Performance of a Flush Airdata Sensor in a Particle Filter-Based Re-entry Navigation System

Thesis Assignment

J. Rijnsdorp

Technische Universiteit Delft

TU Delft

Performance of a Flush Airdata Sensor in a Particle Filter-Based Re-entry Navigation System

Thesis Assignment

by

J. Rijnsdorp

in partial fulfillment of the requirements for the degree of

Master of Science
in Aerospace Engineering

at the Delft University of Technology,
to be defended publicly on Wednesday May 10, 2017 at 1:30 PM.

Student number:	4140427	
Project supervisor:	Dr. Ir. E. Mooij,	TU Delft
Thesis committee:	Prof. Dr. Ir. P. N. A. M. Visser,	TU Delft
	Ir. P. P. Sundaramoorthy,	TU Delft

An electronic version of this thesis is available at http://repository.tudelft.nl/.
Cover image: http://i.space.com/images/i/000/010/869/original/space-shuttle-atlantis-flip-ocean.jpg?1310420924

TU Delft Delft University of Technology

Acknowledgements

Hereby I would like to take the opportunity to thank my supervisor Erwin Mooij for his time, feedback and motivation. During the past year I have learned a lot, not only concerning the research topic at hand, but also in planning and organizing, the approach of a large project, and programming in the MATLAB/Simulink environment.

Furthermore, I would like to state my gratitude towards my friends for their support, as well as for keeping me company in the library. Even though they did not always understand the problems I encountered during this project, they tried to help anyway, which actually lead to some great ideas I had not yet come up with myself.

Finally, I would like to thank my brother for taking the time to read the draft version of this report to provide feedback, and my parents for their support during my studies and the support they have given me during all years of education, whether I was in elementary school, in high school or studying at Delft University of Technology. Without their aid I would not be at this point and I would not be so eager to complete my studies at the Faculty of Aerospace Engineering.

J. Rijnsdorp
Schiedam, April 2017

Abstract

Re-entry is a term that is applied when one would like to transport some (human) payload from an orbit around a certain body towards the surface of that body. With an entry velocity of multiple km/s, having the vehicle carrying this payload not smashed into pieces at the end of flight is one of the key points of re-entry. Using a Guidance, Navigation, and Control (GNC) system, applied to a winged re-entry vehicle, one is capable of defining a path through the atmosphere towards a predefined target, and following that path so that a safe atmospheric entry can be assured. Many methods exist for the definition of such a GNC system, and in this thesis methods concerning the navigation and control module are researched, with a simple attitude profile serving as a guidance system. For the navigation system, the application of the Flush Airdata Sensing system (FADS) is investigated, as well as the use of an Unscented Kalman Filter (UKF) and a Particle Filter (PF) for state estimation. The research question can therefore be defined as

"To what extent can the performance of an estimator-based navigation module for an unpowered, winged re-entry vehicle be optimized between the atmospheric entry point and the Terminal Area Energy Management (TAEM) interface?"

and is answered using four subtopics: the design of a re-entry simulator, the development of a control system, the development of a navigation module and the integration of these components.

The re-entry simulator was designed and developed as a baseline of the research; it was capable of simulating the flight from the atmospheric entry point to the TAEM interface of the HORUS-2B vehicle. This was done using a set of equations of motion that were integrated over time. The gravitational field was approximated up until the second order (thus including the J_2-effect) and the US76 atmospheric model was used to approximate the atmospheric characteristics. For the aerodynamics of the vehicle, tabulated coefficients were used, which were interpolated using the air data variables. In terms of the control system, it was chosen to use a Linear Quadratic Regulator (LQR) controller in combination with the predefined actuator system of HORUS. The controller was integrated into the simulator, and it was seen that using the attitude profile of the guidance system, the vehicle was steered according to this profile by the control system.

As for the navigation module, the Inertial Measurement Unit (IMU), Global Positioning System (GPS), and FADS sensors were implemented, together with the Extended Kalman Filter (EKF), UKF, and PF estimators. The measurement updates were done with a frequency of 10 Hz, and in between the IMU, which measured with 200 Hz, was used to propagate the state using Euler integration. The entry flight was then divided into three phases, namely (I) from entry to blackout, (II) the blackout phase itself, and (III) from blackout to the Terminal Area Energy Management (TAEM) interface. During phase II no GPS and FADS measurements were available and the state estimation was solely performed by IMU propagation. Results of phase I showed that the PF and UKF outperformed the EKF in terms of accuracy, however, the PF induced a rather large computational load. The output of phase I was used as an input for the blackout phase, and it was seen that the state propagation during this phase was sensitive to this input, as no measurement corrections were done. The results obtained from phase III then showed that the state estimation error induced in phase II was corrected easily and all filters were able to perform proper state estimation. The FADS was proven to increase position estimation accuracy, due to the strong dependence of the pressure measurements on the altitude of the vehicle. Again, the PF and UKF outperformed the EKF, due to their capability of using nonlinear equations.

Finally, a trade-off was made in which all results were discussed. The UKF was discarded due to error peaks in the attitude estimation caused by nonlinear transformation of the sigma points for the carrier phase difference measurements. The PF experienced similar peaks, but they were deemed negligible by increasing the number of particles. The computational loads were considered to be insignificant as the computation time did not exceed the real-life flight time, and therefore it was chosen to use the PF in the integrated simulator. The FADS was added to the Inertial Navigation System (INS)/GPS baseline, as it was proven to increase estimator accuracy. Furthermore, it was shown that for a sudden GPS outage of finite time, the system is able to recover the drifted state estimation error from this period, also in lower parts of the atmosphere.

As a conclusion, it was stated that the introduction of the FADS and the use of a PF increased performance of the state estimation process, leading to a mean altitude estimation error of 0.62 m and a mean velocity

estimation error of 0.021 m/s. The roll, pitch and yaw angles were estimated up until an error of 0.012, 0.013 and 0.0089 degrees respectively. Next to that, it was found that the PF is able to estimate GPS clock bias and drift to 0.59 m and 0.025 m/s respectively, whereas the IMU errors were approximated with an accuracy of 10^{-6} m/s^2 for the accelerometer biases, 10^{-6} rad/s for the gyroscope drifts and 10^{-6} for the accelerometer scaling factors. Comparison with the UKF showed that the attitude estimation results for the PF were more accurate, whereas the UKF was able to estimate the altitude and velocity with higher accuracy. The addition of the FADS affected mostly the altitude estimation due to the high dependency of the pressure measurements on the altitude of the vehicle, thereby increasing the altitude estimation accuracy from 0.93 m to 0.62 m.

After testing the integrated GNC-system, it was found that increasing the number of used GPS satellites increases the accuracy of the state estimates, however, this increase is very small. Lowering the frequency of the state propagation has a negative effect during the blackout phase, as the state errors drift further from the true state values. In the phases where the filter is activated, a lower state-propagation frequency increases performance, as the state error has less possibility to drift in between two measurement updates. The reduction of the frequency of the measurement updates then was found to have a negative effect on the performance of the system.

Contents

List of Abbreviations

Abbreviation	Description
AADS	Ascent Air Data System
CDF	Cumulative Distribution Function
CoM	Center of Mass
CRV	Crew Return Vehicle
EKF	Extended Kalman Filter
FADS	Flush Air Data Sensor
DOP	Dilution of Precision
DOF	Degree of Freedom
EoM	Equations of Motion
GDOP	Geometric Dilution of Precision
GNC	Guidance, Navigation, and Control
GPS	Global Positioning System
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
ISS	International Space Station
LKF	Linear Kalman Filter
LLA	Latitude, Longitude, Altitude
LQR	Linear Quadratic Regulator
MATLAB	Matrix Laboratory
MRP	Modified Rodrigues Parameter
MSBLS	Microwave Scan Beam Landing System
NASA	National Aeronautics and Space Administration
ODE	Ordinary Differential Equation
PDF	Probability Distribution Function
PF	Particle Filter
PRNG	Pseudo Random Number Generator
RK	Runge-Kutta
RCS	Reaction Control System
SIR	Sampling Importance Resampling
SSTO	Single Stage To Orbit
STS	Space Transportation System
TACAN	Tactical Air Navigation
TAEM	Terminal Aera Energy Management
UKF	Unscented Kalman Filter
US76	United States Standard Atmosphere of 1976
UT	Unscented Transform

List of Symbols

Symbol	Description	Unit
A	Area	m^2
a	Speed of sound	m/s
a	Temperature gradient	K/m
$\mathbf{a}$	Acceleration	m/s^2
$\mathbf{b}_a$	Accelerometer bias	m/s^2
b_c	Clock bias	m
b_{ref}	Reference length	m
$\mathbf{b}_\omega$	Gyroscope drift	rad/s
C_D	Drag coefficient	-
C_L	Lift coefficient	-
C_S	Side-force coefficient	-
C_l	Roll-moment coefficient	-
C_m	Pitch-moment coefficient	-
C_n	Yaw-moment coefficient	-
c	Speed of light	m/s
c_{ref}	Reference chord length	m
D	Drag force	N
d_c	Clock drift	m/s
E	Energy	J
$\mathbf{F}$	Force	N
f	Frequency	Hz
G	Gravitational parameter	$m^3/(kg \cdot s^2)$
$\mathbf{g}$	Gravitational acceleration	m/s^2
h	Geopotential altitude	m
I	Moment of inertia	kg/m^2
J_2	Jeffery constant for 'flattening' of the Earth	-
K	Gain	-
$\mathscr{L}$	Aerodynamic roll moment	N·m
L	Lift force	N
$\mathscr{M}$	Aerodynamic pitch moment	N·m
M	Mach number	-
M_m	Molar mass	kg/mol
$\mathbf{M}$	Moment	N m
m	Mass	kg
$\mathscr{N}$	Aerodynamic yaw moment	N·m
N	Number of PF particles	-
N_A	Avogadro's constant	1/mol
p	Pressure	N/m^2
p	Roll rate	rad/s
Q	Quaternion	-
$\overline{q}$	Dynamic pressure	N/m^2
q	Pitch rate	rad/s
q_c	Dynamic impact pressure	N/m^2
R	Distance	m
R	Specific gas constant	J/(kg K)
R_E	Earth's equatorial radius	m
r	Distance	m

Symbol	Description	Unit
r	Yaw rate	rad/s
S	Sideforce	N
$\mathbf{S}_a$	Accelerometer scaling and misalignment matrix	-
S_{ref}	Reference area	m^2
$\mathbf{S}_\omega$	Gyroscope scaling and misalignment matrix	-
T	Temperature	K
t	Time	s
U	Gyroscope error particles	rad/s
$\mathbf{u}$	Control vector	*various*
u	Gyroscope error weights	-
V	Velocity norm	m/s
V	Velocity vector particles	m/s
$\mathbf{V}$	Velocity vector	m/s
v	Velocity vector weights	-
W	Accelerometer error particles	*various*
w	Accelerometer error weights	-
$\mathbf{X}$	Position vector	m
X	Position vector particles	m
x	Position vector weights	-
$\mathbf{x}$	State vector	*various*
Y	Attitude vector particles	rad
y	Attitude vector weights	-
Z	GPS error particles	*various*
z	Geometric altitude	m
z	GPS error weights	-
$\mathbf{z}$	Measurement vector	*various*
α	Angle of attack	rad
β	Angle of sideslip	rad
Γ_{xy}	Pressure difference	N/m^2
γ	Flight-path angle	rad
γ	Specific heat ratio	-
δ	Latitude	rad
δ_a	Aileron deflection	rad
δ_b	Body-flap deflection	rad
δ_e	Elevator deflection	rad
δ_r	Rudder deflection	rad
ϵ	Elevation angle	rad
ϵ	FADS calibration factor	-
η_{SIR}	Resampling percentage	-
θ	Pitch angle	rad
λ	Clock angle	rad
μ	Gravitational parameter	m^3/s^2
ρ	Density	kg/m^3
$\tilde{\rho}$	Pseudorange	m
$\dot{\tilde{\rho}}$	Range rate	m/s
σ	Bank angle	rad
τ	Longitude	rad
Φ	Carrier phase	rad
ϕ	Roll angle	rad
ϕ	Cone angle	rad
χ	Heading angle	rad
ψ	Yaw angle	rad
ω_{cb}	Angular rate of central body	rad/s
$\boldsymbol{\omega}$	Rotational velocity	rad/s

1 Introduction

In April of 1981, a breakthrough was noted in the history of space flight, as the first successful orbital tests of the Columbia were conducted. After years of development, this STS-1 mission was completed, leading to an era that was dominated by this so-called Space Transportation System (STS), or, the Space Shuttle. Previous programs, such as Apollo and Gemini, were designed to deliver and deploy their payload, as stated by Goodman (2011). The STS missions were defined to be unique in the sense that they prioritized the rendezvous aspects of the mission, thereby switching from the use of capsules to a lift-generating vehicle. Over the thirty following years, many more missions were to follow for the Space-Shuttle program, up until the retirement in July of 2011. A total of 133 successful re-entry flights could be noted, thereby proving the success and value of this program.

The aforementioned mission aspect of re-entry of a vehicle is a process that is defined by extreme environmental and vehicle related conditions. Passing from a low-Earth orbit towards the surface of the Earth, a re-entry vehicle passes through a large set of flight regimes (Ragsdale, 1985), through all layers of the Earth's atmosphere, thereby experiencing Mach numbers between 27 during flight to 0.25 at landing for winged re-entry vehicles (Dean and Mena, 1983). Due to these velocities the aerodynamic heating in the lower parts of the atmosphere is large, and due to the decelerations that have to take place to ensure that the vehicle has a safe flight to the surface large structural loads are induced as well. Decelerating and safely reaching the surface is one aspect of vehicle re-entry, however, reaching the specified target in the form of a runway is another aspect of high importance; one does not want to have the vehicle end up somewhere in an ocean, or maybe worse, somewhere in the middle of a densely populated area. To ensure the controllability of a winged re-entry vehicle, a GNC-system (Guidance, Navigation, and Control system) is applied, defining where the vehicle is and where it is currently going, where it has to go, and what actions it has to perform to follow that desired trajectory.

For definition of the current position and velocity of the vehicle, the navigation module is used, which estimates the state that is to be compared to the guidance system to provide information for the control system. The latter computes the actions that are to be taken by the vehicle itself. This autonomous system is highly dependent on the measurements of on-board sensors, as these are able to extract information from both close (such as pressure of the surrounding atmosphere) and far (such as the relative position with respect to a satellite or star) external assets. In the STS program, Inertial Measurement Units (IMUs) were used in combination with the Tactical Air Navigation (TACAN) unit, which measured the slant range and magnetic bearing from the vehicle to a specified ground station (Wang and Bochsler, 1987). Additionally, an air data system that computed air pressure (Hillje and Nelson, 1981), and a Microwave Scan Beam Landing System (MSBLS) was incorporated, providing measurements of slant range as well as azimuth angles and elevation angles. Ragsdale (1985) added that the use of the Global Positioning System (GPS) in the navigation module of the Space Shuttle would suffice for even more accurate state estimation.

Incorporation of different sensors would provide a vehicle with information about the local and global environment, but is often not sufficient for direct estimation of a desired vehicle state. In terms of position, the vehicle is for example able to measure the distance between itself and a specific beacon such as a GPS satellite, but not its own location coordinates. A mathematical algorithm, often denoted as an estimator, is therefore to be used to relate the measurements to the desired state variables. Montez and Madden (1982) discussed that the Space Shuttle navigation module contained a so-called Extended Kalman Filter (EKF) that computed a

six-element state of the vehicle using measurements from the aforementioned STS sensors. It was concluded that this approach for the design of a re-entry vehicle navigation system was validated and that performance enhancement had to be included in further research, to increase the accuracy and robustness of the system. In different studies, such as Gross et al. (2010) and Wendel et al. (2010), multiple estimators were considered with respect to a theoretical system with IMU and GPS sensors (INS/GPS), but not applied to the actual re-entry vehicle models.

Montez and Madden (1982) stated that research concerning the performance of the STS-navigation system had to be done, whereas Ragsdale (1985) implied that the use of GPS would definitely improve accuracy and trustworthiness of the module. Mooij and Chu (2002) then conducted research on the application of an EKF-based navigation module containing only the INS/GPS integrated system for a flight between atmospheric entry and the TAEM interface, concluding that the combination of these elements produced acceptable results. It was described that the Space Shuttle also used an air data sensor, which was applied mostly for the flight regimes where the Mach number was deemed small (Hillje and Nelson, 1981). Post-processing of flight results showed two difficulties for this sensor; the operational range was only small in terms of the flight regime that was discussed in Mooij and Chu (2002), and the sensor showed large deviations in Mach number estimation. An innovative air-data system, the Flush Airdata Sensing system (FADS), was introduced by Ellsworth and Whitmore (2007), where it was applied to the X-34 space plane wind tunnel tests. This sensing system contained a built-in estimator that was evaluated in Whitmore and Moes (1994), but was not included in an estimator-based navigation module. This, together with the performance enhancement of the integrated navigation module, using different estimators, defines the gap of information that is to be filled using outcomes of this thesis, resulting in the following research goal:

"To what extent can the performance of an estimator-based navigation module for an unpowered, winged re-entry vehicle be optimized between the atmospheric entry point and the TAEM interface?"

This research question is addressed in this thesis, where it is divided into multiple parts, each assessing a different aspect of the research at hand.

- **Design and development of the re-entry flight simulator** - the dynamics and environmental aspects of the entry flight have to be evaluated thoroughly, resulting in a simulator that is capable of simulating the flight of a specified winged re-entry vehicle between the atmospheric entry point and the TAEM interface.

- **Development and simulation of the control system** - to ensure that the simulated re-entry vehicle is able to perform a controlled flight that is driven by a guidance profile, a controller and an actuator system are to be developed and implemented in the simulator, after which the performance of this system is evaluated.

- **Development and simulation of the navigation module** - as described in this introduction, the focus of this thesis lies on the integration of the FADS system into the navigation module and the performance improvement of this module due to the use of different estimators. In the development and simulation of the navigation system, these elements are evaluated individually, after which the system as a whole is integrated in the simulator.

