spacecraft define a landing target on the moon's surface and then navigate with respect to this point. The idea behind the lander's autonomous navigation system is that ground control defines a target area – in this case, on the Enceladus south pole, near a Tiger Stripe – and the lander then analyzes the region and selects a suitable landing spot. It is decisive that the spacecraft can determine its position with respect to this target point, using the LIDAR and, if necessary, additional instruments such as altimeters, in combination with the IMU. The IMU is an inertial sensor and measures the acceleration and rotational rate of the spacecraft. However, it needs an external reference to begin(Groves, 2008), and must be calibrated well before the descent process starts to minimize the accumulated errors. The external measurements required for this follow from the star sensors and from position data provided by the orbiter. As the orbiter's sensor configuration is unknown at this point, it will be assumed that either the orbiter or the lander combines and processes range and range-rate information to (relative) position data. The navigational filter thus incorporates both loosely and tightly coupled system models, because both raw IMU data and processed position information are filter inputs (see Mooij and Chu (2001)). Note that the filter performance might be further increased by processing the raw range and range-rate measurements inside the filter, at the costs of higher complexity and longer integration times.

This section will start with the derivation of the IMU instrument models. This models are then applied to the spacecraft's equations of motion to derive the system dynamics matrix and all depending elements of the navigation filter. The calibration process and the performance during the actual mission is then tested for different external instruments.

As mentioned in Section 6-1-7, the main error sources of the IMU's accelerometers and gyroscopes are biases, scale factors, misalignments and random errors. Following these explanations, the accelerometer measurement output can be modeled by (see Mooij and Chu (2001) and Groves (2008))

$$a_{x_m} = a_x + s_x a_x + m_{xy} a_y + m_{xz} a_z + b_x + v_{a_x}$$

$$\begin{aligned} a_{y_m} &= a_y + s_y a_y + m_{yx} a_x + m_{yz} a_z + b_y + v_{a_y} \\ a_{z_m} &= a_z + s_z a_z + m_{zx} a_x + m_{zy} a_y + b_z + v_{a_z} \end{aligned} \tag{6-40}$$

where a_x, a_y and a_z are true specific forces, s_x, s_y and s_z are scale factor errors, m_{xy}, m_{xz}, m_{yx}, m_{yz}, m_{zx} and m_{zy} are the misalignments, b_x, b_y and b_z are biases and v_{a_x}, v_{a_y} and v_{a_z} are the random white noises for each axis. Similarly, the gyroscope model is

$$\begin{aligned} p_m &= p + s_p p + m_{pq} q + m_{pr} r + b_p + v_p \\ q_m &= q + s_q q + m_{qp} p + m_{qr} r + b_q + v_q \\ r_m &= r + s_r r + m_{rp} p + m_{rq} q + b_r + v_r \end{aligned} \tag{6-41}$$

where p, q and r are the true rotational rates, s_p, s_q and s_r are the scaling errors, m_{pq}, m_{pr}, m_{qp}, m_{pr}, m_{rp} and m_{rq} are the misalignments, b_p, b_q and b_r are biases and v_p, v_q and v_r are the random noises. The scaling errors and the misalignment errors are usually combined in a single matrix, whose elements would be zero in case the instrument is perfectly aligned and calibrated. Then, Eqs. (6-40) and (6-41) reduce to

$$\mathbf{a}_m = (\mathbf{I} + \mathbf{M}_a)\,\mathbf{a} + \mathbf{b}_a + \mathbf{v}_a \tag{6-42}$$

$$\boldsymbol{\omega}_m = (\mathbf{I} + \mathbf{M}_g)\,\boldsymbol{\omega} + \mathbf{b}_g + \mathbf{v}_g \tag{6-43}$$

with

$$\mathbf{M}_a = \begin{bmatrix} s_x & m_{xy} & m_{xz} \\ m_{yx} & s_y & m_{yz} \\ m_{zx} & m_{zy} & s_z \end{bmatrix} \tag{6-44}$$

$$\mathbf{M}_g = \begin{bmatrix} s_p & m_{pq} & m_{pr} \\ m_{qp} & s_q & m_{qr} \\ m_{rp} & m_{rq} & s_r \end{bmatrix} \tag{6-45}$$

and

$$\mathbf{b}_a = \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix},\ \mathbf{b}_g = \begin{pmatrix} b_p \\ b_q \\ b_r \end{pmatrix},\ \mathbf{v}_a = \begin{pmatrix} v_{a_x} \\ v_{a_y} \\ v_{a_z} \end{pmatrix},\ \mathbf{v}_g = \begin{pmatrix} v_p \\ v_q \\ v_r \end{pmatrix}. \tag{6-46}$$

The estimated true values for the specific force and rotational rate are the inputs for the equations of translational and rotational motion and follow from solving Eqs. (6-42) and (6-43) for $\mathbf{a}$ and $\boldsymbol{\omega}$. The random measurement errors are disregarded in this step, as they have a zero mean and are completely independent of the state history. Then, given the specific force measurements a_{x_m}, a_{y_m} and a_{z_m}, the non-gravitational acceleration components for the equations of translational motion with respect to the inertial reference frame are (Mooij and Chu, 2002)

$$\begin{aligned} \hat{\mathbf{a}}_I &= \mathbf{F}_{I \leftarrow B} \begin{bmatrix} 1+s_x & m_{xy} & m_{xz} \\ m_{yx} & 1+s_y & m_{yz} \\ m_{zx} & m_{zy} & 1+s_z \end{bmatrix}^{-1} \left\{ \begin{bmatrix} a_{x_m} \\ a_{y_m} \\ a_{z_m} \end{bmatrix} - \begin{bmatrix} \hat{b}_x \\ \hat{b}_y \\ \hat{b}_z \end{bmatrix} \right\} \\ &= \mathbf{F}_{I \leftarrow B} \left(\mathbf{I} + \hat{\mathbf{M}}_a \right)^{-1} \left(\mathbf{a}_m - \hat{\mathbf{b}}_a \right) \end{aligned} \tag{6-47}$$

where the hat characters indicate estimations, and given the angular velocity measurements p_m, q_m and r_m, the rotational rates for the equations of rotational motion are (Mooij and Chu, 2002)

$$\begin{aligned}\hat{\boldsymbol{\omega}} &= \begin{bmatrix} 1+s_p & m_{pq} & m_{pr} \\ m_{qp} & 1+s_q & m_{qr} \\ m_{rp} & m_{rq} & 1+s_r \end{bmatrix}^{-1} \left\{ \begin{bmatrix} p_m \\ q_m \\ r_m \end{bmatrix} - \begin{bmatrix} \hat{b}_p \\ \hat{b}_q \\ \hat{b}_r \end{bmatrix} \right\} \\ &= \left(\mathbf{I} + \hat{\mathbf{M}}_g\right)^{-1} \left(\boldsymbol{\omega}_m - \hat{\mathbf{b}}_g\right)\end{aligned} \tag{6-48}$$

The approximate error values for space grade IMU are discussed in Section 6-1-7.

Note that the acceleration is measured with respect to the body reference frame and must be translated into the inertial reference frame using the transformation $\mathbf{F}_{I \leftarrow B}$.

The state vector in the navigational filter will contain the scale errors, misalignments and biases for both the accelerometer and the gyroscope. In this way, the errors can be estimated and tracked during the entire mission. The lander's rotational rate $\boldsymbol{\omega}$ follows directly from the gyroscope measurement and is as such not part of the navigation state vector, but directly added to the output state vector for the guidance system. Furthermore, the attitude is expressed in terms of the quaternion's vector part only, because the navigation filter is unable to distribute the attitude estimation updates in a way that would comply with the rule shown Eq. (4-2) (the norm of an attitude quaternion must be equal to one). To some extent, the error is reduced by the corrective term $K_q \left[1 - \left(q_0^2 + q_1^2 + q_2^2 + q_3^2\right)\right] \bar{\mathbf{q}}$ during each prediction step (see Eq. (4-66)) — the first, uncorrected EKF implementation yielded almost the same results as the revised one —, but the filter will return invalid attitude quaternions most of the time, nevertheless. The scalar part of the attitude quaternion thus follows from the quaternion norm, see Eq. 4-2.

The state vector inside the navigation filter is then

$$\mathbf{x} = \begin{pmatrix} \mathbf{r} \\ \mathbf{v} \\ \mathbf{q}_{I/B} \\ \mathbf{b}_a \\ \mathbf{s}_a \\ \mathbf{m}_a \\ \mathbf{b}_g \\ \mathbf{s}_g \\ \mathbf{m}_g \end{pmatrix} \tag{6-49}$$

where the misalignment vectors $\mathbf{m}_a$ and $\mathbf{m}_g$ consist of the elements of the error matrices in (6-44) and (6-45), sorted row by row:

$$\mathbf{m}_a = \begin{pmatrix} m_{xy} \\ m_{xz} \\ m_{yx} \\ m_{yz} \\ m_{zx} \\ m_{zy} \end{pmatrix}, \qquad \mathbf{m}_g = \begin{pmatrix} m_{pq} \\ m_{pr} \\ m_{qp} \\ m_{qr} \\ m_{rp} \\ m_{rq} \end{pmatrix} \tag{6-50}$$

The system dynamics matrix involves the derivatives of the IMU instrument models with respect to the error state elements. The derivative of the inverse matrices in Eqs. (6-47) and (6-48) lead to very long expressions, but they can be simplified considerably by rewriting the partial derivative of the inverse matrix. Let $\mathbf{A}$ be a n×n non-singular matrix, whose elements are functions of the elements x_i of the vector $\mathbf{x}$. In this case,

$$\mathbf{A}^{-1}\mathbf{A} = \mathbf{I} \tag{6-51}$$

where $\mathbf{I}$ is an identity matrix of the order n. Differentiating Eq. (6-51) with respect to element i of $\mathbf{x}$ leads to

$$\frac{\partial}{\partial x_i}\left(\mathbf{A}^{-1}\mathbf{A}\right) = \frac{\partial \mathbf{I}}{\partial x_i} = \frac{\partial \mathbf{A}^{-1}}{\partial x_i}\mathbf{A} + \mathbf{A}^{-1}\frac{\partial \mathbf{A}}{\partial x_i} = \mathbf{0}. \tag{6-52}$$

The partial derivatives of $\mathbf{A}^{-1}$ can then be written as:

$$\frac{\partial \mathbf{A}^{-1}}{\partial x_i} = -\mathbf{A}^{-1}\frac{\partial \mathbf{A}}{\partial x_i}\mathbf{A}^{-1} \tag{6-53}$$

Applying this new expression to the instrument models in Eqs. (6-47) and (6-48) simplifies the derivation of the system-dynamics matrix, because the partial derivatives of $(\mathbf{I}+\mathbf{M}_a)^{-1}$ and $(\mathbf{I}+\mathbf{M}_g)^{-1}$ with respect to the error states $\mathbf{b}_a$, $\mathbf{s}_a$, $\mathbf{m}_a$, $\mathbf{b}_g$, $\mathbf{s}_g$ and $\mathbf{m}_g$ are matrices consisting of zeros and ones. In the full instrument models, the inverse matrices are multiplied with the difference of the actual measurement vector and the bias vector, which are not a function of the elements of the matrix. Representing this m×1 vector by $\mathbf{y}$, the expression (6-53) can be written as

$$\frac{\partial\left(\mathbf{A}^{-1}\mathbf{y}\right)}{\partial x_i} = \frac{\partial \mathbf{A}^{-1}}{\partial x_i}\mathbf{y} + \mathbf{A}^{-1}\frac{\partial \mathbf{y}}{\partial x_i} = -\mathbf{A}^{-1}\frac{\partial \mathbf{A}}{\partial x_i}\mathbf{A}^{-1}\mathbf{y}, \tag{6-54}$$

because $\frac{\partial \mathbf{y}}{\partial x_i} = 0$. The derivative with respect to the complete vector $\mathbf{x}$ can thus not be simplified further and is

$$\frac{\partial \mathbf{A}^{-1}}{\partial \mathbf{x}} = \begin{bmatrix} -\mathbf{A}^{-1}\frac{\partial \mathbf{A}}{\partial x_1}\mathbf{A}^{-1}\mathbf{y} & -\mathbf{A}^{-1}\frac{\partial \mathbf{A}}{\partial x_2}\mathbf{A}^{-1}\mathbf{y} & \cdots & -\mathbf{A}^{-1}\frac{\partial \mathbf{A}}{\partial x_n}\mathbf{A}^{-1}\mathbf{y} \end{bmatrix} \tag{6-55}$$

The next step is construction of the state derivative $\dot{\boldsymbol{x}}$. The only forces acting on the lander are thrust forces of the engines and the gravitational attraction of Enceladus. As discussed in Section 2-1, the third body perturbations and the effects of the gravitational moments of both Saturn and Enceladus itself are small and will be not be part of the navigation filter. The time derivative of the second state vector element $\mathbf{v}$ is then the sum of the estimated specific acceleration from Eq. (6-47) and the local gravitational acceleration, $\mathbf{g}_{Enc}$. The local gravitational acceleration with respect to the inertial reference frame is (see also 4-4-1):

$$\mathbf{g}_{Enc} = \begin{pmatrix} \ddot{x}_g \\ \ddot{y}_g \\ \ddot{z}_g \end{pmatrix} = -\frac{\mu_{Enc}}{x^2+y^2+z^2} \cdot \frac{1}{\sqrt{x^2+y^2+z^2}} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \tag{6-56}$$

The time derivative of the third state element $\mathbf{q}$ follows from inserting the gyroscope measurement from Eq. (6-48) in a modified version of the quaternion derivative $\dot{\bar{\mathbf{q}}}$ in Eq. (4-64), where the first row of $\boldsymbol{\Omega}_{\bar{q}}$ corresponding to the excluded q_0 is removed:

$$\dot{\mathbf{q}}_{I,B} = \frac{1}{2}\begin{bmatrix} q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} = \frac{1}{2}\boldsymbol{\Omega}_q \boldsymbol{\omega} \tag{6-57}$$

The state-space representation for the Enceladus lander then becomes

$$\dot{\boldsymbol{x}} = f(\mathbf{x}) + \mathbf{w} = \begin{pmatrix} \mathbf{v} \\ \mathbf{T}_{I\leftarrow B}\left(\mathbf{I}+\mathbf{M}_a\right)^{-1}\left(\mathbf{a}_m-\mathbf{b}_a\right)+\mathbf{g}_{Enc} \\ \frac{1}{2}\boldsymbol{\Omega}_q\left(\mathbf{I}+\mathbf{M}_g\right)^{-1}\left(\boldsymbol{\omega}_m-\mathbf{b}_g\right) \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} \mathbf{w}_r \\ \mathbf{w}_v \\ \mathbf{w}_q \\ \mathbf{w}_{b_a} \\ \mathbf{w}_{s_a} \\ \mathbf{w}_{m_a} \\ \mathbf{w}_{b_g} \\ \mathbf{w}_{s_g} \\ \mathbf{w}_{m_g} \end{pmatrix} \tag{6-58}$$

$$\mathbf{z} = h(\mathbf{x}) + \mathbf{v} = \begin{pmatrix} r_x \\ r_y \\ r_z \\ q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} + \begin{pmatrix} v_{r_x} \\ v_{r_y} \\ v_{r_z} \\ v_{q_0} \\ v_{q_1} \\ v_{q_2} \\ v_{q_3} \end{pmatrix} \tag{6-59}$$

The measurement vector in Eq. (6-59) assumes direct external position and attitude measurements in order to limit the filter complexity. The difference between two instrument measuring one state element – for example, range and light tracking and LIDAR for position determination – only lies in the different sample time and measurement accuracy. The measurement matrix $\mathbf{H}$ is accordingly simple and constant for all time steps k:

$$\mathbf{H}_k = \left.\frac{\partial h(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\boldsymbol{x}}_k^-} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{6-60}$$

The system dynamic matrix, on the other hand, is complex and is different for each time step:

$$\begin{pmatrix} \Delta\dot{\mathbf{r}} \\ \Delta\dot{\mathbf{v}} \\ \Delta\dot{\mathbf{q}} \\ \Delta\dot{\mathbf{b}}_a \\ \Delta\dot{\mathbf{s}}_a \\ \Delta\dot{\mathbf{m}}_a \\ \Delta\dot{\mathbf{b}}_g \\ \Delta\dot{\mathbf{s}}_g \\ \Delta\dot{\mathbf{m}}_g \end{pmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{F}_{21} & \mathbf{0} & \mathbf{F}_{23} & \mathbf{F}_{24} & \mathbf{F}_{25} & \mathbf{F}_{26} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{F}_{33} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F}_{37} & \mathbf{F}_{38} & \mathbf{F}_{39} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{pmatrix} \Delta\mathbf{r} \\ \Delta\mathbf{v} \\ \Delta\mathbf{q} \\ \Delta\mathbf{b}_a \\ \Delta\mathbf{s}_a \\ \Delta\mathbf{m}_a \\ \Delta\mathbf{b}_g \\ \Delta\mathbf{s}_g \\ \Delta\mathbf{m}_g \end{pmatrix} \tag{6-61}$$

where

$$\mathbf{F}_{21} = \begin{bmatrix} \frac{\delta\dot{v}}{\delta r_x} & \frac{\delta\dot{v}}{\delta r_y} & \frac{\delta\dot{v}}{\delta r_z} \end{bmatrix} \tag{6-62}$$

$$\mathbf{F}_{23} = \begin{bmatrix} \frac{\delta\dot{v}}{\delta q_1} & \frac{\delta\dot{v}}{\delta q_2} & \frac{\delta\dot{v}}{\delta q_3} \end{bmatrix} \tag{6-63}$$

$$\mathbf{F}_{24} = \begin{bmatrix} \frac{\delta\dot{v}}{\delta b_x} & \frac{\delta\dot{v}}{\delta b_y} & \frac{\delta\dot{v}}{\delta b_z} \end{bmatrix} \tag{6-64}$$

$$\mathbf{F}_{25} = \begin{bmatrix} \frac{\delta \dot{\boldsymbol{v}}}{\delta s_x} & \frac{\delta \dot{\boldsymbol{v}}}{\delta s_y} & \frac{\delta \dot{\boldsymbol{v}}}{\delta s_z} \end{bmatrix} \tag{6-65}$$

$$\mathbf{F}_{26} = \begin{bmatrix} \frac{\delta \dot{\boldsymbol{v}}}{\delta m_{xy}} & \frac{\delta \dot{\boldsymbol{v}}}{\delta m_{xz}} & \frac{\delta \dot{\boldsymbol{v}}}{\delta m_{yx}} & \frac{\delta \dot{\boldsymbol{v}}}{\delta m_{yz}} & \frac{\delta \dot{\boldsymbol{v}}}{\delta m_{zx}} & \frac{\delta \dot{\boldsymbol{v}}}{\delta m_{zy}} \end{bmatrix} \tag{6-66}$$

$$\mathbf{F}_{33} = \begin{bmatrix} \frac{\delta \dot{\boldsymbol{q}}}{\delta q_1} & \frac{\delta \dot{\boldsymbol{q}}}{\delta q_2} & \frac{\delta \dot{\boldsymbol{q}}}{\delta q_3} \end{bmatrix} \tag{6-67}$$

$$\mathbf{F}_{37} = \begin{bmatrix} \frac{\delta \dot{\boldsymbol{q}}}{\delta b_p} & \frac{\delta \dot{\boldsymbol{q}}}{\delta b_q} & \frac{\delta \dot{\boldsymbol{q}}}{\delta b_r} \end{bmatrix} \tag{6-68}$$

$$\mathbf{F}_{38} = \begin{bmatrix} \frac{\delta \dot{\boldsymbol{q}}}{\delta s_p} & \frac{\delta \dot{\boldsymbol{q}}}{\delta s_q} & \frac{\delta \dot{\boldsymbol{q}}}{\delta s_r} \end{bmatrix} \tag{6-69}$$

$$\mathbf{F}_{39} = \begin{bmatrix} \frac{\delta \dot{\boldsymbol{q}}}{\delta m_{pq}} & \frac{\delta \dot{\boldsymbol{q}}}{\delta m_{pr}} & \frac{\delta \dot{\boldsymbol{q}}}{\delta m_{qp}} & \frac{\delta \dot{\boldsymbol{q}}}{\delta m_{qr}} & \frac{\delta \dot{\boldsymbol{q}}}{\delta m_{rp}} & \frac{\delta \dot{\boldsymbol{q}}}{\delta m_{rq}} \end{bmatrix} \tag{6-70}$$

with

$$\frac{\delta \dot{\boldsymbol{v}}}{\delta r_x} = \frac{\mu_{Enc}}{\left(x^2+y^2+z^2\right)^{\frac{5}{2}}} \begin{bmatrix} 2x^2 - y^2 - z^2 \\ 3xy \\ 3xz \end{bmatrix} \tag{6-71}$$

$$\frac{\delta \dot{\boldsymbol{v}}}{\delta r_y} = \frac{\mu_{Enc}}{\left(x^2+y^2+z^2\right)^{\frac{5}{2}}} \begin{bmatrix} 3xy \\ -x^2 + 2y^2 - z^2 \\ 3yz \end{bmatrix} \tag{6-72}$$

$$\frac{\delta \dot{\boldsymbol{v}}}{\delta r_z} = \frac{\mu_{Enc}}{\left(x^2+y^2+z^2\right)^{\frac{5}{2}}} \begin{bmatrix} 3xz \\ 3yz \\ -x^2 - y^2 + 2z^2 \end{bmatrix} \tag{6-73}$$

$$\frac{\delta \dot{\boldsymbol{v}}}{\delta q_1} = 2 \begin{bmatrix} q_1 & q_2 & q_3 \\ q_2 & q_1 & q_0 \\ q_3 & -q_0 & -q_1 \end{bmatrix} \left(\mathbf{I} + \mathbf{M}_a\right)^{-1} \left(\mathbf{a}_m - \mathbf{b}_a\right) \tag{6-74}$$

$$\frac{\delta \dot{\boldsymbol{v}}}{\delta q_2} = 2 \begin{bmatrix} -q_2 & q_1 & -q_0 \\ q_1 & -q_2 & q_3 \\ q_0 & q_3 & -q_2 \end{bmatrix} \left(\mathbf{I} + \mathbf{M}_a\right)^{-1} \left(\mathbf{a}_m - \mathbf{b}_a\right) \tag{6-75}$$

$$\frac{\delta \dot{\boldsymbol{v}}}{\delta q_3} = 2 \begin{bmatrix} -q_3 & q_0 & q_1 \\ -q_0 & -q_3 & q_2 \\ q_1 & q_2 & -q_3 \end{bmatrix} \left(\mathbf{I} + \mathbf{M}_a\right)^{-1} \left(\mathbf{a}_m - \mathbf{b}_a\right) \tag{6-76}$$

$$\frac{\delta \dot{\boldsymbol{v}}}{\delta b_x} = \mathbf{T}_{I \leftarrow B} \left(\mathbf{I} + \mathbf{M}_a\right)^{-1} \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} \tag{6-77}$$

$$\frac{\delta \dot{\boldsymbol{v}}}{\delta b_y} = \mathbf{T}_{I \leftarrow B} \left(\mathbf{I} + \mathbf{M}_a\right)^{-1} \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} \tag{6-78}$$

$$\frac{\delta \dot{\boldsymbol{v}}}{\delta b_z} = \mathbf{T}_{I \leftarrow B} \left(\mathbf{I} + \mathbf{M}_a\right)^{-1} \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \tag{6-79}$$

$$\frac{\delta \dot{\boldsymbol{v}}}{\delta s_x} = -\mathbf{T}_{I \leftarrow B} \left(\mathbf{I} + \mathbf{M}_a\right)^{-1} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \left(\mathbf{I} + \mathbf{M}_a\right)^{-1} \left(\mathbf{a}_m - \mathbf{b}_a\right) \tag{6-80}$$

$$\frac{\delta \dot{\boldsymbol{v}}}{\delta s_y} = -\mathbf{T}_{I \leftarrow B} \left(\mathbf{I} + \mathbf{M}_a\right)^{-1} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \left(\mathbf{I} + \mathbf{M}_a\right)^{-1} \left(\mathbf{a}_m - \mathbf{b}_a\right) \tag{6-81}$$

$$\frac{\delta \dot{\boldsymbol{v}}}{\delta s_z} = -\mathbf{T}_{I \leftarrow B} \left(\mathbf{I} + \mathbf{M}_a\right)^{-1} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \left(\mathbf{I} + \mathbf{M}_a\right)^{-1} \left(\mathbf{a}_m - \mathbf{b}_a\right) \tag{6-82}$$

$$\frac{\delta \dot{\boldsymbol{v}}}{\delta m_{xy}} = -\mathbf{T}_{I \leftarrow B} \left(\mathbf{I} + \mathbf{M}_a\right)^{-1} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \left(\mathbf{I} + \mathbf{M}_a\right)^{-1} \left(\mathbf{a}_m - \mathbf{b}_a\right) \tag{6-83}$$

$$\frac{\delta \dot{\boldsymbol{v}}}{\delta m_{xz}} = -\mathbf{T}_{I \leftarrow B} \left(\mathbf{I} + \mathbf{M}_a\right)^{-1} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \left(\mathbf{I} + \mathbf{M}_a\right)^{-1} \left(\mathbf{a}_m - \mathbf{b}_a\right) \tag{6-84}$$

$$\frac{\delta \dot{\boldsymbol{v}}}{\delta m_{yx}} = -\mathbf{T}_{I \leftarrow B} \left(\mathbf{I} + \mathbf{M}_a\right)^{-1} \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \left(\mathbf{I} + \mathbf{M}_a\right)^{-1} \left(\mathbf{a}_m - \mathbf{b}_a\right) \tag{6-85}$$

$$\frac{\delta \dot{\boldsymbol{v}}}{\delta m_{yz}} = -\mathbf{T}_{I \leftarrow B} \left(\mathbf{I} + \mathbf{M}_a\right)^{-1} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \left(\mathbf{I} + \mathbf{M}_a\right)^{-1} \left(\mathbf{a}_m - \mathbf{b}_a\right) \tag{6-86}$$

$$\frac{\delta \dot{\boldsymbol{v}}}{\delta m_{zx}} = -\mathbf{T}_{I \leftarrow B} \left(\mathbf{I} + \mathbf{M}_a\right)^{-1} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \left(\mathbf{I} + \mathbf{M}_a\right)^{-1} \left(\mathbf{a}_m - \mathbf{b}_a\right) \tag{6-87}$$

$$\frac{\delta \dot{\boldsymbol{v}}}{\delta m_{zy}} = -\mathbf{T}_{I \leftarrow B} \left(\mathbf{I} + \mathbf{M}_a\right)^{-1} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \left(\mathbf{I} + \mathbf{M}_a\right)^{-1} \left(\mathbf{a}_m - \mathbf{b}_a\right) \tag{6-88}$$

$$\frac{\delta \dot{\boldsymbol{q}}}{\delta q_1} = \frac{1}{2} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \left(\mathbf{I} + \mathbf{M}_g\right)^{-1} \left(\boldsymbol{\omega}_m - \mathbf{b}_g\right) \tag{6-89}$$

$$\frac{\delta \dot{\boldsymbol{q}}}{\delta q_2} = \frac{1}{2} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \left(\mathbf{I} + \mathbf{M}_g\right)^{-1} \left(\boldsymbol{\omega}_m - \mathbf{b}_g\right) \tag{6-90}$$

$$\frac{\delta \dot{\boldsymbol{q}}}{\delta q_3} = \frac{1}{2} \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \left(\mathbf{I} + \mathbf{M}_g\right)^{-1} \left(\boldsymbol{\omega}_m - \mathbf{b}_g\right) \tag{6-91}$$

$$\frac{\delta \dot{\boldsymbol{q}}}{\delta b_p} = \frac{1}{2} \boldsymbol{\Omega}_q \left(\mathbf{I} + \mathbf{M}_g\right)^{-1} \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} \tag{6-92}$$

$$\frac{\delta \dot{\boldsymbol{q}}}{\delta b_q} = \frac{1}{2} \boldsymbol{\Omega}_q \left(\mathbf{I} + \mathbf{M}_g\right)^{-1} \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} \tag{6-93}$$

$$\frac{\delta \dot{\boldsymbol{q}}}{\delta b_r} = \frac{1}{2}\boldsymbol{\Omega}_q \left(\mathbf{I} + \mathbf{M}_g\right)^{-1} \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \tag{6-94}$$

$$\frac{\delta \dot{\boldsymbol{q}}}{\delta s_p} = -\frac{1}{2}\boldsymbol{\Omega}_q \left(\mathbf{I} + \mathbf{M}_g\right)^{-1} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \left(\mathbf{I} + \mathbf{M}_g\right)^{-1} \left(\boldsymbol{\omega}_m - \mathbf{b}_g\right) \tag{6-95}$$

$$\frac{\delta \dot{\boldsymbol{q}}}{\delta s_q} = -\frac{1}{2}\boldsymbol{\Omega}_q \left(\mathbf{I} + \mathbf{M}_g\right)^{-1} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \left(\mathbf{I} + \mathbf{M}_g\right)^{-1} \left(\boldsymbol{\omega}_m - \mathbf{b}_g\right) \tag{6-96}$$

$$\frac{\delta \dot{\boldsymbol{q}}}{\delta s_r} = -\frac{1}{2}\boldsymbol{\Omega}_q \left(\mathbf{I} + \mathbf{M}_g\right)^{-1} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \left(\mathbf{I} + \mathbf{M}_g\right)^{-1} \left(\boldsymbol{\omega}_m - \mathbf{b}_g\right) \tag{6-97}$$

$$\frac{\delta \dot{\boldsymbol{q}}}{\delta m_{pq}} = -\frac{1}{2}\boldsymbol{\Omega}_q \left(\mathbf{I} + \mathbf{M}_g\right)^{-1} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \left(\mathbf{I} + \mathbf{M}_g\right)^{-1} \left(\boldsymbol{\omega}_m - \mathbf{b}_g\right) \tag{6-98}$$

$$\frac{\delta \dot{\boldsymbol{q}}}{\delta m_{pr}} = -\frac{1}{2}\boldsymbol{\Omega}_q \left(\mathbf{I} + \mathbf{M}_g\right)^{-1} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \left(\mathbf{I} + \mathbf{M}_g\right)^{-1} \left(\boldsymbol{\omega}_m - \mathbf{b}_g\right) \tag{6-99}$$

$$\frac{\delta \dot{\boldsymbol{q}}}{\delta m_{qp}} = -\frac{1}{2}\boldsymbol{\Omega}_q \left(\mathbf{I} + \mathbf{M}_g\right)^{-1} \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \left(\mathbf{I} + \mathbf{M}_g\right)^{-1} \left(\boldsymbol{\omega}_m - \mathbf{b}_g\right) \tag{6-100}$$

$$\frac{\delta \dot{\boldsymbol{q}}}{\delta m_{qr}} = -\frac{1}{2}\boldsymbol{\Omega}_q \left(\mathbf{I} + \mathbf{M}_g\right)^{-1} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \left(\mathbf{I} + \mathbf{M}_g\right)^{-1} \left(\boldsymbol{\omega}_m - \mathbf{b}_g\right) \tag{6-101}$$

$$\frac{\delta \dot{\boldsymbol{q}}}{\delta m_{rp}} = -\frac{1}{2}\boldsymbol{\Omega}_q \left(\mathbf{I} + \mathbf{M}_g\right)^{-1} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \left(\mathbf{I} + \mathbf{M}_g\right)^{-1} \left(\boldsymbol{\omega}_m - \mathbf{b}_g\right) \tag{6-102}$$

$$\frac{\delta \dot{\boldsymbol{q}}}{\delta m_{rq}} = -\frac{1}{2}\boldsymbol{\Omega}_q \left(\mathbf{I} + \mathbf{M}_g\right)^{-1} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \left(\mathbf{I} + \mathbf{M}_g\right)^{-1} \left(\boldsymbol{\omega}_m - \mathbf{b}_g\right) \tag{6-103}$$

The determination of the discrete process noise $\mathbf{Q}_k$ for the full six degrees-of-freedom lander model requires a different approach as the previous one degree-of-freedom model: the system transition transition matrix $\boldsymbol{\Phi}$ and the continuous process noise matrix $\mathbf{Q}$ now have the dimensions 34×34, which would lead to a complicated analytical integration process. Gibbs (2011) mentions that approximations for the process noise work well in Kalman filters, as the full expression for $\mathbf{Q}_k$ in fact only describes the random error growth during one sample time. It is therefore possible to use again the Taylor-series expansion from Eq. (6-10) to propagate the $\mathbf{Q}$ from one measurement step to the next, which leads to an expression for $\mathbf{Q}_k$. In case $\boldsymbol{\Phi}$ is the first-order approximation $\mathbf{I} + \mathbf{F}T_s$, such as shown in Eq. (6-16), the discrete process noise may be written as (Gibbs, 2011)

$$\mathbf{Q}_k = \mathbf{Q}T_s + \left[\mathbf{F}\mathbf{Q} + \mathbf{Q}\mathbf{F}^T\right]\frac{T_s^2}{2} + \mathbf{F}\mathbf{Q}\mathbf{F}^T\frac{T_s^3}{3} \tag{6-104}$$

where $\mathbf{Q}$ is the diagonal matrix of the squared process errors for each state vector element.

This concludes the mathematical description of the lander's navigational filter. At this point, it is not clear whether a state element can be estimated with sufficient accuracy; this is determined in the filter-tuning process. It may be necessary to modify the instrumentation or the filter state vector.

6-1-6 Filter Tuning

The performance of the on-board EKF can be heavily influenced by tuning the variable parts of the filter. In general, these variable elements are the measurement noise matrix $\mathbf{R}$, the process noise covariance matrix $\mathbf{Q}$, and the initial estimates $\hat{\mathbf{x}}_0$ and $\mathbf{P}_0^+$. Tuning these parameters influences the filter's stability, i.e., the stability of the state estimations over time. In addition to this, the filter might require excitations of the tracked states to derive the internal states (see discussion on observability, Section 6-1-3). The excitation is realized by means of maneuvers before the lander initiates the descent process.

The accuracy of the instruments is usually determined with a series of on-ground tests before the mission start. As an initial estimation, $\mathbf{R}$ will be based on the manufacturer's specifications. The values of the diagonal elements of $\mathbf{P}_0^+$ are chosen to be the squared differences between elements of $\mathbf{x}_0$ and $\hat{\mathbf{x}}_0$:

$$\mathbf{P}_0^+ = \text{diag}\left(\left[x_{1_0} - \hat{x}_{1_0}^+\right]^2, \left[x_{2_0} - \hat{x}_{2_0}^+\right]^2, \ldots, \left[x_{n_0} - \hat{x}_{n_0}^+\right]^2\right) \tag{6-105}$$

Zarchan and Musoff (2005) uses this relation for the EKF implementation in a satellite navigation and a falling body problem simulation. With Eq. (6-105) it is possible to fix $\mathbf{P}_0^+$, if $\hat{\mathbf{x}}_0$ is chosen such that it represents the largest expected deviation from the true state $\mathbf{x}_0$. In reality, the initial state estimation is then always better as $\hat{\mathbf{x}}_0$ used in the tuning process.
The only remaining tuning parameter is $\mathbf{Q}$. A larger $\mathbf{Q}$ leads to larger influence of the measurement $\mathbf{z}_k$ on the state update, as the Kalman gain $\mathbf{K}$ will be larger (see Section 6-1-2). This, however, also means a higher influence of the measurement noise $\mathbf{v}$ on the state estimation, so $\hat{\mathbf{x}}_k$ might become unstable for large values of $\mathbf{Q}$. The process-noise elements are determined in a trial-and-error process for a fixed $\mathbf{P}_0^+$ and $\mathbf{R}$, and a given maneuver that excites all states.

The initial calibration-maneuver simulation uses only the attitude thrusters to change the translational and rotational motions. The lander's attitude control is based on 12 verniers, 4 for each rotation axis and two for each rotation direction (see Section 3-2). The attitude thruster configuration as shown in Fig. 3-2b in fact does not allow for a translational force in the $\pm X_A$-direction, because the thruster anchors all lie within the same geometrical plane. In the simulator it will be assumed that the lander has four additional verniers connected to the two thruster anchors for the pitch control; two pointing in the $+X_B$-direction, and two in the $-X_B$-direction. These thrusters are only used for translational control, so the attitude control theory discussed in Section 5-2 remains untouched. In reality, the thrust direction of all verniers would be determined in a trade-off process between system weight and reliability — this usually leads to thrust directions, that are not aligned with the spacecraft's body axes.

The maximal thrust in one direction in this scenario is thus 2×6 N along each axis. The maximal torque about each body axis is about 8.36 Nm ($\pm Y_B$,$\pm Z_B$) or 9.60 Nm ($\pm X_B$), see

Table 5-8 and the discussion of the thruster selection process in Section 5-2-3. For simplicity, the lower torque value will be used in the gyroscope calibration process.

The values for the true and estimated state vectors $\mathbf{x}_0$ and $\hat{\mathbf{x}}_0$ are listed in Table 6-2. Initially, the differences in position, velocity and attitude are relatively low, because the lander with an assumed mass of 340 kg has just been separated from the orbiter. As indicated before, the orbiter is tracked during the entire mission to reduce the absolute position error to a few kilometers. The accelerometer calibration is based on the relative motion between the orbiter and the lander in the (imperfect) inertial planetocentric reference frame set by the orbiter. This process is valid as long as both spacecraft experience the same apparent disturbing acceleration – which is in fact the acceleration that follows from the difference between the true and the estimated inertial reference frame. For example, the true gravitational acceleration might be different from the estimated one, but this will not result in a relative positioning error while the lander is still close to the orbiter.

The simulation configurations for both the accelerometer and gyroscope calibration runs are listed in Table 6-3. The normally distributed random noise values and the sampling times of the instruments are taken from the respective Tables in Section 6-1-7. Note, that the star tracker random noise value of 30 arcsec is translated in a quaternion value by performing an Euler rotation of 30 arcsec about one axis, and then take the difference between the initial and the final value of the quaternion element corresponding to the rotation axis. Note, that star trackers return attitude measurements in terms of quaternions. The accelerometer and gyroscope calibration must be performed consecutively for maximal thrust levels. The EKF requires external attitude and position measurements. The ideal response of the spacecraft model to the commanded thruster activities is determined in the environment simulator, which in this case consists of the equations of translational and rotational motion for a basic three-dimensional body moving through the idealized $\frac{\mu_{\text{enc}}}{r^2}$ gravity field. The environment simulator thus returns the lander's true state during the entire simulation time T_f. Every sample time T_r and T_q, the true position and attitude and data are passed to the EKF's instrument models, which add the fixed and random error components to produce the measurement vector $\mathbf{z}_k$. The propagation of the estimated state $\hat{\mathbf{x}}_k^-$ uses the IMU data – corrected with the current best estimates for bias, scale error and misalignments – as input. Both the environmental simulator and EKF use a basic RK4 integrator for state propagation. Note, that in principle the EKF propagator may be a lower-order integrator, because even with perfect measurements, the EKF state estimates will diverge from the actual true states due to discrete character of the measurement inputs. Thus, a more precise integrator will not necessarily translate to a more accurate state estimation. The integrator configuration for the actual Enceladus Lander Simulator is different. All calibration simulation runs are executed in `MATLAB`, the full lander simulator in `C++` will then use the results found in this section. This method is also used in the control-system configuration, see Section 5-2-5.

The prerequisite for a successful gyroscope calibration is a varying rotational rate of the spacecraft. The torque command is chosen such that the rotational rates oscillate between positive and negative values, with a varying amplitude and a phase shift with respect to the other axes. The following torque command was found to produce satisfying results for several

Description	Variable	Value	Unit
true initial state	$\mathbf{x}_0$		
– position	$\mathbf{r}_0$	$(352000, 0, 0)^{\mathrm{T}}$	m
– velocity	$\mathbf{v}_0$	$(0, 0, 0)^{\mathrm{T}}$	$\frac{\mathrm{m}}{\mathrm{s}}$
– attitude[1]	$\mathbf{q}_0$	$(0.2836, -0.5088, 0.3050)^{\mathrm{T}}$	–
– rotational rate	$\boldsymbol{\omega}_0$	$(0, 0, 0)^{\mathrm{T}}$	$\frac{\mathrm{rad}}{\mathrm{s}}$
– acc. bias	$\mathbf{b}_a$	$(1, -2, 3)^{\mathrm{T}} \cdot 10^{-4}$	$\frac{\mathrm{m}}{\mathrm{s}^2}$
– acc. scale error	$\mathbf{s}_a$	$(-1, 1, 2)^{\mathrm{T}} \cdot 10^{-4}$	–
– acc. misalignment	$\mathbf{m}_a$	$(-3, 2, 3, 1, 4, -6)^{\mathrm{T}} \cdot 10^{-4}$	–
– gyr. bias	$\mathbf{b}_g$	$(3, 4, -5)^{\mathrm{T}} \cdot 10^{-3}$	$\frac{\mathrm{rad}}{\mathrm{s}}$
– gyr. scale error	$\mathbf{s}_g$	$(1, 2, -1)^{\mathrm{T}} \cdot 10^{-4}$	–
– gyr. misalignment	$\mathbf{m}_g$	$(1, -1, 2, 1, 4, -3)^{\mathrm{T}} \cdot 10^{-5}$	–
estimated initial state	$\hat{\mathbf{x}}_0$		
– position	$\hat{\mathbf{r}}_0$	$\mathbf{r}_0 + (-10, 30, -101)^{\mathrm{T}}$	m
– velocity	$\hat{\mathbf{v}}_0$	$\mathbf{v}_0 + (-1, -3.9, -19)^{\mathrm{T}}$	$\frac{\mathrm{m}}{\mathrm{s}}$
– attitude	$\hat{\mathbf{q}}_0$	$\mathbf{q}_0 + (0.01, 0.01, 0.01)^{\mathrm{T}}$	–
– acc. bias	$\hat{\mathbf{b}}_a$	$(0, 0, 0)^{\mathrm{T}}$	$\frac{\mathrm{m}}{\mathrm{s}^2}$
– acc. scale error	$\hat{\mathbf{s}}_a$	$(0, 0, 0)^{\mathrm{T}}$	–
– acc. misalignment	$\hat{\mathbf{m}}_a$	$(0, 0, 0, 0, 0, 0)^{\mathrm{T}}$	–
– gyr. bias	$\hat{\mathbf{b}}_g$	$(0, 0, 0)^{\mathrm{T}}$	$\frac{\mathrm{rad}}{\mathrm{s}}$
– gyr. scale error	$\hat{\mathbf{s}}_g$	$(0, 0, 0)^{\mathrm{T}}$	–
– gyr. misalignment	$\hat{\mathbf{m}}_g$	$(0, 0, 0, 0, 0, 0)^{\mathrm{T}}$	–

Table 6-2: True and estimated states for the navigation system simulations. [1]Corresponds to the Euler rotation $(\phi, \theta, \psi) = (20^\circ, -70^\circ, 30^\circ)$

simulation runs:

$$\mathbf{M}_{\mathrm{ref}} = 8.36 \text{ Nm} \cdot \begin{pmatrix} \sin\left(\frac{\pi}{2} + 12\frac{2\pi t}{T_f}\right)\left(1 - \frac{t}{T_f}\right) \\ \sin\left(8\frac{2\pi t}{T_f}\right)\frac{t}{T_f} \\ -\cos\left(6\frac{2\pi t}{T_f}\right)\frac{t}{T_f} \end{pmatrix} \tag{6-106}$$

The reference torque as a function of time is visualized in the first graph of Fig. 6-9. The other two graphs show the changes in attitude and in rotational rate that follow from the thruster activities. Note, that the discontinuities in the attitude curves are a consequence of jumps between -180° and 180°, and do not represent sudden changes in orientation. One goal of the navigation system and the EKF is to find the best possible estimates for these *true* rotational state elements. All input data for the gyroscope calibration simulation runs are listed in Table 6-2 and 6-3. The process noise $\mathbf{Q}$ for the rotational state elements was determined in a trial-and-error process; a process noise of 10^{-6} on each gyroscope misalignment element leads to a convergence of all rotational state estimates within the theoretical limits set by $\mathbf{P}_k^+$. Larger noise values, or process noises on other rotational state elements, lead to saturation of the higher-order estimates $\hat{\mathbf{b}}_g$, $\hat{\mathbf{s}}_g$ or $\hat{\mathbf{m}}_g$, which means, that the estimates hardly change over time. Furthermore, it was necessary to deviate from the rule of thumb for $\mathbf{P}_0^+$ in Eq. (6-105)

Description	Variable	Value	Unit
measurement noise	$\mathbf{R}$	$\begin{bmatrix} \mathbf{R}_r & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{R}_q \end{bmatrix}$	
– pos. random noise	$\mathbf{R}_r$	$0.005 \cdot \mathbf{I}_3$	m
– att. random noise	$\mathbf{R}_q$	$0.53 \cdot 10^{-8} \cdot \mathbf{I}_3$	–
random noise acc.	$\mathbf{v}_a$	20	$\mu\text{g}/\sqrt{\text{Hz}}$
random noise gyr.	$\mathbf{v}_g$	0.002	$^\circ/\sqrt{\text{hr}}$
process noise	$\mathbf{Q}$	$\texttt{diag}(\mathbf{0}_{1\times 27}\ \mathbf{Q}_q)$	
– on $\hat{\mathbf{m}}_g$	$\mathbf{Q}_{m_g}$	$10^{-6} \cdot \mathbf{I}_{1\times 6}$	–
integration step size	Δt	0.005	s
sampling time IMU	$T_{s_{\text{imu}}}$	0.005	s
sampling time pos.	T_{s_r}	0.01	s
sampling time att.	T_{s_q}	0.25	s
simulation length	T_f	120	s

Table 6-3: Setup for the navigation system simulations

for elements on the main axis corresponding to $\hat{\mathbf{b}}_g$, $\hat{\mathbf{s}}_g$ and $\hat{\mathbf{m}}_g$: convergence was reached when these values were set to absolute difference between the true and estimated state.

The attitude and the rotational-rate estimation results are shown in Fig. 6-10. The EKF tracks the quaternion attitude (top graph); these results are translated into Euler angles in the middle graph for a better insight. Only the graph for $\hat{\mathbf{q}}$ incorporates the theoretical limit (dotted line), as only the quaternion attitude is part of the EKF filter - the Euler attitude and the rotational rate are both derived variables. Note, that the Euler angle attitude estimation error values explode at about at $t = 92$ s and $t = 112$ s as a consequence of the jumps between the extremes of the angle range [-180°, 180°] (see middle graph, Fig. 6-9) - this behavior is one of the reasons why the attitude is processed in terms of quaternions. The converged attitude and rotational rate estimation accuracies are in the order of 0.3° and 0.002 $\frac{\text{rad}}{\text{s}}$, respectively. The estimation accuracies of the three main error sources vary significantly. The percentage error can be calculated with

$$\text{percentage error} = \frac{|\text{estimated state} - \text{true state}|}{|\text{true state}|} \times 100\% \tag{6-107}$$

Using the best (latest) values for $\hat{\mathbf{b}}_g$, $\hat{\mathbf{s}}_g$ and $\hat{\mathbf{m}}_g$ from several simulation runs, the estimation error of the gyroscope bias was found to be below 0.1%. The estimation errors of the scale factor and the misalignment can be up to 200% and 900%, respectively. The observability Gramian $\mathbf{\Phi}$ as defined in Eq. 6-27 for the rotational state elements ($\mathbf{q}$, $\mathbf{b}_g$, $\mathbf{s}_g$ and $\mathbf{m}_g$) is 16, which is the full rank and indicates a fully observable system. The values of $\mathbf{\Phi}$, however, range from 480 for the attitude, to about 8 for the bias, to below 1 for the scale factor and misalignments. A lower value indicates a lower observability, see Section 6-1-3. Choosing a different calibration maneuver does change these values, but it does not significantly increase the observability of $\mathbf{s}_g$ and $\mathbf{m}_g$. The main reason high error values are – apart from the star sensor's low update frequency – the limited capabilities of the attitude thrusters: higher thrust levels would lead to a larger – and more correctable – error accumulation, as the spacecraft's rotational velocity has a larger amplitude. On the other hand, the star sensors

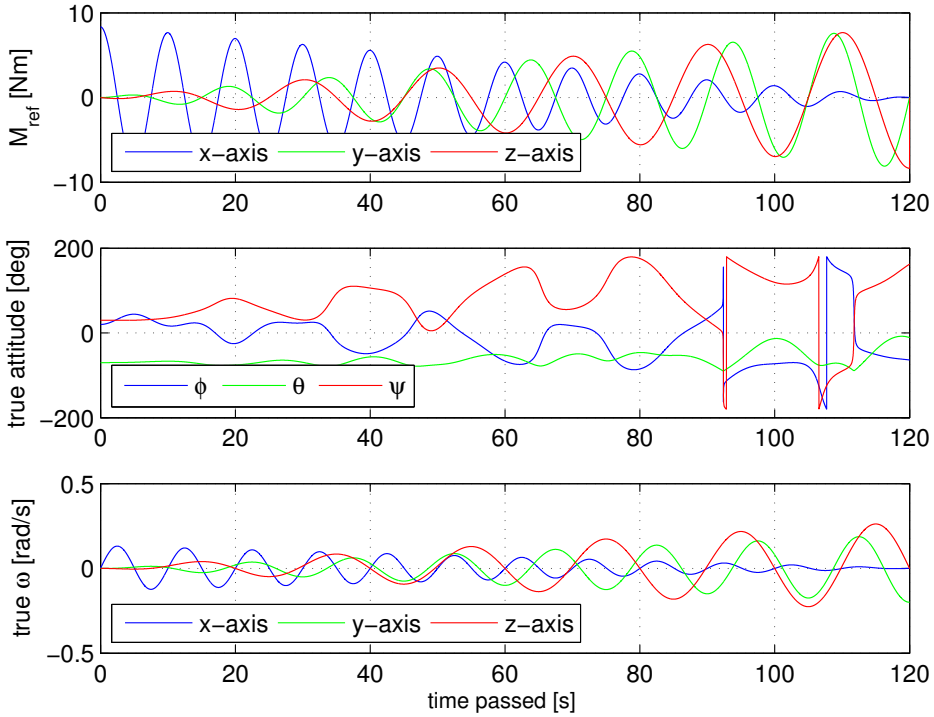


Figure 6-9: Input torque and the resulting true attitude and rotational rate changes

precision deteriorates for higher values of $\boldsymbol{\omega}$ (see Section 6-1-7), so it is questionable whether the estimation accuracy can be improved significantly. At this point, the scale factor and the misalignment cannot be estimated with acceptable accuracy, so they are both not included in the EKF of the Enceladus Lander Simulator.

Similar to the gyroscope calibration process, the thrust commands are chosen such that the resulting acceleration a_{ref} excites all axis periodically with a varying amplitude and a phase shift with respect to the other axes:

$$\mathbf{a}_{\text{ref}} = \frac{12\ \text{N}}{340\ \text{kg}} \cdot \begin{pmatrix} -\sin\left(\frac{\pi}{2} + 2\frac{2\pi t}{T_f}\right)\left(1 - \frac{t}{T_f}\right) \\ \sin\left(\frac{\pi}{2} + 2\frac{2\pi t}{T_f}\right)\frac{t}{T_f} \\ -\sin\left(\frac{\pi}{2} + 3\frac{2\pi t}{T_f}\right)\frac{t}{T_f} \end{pmatrix} \tag{6-108}$$

The maximal thrust of 12 N produces an acceleration of $\frac{12\text{N}}{340\text{kg}} = 0.0353\frac{\text{m}}{\text{s}^2}$, which is not sufficient to overcome the gravitational acceleration of 0.0583 $\frac{\text{m}}{\text{s}^2}$ at an orbital radius of 352 km, so a change of the lander's X_I-position will not oscillate between positive and negative values – in contrast to the Y_I- and Z_I-position changes. The position measurements follow from range and light tracking measurements. The chosen frequency of 100 Hz does not fully exploit the instrument's potential, but for the simulations, the IMU's sampling time must always be below the sampling times of all external instruments. In exchange, it is assumed that the accuracy of the range measurements with a lower measurement frequency can be reduced to 0.5 cm (see Section 6-1-7 for more details on this instrument).

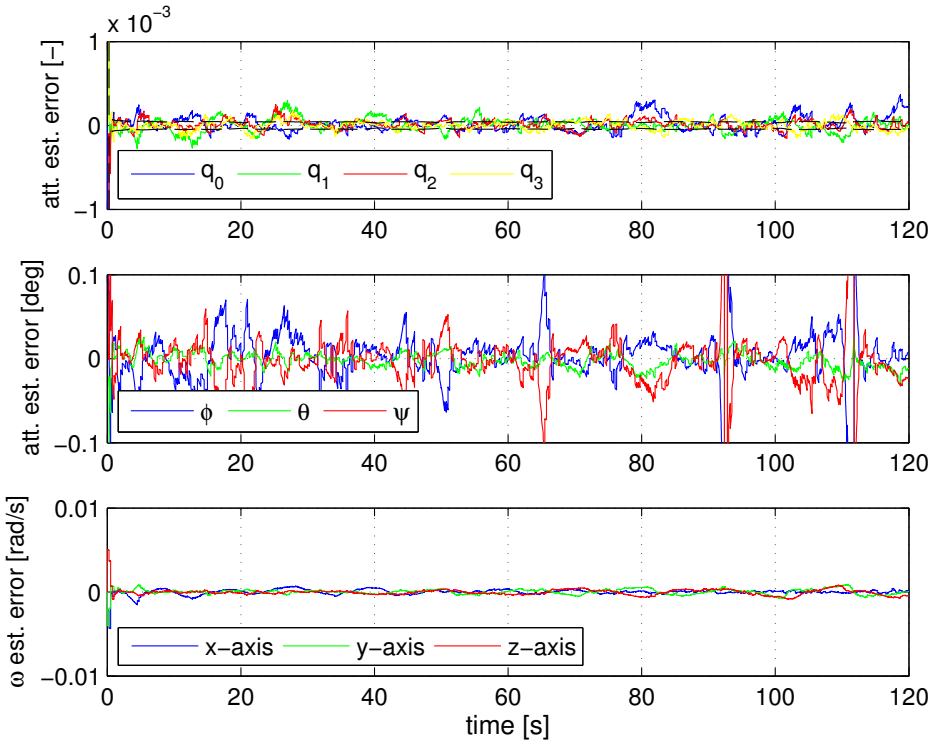


Figure 6-10: Attitude and rotational rate estimation errors for the input torque from 6-9

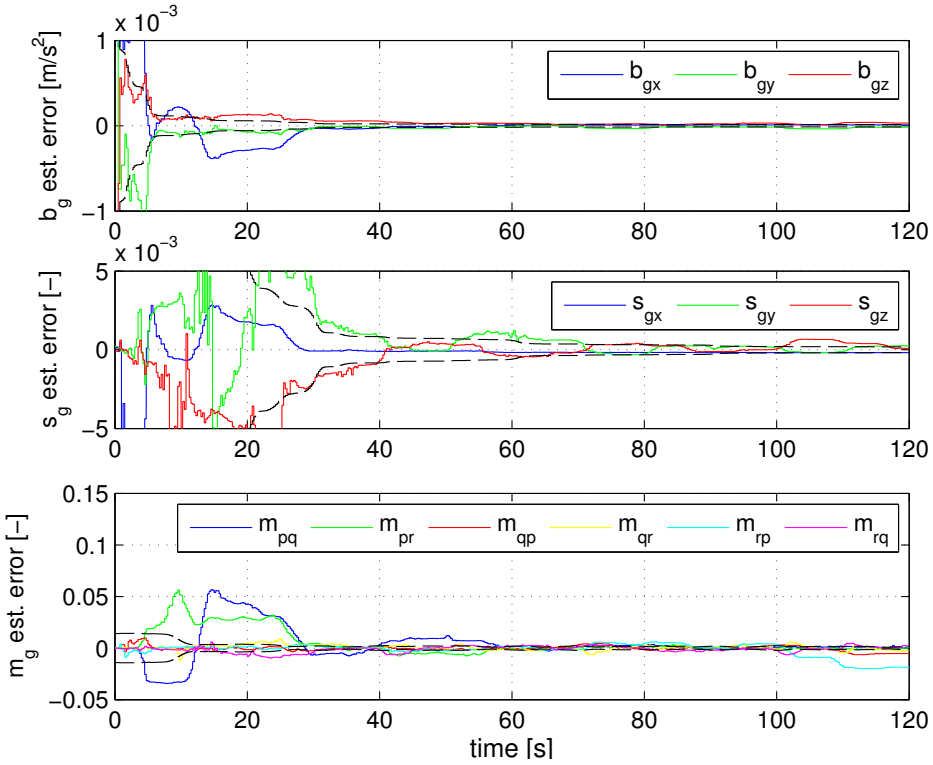


Figure 6-11: Bias, scale factor and misalignment estimation error

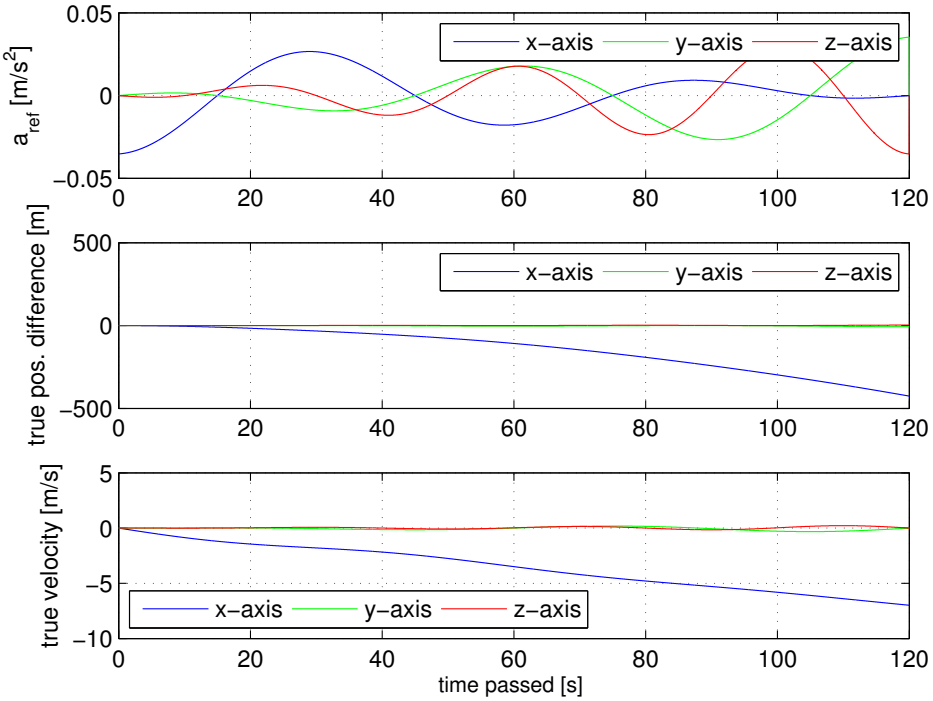


Figure 6-12: Input acceleration and the resulting true position and velocity changes

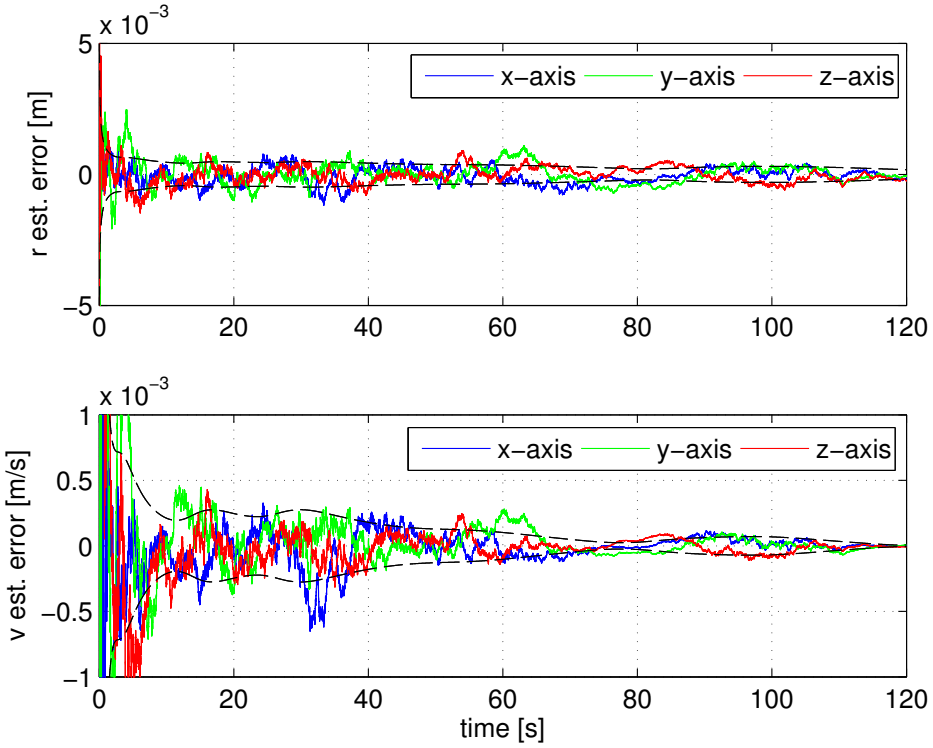


Figure 6-13: Position and velocity estimation errors for the input acceleration from 6-12

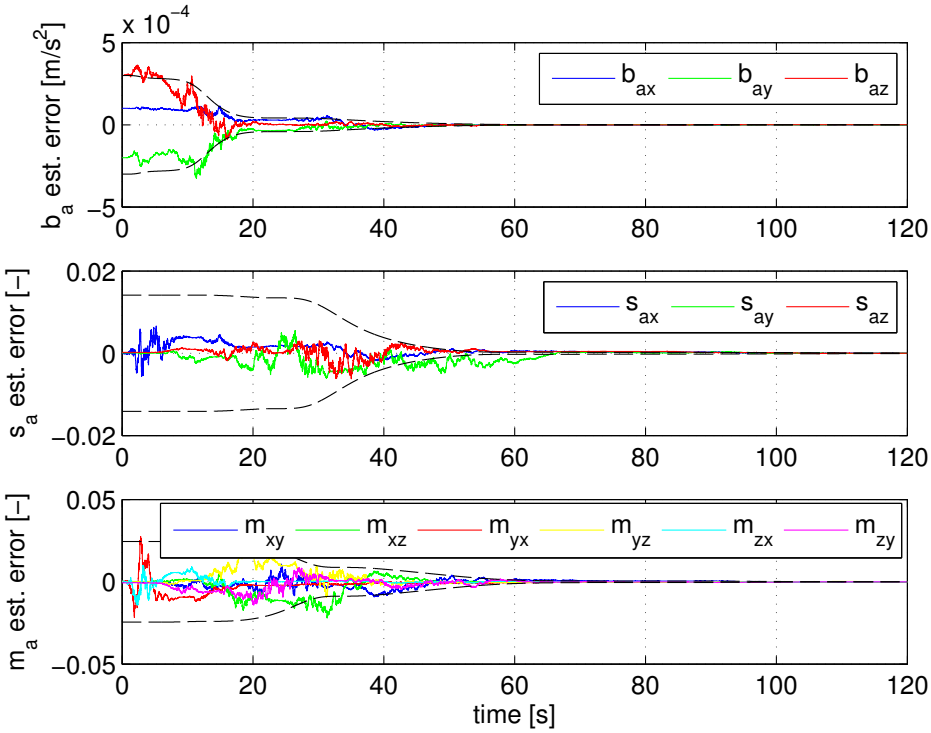


Figure 6-14: Bias, scale factor and misalignment estimation error

The top graph of Fig. 6-12 shows the reference acceleration as a function of time. The mid and bottom picture show the resulting changes in true position and velocity. As indicated before, both the position and the velocity along the X_I-axis decreases and is merely decelerated and accelerated by the thrust firings. The process noise $\mathbf{Q}_r$ for the estimation of the translational state elements is zero for all elements, because all estimations converge within the theoretical limits. However, convergence was only possible when the elements in $\mathbf{P}_0^+$ for $\hat{\mathbf{s}}_a$ and $\hat{\mathbf{m}}_a$ were increased to the maximum error values $2 \cdot 10^{-4}$ and $6 \cdot 10^{-4}$, respectively (see Table 6-2). So in this case, not all elements in $\mathbf{P}_0^+$ follow from Eq. (6-105).

The position and velocity estimation converge rapidly and are in the order of 0.01 m and 0.002 $\frac{\text{m}}{\text{s}}$, see Fig. 6-13. Note, that these estimations might be further improved by including range-rate measurements from the same instrument - currently, they are not included due to the concept character of the range and light tracking system. The best estimates for $\hat{\mathbf{b}}_a$, $\hat{\mathbf{s}}_a$ and $\hat{\mathbf{m}}_a$ from several simulation runs are in the order of 0.1%, 100% and 100%, respectively. An example of the estimation precision for these state vector elements over time is shown in Fig. 6-14. The Gramian for the translational state elements indicates a very low observability for $\hat{\mathbf{s}}_a$ and $\hat{\mathbf{m}}_a$ in the order of 10^{-8}, whereas the values for $\hat{\mathbf{r}}$, $\hat{\mathbf{v}}$ and $\hat{\mathbf{b}}_a$ are 12000, 1.2 and $0.3 \cdot 10^{-4}$, respectively. In fact, some values of $\mathbf{\Phi}$ for the scale factor and misalignment estimation error are so small, that `MATLAB`'s `rank`-function returns 15 instead of the full 18. Adding a velocity measurement to the system, the rank increases to 18, but the estimation accuracy does not improve significantly. Just like the gyroscope calibration simulation, the accelerometer calibration simulations indicates that both scale factor and misalignment cannot be estimated with sufficient accuracy – they will also be excluded from the EKF of the Enceladus Lander Simulator. The true values for $\mathbf{s}_a$ and $\mathbf{m}_a$ remain part of the environment simulator, but their

effect is transferred to $\hat{\mathbf{b}}_a$ and the other state elements during the estimation process.

6-1-7 Available Sensors

This section presents the most important attitude and position determination instruments. Each instrument has specific advantages and disadvantages - a good combination of different instruments will not only increase the redundancy, but also allows for a more rapid data processing and more accurate results. Some instruments used for hazard detection and avoidance, for example the LIDAR, also provide position and attitude data, but these instruments are discussed in Section 6-2.

Optical sensors, such as used for the HAYABUSA mission, are not feasible for the Enceladus mission, because the approach and touchdown phases are much shorter and the velocities much higher without extensive thruster firings – the requirements for the image processing would be too high. To what extent the on-board descent imager could provide additional navigation data might be an interesting subject for further studies, but this is beyond the scope of this report.

Inertial Measurement Unit

A Inertial Measurement Unit IMU is a combination of gyroscopes and accelerometers for determining the angular rate and acceleration of a vehicle. Usually, they consist of three accelerometers and three gyroscopes, and sometimes additional sensors for redundancy. These sensors can only measure *changes* in velocity and orientation, the absolute values must be determined with other instruments. A typical navigation system is based on a combination of star sensors and an IMU (Wertz et al., 2009): The absolute attitude information provided by the star sensors can be updated over time with estimations based on the IMU data. Gyroscopes commonly have a high update rate in the order of 100 to 1000 Hz (Groves, 2008) and are as such very useful to track the spacecraft's orientation during rapid changes induced by the thrusters (Wertz et al., 2009). The IMU processor usually integrates the measured specific forces and angular rates during one sampling interval, and outputs non-dimensional values that provide information about the change in specific impulse and in attitude. The instrument output is then translated into SI units using the IMU characteristic scaling factors (Groves, 2008). The simulator IMU model assumes that the scaling is performed internally, so the instrument outputs are given in terms of the specific force $\mathbf{a}_m$ with respect to the body reference frame, and of the rotational rate $\boldsymbol{\omega}_m$ of the body reference frame with respect to the inertial reference frame. The integration of $\mathbf{a}_m$ and $\boldsymbol{\omega}_m$ to velocity and attitude changes is often performed by the IMU processor, but in the Enceladus lander simulator, this is part of the state propagation in the navigation filter.

The accuracies of the accelerometers and the gyroscopes are limited and change over time. This has a direct influence on the navigation system, so it is necessary to add error models to the instrument simulator. According to Groves (2008), the main error sources are *biases*, *scaling errors*, *misalignment errors* and *random errors*. The IMU processor uses internal calibration libraries, a clock and a temperature sensor to minimize constant offsets, temperature-dependent errors and the errors due to sensor location with respect to the spacecraft's center of mass. The magnitudes of the other error sources change in-between or even during one measurement cycle, and must be corrected or tracked outside of the IMU.

A bias is a error independent of the current specific force and rotational rate. They have, in general, the largest influence on the total measurement error (Groves, 2008). The bias in accelerometers is constant, whereas the bias of gyroscopes increases over time. The gyroscope bias is called *drift*, and is measured in the non-SI unit deg/hr and describes the loss of accuracy over time. The drift rate depends on the used technique, and can range from 0.0007 to 0.2 deg/hr in space applications (Wertz et al., 2009). As the gyroscopes can be restarted on Enceladus' surface and a repositioning cycle lasts a few minutes at most, the IMU is a very suitable attitude determination system for a moving lander.

Scaling errors are a consequence of internal imprecisions in the conversion process between the instrument input and output. They scale linearly with the true specific force or rotational rate.

Misalignment errors arise, when the instrument axes are not perfectly aligned with the spacecraft's body reference frame. In this case, the measurement of an excitation along one body axis will have small components along the other axes. Both scaling errors and misalignments can be minimized by an elaborated on-ground calibration process, but they will always be present to some extent. Random noise in a given system, on the other hand, cannot be calibrated. It is commonly expressed in terms of root PSD, namely $\mu g/\sqrt{Hz}$ for the accelerometers, and $^\circ/hr/\sqrt{Hz}$ for the gyroscopes (Groves, 2008). The standard deviation then follows from multiplying the root PSD with the root measurement frequency.

Accelerometers measure the relative movements of the internal test masses with respect to the moving body on which they are mounted. This movement is the result of a specific force and is translated internally to the output acceleration. Usually, the distance between the test masses and the lander's center of gravity is relatively small, so the gravitational acceleration on the lander is the same as on the instrument, and is not registered by the accelerometer until touchdown. Consequently, the gravitational acceleration should always be included as a separate element in the state propagation. This requires the implementation of gravitational models in the navigation system, which will inevitably lead to an offset between the true and the estimated total acceleration.

IMU element	Name	Variable	Value	Comment
Accelerometer	bias	$\mathbf{b}_a$	0.01 mg	$\propto$ local g
	misalignment	$\mathbf{m}_a$	10^{-4}-10^{-3}	no unit
	scale error	$\mathbf{s}_a$	10^{-4}-10^{-3}	no unit
	random error	$\mathbf{v}_a$	$20\mu g/\sqrt{Hz}$	$\propto$ local g
Gyroscope	bias	$\mathbf{b}_g$	0.001 hr^{-1}	-
	misalignment	$\mathbf{s}_g$	10^{-4}-10^{-3}	no unit
	scale error	$\mathbf{m}_g$	10^{-5}-10^{-4}	no unit
	random error	$\mathbf{v}_g$	$0.002\ ^\circ/\sqrt{\mathrm{hr}}$	-
System	measurement frequency	f_{imu}	100 - 1000 Hz	

Table 6-4: IMU model characteristics (Groves, 2008). Note that this are approximate values for space grade systems.

The preliminary lander design includes an IMU with a ring laser gyroscope, which offers higher accuracies then the spinning mass gyroscopes. Table 6-4 lists the instrument characteristics

that will be used in the navigation filter of the Enceladus lander simulator. Note that there exist a wide range of different gyroscopic systems, so the actual numbers might diverge from the given specifications. The random error here has a uniform distribution.