- **Simulation of the integrated system** - this final step defines the performance of the integrated control and navigation system, with the newly implemented FADS system.

In the final part of this thesis, the development and integration of a GNC-system is described, where the guidance profile is obtained directly from a study done by Mooij and Chu (2002), and consists only of an attitude profile. The final simulator will contain the properties of the HORUS-2B reference vehicle, which motion through the atmosphere is approximated by a form of the Equations of Motion (EoM), which are integrated over time, using models for the gravitational field, planetary shape, atmosphere and aerodynamics of the vehicle. With the integrated control system and the retrieved guidance profile, the simulator is able to perform controlled flight, *i.e.*, it is capable of directly processing the commands provided by the guidance system and steering the vehicle in such a way that this guidance profile is approximated. The integrated navigation module finally will complete the GNC-system by accurately estimating the state of the vehicle, which will be used as input for the control system.

This navigation module serves as the most important system during this project, as the research question is formulated with respect to this system. To investigate the performance enhancement that is achieved by implementation of the FADS, comparisons will be made between a base configuration and an extended configuration, with the INS/GPS combination being the former and the INS/GPS/FADS combination being the latter for the sensor module. The built-in FADS estimator is used for comparison with the integrated estimator-based system, to research the performance improvement that is made by applying the raw FADS measurements to an estimator. Finally, for the estimator module of the navigation system, the EKF is defined as the baseline design, as this filter was also used in Mooij and Chu (2002) and has proven its trustworthiness on the STS flights (Montez and Madden, 1982). Performance optimization in terms of the estimators is done by comparison of this baseline with two extended models: the Unscented Kalman Filter (UKF) and the Particle Filter (PF), which are all evaluated on estimation accuracy, computational load, and capability of recovery after sensor failure.

In this report an overview of the theory that was used to build the simulator is provided, and the design process is thoroughly discussed. Furthermore, the implementation and verification of the different navigation system components is described and the obtained results are discussed. First, in Chapter 2 the mission heritage is discussed, and the mission and system requirements are formulated. Chapter 3 then provides the main aspects of the vehicle's flight dynamics, which together with the environmental aspects of the entry trajectory, described in Chapter 4, make up the theory of the core of the re-entry flight simulator. Chapter 5 elaborates on the sensor and estimator theory, as well as the theory that is used for implementation of the control system. Chapter 6 then describes the implementation of the theory of the aforementioned chapters into one simulator. Subsequently, the implementation, verification, and simulation of the control system is addressed in Chapter 7. Then, a similar approach is taken for the navigation system, where the implementation and verification of the sensors and estimators are described in Chapter 8. Subsequently, they are evaluated on performance, which is discussed in Chapter 9. This chapter is concluded by a section on the fully integrated navigation and control system and its performance. Finally, the conclusions that are drawn from the research that was done in this report are stated in Chapter 10, which is ended by recommendations for future research.

2

Mission Heritage and System Requirements

In Chapter 1, the motivation for the research that will be described in this report was explained, which was captured in the research question that was stated in the introduction. Then, before any theory is discussed, the mission heritage of the vehicle that was used in this thesis is defined in Section 2.1, as well as the application of the FADS, UKF and PF in the GNC-system of the vehicle. Then, concluding this chapter, a section is devoted to the statement of the mission and system requirements, which is captured in Section 2.2.

2.1 Mission Heritage

In the past century, much research has been conducted on traveling to and in space. One of the concepts that was designed was a system that would be able to go into orbit and return, after which it could be used again. At the end of the sixties, this principle lead to the design of the Space Shuttle, which was launched for the first time on April 12th, 1981. After 30 years and 135 missions, of which 133 were deemed successful, the Shuttle retired from service (Goodman, 2011). The final system that flew into space was indeed partly reusable, which was defined as a requirement during the preliminary part of the design cycle. The system consisted of a main launcher, which was expandable, two boosters, which were reusable after they were picked-up from landing, and the Shuttle itself, which was a winged re-entry vehicle. These wings were included for the planning of a horizontal landing on a runway, similar to most aircraft. Figure 2.1 shows a picture of the Space-Shuttle system, in which the whole system can be seen; on the front the winged Space Shuttle is shown, which has an aircraft-like structure. The main booster is located in the middle and at the back, and is the only part of the system that is not reusable. The two boosters on the side of the main tank are discarded shortly after launch, after which they can be retrieved, refurbished, and reused.

As the Space Shuttle program was an American concept, the Russians came up with a counter-move to produce their own reusable spacecraft, developed by the Buran program. The first flight of the Buran was on November 15th, 1988, more than seven years later than the Space Shuttle. The orbiter was very similar to the Space Shuttle and again concerned a winged re-entry vehicle. The launcher system however was not mounted to the orbiter, which was a difference between this system and the American system. The Buran program funding was cut later on and therefore this system flew only once, despite a successful launch and return.

A third system that was evaluated before starting the research is the HORUS 2B vehicle, which is a purely conceptual vehicle and has never flown. It was originally designed to be a fully reusable second stage to the Ariane-5 launcher (Mooij, 1998). Although it was designed to be equipped with a deorbitation system and control thrusters, it was referred to as an unpowered vehicle. This vehicle is considered to be very convenient, as a complete database exists in the document from MBB (1988), which is summarized by Mooij (1998) and used for determination of aerodynamic values in Mooij (1995).

Taking a different step in the field of winged re-entry vehicles, the family of space-planes is adressed. The Lockheed Martin-NASA X-33 Venture Star (shown on the right side of Figure 2.1) was a planned space-plane developed late in the previous century. Unfortunately this vehicle never flew; it was designed to be a SSTO (Single Stage To Orbit) vehicle that could be retrieved after the mission and reused for further missions. The Boeing X-37 Orbital Test Vehicle then was another reusable spacecraft designed for unmanned missions. This

(a)

(b)

Figure 2.1: (**a**) - The Space-Shuttle system on the STS-122 mission during launch, taken on February 7th, 2008[1]. (**b**) - The Lockheed Martin X-33 space-plane concept, this concept however has never flown[2].

vehicle has completed three successful missions, under the name of OTV-1, OTV-2 and OTV-3, and therefore could serve as the reference vehicle that is used in this thesis. The third space plane that is considered is NASA's X-38 mission. This was originally set up using a lifting-body concept, which would serve as an emergency Crew Return Vehicle (CRV) for the International Space Station (ISS) (Stein et al., 2007). This program contained three prototype flights before its budget was cut in 2002 [3].

In this section, only a set of winged re-entry vehicles were discussed. In Mooij (2015) a different type of vehicles was described, namely the capsules, which perform a so-called ballistic re-entry. The trajectories that are flown by ballistic re-entry vehicles are very steep and thus severely reduce the controllability of the vehicle performing the descent. Therefore, it was chosen to use only the aforementioned concepts, as they perform a gliding flight, which is less steep and has a longer flight time. As will be seen further on in this chapter, these winged re-entry vehicles may be equipped with aerodynamic control surfaces, which are used to control the vehicle in its flight towards Earth's surface.

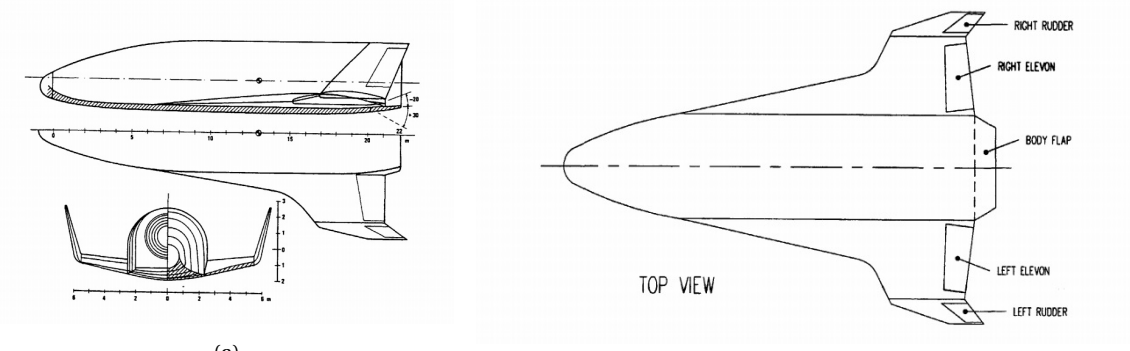
After evaluating the previously described concepts, a trade-off was performed. From this trade-off, it was concluded that the HORUS-2B was most useful as a reference vehicle during the research that is described in this report, as MBB (1988) provided a database containing the vehicle's geometry, mass and inertia properties and tabulated values of the aerodynamic coefficients, which were all summarized in Mooij (1995). Next to that, Mooij and Chu (2002) used this vehicle in the research on an EKF-based navigation module, thereby verifying the usability of the vehicle in similar research.

As the HORUS-2B vehicle was chosen as the reference vehicle for this study, it may be useful to create a short recap concerning its main characteristics. In the study from MBB (1988) a sketch of the HORUS 2B vehicle is found, which is presented in Figure 2.2a.

Earlier in the MBB study it was stated that this vehicle had multiple control surface actuators, namely two rudders, two elevons and one body flap, which are shown in Figure 2.2b. The configuration and characteristics of these actuators are discussed in Section 5.2.2. The aerodynamics of HORUS 2B that are discussed in Mooij (1995) are briefly described in Section 4.3, whereas all other characteristics are shown in Appendix A.3.

It was described that Mooij and Chu (2002) performed research on the application of an EKF-based navigation module on the HORUS-2B vehicle, in which an integrated Inertial Navigation System/Global Positioning System (INS/GPS) was applied. Gross et al. (2010) then studied the performance of the application

[1] http://spaceflight.nasa.gov/gallery/images/shuttle/sts-122/html/sts122-s-046.html, retrieved on 06-04-2017
[2] http://www.gagdaily.com/appealing/1514-lockheed-martin-x-33.html, retrieved on 06-04-2017
[3] http://www.chron.com/news/nation-world/article/X-38-project-s-cancellation-irks-NASA-partners-2064969.php, retrieved on 06-04-2017



(a) (b)

Figure 2.2: (**a**) Front, top, and side view of the HORUS 2B reference vehicle. (**b**) Lay-out of the control surface actuators of the HORUS 2B reference vehicle. Both images are based on MBB (1988) and Mooij (1995).

of EKF, UKF, and PF on an INS/GPS navigation system for the WVUYF-22 research aircraft, whereas Wendel et al. (2005) and Crassidis (2005) evaluated the performance of only the EKF and UKF. It was seen that in none of these studies, except the one from Mooij and Chu (2002), the performance of such a system on a re-entry vehicle flight was studied. In Mooij and Chu (2002) then only the EKF in combination with INS/GPS measurements was evaluated, and therefore it was chosen to research the application of UKF and PF on the HORUS-2B navigation module. Then, to ensure proper comparison with earlier research, it is chosen to use the same mission that was defined in Mooij and Chu (2002), along with the same initial conditions of the vehicle, which are described further on in this report.

Then, to expand the research somewhat more, it was chosen to include an air-data sensor in the system, which was not used in similar applications yet. Hillje and Nelson (1981) described that the Space Shuttle was equipped with an Ascent Air Data System (AADS), which was mainly used for low Mach numbers. From post-processed data it was found that the AADS showed large deviations in Mach number estimation, which, together with the small operational range, showed room for improvement. Whitmore and Moes (1994) described the FADS, which was said to be operational up until Mach numbers of approximately 13 and thus showed a relatively large operational range. This system also contained a built-in estimator that was able to approximate the Mach number, angle of attack and angle of sideslip of the vehicle, as well as the current free stream air pressure and dynamic impact pressure, as described by Cobleigh et al. (1999). The application of the pressure data in a navigation system, such as was described in Mooij and Chu (2002), however, was not performed, which served as an opportunity to research the applicability of this sensor in such systems. By doing this, the performance of the sensor integrated in a navigation module could be evaluated and compared to the stand-alone performance of the built-in estimator.

2.2 Mission and System Requirements

In the previous section, different re-entry vehicles were evaluated, after which a reference vehicle for this research was chosen and briefly discussed. Subsequently, previous research on the EKF, UKF, and PF was described, from which it was concluded that the application of the UKF and PF in a re-entry navigation module was missing, which drives the research described in this report. In addition, the FADS sensor was introduced, which is included in this research because its application in the navigation module is of interest. Then, finally, the mission and system requirements are to be formulated, which will be done in this section. The former contain the requirements with respect to the research that is performed in this thesis and are shown in Table 2.1, whereas the latter define the requirements of the system that is designed and evaluated and are presented in Table 2.2.

Table 2.1: Mission requirements for the thesis that is described in this report.

Label	Requirement
THESIS-M-1	The research shall be completed within the time span that is equal to 42 ECTS.
THESIS-M-2	All research elements that are planned shall be performed.
THESIS-M-3	A thesis report containing all research elements shall be delivered at the end of the research period.
THESIS-M-3-1	The research will be presented on the graduation day.
THESIS-M-3-2	The research will be evaluated during the thesis defence.
THESIS-M-4	The focus of the research will lie on the performance of the integrated navigation module.
THESIS-M-4-1	The focus of the research will contain the performance of the EKF, UKF, and PF.
THESIS-M-4-2	The focus of the research will contain the performance of the INS/GPS integrated system.
THESIS-M-4-3	The focus of the research will contain the implementation and performance of the FADS in the navigation module.

Table 2.2: System requirements for the simulator that is designed during the research that is described in this report.

Label	Requirement
THESIS-S-1	The system shall contain a complete vehicle model.
THESIS-S-1-1	The vehicle model shall contain all necessary geometric parameters and mass and inertia properties for performing the research.
THESIS-S-1-2	The vehicle model shall contain all necessary aerodynamic parameters for performing the research.
THESIS-S-2	The system shall contain an environmental model.
THESIS-S-2-1	The environmental model shall contain a representation of the gravitational field.
THESIS-S-2-2	The environmental model shall contain a representation of the atmosphere.
THESIS-S-3	The system shall describe the motion of the vehicle during its re-entry flight.
THESIS-S-3-1	Both the translational and rotational motion of the vehicle shall be considered.
THESIS-S-4	The system shall contain a fully implemented GNC-system for the 6DoF closed loop simulations.
THESIS-S-4-1	The GNC-system shall contain a guidance module.
THESIS-S-4-2	The GNC-system shall contain a navigation module.
THESIS-S-4-2-1	The navigation module shall contain a sensor system.
THESIS-S-4-2-2	The navigation module shall contain a state propagation filter.
THESIS-S-4-3	The GNC-system shall contain a control module.
THESIS-S-4-3-1	The control module shall contain control actuators.
THESIS-S-4-3-2	The control module shall provide control throughout the whole flight.
THESIS-S-5	The subsystems shall be verified by module verification.
THESIS-S-6	The system shall be verified by system verification.
THESIS-S-7	The system shall be evaluated on performance.

3

Flight Dynamics

Starting the design of the simulator, first the theory of the vehicle's motion is to be discussed, along with the environmental aspects that are encountered during flight. On the former, which is described in this chapter, different models can be used, which are dependent on the frame of reference that is used. A number of relevant reference frames is defined in Section 3.1, where also the transformation between different frames is touched upon. Then, a decision is made on the state variables that are used in the simulator, which is elaborated on in Section 3.2. Subsequently, the fundamentals of motion are described in Section 3.3, after which the external forces and moments are defined in Section 3.4. Then, the EoM are shown for both translational (Section 3.5) and rotational motion (Section 3.6). Chapter 4 then continues the basic theory with respect to the aforementioned environmental aspects.

3.1 Reference Frames

First, the reference frames that are used throughout the simulation are to be defined, as they are required for several computations. Forces and moments that are exerted on the re-entry vehicle, as well as the position, velocity, attitude and attitudinal rates of the vehicle itself are vectors, and to perform correct computations, it has to be clear in what reference frame these vectors are expressed. Transformation between different reference frames can be achieved by the so-called transformation matrices, which are discussed further on in this section.

3.1.1 Reference Frame Definitions

The following reference frame definitions are retrieved from Mooij (1994):

- **Inertial planetocentric reference frame** (I) - this reference frame has its origin in the center of mass (CoM) of the main body, which in this case is the Earth. The Z_I-axis is directed along the rotational axis of the main body towards the North and the X_I-axis is directed along the so-called reference meridian, which is defined at a specific point in time, as is done in the $J2000$-system. The Y_I-axis then completes the right-handed system.
- **Rotating planetocentric reference frame** (R) - the rotating frame is the second frame with its origin in the CoM of the main body. The Z_R-axis is coincident with the Z_I-axis, however, the OX_RY_R plane is fixed to the body itself. The X_R-axis is defined through the Greenwich meridian (zero longitude) and the Y_R-axis completes the right-handed system. At the $J2000$ epoch the inertial frame and the reference frame coincide, the R-frame then co-rotates with the main body whereas the I-frame remains fixed in inertial space.
- **Vertical reference frame** (V) - this reference frame has its origin in a reference point on the vehicle, which normally is the CoM The Z_V-axis is directed towards the CoM of the main body, which defines the OX_VY_V plane as the local horizontal plane. In this plane the X_V-axis is directed to the local north and the Y_V-axis completes the right-handed system.
- **Body reference frame** (B) - this reference frame also has its origin in a reference point on the vehicle. The OX_BZ_B plane then coincides with the plane of symmetry of the vehicle, where the Z_B-axis is directed

downwards and the X_B-axis is directed forwards through the nose of the vehicle. The Y_B-axis then completes the right-handed system.

- **Trajectory reference frame** (T) - as no wind will be used, no distinction between airspeed and groundspeed has to be made. The X_T-axis is positive in the velocity direction, the Z_T-axis is directed downwards in the vertical plane and the Y_T-axis completes the right-handed system.
- **Aerodynamic reference frame** (A) - for this system, the X_A-axis is defined in the same direction as the velocity vector. The Z_A-axis then is oriented in the opposite direction of the lift force and the Y_A-axis completes the right-handed system.

3.1.2 Reference Frame Transformations

The reference frames that were defined earlier in this section are used for the expression of state variables, forces and moments. To switch between different frames, a transformation matrix is required, which is comprised of a set of unit rotation matrices that are individually defined in Equation (3.1)

$$\mathbf{C_1}(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha \\ 0 & -\sin\alpha & \cos\alpha \end{bmatrix} \tag{3.1a}$$

$$\mathbf{C_2}(\alpha) = \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{bmatrix} \tag{3.1b}$$

$$\mathbf{C_3}(\alpha) = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.1c}$$

in which α represents the rotation angles over a specific axis. By defining a sequence of rotations the transformation matrix from frame a to frame b can be defined, which then is denoted as $\mathbf{C_{b,a}}$. Due to the fact that the unit axis rotation matrices are all orthonormal, any product of these matrices will also be orthonormal, hence it can be stated that (Mooij, 2015):

$$\mathbf{C_{a,b}} = \mathbf{C_{b,a}^{-1}} = \mathbf{C_{b,a}^{T}} \tag{3.2}$$

The transformation matrices that are used in the simulator are shown in Appendix A.1.

3.2 State Variables

This section describes the definition of the state variables, in terms of position and velocity of the vehicle and in terms of attitude, or, in other words, vehicle orientation. Note that different state variables may be used for different components of the simulator.

3.2.1 Position and Velocity

Concerning the position and velocity of the vehicle itself, it is chosen to use two systems for expression of the state variables, namely the Cartesian system and the spherical system. Both are used in different parts of the simulator, and they are mainly switched between for convenience or visualization of results. The Cartesian components possess the advantage of being straightforward and easily understood. Spherical components then are advantageous due to the fact that the Earth's shape is nearly round, and thus they can be visualized more easily with respect to the Earth. Both elements are also used in similar research (Mooij and Chu, 2002). The system of orbital elements is normally used to express the position and velocity of a vehicle in an orbit around a body. In Chapter 5, the use of GPS satellites is described, and to model the motion of these satellites, orbital elements will be used; they however will be briefly discussed in Section 8.1.2.

For the Cartesian system, the position and velocity vector are decomposed in three directions, each along one axis of the reference system that is currently used. For this, the type of reference frame is not of importance for the representation of the vector. The system of spherical components is directly related to the Cartesian system as a vector in this system is decomposed in a magnitude and two angles. Switching from the Cartesian to the spherical system and vice versa is done using the equations shown in Appendix A.2. In the simulator, the state of the vehicle was chosen to be expressed in the I-frame, as this frame is fixed in space and thus is

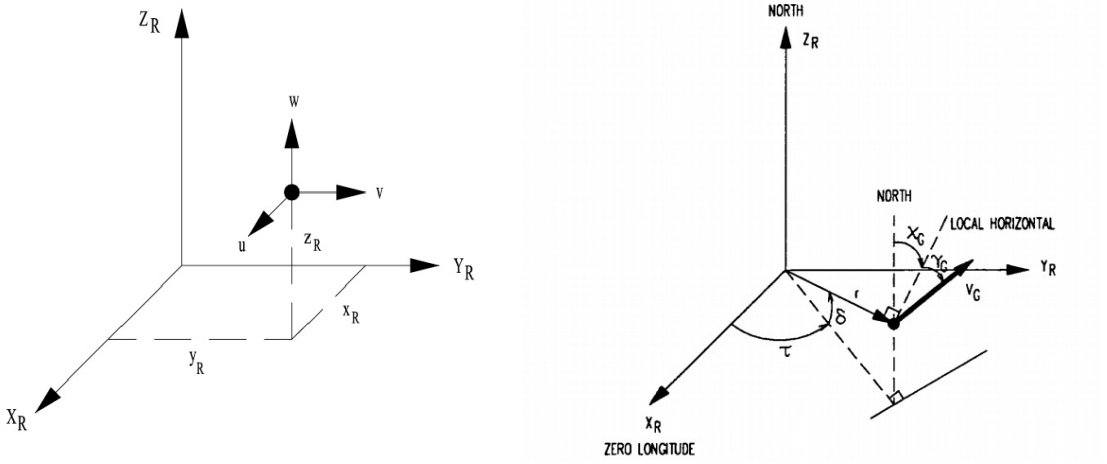


Figure 3.1: Left: Cartesian representation of the state variables in the rotating frame, which is similar to the representation in the inertial frame. Right: spherical representation of the position in the rotating frame and velocity in the vertical frame (Mooij, 2015).

independent of the rotation of the Earth. In this frame, it was chosen to use a Cartesian representation for the position and velocity, which is shown in Equation (3.3). For visualization of the results and in the equations of motion that are used in the design of the control system (Chapter 7) however, spherical coordinates are used, where the radius R, longitude τ, and latitude δ are defined for the position and are normally expressed in the co-rotating frame; the R-frame. When switching from the I-frame to the R-frame or a vehicle oriented frame, a correction is to be done for the rotation of the Earth. The V-frame is used in the simulator for expression of the velocity of the vehicle with respect to the local horizontal plane, which is directly related to the position of the vehicle with respect to the Earth. In this frame, velocity can be defined in Cartesian components, namely in north, east and downward direction (V_N, V_E, and V_D). The velocity in the V-frame can also be expressed in spherical components, providing a better visualization of the motion of the vehicle. This representation contains the velocity modulus V, the flight-path angle γ, and the heading angle χ.

$$\mathbf{X_I} = \begin{pmatrix} x_I \\ y_I \\ z_I \end{pmatrix}, \mathbf{V_I} = \begin{pmatrix} \dot{x}_I \\ \dot{y}_I \\ \dot{z}_I \end{pmatrix} \tag{3.3}$$

In the representation shown in Equation (3.3) $\dot{x}_I$, $\dot{y}_I$, and $\dot{z}_I$ define the velocity components in the directions of the I-frame axes. For the spherical system, δ is defined to be the latitude ($-90^\circ \leq \delta \leq 90^\circ$), which is zero at the equator and is positive in the northern hemisphere. The longitude τ is zero at the Greenwich meridian and is positive to the east (Mooij, 1997), where $-180^\circ \leq \tau < 180^\circ$. The distance R is then measured from the CoM of and completes the spherical vector representation. When evaluating the velocity, the velocity V represents the magnitude of the vector (similar to R for the position). The flight-path angle γ is then defined as the angle between $\mathbf{V}$ and the local horizontal plane, where it ranges from $-90^\circ \leq \gamma \leq 90^\circ$ and is measured positively when $\mathbf{V}$ is directed above the local horizon. The heading angle χ then ranges from $-180^\circ \leq \chi < 180^\circ$ and represents flying to the east parallel to the equator for $\chi = +90^\circ$. A heading to the north is represented by $\chi = 0^\circ$.

3.2.2 Attitude

Earlier it was stated that both translational and rotational motion of the vehicle were to be considered. Therefore, the attitude and rotational rates of the vehicle have to be evaluated. The attitude itself can be expressed via Euler angles, quaternions and modified Rodrigues parameters (MRPs) (Mooij, 1994 and Crassidis and Markley, 1996). It was chosen that in the simulator all computations were to be done using quaternions, as this avoids the occurrence of singularities. MRPs only experience singularities for angles of 360°, but still possess the possibility of encountering them. Therefore it was decided to use quaternions for all internal computations of the simulator. However, due to the fact that quaternions are not easily understood when visualized, it was chosen to compute the roll, pitch, and yaw angles via the conversions that are shown in Appendix A.2. These Euler angles are used to describe the orientation of the vehicle's body frame with respect

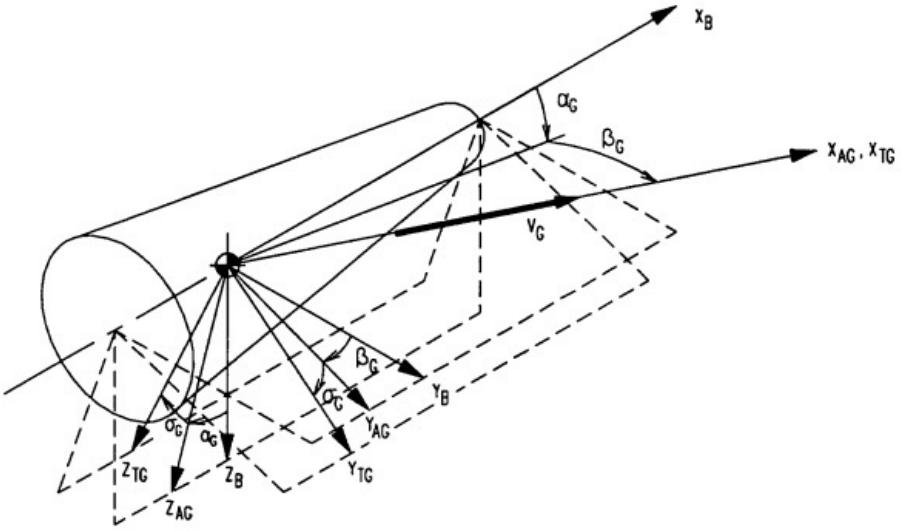


Figure 3.2: Definition of the set of aerodynamic angles (angle of attack, angle of sideslip and bank angle). The subscript G denotes the so-called groundspeed, which, in the absence of wind, is equal to the velocity of the vehicle. (Mooij, 2015).

to the inertial frame. However, in Chapter 7 it will be seen that the input of the control system requires aerodynamic Euler angles, which can be computed by transforming the roll, pitch, and yaw angle.

The Euler angles ϕ, θ and ψ represent rotations around the respective x, y and z axes of the reference frames. For visualization of the attitude of the vehicle these angles represent the orientation of the vehicle with respect to the inertial frame. The roll angle ϕ is measured positive when a right turn is initiated ($-180° \leq \phi < 180°$), the pitch angle is measured positive for pitch up ($-90° \leq \theta \leq 90°$) and the yaw angle is measured positive for nose right ($-180° \leq \psi < 180°$). The aerodynamic Euler angles consist of the angle of attack α with $-180° \leq \alpha < 180°$, measured positive for nose up, the angle of sideslip β with $-90° \leq \beta \leq 90°$, measured positive for nose right, and the bank angle σ with $-180° \leq \sigma < 180°$, measured positive when a right turn is initiated. The former two angles are required for transformation between the body frame and the aerodynamic frame, whereas the bank angle is required for the transformation between the trajectory frame and the aerodynamic frame. The definition of the aerodynamic angles is shown in Figure 3.2.

The system that is used in the simulator are the quaternions, which are convenient in the sense that no singularities can occur. A quaternion consists of four elements that together describe a set of rotation angles. It is defined as (Kuipers, 1999)

$$Q_1 = a_1 \sin\left(\frac{\theta}{2}\right) \tag{3.4a}$$

$$Q_2 = a_2 \sin\left(\frac{\theta}{2}\right) \tag{3.4b}$$

$$Q_3 = a_3 \sin\left(\frac{\theta}{2}\right) \tag{3.4c}$$

$$Q_4 = \cos\left(\frac{\theta}{2}\right) \tag{3.4d}$$

where the first three elements denote a 3×1-vector and the fourth element defines a scalar. One of the main characteristics of quaternions is that the norm of the four elements is equal to one:

$$Q_1^2 + Q_2^2 + Q_3^2 + Q_4^2 = 1 \tag{3.5}$$

Finally, the angular rates of the attitude angles of the vehicle have to be discussed. These rates are normally contained in an angular rotation vector that describes the rotation of the body-fixed reference frame with respect to the inertial or rotating frame (Mooij, 2015). Note that these variables denote the time derivatives of the roll, pitch, and yaw angle, and are defined to be the roll rate p, the pitch rate q and the yaw rate r.

3.3 Fundamentals of Motion

Before anything can be said concerning the equations of motion, the physical laws that are to be obeyed have to be defined, as well as the assumptions that are made to simplify the problem.

Newton's Laws

It is started a brief recap of Newton's laws as these were used in the derivation of the equations of motion. Although this derivation is not included in this report, these fundamentals are formulated for understanding. The four laws are listed below (Mulder et al., 2013):

1. Any object on which there is no net force travels with constant velocity. This law can be put in a mathematical format:
$$\text{if} \sum \mathbf{F}_I = 0, \text{then} \frac{d\mathbf{V}_I}{dt} = 0 \tag{3.6}$$
2. The resulting vector force on an object is equal to the mass of the object multiplied by its acceleration vector, or, in mathematical terms:
$$\mathbf{F}_I = m\mathbf{a}_I \tag{3.7}$$
3. When one object exerts a force on a second object, that second object exerts an opposed force of equal magnitude on the first object at the same time:
$$\mathbf{F}_{I,a} = -\mathbf{F}_{I,b} \tag{3.8}$$
4. Two objects with masses m_1 and m_2 always exert a force on each other. This is the law of gravitation and will be further explained in Section 4.1, it can be reduced to the following equation:
$$\mathbf{F}_I = G\frac{m_1 m_2}{|r^3|}\mathbf{r}_I \tag{3.9}$$

Fundamental Assumptions

Finally, some assumptions are to be discussed, which are to be used in the simulator. Note that these assumptions are made to simplify the derivation of the final set of equations of motion, without losing very much accuracy of the approximation of the vehicle's motion.

- The flattening of the Earth is set to zero, which makes it a spherical model.
- The vehicle that is considered has a constant mass and does not deform during flight.
- The vehicle is rotationally symmetric, causing the products of inertia to be zero.
- The initial rotation of the vehicle is set to zero.

3.4 Forces and Moments

During its flight towards the surface of the Earth, different external forces and moments are exerted on the re-entry vehicle, thereby influencing its trajectory. As an unpowered vehicle is considered, the only two sources that remain in the simulator are the aerodynamic and the gravitational forces and moments, which are described in this section.

Aerodynamics

Due to interaction of the vehicle with the surrounding atmosphere a source of forces and moments is created that is categorized as aerodynamics. These forces and moments depend on the airspeed variables, according to Mooij (1994), and are discussed in this paragraph. They are described by Mooij (1994) as

$$\mathbf{F}_{\mathbf{A,A}} = \begin{pmatrix} -D \\ -S \\ -L \end{pmatrix} = \begin{pmatrix} -C_D \bar{q} S_{\text{ref}} \\ -C_S \bar{q} S_{\text{ref}} \\ -C_L \bar{q} S_{\text{ref}} \end{pmatrix} \tag{3.10}$$

and are dependent on the aerodynamic coefficients for drag C_D, sideforce C_S and lift C_L, as well as the reference area S_{ref} and the dynamic pressure $\bar{q}$, which is related to the atmospheric density ρ and the airspeed via:

$$\bar{q} = \frac{1}{2}\rho V^2 \tag{3.11}$$

The coefficients that are described in Equation (3.10) depend on the Mach number, the angle of attack and the angle of sideslip of the vehicle. The Mach number is defined as

$$M = \frac{V}{a} \tag{3.12}$$

with V being the velocity in m/s and a being the speed of sound, which is also expressed in m/s. The definition of the speed of sound is

$$a = \sqrt{\gamma R T} \tag{3.13}$$

where γ is the specific heat ratio of the air, R is the air specific gas constant and T is the air temperature, which varies with altitude.

In terms of aerodynamic moments, a similar equation to Equation (3.10) can be set up, where the roll, pitch, and yaw moment coefficients C_l, C_m and C_n are used respectively. These coefficients again depend on the Mach number, the angle of attack and the angle of sideslip. The aerodynamic moments are computed via

$$\mathbf{M}_{\mathbf{A_M},\mathbf{B}} = \begin{pmatrix} \mathscr{L} \\ \mathscr{M} \\ \mathscr{N} \end{pmatrix} = \begin{pmatrix} C_l \bar{q} S_{\text{ref}} b_{\text{ref}} \\ C_m \bar{q} S_{\text{ref}} c_{\text{ref}} \\ C_n \bar{q} S_{\text{ref}} b_{\text{ref}} \end{pmatrix} \tag{3.14}$$

where b_{ref} and c_{ref} are aerodynamic reference lengths. The total moment however does not only consist of these computed values, but also has a contribution from the aerodynamic forces, using the definition for an arbitrary moment:

$$\mathbf{M}_{A_F,B} = \mathbf{r}_{\text{cm}} \times \mathbf{F}_{A,B} \tag{3.15}$$

where $\mathbf{r}_{\text{cm}}$ is the arm and defines the distance from the acting point of the force towards the center of mass of the vehicle. The total aerodynamic moment is retrieved by simply adding both moment contributions:

$$\mathbf{M}_{A,B} = \mathbf{M}_{A_M,B} + \mathbf{M}_{A_F,B} \tag{3.16}$$

Note that the aerodynamic forces that were stated at the beginning of this section are expressed in the aerodynamic frame, whereas the aerodynamic moments as well as the contribution of the force in Equation (3.15) are expressed in the body frame. Using the reference frame transformations from Section 3.1 all moments and forces from this paragraph can be used together.

Gravity

The second and also the final source of external forces and moments that is to be discussed is the gravitational field of the Earth. As modeling this field can be a research topic on its own, a simplified model is used, which is sufficiently accurate for the simulator and will be described in Chapter 4. This model is based on a spherical field, and therefore a spherical representation is most convenient, which is done in the R-frame, as described earlier in this chapter. The external force related to the gravitational field is computed using Newton's second law:

$$\mathbf{F}_{G,R} = m\mathbf{g}_R \tag{3.17}$$

where m is the mass of the vehicle and $\mathbf{g}_R$ is the gravitational acceleration expressed in the rotating frame.

For the gravitational moments, it is stated that they are induced by the fact that the CoM of the vehicle is not perfectly aligned with the center of gravity of the vehicle, which is due to differences in the gravitational field on different parts of the vehicle. As was stated, only a simplified model of the gravitational field will be used, which makes these differences negligible over an object that has the size of the vehicle. Therefore, the gravitational moments can be neglected, which concludes the final equations for the external forces and moments:

$$\mathbf{F}_I = \mathbf{F}_{A,I} + \mathbf{F}_{G,I} \tag{3.18}$$

which is expressed in the I-frame as can be seen. This is done as the state propagation is done in the I-frame, which was discussed earlier in this chapter. Due to this, the gravitational force has to be transformed from the rotating frame to the inertial frame, whereas the aerodynamic forces and moments have to be transformed from the aerodynamic to the inertial frame.