Star Sensors

Star sensors determine the direction of one or more stars within their instrument reference frame. The spacecraft attitude then follows from comparing the measurement results with a star catalog. The star sensors evolved rapidly during the last years as a consequence of the advances in the charge-coupled device (CCD) technology. Current star sensors have low power requirements, are relatively light and have built-in processing electronics, including the star catalog. Many instruments can also monitor multiple stars over time, which has positive effects on the accuracy; these star sensors are called *star trackers*. CCD star sensors consist of an optical system with a lens and a measurement plane with photocells. The light rays of a star illuminate a number of the light-sensitive cells. The processing system analyzes the light intensity from each cell and calculates the focus. The plane angles of the star follow combining the focus location with the characteristics of the optical system. The light intensity also allows a correlation with a certain star of the star catalog, which ultimately yields the orientation of the spacecraft with respect to that star.

Star trackers have a higher accuracy about the axes normal to the center line (boresight) of the CCD plane, which is consequence of the more complex star vector separation in small field-of-views (Pisacane, 2005). The accuracy of star trackers decreases significantly for rotational velocities above some degrees per second, unless some image stabilization software is incorporated. Additionally, higher rotational speeds reduce the probability of a successful attitude determination (VECTRONIC, 2012). The preliminary design uses two VST-41M star trackers, mounted on the sides of vehicles body. The specifications of this instrument can be found in Table 6-5. The accuracy is given in terms of 2σ values off boresight and about boresight, but according to Pisacane (2005), it is necessary to consider additional error sources, such as optical errors, velocity aberration, catalog uncertainties, noise equivalent angle and more. Table 6-6 lists some general star tracker data, including information about systematic and random errors.

Characteristic	Value	Comment
accuracy 2σ x-,y-axis	18 arcsec	off boresight
accuracy 2σ z-axis	122 arcsec	about boresight
acquisition probability	>99.7 %	for angular rates <0.8 degs^{-1}
field of view	$14^\circ \times 14^\circ$	–
time to first acquisition	800 ms	typically
update rate	4 Hz	after first acquisition
dimensions	80 mm × 100 mm × 180 mm	–
power requirement	2.5 W	-
mass	1.1 kg	–

Table 6-5: VST-41M characteristics (VECTRONIC, 2012)

Characteristic	Value	Comment
accuracy	0.0003° - 0.01°	–
random error	below 0.1 arcsec	$= 2.78 \times 10^{-5}$ deg
sampling time	0.25 s	= 4 Hz
power requirement	5 - 20 W	for a pair
mass	2 - 5 kg	for a pair

Table 6-6: Star sensor characteristics (Wertz et al., 2009; Pisacane, 2005). Note that star sensors currently undergo a rapid evolution.

Horizon Sensors

Horizon sensors are passive infrared instruments that determine the location of the transition between the warm surface and the cold space. There exists a wide range of horizon sensor types, more than for any other attitude sensor (Wertz et al., 2009). The focus in this section will be on conical scanning horizon sensors for 3-axis stabilized spacecrafts, as these instruments are most suitable for near-surface measurements with a relatively high level of accuracy. The direction of the center of the planet can be inferred from three or four sufficiently separated transition points (Parkes and Silva, 2002). However, if the spacecraft's altitude is known from other sensors, two points are sufficient. This allows the use less complex and lighter instruments. Horizon sensors in general suffer from two problems. Firstly, the terrain relief disturbs near-surface measurements. Attitude data from altitudes below ca. 1 km is probably not useful (Parkes and Silva, 2002). Secondly, horizon sensors used on celestial bodies with an atmosphere are less accurate as a consequence of the temperature extremes between the lit and dark sections of the surface (Wertz et al., 2009). The second downside is not applicable to Enceladus with its negligible atmospheric density. Horizon sensors are an important attitude determination system due to their reliability, proven design and technique, which is very different from the working principle of the other attitude sensors - this is beneficial for the redundancy.

Table 6-7 lists some general characteristics static and scanning horizon sensors.

Characteristic	Value	Comment
accuracy	0.02° - 1°	scanning sensor more accurate
random error	0.05° - 0.5°	
update rate	1 Hz	estimation for scanning sensor
power requirement	0.3 - 10 W	scanning sensor more demanding
mass	0.5 - 4 kg	scanning heavier than static

Table 6-7: Static and scanning horizon sensor characteristics (Wertz et al., 2009). Comparisons under *Comment* always with respect to static sensors.

Magnetometer

The investigation of Enceladus' influence on the magnetic field of the other bodies in the Saturnian system is part of the scientific goals, so a combination of a plasma detector and magnetometer is part of the lander's preliminary scientific instrumentation package (Ampe et al., 2009). This three-axes fluxgate magnetometer measures the local gradient of the magnetic field with a high accuracy. In case the magnetic field of the central body is mapped and its changes due to solar activity, orbital position of the body and others can be modeled, magnetometers offer a very simple, low-cost, lightweight and reliable way to determine a spacecraft's attitude (Acuna, 2002).

A major disadvantage on the other hand is the relatively low accuracy, especially for low-altitude orbits: Even for Earth, whose magnetic field is the extensively studied, the accuracy is about 5° at 200 km altitude (Wertz and Larson, 1999). Enceladus magnetic field and its variability is widely unknown, and many measurements are taken near the surface during the repositioning process. Consequently, magnetometers will not be part of the attitude determination system, but will be used as scientific payload. In case of emergency, they can, however, provide rough attitude information.

Characteristic	Value	Comment
accuracy	0.5° - 5°	highly conditional, from Wertz and Larson (1999)
random error	0.35° - 1.1°	
update rate	high	continuous
power requirement	less than 1 W	-
weight	0.3 - 1.2 kg	-

Table 6-8: Magnetometer characteristics (Wertz et al., 2009)

Directional Antenna

The preliminary design of the lander includes two helical, redundant antennas with a length of 40 cm and a diameter of 5 cm (Ampe et al., 2009). These antennas have a beamwidth of about 25°, and can each be aligned with a tracking and pointing system. The directional characteristics of any antenna allow the determination of the antenna adjustment, which in turn provides information about the attitude of the spacecraft. Directional antennas in general not part of the attitude determination system, because it requires a constant link to a second signal source. In case of Silenus, the orbiter can act as a reference during the contact phases. The accuracy of this system is between 0.01° and 0.5°, typically 1 % of the antenna bandwidth (Wertz and Larson, 1999), thus about 0.25° for the preliminary lander design. The directional antennas can be uses as an additional source of attitude data, which increases the redundancy and (possibly) the accuracy of the attitude determination system. Table 6-9 lists some general characteristics of the attitude determination with directional antennas.

Characteristic	Value	Comment
Accuracy	0.01° - 5.0°	about 1% of bandwidth (Wertz and Larson, 1999)
Systematic Error	various	instrument is part of payload
Random Error	various	instrument is part of payload
Total Error	various	instrument is part of payload
Power Requirement	[side product]	-
Weight	various	-

Table 6-9: Directional antenna characteristics (Wertz et al., 2009)

Range and light tracking

Range and range-rate navigation derives position and velocity information from the travel time and the phase shift of an emitted electromagnetic signal. The signal itself contains the sending time, so the receiving spacecraft on-board computer compares this time with the arrival time to determine the travel time. Obviously, the on-board clock must be synchronized with the transmitter's clock. The Doppler shift of the signal contains information about the relative velocity between the transmitter and receiver, so if the position and velocity of the transmitter is known, the velocity of the receiver can be derived. This system is, in fact, the working principle of the GPS satellites. When range measurements from at least two different sources are available and the spacecraft position is known, it even possible to determine the receiver's attitude by carrier phase measurements, which requires two receiver antennas at different locations on the spacecraft. An unambiguous position estimation, on the other hand, requires three different sources, and only one receiving antenna.

A limiting factor to signal range measurements is the exact determination of the arrival time. Modern military GPS receivers have a timing accuracy of about 1 ns, which leads to a range accuracy of 0.3 m (Mio, 2013). The LIDAR system discussed in the next section suffers from similar limitations. Such a ranging accuracy would to large pointing errors and thus position errors, in case the distance between the three transmitters and receivers is too small. A solution might be to install a light or a beacon on the lander and use a instrument similar to a sun sensor to determine the elevation and azimuth of the signal with respect to the orbiter. A single range measurement, using either time information in the beacon signal or a separate range measurement system, would then fully determine the position of the lander with respect to the orbiter. This also would allow to chose a more complex range finder with a higher accuracy. Modern LIDAR systems can reach accuracies in the order of 2 to 10 cm with range measurement frequencies up to 10 kHz, see Liebe et al. (2003) and Section 6-1-7. A laser altimeter developed recently developed by John Hopkins APL for landing applications on low-gravity planetary bodies reached accuracies up to 1 cm with an desired update rate between 1 and 5 kHz (Bruzzi et al., 2012).

Sun sensors are widely used attitude determination system, that determines the position of the Sun on the spacecraft-centered celestial sphere by identifying the transition between light and shadow on the optical cells. The accuracy is about 0.18° for simple analogue sun sensors (Astrium, 2013), so the accuracy of a specially designed system to track a distinct

light on another spacecraft will be higher. Analogous Sun sensors allow a continuous read-out of measurement data, so in case the same technique can be used, the limiting factor for the update frequency for a combined range and light tracking instrument are the range measurements. Table 6-10 lists the estimated performance of a combined range and light tracking system. The error off the line of sight will depend on the distance to the target, but as a first approximation, the distance between the orbiter and lander is assumed small, which is true for some time after the separation.

Characteristic	Value	Comment
update rate	$>$ 1kHz	
accuracy	0.02 m	estimated 1σ, varies off-axis
mass	ca. 1.5 kg	sun sensor about 0.25 kg

Table 6-10: Estimated characteristics of a range and light tracking system for relative position determination, see text for discussion

LIDAR

The light detection and ranging LIDAR instrument measures the distance to an object by emitting a laser pulse and recording the time of the return. The range then follows from dividing the time-of-flight by two times (one way) the speed of light. Scanning LIDARs use a continuous stream of light pulses, which is reflected by a rotating mirror scan a target area (Johnson et al., 2002). The mirror kinematics must be precisely known for a correct range calculation, but there will be an extra noise. The measurement results are then combined in an elevation map, which forms the basis of the hazard avoidance system, and of course gives distance information for close-target navigation. LIDARs are usually designed specifically for one mission, which results in very different instrument characteristics. The preliminary lander design uses a LIDAR comparable to the HAYABUSA mission; its specifications can be found in Table 6-11. Currently, the Jet Propulsion Laboratory is building a dual-axis gimbaled mirror LIDAR for a hazard avoidance purposes. The specifications are unknown yet, some estimations can be found in Table 6-12. For more information about the LIDAR in the context of hazard avoidance, see Section 6-2.

Characteristic	Value	Comment
repetition rate	1 Hz	-
footprint size	12.0 m $\times$ 4.9 m	at 7 km
range	50 m to 50 km	also used at 30 m
range accuracy	10 m at 50 km, 1 m at 50 m	relatively constant
range resolution	0.5 m	-
power requirement	22 W	-
mass	3.56 kg	-

Table 6-11: Characteristics of the HAYABUSA LIDAR (NASA, 2012a)

Characteristic	Value	Comment
sample rate	10 kHz	sample = 1 range point
field of view	$10^\circ \times 10^\circ$	-
range	about 50 m to 2 km	-
range resolution	0.02 m	-
range error	0.02 m	1σ-value
pointing error	0.001°	1σ-value
pointing resolution	0.001°	-
divergence	0.1°	-

Table 6-12: Estimated characteristics of the JPL LIDAR (Johnson et al., 2002)

6-1-8 Chosen Sensors

The navigation system of the preliminary design of the Enceladus lander is based on two star sensors, two horizon sensors, one IMU and one LIDAR. This choice is still reasonable, because it covers the required input of external measurements including redundancy. The IMU is a very efficient instrument to propagate the vehicle's velocity and position, in particular during the short repositioning cycles. The IMU is thus the critical navigation sensor for soft landings, as it allows for a continuous measurement of accelerations and rotations. The star sensors are the main attitude determination instrument, but they cannot be used on a (rapidly) rotating spacecraft. Star sensors are very often used in combination with IMUs, because they provide absolute attitude data needed to propagate the relative measurements from the IMU. The periods between the reposition cycles allow for a very precise attitude estimation, as the stars can be tracked over a long period of time. The accuracies of the attitude and acceleration measurements are very high (see Table 6-4 and 6-6) and meet the requirements stated in the previous section. The two horizon sensors were intended to aim at Saturn and at Enceladus during the orbital phase of the spacecraft, as an additional way to determine the attitude. However, star sensors are more accurate, have a lower weight and can be used on the surface of moon. Horizon sensors might be an option for redundancy, but they will not increase the overall accuracy of the attitude measurements and are thus excluded from the lander. The LIDAR is the main hazard avoidance instrument. This instrument was successfully used for the HAYABUSA mission, in combination with optical navigation cameras. The preliminary hazard avoidance system of the *Altair* lunar lander is also based on an active scanning LIDAR (see Section 5-1).

6-1-9 Simulator Configuration: Navigation System Parameters

The navigation system in the Enceladus Lander Simulator consists of the EKF with a state vector consisting of the position, velocity, attitude, accelerometer bias and gyroscope bias. As discussed in Section 6-1-5, the misalignment and scale factor errors cannot be estimated with sufficient with sufficient accuracy and are not included in the actual filter – their true values, however, remain part of the IMU model.

The navigation system related simulator parameters are listed in Table 6-13, with references to the variable names used in the previous navigation system sections. Just like the previous parameter lists, variable names are only listed once in the table if only their axis designation changes – in those cases, an indication of the number of excluded variable names is added in brackets.

Variable Name	Comment
General	
`propagationStepSizeFactor`	[s], Δt in Table 6-3
External Instruments	
`frequencyRangeInstrument`	[1/s], f_{pos}, see Table 6-10
`frequencyStarSensors`	[1/s], f_{att} , see Table 6-6
`measurementNoiseRange`	[m], 1σ error, $\mathbf{v}_r$
`measurementNoiseStarS`	1σ **q**-error, $\mathbf{v}_q$
IMU Data	
`frequencyIMU`	[1/s], f_{imu}, see Table 6-4
`randomNoiseAccelerometer`	[m^2], 1σ error, $\mathbf{v}_a$ in Table 6-3
`randomNoiseGyroscope`	1σ **q**-error, $\mathbf{v}_g$ in Table 6-3
`processNoisePosition`	[m^2], σ^2, $\mathbf{Q}_r$
`processNoiseVelocity`	[m^2/s^2], σ^2, $\mathbf{Q}_v$
`processNoiseAttitude`	[-], σ^2, $\mathbf{Q}_q$
`processNoiseBiasAcc`	[m^2/s^4], σ^2, $\mathbf{Q}_{b_a}$
`processNoiseBiasGyr`	[rad^2/s^2], σ^2, $\mathbf{Q}_{b_g}$
`covariancePosition`	[m^2], initial σ^2, $\mathbf{P}^+_{0,r}$
`covarianceVelocity`	[m^2/s^2], initial σ^2, $\mathbf{P}^+_{0,v}$
`covarianceAttitude`	[-], initial σ^2, $\mathbf{P}^+_{0,q}$
`covarianceBiasAcc`	[m^2/s^4], initial σ^2, $\mathbf{P}^+_{0,b_a}$
`covarianceBiasGyr`	[rad^2/s^2], initial σ^2, $\mathbf{P}^+_{0,b_g}$
`biasAccelerometerX` (+2)	[m/s^2], $b_{a,x}$ in Table 6-2
`biasAccelerometerXest` (+2)	[m/s^2], initial estimated $b_{a,x}$
`biasGyroscopeX` (+2)	[rad/s], $b_{g,x}$ in Table 6-2
`biasGyroscopeXest` (+2)	[rad/s], initial estimated $b_{g,x}$
`scaleErrorAccelerometerX` (+2)	[-], $s_{a,x}$ in Table 6-2
`scaleErrorAccelerometerXest` (+2)	[-], initial estimated $s_{a,x}$
`scaleErrorGyroscopeX` (+2)	[-], $s_{g,x}$ in Table 6-2
`scaleErrorGyroscopeXest` (+2)	[-], initial estimated $s_{g,x}$
`misalignmentAccelerometerXY` (+5)	[-], $m_{a,xy}$ in Table 6-2
`misalignmentAccelerometerXYest` (+5)	[-], initial estimated $m_{a,xy}$
`misalignmentGyroscopeXY` (+5)	[-], $m_{g,xy}$ in Table 6-2
`misalignmentGyroscopeXYest` (+5)	[-], initial estimated $m_{g,xy}$

Table 6-13: State vector program parameters

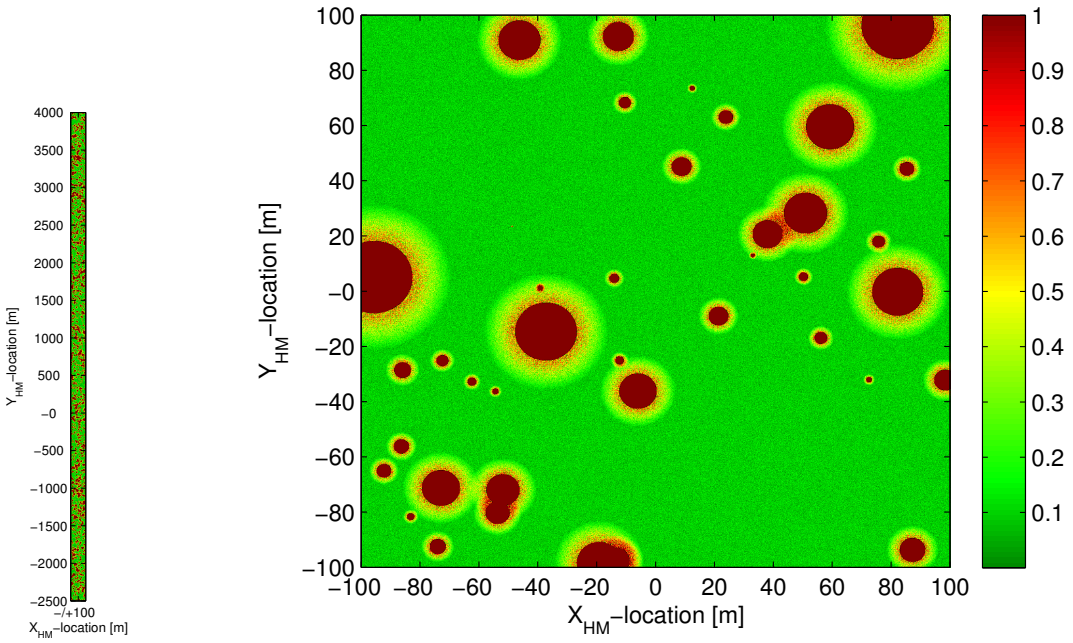
6-2 Hazard Avoidance

The hazard avoidance system of the Enceladus Lander is based on the analysis of the *hazard map* of the target area. The hazard map indicates the landing risk for a certain landing area by assigning values between 0 for no risk and 1 for maximum risk to each image pixel (Parreira et al., 2008). The hazard map is the result of the LIDAR image analysis process: The LIDAR generates a topography model of the target area's surface. This three-dimensional map is then translated into incidence angle and surface roughness maps (Johnson et al., 2002). The maximal incidence angle and maximum obstacle height for Enceladus lander are 15° and 0.5 m, respectively, see Chapter 3 and Ampe et al. (2009). Each pixel on both maps receives a hazard value between 0 and 1. The normalized combination of these two maps is a surface hazard map; the mission will fail if the touchdown takes place in the region of a pixel with a value equal to 1. The area required for a safe touchdown is called *Vehicle Dispersion Footprint Ellipse* (VDFE), and it consists of the area defined by the lander model diameter and uncertainties caused by the GNC system errors (Johnson et al., 2008). Consequently, all areas on the surface hazard map within one VDFE around a pixel with a value equal to 1 are inaccessible, so the according pixels all receive the value 1 as well. The result is the full hazard map, which gives a realistic impression of the landing risks in the target area. The derivation the incidence angle and the roughness map from the LIDAR topography model is complex and beyond the scope of this report. The surface hazard map is thus assumed to be an input for the lander's hazard avoidance system. The *true* surface hazard maps for both descent and repositioning are generated before the actual landing simulations for the regions around the nominal target spots.

6-2-1 True Surface Hazard Map Generation

The true surface hazard map (SHM) is defined in the $Y_S Z_S$-plane, with the origin coinciding with the $y_S - /z_S-$coordinates of the nominal target position. The X_{SHM}-axis is collinear with the Y_S-axis and points in the same direction, while the Y_{SHM}-axis is collinear with the Z_S-axis, but points in the opposite direction (see the definition of the surface-fixed frame, Fig. 4-4). The true SHM must be large enough to allow for LIDAR scans from all viewing angles about the target position, including aiming errors. The instrument's field-of-view is about 100 m × 100 m for the scan altitude (see Section 6-2-2), while the repositioning target can be anywhere between approximately -2500 m and 4000 m on the Y_S-axis (see Section 8-4). Consequently, the true SHM has the dimensions of 200 m × 6500 m, with the origin located at the nominal repositioning target of $y_S = 1000$ m and $z_S = 0$ m.

The topography of Enceladus' is subject of current scientific research, so there currently exists no exact data on the average rock size and rock density on the surface. Based on the discussion in Section 2-2 and the surface images shown in Fig. 2-6, the average rock size is chosen to be 10 m. These rocks will cover about 15% of the target area, which is more than comparable terrain models for Mars (Sinclair and Fitz-Coy, 2003). The SHM generator assumes, that the boulders have a circular shape with a normally distributed diameter $d_{rock} \sim \mathcal{N}(10\text{m}, 1\text{m}^2)$. The rocks are randomly placed on the SHM and may overlap. The lander's ground clearance is 0.5 m in the preliminary design (Ampe et al., 2009), so all map pixels with values larger than 0.5 m indicate a hazardous landing spot and are thus set to 1. The maximal inclination angle of 15° does not add further constraints, because a boulder of 0.5 m in combination with



(a) scaled map **(b)** detail of origin

Figure 6-15: True surface hazard with a resolution of 0.1 m

the lander diameter of 4.5 m only leads to an inclination angle of about 6°. Figure 6-15b shows a detail of the true SHM for the repositioning with the focus on the map origin – the actual Y_{SHM} ranges over 6.5 kilometers, as indicated in Fig. 6-15a. The resolution is set 0.1 m, which is clearly below the maximal boulder size. The map indicates the landing risk for the terrain map only – the LIDAR sees only the section within its field-of-view, and the lander's on-board hazard avoidance system uses this input to determine the best landing spot.

6-2-2 LIDAR Surface Hazard Map

The sub-satellite point (SSP) on the true surface hazard map for a spacecraft at the position $\mathbf{x}_S$ depends on the nominal target $\mathbf{x}_{t_{nom},S}$, which defines the origin of the SHM:

$$SSP_{SHM} = \begin{pmatrix} -z_S \\ y_S \end{pmatrix} - \begin{pmatrix} -z_{t_{nom},S} \\ y_{t_{nom},S} \end{pmatrix} \tag{6-109}$$

The minus sign in front of the first vector elements is a consequence of the opposite pointing direction of the Z_S- and the X_{SHM}-axis. The Enceladus lander aims at the initial target $\mathbf{x}_{t_0,S}$, which coincides with the origin of the true SHM for the nominal simulation run. The spacecraft position error causes a divergence of the line-of-sight on the horizontal plane, so the LIDAR now aims at the position

$$\mathbf{x}_{t,SHM} = \begin{pmatrix} -z_{t_0,S} \\ y_{t_0,S} \end{pmatrix} - \begin{pmatrix} -z_{t_{nom},S} \\ y_{t_{nom},S} \end{pmatrix} + (SSP_{est,SHM} - SSP_{true,SHM}) \tag{6-110}$$

where the difference between the estimated and the true position is the position estimation error, on the $Y_S Z_S$-plane. The lander is generally not exactly located above the Y_S-axis in

the horizontal plane, so the LIDAR sees the target $\mathbf{x}_{t,SHM}$ at a certain azimuth angle. The SHM image rotation depends on the true SSP and the true target $\mathbf{x}_{t,SHM}$, and follows from the problem geometry. The counter-clockwise rotation angle ϕ_{SHM} is:

$$\phi_{SHM} = -\tan^{-1}\left(\frac{SSP_{x,true,SHM} - x_{t,SHM}}{SSP_{ytrue,SHM} - y_{t,SHM}}\right) \tag{6-111}$$

The target area is scanned at the maximum altitude for which all hazardous surface elements are detectable to maximize the field of view. With a LIDAR angular resolution of $\Delta\theta = 0.1^\circ$ and a the maximum rock size of 0.5 m, the scan altitude is about 290 m. It will be assumed, that the elevation angle between the local vertical and the LIDAR line-of-sight is 90° for the image acquisition – other elevation angles introduce tilt effects, which significantly increase the problem complexity. The LIDAR's field-of-view now is a square area, with the center $\mathbf{x}_{t,SHM}$, rotated over the angle ϕ_{SHM} with respect to the true SHM, and with a side length of

$$d_{FOV} = 2h\tan\frac{\theta_{FOV}}{2} \tag{6-112}$$

where θ_{FOV} is the instrument's angular field-of-view. The LIDAR takes measurements each $\Delta\theta$ along the scan lines. The relation between a position on the LIDAR image frame and the true SHM is

$$\begin{aligned}\mathbf{x}_{SHM} &= \mathbf{x}_{t,SHM} + \begin{bmatrix}\cos\phi_{SHM} & \sin\phi_{SHM}\\ -\sin\phi_{SHM} & \cos\phi_{SHM}\end{bmatrix}\mathbf{x}_{lidar}\\ &= \mathbf{x}_{t,SHM} + \mathbf{T}_{SHM\leftarrow lidar}\mathbf{x}_{lidar}.\end{aligned} \tag{6-113}$$

Note, that a LIDAR image pixel does not generally have a resolution of 1 m, so there is an additional translation – derived from Eq. (6-112) – required to get from a position on the image to $\mathbf{x}_{lidar}$.

Name	Value
Initial State	
estimated position	$[290.00, 200.00, 50.00]$ m
true position	$[290.00, 198.00, 52.00]$ m
estimated velocity	$[-2.00, 3.00, -1.00]$ m/s
true velocity	$[-2.00, 3.00, -1.00]$ m/s
Target State	
nominal target position	$[0.00, 500.00, 0.00]$ m
current target position	$[0.00, 500.00, 0.00]$ m

Table 6-14: Inital and target state information for the hazard avoidance simulations

The initial and target state information for the hazard avoidance simulation in this section are listed in Table 6-14. The target is chosen to be $\mathbf{x}_{t_{nom},S}$, so $\mathbf{x}_{t_0,S}$ coincides with the origin of the true SHM. The true initial SSP is not located above Y_S-axis relative to the Y_SZ_S-plane; the image rotation ϕ_{SHM} is -9.15° based on Eq. (6-111). The position estimation error causes

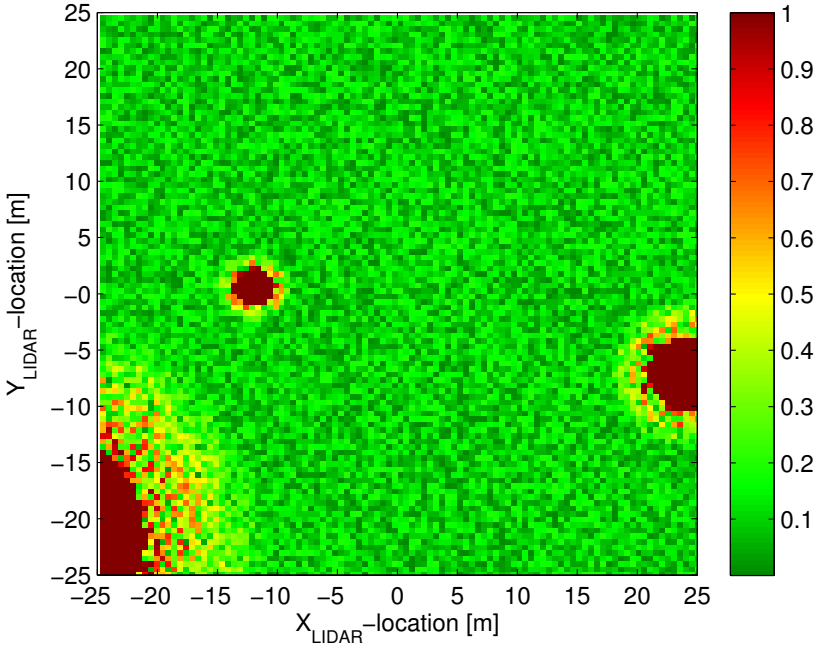


Figure 6-16: Surface hazard map generated with the LIDAR

an image shift of -2 m in both directions of the true SHM. Figure 6-16 shows the surface hazard map generated with the LIDAR based on the true SHM given in Fig. 6-15b.

The field-of-view at an altitude of 290 m is about 76 m for $\theta = 15°$, which is relatively small for a lander with a base diameter of 4.5 m. It might be necessary to use a LIDAR with higher angular resolution or with a wider field-of-view.

6-2-3 Piloting

Piloting is the complete process of selecting a safe landing spot on a given risk map. In general, a piloting function first determines values for the fuel consumption, the visibility, the thrust profile and more for each candidate point, and then combines this data in a score map, which forms the basis for the retargeting choice (Parreira et al., 2008). The Enceladus Lander Simulator is limited to a basic reachability and fuel-consumption analysis.

The surface hazard map in Fig. 6-16 does only contain hazard information about the (small) areas related to each image pixel. The Enceladus lander, however, has a certain base diameter, and can never exactly land at the desired position due to the GNC errors during descent and repositioning. The *Vehicle Dispersion Footprint Ellipse* (VDFE) is a combination of these two values, and thus describes the minimal (circular) area that should be free of any hazardous elements for a safe landing (Johnson et al., 2008). The landing precision of the lander is in the order of 2 m, see Section 8-4. With a lander base diameter of 4.5 and some contingency, the VDFE is chosen to be 9 m. To determine the safe landing areas from the LIDAR image, a circle with a radius equal to the VDFE is drawn around each pixel on the SHM with a value larger than 0.9. All areas inside one or more circles are inaccessible, and all areas outside are considered candidate points for a safe touchdown. The reachability map for Fig. 6-16

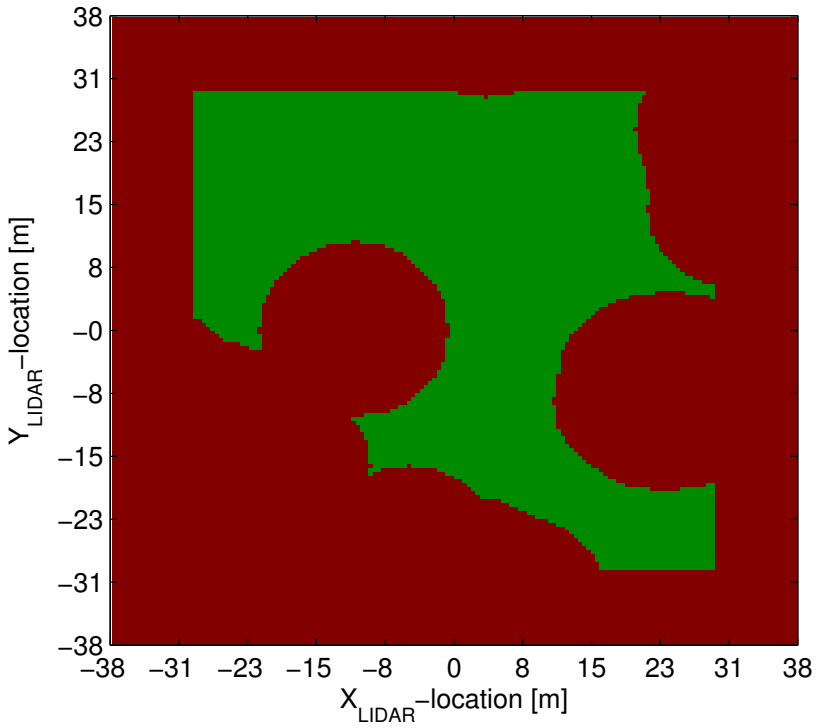


Figure 6-17: Reachability map based on Fig. 6-16 – green marks the candidate points, red the hazardous area.

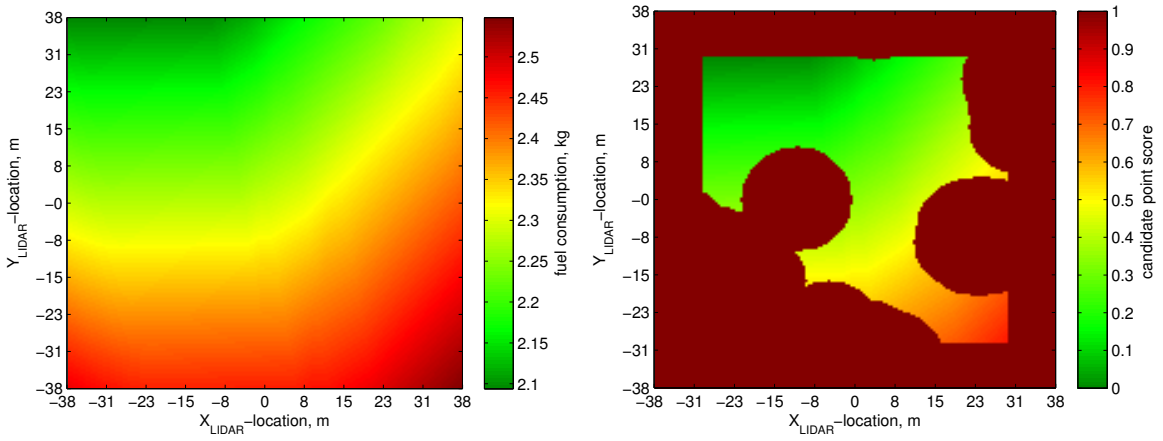
is shown in Fig. 6-17. Note, that also the areas within a distance of one VDFE diameter is inaccessible, because there is no information available about the hazards just outside the LIDAR image.

At this point, the hazard avoidance system checks whether the original target location (here: $x_{LIDAR} = y_{LIDAR} = 0$ m) is a candidate point: The target should only be changed when it is critical to the mission success, because the new target not the desired touchdown position, and retargeting can introduce new errors. In case the original target is not reachable, a new target must be selected from the available candidate points. This choice will be based on the amount of fuel required to reach a given target, because this system has a higher degree of autonomy. It is also possible to chose the candidate point closest the to original target, or apply a variety of different filters to find the optimal retargeting option, but this is beyond the scope of this report.

The relation between a position in the LIDAR image frame and a position in the S-frame follows from a combination of Eqs. (6-109) and (6-113) and can be written as

$$\begin{pmatrix} z_S \\ y_S \end{pmatrix} = \begin{pmatrix} z_{t_{nom},S} \\ y_{t_{nom},S} \end{pmatrix} + \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \left(\mathbf{x}_{t,SHM} + \mathbf{T}_{SHM \leftarrow lidar} \mathbf{x}_{lidar} \right). \tag{6-114}$$

It is now possible to determine a target state vector for every position on the LIDAR image by combining the result of Eq. (6-114) with the x_S-value and the target velocity of the true target state vector. The target state and the estimated initial state are then inserted quadratic guidance logic as shown in Section 5-1-2 – the fuel consumption is one of the byproducts of



(a) full LIDAR image **(b)** combined with reachability map

Figure 6-18: Fuel consumption for the LIDAR image

the time-to-go determination process. The fuel consumption for the complete LIDAR hazard map is shown in Fig. 6-18a. The plot is not symmetrical, because the lander initially has a sidewards velocity, see Table 6-14. The fuel consumption is lower for more distant targets, because the deceleration and thus the thrust level is lower. The Enceladus lander piloting function only investigates the fuel consumption of the candidate points; the result is shown in Fig. 6-18b. The fuel consumption is translated into a score with values between 0 and 0.8 to discriminate between the inaccessible areas with a score of 1.

The new target is then chosen to be the area related to the pixel with the lowest candidate point score – for Fig. 6-18b, the new target would be $x_{LIDAR} = y_{LIDAR} = -31$ m, if the original target was unreachable. In case there is not a single candidate point on the LIDAR image, the lander automatically interrupts its descent and selects a target one field-of-view side length further ahead. The entire process is then restarted.

The Enceladus lander hazard avoidance system is relatively basic, and its refinement is an interesting subject of future work.

6-2-4 Simulator Configuration: Hazard Avoidance System Parameters

The hazard avoidance system of the Enceladus lander requires a true SHM written in a comma-separated `*.txt`-file. Each lander model scans a segment of the true SHM depending on the current sub-satellite point and the current position estimation error, and generates a reachability map from this image. In case the original target cannot be reached, each reachable target point is related to the required amount of fuel. The candidate point with the lowest fuel consumption score is then selected as new target.

All hazard avoidance system related simulator parameters are listed in Table 6-15.

Variable Name	Comment
`isHazardAvoidanceActive`	on/off
`scanAltitude`	[m], 290 m to see 0.5 m hazards with $\Delta\theta = 0.1^\circ$
`resolutionOfTrueSHM`	[m], set to 0.1
`VDFE`	[m], 9 m incl. 0.5 m contingency
`reachabilityMapHazardThreshold`	[-], set to 0.9
`originOfSHMinMapcoordinateX`	[-], relates the `MATBLAB` map to the SHM
`originOfSHMinMapcoordinateY`	[-], see above
`nominalTargetinSframeZ`	[-], $z_{t_{nom}}$ in Eq. (6-109)
`nominalTargetinSframeY`	[-], $y_{t_{nom}}$ in Eq. (6-109)

Table 6-15: Control system related program parameters

Chapter 7

Simulator Setup

The following sections describe the major elements of the software package designed to evaluate the behavior of a lander during descent, touchdown and repositioning.

The program design turned out to be far more complex than initially assumed due to the large number of parameters and interaction of virtually every single system. The program must be able to generate a continuous data flow, and thus connect all isolated system elements discussed in the previous sections. The result, however, is a comprehensive simulation software, which is able to simulate a complete GNC system of a spacecraft in different situations.

The discussion in this chapter will be limited to basic program functionality, the actual `C++` code is not discussed here.

7-1 Program Structure

The Enceladus Lander Simulator must be able to simulate a large number of lander elements with different initial states and configurations in order to determine the mission success rate under realistic circumstances. Ideally, the program structure is modular to allow future additions and to simplify the program maintenance and testing.

The program overview is shown in Fig. 7-1. The simulator has three main elements, or namespaces in `C++`: the flight software, the flight hardware and the flight dynamics. The flight software represents the lander's on-board computer system, which is connected to the flight hardware – the main engine, the attitude verniers and the sensors for internal and external measurements. The flight dynamics represents the true environment of the spacecraft; it determines the lander's motion under the influence of various forces. The true state outputs follow from the state propagator inside the flight dynamics section, while the lander software only works with the estimated states.

All elements in Fig. 7-1 are briefly discussed in Section 7-2. The four orange blocks are the user interfaces – the lander and simulator configuration files as well as simulator outputs are discussed in Section 7-7.

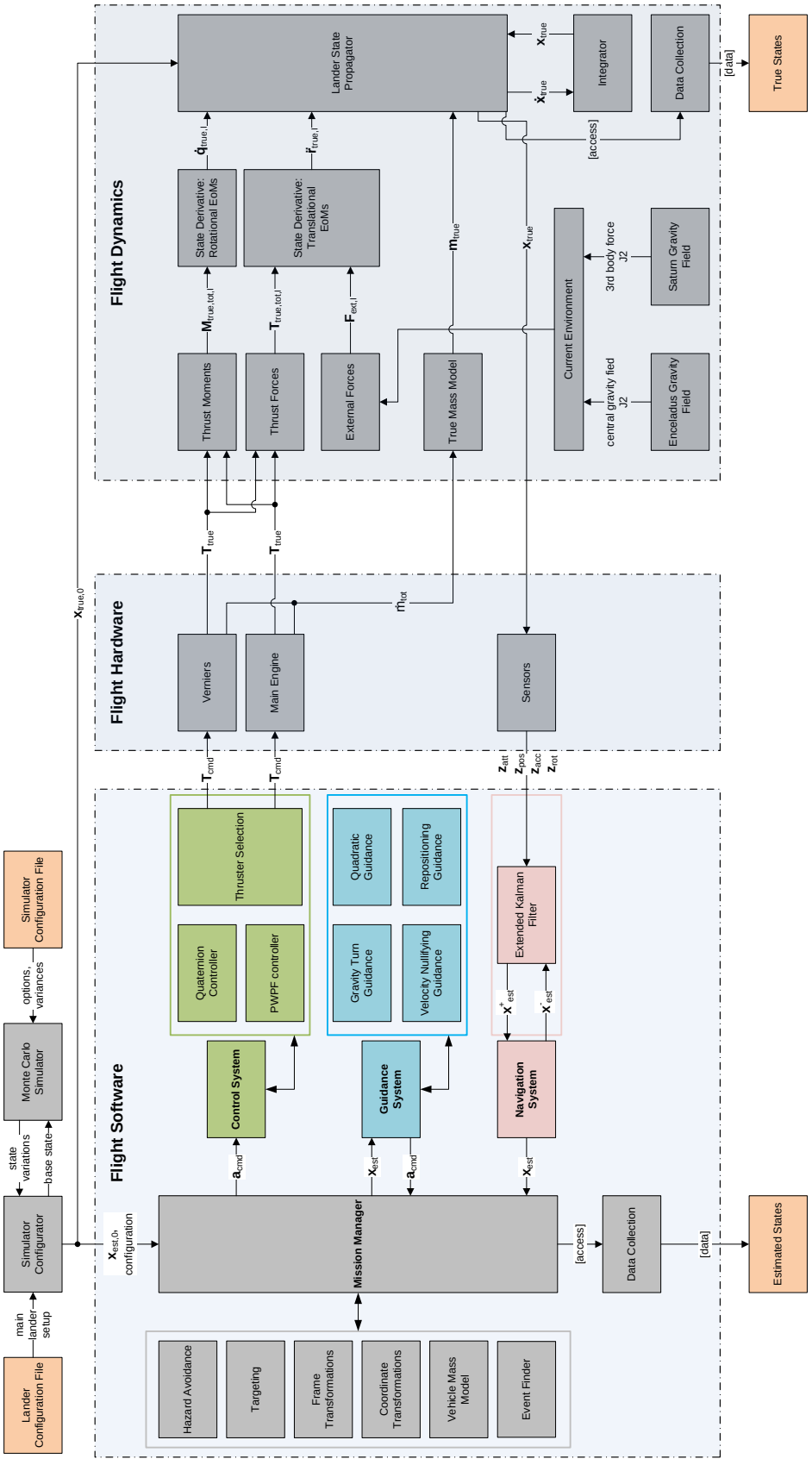


Figure 7-1: Enceladus Lander Simulator program overview

7-2 Program Elements and Testing

Every code module of the Enceladus lander simulator is tested for its correct functionality. Most test routines use the unit test framework of the `boost C++` test library, which allows performing tests in an environment separate from the main function. It offers simple tools to compare data types, to execute successively multiple test cases and to collect pass/fail information.

Some classes, in particular the integrators and spherical harmonics gravity field model, are based on functions inside the `Tudat` toolbox and tested independently. `Tudat` is the TU Delft Astrodynamics Toolbox written in `C++`, and a project of the astrodynamics and satellite mission research group of the faculty of aerospace engineering.

The unit tests are designed such that all interfaces of a code block are tested. Some code modules are tested in conjunction with other modules, as they require data from each other. For example, the `thirdBodyForceModel` needs pointers to an existing `thirdBodyModel` object, a `massModel` object and a `referenceFrames` model. In cases like this, the lower-class modules are tested beforehand, and the higher-class modules often work with input objects in their default state.

The unit tests check whether the code block returns the expected data for the given input. This expected data can come from an external source, such as data bases or text books, or from own calculations. Furthermore, the unit tests check if the module is stable for all inputs, in particular, in the boundary regions or in the default state.

The following lists describes the functionality of the Enceladus Lander Simulator program elements together with the used test method. Most of these elements are shown in Fig. 7-1.

Enceladus Lander namespace

`coordinateSystemTransformations`
: Includes functions to translate between degree and radians, between Euler rotation angles and quaternions, and between Cartesian and spherical coordinates.
Unit Tests: basic axis rotations and back-and-forth translations.

`output`
: Contains several functions to write matrices and vectors to `*.txt`-files and print the contents of data maps onscreen.
Unit Tests: inspection of generated files.

`referenceFrames`
: Collects the attitude data of the current lander model, and can return the orientation of the I-, R-, V-, S- and B-frames with respect to each other, all in terms of both quaternions and rotation matrices. Updated constantly by the state propagator.
Unit Tests: axis rotations and back-and-forth translations.

`simulatorConfiguration`
: Reads the lander and simulator configuration files, initiates all systems and passes pointers and setup data. By far the largest file.
Unit Tests: checking of correct initiation of each class.

Flight Dynamics namespace

`thrustForceModel, thrustMomentModel`
Base classes to determine forces and moments in the I-frame.
Unit Tests: force model in `Tudat`, moment model by recalculation.

`constantPerturbingForce, thirdBodyForceModel, j2GravityField`
Extensions of the force/moment models, with different setup but inherited functions to determine the forces/moments in the I-frame.
Unit Tests: gravity fields in `Tudat`, constant force by recalculation.

`equationsOfRotational/-TranslationalMotion, massDerivative`
Determines the translational and rotational state derivatives and the current mass flow, given the force and moment models and the current state estimations.
Unit Tests: equations of translational motion in `Tudat`, equation of rotational motion using results from literature (quaternion derivative) and `Simulink`, mass derivative by recalculation.

`stateDerivativeCombined`
Collects the results from the equations of motion and combines them with the mass flow to generate the complete state derivative vector.
Unit Tests: inspection of end result.

`massModel`
Comprises the current true and estimated values for the lander total mass, inertia tensor and center of mass shift. Constantly updated by the propagator and control system.
Unit Tests: inspection of saved data.

`stateIntegrator, statePropagator`
Propagates the true state using the chosen RK or RKF integration techniques, and checks whether the simulation end conditions are reached.
Unit Tests: integrators in `Tudat`, propagator as part of flight dynamics system test with `Simulink`.

Flight Hardware namespace

`thrustModel`
Base class for all thrusters; includes engine performance and alignment data, and applies thruster magnitude and alignment errors.
Unit Tests: inspection of returned values.

`attitudeThrusterModel, mainEngineThrusterModel`
Extend the thrust model and add additional values (minimum impulse bit).
Unit Tests: inspection of returned values.

Flight Software namespace

`missionManager`
Central class for the lander models. Calls the guidance, navigation and control system, organizes the hazard avoidance and targeting cycles, checks for landing failures, changes between guidance modes, initiates the true state propagation and saves the system data for the simulation output.
Unit Tests: check for correct function calls by inspection of output.

`eventFinder`
Determines which GNC system or other software element must be called after the next time step, based on the system frequencies. Constantly called by the mission manager.
Unit Tests: inspection of results for various input frequencies.

`targeting`
Transforms target state between different frames, and contains the hazard avoidance routines such as scanning the true SHM, generating reachability and fuel maps, and selecting a new target.
Unit Tests: comparison of results with `referenceFrames`, hazard avoidance: optical inspection of maps.

`guidanceSystem, controlSystem, navigationSystem`
Main system files to coordinate the different subsystems; interface with the mission manager.
Unit Tests: check for correct function calls by inspection of output.

`guidance: gravityTurnGuidanceLogic`
Returns the acceleration command for a spherical and flat moon gravity turn.
Unit Tests: Comparison with literature (see Section 5-1-1).

`guidance: quadraticGuidanceLogic`
Returns the acceleration command for the quadratic guidance, given a target and a time-to-go search option.
Unit Tests: Integration of acceleration command and comparison with target.

`guidance: velocityNullifyingGuidance`
Returns the acceleration command to nullify the current horizontal and – if demanded – vertical velocity.
Unit Tests: Integration of acceleration command and comparison with target.

`guidance: repositioning`
Returns the acceleration commands for the hybrid ballistic-quadratic and the purely quadratic repositioning guidance.
Unit Tests: from quadratic guidance; integration of acceleration command.

`navigation: navigationSystem`
Determines the state estimations given the previous state estimations. The instrument models are not exported to external files for simplicity, so the true state for the measurement generation is also also an required input.
Unit Tests: basic version with literature, end version by inspection.

`control: quaternionController`
Determines the reference moment using a linear quaternion controller, given the current state estimation.
Unit Tests: Integration of result and comparison with target.

`control: pwpfController`
Calculates the thrust pulses required to approximate the moment command from the quaternion controller.
Unit Tests: comparison with literature (see Section 5-2-4)

`control: thrusterManager`
Determines the appropriate thrust levels for the PWPF controller and directs its

output to the correct thrusters.
Unit Tests: Checking the results for different moment commands.

7-3 Monte Carlo Simulation

The Monte Carlo simulation is a technique in the field of probability theory and uses a large number of random experiments to analyze complex numerical problems. In this report, the Monte Carlo technique is used to determine the effects of varying initial conditions on the spacecraft's landing precision. The sensitivity of the system to deviations from the nominal state is a valuable input for the further system development, because it translates into the system fault tolerance. In case a certain variable seems to have a large influence on the end result, the detailed lander design should ensure that its value is always within the given limits.

The Enceladus Lander Simulator uses default values for each base lander model. These nominal values are saved in the lander configuration files, see Appendix A-1. The other lander sample models are the result of modifications of the base lander parameters, initiated by the Monte Carlo simulator. The deviation from the nominal value is determined using a *Gaussian* – or *normal* – distribution. The probability density function of the normal distribution for the random variable x is a bell-shaped curve with the Eq. (Teunissen et al., 2006)

$$f(x) = \frac{1}{\sigma_x\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\left(\frac{x-\mu}{\sigma_x}\right)^2\right\} \tag{7-1}$$

where μ is the *mean* and σ_x the *standard deviation*. μ and σ_x fully define the normal distribution, so Eq. (7-1) is often written as $x \sim \mathcal{N}(\mu, \sigma_x^2)$. In this notation, σ_x^2 is the *variance*. Table 7-1 lists all variances implemented in the Enceladus Lander simulator. The symbols shown here will be used later in this report to present the simulator setup for each simulation run. Note, that each variance is related to a certain subsystem: the velocity estimation variance, for example, has no influence on guidance (G), or guidance and control (GC) simulations – it is only relevant for the full guidance, navigation and control (GNC) simulations. The variance values of the initial conditions are listed in the simulator configuration file, see Appendix A-2, and may be set to any desired value.

For example, the mass of a sample lander model follows from the simulation parameters listed in Table 8-3 and is

$$m = \mathcal{N}(m_0, \sigma_m^2) = \mathcal{N}(335 \text{ kg}, 10 \text{ kg}^2).$$

Due to the probability density function shown in Eq. (7-1), the mass deviation of the lander model from the nominal value is $\sigma_m = 3.16$ kg or less in 68.3% of all cases, and in 99.7% of all cases, the deviation is $2\sigma_m = 6.32$ kg at most.

Variable Name	Symbol	Applicable Simulation Type
`massVariance`	σ_m^2	GNC, GC, G
`massMomentInertiaVariance`[1]	σ_{moi}^2	GNC, GC
`COMshiftVariance`[1]	σ_{com}^2	GNC, GC
`thrustMagnitudeVariance`[2]	σ_{tm}^2	GNC, GC
`thrustAlignmentVariance`[2]	σ_{ta}^2	GNC, GC
`positionVariance`	σ_r^2	GNC, GC, G
`velocityVariance`	σ_v^2	GNC, GC, G
`eulerAngleVariance`	σ_q^2	GNC, GC
`positionEstimationVariance`	$\sigma_{r,e}^2$	GNC
`velocityEstimationVariance`	$\sigma_{v,e}^2$	GNC
`eulerAngleEstimationVariance`	$\sigma_{q,e}^2$	GNC
`hopDistanceVariance`	σ_y^2	GNC, GC, G (repos. only)
`hopAltitudeReqVariance`	σ_h^2	GNC, GC, G (repos. only)

[1] see explanation at the end of this section
[2] for main engine and verniers

Table 7-1: Program parameters for the Monte Carlo Simulation

The meaning of the center of mass shift variance σ_{com}^2 and the mass moment of inertia variance σ_{moi}^2 is not directly evident, and will be explained in the following.

As indicated in Section 3-1, the lander's nominal center of mass is located in the geometrical center of the spacecraft body. It is furthermore assumed, that the mass is homogeneously distributed in the shape of a cube, which leads to a diagonal inertia tensor. The Monte Carlo simulations include deviations from the cubic shape – which will change the mass moments of inertia I_{xx}, I_{yy} and I_{zz} – and a shifting center of mass – which will lead to non-zero products of inertia I_{xy}, I_{xz} and I_{yz}. The implementation of a rectangular parallelepiped instead of a cube is straightforward, and is achieved by modifying the mass moments of inertia with the according variance values (see Table 3-5. The center of mass shift is achieved by shifting the mass block inside the spacecraft bus, so effectively, the B-frame is shifted with respect to the A-frame. This will influence the lander rotation behavior, because all thrusters remain fixed in the A-frame and now have different thrust arms. In particular the main engine generates a thrust moment, if the thrust vector does not pass through the center of mass. For the center of mass shift $[\Delta x_{\text{com}}, \Delta y_{\text{com}}, \Delta z_{\text{com}}]$ with respect to the original center of mass, the new inertial tensor then becomes

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} + m \begin{bmatrix} \Delta y_{\text{com}}^2 + \Delta z_{\text{com}}^2 & -\Delta x_{\text{com}} \Delta y_{\text{com}} & -\Delta x_{\text{com}} \Delta z_{\text{com}} \\ -\Delta y_{\text{com}} \Delta x_{\text{com}} & \Delta x_{\text{com}}^2 + \Delta z_{\text{com}}^2 & -\Delta y_{\text{com}} \Delta z_{\text{com}} \\ -\Delta z_{\text{com}} \Delta x_{\text{com}} & -\Delta z_{\text{com}} \Delta y_{\text{com}} & \Delta x_{\text{com}}^2 + \Delta y_{\text{com}}^2 \end{bmatrix} \tag{7-2}$$

where the first term is the moment of inertia about the center of mass, and the second term follows from the parallel-axis theorem. The theory behind Eq. (7-2) is discussed, for example, in Meriam et al. (2002).

The Enceladus Lander Simulator uses the external `Boost C++` library to determine the value for x given the mean value of the base lander model and the according variance for the Monte

Carlo simulations. The random number generator has fixed seed numbers. In this way, the respective random values are equal each simulation run, which allows an easier problem analysis and is necessary for the reproducibility of the results.

7-4 Simulator Configuration and Output

The Enceladus Simulator is fully configured by two separate `*.txt` files.

The complete true lander state for the Enceladus Lander Simulator is defined in the *lander configuration file*. This file also includes all data that define the nominal – or main – lander model characteristics, such as the inertial tensor, the thruster orientation and the thrust limits. The simulation parameters, on the other hand, are listed in the *simulator configuration file*, and range from the maximal simulation time and the propagator selection to the Monte Carlo Simulation parameters to the various GNC options and success conditions.

The implemented file reader automatically parses the a text string that is not started with a number sign. The reader will return an error if the text line does not correspond to the format 'variable name' – '=' – 'number'. Activations and deactivations are thus represented by the values 1 and 0, respectively. This principle is illustrated in the following extract from the simulator configuration file:

```
# CONTROL OPTIONS
# PWPF controller options:
# 0.0 deactivated: moment_B = torque_ref from controller, no thrust limits
# 1.0 activated: filter & trigger used, thrust limitations
proportionalGainQuaternionController = 2.5
derivativeGainQuaternionController = 3.5
PWPFactive = 0.0
```

The proportional and derivative control gains will have the values 2.5 and 3.5, respectively, while the PWPF controller is inactive.

The Enceladus Lander Simulator generates a series of result `*.txt`-files that collect the estimated and true initial and final states, and the position data as a function of time for each lander model. The complete simulation data is only saved for the main lander model, but may be activated for all Monte Carlo elements if desired. The simulation parameters are discussed in Section 7-7. Note, that this requires a lot of disk space, and is only necessary to investigate anomalies. All figures shown in this chapter are generated in `MATLAB` using the output files of the Enceladus Lander Simulator.

7-5 System Test of Flight Dynamics

Many code modules of the simulator's flight dynamics part can be tested with the `MATLAB Simulink` block *Simple Variable Mass 6DoF (Quaternion)*, which is part of the *Aerospace Blockset*. This Simulink block can propagate the translational and rotational state of body with respect to the inertial and the body-fixed reference frame, given the forces and moments acting on the body.

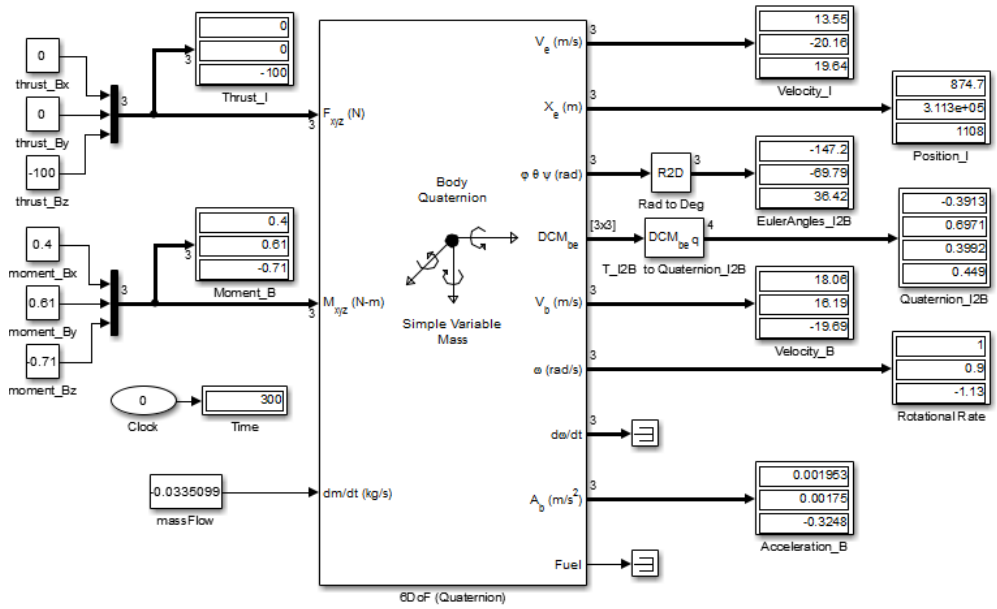


Figure 7-2: Simulink model used for the flight dynamics system test

In the flight dynamics system test, the state of a lander model is propagated for a specific set of initial thruster settings. The result should ideally be the same as a propagation with the same settings in `Simulink`.

Although the basic equations of the `Simulink` model and the Enceladus Lander Simulator are the same, the program structure and thus the method for arriving at the result is different. This inevitably will lead to (small) differences.

Floating-point numbers as used in both simulations have a high precision, but suffer from a limited accuracy. Almost all decimal numbers have no exact binary counterpart, which leads to the introduction of roundoff errors when calculating with floating-point numbers. The magnitude of this error is limited by the so-called *machine epsilon*, which is defined as the smallest increment of the floating-point number 1.0 that can be represented as a new floating-point number. The machine epsilon for the Enceladus Lander Simulator can be determined with the `C++` standard library class *numerical_limits* and is 2.22045×10^{-16} for double-precision floating-point values. An additional source of roundoff errors are the `C++` standard library trigonometric functions used in the reference frame transformations class, because these function are based on polynomial approximations that return (small) non-zero results in situations where they are expected to return zero (Bezanson, 2013). In general, these errors are very small, but they are magnified when divided by small numbers, for example during a numerical integration process. The latter can introduce additional error sources, in particular, a loss of significance. This effect occurs on the one hand, when a very small floating-point number is added to a comparably large floating-point number: During the arithmetic operation, the least significant bits of the smaller number are shifted and subsequently lost (Press et al., 2002). On the other hand, a subtraction of two very close values will result in a large relative error, as the result loses significant bits (Bezanson, 2013).

These effects can be minimized and quantified by a numerical analysis of all code blocks of the Enceladus Lander Simulator, but this is beyond the scope of this thesis. Consequently, it is expected that the results produced with the Simulink model and with the Enceladus Lander Simulator will not perfectly match, but the discrepancies must be within acceptable limits.

The system test consists of a state vector propagation for a lander with at an initial mass of 320 kg, a fixed diagonal inertia matrix with $I_{xx} = I_{yy} = I_{zz} = 150$ kgm^2 and the following translational and rotational state vector elements:

$$\mathbf{x}_{translational} = \begin{bmatrix} 0.0 \\ 312345.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \end{bmatrix}, \quad \mathbf{x}_{rotational} = \begin{bmatrix} 1.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.2 \\ -0.32 \\ 0.29 \end{bmatrix} \tag{7-3}$$

The main engine produces a constant thrust of 100 N in the thruster reference frame, and the verniers generate constant thrust moments of [0.4, 0.61, -0.71] Nm. Both simulators use a Runge-Kutta 4 integrator with a fixed stepsize of 0.1 s over a total simulation time of 300 s. The variable step-size integrator is not suitable for the system test, because the step-size determination logic is different in the `Simulink` model, which will lead to different step-sizes and thus to meaningless discrepancies in the results. Table 7-2 shows the absolute and the relative deviation of the Enceladus Lander Simulator results from the `Simulink` model results; the actual final state after the 5-minute-propagation is given in Fig. 7-2. Differences are listed as approximately zero for values below 1.0×10^{-8}. The attitude quaternions are translated into the corresponding Euler angles for a better insight. The final mass of the body is no output of the basic *6DoF*-block - but this is not even necessary, as the mass flow is kept constant during the integration process, which leads to a linearly decreasing vehicle mass. The mass loss is significant (about 10 kg), so faulty mass integration would quickly lead to large errors in the equations of translational motion.

State Element	Absolute Difference	Relative Difference
X_I	0.990 m	0.113 %
Y_I	2.936 m	0.001 %
Z_I	-1.496 m	-0.135 %
$V_{x,I}$	0.094 m/s	0.701 %
$V_{y,I}$	-0.117 m/s	0.582 %
$V_{z,I}$	0.120 m/s	0.615 %
$\psi_{B/I}$	$-0.307\times10^{-6}\,^\circ$	$\approx$ 0 %
$\theta_{B/I}$	$-0.320\times10^{-6}\,^\circ$	$\approx$ 0 %
$\phi_{B/I}$	$0.155\times10^{-6}\,^\circ$	$\approx$ 0 %
p	$\approx$ 0 rad/s	$\approx$ 0 %
q	$\approx$ 0 rad/s	$\approx$ 0 %
r	$\approx$ 0 rad/s	$\approx$ 0 %

Table 7-2: Absolute and relative deviation of the Enceladus Lander Simulator results from the Simulink model results

As can be seen, the relative and absolute error of the rotational rates and attitude angles is very low, while the errors in velocity and position are higher, reaching a maximal relative error of 0.7% for the V_x-velocity. The increase in error from the table's bottom to top is expected, because errors in rotational rates are passed to the attitude angles, which in turn pass their errors in terms of imperfect transfer matrices to the acceleration vector and thus subsequently the velocity vector. All errors are in the end collected during the position determination. Further testing showed, that the errors increase in particular with increasing rotation velocities. It is therefore decided to design an additional test case focusing only on a rotating thrust vector, and compare the results with precise analytical solutions.

7-6 Rotating thrust vector

The exhaust of the main engine points in the direction of the $+Z_B$-axis, so the T- and the B-frame coincidence at all times (see Section 3-2). When the spacecraft is in a rotational motion, the thrust force vector $\mathbf{T}_B$ always points in the $-Z_B$-direction - in contrast to the continuously changing components of $\mathbf{T}_I$ and the accompanying acceleration $\mathbf{a}_I$. The integration of the equations of translational motion requires precise values for $\mathbf{a}_I$.

Trial runs of the software have shown that an inaccurate definition of $\mathbf{a}_I$ - in particular, the assumption that the thrust vector $\mathbf{T}_I$ is constant during one integration step and thus *not* independent on the current rotational state - will quickly lead to large errors in the translational part of the propagated state vector. Consequently, a separate test case was designed to benchmark the integration of $\mathbf{a}_I$. The test scenario is shown in Fig. 7-3. The spacecraft is rotating about the Z_B-axis with a rotational rate q with an activated main engine, resulting in an acceleration $\mathbf{a}_B$. The X_I- and the Y_B-axis are aligned, so the scenario is viewed in the $X_I Z_I$-plane only.

Assuming that no moments are acting on the spacecraft and that the rotational rates are zero for the other axes, the inertial velocity of the lander at time t then follows from integrating the X_I- and Z_I-components of $\mathbf{a}_T$ and applying the boundary condition $\mathbf{v}_{y,I}(0) = \mathbf{v}_{z,I}(0) = 0$ ms^{-1}:

$$\mathbf{v}_I = \frac{T}{m}\frac{1}{q}\begin{bmatrix} 0 \\ \cos(qt) - 1 \\ \sin(qt) \end{bmatrix} \tag{7-4}$$

Similarly, the inertial position $\mathbf{d}_I$ should be

$$\mathbf{d}_I = \frac{T}{m}\frac{1}{q^2}\begin{bmatrix} 0 \\ \sin(qt) - tq \\ 1 - \cos(qt) \end{bmatrix} \tag{7-5}$$

In the test scenario, the spacecraft is rotating with a relatively high angular velocity of 0.2π rads^{-1} in combination with a thrust setting of 100 N. Figure 7-4 shows the deviations of the simulator and the Simulink results from the exact analytical solutions (equations (7-4) and (7-5)) over a period of 10 minutes, using again the RK4 integrator with a step-size of 0.1 s. The velocity errors along the Z_I-axis are in the order of 10^{-5} and grow over time, while the errors along the Y_I-axis are very small and periodic. The resulting accumulated position error

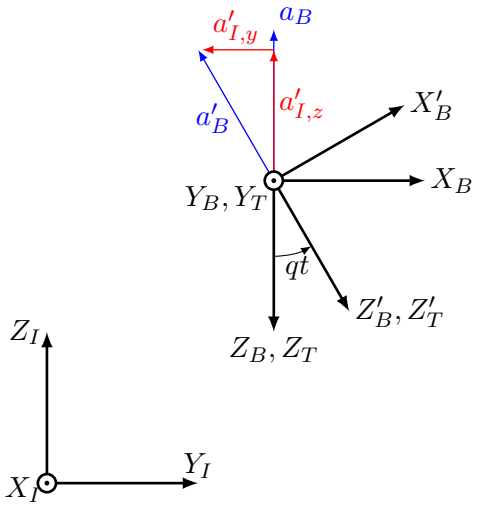
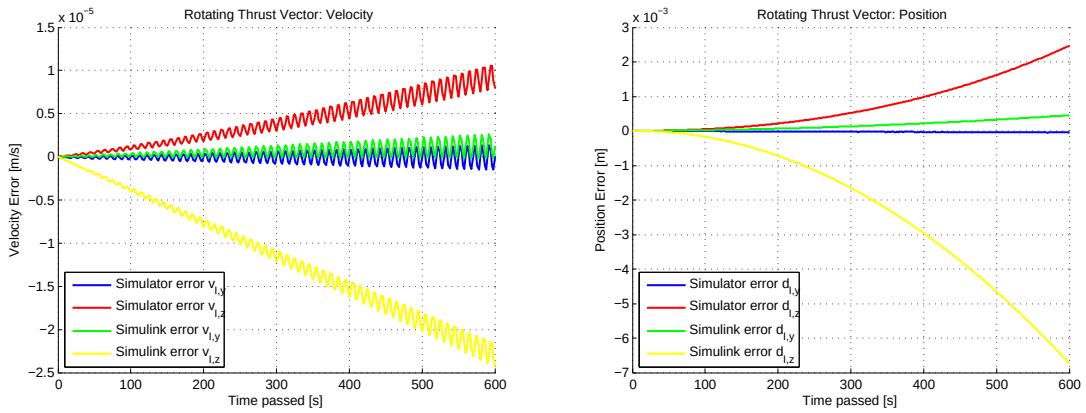


Figure 7-3: The acceleration of a lander model rotating about the Y_B-axis, with respect to the inertial and the body reference frame I and B



(a) Velocity error **(b)** Position error

Figure 7-4: Analytical and simulated results for a rotating body with a body-fixed thrust vector

for the Enceladus Lander Simulator is about 2 mm and 0.5 mm, respectively. Considering the high angular velocity of the spacecraft over a long period of time and the comparable results from the Simulink model, the state propagator seems to work as intended. The accuracy of the results can be increased by selecting a higher-order integrator or reducing the step-size.

7-7 Simulator Configuration: Simulation Parameters

The Enceladus lander simulator incorporates five different simulation modes (see Table 7-3; three for the descent and two for the repositioning process. The mode selection activates or deactivates the different subsystems and prepares the guidance mode selection (see Table 5-5. A deactivated navigation systems sets all estimated states equal to the true states, because the flight software has no means to propagate the state estimations. The deactivation of the

control system limits the lander dynamics to three degrees of freedom; the lander becomes a point mass, and the desired thrust vector based on the commanded acceleration is aligned perfectly.

Mode Number	Description	Comment
1	Descent: G simulation	see Section 8-3-1
2	Descent: GC simulation	see Section 8-3-2
3	Descent: GNC simulation	see Section 8-3-3
4	Repositioning: GC simulation	not used
5	Repositioning: GNC simulation	see Section 8-4

Table 7-3: Simulation mode numbers

The parameters to define the simulation output and the boundary conditions that stop the current simulation run, are listed in Table 7-4. The initial and final states for each lander element together with the according parameter variations are always saved. The full simulation data, such the true position in different reference frames, as a function of time is only saved if requested. It is possible to simulate single lander elements, which the fastest way to analyze landing failures or unexpected behavior of single lander elements. The state results mentioned in Table 7-4 refers to a single file collecting the position data of all lander elements. This data package is necessary to plot the trajectories of all lander elements in a single figure, for example in Fig. 8-2. The state results data package is a large file, and its generation can increase the simulation time considerably.

Variable Name	Comment
General	
`simulationMode`	option 1 to 5, see Table 7-3
Output Options	
`saveResultsBaseLander`	on/off save all data
`saveResultsAllLander`	on/off save all data
`saveStateResultsAllLander`	on/off generate position data
`saveResultsLanderNumber`	save all data for specific element
`onlySimulateLanderNumber`	simulate specific element only
`saveResultsStepSize`	[s], step size for all data saves
`excludeFailLandings`	on/off include fails in state
Boundary Conditions	
`simulationStartEpoch`	[s]
`simulationEndEpoch`	[s], $t > t_{max}$ means failure
`maxVelocityErrorForFail`	[m/s], higher Δv_{end} means failure
`maxPositionErrorForFail`	[m], higher Δr_{end} means failure
`minimalAltitude`	[m], $h < h_{min}$ means failure

Table 7-4: Simulation parameters, all in the simulator configuration file

Chapter 8

Simulation Results

This chapter presents the simulation results of Enceladus Lander Simulator. The theory behind the program and the software setup were discussed in the previous chapters.

The simulations are performed separately for the descent phase and the repositioning phase, and not all elements of the GNC system are activated at the same time. The simulation plan presented in Section 8-1 gives more detail about the exact simulation sequences.

The Enceladus Lander Simulator uses different reference frames for the data input and output: the initial state must be specified with respect to the I-frame, the target state with respect to the R-frame, and the simulation results are mainly expressed with respect to the S-frame. The initial and target states are derived in Section 8-2, including a discussion of the general simulator setup.

Section 8-3 discusses the simulation results for the descent phase, with a partially and fully activated GNC system to analyze the effects of the single subsystems on the landing precision. Section 8-4 deals with the repositioning of the lander on the surface, with a fully activated GNC system and a large number of different targets.

8-1 Simulation Plan

The main target of the Enceladus Lander Simulator is finding the answer to the question whether a hopping planetary lander can safely land and reposition on the surface of Enceladus. There are two main characteristics to track: on the one hand, the landing precision under the influence of disturbing forces, because it is critical for a successful touchdown at the end of the descent phase and of each repositioning cycle. On the other hand the fuel consumption, as it has the largest influence on the lander's surface coverage, and is the decisive factor when comparing the different guidance options.