3.5 Equations of Translational Motion

This section contains the derivation of the equations of translational motion for the re-entry flight that is to be controlled. In the previous sections the fundamentals of motion in the form of assumptions and basic laws were shown, as well as the reference frames that will be used, the transformations between the reference frames that are to be used, the systems in which the state variables were defined and the external forces and moments that act upon the vehicle. The equations of motion that are shown in this section are used for the state propagation of the vehicle in the simulator. For the integration of the equations of motion (EoM) the inertial frame is used, however, for the control system, the EoM in the vertical frame are used, which are handled in Chapter 7. As will be shown further on in this report, the input state contains a position and velocity expressed in spherical components. However, as the EoM will be less complex when using Cartesian components, the initial state is first transformed to the Cartesian system.

The equations for translational motion are then defined as (Mooij, 2015):

$$\frac{d\mathbf{r}_I}{dt} = \mathbf{V}_I \tag{3.19}$$

$$\frac{d\mathbf{V}_I}{dt} = \mathbf{a}_I = \frac{\mathbf{F}_I}{m} \tag{3.20}$$

where $\mathbf{r}_I$ is the position of the vehicle, $\mathbf{V}_I$ is the velocity, $\mathbf{a}_I$ is the acceleration, m is the mass of the vehicle and $\mathbf{F}_I$ are the forces that are exerted on the vehicle, which were defined in Section 3.4. In these equations it is simply defined that the derivative of the position is the velocity and that the derivative of the velocity results in the acceleration. The acceleration can then be computed using Newton's second law, as was described in Section 3.3. The resulting force $\mathbf{F}_I$ was obtained by adding all forces that are exerted on the vehicle, which are in this case the aerodynamic forces and the gravitational forces, as seen in Equation (3.18). Note that wind is not taken into account, as this topic lies beyond the scope of this research.

Earlier it was stated that when transformation from the inertial to the rotating frame was to be done, a correction for the rotation of the Earth had to be applied to the velocity of the vehicle, which is defined by:

$$\mathbf{V}_R = \mathbf{C}_{R,I}\left(\mathbf{V}_I - \boldsymbol{\omega}_{cb} \times \mathbf{r}_I\right) \tag{3.21}$$

where $\mathbf{C}_{R,I}$ is the transformation matrix from the inertial to the rotating frame and $\boldsymbol{\omega}_{cb}$ is the rotation of the Earth. This concludes the translational equations of motion in Cartesian components, expressed in the inertial frame. The aforementioned correction for the rotation of the Earth is to be applied when transforming velocity to a different reference frame.

3.6 Equations of Rotational Motion

Previously it was described that quaternions were to be used for the attitude computations, whereas Euler angles were chosen for the visualization in interpretation of the results. In this section the rotational equations of motion are provided, which are integrated to define the attitude propagation of the vehicle. The quaternion representation from Kuipers (1999) can be shown as a vector

$$\mathbf{Q}_{I,B} = (Q_1, Q_2, Q_3, Q_4)^T \tag{3.22}$$

which denotes the attitude of the body frame with respect to the inertial frame. According to Mooij (1994) the time derivative of this vector is stated as

$$\dot{\mathbf{Q}}_{I,B} = \frac{1}{2}\begin{bmatrix} -Q_4 & -Q_3 & Q_2 \\ Q_3 & -Q_4 & -Q_1 \\ -Q_2 & Q_1 & -Q_4 \\ Q_1 & Q_2 & Q_3 \end{bmatrix}\begin{pmatrix} p \\ q \\ r \end{pmatrix} \tag{3.23}$$

where p, q and r are the roll rate, pitch rate and yaw rate respectively. Note that this equation is already sufficient for the state propagation of the quaternion values.

The only remaining process that has to be evaluated is the computation of the angular accelerations, which will complete the rotational equations of motion. In Mooij (1994) it can be seen that the roll, pitch and yaw accelerations are dependent on the moments that exert on the vehicle, the moments of inertia of the vehicle

and the current roll, pitch and yaw rates. The angular accelerations are defined as

$$\dot{p} = \frac{M_x}{I_{xx}} + \frac{I_{yy} - I_{zz}}{I_{xx}} qr \tag{3.24a}$$

$$\dot{q} = \frac{M_y}{I_{yy}} + \frac{I_{zz} - I_{xx}}{I_{yy}} pr \tag{3.24b}$$

$$\dot{r} = \frac{M_z}{I_{zz}} + \frac{I_{xx} - I_{yy}}{I_{zz}} pq \tag{3.24c}$$

in which I denotes the moment of inertia and M defines the moment around a certain axis. Note that these equations are independent of the use of quaternions or Euler angles, and can be used for both systems.

4

Entry Environment

This chapter will describe all environmental aspects that are encountered during the simulated re-entry flight, and which are required to model in order to ensure a proper simulation. For the entry problem discussed in this thesis, the Earth is considered to be the central body, which defines two topics that are to be considered: the planetary atmosphere (Section 4.2) and the gravitational field (Section 4.1), as in the design of the simulator it was assumed that the Earth is spherical and that the geometric altitude can be used in the calculations. As is described in Chapter 3, the main forces that act upon the vehicle are gravitational and aerodynamic, which relates the previously mentioned three topics to the dynamics of the entry problem. In Section 4.3 then the HORUS aerodynamics are considered.

4.1 Gravitational Field

Gravity is one of the main external forces that defines the trajectory of the re-entry vehicle, as it pulls the vehicle towards the central body during its whole flight. It was chosen to use the basic gravitational field model, which is derived from Newton's fourth law (Equation (3.9)) in combination with some of the so-called gravitational harmonics.

4.1.1 Basics of Gravity

Chapter 3 contained the derivation of the EoM, where it was stated that the dynamics of the vehicle would be expressed in the inertial frame (the I-frame). The gravitational field is a spherical representation and thus the use of either the I-frame or the R-frame is sufficient, where a Cartesian representation is most convenient. In Section 3.4 it was shown that the gravitational force on an object is defined as

$$\mathbf{F}_{G,R} = m\mathbf{g}_R \tag{4.1}$$

where $\mathbf{F}_{G,R}$ is the force in the rotating frame, m is the mass of the object and $\mathbf{g}_R$ is the gravitational acceleration vector, which is also expressed in the rotating frame. This equation is formulated using Newton's second law; the gravitational attraction between two objects is defined by Newton's fourth law (Wakker, 2010)

$$\mathbf{F}_{G,R} = G\frac{m_1 m_2}{r^3}\mathbf{r}_R \tag{4.2}$$

where G is the universal gravitational constant (and has a value of $6.67428 \cdot 10^{-11}$ m^3 kg^{-1} s^{-2} (Wakker, 2010)) and m_1 and m_2 represent the masses of both objects, which are in this case the re-entry vehicle and the Earth. The vector $\mathbf{r}_R$ and its magnitude r define the distance between both objects. As Newton's fourth law is universal and applies to all objects, it also holds for the case described in this thesis with the re-entry vehicle and the Earth. When m_1 is substituted by the mass of the Earth M_E, the gravitational parameter of the Earth can be calculated by multiplying this value with the universal gravitational constant. This value μ will be used in the simulator and has a value of $3.986004356 \cdot 10^{14}$ m^3 s^{-2} (Wertz, 2009). In Equation (4.1) it was seen that for the computation of the gravitational force, the gravitational acceleration vector was required, which will be defined in the following paragraph.

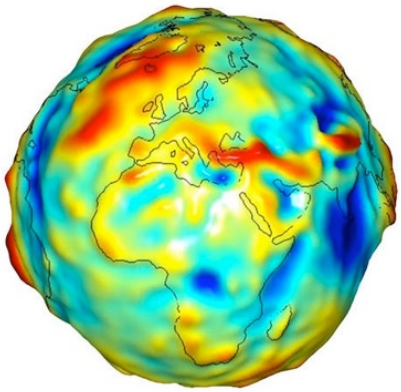


Figure 4.1: Gravity field over the Earth's surface; it displays the strength of the gravitational field distributed over the Earth, where red denotes a strong gravity field and blue denotes a weak(er) gravity field[1].

4.1.2 Gravity Perturbations and Harmonics

Earlier, two equations were provided that can be used to compute the gravitational force that is exerted on an object. However, in Equation (4.2) it was assumed that the Earth is a sphere with an evenly distributed density. Nonetheless, in real life this is not the case, and therefore the computation of the gravitational force via Equation (4.2) is only an approximation. In real life, perturbations and harmonics in the gravitational field affect the magnitude of the force that is exerted on the re-entry vehicle. The theoretical aspects of gravity are now continued with the relation between the gravitational field and the shape of the Earth.

Figure 4.1 shows the strength of the gravitational field of the Earth. It can be seen that this does not comply with the gravitational law that was derived earlier in this section, as the strength of the gravitational field was assumed equal everywhere, due to the spherical approximation of the Earth's shape together with the assumption that the mass is evenly distributed.

If the gravitational field is to be modeled with more accuracy than was done in Equation (4.2), so-called Jeffery constants have to be included, which represent the harmonic expansion and reduction of the gravity field of the Earth. Every J_x-constant denotes a perturbation in the field due to a different factor. For example, the J_2-constant (or the J_2-effect) is used to compensate for the oblateness of the Earth; due to the rotational speed, more mass will be located around the equator of the rotational plane, and less mass will be located around the poles of this plane, thereby 'flattening' the Earth. The J_2-term will account for this effect in the value of the gravitational field. And so, the gravitational acceleration $\mathbf{g}_R$ can be estimated more and more accurately by adding more harmonic terms.

As the focus of this thesis does not lie on the modeling of the gravitational field of the Earth, it is decided to only include the Jeffery constants up to the J_2-effect. The values of these constants are defined by Wertz (2009):

- Zeroth order, $J_0 = 1$
- First order, $J_1 = 0$
- Second order, $J_2 = 0.00108263$

Higher orders are neglected as their influence is so small that the simulated trajectory will only show a negligible difference with the real life trajectory concerning this topic. Next to that, Regan and Anandakrishnan (1993) stated that the J_2-effect is estimated with a much larger accuracy than the higher order terms, and therefore it is reasonable to discard these higher order terms as they bring along a relatively larger inaccuracy to the magnitude of the gravitational field than the lower order terms.

Taking the J_2-effect into account, the following representation of the gravitational acceleration in the Earth-fixed reference frames can be defined

$$\mathbf{g}_R = \begin{bmatrix} g_r \\ g_\tau \\ g_\delta \end{bmatrix} \tag{4.3}$$

where the individual components are computed via (Mooij, 1994):

[1] `http://www.jarredwilson.com/wp-content/uploads/2013/02/geoid.png` - retrieved: 02-09-2016

$$g_r = \frac{\mu}{r^2}\left[1 - \frac{3}{2}J_2\left(\frac{R_E}{r}\right)^2\left(3\sin^2\delta - 1\right)\right] \tag{4.4}$$

$$g_\delta = -3\frac{\mu}{r^2}J_2\left(\frac{R_E}{r}\right)^2 \sin\delta\cos\delta \tag{4.5}$$

The gravitational component in the longitudinal direction can be neglected, as it is assumed that the Earth is symmetric around its rotational axis. The value r represents the distance between the object and the center of gravity of the Earth, μ is the gravitational parameter, R_E is the Earth's radius and J_2 is the aforementioned coefficient for the oblateness of the Earth. Note that as the state propagation in the simulator is done in the I-frame, the gravitational force is transformed when applied to the flight dynamics block, as was described in Section 3.4.

4.2 Atmosphere

In Chapter 3 it was discussed that two sources of external forces and moments are considered, which are the Earth's gravitational field and the motion of the vehicle through an atmosphere, which induces aerodynamic forces and moments. In this section a model of the Earth's atmosphere will be discussed, which will be used for calculation of these aerodynamic forces and moments that are defined later on in this document.

Before this research was started, different atmospheric models were evaluated and traded off against each other. After this, it was chosen to use the US76 Standard Atmosphere model, which has the advantage of the availability of verified tabulated data and thus can easily be verified. In similar research topics (Costa et al., 2002 and Mooij et al., 2007) the US76 Standard Atmosphere is also used and did not oppose structural limitations to the research at hand. This model, which will be referred to as US76, is divided in to two regions; below and above a geopotential altitude of 84.85 km (NASA, 1976), which relates to geometric altitude of 86 km. It is based on the following three assumptions:

- The atmosphere behaves as an ideal gas
- The atmosphere is in hydrostatic equilibrium
- The atmosphere is homogeneous in the lowest part

The US76 model states that the atmosphere can be divided into several regions, depending the behavior of the temperature. Up until a geometric altitude of 86 km, the model defines that the temperature either increases or decreases for increasing altitude or is constant. Using this, it is defined that (Anderson Jr., 2012):

$$\frac{\partial T}{\partial z} \equiv a \tag{4.6}$$

where z is the geometric altitude, T is the temperature and a is the temperature gradient. For the aforementioned regions the density, pressure and temperature functions were derived, which are summarized in Table 4.1. In these equations, p defines the air pressure, ρ represents the air density, g is the magnitude of the gravitational acceleration, which is assumed to be constant in this model and R is the universal gas constant.

Table 4.1: Summary of all governing atmospheric equations from Section 4.2. Subscript 2 denotes the final altitude, subscript 1 denotes the initial altitude. Equations hold until a geopotential altitude of 84.8520 km (NASA, 1976)

	Temperature increases with altitude	Temperature is constant with altitude	Temperature decreases with altitude
a	$a > 0$	$a = 0$	$a < 0$
T_2	$T_2 = T_1 + a(z_2 - z_1)$	$T_2 = T_1$	$T_2 = T_1 + a(z_2 - z_1)$
p_2	$p_2 = p_1 \cdot \left(\frac{T_2}{T_1}\right)^{-\frac{g}{aR}}$	$p_2 = p_1 \cdot e^{\left[-\frac{g}{RT}(z_2 - z_1)\right]}$	$p_2 = p_1 \cdot \left(\frac{T_2}{T_1}\right)^{-\frac{g}{aR}}$
ρ_2	$\rho_2 = \rho_1 \cdot \left(\frac{T_2}{T_1}\right)^{-\left(\frac{g}{aR}\right)+1}$	$\rho_2 = \rho_1 \cdot e^{\left[-\frac{g}{RT}(z_2 - z_1)\right]}$	$\rho_2 = \rho_1 \cdot \left(\frac{T_2}{T_1}\right)^{-\left(\frac{g}{aR}\right)+1}$

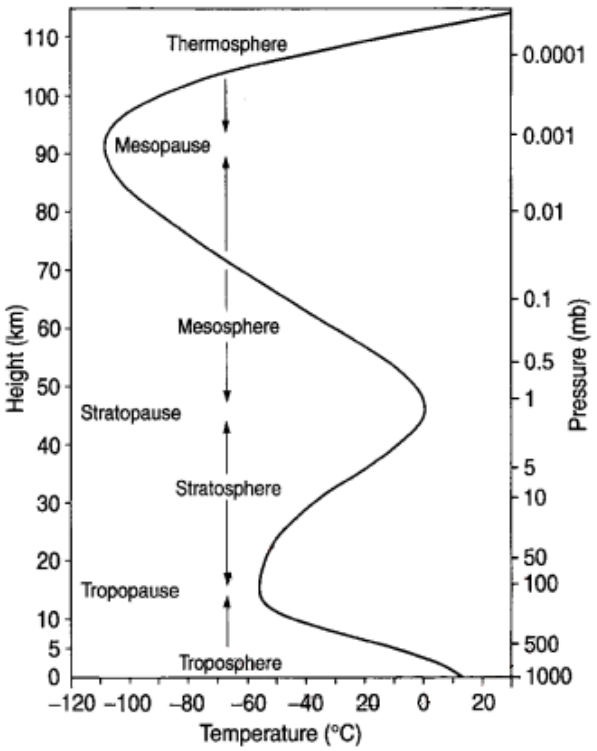


Figure 4.2: Thermal profile of the Earth's atmosphere, containing all layers that were described earlier in Section 4.2[2].

4.2.1 Earth's Atmosphere

In the previous subsection the resulting equations of the basic laws of the US76 atmosphere were demonstrated, which are used for the computation of temperature, density and pressure. However, the model states that the temperature either increases linearly, stays constant or decreases linearly, which may not be the case in any real atmosphere. This section evaluates the Earth's atmosphere, as this atmosphere will be used throughout this study.

In the US76 model, the computation of air pressure and air density is based on temperature behavior in the atmosphere, and therefore the thermal profile is of importance. The Earth's thermal profile is shown in Figure 4.2 and it can be seen that it resembles an exponential atmosphere in the sense that the different parts are either increasing or decreasing almost linearly or are constant for different altitudes. The profile observed in Figure 4.2 is somewhat smoother than the exponential atmosphere and that is where small inaccuracies are introduced.

Using Figure 4.2 it can be stated that a re-entry vehicle thus has to cope with various temperature regions during its flight towards the surface. The regions up until a geometric altitude of 86 km can be covered by the equations that were shown in Table 4.1. For higher altitudes, different temperature behavior is found, and thus they will be described individually.

Region 8: 86 - 91 km: This is an isothermal region and thus the temperature stays constant at a value of $T_7 = T_8 = 186.8673$ K.

Region 9: 91 - 110 km: This is a region where the temperature increases along an elliptical form. It can be computed using:

$$T = T_c + A\left[1 - \left(\frac{z - z_8}{a}\right)^2\right]^{\frac{1}{2}} \tag{4.7}$$

where $T_c = 263.1905$ K, $A = -76.3232$ K and $a = -19.9429$ km.

[2] https://scienceofdoom.files.wordpress.com/2010/03/pressure-and-height-vs-temp-highlight-bigg-2005.png - retrieved: 06-09-2016

Table 4.2: Thermal gradients, region boundary value ratios and their respective geopotential altitude ranges according to NASA (1976), up until $h = 84.85$ km.