The guidance, navigation and control systems are activated sequentially to investigate the effects on the end results. Each system adds new, system-related Monte Carlo variances to simulations. The random number generators inside the Enceladus Lander Simulator use fixed

seed numbers, so parameter variations are equal each simulation run – this is a prerequisite to allow for a comparisons between multiple simulation runs.

Table 8-1 below gives an overview of the simulation runs that will discussed in the following section. The letters G, GC and GNC refer to guidance, guidance and control, and guidance, navigation and control simulation runs, respectively. The guidance system is active in all cases. It is decided against a separate GN run due to time constraints. The choice fell on GC instead of GN, simply because the control system was implemented in the `C++` code before the navigation system.

Simulation Mode	Objective	Reference
Orbital Descent		Section 8-3
G	landing precision, fuel consumption	Section 8-3-2
GC	landing precision, error tolerance	Section 8-3-2
GNC	landing precision, error tolerance	Section 8-3-3
Repositioning		Section 8-4
GNC	landing precision, fuel consumption	–

Table 8-1: Simulation plan for the Enceladus Lander Simulator

8-2 Initial conditions and Simulation Setup

The Tiger Stripes cover a large part of Enceladus' southern hemisphere. The target area for the GNC simulations is chosen to be the depression with the name *Baghdad* (see Fig. 2-5) along the 70° south longitude and between the 10° and 40° latitude. The rotating reference frame in the Enceladus Lander Simulator is not directly connected to the a base-map such as shown in Fig. 2-5, so the actual target latitude depends on the initial orientation of these two reference frames. The target state must be expressed with respect to the rotating frame. Thus, for a simpler translation to Cartesian coordinates, the desired target latitude is set to 0°. The target position $\mathbf{r}_{t_R}$ with $\|\mathbf{r}_{t_R}\| = R_{\text{enc}}$ thus is a vector that initially pointed in the X_R-direction and then rotated 70° about the Y_R-axis, see Fig. 4-2 and 4-7. This leads to a target position of $[86223.28,\ 0,\ -236896.51]$ m.

The target velocity with respect to the surface-fixed frame is set to $[-0.5,\ 0,\ 0]\ \frac{\text{m}}{\text{s}}$, as horizontal velocities should always be avoided and a vertical velocity of $0.5\ \frac{\text{m}}{\text{s}}$ is below the maximum velocity that the landing gear can absorb (see Section 3-3). Using the same transformation as for the position vector, this leads to a target velocity $\mathbf{v}_{t_R}$ at $\mathbf{r}_{t_R}$ of $[-0.17,\ 0.02,\ 0.47]\ \frac{\text{m}}{\text{s}}$.

The circular velocity at the initial altitude h_0 can be calculated with

$$V_c = \sqrt{\frac{\mu_{enc}}{R_{enc} + h_0}} \tag{8-1}$$

and is $168.11\ \frac{\text{m}}{\text{s}}$ at $h_0 = 3$ km. The descent phase is initiated at the point where a pure gravity turn will guide the lander close to the actual target point. This approach is necessary to allow for a (fair) comparison between the implicit gravity turn and the explicit quadratic guidance logic. In contrast to the target state, the lander's initial state in the Enceladus Lander Simulator must be expressed with respect to the inertial planetocentric reference

frame. The approximate gravity turn downrange for an assumed initial velocity of 0.4 V_c or 67.25 $\frac{\text{m}}{\text{s}}$ follows from Eq. (5-25) and is 10.63 km. The initial position then is the location in a 70° inclination orbit, 10.63 km in front of the target measured along the surface. The orbit is in prograde motion with respect to the rotating moon to reduce the required ΔV. As a vector rotation follows from a reference frame rotation in the opposite direction, the initial inertial lander position is

$$\mathbf{r}_{0_I} = \mathbf{T_2}\left(-70^\circ\right)\mathbf{T_3}\left(2\pi\frac{10,630\text{ m}}{2\pi R_{\text{enc}}}\right)\begin{pmatrix} r_{\text{enc}}+h_0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 87180.71 \\ -10116.35 \\ -239527.02 \end{pmatrix}\text{ m}$$

where the expressions for $\mathbf{T_2}$ and $\mathbf{T_3}$ are given in Eqs. (4-12b) and (4-12c). This can be translated to the spherical coordinates $\tau = -6.62^\circ$, $\delta = -69.88^\circ$ and $r = 255.1$ km.

The surface velocity with respect to a motionless spacecraft at the 69.88° south longitude is

$$V_{\text{surf}} = \omega_{\text{enc}} R_{\text{enc}} \cos\left(69.88^\circ\right) = 4.60\text{ ms}^{-1}$$

which leads to an initial velocity with respect to the I-frame at the previously determined initial position $\mathbf{r}_{0_I}$ of

$$\mathbf{v}_{0_I} = \mathbf{T_2}\left(-70^\circ\right)\mathbf{T_3}\left(2\pi\frac{10,630\text{ m}}{2\pi R_{\text{enc}}}\right)\begin{pmatrix} 0 \\ 67.25 \\ 0 \end{pmatrix}\text{m/s} - \begin{pmatrix} 0 \\ 4.60 \\ 0 \end{pmatrix}\text{m/s} = \begin{pmatrix} 0.91 \\ 62.60 \\ -2.51 \end{pmatrix}\text{m/s}.$$

The above expression for $\mathbf{v}_{0_I}$ does not include the Coriolis effect – the resulting small deviations will be corrected by the quadratic guidance logic.

The lander's initial attitude is chosen such that the X_B-axis points in the direction of velocity vector $\mathbf{v}_{0_I}$. In this way, the cross-range (z_S-value) between the initial and the final position is zero, which allows an easier interpretation of the results. The rotation matrix from the I-frame to the B-frame at $\mathbf{r}_{0_I}$ located at the 69.88° south longitude is

$$\mathbf{T}_{B\leftarrow I} = \mathbf{T_1}\left(69.88^\circ\right)\mathbf{T_1}\left(-90^\circ\right)\mathbf{T_3}\left(90^\circ\right)$$

which can be translated to the Euler angles $\phi = -20.12^\circ$, $\theta = 0^\circ$ and 90° or the attitude quaternion $[0.6962, -0.1235, -0.1235, 0.6962]$ (see Section 4-2-4 for the appropriate transformations). The S-frame is defined using a series of transformations based on the initial position, velocity and attitude in the I-frame (see Section 4-3-2). The initial and final lander state vectors with respect to the S-frame together with the other state vectors as derived up to now are collected in Table 8-2. Note, that the I- and R-frame are used for the simulation setup, while the S-frame is the primary tool for the evaluation of the simulation results.

Name	Value
Initial State	
position in I-frame	$[87180.71,\ -10116.35,\ -239527.02]\,\mathrm{m}$
velocity in I-frame	$[0.91,\ 62.60,\ -2.51]\,\mathrm{m/s}$
position in S-frame	$[3000.00,\ 0.00,\ -0.00]\,\mathrm{m}$
velocity in S-frame	$[0.19,\ 58.03,\ -0.51]\,\mathrm{m/s}$
attitude $B \leftarrow I$ in	
– Euler angles	$(-20.12^\circ, 0^\circ, 90^\circ)$
– quaternions	$[0.6962, -0.1235, -0.1235, 0.6962]$
Target State	
– position in R-frame	$[86223.28,\ 0,\ -236896.51]\,\mathrm{m}$
– velocity in R-frame	$[-0.17,\ 0.02,\ 0.47]\,\mathrm{m/s}$
– position in S-frame	$[-198.25,\ 9997.34,\ 0.00]\,\mathrm{m}$
– velocity in S-frame	$[-0.50,\ 0.00,\ 0.00]\,\mathrm{m/s}$
attitude $B \leftarrow I$	(Z_B-axis pointing nadir)

Table 8-2: Nominal initial and target conditions for the simulations in the next sections.

The Enceladus Lander Simulator considers a landing as a success, if the total velocity at touchdown in the S-frame is below 2 ms^{-1}, and the horizontal distance to the actual target spot is 10 m at most. The velocity end condition is relaxed by 0.6 ms^{-1} from the 1.4 ms^{-1} as mentioned in Chapter 3-3. The landing gear design is conceptual only, and the simulations indicated that the end velocity is the most critical parameter for the given system design. While the current basic velocity nullifying guidance is working as intended, there is still room for improvement, and it would be wrong to count a lander element for lost, if its end velocity slightly exceeds the original boundary. Note that only a very low number of landers in fact ends with a velocity between 2 ms^{-1} and 1.4 ms^{-1}, see for example Fig. 8-13a. The maximal position error of 10 m as discussed in Section 5-1 remains unchanged. Furthermore, any simulation run is stopped and counted as failure, when the lander model rotates faster than 1 rads^{-1} or does not hit the ground within the maximal simulation time,

Most of the lander and simulator configuration parameters do not change from one simulation run to another, so for better structuring, each of the following sections will specify the deviations from the basic configuration, and list the most important parameters for the current simulation series. The contents of the basic lander and simulator configuration files are shown in appendix A-1 and A-2, respectively. The exact meaning and nominal value of each parameter has been defined in the corresponding chapters.

8-3 Orbital Descent

This section discusses the simulation results for the orbital descent for three different GNC modes: Only guidance is active (subsection 8-3-1), guidance and control is active (subsection 8-3-2), and guidance, control and navigation is active (subsection 8-3-3). The GNC modes in the Enceladus Lander Simulator are discussed in Section 7-7.

8-3-1 Guidance System Simulations

The guidance system simulations evaluate the landing precision for the spacecraft in case the control navigation system work perfectly, while the guidance system uses all available guidance logics in combination with varying initial conditions and settings. In this configuration, the spacecraft model has only three degrees of freedom, because the rotational motion is steered by the (ideal) control system. The simulator software uses a separate main engine model which is not related to the lander's attitude and is always aligned perfectly with the commanded acceleration vector.

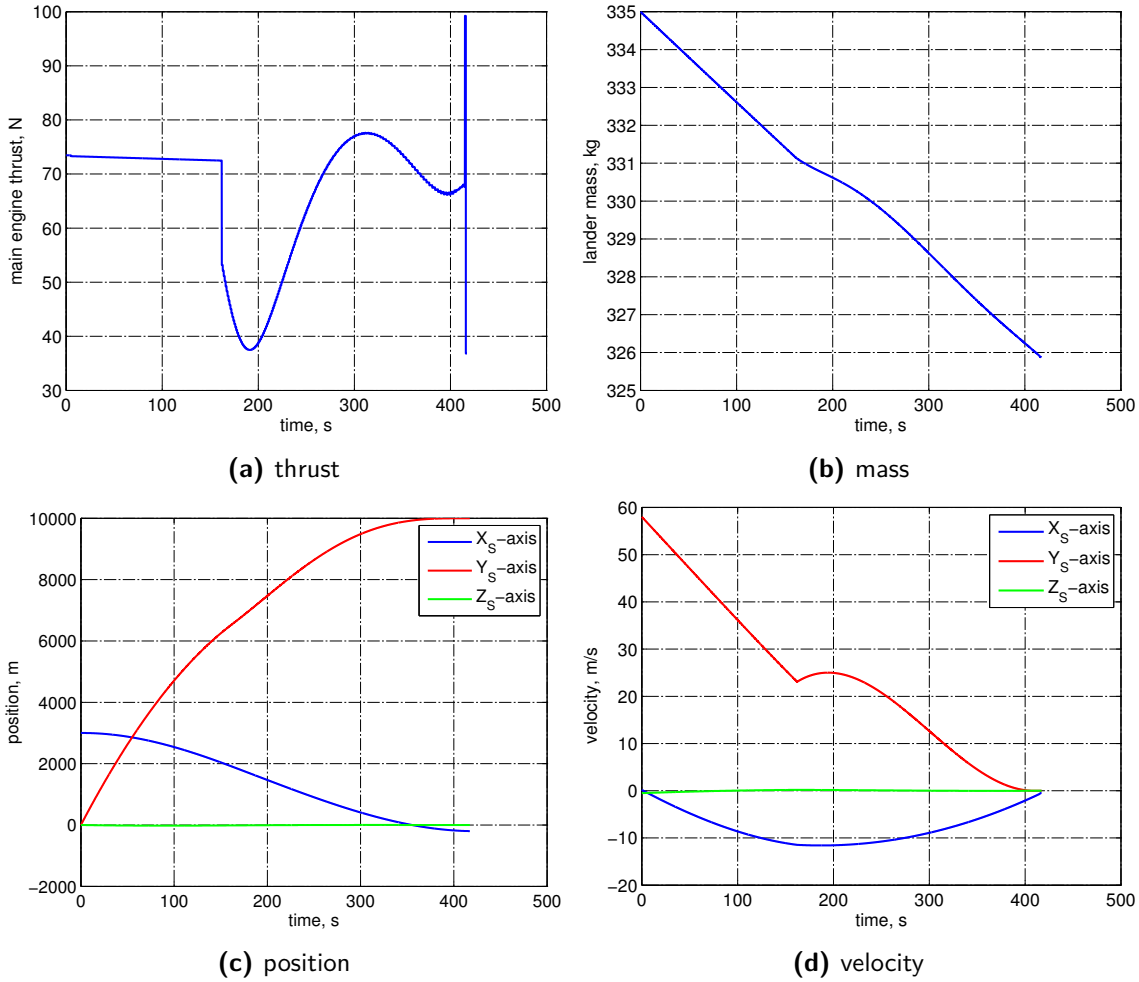
Name	Value	Comment
integrator	RK 4	with $\Delta t = 0.5$ s
lander number	1000	–
true position $1\sigma_r$	200 m	–
true velocity $1\sigma_v$	4.5 ms^{-1}	–
mass variance $1\sigma_m$	3.16 kg	$\sigma_m^2 = 10$ kg^2
altitude switch $1\sigma_{as}$	200 m	mean is 2000 m
guidance frequency	1 Hz	higher freq. unnecessary
force models	– gravity field Enceladus	J_2 and central
	– gravity field Saturn	J_2 and central
other options	– rotating moon	–
	– variable lander mass	–

Table 8-3: Important simulation parameters

Ideally, the equations for the quadratic guidance discussed in Section 5-1-2 should ensure that the lander's final position and velocity match the target state given in Table 8-2, and negate all deviations caused by the implicit gravity turn guidance - for all acceptable initial lander states. The quadratic guidance is activated at the *switch altitude* of 2000 m in the nominal case, deactivated for the horizontal plane 10 s before touchdown, and completely disabled 2 s before touchdown in order to avoid exploding terms for time-to-go values close to 0 s. The velocity nullifying guidance, which is active from 10 s before touchdown until the end, is not designed to reduce position errors, so it is expected that the lander will have always have a (small) landing error.

The initial conditions for the nominal lander modal are listed in Table 8-2 and in full detail in the configuration files in A-1, while the simulator configuration parameters together with the Monte Carlo simulation options are given in simulator configuration file in A-2. The exact meaning of all parameters together with their influence on the lander's behavior is discussed in the respective chapters of this report. Table 8-3 gives an overview of the most important simulation parameters for the guidance system simulations. In particular, these are the number of simulated lander models, the variance values for position, velocity, mass and switch altitude, and the basic integrator setup. The variance values for the inactive control and navigation system are set to zero for this simulation run, as indicated in the simulation plan.

The S-frame for all lander models is initiated using the initial state of the *nominal* lander model, because the initial and final states of the Monte Carlo lander element can then be



(a) thrust **(b)** mass

(c) position **(d)** velocity

Figure 8-1: Nominal mission parameters: descent G simulations

expressed as a deviation from the nominal lander state. In particular, the target location in the S-frame is identical and thus comparable only for this case.

The plots in Fig. 8-1 give an overview of the most important flight parameters for the nominal lander mission. In this section, the spacecraft is modeled as a point mass with perfect thrust, so the nominal mission parameters shown here are limited to the thrust, mass, position and velocity as functions of time. The gravity turn guidance logic returns a constant acceleration command, so the initial main engine thrust shown in Fig. 8-1a is approximately 72 N during the first guidance phase, and decreases only slightly due to the decreasing lander mass (see Fig. 8-1b). At an altitude of 2000 m – represented by the blue line in Fig. 8-1c – the quadratic guidance is activated, and the thrust level drops to about 55 N. The thrust then follows a S-shaped curve, which is the result of the quadratic guidance solution between the current and the target state. The horizontal velocity is about twice as high as the vertical velocity at the start of the quadratic guidance phase – the linear thrust profile for the X_S-direction is overlaid by the quadratic profile for the Y_S-direction. The velocity nullifying guidance is activated 10 s before touchdown, and returns relatively large thrust values up to 100 N, which is a consequence of the shot update period – the guidance system tries to eliminate all

velocities over a period of 1 s. The target position and velocities are all reached; the final values are close to the target state defined in Table 8-2.

Figure 8-2 shows the descent trajectory for the nominal lander model (cyan) and for the 1000 other lander models with diverting initial states from the Monte Carlo simulation (blue). The guidance system is able to handle all initial states and bring each lander close to the specified target position. The maximal initial position, velocity and mass divergence is about 720 m, 17 ms^{-1} and 9 kg. In particular the velocity divergence has a large influence on the interim cross-range and downrange deviations from the nominal trajectory. These deviations can add up to values in the order of kilometers, but they have no influence on the actual landing precision. The individual Monte Carlo element trajectories are removed from Fig. 8-3 and replaced by gray envelopes for a better overview. The shape for the $X_S Z_S$- and $X_S Y_S$-plane is mainly defined by singular outliers.

The landing precision on the horizontal $Y_S Z_S$-frame in terms of distance to the actual target point is illustrated in Fig. 8-4. The simulation results cluster around two landing spots and display a maximal landing error in the order of 6 cm. This is a direct consequence of the simulation parameter for the smallest time-to-go for which the C-factors may still be updated (`minimalT2GoForCalculationCfactors`, see Table 5-6, here: 2 s) and the comparatively low guidance frequency of 1 Hz: The simulator propagates the state vectors with an overall integration step size based on the highest GNC frequency, thus 1 s for a 1 Hz guidance frequency. The propagation is stopped at the last valid state, directly before touchdown. In the worst-case scenario, the end condition is not exactly reached two seconds after the last C-factor update, and the quadratic guidance command is based on state vector estimations older than 2 s. For all other cases, the last C-factor update was 2 s ago. In combination with the effects of the rotating reference frame and the gravity fields of Saturn and Enceladus itself, this leads to two distinct landing spots. The effect can be reduced by reducing the minimal time-to-go requirement for C-factor updates to 1 second and eliminated by selecting a higher guidance frequency in addition.

On a related note, the time-to-go should not be updated too often with the search scheme discussed at the end of Section 5-1-2, because this sometimes leads to time-to-go jumps and, accordingly, strong thrust fluctuations. The base time-to-go in this section is only determined at the very beginning of the simulation: the second update possibility occurs at the end of the hazard avoidance process, but the original target point is accessible for all lander models, and the target is thus not changed. The time-to-go is in this case simply updated by subtracting the current system time.

The end velocity errors for the horizontal and the vertical plane are shown in Fig. 8-5a and 8-5b, respectively. The precision is in the order of $\frac{\mathrm{mm}}{\mathrm{s}}$ due to the velocity nullifying guidance, which uses nullifying cycle lengths equal to the overall integration step size of 1 s. This guidance logic only considers the current velocity, and therefore cannot account for the current disturbing accelerations from the gravity field models. A small steady-state error is thus always expected.

The plot of the fuel consumption as a function of divergence from the nominal switch altitude is shown in Fig. 8-6a. The point distribution indicates, that a higher switch altitude – thus an earlier switch from the gravity turn guidance to the quadratic guidance – tends to have a positive effect on the fuel consumption. On the other hand, the data points have a high spread in general, so other factors probably have a higher influence. The initial vertical

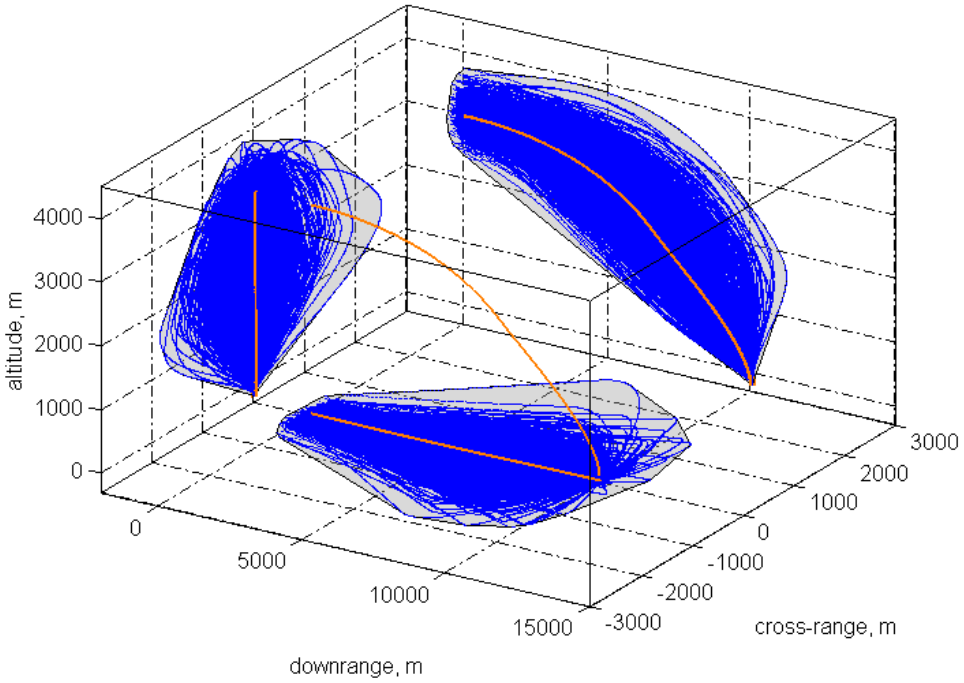


Figure 8-2: Nominal (orange) and diverging (blue) descent trajectory and their envelope (gray)

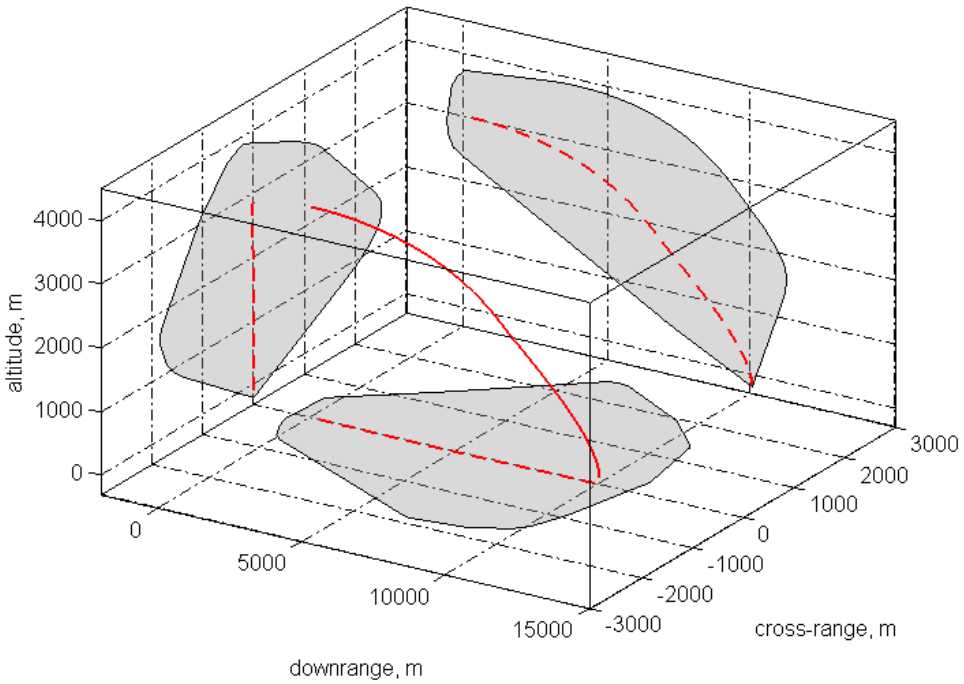


Figure 8-3: Nominal descent trajectory (red) and envelopes of the diverging lander models (gray)

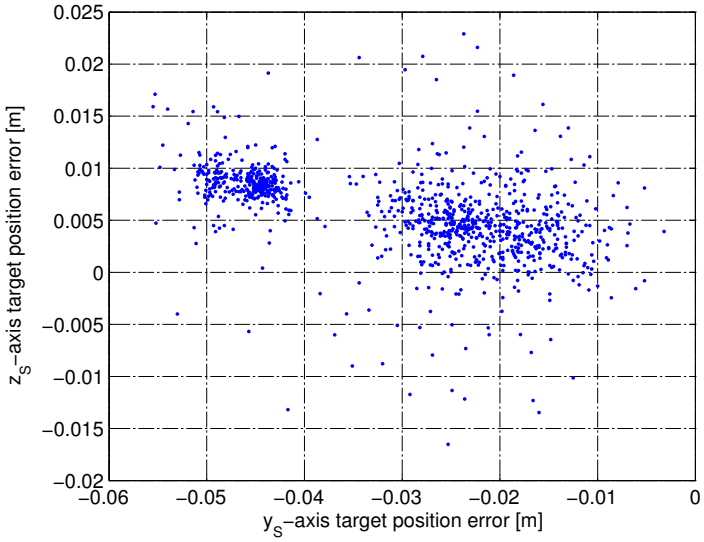
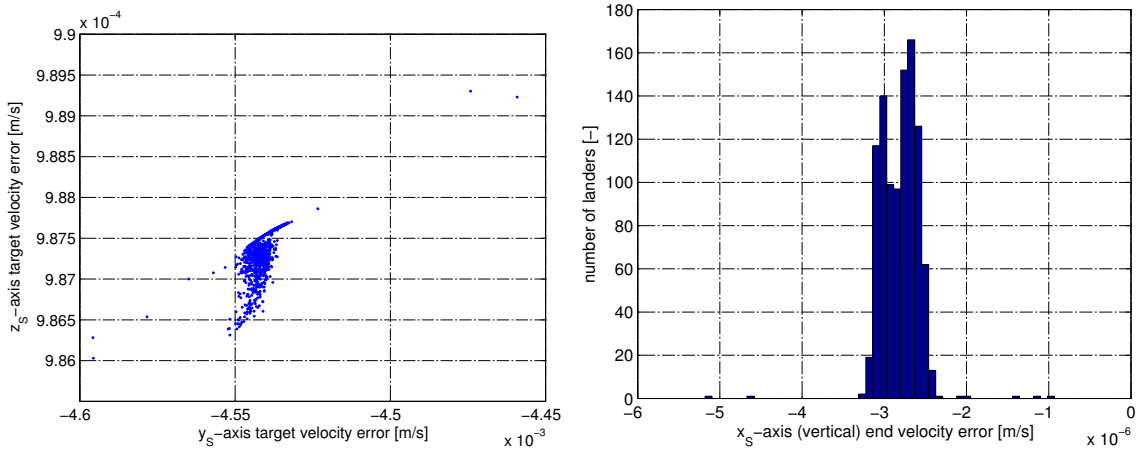
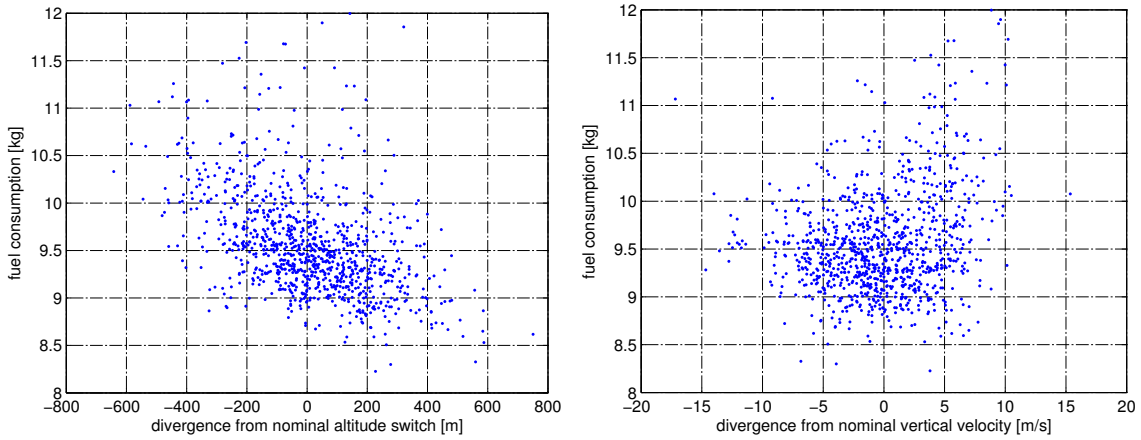


Figure 8-4: Landing precision on the horizontal plane



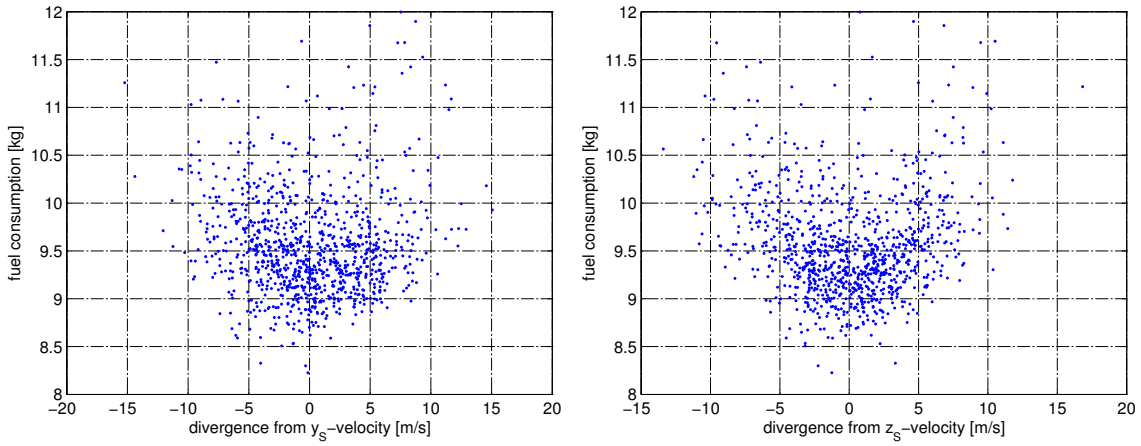
(a) Horizontal velocities **(b)** Vertical velocity

Figure 8-5: Velocities at touchdown



(a) Altitude switch **(b)** Vertical velocity

Figure 8-6: Correlation between the fuel consumption and the initial vertical velocity, and altitude at which the guidance system switches from gravity turn to quadratic guidance



(a) Downrange velocity **(b)** Cross-range velocity

Figure 8-7: Correlation between the fuel consumption and the initial vertical velocities with respect to the nominal lander

velocity divergence, for example, also as a influence on the fuel consumption, see Fig. 8-6b – in fact, the spread is comparable. The fuel consumption as a function of the initial horizontal velocity, altitude and mass is much more intuitive. The correlation between the fuel consumption and the initial downrange and cross-range velocities is exemplary illustrated in Figs. 8-7a and 8-7b. A non-zero Z_S-velocity requires additional fuel to decelerate, while a slightly lower forward velocity – down to 5 m/s slower than the nominal lander model – can reduce the fuel consumption.

8-3-2 Guidance & Control Simulations

The combined guidance and control (GC) system simulations evaluate the landing precision of a spacecraft in case the navigation system is the only deactivated GNC system, returning perfect state estimations. The lander models are now no longer point masses, but three-dimensional objects, whose rotational and translational motion is coordinated by the control system, using the outputs of the guidance system. The control system determines the reference torque necessary to align the thrust vector ($-Z_B$-axis) with the acceleration command produced in the guidance system. Then, the PWPF controller and the thruster selection logic are called to find the appropriate main engine and vernier thrust settings. The activation of the control system will lead to a delayed translation of the acceleration command, and a limitation of the thrust magnitudes between the specified minimal and maximal thrust levels – the guidance system simulation in the previous section assumed a perfectly aligned main engine thrust without any thrust magnitude limitations. It is expected, that some of the previously successful landings will now fail because the GC system is unable to handle the involved errors. The GC system simulation use the same initial position, velocity and mass values for each lander model as used in the previous guidance system simulation due to unchanged variance values and a fixed seed value for the random number generators in the Monte Carlo simulation setup. In this way it is possible to investigate the effects of activated control system. Note that the true attitude standard deviation was active during the guidance system simulation runs, but did not have any effect the results, because all lander models were point masses.

The initial state as defined in Table 8-2 introduces a large initial alignment error in the order of 90°, because the main engine aligned with the Z_B-axis points in the northern direction instead of in the flight direction. The moment commands and thus the vernier thrust levels are expected to be large during the first couple of seconds. An exemplary thrust moment history is given in Fig. 8-8, which also shows the discrepancy between expected and the true thrust levels as a consequence of the control variances.

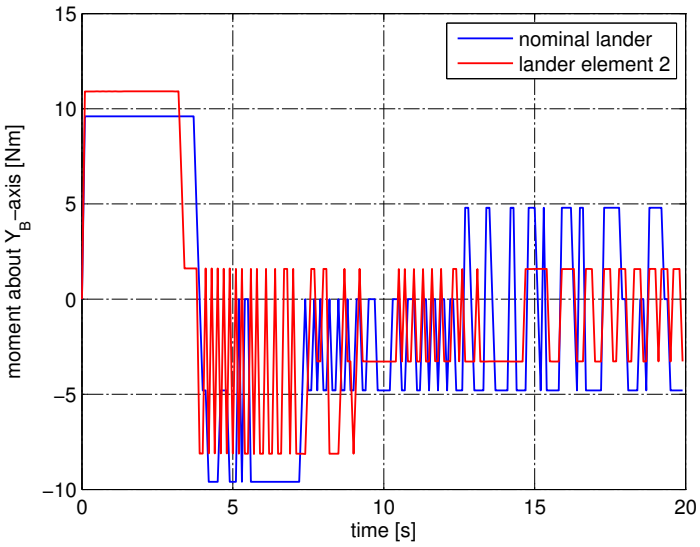
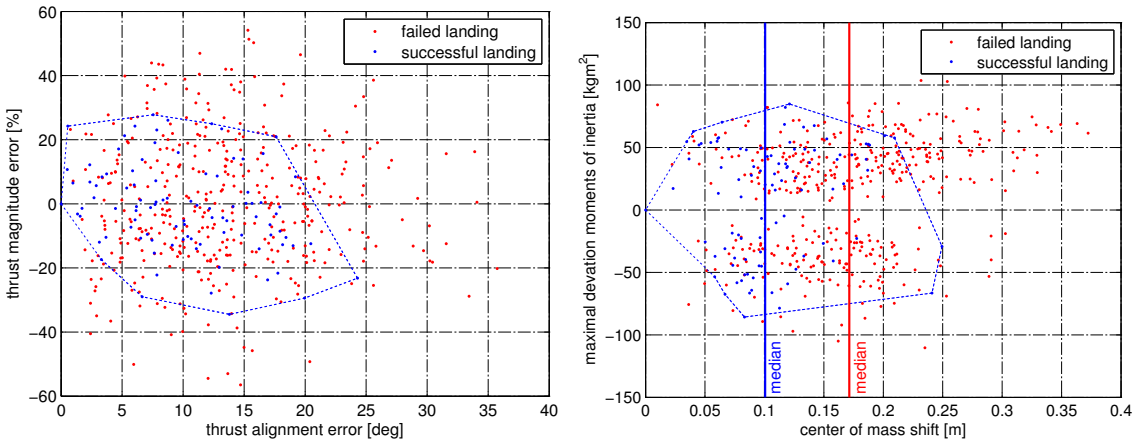


Figure 8-8: Thrust moments for the nominal lander and the first Monte Carlo lander model

Name	Value	Comment
integrator	RKF45	with $\Delta t_0 = 0.01$ s
altitude switch $1\sigma_{as}$	0 m	not of interest here
true attitude $1\sigma_q$	5°	now has effect
c.o.m. shift $1\sigma_{com}$	0.04 m	5% of lander base radius
moment of inertia $1\sigma_{moi}$	7.5 kgm^2	5% of nominal value
thrust magnitude $1\sigma_{tm}$	5%	–
thrust alignment $1\sigma_{ta}$	0.5°	–
guidance frequency	10 Hz	increased
control frequency	20 Hz	–
velocity nullifying duration	10 s	increased

Table 8-4: Changes and additions to the simulation parameters for the combined GC system, with respect to the previous section

The simulation parameters for the GC system are collected in Table 8-4. The list comprises only the changes and additions with respect to the setup used in the previous section. The guidance frequency is increased from 1 Hz to 10 Hz to cope with the new rotational motion. The control frequency of 20 Hz is sufficiently high to eliminate situations where there spacecraft becomes uncontrollable due to slow thrust setting updates. The velocity nullifying guidance duration is increased from 1 s to 10 s, because the attitude control generates large moment commands in case the duration is small. If this value was left unchanged, the nullifying guidance causes a high percentage of all lander simulation models to fail during the terminal guidance phase. The modification solves this problem, but it is another indication (see Section 5-1-3) that the implementation of a feedback nullifying guidance logic should be a subject of future work.



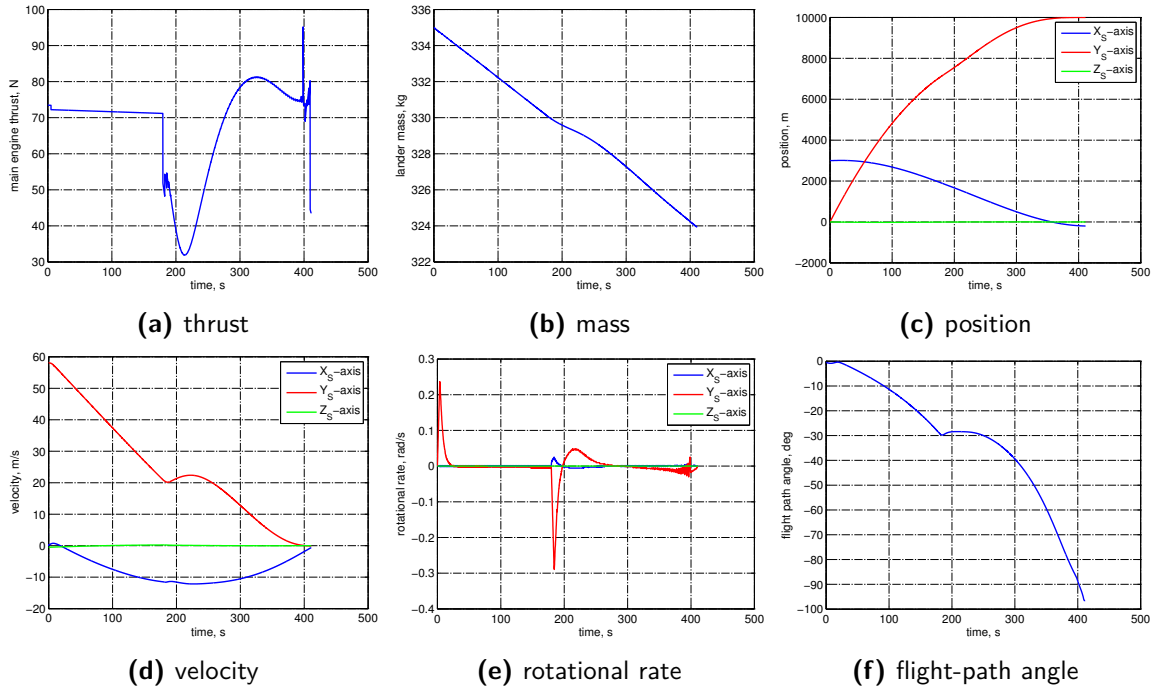
(a) main thruster errors **(b)** mass distribution

Figure 8-9: Influence of control variances on the landing success

The robustness of the control system is tested by applying variances on the center of mass position, the moment of inertia, the thrust magnitudes and the thrust alignment. The system sensitivity to the control variances was determined in a separate simulation run, where the standard deviations were so high that more than half of all landings failed. Figure 8-9 shows the simulation results for standard deviations $\sigma_{com} = 0.1$ m, $\sigma_{moi}= 30$ kgm^2, $\sigma_{tm} = 20\%$ and $\sigma_{ta} = 10^\circ$. The system sensitivity to variations of the moment of inertia, and to the thrust magnitude and alignment error is relatively low; the lander can handle errors in the order of 80 kgm^2, 25% and 20°, respectively. The main reason for the low sensitivity is the feedback loop in the PWPF controller: the motion errors from a previous control cycle have a direct, corrective influence on the subsequent cycles. The center of mass shift, on the other hand, has a large influence on the mission success: the position shift is less than 0.1 m for 50% of all successful landings, while the median for the failed landings is 0.17 m. The main reason for failures in this case is the moment induced by the main engine as a consequence of the wrong mass distribution. The attitude verniers are not powerful enough to compensate for this effect beyond a certain threshold.

The standard deviations for the guidance and control system simulations are considerably lower, but still large enough to account for other error sources that are not included in the lander models. The 1σ-value for the center of mass shift is 5% of the lander base radius, and the one for the moment of inertia 5% of the nominal value of 150 kgm^2. A normal thruster has a misalignment in the order 0.1° (Cornelisse et al., 1979), while the thrust magnitude error can determined in experiments with high accuracy.

The plots in Fig. 8-10 give an overview of the most important flight parameters for the nominal lander mission. The spacecraft model is no longer a point mass, so the rotational rate and flight-path angle are now added to the nominal mission parameters (compared with Fig. 8-1). The gravity turn guidance phase is approximately the same as for the G simulations. At the beginning of the quadratic guidance phase, the main engine is not perfectly aligned with the thrust command vector. The control system requires about 10 s to reorient the spacecraft; in this phase, the rotational rate values increase (see Fig. 8-10e), and thrust main engine thrust fluctuates, as the quadratic guidance updates the acceleration commands based on



(a) thrust **(b)** mass **(c)** position

(d) velocity **(e)** rotational rate **(f)** flight-path angle

Figure 8-10: Nominal mission parameters: descent GC simulations

the current state. The velocity nullifying guidance again returns comparatively large thrust values up to 100 N, but the values with the original update period of 1 s were much higher and exceeded the engine thrust limit. The target position and velocities are all reached; the final values are close to the target state defined in Table 8-2.

The failure rate for current GC system configuration is 7.3%. The center of mass shift in all these cases is more than 0.053 m and thus clearly above the 1σ-value of 0.04 m listed in Table 8-4. As might be expected based on the sensitivity analysis from Fig. 8-9, the other parameters have no significant influence on the simulation results. The success rate can be increased further by applying stricter requirements on lander's mass distribution. Most of the failed lander simulations were stopped due to large, uncontrollable rotational rates. The end values for position, velocity and attitude are not meaningful and thus excluded from the following discussions in this section.

Figure 8-11 shows the decent trajectory envelopes for the successful landings with activated GC system. The envelope shapes are similar to the ones generated with deactivated control system given in Fig. 8-3. The extreme values increase slightly as a consequence of the new control reaction time and thrust limitations.

The landing precision in the horizontal plane for the GC system is illustrated in Fig. 8-12a The result clusters present in the results for the guidance system simulations in Fig. 8-4 have disappeared due to the higher guidance frequency. The mean cross-range and downrange errors are 0.11 m and -0.69 m, respectively. The position errors are not corrected while the velocity nullifying guidance is active, so the slightly negative downrange error is a consequence of the last quadratic guidance logic commands, the disturbing forces during the terminal approach and possibly the common approach direction along the $+Y_S$-axis. This tendency

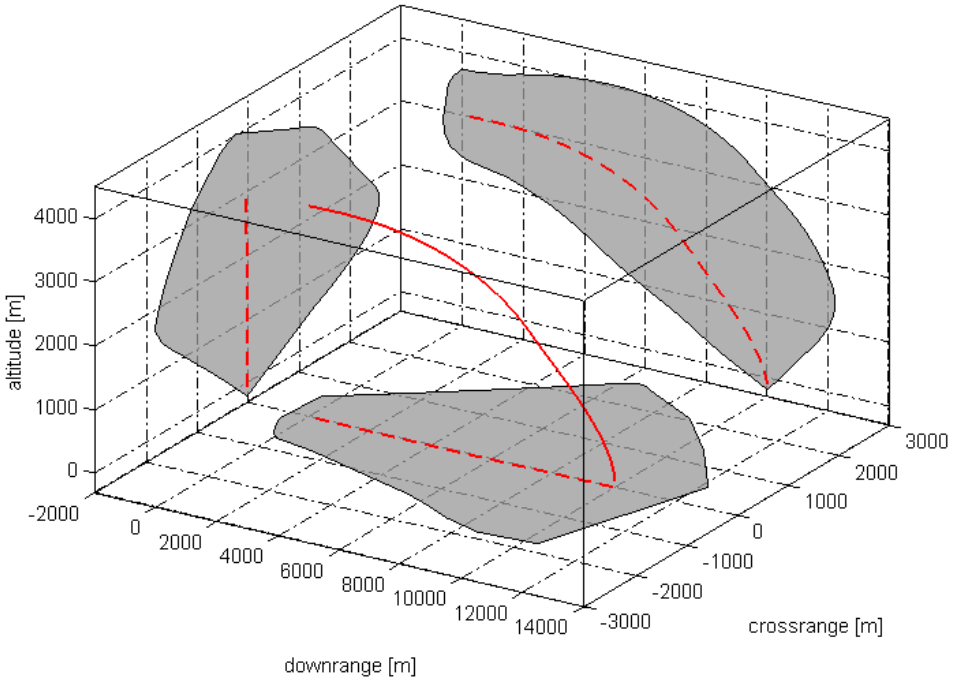
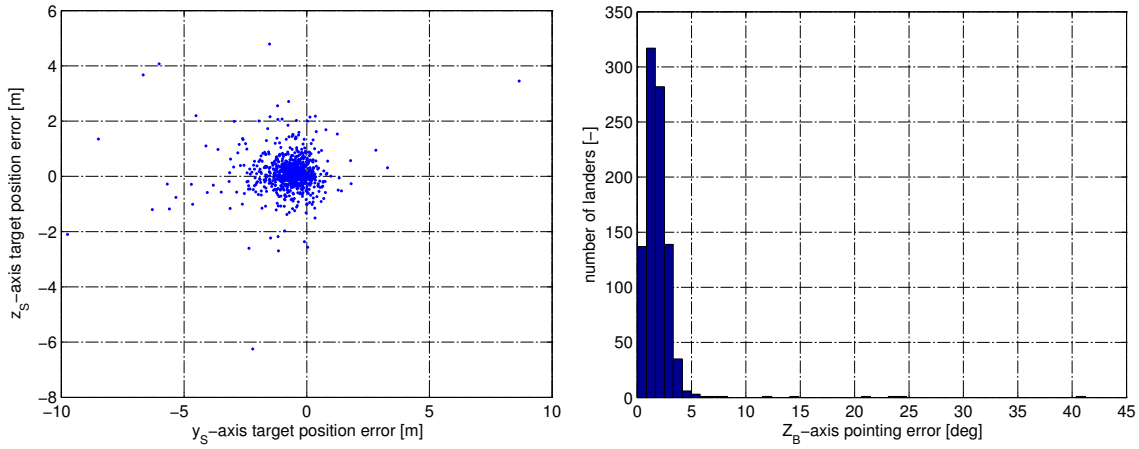
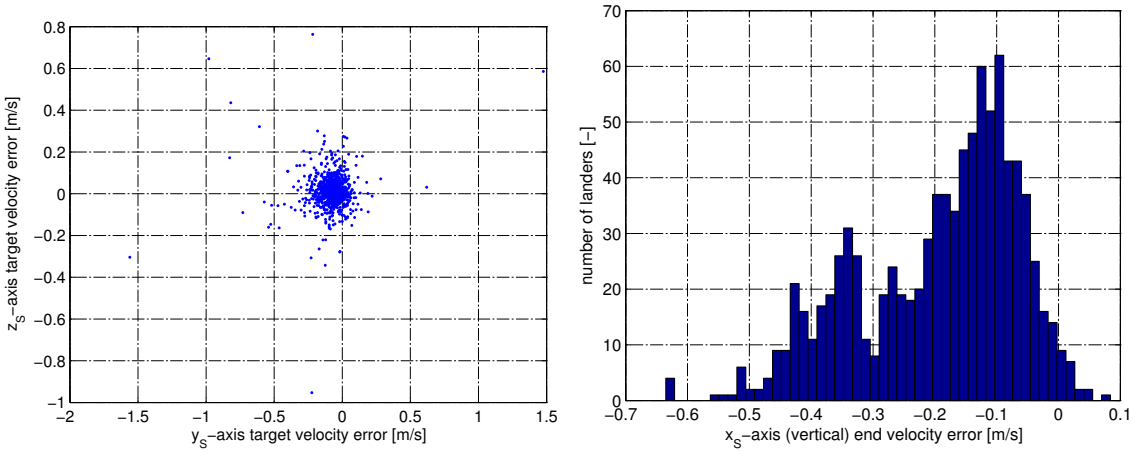


Figure 8-11: Nominal descent trajectory (red) and envelopes of the divergent lander models (gray)



(a) landing precision **(b)** pointing error

Figure 8-12: Landing accuracies and pointing errors for the combined GC system



(a) horizontal velocities **(b)** vertical velocity

Figure 8-13: Velocities at touchdown for the combined GC system

is also indicated in Fig. 8-4 and further amplified by the control system activation. The standard deviation from the mean landing point is 0.94 m.

The pointing error, or the angle between the Z_B-axis and nadir, is a variable that could not be measured in the previous simulations with only three degrees of freedom. Ideally, it is zero, because the quadratic guidance system is programmed to eliminate the horizontal velocities 10 s before touchdown, so the acceleration vector should be vertical at that instance of time. The velocity nullifying guidance, however, aims at removing of the translational and rotational velocities by realigning the main engine and activating appropriate vernier pairs. These orientation errors cannot be corrected due to the high velocity nullifying duration as discussed before. Figure 8-12b indicates, that the pointing error is below 5° for majority of all landings; the pointing error has a mean value 1.92° with a standard deviation of 2.06°. These small errors allow the landing gear to absorb the impact energy more efficiently, see Section 3-3.

The end velocity errors for the horizontal and vertical plane are shown in Fig. 8-13a and 8-13b, respectively, and can be directly compared with Fig. 8-5a and 8-5b in the previous section. The downrange and cross-rang velocities are now clustered around the mean values -0.08 ms^{-1} and 0.01 ms^{-1} with a standard deviation of 0.12 ms^{-1}. The mean value of the vertical velocity error is -0.19 ms^{-1} with $1\sigma = 0.13$ ms^{-1} and thus higher than its horizontal counterparts. The main reason for this is the combination of the lander's pointing error and the main engine's thrust magnitude error, which are not recognized during the terminal descent phase, where quadratic and velocity nullifying guidance are active simultaneously (guidance mode 5, see Table 5-4). Note, that the vertical target velocity is -0.5 ms^{-1}, so a positive velocity error does describe an upward velocity, but merely a lower touchdown velocity. Despite the disadvantages of the current velocity nullifying guidance setup, the GC system is able to bring the end velocities very close the desired target values.

All four control-system variances given in Table 8-4 have only a low influence on the landing precision of the successful touchdowns. The most precise landings tend to have lower divergences from the nominal values, but higher divergences only slightly increase the result spread.

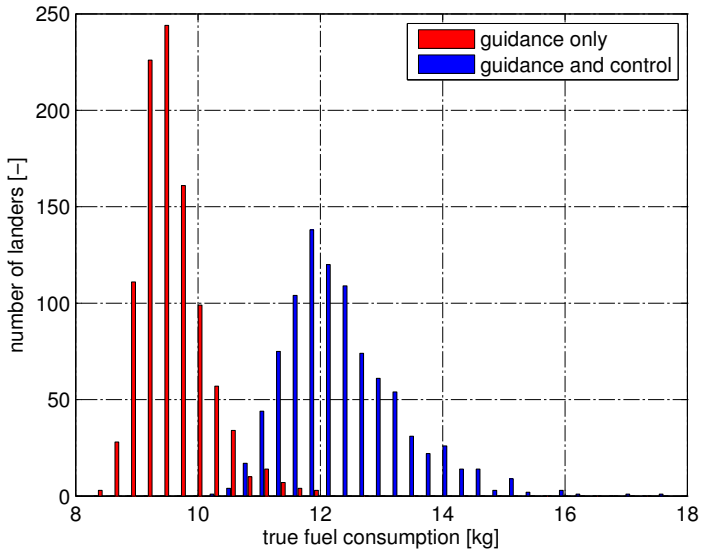


Figure 8-14: Comparison of the fuel consumption for G- and GC-simulations

The average fuel consumption for the GC system simulations is 12.39 kg and thus – as expected – higher than the 9.56 kg for the G system simulations, because the attitude thrusters are now active, and the main engine is not perfectly aligned. A comparison of the two values is shown in Fig. 8-14, the discussion on the fuel consumption as consequence of the PWPF controller can be found in Section 5-2-4.

8-3-3 Guidance, Control & Navigation Simulations

The guidance, navigation and control system simulations evaluate the landing precision of the Enceladus lander with the fully operational flight software. With the activation of the navigation system, the input for the spacecraft GC system are now state estimations, which are generated and regularly updated with the on-board navigation system. The navigation system uses the IMU to propagate the previous state estimation to the instance of time, where external measurements from the range instrument or the star sensors or both are available. These noisy data are then used to update the state predication and the error estimations. The IMU model includes bias, alignment errors and scale factor errors for both the accelerometers and the gyroscopes, but as discussed in Chapter 5, only the biases can be estimated with sufficient accuracy. The alignment and scale-factor errors must be determined in pre-flight calibrations as accurately as possible, so that the navigation filter can use the best estimates for these values. The remaining uncertainty is thus divided over all state estimations, which will cause the bias estimations to diverge from the true bias values. For the simulations it is assumed that pre-flight calibrations provide alignment and scale-factor error estimation with an accuracy of 90%. The exact parameters are based on the IMU characteristics listed in Table 6-4, and can be found in the simulator configuration file in Appendix A-2. All values are slightly higher than the typical maximal errors in order to account for error sources not included in the IMU model and to test the robustness of the system. The biases have a large influence on the IMU output and on the state estimations, and should always be

estimated with by means of in-situ measurements from other instruments, in this case, the range instrument and the star sensors. The initial bias estimations are thus zero for all axes.

Name	Value	Comment
position estimation $1\sigma_{r,e}$	630 m	$>\sigma_r$, Table 8-3
velocity estimation $1\sigma_{v,e}$	10 ms^{-1}	$>\sigma_v$, Table 8-3
attitude estimation $1\sigma_{q,e}$	10°	$>\sigma_q$, Table 8-4
IMU frequency	20 Hz	low, see discussion
Range Instrument		
– frequency	10 Hz	low estimate
– 1σ noise	0.2 m	high estimate
Star Sensors		
– frequency	4 Hz	–
– 1σ noise	0.0001	quaternion element error

Table 8-5: Additions to the simulation parameters for the GNC system

The relevant GNC system simulation parameters are listed in Table 8-5. No values were changed with respect to the previous section. The range instrument frequency and measurement noise values follow from conservative estimations, because the device is a concept design only, as discussed in Section 6-1-7. The performances of the laser rangers and the light sensors are individually much better, so it is expected, that system will have a higher frequency and a lower measurement noise than the assumed 10 Hz and 0.2 m, respectively. The performance of the star sensors, on the other hand, is well-known; their frequency and noise values follow from Table 6-6. The 1σ noise must be specified in terms of quaternion elements. The conversion between the accuracy in terms of degree and quaternion elements is discussed in Section 6-1-5. The chosen IMU frequency of 20 Hz is clearly below the instrument capabilities, but initial simulations showed that a reduction from 100 Hz to 20 Hz has no significant influence on the end results. On the other hand, a lower IMU frequency greatly reduces the required simulation time, which is already high with more than 8 hours for 1000 lander elements with a fully activated GNC system. The numerous navigation filter parameters are not listed here; their values follow from the discussions in the respective sections of this report and can be directly in the simulator configuration file in Appendix A-2.

Similar to the approach in the previous section, the sensitivity of the navigation system is tested in a separate simulation run with very large values for the navigation system related Monte Carlo variance parameters of position, velocity and attitude estimation. Note, that a variance on the initial rotational rate estimation has no effect, because the navigation filter extracts this information directly from the gyroscope measurements; the rotational rate is not part of the EKF state vector, see discussion in Section 6-1-5. For very large values $\sigma_{r,e} = 15$ km, $\sigma_{v,e} = 20$ ms^{-1} and $\sigma_{q,e} = 90^\circ$, the failure rate increases to 46%. The main reason for this is in fact not directly related to the navigation system, but to the gravity-turn guidance: in almost halve of all cases, the estimated altitude is below Enceladus' surface, as the initial lander altitude is only 3 km (varies based on σ_r). A negative altitude estimation leads to undefined behavior particularly in the gravity turn guidance system – not a singe successful touchdown can be achieved with a negative altitude estimation, see Fig. 8-15 . The state estimates are corrected quickly, but the largely implicit gravity turn guidance logic is not able

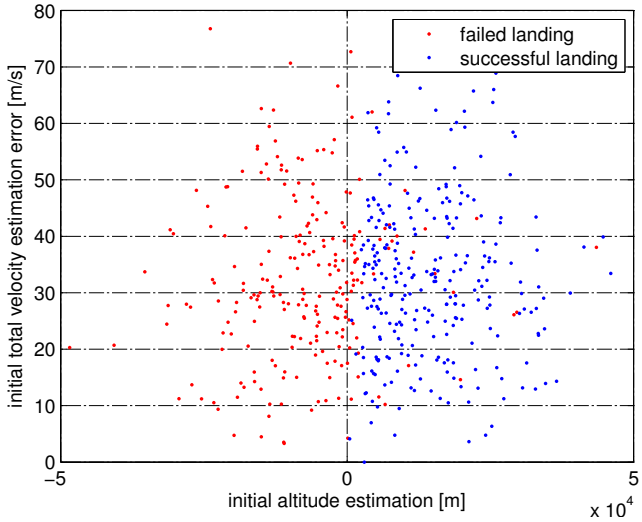
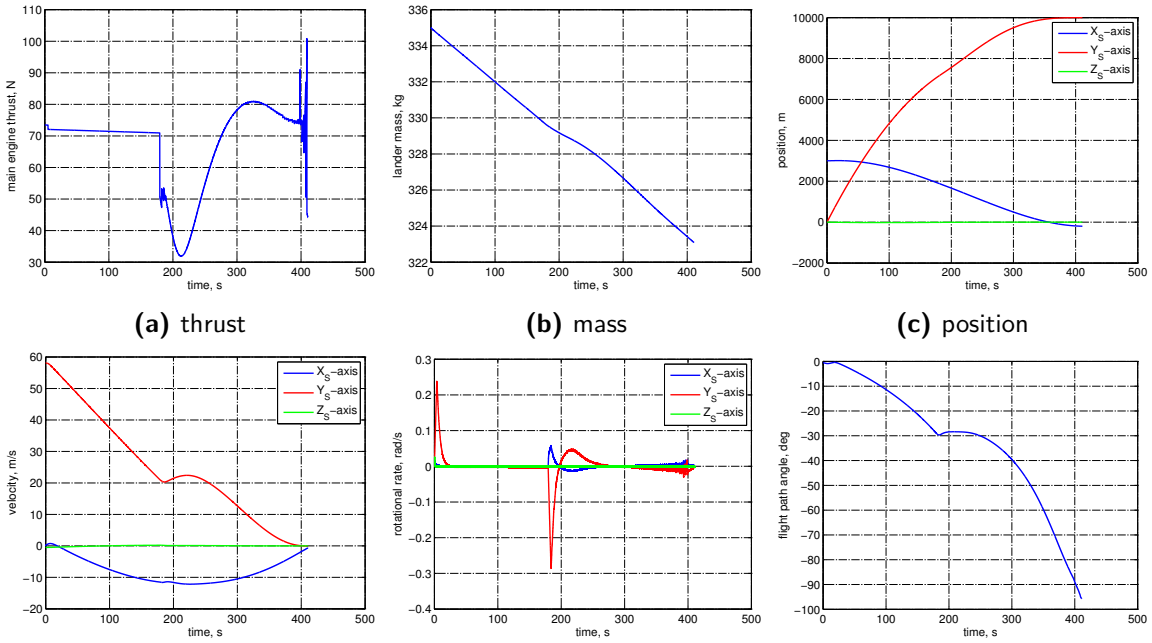


Figure 8-15: Influence of navigation variances on the landing success

to update the acceleration commands. The guidance system may be updated in order to deal with these situations, but the simplest solution is to reject any negative altitude estimation. For the remaining unsuccessful landings with a positive initial altitude estimation, there is only a very weak correlation to the initial vertical velocity estimation error, but it is not possible derive upper boundaries for the position, velocity and attitude estimation standard deviations. Again, the current setup of the gravity turn guidance logic is the reason for the failures: For unfavorable combinations of true and estimated state vectors, the commanded thrust direction is too shallow to bring the lander to switch altitude of 2000 m. In these cases, the singular gravity-turn update (see Table 5-6) after 5 s is ineffective. The activation of continuous gravity turn acceleration command updates (see discussion Eq. (5-15)) in the simulator configuration in theory solves this problem at some point, but simulations with the GC system indicated a low tolerance for the variances on the control system related parameters and an increased failure rate. The effect of state estimation errors and an imperfect lander models on the gravity turn guidance is an interesting subject for further studies.

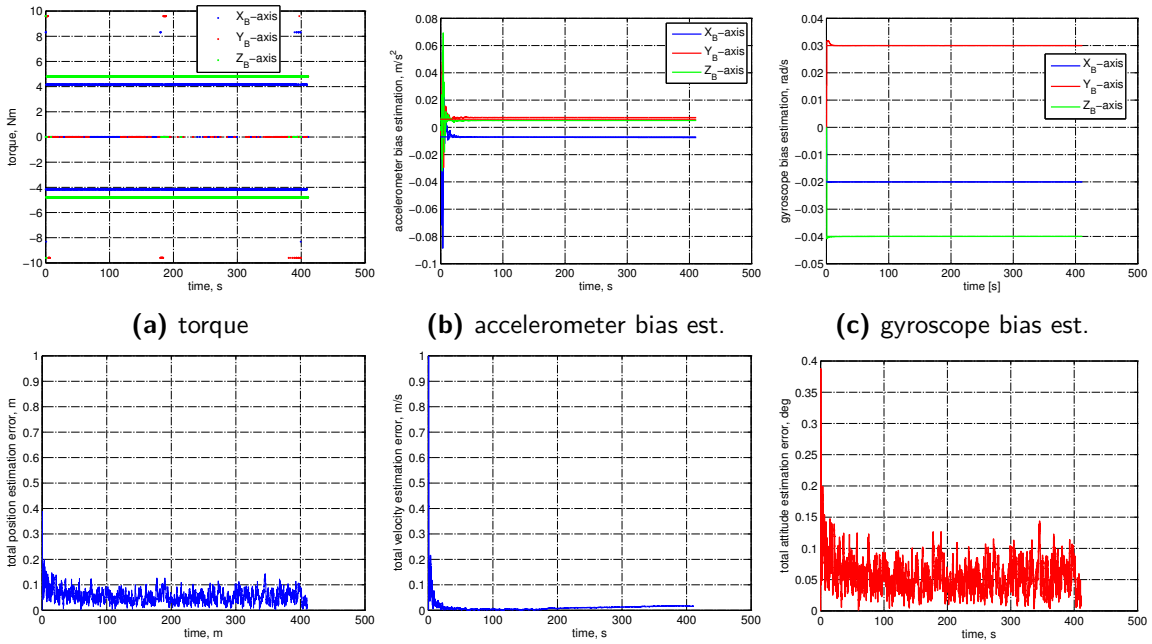
The plots in Figs. 8-16 and 8-17 give an overview of the most important flight parameters for the nominal lander mission. The navigation system is now active, so the evolutions of the state estimations are now added to the nominal mission parameters (compared with Fig. 8-10). The differences between the thrust, mass, position, velocity rotational rate and flight-path angle results for the GC- and the GNC-simulations are hardly visible. This is a consequence of the good estimation results: the accelerometer and gyroscope bias estimation errors are below 5%, see Figs. 8-17b and 8-17c, and the position, velocity and attitude estimation errors are in the order of 0.1 m, 0.02 m/s and 0.1°, respectively – see Figs. 8-17d, 8-17e and 8-17f. The torque levels shown in Fig. 8-17a are only maximal, when – according to the guidance system – the orientation error of the lander is high; this is the case at the beginning of the simulation, at the switchover time between the gravity turn guidance and the quadratic guidance after about 190 s, and at the switchover time between the quadratic guidance and the velocity nullifying guidance after 390 s.



(a) thrust **(b)** mass **(c)** position

(d) velocity **(e)** rotational rate **(f)** flight-path angle

Figure 8-16: Nominal mission parameters: descent GNC simulations, part 1



(a) torque **(b)** accelerometer bias est. **(c)** gyroscope bias est.

(d) position est. error **(e)** velocity est. error **(f)** attitude est. error

Figure 8-17: Nominal mission parameters: descent GNC simulations, part 2

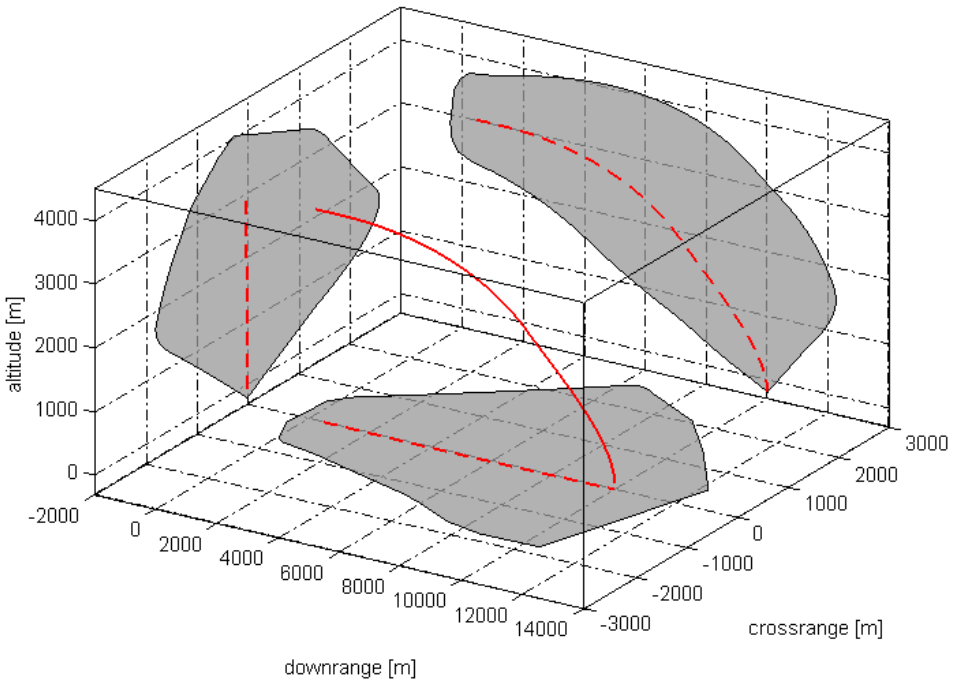
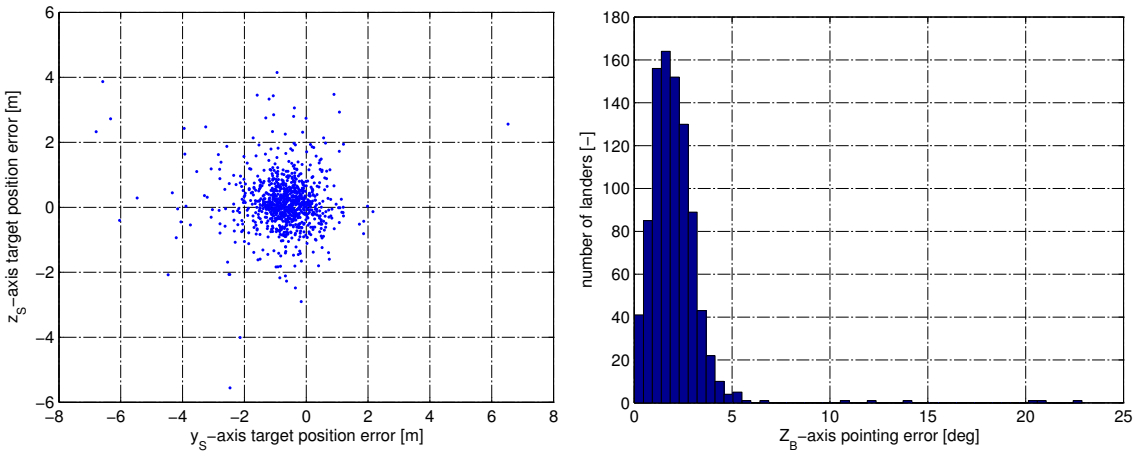


Figure 8-18: Nominal descent trajectory (red) and envelopes of the divergent lander models (gray)

The GNC simulations with the normal system configuration based on Table 8-5 have a failure rate of 10.6%, which is an increase of 3.3 percent points with respect to the GC simulations in the precious section. The main reason for this does not actually lie in the navigation system, but – as mentioned before – in the guidance system: the guidance system now is confronted with state *estimations* instead of the true state – large initial state estimation errors will lead to inadequate guidance commands. The navigation is able to accurately determine the accelerometer and gyroscope biases within seconds, see Figs. 8-17b and 8-17c. The accelerometer biases deviate slightly more from the true values than the gyroscope biases, because the EKF does not include the third body perturbation from Saturn and the effects of the J_2-terms from the gravitational field of both Enceladus and Saturn. The initial estimation errors are reduced rapidly due to the comparatively high instrument accuracy.

Figure 8-18 shows the descent trajectory envelopes for the successful landings with fully activated GNC systems. The activation of the navigation system has no visible impact on the envelope shapes compared with results from the GC simulations shown in Fig. 8-11. The actual values obviously change due to the additional uncertainties, but the differences are minimal. For a better insight into the true trajectory shapes, a scaled version of Fig. 8-18 is given in Fig. 8-20.