Region	Altitude range h [km]	Thermal gradient [K/km]	T_0 [K]	p/p_0 [-]	ρ/ρ_0 [-]
1	0.0 - 11.0	−6.5	288.15	1.0000	1.0000
2	11.0 - 20.0	0.0	216.65	$2.2336 \cdot 10^{-1}$	$2.9708 \cdot 10^{-1}$
3	20.0 - 32.0	+1.0	216.65	$5.4032 \cdot 10^{-2}$	$7.1865 \cdot 10^{-2}$
4	32.0 - 47.0	+2.8	228.65	$8.5666 \cdot 10^{-3}$	$1.0796 \cdot 10^{-2}$
5	47.0 - 51.0	0.0	270.65	$1.0945 \cdot 10^{-3}$	$1.1653 \cdot 10^{-3}$
6	51.0 - 71.0	−2.8	270.65	$6.6063 \cdot 10^{-4}$	$7.0335 \cdot 10^{-4}$
7	71.0 - 84.85	−2.0	214.65	$3.9046 \cdot 10^{-5}$	$5.2417 \cdot 10^{-5}$

Region 10: 110 - 120 km: This region has a linear increase in temperature, starting at the end temperature of the previous region.

$$T = T_9 + a_{10}(z - z_9) \tag{4.8}$$

where $T_9 = 240.0$ K and $a_{10} = 12.0$ K km^{-1}.

Region 11: 120 - 1000 km: The temperature increases exponentially, which is represented by:

$$T = T_\infty - (T_\infty - T_{10})\, e^{-\lambda \xi} \tag{4.9}$$

where $T_\infty = 1000$ K, $T_{10} = 360.0$ K, $\lambda = 0.01875$ and $\xi = (z - z_{10})\left(\frac{R_E + z_{10}}{R_E + z}\right)$. The pressure in region 8 to 11 can then be computed using

$$p = \frac{\sum n_j R^* T}{N_A} \tag{4.10}$$

where $R^* = 8.3144621$ J kg^{-1} mol^{-1} and $N_A = 6.022169 \cdot 10^{26}$ kmol^{-1}, which is Avogadro's constant. According to NASA (1976), the factor $\sum n_j$ is the sum of the number densities of the individual gas species and can be computed using:

$$\sum n_j = N = N_i e^{(z - z_i) \ln\left[\frac{N_{i+1}}{N_i (z_{i+1} - z_i)}\right]} \tag{4.11}$$

which is an interpolation from the values that are mentioned in NASA (1976). Subsequently, the density can be computed using the ideal gas law.

It was seen that all temperature, air pressure and air density equations were defined with respect to a reference point, which is located on the lower boundary of the respective region. In Table 4.2 and Table 4.3 the boundary values are shown for the whole atmospheric model, where a distinction is made between the lower model (region 1 to 7) and the upper model (region 8 to 11). In the lower model, the atmosphere is modeled exponentially, and as the regions are defined as function of geopotential altitude, this is also used in Table 4.2. The air pressure and density can be computed using the ratios shown in this table and the sea-level values, which were defined by NASA (1976) to be:

- $p_0 = 1.01325 \cdot 10^5$ N m^{-2}
- $\rho_0 = 1.2250$ kg m^{-3}
- $T_0 = 288.15$ K

In the upper region the density is so small that number densities of the individual gas species are used, as was described earlier in this section. The regions are defined as function of geometric altitude, and therefore this is also used in Table 4.3.

Using these boundary values in combination with Table 4.1 and the equations for regions 8 to 11, the temperature, air density and air pressure can be computed for all altitudes up until 1000 km, and due to the fact that the initial altitude of the vehicle will be around 120 km, this model is sufficient for computation of the atmospheric properties.

Table 4.3: Thermal gradients and their respective geometric altitude ranges according to NASA (1976) up until $z = 1000$ km.

Region	Altitude range h [km]	T_0 [K]	p/p_0 [-]	ρ/ρ_0 [-]
8	86.0 - 91.0	186.87	$3.6850 \cdot 10^{-6}$	$5.680 \cdot 10^{-6}$
9	91.0 - 110.0	186.87	$1.5179 \cdot 10^{-6}$	$2.335 \cdot 10^{-6}$
10	110.0 - 120.0	240.00	$7.0113 \cdot 10^{-8}$	$7.925 \cdot 10^{-8}$
11	120.0 - 1000.0	360.00	$2.5050 \cdot 10^{-8}$	$1.814 \cdot 10^{-8}$

4.3 Aerodynamics

This part of the re-entry fundamentals concerns the aerodynamic aspects of the re-entry flight. As the simulated re-entry vehicle performs gliding flight from an initial altitude of roughly 120 km, it is assumed that there is continuum flow throughout the whole flight, which means that the mean path length between particles in the air is negligible with respect to the size of the vehicle. Due to this assumption, it can be stated that air density and pressure can be defined throughout the whole flow region. In terms of aerodynamics, only the vehicle related equations have to be discussed, which is done in this section.

Mooij (1995) describes the aerodynamic properties of HORUS 2B, which are used in the re-entry simulation. Section 3.4 described the aerodynamic forces and moments that are present during flight, and each of these forces and moments was expressed using a coefficient. The force coefficients are defined by MBB (1988) and Mooij (1995) as:

$$C_D = C_{D_0} + \Delta C_{D_{r,l}} + \Delta C_{D_{e,i}} + \Delta C_{D_b} + \Delta C_{D_{e,r}} + \Delta C_{D_{r,r}} - \Delta C_{D_h} \tag{4.12a}$$

$$C_S = \Delta C_{S_{r,l}} + \Delta C_{S_{e,l}} + \Delta C_{S_{e,r}} + \Delta C_{S_{r,r}} + \left[\left(\frac{\partial C_S}{\partial \beta} \right)_0 + \Delta \left(\frac{\partial C_S}{\partial \beta} \right)_{e,l} + \Delta \left(\frac{\partial C_S}{\partial \beta} \right)_{e,r} \right] \beta \tag{4.12b}$$

$$C_L = C_{L_0} + \Delta C_{L_{e,l}} + \Delta C_{L_b} + \Delta C_{L_{e,r}} \tag{4.12c}$$

These coefficients depend on the angle of attack, the angle of sideslip, the deflections of the rudders, elevons and the body flap, the Mach number and the altitude. The moment coefficients are computed by:

$$C_l = \Delta C_{l_{e,l}} + \Delta C_{l_{e,r}} + \left(\frac{\partial C_l}{\partial \beta} \right)_0 \beta \tag{4.13a}$$

$$C_m = C_{m_0} + \Delta C_{m_{e,l}} + \Delta C_{m_b} + \Delta C_{m_{e,r}} \tag{4.13b}$$

$$\begin{aligned} C_n = {} & \Delta C_{n_{r,l}} + \Delta C_{n_{e,l}} + \Delta C_{n_{e,r}} + \Delta C_{n_{r,r}} \\ & + \left[\left(\frac{\partial C_n}{\partial \beta} \right)_0 + \Delta \left(\frac{\partial C_n}{\partial \beta} \right)_{r,l} + \Delta \left(\frac{\partial C_n}{\partial \beta} \right)_{e,l} + \Delta \left(\frac{\partial C_n}{\partial \beta} \right)_{e,r} + \Delta \left(\frac{\partial C_n}{\partial \beta} \right)_{r,r} \right] \beta \end{aligned} \tag{4.13c}$$

In these equations, the coefficients also depend on the control-surface deflections, the angle of attack and the Mach number. In Mooij (1995) tabulated values for the coefficient contributions can be found. These values are displayed for Mach numbers ranging from 1.20 to 20, for angles of attack ranging from 0° to 45° degrees and for altitude ranging from 0.0 to 20.0 km. All coefficient contributions are measured for two of these three variables, and by interpolation the values between the step sizes of these variables can be found. When an input value that is outside the aforementioned ranges is given, the last tabular value is used, as the behavior of the aerodynamic coefficients outside these bounds is unknown. For coefficient derivatives, the linear increase or decrease between two table entries is computed.

Using the aerodynamic tables, all coefficient contributions can be computed, which result in the final six aerodynamic coefficients that are used to calculate the aerodynamic forces and moments according to the equations stated in Section 3.4.

5

Guidance, Navigation, and Control

In this thesis the re-entry mission was defined as the trajectory from the atmospheric entry point to some target on Earth, where the vehicle has to land. However, if one would do nothing after de-orbiting, there remains very little or no control over the vehicle to get it to its desired location. Due to the requirement for a safe flight to the Earth's surface, a GNC-system is designed. Such a system consists of three modules, which are described in this chapter. Figure 5.1 shows how these three segments are related. In the ellipse-shaped boxes the type of trajectory of the vehicle is defined (either reference, which is defined in the software, or the actual trajectory, which is the path that is flown at that instant). The more rectangular boxes contain parts of the GNC-system. The two dashed rectangular boxes contain every component that either belongs to the navigation segment or the control segment. The guidance segment consists of the box with 'guidance laws' on it.

As the GNC system of a re-entry vehicle operates with a specific frequency one can distinguish a discrete system. At every time interval, sensors (that may operate with a different frequency) send measurement data to the state estimator in the navigation segment, which produces an estimate of all state variables. This estimated state is compared to a reference trajectory and using the guidance laws, which describe what the vehicle has to do to achieve this ideal reference trajectory, one is able to compute the so-called guidance commands. These commands are converted to signals that can actually be applied to the control hardware to adjust the trajectory of the vehicle in such a way that the desired trajectory is approximated.

In Section 5.1 the theory of the navigation segment will be described in full detail, as part of the research that is done concerns the different sensor combinations and state estimators that are used. The results of this research however are discussed in Chapter 9. Subsequently, a basic control system is designed to evaluate the robustness of the parts of the navigation module; this control system is discussed in Section 5.2. Its

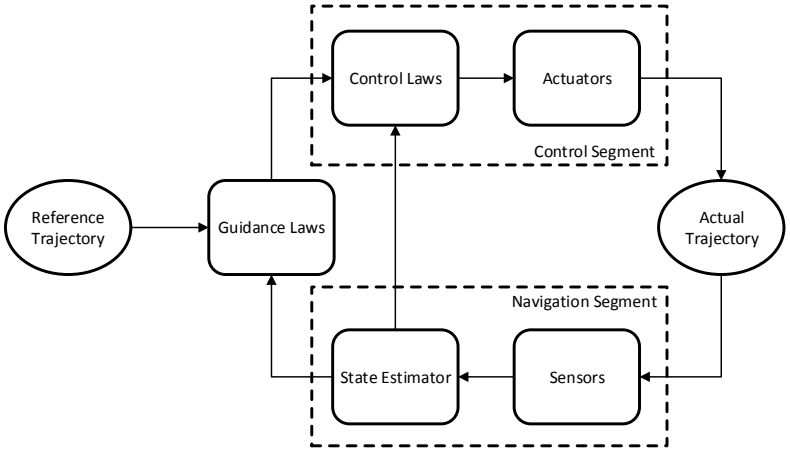


Figure 5.1: GNC-system architecture for an arbitrary re-entry vehicle.

implementation and results are shown in Chapter 7. As no research is done with respect to the guidance segment of the GNC system, the theory of the guidance module of the HORUS-2B vehicle is described in **??**.

5.1 Navigation

As was stated in the introduction of this chapter, in this section it is elaborated on the navigation segment, which is divided into two parts. First the sensors that were used in the simulator are discussed, where their error sources are touched upon. The measurements that are generated in the simulator are fed to the state estimator, for which the used methods are described in the latter part of this section.

5.1.1 Sensors

Earlier in this report the dynamics of the re-entry flight were discussed. These dynamics are used to describe the state of the vehicle. The actual exact state of the vehicle, however, cannot be known in real life, as error sources are always present. The navigation module is used to estimate the state and these error sources as accurate as possible so that the desired trajectory can be approached very closely. Measurements from the on-board sensors are used to aid in the process of accurate state and error estimation.

As a simulation of an arbitrary re-entry flight is considered, the absence of real life measurements exists, hence they have to be generated during the simulation. Using an IMU, values for non-gravitational acceleration and rotational rates will be obtained. These measurements are used in an initial prediction of the position, velocity and attitude of the vehicle, as will be described in Section 5.1.2. Then, the estimator will contain a correction step in which measurements from the IMU are again included to estimate the biases of this sensor. To increase accuracy on the state estimation, GPS measurements will be added to the correction step of the estimator. In this step the error source from the clock bias and drift can also be estimated. These two sensors in combination with an estimator provide a suitable system for the re-entry trajectory, as was concluded by Mooij and Chu (2002). A third sensor, the FADS can then be added, and due to the fact that this is a relatively new sensor its ease of implementation and its performance in a simulated re-entry flight are to be researched. The difference between the different sensor configurations (IMU, GPS and FADS) is discussed in Chapter 9.

IMU

The Inertial Measurement Unit is a so-called inertial sensor and consists of a set of accelerometers and gyroscopes (Mooij, 2015). The former ones are used to measure the translational motion of the object and are mostly comprised of a proof-mass that moves in a closed environment, in which the accelerations of that mass in the different directions are measured. Gyroscopes then are used for the estimation of the rotational motion of the object, in which the conservation of angular momentum is used. As for the GPS sensors, also for the IMUs several errors can occur, which are described and visualized by Grewal et al. (2001). The following error models can be defined:

- A constant error can occur that gives a non-zero output when there is a zero input, a so-called bias.
- A scale factor error may be introduced, that enlarges or diminishes the output with respect to the output that should be generated.
- Nonlinearities can occur, which causes inaccuracies due to the linearized modeling of the measurements.
- A so-called scale-factor sign asymmetry can occur, which is a difference in output scaling for the negative and positive parts of the output variables.
- A dead zone may be introduced, for which no output value is defined for specific input values.
- Quantization errors occur in every digital system and are basically related to the number of steps that are used to quantize the signal. The exact value is always rounded to one of the quantization steps and therefore (small) errors are introduced.

These error sources are visualized in Figure 5.2, where the dashed line represents the true value and the solid line states the behavior of the measured value due to the error source. Mooij (2015) then adds three more error sources, of which only the first two are visualized, which are shown in Figure 5.3:

- Saturation points: these inaccuracies occur when the maximum or minimum capabilities of the sensor are reached. The output value then can not handle the extreme input values and will most likely show the maximum or minimum output value the sensor can provide.

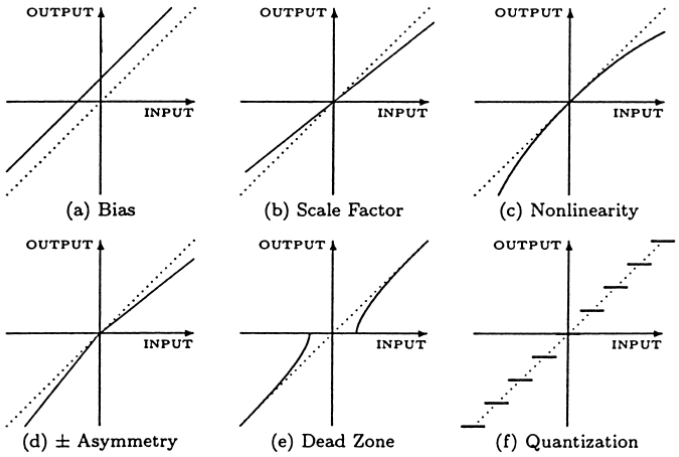


Figure 5.2: Error models belonging to the IMU sensors. Note that any combination of these errors can occur within the sensor, as described by Grewal et al. (2001).

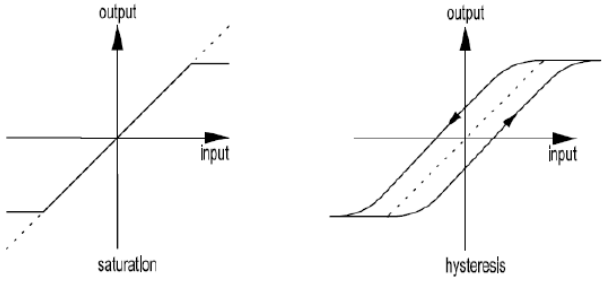


Figure 5.3: Two additional error models belonging to the IMU sensors (Mooij, 2015).

- Hysteresis: this type of error occurs when the output value of the sensor does not only depend on the input value at the current time but also at a previous time step.
- Misalignments: these errors occur when the axis of the sensor are not perfectly oriented perpendicular to each other.

When modeling the IMU in the simulator, it is chosen to use only three of the above mentioned inaccuracies: the sensor bias, the scale factor and the axis misalignments. In the simulator, the sensor is implemented and measurements are generated in every time step. For the accelerometer, this is done via

$$\mathbf{a}_m = (\mathbf{I} + \mathbf{S}_a)\,\mathbf{a} + \mathbf{b}_a + \boldsymbol{\nu}_a \tag{5.1}$$

with

$$\mathbf{S}_a = \begin{bmatrix} s_x & m_{xy} & m_{xz} \\ m_{yx} & s_y & m_{yz} \\ m_{zx} & m_{zy} & s_z \end{bmatrix} \tag{5.2}$$

In these equations, $\mathbf{b}_a$ represents the bias vector with components over three axes and the matrix $\mathbf{S}_a$ contains the scaling factors per axis (s_x, s_y and s_z) and the misalignments from one axis relative to another one (m_{xy}, m_{xz}, m_{yx}, m_{yz}, m_{zx} and m_{zy}). Next to that the a three-component noise vector is added, which is represented by $\boldsymbol{\nu}_a$. The accelerometer is able to measure the external forces that act upon the surface of the vehicle, which excludes gravitational force. As everything related to the measurements of the IMU happens on-board, the accelerations are expressed in the body frame. When the state estimation is done they have to be transformed into the inertial frame. The gyroscope measures the rotational rate of the body frame with respect to the inertial frame, and has a similar set up:

$$\boldsymbol{\omega}_m = (\mathbf{I} + \mathbf{S}_\omega)\,\boldsymbol{\omega} + \mathbf{b}_\omega + \boldsymbol{\nu}_\omega \tag{5.3}$$

and

$$\mathbf{S}_\omega = \begin{bmatrix} s_p & m_{pq} & m_{pr} \\ m_{qp} & s_q & m_{qr} \\ m_{rp} & m_{rq} & s_r \end{bmatrix} \tag{5.4}$$

In these equations, $\mathbf{b}_\omega$ represents the gyroscope drift, the s parameters represent the scaling factors, the m parameters represent the misalignments and the noise is again defined by $\boldsymbol{\nu}_\omega$. For both the accelerometer and the gyroscope, the scaling factors, misalignments and the accelerometer bias and gyroscope drift are estimated by ground calibration. In the simulator, these ground-calibrated values are taken as the real IMU errors, where some of these are estimated during flight using the navigation filter, which will be described in Chapter 9.