The true horizontal landing precision for the GNC simulations is displayed in Fig. 8-19a. The mean downrange and cross-range errors are -0.78 m and 0.16 m, respectively, with a standard deviation of 1.04 m. Compared with the GC simulations, the landing precision decreases by a few centimeters, combined with a higher result spread. The accuracy loss is in the same order as the position estimation errors. The overall accuracy is still well within the required



(a) landing precision **(b)** pointing error

Figure 8-19: Landing precision and pointing errors at touchdown for the complete GNC system

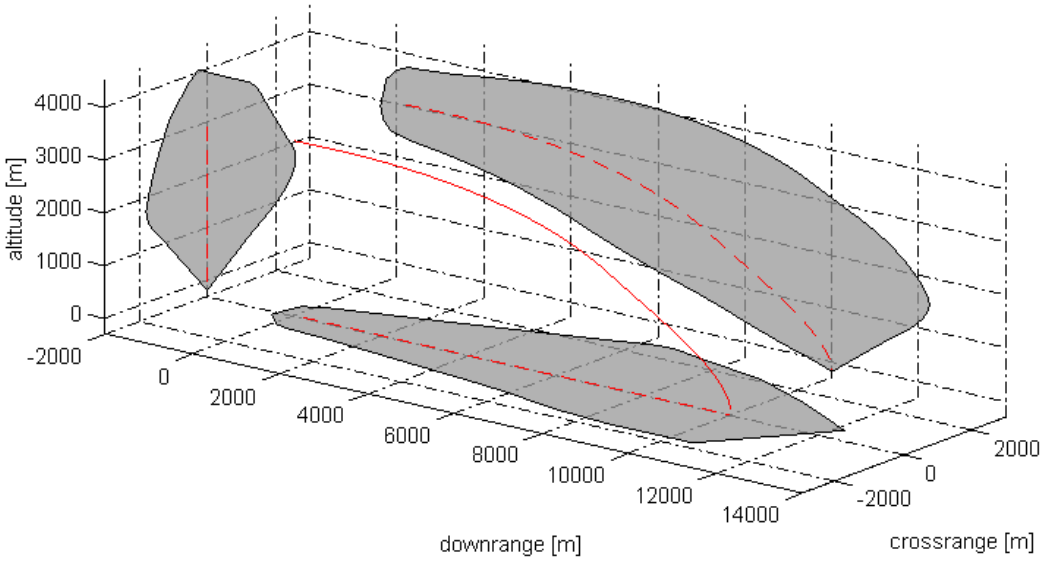
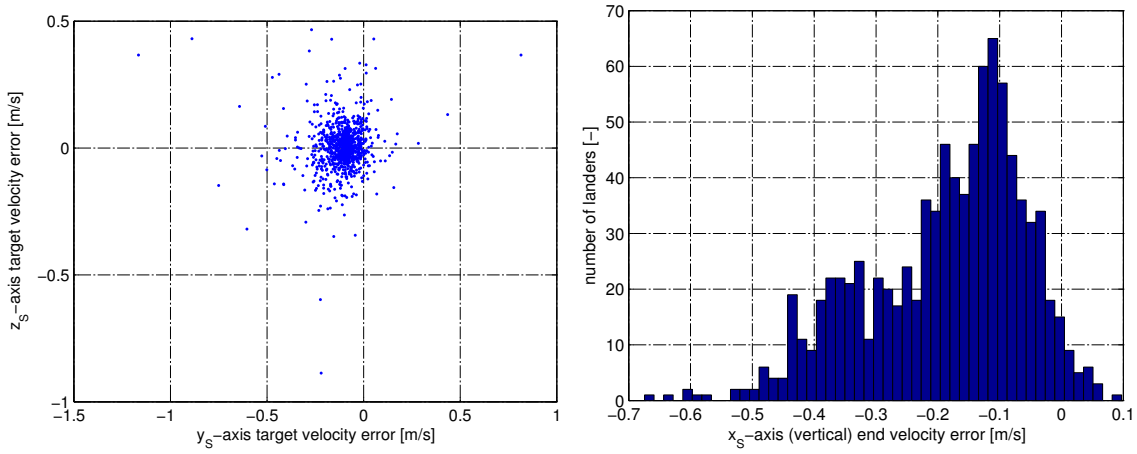


Figure 8-20: Scaled descent trajectories for the full GNC simulations



(a) horizontal velocities **(b)** vertical velocity

Figure 8-21: Velocities at touchdown for the complete GNC system

10 m boundary.

The influence of the navigation system on the pointing error is also small; the mean error increases by 0.15° to 2.08° with a decreased standard deviation of 1.55°. As discussed before, the attitude estimations errors are very small, so any changes in the pointing errors are mainly a consequence of the position and velocity estimation errors and their influence on the commanded thrust direction.

The horizontal velocity errors are illustrated in Fig. 8-21a. The mean downrange and cross-range velocity errors are -0.08 m and 0.01 m, respectively, with a standard deviation of 0.11 m. The vertical velocity error shown in Fig. 8-21b has a mean value of -0.19 ms^{-1} with a standard deviation of 0.14 ms^{-1}. The true end velocity errors are almost equal to the results from the GC simulations.

As might have been expected, the values for the initial position, velocity and attitude estimation errors have no direct influence on the landing precision for successful touchdowns. It is therefore decided to not include graphs showing precision as functions of the deviations from the nominal states in this section.

The average fuel consumption for the complete GNC simulations is 12.43 kg, which is only just above the average of 12.39 kg for the GC system. The navigation system tends to have a negative effect on the landers with an already above-average fuel consumption – for the lower end, the opposite is true. A comparison of the true fuel consumptions for the G, GC and GNC simulations can be found in Fig. 8-22.

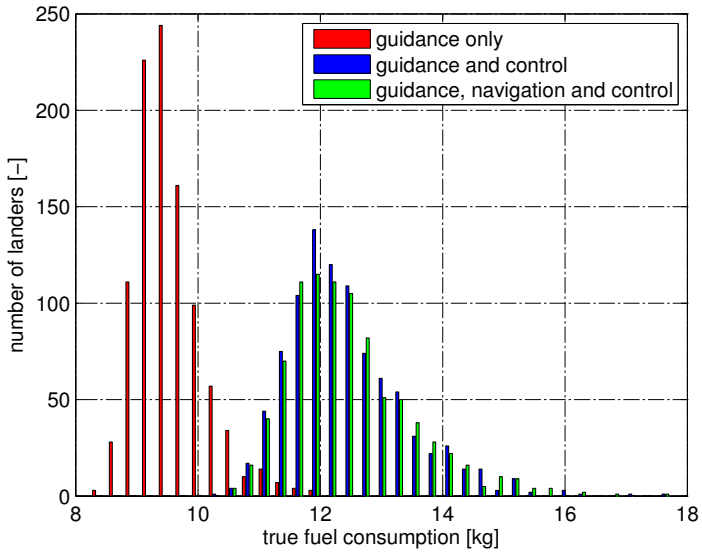


Figure 8-22: Comparison of the fuel consumption for G-, GC- and GNC-simulations

8-4 Repositioning

The reposition simulations evaluate the precision and fuel consumption of the purely quadratic repositioning guidance and the hybrid ballistic-quadratic guidance system discussed in Section 5-1. Both systems use the full GNC system as in the previous sections with the same configuration parameters unless specified otherwise. The nominal jump distance and required altitude are 1000 m and 300 m, respectively. The latter generally follows from the hazard avoidance requirements, in particular the LIDAR specifications. All landers start on the same point, but their respective target on the Y_S-axis different depending on the jump distance variance in the Monte Carlo simulation.

The hybrid ballistic-quadratic guidance system is initiated with a 20 s low thrust command (15% of maximum value) in the direction of the calculated elevation angle. This technique is necessary to ensure that the lander is oriented correctly by the control system, before the short phase of maximal thrust is initiated. The main engine is completely shut down as soon as the lander reaches the desired impulsive shot velocity. A virtual acceleration command in the $-Y_S$-direction reorients the main engine in the opposite flight direction, which reduces the initial pointing error for the deceleration process. As soon as the lander is descending and reaches a certain altitude (set to 200 m), the quadratic guidance system is initiated, which in turn activates the velocity nullifying guidance in the terminal guidance phase. This guidance sequence with the logic identification number 4 – see Table 5-4 – has been successfully used in the previous simulations.

The pure quadratic guidance logic first brings the lander to the pre-programmed intermediate target. As soon as the guidance system indicates that the time-to-go for the first flight phase is reached, the quadratic guidance system is reset and reprogrammed with the actual target location on the $+Y_S$-axis, initiating the second repositioning phase. The terminal phase is, again, controlled by the velocity nullifying guidance.

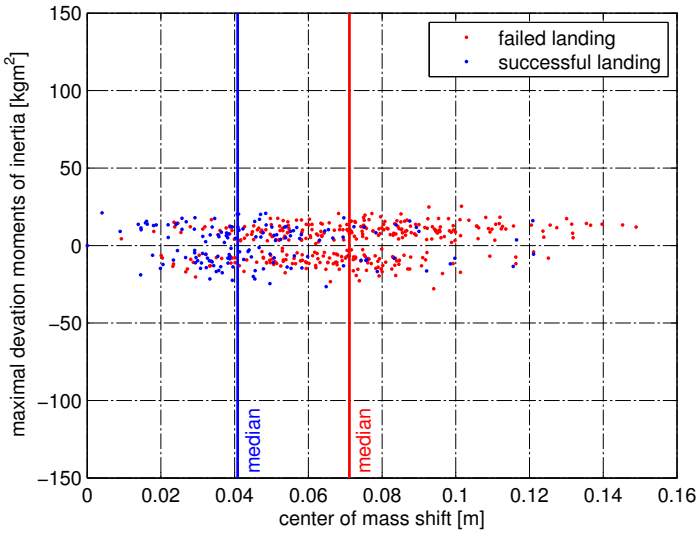


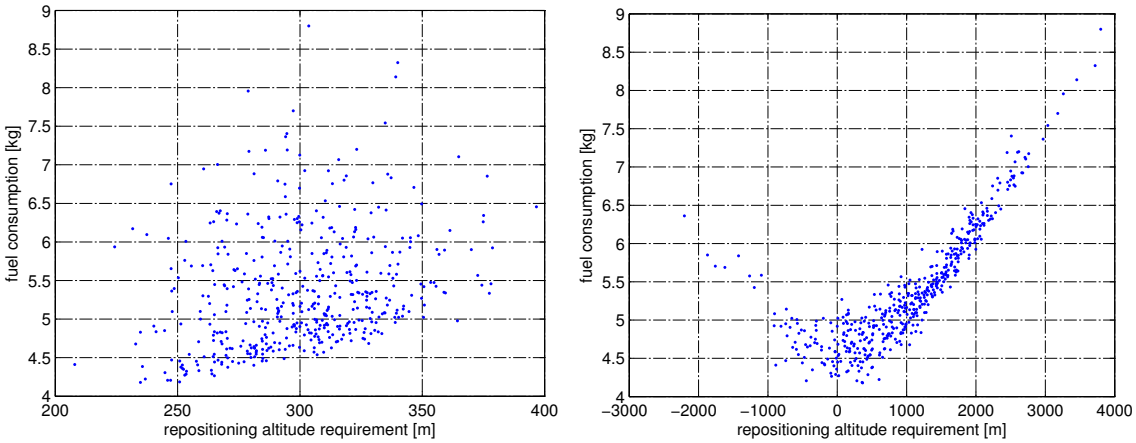
Figure 8-23: Influence of the center of mass shift on the success rate of the hybrid ballistic-quadratic guidance

Name	Value	Comment
lander number	500	reduced for better trajectory visibility
true position $1\sigma_r$	0 m	lander on ground
true velocity $1\sigma_v$	0 m/s	lander on ground
true attitude $1\sigma_q$	0°	lander on ground
position estimation $1\sigma_{r,e}$	5 m	landing accuracy
velocity estimation $1\sigma_{v,e}$	0 m/s	lander on ground
attitude estimation $1\sigma_{q,e}$	0°	very low
hop altitude req. $1\sigma_h$	100 m	–
hop distance $1\sigma_r$	1000 m	–
proportional gain k	2.5	increased for hybrid ballistic-quadratic g.

Table 8-6: Changes and additions to the simulation parameters for the combined GC system, with respect to the GNC simulations.

Table 8-6 lists the relevant simulation parameters for the repositioning simulations. All true state variances – except for the mass – are set zero, because each all lander start from the same position. Similarly, the velocity estimation standard deviation is also removed, as the lander standing on the moon's surface. The position and attitude estimation errors exist, but are small due to the foregoing successful descent process. It was necessary to increase the proportional gain k of the quaternion controller to decrease the reaction time during the ballistic guidance high-acceleration phase.

The *hybrid ballistic-quadratic repositioning guidance system* in the current configuration has a failure rate in the order of 70%. The overwhelming majority of unsuccessful landings are caused by large rotational velocities after the acceleration phase. The main reason for this are the center of mass shifts: in contrast to the previous GC and GNC descent simulations, the



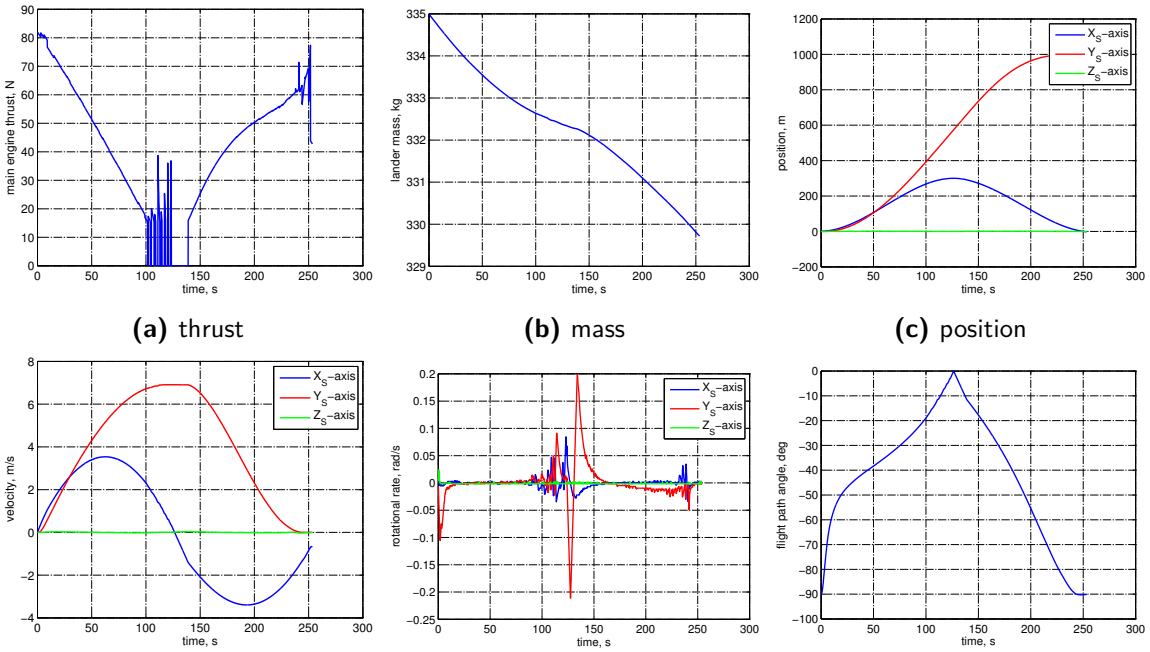
(a) required altitude **(b)** jump distance

Figure 8-24: Fuel consumption for different repositioning conditions

guidance system commands maximum thrust. The disturbing moments induced by the main engine cannot be compensated for by the comparatively weak thrust of the attitude verniers, and the linear quaternion controller is not designed to handle large rotational velocities, see discussion in Section 5-2-1. The influence of mass variations on the landing success is illustrated in Fig. 8-23. As might be expected, the requirements on the maximal allowable center of mass shift value are much stricter than for the low-thrust situation shown in Fig. 8-9b; the success median is now at 4 cm instead of 10 cm. The mean fuel consumption is for the successful landings is 4.8 kg. The fuel consumption for the nominal jump over a distance of 1000 m with a minimal required altitude of 300 m is 4.24 kg, which is clearly above the ideal impulsive shot fuel consumption of 2.35 kg (from Eq. 5-45) mainly due to the quadratic guidance phase, but below the value for the purely quadratic guidance repositioning discussed later in this section.

The plots in Figs. 8-25 and 8-26 give an overview of the most important flight parameters for the nominal lander repositioning phase. The full GNC system is active, so the nominal mission parameters are the same as in Section 8-3. The intermediate target is reached after about 125 s. The main engine is not active shortly before this time instance due to the small t_{go}. The flight software imitates a 10 s coasting phase during which the lander is rotated such, that the main engine is pointing in the direction opposite to the flight direction. The rotational rates increase during this period of time, see Fig. 8-25e, and the flight-path angle changes accordingly, see Fig. 8-25f. The position, velocity and attitude estimation errors are comparable to the results in the previous section, and are still in the order of 0.1 m, 0.02 m/s and 0.1°, respectively – see Figs. 8-17d, 8-17e and 8-17f.

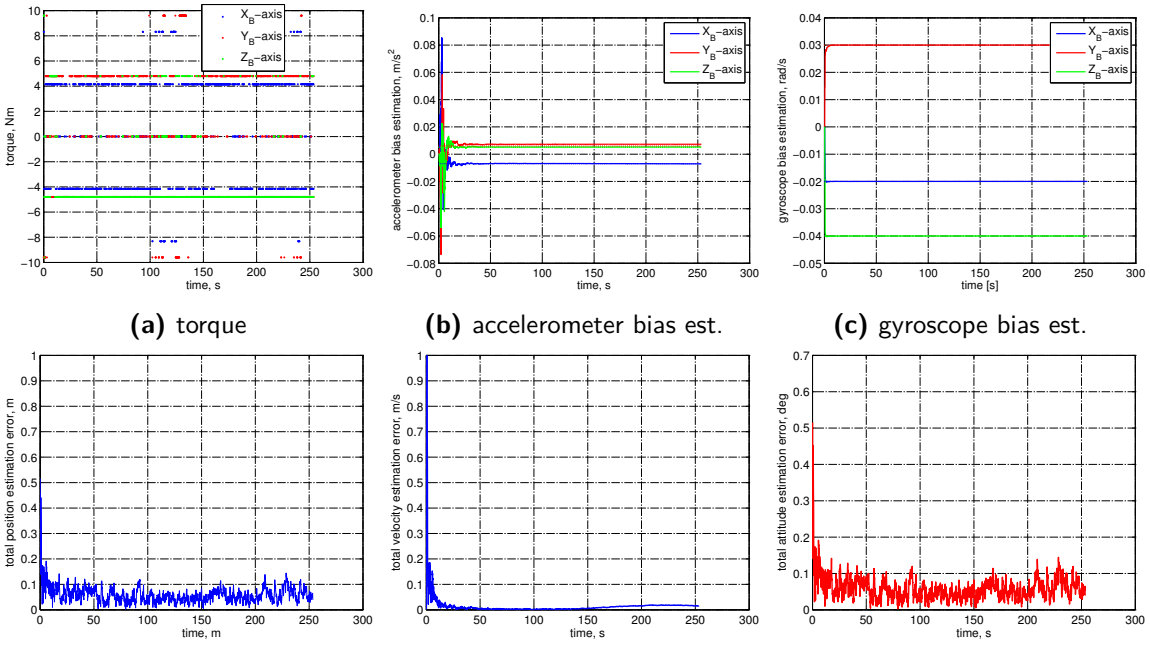
The success rate for the *quadratic repositioning guidance* is 98.2%, the only failures are caused by lander-model elements with center-of-mass shifts larger than 15 cm. The acceleration commands in general very smooth, which allows for a steady adaption to the reference attitude. The explicit guidance scheme constantly corrects for small deviations, unlike the gravity turn guidance used in the descent GNC simulations. In about 40% of all cases, the hazard avoidance indicated, that the original target is not reachable. The retargeting process is successful in all cases. The average fuel consumption is 5.4 kg, and 5.3 kg for the nominal lander model.



(a) thrust **(b)** mass **(c)** position

(d) velocity **(e)** rotational rate **(f)** flight-path angle

Figure 8-25: Nominal mission parameters: repositioning GNC simulations, part 1



(a) torque **(b)** accelerometer bias est. **(c)** gyroscope bias est.

(d) position est. error **(e)** velocity est. error **(f)** attitude est. error

Figure 8-26: Nominal mission parameters: repositioning GNC simulations, part 2

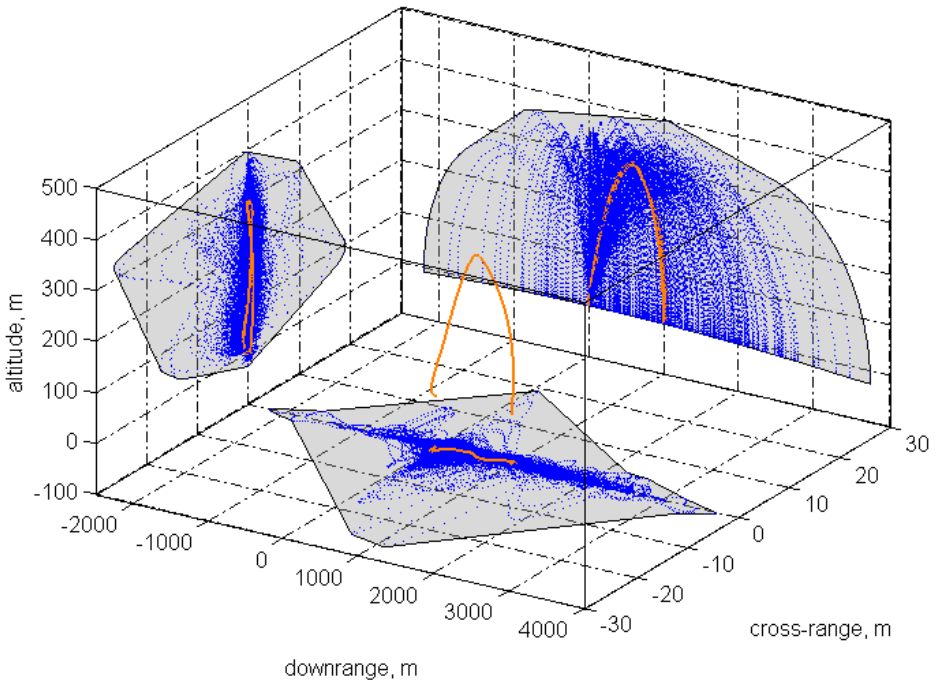


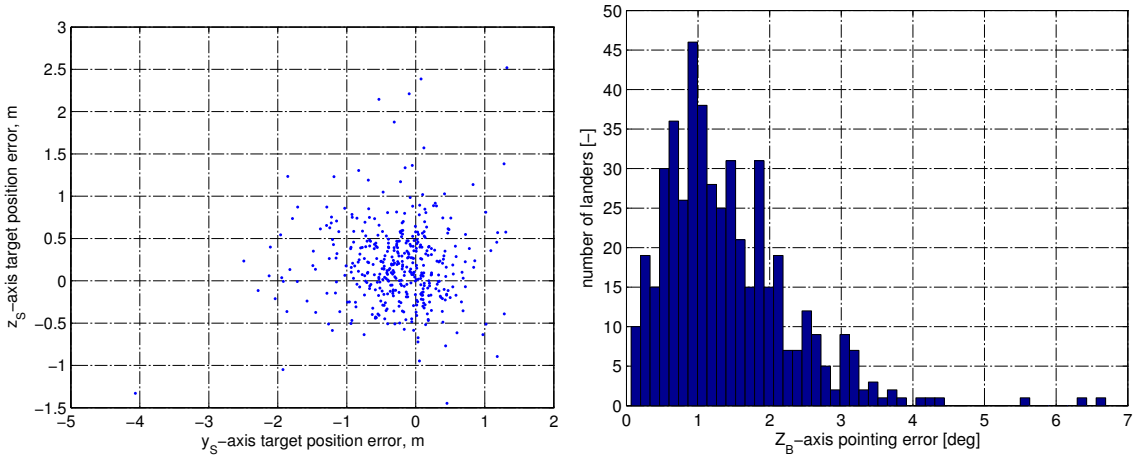
Figure 8-27: Nominal (orange) and divergent (blue) repositioning trajectories

This is a 24% increase with respect to the hybrid repositioning guidance system. The impact of the hazard avoidance system on the trajectory plot is not visible, because the maximal deviation from the initial target is smaller than the LIDAR field-of-view side length, which is about 76 m.

The fuel consumption as a function of the altitude requirement and the jump distance is shown in figures 8-24a and 8-24b. The fuel requirements are approximately proportional to the jump distance, and increase with for higher altitude requirements. In absolute numbers, the impact of altitude changes is much higher: a increase of h_{req} by 100 m costs about the same amount of fuel as doubling the jump distance from 1000 m to 2000 m.

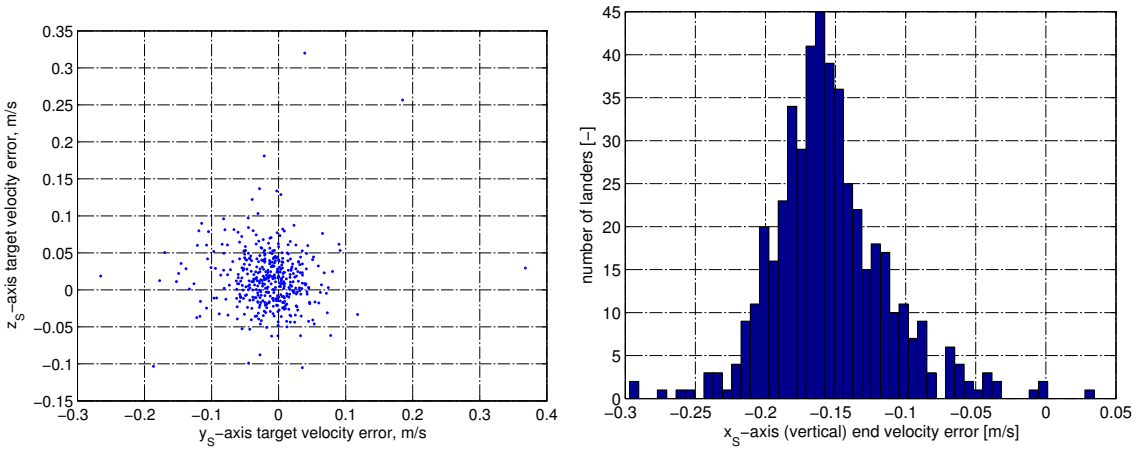
The repositioning trajectories for the successful landings are shown in Fig. 8-27. The cross-range errors are below 5 m for all cases. The maximal downrange shown here is in the order of 3 km, but this is purely a limit set by the combination of the nominal value jump distance and the chosen standard deviation; the lander is in fact able to cover a much larger range, with the available amount of fuel as only limiting factor. The available fuel mass for repositioning is about 20 kg (see Table 3-1), so the lander is able to execute three separate jumps of approximately 2 km.

The landing accuracy in the horizontal plane is shown in Fig. 8-28a. With mean downrange and cross-range errors of -0.27 m and 0.18 m, respectively, and a standard deviation of 0.61 m, the landing accuracy for the repositioning is slightly higher than for the GNC descent simulations. This is the consequence of lower overall velocity, the exclusion of the gravity turn guidance and the lower number of parameter variances. The same applies to the pointing errors illustrated in Fig. 8-28b with a mean value of 1.41° and a standard deviation of 0.9°.



(a) landing precision **(b)** pointing error

Figure 8-28: Landing accuracies and pointing errors for the quadratic repositioning system



(a) horizontal velocities **(b)** vertical velocity

Figure 8-29: Velocities at touchdown for quadratic repositioning system

The mean downrange and cross-range velocity errors are -0.02 ms^{-1} and 0.02 ms^{-1}, respectively, with a standard deviation of -0.05 ms^{-1}, see Fig. 8-29a. The mean vertical velocity error is -0.15 ms^{-1} with a 0.05 ms^{-1} standard deviation. These values are again lower than the corresponding values in the previous simulations.

It is possible that the range and light tracking instrument cannot be used during the repositioning phase. In that case, another instrument – or the LIDAR – must be able to produce relative position data. Tests show that a reduction of the position measurement frequency from 10 Hz to 0.5 Hz still has a very high success rate.

Chapter 9

Conclusions and Recommendations

The answer to the initial research question is, that it is possible to safely land and reposition a hopping lander on Enceladus with current technology.

The correct implementation of propagator in combination with the translational and rotational equations of motion was tested with `Simulink`. The integration error with the RK4 propagator is below 3 m even extreme case with a constant thrust of 100 N combined with increasing rotational velocities. The unit test of the gravity turn guidance logic is based on a lunar landing situation from literature and showed very good agreement. The remaining guidance logics were tested by comparing the simulation results with the expected results, which is in particular for the quadratic guidance and the quadratic repositioning guidance a valid method, because they represent a two-point boundary value problem. The PWPF modulator implementation is tested with results from literature, while the linear quaternion controller functionality is evaluated by observing the effects of the proportional and derivative gains on the lander dynamics and comparing the settling time and the steady-state errors with the expected values. The testing of the navigation system is a bit more complex, because its design is tailored to the Enceladus landing mission, but the state estimates were close the to the true states, and the filter tuning process indicated a normal EKF behavior.

The Monte Carlo simulations include variances of the lander's position, velocity, attitude, mass, moment of inertia, center of mass, thrust magnitude and alignment, quadratic guidance switch altitude, hop distance and altitude, and the estimations of position, velocity and attitude. This simulator setup allows for a realistic testing of the system performance in presence of various error sources.

The full descent GNC simulations have a failure rate of 10.6%. The mass distribution has been identified as the most critical factor – a center of mass shift of more than 0.05 m becomes dangerous in combination with higher thrust levels, because the perturbing moment caused by the main engine cannot be compensated with the comparatively weak attitude thrusters. This effect is even more obvious in the repositioning simulations using the ballistic guidance logic: The maximal thrust causes almost 70% of all lander model elements to fail; the critical center of mass shift here is 0.04 m. A reposition sequence based on quadratic guidance, on

the other hand, has a success rate of 98.2% due to smooth acceleration commands with much lower magnitudes.

The landing errors for the descent and the repositioning phase are below 0.5 m and 1 m, respectively, in most cases, with velocity errors in the order of 0.2 ms^{-1}. The pointing errors are generally below 5°, and highly depend on the interaction between the quadratic guidance logic and the velocity nullifying guidance during the last seconds before touchdown.

The hazard avoidance system re-targeted about 40% of lander during the repositioning simulations. This process was successful in all cases. Retargeting was not necessary during the descent phase, because the initial target site is reachable and free of hazardous surface elements.

The navigation system proved to be capable of finding good state estimations within seconds. A decrease of the measurement frequencies and the measurement accuracies does not have a significant effect on the mission success, in particular for the repositioning cycle.

The control system is fully capable of controlling the lander's attitude in all mission phases and respond errors, but it cannot handle larger rotational velocities due to the limitations of the linear quaternion controller.

The GNC subsystems all worked as intended. The reasons for the landing failures can all be identified, and are a consequence of large mass shifts inside the lander body, and large initial state estimation errors, which lead to problematic gravity turn guidance commands. With further adaptions it is expected, that the success rate can be increased even further. The gravity turn logic and the velocity nullifying logic were identified as the critical components – modifications to these systems will lead a higher reliability.

During the thesis work, some shortcomings of the Enceladus Simulator became obvious.

The simulation time increases drastically for large numbers of lander model elements. This increase is not linear, as might be expected: the full GNC descent simulations for 500 lander models take approximately 1.5 hour, while 1000 elements require more than 8 hours. This is not a problem of the state propagation or the high control and navigation frequencies, but probably of the simulator configuration data and result information stored in maps. A large number of map keys has a negative effect on the access time. This problem is not easy to solve at this point without completely rewriting the simulator configuration, which is a core function in the Enceladus Lander Simulator.

The definitions of the initial and target states in the Enceladus Lander Simulator are not straightforward, because they use both the inertial frame and the rotating frame, while the results are in fact returned with respect to the surface-fixed frame. A routine, that accepts state definitions in any frame would significantly improve the user-friendliness of the software package.

The hazard avoidance system is based on a true surface hazard map, that is generated with rock sizes and rock densities derived from single surface images. A more detailed topography model of Enceladus will increase the trustworthiness of the hazard avoidance system. Furthermore, the hazard avoidance system in its current state only uses a reachability and fuel consumption map to determine a new target spot. It is very interesting to investigate the effect of additional filters on the retargeting process.

The velocity nullifying guidance in its current form is not inherently stable, and can cause landing failures if not configured well. The initial idea was to design a very basic velocity nullifying guidance logic, because the effort to implement a feedback version would be disproportionate to the benefits. The time required to track the reasons for the landing failures and to update and change the logic parameters, however, was probably longer than the time needed to implement a explicit velocity nullifying guidance system from the start. This guidance logic currently works, but still is not trustworthy and should be updated.

The performance of the gravity turn is considerably lower, if the state estimations are too imprecise. Currently, the gravity turn guidance is activated at the same time of the first navigation system cycle, so the state estimates can diverge considerably from the true state. As first solution, the gravity turn acceleration command is recalculated after a few seconds, but there must a more stable solution. The gravity turn with constant recalculations of the required acceleration works fine in the isolated system simulations, but in combination with the full GNC system with added errors, this system proved to be unstable. The reasons for this are unclear at this point, but this might be an interesting subject for future work.

The lander has currently no means to control the rotation about the Z_B-axis, because the quaternion controller is only designed to align the main engine's thrust direction ($-Z_B$-axis) with the current thrust command. In the simulations it is assumed, that the LIDAR always aims at the target area during the scan process – regardless of the current lander orientation. The implementation of an additional controller for the $X_S - /Y_S -$axis pointing direction will give a more realistic hazard avoidance system, as it then imposes more requirements on the lander's state at the time of the image acquisition.

Appendix A

Simulation Configuration Files

This Appendix shows the basic lander and simulator configuration files. The former comprises the main lander data such as the initial and target state variables, whereas the latter is used to set up the simulator itself, which includes the propagator, the active disturbing forces and the Monte Carlo simulation options. The variable names in general are self-explanatory, and their exact meaning and influence is discussed in the corresponding chapters.