As stated earlier in this section, the IMU measurements are generated in the simulator, where the above equations are used and where constant values are taken for the error sources, representing the ground calibrations. The sources that have a significant effect on the state estimation are included in the process, which will be described later on in this section. To ensure that the IMU measurements resemble reality as accurate as possible, white noise is added.

In real-life applications the IMU measurements are used for basic estimation of the position, velocity and attitude of the vehicle. To ensure more accurate estimation, which is required, one needs to implement another sensing system, which are in this case the GPS and the FADS. In Chapter 8 it will be shown that the IMU is not sufficient for accurate state estimation by itself, and therefore the use of GPS or a similar sensor is required. At some point in the re-entry phase, the vehicle will pass through a part of the atmosphere where no transmissions from and to the vehicle can occur, which is the so-called blackout phase. In this part of the trajectory GPS signals are not received and the pressure is insufficient to produce trustworthy FADS measurements and thus, in this phase the IMU will operate as a stand-alone sensor.

In this subsection only the theory of the IMU was discussed; the implementation, verification and sensor results will be described in Section 8.1.1.

GPS

The GPS (Global Positioning System) is a system which consists of between 24 and 32 satellites that orbit the Earth in five different planes (Grewal et al., 2001). When a vehicle is in an arbitrary position around the Earth and visible by at least four of these GPS satellites, it is possible to compute the position of the vehicle. The system is a so-called one way ranging system, which means that a signal is only sent from a GPS satellite to the vehicle. When a GPS receiver is on-board, the signals that are received can be evaluated on the pseudorange, the Doppler shift from the carrier frequency and the carrier phase or carrier-phase difference from different GPS receivers (Mulder et al., 1999).

As for the IMU, also for the GPS a distinction is made between some error sources. The first one lies in the clocks of both the GPS satellite and the GPS receiver that is on-board the vehicle. According to Tapley et al. (2004), due to general relativity, clocks that are located far from the center of gravity of a massive body run faster than clocks that are located nearby (for example, at the surface). This is called time dilation and has a fractional influence on the measurements that are obtained. The second error source has its origin in the signals that are sent from the GPS satellites to the receiver. When a re-entry case is evaluated, these signals have to travel through (part of) the atmosphere, which means that they are slightly modified due to particles in the air. The two main sources of signal modification are due to the:

- **Troposphere** - In the troposphere, which lies in the lower part of the atmosphere, the density of the air is highest and thus there are more particles due to which the signal can be refracted. Due to these refractions the signal that is received by the vehicle contains a certain error which is defined by

$$\delta\rho_{\text{trop}} = 10^{-6} \int N ds \tag{5.5}$$

  where N is the refractivity, which depends on the index of refraction of the particles the signal encounters. The ds component of the integral is basically a very small length along the path the signal travels (Tapley et al., 2004).

- **Ionosphere** - The ionosphere, which is positioned somewhat higher in the Earth's atmosphere, contains ionized particles and electrons. When a signal is sent with non-optical frequencies, it will be affected by

the ions and electrons, related by the total electron content and the frequency of the signal (Tapley et al., 2004). This error source, however, can be canceled by using dual-frequency measurements (Zhang et al., 1999).

Thirdly, there is a relatively small uncertainty in the position and velocity of the GPS satellites themselves, which introduces another small error to the measurements. As was stated, at least four GPS satellites are required to approximate the position and velocity of the vehicle, however, the constellation of these satellites is of importance as well. When the used satellites are spread across the sky the uncertainty in calculated position will be less than when they are 'close to each other' from the vehicle point of view. This is called Dilution of Precision (DOP) and will be evaluated further on in this section. Finally, the evaluation of the errors that occur on the vehicle itself is done. Tapley et al. (2004) describes that a signal can reach the antenna directly, but also by bouncing off of another part of the vehicle. When both of these signals (that come from the same signal) are picked up by the antenna, it is denoted as multipath. Next to that, it is found that noise occurs in both the antennas and in the receiver itself.

Pseudorange
The range ρ is the distance between the vehicle and the GPS satellite and can simply be computed by the general formula for any distance between two objects. However, the position of the vehicle is in this case unknown, so a different method has to be used, which uses the clocks that are on-board the vehicle and the GPS satellite. The range is then found by:

$$\tilde{\rho} = c\,(t_R - t_T) \tag{5.6}$$

where c is the speed of light in vacuum, t_R and t_T are the receiver clock-time and the transmitter clock-time respectively and $\tilde{\rho}$ is the so-called pseudorange. It is not the true range, as the clocks may not be synchronized and thus errors will occur, such as an initial offset, a clock drift or other errors due to for example stochastic components (Tapley et al., 2004).

According to Mooij (2015) and Mulder et al. (1999), the pseudorange of a signal from GPS satellite i to receiver can be computed via

$$\tilde{\rho} = c\left(t_r - t_{s_i}\right) = \sqrt{\left(\mathbf{r}_{\text{GPS,i}} - \mathbf{r}\right)^T \left(\mathbf{r}_{\text{GPS,i}} - \mathbf{r}\right)} + c \cdot b_c + v_{\rho_i} \tag{5.7}$$

where $\mathbf{r}$ contains the position of the vehicle, $\mathbf{r}_{\text{GPS,i}}$ contains the position of GPS satellite i, c is the speed of light ($2.99792458 \cdot 10^8$ m/s, according to Mooij (2015)), b_c denotes the clock bias and v_{ρ_i} is the noise component.

Doppler shift
When an object is moving over the line-of-sight of the observer, a signal that is sent with a certain frequency will not be received with that frequency due to an effect that is called a Doppler shift. According to Mulder et al. (1999) this Doppler shift is related to the range rate via:

$$\Delta f = f_s - f_r = \frac{\dot{\tilde{\rho}} f_s}{c} \tag{5.8}$$

where Δf is the Doppler shift (the difference in transmitted and received frequency) and $\dot{\tilde{\rho}}$ is time derivative of the pseudorange that was described above; the range rate. Using Equation (5.7) this parameter can be defined:

$$\dot{\tilde{\rho}} = \frac{\left(\mathbf{r}_{\text{GPS,i}} - \mathbf{r}\right)^T \left(\dot{\mathbf{r}}_{\text{GPS,i}} - \dot{\mathbf{r}}\right)}{\sqrt{\left(\mathbf{r}_{\text{GPS,i}} - \mathbf{r}\right)^T \left(\mathbf{r}_{\text{GPS,i}} - \mathbf{r}\right)}} + c \cdot d_c + v_{\dot{\rho}} \tag{5.9}$$

where $\dot{\mathbf{r}}$ is the velocity vector of the vehicle, $\dot{\mathbf{r}}_{\text{GPS,i}}$ is the velocity vector of GPS satellite i, d_c is the receiver clock drift and $v_{\dot{\rho}}$ is again a noise component.

Carrier phase
The aforementioned measurements that are done using GPS are used to estimate the position and the velocity of the re-entry vehicle. However, with a third set of measurements, its attitude can be estimated as well. This is done using carrier phase measurements and for this the layout of the receiver within the vehicle has to be defined. As is described in Mooij and Chu (2002) the HORUS 2B reference vehicle is equipped with four antennas, of which one is the master antenna and the other three are so-called slave antennas. Then, when a signal is received from a GPS satellite, it is picked up by all four antennas, and using this the difference in phase can be measured, which is shown in Figure 5.4.

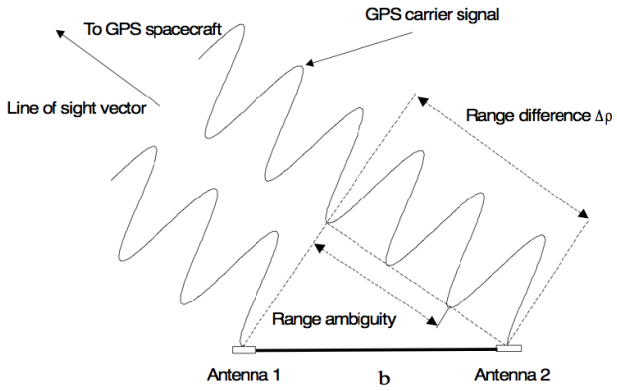


Figure 5.4: Visualization of the carrier phase difference for two different antennas, which are in this case located on the vehicle (Mulder et al., 1999).

It can be seen that the wave on the right travels further and that the received waves may be different in phase. This is measured for all three slave antennas with respect to the master antenna, where the measurements can be computed via

$$\Phi_m(t_r) = \Phi_{CR}(t_r) - \Phi_0(t_r) - \frac{f_s}{c}\left(\tilde{\rho} - \dot{\tilde{\rho}} b_c\right) + N + \upsilon_\Phi \tag{5.10}$$

where Φ_m is the measured phase, Φ_{CR} is the phase that is received, Φ_0 is a specific reference phase, f_s is the carrier frequency, c is the speed of light, $\tilde{\rho}$ and $\dot{\tilde{\rho}}$ are the range and range rate respectively, b_c is the clock bias, N is the integer ambiguity and υ_Φ is a noise component. The measurements that are used in the estimator however are carrier phase difference measurements, for which the layout of the antennas on the vehicle is to be defined, as was stated earlier. In Mooij and Chu (2002) it was defined that the master antenna has body frame coordinates $\mathbf{r}_m = [-3, 0, -2.7]^T$, whereas the three slave antennas have coordinates:

- $\mathbf{r}_{s1} = [4, 0, -2.5]^T$
- $\mathbf{r}_{s2} = [-3, 1, -2.5]^T$
- $\mathbf{r}_{s3} = [-3, -1, -2.5]^T$

The baseline vector $\mathbf{b}$ that is shown in Figure 5.4 then is defined as the difference between master and slave antenna positions. The carrier phase difference is computed by

$$\Delta\Phi_{ij} = -\frac{f_s}{c}\mathbf{b}_{ij_B}^T \mathbf{C}_{B,I}\mathbf{e}_{i,I} + \Delta N + \beta + \upsilon_\Phi \tag{5.11}$$

where $\mathbf{C}_{B,I}$ is the transformation matrix from inertial to body frame. The carrier phase difference is measured for GPS satellite i and slave antenna j. In the simulator, the electrical line bias β will be neglected and it is assumed that the integer ambiguity ΔN is solved. The line-of-sight unit vector $\mathbf{e}_{i,I}$ is computed by

$$\mathbf{e}_{i,I} = \frac{\mathbf{r}_{\text{LOS,i}}}{\sqrt{\left(\mathbf{r}_{\text{GPS,i}} - \mathbf{r}\right)^T \left(\mathbf{r}_{\text{GPS,i}} - \mathbf{r}\right)}} \tag{5.12}$$

where the line of sight is

$$\mathbf{r}_{\text{LOS,i}} = \mathbf{r}_{\text{GPS,i}} - \mathbf{r} \tag{5.13}$$

which concludes the theory behind the GPS measurements. Note that for every visible GPS satellite, the vehicle can retrieve five measurements: pseudorange, range rate and three carrier phase difference measurements.

Finally, the concept of Dilution of Precision (DOP) is to be evaluated. It was stated that at least four satellites have to be visible to determine the vehicle's position and velocity, as the fourth satellite is required to solve for the time dependency. In Figure 5.5 the effect of the distribution of the visible satellites across the sky (from point of view of the vehicle) is shown and it is seen that when the satellites appear to be close to each other, as

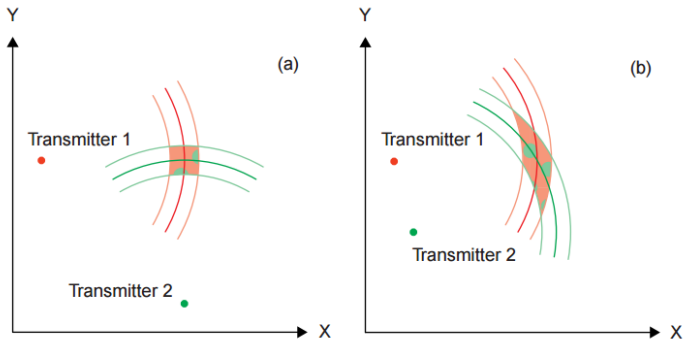


Figure 5.5: Representation of the effect of the constellation of the visible satellites in the sky.[1]

seen on the right side, the uncertainty in position determination is larger than for the case where the satellites are far from each other, as is seen on the left side of Figure 5.5.

In mathematical terms, the Geometric Dilution of Precision (GDOP) can be computed, which is a measure for the uncertainty of the position determination of the receiver. According to Spilker (1994), the DOP values can be computed via

$$\mathbf{A} = \begin{bmatrix} \frac{x_a - x}{R_a} & \frac{y_a - y}{R_a} & \frac{z_a - z}{R_a} & -1 \\ \frac{x_b - x}{R_b} & \frac{y_b - y}{R_b} & \frac{z_b - z}{R_b} & -1 \\ \frac{x_c - x}{R_c} & \frac{y_c - y}{R_c} & \frac{z_c - z}{R_c} & -1 \\ \frac{x_d - x}{R_d} & \frac{y_d - y}{R_d} & \frac{z_d - z}{R_d} & -1 \end{bmatrix} \tag{5.14}$$

where x_i, y_i and z_i are the coordinates of GPS satellite i; x, y and z are the coordinates of the receiver, and R_i is the distance between GPS satellite i and the receiver. Then:

$$\mathbf{Q} = \left(\mathbf{A}^T \mathbf{A}\right)^{-1} = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{xz} & \sigma_{xt} \\ \sigma_{xy} & \sigma_y^2 & \sigma_{yz} & \sigma_{yt} \\ \sigma_{xz} & \sigma_{yz} & \sigma_z^2 & \sigma_{zt} \\ \sigma_{xt} & \sigma_{yt} & \sigma_{zt} & \sigma_t^2 \end{bmatrix} \tag{5.15}$$

The diagonal of $\mathbf{Q}$ now contains four values, the first three represent the uncertainties for the x, y and z coordinates respectively whereas the fourth value represents the uncertainty with respect to time. The Geometric Dilution of Precision (GDOP) can then be defined by:

$$\text{GDOP} = \sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_z^2 + \sigma_t^2} \tag{5.16}$$

The GDOP should be as low as possible, as this represents the smallest uncertainty in position and time determination. Note that the GDOP can also be computed for more than four satellites. In Chapter 9 a dependency will be shown of the number of satellites that are used with respect to the GDOP value. Note that in the research that is done on the estimators, the minimum number of GPS satellites will be used.

Similar to the IMU, also for the GPS only the theory of the sensor is handled in this subsection. The implementation, verification and sensor discussion is done in Section 8.1.2.

FADS

Next to the IMU and GPS, the FADS will be evaluated, which was developed at the end of the previous century. Part of the research that is done is with respect to implementation of this sensor and the constraints, difficulties, but also advantages it provides. In previous missions, the atmospheric data, or the so-called air-data were measured using intrusive systems such as pitot tubes (Siemers III, 1983). Cobleigh et al. (1999) defined a non-intrusive sensing system concept which is called the FADS system. This concept, which is already windtunnel-tested on several space-planes, such as the X-33, X-34 and X-38, is dependent on surface pressures that are measured on the front-side of the body. A mathematical algorithm then relates these pressures to the state of the air-data. On the Space Shuttle, a variant of the system was applied that used a matrix of orifices on the nose of the vehicle, which provided the surface pressure observations (Hillje and Nelson, 1981).

[1] `http://www2.unb.ca/gge/Resources/gpsworld.may99.pdf` - retrieved: 18-10-2016

The pressure model that is used for the FADS system is related to a simple spherical object in a flow, for which the flow characteristics can be computed. This can be modeled as:

$$C_p(\theta) = \frac{p(\theta) - p_\infty}{q_c} \tag{5.17}$$

in which p_∞ is the free-stream static pressure, p is the local pressure as a function of angle θ and q_c is the dynamic impact pressure. For modified Newtonian flow theory for blunt objects in hypersonic flow, it is found that (Cobleigh et al., 1999):

$$C_p(\theta) = \frac{p_{t_2} - p_\infty}{\bar{q}} \cos^2\theta \tag{5.18}$$

where now the dynamic pressure $\bar{q}$ is also at an undisturbed free point. The incidence angle θ can be written as a function of the local angle of attack, the sideslip angle and the local clock angle λ and cone angle ϕ of the nose cap. Hereby is λ the angle from the point on the cone with respect to its radial symmetry axis and ϕ the radial angle from a specific reference point. The relation is written as:

$$\cos\theta = \cos\alpha_e \cos\beta_e \cos\lambda + \sin\beta_e \sin\phi \sin\lambda + \sin\alpha_e \cos\beta_e \cos\phi \sin\lambda \tag{5.19}$$

which is different for every sensor port, as the surface coordinates differ per port. Then, the equations for the pressure coefficient are combined to:

$$C_p(\theta) = X\left(M_\infty, \alpha_e, \beta_e\right) + Y\left(M_\infty, \alpha_e, \beta_e\right)\cos^2\theta \tag{5.20}$$

Stating that at $\theta = 0$ this equation has to satisfy subsonic and supersonic conditions, the final formula for the pressure as a function of the incidence angle θ can be obtained (Cobleigh et al., 1999)

$$p(\theta) = q_c\left[\cos^2\theta\left(\alpha_e, \beta_e\right) + \epsilon\left(M_\infty, \alpha_e, \beta_e\right)\sin^2\theta\left(\alpha_e, \beta_e\right)\right] + p_\infty \tag{5.21}$$

where θ is the incidence angle between the vector normal to the surface orifice and the local velocity vector, q_c is the impact pressure, ϵ is a calibration parameter, p_∞ is the free-stream static pressure, α_e is the effective angle of attack, β_e is the effective angle of sideslip and M_∞ is the free-stream Mach number. Note that θ, q_c and ϵ are functions of α_e, β_e and M_∞.

The built-in solution algorithm for the FADS sensor then consists of different estimators for the different parameters. First, the angle of attack is considered, which is found by:

$$\text{for} |\alpha_e| \leq 45^\circ, \alpha_e = \frac{1}{2}\arctan\left(\frac{A}{B}\right) \tag{5.22a}$$

$$\text{for} |\alpha_e| > 45^\circ, \alpha_e = \frac{1}{2}\left[\pi - \arctan\left(\frac{A}{B}\right)\right] \tag{5.22b}$$

where A and B are defined as:

$$A = \Gamma_{ik}\sin^2\lambda_j + \Gamma_{ji}\sin^2\lambda_k + \Gamma_{kj}\sin^2\lambda_i \tag{5.23a}$$

$$B = \Gamma_{ik}\cos\phi_j \sin\lambda_j \cos\lambda_j + \Gamma_{ji}\cos\phi_k \sin\lambda_k \cos\lambda_k + \Gamma_{kj}\cos\phi_i \sin\lambda_i \cos\lambda_i \tag{5.23b}$$

where $\Gamma_{xy} = p_x - p_y$ and denotes the difference between two port pressures. The i, j and k subscripts represent three different ports and therefore this algorithm is called the 'triples' algorithm.

The angle of sideslip is estimated in a similar matter:

$$A'\tan^2\beta_e + 2B'\tan\beta_e + C' = 0 \tag{5.24}$$

where:

$$A' = \Gamma_{ik}b_j^2 + \Gamma_{ji}b_k^2 + \Gamma_{kj}b_i^2 \tag{5.25a}$$

$$B' = \Gamma_{ik}a_j b_j + \Gamma_{ji}a_k b_k + \Gamma_{kj}a_i b_i \tag{5.25b}$$

$$C' = \Gamma_{ik}a_j^2 + \Gamma_{ji}a_k^2 + \Gamma_{kj}a_i^2 \tag{5.25c}$$

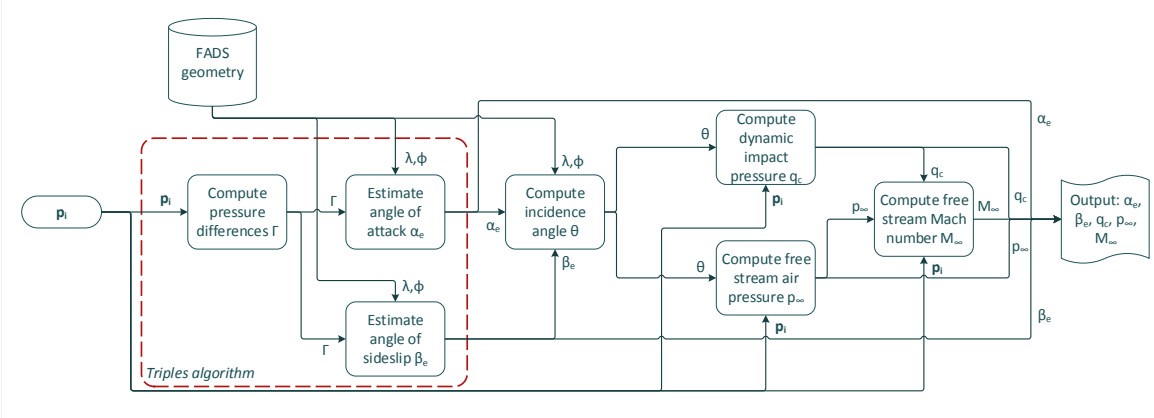


Figure 5.6: Schematic overview of the built-in FADS estimator.

and

$$a_x = \cos\alpha_e \cos\lambda_x + \sin\alpha_e \sin\lambda_x \cos\phi_x \tag{5.26a}$$

$$b_x = \sin\lambda_x \sin\phi_x \tag{5.26b}$$

The Mach number is then estimated using the impact pressure and free-stream static pressure via:

$$\frac{q_c}{p_\infty} = \left[1 + 0.2M^2\right]^{3.5} - 1 \tag{5.27}$$

for subsonic flow. For supersonic flow it is found that:

$$\frac{q_c}{p_\infty} = 166.92M_\infty^2 \left[\frac{M_\infty^2}{7M_\infty^2 - 1}\right]^{2.5} - 1 \tag{5.28}$$

This supersonic equation is solved using a Taylor's series expansion and reduces to:

$$M_\infty = \frac{\sqrt{\sum_{i=0}^{9} r_i \left(W^i\right)}}{W} \tag{5.29}$$

where $W = 1.839371\left(\frac{p_\infty}{p_t}\right)$. The summation coefficients r_i are summarized in the following vector:

$$\mathbf{r} = [1.42857, -0.357143, -0.0625, -0.025, -0.012617, -0.00715, -0.0043458, 0, 0, -0.0087725]^T \tag{5.30}$$

Using the previously mentioned equations the flight conditions can be approximated by the FADS. This built-in estimator is summarized for implementation purposes in Figure 5.6. This figure shows that the pressure measurements are used in combination with the triples algorithm and the pressure port locations to estimate the angle of attack and angle of sideslip. Using these aerodynamic angles the incidence angle θ is computed, which is used, together with the pressure measurements, to compute the free stream air pressure and dynamic impact pressure. Finally, the free stream Mach number can be estimated, after which an output vector of the air data variables (α, β, p_∞, q_c, M_∞) is produced.

In most cases five or more measured pressures from Equation (5.21) are required to obtain the airdata state vector, containing q_c, p_∞, α and β (Cobleigh et al., 1999), from which the other relevant parameters such as the Mach number and pressure altitude can be calculated. The last airdata variable, the true airspeed, has to be measured using the free-stream temperature, as the speed of sound depends on this variable, according to Equation (3.13). The speed of sound and the Mach number are then sufficient to compute the true airspeed.

As for the other two sensor types, also the FADS system contains error sources, which are thoroughly explained in Whitmore and Moes (1994). First of all, there is the calibration of the compressibility factor ϵ, which, according to Cobleigh et al. (1999), nears zero for high Mach numbers. The calibration is done iteratively using

$$\epsilon\left(M_\infty, \alpha_e, \beta_e\right) = \frac{\sum_{i=1}^{N} \sin^2\theta_i \left[C_{p_c}(\theta_i) - \cos^2\theta_i\right]}{\sum_{i=1}^{N} \sin^4\theta_i} \tag{5.31}$$

where i denotes the pressure port, and N defines the total number of pressure ports. This equation provides the least-squares estimate of the compressibility factor. It is also stated that this value may be approximated even more accurate, which is done iteratively and is described in Cobleigh et al. (1999). Then, two errors in angle of attack and angle of sideslip can be defined, denoted as the upwash and sidewash corrections $\delta\alpha$ and $\delta\beta$ respectively. These parameters near constant values for high Mach numbers, making them a bias for the true value (Cobleigh et al., 1999). They can be defined as

$$\delta\alpha\left(M_\infty, \alpha_e\right) = \alpha_e - \alpha_{\text{true}} \tag{5.32a}$$

$$\delta\beta\left(M_\infty, \beta_e\right) = \beta_e - \beta_{\text{true}} \tag{5.32b}$$

and are to be calibrated using windtunnel tests.

Subsequently, in Whitmore and Moes (1994) it is described that the pressure ports that are used for the measurements can be misaligned over both the clock angle λ and the cone angle ϕ. Furthermore, Whitmore and Moes (1994) describe some more complicated error sources, which will briefly be touched upon. The viscous interaction error is an inaccuracy that is induced by the fact that for high velocities, the boundary layer thickness may be increased due to viscousness of the air flow, leading to uncertainties in the measured pressures.

$$\frac{\partial P_{\text{viscous}}}{P_{\text{inviscid}}} = 1 + 0.31\bar{\chi} + 0.05\bar{\chi}^2 \text{ for weak interactions} \tag{5.33a}$$

$$\frac{\partial P_{\text{viscous}}}{P_{\text{inviscid}}} = 0.514\hat{\chi} - .0.241 \text{ for strong interactions} \tag{5.33b}$$

The parameter $\hat{\chi}$ is the so-called similarity parameter and is a function of Mach number, Reynolds number and the specific heat ratio of the air. This error is negligible for low Mach numbers and has a value of roughly 0.1 for Mach numbers of 12, and thus is so small that modeling this error is beyond the scope of this thesis.

Then, when the hardware of the sensing system is evaluated, it is found that the pressure ports are connected to transducers by tubes. In these tubes the air stream experiences drag with respect to the walls of the tube, thereby modifying the air stream that reaches the transducer, which is called pneumatic lag:

$$\frac{P_L}{P_0} = \frac{1}{\frac{1}{\omega_n^2}s^2 + 2\frac{\xi}{\omega_n}s + 1} \tag{5.34}$$

This equation shows the ratio of the measured pressure at the transducer and at the surface of the pressure port. It is modeled as a transfer function of the second order, where the damping ratio ξ and natural frequency ω_n are included. These parameters depend on the tube, transducer, and pressure port characteristics. Whitmore and Moes (1994) modeled this error and found that the effects are negligible, both for high and for small Mach numbers, as the magnitude of this error does not exceed 0.05 for the Mach number and 0.05° for the angle of attack.

Next to that, due to the high velocity of the air stream to the tubes, thermodynamic effects may also modify the measured stream, which is referred to as the thermal transpiration error:

$$\frac{\partial P(x)}{\partial T(x)} = \frac{1}{2}\frac{P(x)}{T(x)}\left[\frac{\kappa(\kappa + a)}{\kappa^2 + (a+b)\kappa + bc}\right]^2 \tag{5.35}$$

where a, b, and c are empirical constants and κ represents Knudsen's number, which is a function of the pressure and temperature in the tube and the tube diameter. The pressure P and temperature T that are described in this equation are a function of position x in the tube, and can be computed via integration, as this equation defines a differential equation. In Whitmore and Moes (1999) it is stated that the thermal transpiration error has a negligible influence on the angle of attack and increases for high Mach numbers, however, the error magnitude remains below 0.05 for the Mach number.

These thermodynamic effects may also cause structural deformation of the hardware due to high temperatures, however, as this error is much smaller than the previously described errors, it can be discarded directly.

Finally, it is found that the transducers of the FADS may also contain inaccuracies due to quantization, which completes the error model of the FADS. Of all the aforementioned errors the magnitude is small, especially for low Mach numbers, as is seen in Whitmore and Moes (1994). Only the angle of attack error due to port misalignment is denoted as significant for low Mach numbers, and therefore this error is evaluated further on in this thesis. Modeling and evaluating the other described errors is beyond the scope of this thesis, but is recommended for future work.

Again, in this subsection only the theory of the FADS is described. The implementation and verification, as well as the results of the built-in estimator are discussed in Section 8.1.3.

5.1.2 Estimators

As was seen in Figure 5.1 the second block that belongs to the navigation module is the estimator block. During flight, different sensors measure the variables that were discussed in the previous paragraph with a sensor-specific operating frequency. The estimator then operates with its own frequency and propagates an initially set state over time. The main focus of the estimator is to approach the real state as close as possible, which is done by also focusing on determination of error sources and their magnitudes. Three different estimators are used in this report, as part of the research concerns the differences in functionality and performance of these estimators.

Extended Kalman Filter

The Extended Kalman Filter (EKF) is a member of the Kalman filter family and it has a close relation to the Linear Kalman Filter (LKF), which simply cannot be applied to most real-life problems, as it can only handle linear systems. The EKF can be applied to nonlinear systems, as it uses a first-order Taylor approximation around the estimate of the previous time step. For the derivation of the theory of the EKF, a summary that is described in Welch and Bishop (2001), Tapley et al. (2004) and Mooij (2015) is provided, where it is started with the definition of a nonlinear difference equation

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}) \tag{5.36}$$

with measurements

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k, \mathbf{v}_k) \tag{5.37}$$

where $\mathbf{x}$ is the state, $\mathbf{u}$ is the input vector, $\mathbf{z}$ is the measurement vector and $\mathbf{v}$ and $\mathbf{w}$ are noise vectors. Subscript k denotes the current time step k, whereas $k-1$ denotes the previous time step. The filter is then divided into two parts, prediction and correction. In the prediction step, the so-called a-priori estimate is computed:

$$\hat{\mathbf{x}}_k^- = \mathbf{f}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}, \mathbf{0}) \tag{5.38}$$

where the $\hat{\mathbf{x}}$ denotes that an estimate is considered and where the superscript minus denotes that this estimate is done a-priori. It is seen that for the a-priori estimate of the state at time step k, the a-posteriori estimate from time step $k-1$ is used. Subsequently, using the measurement equations, an estimate of the observations is computed as well:

$$\hat{\mathbf{z}}_k = \mathbf{h}\left(\hat{\mathbf{x}}_k^-, \mathbf{0}\right) \tag{5.39}$$

Then, according to Welch and Bishop (2006), equations for the state and the observations can be defined by linearizing an estimate about Equations (5.38) and (5.39).

$$\mathbf{x}_k \approx \hat{\mathbf{x}}_k^- + \mathbf{A}_k(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}) + \mathbf{W}_k\mathbf{w}_{k-1} \tag{5.40}$$

$$\mathbf{z}_k \approx \hat{\mathbf{z}}_k + \mathbf{H}_k\left(\mathbf{x}_k - \hat{\mathbf{x}}_k^-\right) + \mathbf{V}_k\mathbf{v}_k \tag{5.41}$$

Equation (5.40) and Equation (5.41) contain the state and observation vectors of the current time step $\mathbf{x}_k$ and $\mathbf{z}_k$, but also the a-posteriori estimation $\hat{\mathbf{x}}_k$ and the a-priori estimations $\hat{\mathbf{x}}_{k+1}^-$ and $\hat{\mathbf{z}}_k$. The noises that are included in these equations can be related to covariance matrices, as is stated in Welch and Bishop (2006) and Mooij (2015). In this case, the notation from Mooij (2015) is used.

$$p(\mathbf{v}_k) \sim N(0, \mathbf{R}_k) \tag{5.42a}$$

$$p(\mathbf{w}_k) \sim N(0, \mathbf{Q}_k) \tag{5.42b}$$

In these two equations it can be seen that both the process noise as the measurement noise are modeled as probability functions with a normal distribution. Matrices $\mathbf{R}_k$ and $\mathbf{Q}_k$ are the measurement noise and process noise matrices respectively and are diagonal, so that for each probability element of vectors $\mathbf{v}_k$ and $\mathbf{w}_k$ a normal distribution holds with a mean of zero and a variance from the matrices $\mathbf{R}_k$ and $\mathbf{Q}_k$.

Next to that, four matrices are used in Equation (5.40) and Equation (5.41), which are called Jacobians. These matrices contain derivatives of functions **f** and **h** from Equation (5.38) and Equation (5.39) with respect to the state variables and the noise variables. In mathematical terms:

$$\mathbf{A}_k = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{(\hat{\mathbf{x}}_k, \mathbf{u}_k, \mathbf{0})} \tag{5.43a}$$

$$\mathbf{W}_k = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{w}}\right|_{(\hat{\mathbf{x}}_k, \mathbf{u}_{k+1}, \mathbf{0})} \tag{5.43b}$$

$$\mathbf{H}_k = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\left(\hat{\mathbf{x}}_k^-, \mathbf{0}\right)} \tag{5.43c}$$

$$\mathbf{V}_k = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{v}}\right|_{\left(\hat{\mathbf{x}}_k^-, \mathbf{0}\right)} \tag{5.43d}$$

In Equation (5.40) and Equation (5.41) approximations for the state and the measurements were made, which induces errors in both estimations. These errors are defined as:

$$\hat{\mathbf{e}}_{xk}^- \equiv \mathbf{x}_k - \hat{\mathbf{x}}_k^- \approx \mathbf{A}_{k-1}\left(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}\right) + \boldsymbol{\epsilon}_k \tag{5.44a}$$

$$\hat{\mathbf{e}}_{zk}^- \equiv \mathbf{z}_k - \hat{\mathbf{z}}_k \approx \mathbf{H}_k\left(\hat{\mathbf{e}}_{xk}^-\right) + \boldsymbol{\eta}_k \tag{5.44b}$$

In these equations ϵ and η can be found, which are linear approximations of the noise parameters that were mentioned earlier. According to Mooij (2015) these are defined as:

$$p\left(\boldsymbol{\epsilon}_k\right) \sim N\left(\mathbf{0}, \mathbf{W}_{k-1}\mathbf{Q}_{k-1}\mathbf{W}_{k-1}^T\right) \tag{5.45a}$$

$$p\left(\boldsymbol{\eta}_k\right) \sim N\left(\mathbf{0}, \mathbf{V}_k\mathbf{R}_k\mathbf{V}_k^T\right) \tag{5.45b}$$

Equation (5.45) states that also the linearized parts of the noise parameters are probability functions with a normal distribution. Finally, the state estimation can be updated, using **K** as the Kalman gain (Welch and Bishop, 2006):

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k\left(\mathbf{z}_k - \hat{\mathbf{z}}_k^-\right) \tag{5.46}$$

where **K** is defined as:

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}_k^T\left(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{V}_k\mathbf{R}_k\mathbf{V}_k^T\right)^{-1} \tag{5.47}$$

Covariance matrix **P** can be propagated via:

$$\mathbf{P}_k^- = \mathbf{A}_k\mathbf{P}_{k-1}\mathbf{A}_k^T + \mathbf{W}_k\mathbf{Q}_{k-1}\mathbf{W}_k^T \tag{5.48}$$

where

$$\mathbf{P}_k = \left(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\right)\mathbf{P}_k^- \tag{5.49}$$

To summarize and clarify all used equations from this paragraph Algorithm 1 is provided, as is done in a figure from Welsh and Bishop (2006). This algorithm was visualized for easy implementation of the EKF. Note that this paragraph only described the theory behind the EKF, the implementation and verification is discussed in Section 8.2.1.