A-1 Lander Configuration File

This is the lander configuration file at the beginning of the G system simulations. All changes during the simulation process are discussed in the respective sections of Chapter 8.

```
# Lander configuration file reader for the NOMINAL LANDER

# Set the file name. This line is required in every configuration file.
fileName = lander1

# Initial lander mass [kg]
landerMass = 335

# Inertia tensor [kgm^2]
inertiaTensorXX = 150.0
inertiaTensorYY = 150.0
inertiaTensorZZ = 150.0
inertiaTensorXY = 0.0
inertiaTensorXZ = 0.0
inertiaTensorYZ = 0.0
xCOMshift = 0.0
yCOMshift = 0.0
zCOMshift = 0.0

# DESCENT: Initial true translational state vector [m,...,m/s] in I-frame
xPosition = 87180.71
yPosition = -10116.35
zPosition = -239527.02
xVelocity = 0.91
yVelocity = 62.60
zVelocity = -2.51
```

```
# DESCENT: Initial true rotational state vector [deg,...,rad/s] in I-frame
# Note: attitude is expressed in terms of Euler angles for a rotation from I-frame
# to B-frame, giving the transformation matrix T_I2B. (phi,theta,psi = X-,Y-,Z-axis)
eulerAnglePhi = -20.1234
eulerAngleTheta = 0.0
eulerAnglePsi = 90.0
rotationalRateX = 0.0
rotationalRateY = 0.0
rotationalRateZ = 0.0

# DESCENT: Target state vector [m,...,m/s,...,m/s^2] in R-frame
xPositionTarget = 86223.28
yPositionTarget = 0.0
zPositionTarget = -236896.51
xVelocityTarget = -0.17
yVelocityTarget = 0.02
zVelocityTarget = 0.47
xAccelerationTarget = 0.0
yAccelerationTarget = 0.0
zAccelerationTarget = 0.0

# REPOSITIONING: Initial true translational state vector [m,...,m/s] in I-frame
# Note: 1 m initial altitude was added, or the propagator ends program!
xPositionRepositioning = 86223.62015
yPositionRepositioning = 0.0
zPositionRepositioning = -236897.44939
xVelocityRepositioning = 0.0
yVelocityRepositioning = 4.576157
zVelocityRepositioning = 0.0

# REPOSITIONING: Initial true rotational state vector [deg,...,rad/s] in I-frame
# Note: attitude is expressed in terms of Euler angles for a rotation from I-frame
# to B-frame, giving the transformation matrix T_I2B. (phi,theta,psi = X-,Y-,Z-axis)
eulerAnglePhiRepositioning = -20.0
eulerAngleThetaRepositioning = 0.0
eulerAnglePsiRepositioning = 90.0
rotationalRateXRepositioning = 0.0
rotationalRateYRepositioning = 0.0
rotationalRateZRepositioning = 0.0

# REPOSITIONING: Target in S-frame
# hopDistance: along +Y_S-axis
# altitudeRequirement: altitude the lander has to reach at least
hopDistance = 1000.0
altitudeRequirement = 300.0

# Initial thrust and moment commands w.r.t. B-frame [N]
# Note: thrust of main engine must be positive (or zero)!
mainEngineThrust = 0.0
xThrustMoment = 0.0
yThrustMoment = 0.0
zThrustMoment = 0.0

# GUIDANCE SYSTEM
# Note: for guidance mode 4 only: conditions for switching (whatever happens earlier)
altitudeForSwitchToQuadratic = 2000.0
timeForSwitchToQuadratic = 300.0

# NAVIGATION SYSTEM
# Note: w.r.t. initial position R-frame) [m/s2, m/s2, m/s2]
constantDisturbingAccelerationX = 0.0
constantDisturbingAccelerationY = 0.0
constantDisturbingAccelertaionZ = 0.0

# CONTROL SYSTEM: varying parameters
# thrustMagnitude: output thrust = nominal thrust * thrustMagnitude[factor]
thrustMagnitude = 1.0

# CONTOL SYSTEM: Main engine data
```

```
# Note: Location w.r.t. B-frame [m], orientation w.r.t B-frame [rad] (Z->Y->X)
xPositionMainEngine = 0.0
yPositionMainEngine = 0.0
zPositionMainEngine = 0.0
eulerOrientationXmain = 0.0
eulerOrientationYmain = 0.0
eulerOrientationZmain = 0.0
# Characteristics:
# - thrust range [N]
# - specific impulse [s]
# - minimum impulse bit [Ns]
minimumThrustMainEngine = 15.5
maximumThrustMainEngine = 490.0
specificImpulseMainEngine = 312.0
# Switch engine off/on (for testing)
thrustForce = 1.0

# CONTROL SYSTE: Attitude thruster data
# Note: Location w.r.t. B-frame [m], orientation w.r.t B-frame [rad] (Z->Y->X)
# Note: Do NOT chance orientation without rewriting the thruster selection class!
# Nr. 1: x-axis, 1 out of 4
xPositionVernier1 = 0.0
yPositionVernier1 = -0.693
zPositionVernier1 = 0.0
eulerOrientationX1 = 0.0
eulerOrientationY1 = 0.0
eulerOrientationZ1 = 0.0

# Nr. 2: X_B-axis, 2 out of 4
xPositionVernier2 = 0.0
yPositionVernier2 = -0.693
zPositionVernier2 = 0.0
eulerOrientationX2 = 180.0
eulerOrientationY2 = 0.0
eulerOrientationZ2 = 0.0

# Nr. 3: X_B-axis, 3 out of 4
xPositionVernier3 = 0.0
yPositionVernier3 = 0.693
zPositionVernier3 = 0.0
eulerOrientationX3 = 180.0
eulerOrientationY3 = 0.0
eulerOrientationZ3 = 0.0

# Nr. 4: X_B-axis, 4 out of 4
xPositionVernier4 = 0.0
yPositionVernier4 = 0.693
zPositionVernier4 = 0.0
eulerOrientationX4 = 0.0
eulerOrientationY4 = 0.0
eulerOrientationZ4 = 0.0

# Nr. 5: Y_B-axis, 1 out of 4
xPositionVernier5 = 0.8
yPositionVernier5 = 0.0
zPositionVernier5 = 0.0
eulerOrientationX5 = 0.0
eulerOrientationY5 = 0.0
eulerOrientationZ5 = 0.0

# Nr. 6: Y_B-axis, 2 out of 4
xPositionVernier6 = 0.8
yPositionVernier6 = 0.0
zPositionVernier6 = 0.0
eulerOrientationX6 = 180.0
eulerOrientationY6 = 0.0
eulerOrientationZ6 = 0.0

# Nr. 7: Y_B-axis, 3 out of 4
xPositionVernier7 = -0.8
```

```
yPositionVernier7 = 0.0
zPositionVernier7 = 0.0
eulerOrientationX7 = 180.0
eulerOrientationY7 = 0.0
eulerOrientationZ7 = 0.0

# Nr. 8: Y_B-axis, 4 out of 4
xPositionVernier8 = -0.8
yPositionVernier8 = 0.0
zPositionVernier8 = 0.0
eulerOrientationX8 = 0.0
eulerOrientationY8 = 0.0
eulerOrientationZ8 = 0.0

# Nr. 9: Z_B-axis, 1 out of 4
xPositionVernier9 = 0.8
yPositionVernier9 = 0.0
zPositionVernier9 = 0.0
eulerOrientationX9 = 90.0
eulerOrientationY9 = 0.0
eulerOrientationZ9 = 0.0

# Nr. 10: Z_B-axis, 2 out of 4
xPositionVernier10 = 0.8
yPositionVernier10 = 0.0
zPositionVernier10 = 0.0
eulerOrientationX10 = -90.0
eulerOrientationY10 = 0.0
eulerOrientationZ10 = 0.0

# Nr. 11: Z_B-axis, 3 out of 4
xPositionVernier11 = -0.8
yPositionVernier11 = 0.0
zPositionVernier11 = 0.0
eulerOrientationX11 = -90.0
eulerOrientationY11 = 0.0
eulerOrientationZ11 = 0.0

# Nr. 12: Z_B-axis, 4 out of 4
xPositionVernier12 = -0.8
yPositionVernier12 = 0.0
zPositionVernier12 = 0.0
eulerOrientationX12 = 90.0
eulerOrientationY12 = 0.0
eulerOrientationZ12 = 0.0

# Characteristics:
# - thrust range [N]
# - specific impulse [s]
# - minimum impulse bit [Ns]
minimumThrustVerniers = 1.85
maximumThrustVerniers = 6.0
specificImpulseVerniers = 220.0
minimumImpulseVerniers = 0.0028

# Switch verniers off/on (for testing)
verniersForce = 1.0
```

A-2 Simulator Configuration File

This is the simulator configuration file at the beginning of the G system simulations. All changes during the simulation process are discussed in the respective sections of Chapter 8.

```
# Simulation configuration file reader. This file is unique.
```

```
# Set the file name. This name is used during in simulatorconfiguration.cpp.
fileName = lander1simulatorConfiguration

################################################################################
########## PART A.): Propagator setup ##########################################
################################################################################

# Maximal value that is considered equal to zero
toleranceForZero = 0.00001

# Integrator data [s]
simulationStartEpoch = 0.0
simulationEndEpoch = 700.0

# Choice of integrator for the true state: Select one of the following IDs
# 1.0 - RK4
# 2.0 - RKF45
# 3.0 - RKF56
# 4.0 - RKF78
integratorID = 1.0

# Activate rotating moon and variable mass model
rotatingReferenceFrame = 1.0
variableMassModel = 1.0

# STATE INTEGRATION
# Note: for fixted-step integrators: initialStepSize = fixed step size
minimumStepSize = 0.01
maximumStepSize = 2.0
relativeErrorTolerance = 0.001
absoluteErrorTolerance = 0.001
initialStepSize = 0.01

# Minimal altitude considered touchdown [m]
minimalAltitude = 252100.0

# MONTE CARLO OPTIONS
# Note: variance = squared standard deviation
# thrustAlignmentVariance - [deg], about all axes
# eulerAngleEstimateionVariance - [deg], about all axes
# thrustMagnitudeVariance - [%], deviation from nominal
landerNumber = 1000.0
velocityVariance = 20.0
positionVariance = 40000.0
eulerAngleVariance = 0.0
massVariance = 10.0
guidanceSwitchVariance = 40000.0
massMomentInertiaVariance = 56.25
COMshiftVariance = 0.0016
thrustMagnitudeVariance = 25.0
thrustAlignmentVariance = 0.25
positionEstimationVariance = 25.0
velocityEstimationVariance = 0.0
eulerAngleEstimationVariance = 0.0
hopDistanceVariance = 1000000.0
hopAltitudeReqVariance = 1000.0

# SAVE OPTIOINS
# Note: - initial and final results for all models always saved
# - if onlySimulateLanderNumer != 0, then no collectedResults and no
#         base lander results
#       - excludeFailLandings: only in collected position results; use indicator
#         in landingFailed.txt to remove invalid collected state results
# - maximal error numbers: refer to total error (vector norm)
saveResultsBaseLander = 1.0
saveResultsAllLander = 0.0
saveStateResultsAllLander = 1.0
saveResultsLanderNumber = 12.0
onlySimulateLanderNumber = 0.0
```

```
saveResultsStepSize = 4.0
excludeFailLandings = 1.0
maxVelocityErrorForFail = 2.0
maxPositionErrorForFail = 10.0

################################################################################
########## PART B.): GNC setup #################################################
################################################################################

# SIMULATION MODE
# 1.0 Descent: Guidance only
# 2.0 Descent: Guidance and control
# 3.0 Descent: Guidance, navigation and control
# 4.0 Repositioning: Guidance and control
# 5.0 Repositioning: Guidance, navigation and control
simulationMode = 1.0

# GNC UPDATE FREQUENCIES [Hz]
# Note: navigation frequency follows from highest instrument frequency
guidanceFrequency = 1.0
controlFrequency = 20.0

# Guidance modes, both descent and repositioning
# 1.0 Gravity turn only
# 2.0 Quadratic guidance only
# 3.0 Velocity nullifying only
# 4.0 First gravity turn, then quadratic, then velocity nullifying
# 5.0 Ballistic and quadratic guidance (mode must be 4 or 5)
#       6.0 Quadratic guidance repositioning (mode must be 4 or 5)
guidanceMode = 4.0

# GRAVITY TURN GUIDANCE LOGIC OPTIONS
gravityTurnConstantThrust = 1.0
velocityRatioSwitchToSphericalModel = 0.5
timeOfSingularUpdate = 5.0

# QUADRATIC GUIDANCE LOGIC OPTIONS
# Note: stepsize for numerical integration to find the fuel consumption is
#       [t2go under investigation]/[fuelConsumptionCalculationStepSizeFactor]
#
# Update options for the C-factors:
# -1.0 Use frequency of navigation system
#          value > 0 Set number of seconds between updates
#
# Update options for the underlying time-to-go:
# -1.0 Use instant of time of retargeting with hazard map
#    value > 0 Set number of seconds between updates
#
# Time-to-go search options:
# 1.0 a_target is zero
# 2.0 a_target as specified
# 3.0 a_target free (standard)
# Guidance options in case time-to-go search not successful
# 1.0 gravity turn (standard)
# 2.0 Q-guidance
# 3.0 no guidance
#
# YZaxisT2GoLead:
# how many seconds the Y_S- and Z_S-axis target states must be reached
# before the final state along the X_S-axis (avoid critical maneuvers
# just above the surface). If t2go < lead: lead set 0. 10s standard.
#
# minimalT2GoForCalculationCfactor:
# in case there is a YZ-lead, the C-factors for the Y_S-/Z_S-axes
#       will be 0 when the X_S-t2go is still large enough
#
# t2GoWhenQuadraticGuidanceEnds:
# if in mode 4 -> velocity nullifying guidance,
#       if in mode 2 -> no guidance
#
```

```
stepsizeTargetAccSearch = 0.01
fuelConsumptionCalculationStepSizeFactor = 50.0
timeBetweenCfactorsUpdate = 3.0
timeBetweenT2GoUpdate = -1.0
t2GoSearchOption = 3.0
guidanceOptionT2GoUnsuccessful = 3.0
YZaxisT2GoLead = 10.0
minimalT2GoForCalculationCfactors = 2.0
t2GoWhenQuadraticGuidanceEnds = 2.0

# VELOCITY NULLIFYING GUIDANCE LOGIC OPTIONS
#  Switch options:
# -1.0 Use instance of time where t2go = YZaxisT2GoLead
#    value > 0 Set number of seconds before touchdown.
#              Must be <= YZaxisT2GoLead and <= nullifyingDuration.
nullifyingDuration = 1.0
nullifyingActivationT2Go = -1.0
xSaxisTargetVelocity = -0.5

# BALLISTIC GUIDANCE OPTIONS
# Note: turn refers to the rotation from vertical to desired flight direction
durationLowThrustTurn = 20.0
percentageMaxThrustDuringTurn = 15.0
altitudeBallisticToQuadratic = 200.0

# HAZARD AVOIDANCE OPTIONS
isHazardAvoidanceActive = 1.0
scanAltitude = 290.0
resolutionOfTrueSHM = 1.0
VDFE = 9.0
reachabilityMapHazardThreshold = 0.9
originOfSHMinMapcoordinateX = 100.0
originOfSHMinMapcoordinateY = 2500.0
nominalTargetinSframeZ = 0.0
nominalTargetinSframeY = 0.0

# CONTROL OPTIONS
# PWPF controller options:
# 0.0 deactivated: moment_B = torque_ref from controller
# 1.0 activated: filter & trigger used, thrust limitations
proportionalGainQuaternionController = 1.5
derivativeGainQuaternionController = 3.5
PWPFactive = 1.0
PWPFfilterGain = 8.0
PWPFtimeConstant = 0.45
PWPFsamplingTime = 0.1
PWPFcutInFactor = 0.6
PWPFcutOutFactor = 0.1
PFPWfilterLimitFactor = 1.5

# NAVIGATION OPTIONS
# Changeable navigation parameters
frequencyIMU = 20.0
frequencyRangeInstrument = 10.0
frequencyStarSensors = 4.0
propagationStepSizeFactor = 0.1
processNoisePosition = 0.001
processNoiseVelocity = 0.0
processNoiseAttitude = 0.00001
processNoiseBiasAcc = 0.0
processNoiseBiasGyr = 0.0
covariancePosition = 1000.0
covarianceVelocity = 25.0
covarianceAttitude = 0.0004
covarianceBiasAcc = 0.01
covarianceBiasGyr = 0.01
biasAccelerometerXest = 0.0
biasAccelerometerYest = 0.0
biasAccelerometerZest = 0.0
biasGyroscopeXest = 0.0
```

```
biasGyroscopeYest = 0.0
biasGyroscopeZest = 0.0
scaleErrorAccelerometerXest = 0.0022
scaleErrorAccelerometerYest = -0.0063
scaleErrorAccelerometerZest = -0.0033
scaleErrorGyroscopeXest = -0.0011
scaleErrorGyroscopeYest = -0.0045
scaleErrorGyroscopeZest = -0.0066
misalignmentAccelerometerXYest = 0.009
misalignmentAccelerometerXZest = 0.022
misalignmentAccelerometerYXest = -0.027
misalignmentAccelerometerYZest = 0.0
misalignmentAccelerometerZXest = 0.0022
misalignmentAccelerometerZYest = -0.0066
misalignmentGyroscopeXYest = -0.0009
misalignmentGyroscopeXZest = 0.0022
misalignmentGyroscopeYXest = -0.0027
misalignmentGyroscopeYZest = 0.0044
misalignmentGyroscopeZXest = -0.0055
misalignmentGyroscopeZYest = -0.0056
#  IMU instrument model parameters
# NOTE: - random errors have a uniform distribution
#       - random error gyroscope: in terms of quaternion elements
biasAccelerometerX = -0.007
biasAccelerometerY = 0.006
biasAccelerometerZ = 0.005
biasGyroscopeX = -0.02
biasGyroscopeY = 0.03
biasGyroscopeZ = -0.04
scaleErrorAccelerometerX = 0.002
scaleErrorAccelerometerY = -0.007
scaleErrorAccelerometerZ = -0.003
scaleErrorGyroscopeX = -0.001
scaleErrorGyroscopeY = -0.005
scaleErrorGyroscopeZ = -0.006
misalignmentAccelerometerXY = 0.01
misalignmentAccelerometerXZ = 0.02
misalignmentAccelerometerYX = -0.03
misalignmentAccelerometerYZ = 0.0
misalignmentAccelerometerZX = 0.002
misalignmentAccelerometerZY = -0.006
misalignmentGyroscopeXY = -0.001
misalignmentGyroscopeXZ = 0.002
misalignmentGyroscopeYX = -0.003
misalignmentGyroscopeYZ = 0.004
misalignmentGyroscopeZX = -0.005
misalignmentGyroscopeZY = -0.006
randomNoiseAccelerometer = 0.00002
randomNoiseGyroscope = 0.00002
# Range instrument and star sensor model options
# Note: 1-sigma values; star sensor in terms of quaternion elements
measurementNoiseRange = 0.2
measurementNoiseStarS = 0.0001


################################################################################
########## PART C.): Force Options #############################################
################################################################################

# NAVIGATION SYSTEM: Choice of gravity fields
#  Note: 3rd body perturbation force directional vector in rotating RF from Enceladus
#             is fixed at [0;1;0] - always along y-axis. Change this if needed.
centralGravityFieldEnceladus = 1.0
centralGravityFieldSaturn = 1.0
J2GravityFieldSaturn = 1.0
J2GravityFieldEnceladus = 1.0
```

Bibliography

Acuna, M. H.: "Space-based magnetometers". *Review of Scientific Instruments* Vol. 73, No. 11, 2002, pp. 3717–3736.

Aeyels, D.: "Stabilization by smooth feedback of the angular velocity of a rigid body". *Systems & Control Letters* Vol. 6, No. 1, 1985, pp. 59–63.

Ampe, A., Bertels, F., Bos, R., Faber, A., Gijsen, H.P., Haagsma, A., Meerbeeck, v.M., Schollaart, V., and Vrebosch, T.: *METOPE - Mission to Enceladus for Terrain, Ocean and Plume Exploration*. Final Report - Version 1. 2009.

Antreasian, P. G., Bordi, J. J., Criddle, K. E., Ionasescu, R., Jacobson, R. A., Jones, J. B., MacKenzie, R. A., Meek, M. C., Pelletier, F. J., Roth, D. C., Roundhill, I. M., and Stauch, J.: "Cassini orbit determination performance during the first eight orbits of the Saturn satellite tour". *Astrodynamics 2005, Vol 123, Pts 1-3* Vol. 123, 2006, pp. 933–962.

Astolfi, A. and Rapaport, A.: "Robust stabilization of the angular velocity of a rigid body". *Systems & Control Letters* Vol. 34, No. 5, 1998, pp. 257–264.

Astrium: *EADS Astrium about the Linear Accurate Sun Sensor*. Website, available via `http://www.astrium.eads.net/en/equipment/liass.html`. Last checked: Oktober 2013. 2013.

Belló-Mora and Baeza-Martin, M.: "Moderate Lift/Drag Vehicle Reentry "RATT"". Final Report, Contract No. 9982/92/NL/JG, Work Order No. 8, GMVSA 2069/95, G.M.V., S.A., Madrid. 1995.

Bezanson, J.: *Understanding and Using Floating Point Numbers*. Website, available at `http://www.cprogramming.com/tutorial/floating_point/understanding_floating_point.html`. Last checked: May 2013. 2013.

Boyce, W. E. and DiPrima, R. C.: *Elementary differential equations and boundary value problems*. 8th. Hoboken, NJ: Wiley, 2005.

Brown, C.: *Elements of spacecraft design*. AIAA education series. Reston, VA: American Institute of Aeronautics and Astronautics, Inc., 2002.

Brown, R.C., Clark, R., et al.: "Composition and Physical Properties of Enceladus' Surface". *Science* Vol. 311, No. 5766, 2006, pp. 1425–1428.

Bruzzi, J.R., Strohben, K., Boone, B.G., Keren, S., Layman, R.S., and Noble, M.W.: "A Compact Laser Altimeter for Spacecraft Landing Applications". *John Hopkins APL Technical Digest* Vol. 30, No. 4, 2012, p. 15.

Cavallo, A. and Ferrara, F.: "Atmospheric re-entry control for low lift drag vehicles". *Journal of Guidance Control and Dynamics* Vol. 19, No. 1, 1996, pp. 47–53.

Chu, Q.P.: "Spacecraft Attitude Dynamics and Control II, AE4-305". Lecture Notes. 2012.

Citron, S. J., Dunin, S. E., and Meissinger, H. F.: "A Terminal Guidance Technique for Lunar Landing". *Aiaa Journal* Vol. 2, No. 3, 1964, pp. 503–509.

Cornelisse, J. W., Schöyer, H. F. R., and Wakker, K. F.: *Rocket propulsion and spaceflight dynamics*. London ; San Francisco: Pitman, 1979.

Coustenis, A., Atreya, S.K., Balint, T., Brown, R.H., Dougherty, M.K., Ferri, F, Fulchignoni, F., Gaultier, D., Gowen, R.A., Griffith, C.A., et al.: "TandEM: Titan and Enceladus mission". *Experimental Astronomy* Vol. 23, No. 3, 2009, pp. 893–946.

Crassidis, J. L., Markley, F. L., and Cheng, Y.: "Survey of Nonlinear Attitude Estimation Methods". *Journal of Guidance Control and Dynamics* Vol. 30, No. 1, 2007, pp. 12–28.

Davailus, G. and Newman, B.: "The Application of Quaternion Algebra to Gyroscopic Motion, Navigation, and Guidance". *AIAA Guidance, Navigation, and Control Conference and Exhibit*. AIAA 2005-5945. 2005.

ESA: *ESA science and technology*. Website, available at `http://sci.esa.int/science-e`. Last checked: March 2012. 2012.

Fehlberg, E.: *Classical fifth-, sixth-, seventh-, and eighth-order Runge-Kutta formulas with stepsize control*. Washington: NASA; for sale by the Clearinghouse for Federal Scientific and Technical Information, Springfield, Va., 1968.

Gibbs, Bruce P.: *Advanced Kalman Filtering, Least-Squares and Modeling*. John Wiley and Sons, Inc., 2011, p. 627.

Gregory, W., Ottinger, W., Robinson, M., Schmitt, H., and Lawrence, S. J., eds.: *Go for Lunar Landing: From Terminal Descent to Touchdown Conference Report*. Arizona State University, University of Arizona, Lunar and Planetary Institute. 2008.

Groves, P. D.: *Principles of GNSS, inertial, and multisensor integrated navigation systems*. GNSS technology and applications series. Boston: Artech House, 2008.

Guermah, S., Djennoune, S., and Bettayed, M.: "Controllability and observability of linear discrete-time fractional-order systems". *International Journal of Applied Mathematics and*

Computer Science Vol. 18, No. 2, 2008. 321TW Times Cited:11 Cited References Count:36, pp. 213–222.

Guo, G. N., Hawkins, M., and Wie, B.: "Waypoint-Optimized Zero-Effort-Miss/Zero-Effort-Velocity Feedback Guidance for Mars Landing". *Spaceflight Mechanics 2012* Vol. 143, 2012, pp. 1383–1402.

Guo, J. A. and Han, C. Y.: "Design of Guidance Laws for Lunar Pinpoint Soft Landing". *Astrodynamics 2009, Vol 135, Pts 1-3* Vol. 135, 2010, pp. 2133–2145.

Es-hagh, M.: "Step variable numerical orbit integration of a low earth orbiting satellite". *Journal of the Earth & Space Physics* Vol. 31, No. 1, 2005, pp. 1–12.

Hansen, C., Esposito, L., Stewart, A. I. F., Colwell, J., Hendrix, A., Pryor, W., Shemansky, D., and West, R.: "Enceladus' water vapor plume". *Science* Vol. 311, No. 5766, 2006, pp. 1422–1425.

Huang, X. Y. and Wang, D. Y.: "Autonomous navigation and guidance for pinpoint lunar soft landing". *2007 Ieee International Conference on Robotics and Biomimetics, Vols 1-5*, 2007, pp. 1148–1153.

Ingoldby, R.N.: "Guidance and Control System Design of the Viking Planetary Lander". *Journal of Guidance and Control* Vol. 1, No. 3, 1978, pp. 189–196.

Iserles, A.: *A first course in the numerical analysis of differential equations*. Cambridge texts in applied mathematics. Cambridge ; New York: Cambridge University Press, 1996.

Jacobson, R. A., Antreasian, P. G., Bordi, J. J., Criddle, K. E., Ionasescu, R., Jones, J. B., Mackenzie, R. A., Meek, M. C., Parcher, D., Pelletier, F. J., et al.: "The gravity field of the Saturnian system from satellite observations and spacecraft tracking data". *Astronomical Journal* Vol. 132, No. 6, 2006, pp. 2520–2526.

Jaumann, R., Stephan, K., Hansen, G. B., Clark, R. N., Buratti, B. J., Brown, R. H., Baines, K. H., Newman, S. F., Bellucci, G., Filacchione, G., et al.: "Distribution of icy particles across Enceladus' surface as derived from Cassini-VIMS measurements". *Icarus* Vol. 193, No. 2, 2008, pp. 407–419.

Johnson, A. E. et al.: "Lidar-based hazard avoidance for safe landing on Mars". *Journal of Guidance Control and Dynamics* Vol. 25, No. 6, 2002, pp. 1091–1099.

Johnson, A. E., Huertas, A., Werner, R. A., and Montgomery, J. F.: "Analysis of on-board hazard detection and avoidance for safe lunar landing". *2008 Ieee Aerospace Conference, Vols 1-9*, 2008, pp. 699–707.

Kargel, J. and Pozio, S.: "The Volcanic and Tectonic History of Enceladus". *Icarus* Vol. 119, No. 2, 1996, pp. 385–404.

Körner, T. W..: *Vectors, Pure and Applied - A General Introduction to Linear Algebra*. Cambridge University Press, 2012, p. 452.

Kos, L. D., Polsgrove, T. P., Sostaric, R. R., Braden, E. M., and Sullivan, J. J., eds.: *Altair Descent and Ascent Reference Trajectory Design and Initial Dispersion Analyses*. AIAA Guidance, Navigation, and Control Conference. 2010.

Krøvel, T.D.: "Optimal Tuning of PWPF Modulator for Attitude Control". MA thesis. Norwegian University of Science and Technology, 2005.

Kubota, T., Hashimoto, T., Kawaguchi, J., Uo, M., and Shirakawa, K.: "Guidance and Navigation of Hayabusa Spacecraft for Asteroid Exploration and Sample Return Mission". *SICE-ICASE, 2006. International Joint Conference*. 2006, pp. 2793–2796.

Kuipers, J. B.: *Quaternions and rotation sequences : a primer with applications to orbits, aerospace, and virtual reality*. Princeton, N.J.: Princeton University Press, 1999.

Lee, A. Y.: "Fuel-efficient Descent and Landing Guidance Logic for a Safe Lunar Touchdown". *AIAA Guidance, Navigation, and Control Conference*. AIAA 2011-6499. 2011.

Lee, A. Y., Ely, T., Strahan, A., Riedel, J., Ingham, M., Wincentsen, J., and Sarani, S.: "Preliminary Design of the Guidance, Navigation and Control System of The Altair Lunar Lander". *AIAA Guidance, Navigation, and Control Conference*. AIAA 2010-7717. 2010.

Liebe, C.C., Abramovici, A., Bartman, R.K., Bunker, R.L., Chapsky, J., Chu, Cheng-Chih, Clouse, D., Dillon, J.W., Hausmann, B., Hemmati, H., Kornfeld, R.P., Kwa, C., Mobasser, S., Newell, M., Padgett, C., Roberts, W.T., Spiers, G., Warfield, Z., and Wright, M.: "Laser radar for spacecraft guidance applications". *Aerospace Conference, 2003. Proceedings. 2003 IEEE*. Vol. 6. 2003. DOI: 10.1109/AERO.2003.1235190.

Mathews, J.H. and Fink, K.D.: *Numerical methods using MATLAB*. 3rd. 98048648 John H. Mathews, Kurtis D. Fink. ill. ; 25 cm. Rev. ed. of: Numerical methods for mathematics, science, and engineering. 2nd ed. 1992. Includes bibliographical references (p. 619-630) and index. Upper Saddle River, N.J.: Prentice Hall, 1999.

Mathworks: *6DoF Quaternion Block description, Aerospace Blockset R2013B Documentation*. Website, available via `http://www.mathworks.nl/help/aeroblks/6dofquaternion.html`. Last checked: July. 2014.

McCelland, R.S.: "Spacecraft attitude control system performance using pulse-width pulse-frequency modulated thrusters". MA thesis. Naval Postgraduate School, 1994.

McInnes, C. R.: "Nonlinear transformation methods for gravity-turn descent". *Journal of Guidance Control and Dynamics* Vol. 19, No. 1, 1996, pp. 247–248.

McKay, C. P., Porco, C. C., Altheide, T., Davis, W. L., and Kral, T. A.: "The Possible Origin and Persistence of Life on Enceladus and Detection of Biomarkers in the Plume". *Astrobiology* Vol. 8, No. 5, 2008, pp. 909–919.

Meriam, J. L., Kraige, L. G., and Palm, William J.: *Engineering mechanics*. 5th. New York: J. Wiley, 2002.

Mio: *Mio Technology Corporation about GPS receivers*. Website, available via `http://www.mio.com/technology-gps-accuracy.htm`. Last checked: Oktober 2013. 2013.

Montenbruck, O. and Gill, E.: *Satellite Orbits : Models, Methods, and Applications*. Berlin New York: Springer, 2000.

Mooij, E.: *The motion of a vehicle in a planetary atmosphere*. Delft: Delft University Press, 1997.

Mooij, E. and Chu, Q.: "IMU/GPS integrated navigation system for a winged re-entry vehicle". *AIAA Guidance, Navigation, and Control Conference*. AIAA 2001-4169. 2001.

– *Tightly-Coupled IMU/GPS Re-Entry Navigation System*. Tech. rep. AIAA 2002-5005. 2002.

NASA: *HAYABUSA LIDAR Instrument Information*. Website, available via `http://starbrite.jpl.nasa.gov/pds`. Last checked: February 2012. 2012.

– *NASA image photo journal*. Website, available at `http://www.jpl.nasa.gov/spaceimages`. Last checked: March 2012. 2012.

– *Planetary fact sheets*. Website, available at `http://nssdc.gsfc.nasa.gov/planetary/planetfact.html`. Last checked: February 2013. 2013.

Parkes, S. M. and Silva, V.: "GNC sensors for planetary landers - a review". *Data Systems in Aerospace*. Ed. by R. A. Harris. Vol. 509. ESA Special Publication. July 2002.

Parkinson, C., Liang, M.-C., Yung, Y. L., and Kirschivnk, J. L.: "Habitability of Enceladus: Planetary Conditions for Life". *Origins of Life and Evolution of Biospheres* Vol. 38, 4 2008, pp. 355–369.

Parreira, B., Sotto, E.D., Caramagno, A., and Rebordäo, J.: "Hazard avoidance for planetary landing: GNC design and performance assessment". *7th International ESA Conference on Guidance, Navigation and Control Systems*. 2008, p. 14.

Pater, I. D. and Lissauer, J. J.: *Planetary sciences*. Cambridge University Press, 2001.

Pisacane, V. L.: *Fundamentals of space systems*. 2nd. The Johns Hopkins University/Applied Physics Laboratory series in science and engineering. Oxford ; New York: Oxford University Press, 2005.

Porco, C., Helfenstein, P., Thomas, P. C., Ingersoll, A. P., Wisdom, J., West, R., Neukum, G., Denk, T., Wagner, R., Roatsch, T., et al.: "Cassini observes the active South Pole of Enceladus". *Science* Vol. 311, No. 5766, 2006, pp. 1393–1401.

Press, W. H., Teukolsky, S.A., Vetterling, W.T., and Flannery, B.P.: *Numerical Recipes in C++. The Art of Scientific Computing*. 2nd. Cambridge: Cambridge University Press, 2002, xxviii, 1002 p.

Regan, F. J.: *Re-entry vehicle dynamics*. AIAA education series. New York, N.Y.: American Institute of Aeronautics and Astronautics, 1984.

Ribeiro, M. I.: *Kalman and Extended Kalman Filters: Concept, Derivation and Properties*. Tech. rep. Available at `http://users.isr.ist.utl.pt/~mir/pub/kalman.pdf`. Last checked: December 2011. Institute for Systems and Robotics, Instituto Superior Técnico, 1049-001 Lisboa, 2004.

Roatsch, Th., Jaumann, R., Stephan, K., and Thomas, P. C.: *Cartographic Mapping of the Icy Satellites Using ISS and VIMS Data*. Springer Netherlands, 2009, pp. 763–781.

Russell, R. P. and Lara, M.: "On the design of an Enceladus science orbit". *Acta Astronautica* Vol. 65, No. 1-2, 2009, pp. 27–39.

Sinclair, A. J. and Fitz-Coy, N. G.: "Comparison of obstacle avoidance strategies for Mars landers". *Journal of Spacecraft and Rockets* Vol. 40, No. 3, 2003, pp. 388–395.

Sostaric, R. R. and Rea, J. R.: "Powered Descent Guidance Methods For The Moon and Mars". *AIAA Guidance, Navigation, and Control Conference and Exhibit*. AIAA 2005-6287. 2005.

Spahn, F., Schmidt, J., Albers, N., Horning, M., Makuch, M., Seiss, M., Kempf, S., Srama, R., Dikarev, V., Helfert, S., et al.: "Cassini Dust Measurements at Enceladus and Implications for the Origin of the E Ring". *Science* Vol. 311, No. 5766, 2006, pp. 1416–1418.

Spencer, J. R., Pearl, J. C., Segura, M., Flasar, F. M., Mamoutkine, A., Romani, P., Buratti, B. J., Hendrix, A. R., Spilker, L. J., and Lopes, R. M. C.: "Cassini Encounters Enceladus: Background and the Discovery of a South Polar Hot Spot". *Science* Vol. 311, No. 5766, 2006, pp. 1401–1405.

Squyres, S., Reynolds, R., and Cassen, P.: "The Evolution of Enceladus". *Icarus* Vol. 53, No. 2, 1983, pp. 319–331.

Stewart, J.: *Calculus - Early Transcendentals*. Princeton, N.J.: Thomson Brooks/Cole, 2003.

Stone, E. C. and Miner, E. D.: "Voyager 2 Encounter with the Saturnian System". *Science* Vol. 215, No. 4532, 1982, pp. 499–504.

Terejanu, G.A.: *Extended Kalman Filter Tutorial*. Tech. rep. Available at `http://users.ices.utexas.edu/~terejanu/files/tutorialEKF.pdf`. Last checked: December 2011. Department of Computer Science and Engingeering, University of Buffalo, 2004.

Terrile, R. J. and Cook, A. F.: "Enceladus: Evolution and Possible Relationship to Saturn's E-Ring". *LPI Contributions* Vol. 428, 1981, p. 10.

Teunissen, P.J.G., Simons, D.G., and Tiberius, C.C.J.M: "Probability and Observation Theory, AE2-E01". Lecture Notes. 2006.

Thomas, P. C., Burns, J. A., Helfenstein, R., Squyres, S., Veverka, J., Porco, C., Turtle, E. P., McEwen, A., Denk, T., Giese, B., et al.: "Shapes of the saturnian icy satellites and their significance". *Icarus* Vol. 190, No. 2, 2007, pp. 573–584.

Török, J. S.: *Analytical mechanics : with an introduction to dynamical systems*. New York: Wiley, 2000.

Vallado, D.A. and McClain, W.D.: *Fundamentals of astrodynamics and applications*. 2nd. Space technology library. Dordrecht; Boston: Kluwer Academic Publishers, 2001.

VECTRONIC: *Star Tracker VST-41M information sheet*. Website, available at `http://www.vectronic-aerospace.com/space.php?p=starsensor`. Last checked: February 2012. 2012.

Wakker, K.F.: "Astrodynamics-I, AE4-874 Part I". Lecture Notes. 2010.

Welch, G. and Bishop, G.: "An Introduction to the Kalman Filter". Department of Computer Science, University of North Carolina at Chapel Hill. 2006.

Wertz, J. R. and Larson, W. J.: *Space mission analysis and design*. 3rd. Space technology library. El Segundo, Calif. Dordrecht ; Boston: Microcosm ; Kluwer, 1999.

Wertz, J. R., Meissinger, H. F., Newman, L. K., and Smit, G. N.: *Orbit & Constellation Design & Management*. 2nd. Microcosm Press and Springer New York, 2009.

Wie, B.: *Space vehicle dynamics and control*. AIAA education series. Reston, VA: American Institute of Aeronautics and Astronautics, 1998.

– *Space vehicle dynamics and control*. 2nd. AIAA education series. Reston, VA: AIAA, 2008, xvii, 950 p.

Wie, B., Weiss, H., and Arapostathis, A.: "Quarternion feedback regulator for spacecraft eigenaxis rotations". *Journal of Guidance, Control, and Dynamics* Vol. 12, No. 3, 1989, pp. 375–380.

Wong, E. C., Singh, G., and Masciarelli, J. P.: "Autonomous guidance and control design for hazard avoidance and safe landing on Mars". *AIAA Atmospheric Flight Mechanic's Conference*. AIAA 2002-4619. 2002.

Zarchan, P. and Musoff, H.: *Fundamentals of Kalman Filtering - A Practical Approach, Second Edition*. 2nd. American Institute of Aeronautics and Astronautics, 2005.