Algorithm 1 Extended Kalman Filter

Initialize filter:
Set $\hat{\mathbf{x}}_0$, $\mathbf{P}_0$
while $t_k \leq t_{\text{end}}$ **do**
Time update (prediction):
1. $\hat{\mathbf{x}}_k^- = \mathbf{f}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}, \mathbf{0})$
2. $\mathbf{P}_k^- = \mathbf{A}_k \mathbf{P}_{k-1} \mathbf{A}_k^T + \mathbf{W}_k \mathbf{Q}_{k-1} \mathbf{W}_k^T$
Measurement update (correction):
3. $\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T \left(\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{V}_k \mathbf{R}_k \mathbf{V}_k^T \right)^{-1}$
4. $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \left(\mathbf{z}_k - \hat{\mathbf{z}}_k^- \right)$
5. $\mathbf{P}_k = (I - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^-$
$k = k + 1$
end while

Unscented Kalman Filter

Secondly, the Unscented Kalman Filter (UKF) will be discussed, which is strongly related to the previously described Extended Kalman Filter. Van der Merwe (2004) states that the EKF uses a first-order linearization of the equations of motion, which can lead to large uncertainties in the state estimation. The UKF uses a set of so-called sigma points, which are fed to the nonlinear equations of motion, and thereby is able to estimate the uncertainty of the model up to the third order (Van der Merwe, 2004). However, the structure of both filters is similar, as for the unscented filter also a time update and a measurement update are used. Next to this, one extra step is to be added for the UKF, as the sigma points have to be selected. This subsection will handle the equations and theory that is useful for the design of the UKF in the navigation system; the results are shown in Chapter 9.

For every time step the filter is initiated with the sigma points, which are selected in a smart way and according to Van der Merwe (2004), they have to be chosen such that they contain the important statistical characteristics of the state variables. Crassidis (2005) and Van der Merwe (2004) describe the selection of the sigma points as follows. Using the a-posteriori state $\hat{\mathbf{x}}_{k-1}$ and state covariance matrix $\mathbf{P}_{x,k-1}$ from time step $k-1$ and the process noise matrix $\mathbf{Q}_k$ a set of $2L+1$ points is generated, where L is the number of state variables.

$$\mathscr{X}_{0,k-1} = \hat{\mathbf{x}}_{k-1} \tag{5.50a}$$

$$\mathscr{X}_{i,k-1} = \hat{\mathbf{x}}_{k-1} + \left(\sqrt{(L+\lambda)\left(\mathbf{P}_{x,k-1} + \mathbf{Q}_k\right)} \right)_i \text{ for } i = 1, ..., L \tag{5.50b}$$

$$\mathscr{X}_{i,k-1} = \hat{\mathbf{x}}_{k-1} - \left(\sqrt{(L+\lambda)\left(\mathbf{P}_{x,k-1} + \mathbf{Q}_k\right)} \right)_i \text{ for } i = L+1, ..., 2L \tag{5.50c}$$

where $\lambda = \alpha^2 (L + \kappa) - L$. Then, every sigma point is assigned to a specific weight, which is computed via:

$$w_0^{(m)} = \frac{\lambda}{L+\lambda} \tag{5.51a}$$

$$w_0^{(c)} = \frac{\lambda}{L+\lambda} + (1 - \alpha^2 + \beta) \tag{5.51b}$$

$$w_i^{(m)} = w_i^{(c)} = \frac{1}{2(L+\lambda)} \text{for } i = 1, ..., 2L \tag{5.51c}$$

In these equations the scaling parameter α occurs, which can be set to $0 < \alpha \leq 1$. This parameter defines the spread of the sigma points, where a small value resembles a cluster of points that lie close together, whereas a large value resembles a wider spread of points. Then, also the parameter β can be seen, which is added to serve as an extra degree of freedom (Van der Merwe, 2004), and usually has a value of 2 for Gaussian distributions. Finally, it is defined that $\kappa = 3 - L$, which also serves as a scaling parameter. The square root that is to be computed in Equation (5.50) can be calculated using Cholesky decomposition (Crassidis, 2005).

Using the previous equations and the a priori covariance and process noise matrix, a different set of sigma points is computed for every time step. When this is done, the time update is performed, which is similar to

the EKF principle. However, as described earlier in this subsection, the nonlinear equations of motion are used, and thus the time update can be described as

$$\mathscr{X}_{k|k-1} = f\left(\mathscr{X}_{k-1}, \mathbf{u}_{k-1}, \mathbf{v}_{k-1}\right) \tag{5.52}$$

which is known as the Unscented Transform (UT). The above stated equation leads to a set of $[L \times 2L+1]$ transformed sigma points. Then, according to Julier and Uhlmann (1997), the mean and covariance of the transformed sigma points can be computed using

$$\hat{\mathbf{x}}_k^- = \sum_{i=1}^{2L+1} w_i^{(m)} \mathscr{X}_{i,k|k-1} \tag{5.53}$$

$$\mathbf{P}_{x,k}^- = \sum_{i=1}^{2L+1} w_i^{(c)} \left(\mathscr{X}_{i,k|k-1} - \hat{\mathbf{x}}_k^-\right)\left(\mathscr{X}_{i,k|k-1} - \hat{\mathbf{x}}_k^-\right)^T \tag{5.54}$$

which completes the time update and results in the a-priori estimated state vector $\hat{\mathbf{x}}_k^-$ and its covariance matrix $\mathbf{P}_{x,k}^-$. The third step in the UKF is similar to the second step in the EKF and is again denoted as the measurement update. For this step the actual measurement vector $\mathbf{z}_k$ and the transformed sigma points $\mathscr{X}_{k|k-1}$ are required. First, the Unscented Transform is applied with respect to the measurement equations, resulting in

$$\mathscr{Z}_{k|k-1} = h\left(\mathscr{X}_{k|k-1}, \mathbf{w}_{k-1}\right) \tag{5.55}$$

which now results in a set of $[M \times 2L+1]$ measurement points, where M is the number of measurements. From this set the mean and covariance can be computed again, using:

$$\hat{\mathbf{z}}_k^- = \sum_{i=1}^{2L+1} w_i^{(m)} \mathscr{Z}_{i,k|k-1} \tag{5.56}$$

$$\mathbf{P}_{\overline{z},k} = \sum_{i=1}^{2L+1} w_i^{(c)} \left(\mathscr{Z}_{i,k|k-1} - \hat{\mathbf{z}}_k^-\right)\left(\mathscr{Z}_{i,k|k-1} - \hat{\mathbf{z}}_k^-\right)^T + \mathbf{R}_k \tag{5.57}$$

Next to this, the cross-covariance is computed by

$$\mathbf{P}_{xz,k} = \sum_{i=1}^{2L+1} w_i^{(c)} \left(\mathscr{X}_{i,k|k-1} - \hat{\mathbf{x}}_k^-\right)\left(\mathscr{Z}_{i,k|k-1} - \hat{\mathbf{z}}_k^-\right)^T \tag{5.58}$$

which is used in the computation of the Kalman gain:

$$\mathbf{K}_k = \mathbf{P}_{xz,k}\mathbf{P}_{\overline{z},k}^{-1} \tag{5.59}$$

The measurement update is completed by

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k\left(\mathbf{z}_k - \hat{\mathbf{z}}_k^-\right) \tag{5.60}$$

and

$$\mathbf{P}_{x,k} = \mathbf{P}_{x,k}^- - \mathbf{K}_k\mathbf{P}_{\overline{z},k}\mathbf{K}_k^T \tag{5.61}$$

As for the EKF, the steps that are to be taken in the UKF are summarized in Algorithm 2. The implementation of the UKF together with the results are described in Section 8.2.2.

Particle Filter

Finally, the theory of the third filter that is used in the research that is described in this document is evaluated, namely the Particle Filter (PF). Previously the EKF and UKF were discussed, which are both Kalman filters and are therefore strongly related. The PF, however, is a Bayesian filter, and uses a stochastic set of points (particles) to estimate the state of the vehicle based on the nonlinear equations of motion, which thus is similar to the UKF (Gustafsson, 2009). The main algorithm consists of an initialization and a set of computations that are repeated at each time step. During the initialization a particle cloud is generated and a weight is assigned to each individual particle, after which the loop is initiated. First a step that is similar to the time update of the Kalman filters that were described earlier is found, which is denoted as the evolution of the particles (Ren,

Algorithm 2 Unscented Kalman Filter

Initialize filter
Set $\hat{\mathbf{x}}_0$, $\mathbf{P}_0$
while $t_k \leq t_{\text{end}}$ **do**
Calculate sigma points and weights:
1. $\mathscr{X}_{k-1} = \left[\hat{\mathbf{x}}_{k-1}, \hat{\mathbf{x}}_{k-1} + \sqrt{(L+\lambda)\left(\mathbf{P}_{x,k-1} + \mathbf{Q}_k\right)}, \hat{\mathbf{x}}_{k-1} - \sqrt{(L+\lambda)\left(\mathbf{P}_{x,k-1} + \mathbf{Q}_k\right)}\right]$
2. $\mathbf{w}^{(m)} = \left(\frac{\lambda}{L+\lambda}, \frac{1}{2(L+\lambda)}\right)$
3. $\mathbf{w}^{(c)} = \left(\frac{\lambda}{L+\lambda} + 1 - \alpha^2 + \beta, \frac{1}{2(L+\lambda)}\right)$
Time update (prediction):
4. $\mathscr{X}_{k|k-1} = f\left(\mathscr{X}_{k-1}, \mathbf{u}_{k-1}, \mathbf{v}_{k-1}\right)$
5. $\hat{\mathbf{x}}_k^- = \sum_{i=1}^{2L+1} w_i^{(m)} \mathscr{X}_{i,k|k-1}$
6. $\mathbf{P}_{x,k}^- = \sum_{i=1}^{2L+1} w_i^{(c)} \left(\mathscr{X}_{i,k|k-1} - \hat{\mathbf{x}}_k^-\right)\left(\mathscr{X}_{i,k|k-1} - \hat{\mathbf{x}}_k^-\right)^T$
Measurement update (correction):
7. $\mathscr{Z}_{k|k-1} = h\left(\mathscr{X}_{k|k-1}, \mathbf{w}_k\right)$
8. $\hat{\mathbf{z}}_k^- = \sum_{i=1}^{2L+1} w_i^{(m)} \mathscr{Z}_{i,k|k-1}$
9. $\mathbf{P}_{\bar{z},k} = \sum_{i=1}^{2L+1} w_i^{(c)} \left(\mathscr{Z}_{i,k|k-1} - \hat{\mathbf{z}}_k^-\right)\left(\mathscr{Z}_{i,k|k-1} - \hat{\mathbf{z}}_k^-\right)^T + \mathbf{R}_k$
10. $\mathbf{P}_{xz,k} = \sum_{i=1}^{2L+1} w_i^{(c)} \left(\mathscr{X}_{i,k|k-1} - \hat{\mathbf{x}}_k^-\right)\left(\mathscr{Z}_{i,k|k-1} - \hat{\mathbf{z}}_k^-\right)^T$
11. $\mathbf{K}_k = \mathbf{P}_{xz,k}\mathbf{P}_{\bar{z},k}^{-1}$
12. $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k\left(\mathbf{z}_k - \hat{\mathbf{z}}_k^-\right)$
13. $\mathbf{P}_{x,k} = \mathbf{P}_{x,k}^- - \mathbf{K}_k\mathbf{P}_{\bar{z},k}\mathbf{K}_k^T$
$k = k+1$
end while

2010) or the prediction of the particle state (Bistrovs, 2011). After this step, a measurement update is followed with, which is similar to the step that is taken in the Kalman filters. Then, resampling of the particles is done, which filters out outliers and duplicates particles with a high probability of being close to the actual state. Finally, the propagated particles from the time update step are multiplied by their respective weights from the measurement update. The summation of this product then leads to the estimated state. Several PF algorithms are described in Bistrovs and Kluga (2011), Ren and Ke (2010), Zhao (2015) and Gross et al. (2010), which are all based on the above described principle.

The initialization is started by the generation of a set of N particles, which are distributed over a specified range. Each particle represents a possible state, and, as the correctness of this state is unknown, equal weights are assigned to all particles, hence

$$w_{i,0} = \frac{1}{N} \tag{5.62}$$

which leads to a combined weight of one. The distribution at the initialization point can be denoted as (Ren and Ke, 2010):

$$p\left(\hat{\mathbf{x}}_{i,0}\right) \equiv p\left(\hat{\mathbf{x}}_{i,0}|\mathbf{z}_0\right) \tag{5.63}$$

Then the PF repeating computations can be started, which were described earlier in this subsection. First, all particles are propagated using the nonlinear equations of motion

$$p\left(\hat{\mathbf{x}}_{i,k}^-|\hat{\mathbf{x}}_{i,k-1}\right) = f\left(\hat{\mathbf{x}}_{i,k-1}, \mathbf{Q}_k\right) \tag{5.64}$$

where $\mathbf{x}_{i,k-1}$ are the a-posteriori particles from time step $k-1$ and $\mathbf{Q}_k$ is the process noise matrix at current time step k. The left-hand side of this equation is defined as the conditional prior of the state vector. Subsequently, the a-priori particle states at time step k are used to predict the measurements, which is similar to the UKF approach. Then, using the real measurements it can be defined that

$$p\left(\mathbf{z}_k|\hat{\mathbf{x}}_{i,k}^-\right) = \mathbf{z}_k - \hat{\mathbf{z}}_{i,k}^- \tag{5.65}$$

where $\mathbf{z}_k$ are the real measurements and $\hat{\mathbf{z}}_{i,k}^-$ are the estimated measurements at time step k for every particle i. This step seems similar to the one that was used in the Kalman filter approach, however, the difference between measured and estimated value is used differently. For every particle, which defines a possible state, this computed difference defines the likelihood of that particle being close to the actual value of the state. If for example a particle is taken that generates a small difference in pseudorange, it has a high likelihood of being close to the actual state, whereas a particle that is related to a large pseudorange difference has a low likelihood and is most likely far from the actual state. The measurement update for the PF uses the likelihood of every particle to compute the weight of every propagated particle (Bistrovs and Kluga, 2011, and Zhao, 2015):

$$\hat{w}_{i,k}^- = w_{i,k-1} \frac{p\left(\mathbf{z}_k|\hat{\mathbf{x}}_{i,k}^-\right) p\left(\hat{\mathbf{x}}_{i,k}^-|\hat{\mathbf{x}}_{i,k-1}\right)}{q\left(\hat{\mathbf{x}}_{i,k}^-|\hat{\mathbf{x}}_{i,k-1}, \mathbf{z}_{1:k}\right)} \tag{5.66}$$

Then, to ensure that the sum of all weights is equal to one, the above computed weights have to be normalized:

$$w_{i,k}^- = \frac{\hat{w}_{i,k}^-}{\sum_{i=1}^N \hat{w}_{i,k}^-} \tag{5.67}$$

The denominator of Equation (5.66) is the so-called importance density function and according to Bistrovs and Kluga (2011) this is the selection of this function is the most important step in the PF algorithm. However, it is also described that a very popular choice is to set the importance density function equal to the conditional prior of the state vector, thereby reducing Equation (5.66) to

$$\hat{w}_{i,k}^- = w_{i,k-1} p\left(\mathbf{z}_k|\hat{\mathbf{x}}_{i,k}^-\right) \tag{5.68}$$

which means that the propagation of the particle weights only depends on the likelihood of the estimated measurements.

At this point the propagated particles and their propagated weights are computed, which were dependent on the likelihood of the particles. It can be imagined that when a cloud of uniform distributed particles is chosen, most of the particles are located quite far from the true value of that state. This leads to the weights of those particles being very small with respect to the ones that are located close to the truth. After some iterations, it will most likely occur that only a really small percentage of all particles is usable for state estimation, the rest will then be assigned to negligible weights, which is called filter degeneracy. Gross et al. (2010) clearly describe that this can be omitted by resampling; some particles with very low weight are replaced by duplicates of the particles with the highest weights. For doing this, the effective number of particles is computed to be:

$$N_{\text{eff}} = \frac{1}{\sum_{i=1}^N \left(w_{i,k}^-\right)^2} \tag{5.69}$$

Then, the filter evaluates whether $N_{\text{eff}} < N_{\text{threshold}}$, and in case this is true, the resampling process is started. During this process the cumulative sum of the weights is calculated, which can be defined as the Cumulative Distribution Function (CDF) (Fay and Speyer, 2008)

$$c_i = c_{i-1} + w_{i,k}^- \text{ for } i = 2, ..., N \tag{5.70}$$

with $c_1 = 0$. Then a random starting point u_1 is generated using a uniform distribution with zero mean and a standard deviation of $1/N$. Subsequently a loop through all points is initiated where the value of u is increased by linear parts that are proportional to $1/N$:

$$u_j = u_1 + \frac{j-1}{N} \text{ for } j = 1, ..., N \tag{5.71}$$

Then, when the linear increment between point i and $i-1$ is larger than the difference between c_i and c_{i-1}, point i is replaced by a duplicate of point $i-1$. This is repeated for all particles, up until a new, updated distribution is found, containing duplicates of particles with a high likelihood. Finally, at the end of the resampling process, the weights are reset to

$$w_{i,k} = \frac{1}{N} \tag{5.72}$$

When no resampling is done:

Algorithm 3 Particle Filter (for $i = 1 : N$)

Initialize filter:
Set $p\left(\hat{\mathbf{x}}_{i,0}\right) \equiv p\left(\hat{\mathbf{x}}_{i,0}|\mathbf{z}_0\right), w_{i,0}$
while $t_k \leq t_{\text{end}}$ **do**
Time update (prediction):
1. $p\left(\hat{\mathbf{x}}_{i,k}^{-}|\hat{\mathbf{x}}_{i,k-1}\right) = f\left(\hat{\mathbf{x}}_{i,k-1}, \mathbf{Q}_k\right)$
Measurement update (correction):
2. $p\left(\mathbf{z}_k|\hat{\mathbf{x}}_{i,k}^{-}\right) = \mathbf{z}_k - \hat{\mathbf{z}}_{i,k}^{-}$
3. $\hat{w}_{i,k}^{-} = w_{i,k-1}\, p\left(\mathbf{z}_k|\hat{\mathbf{x}}_{i,k}^{-}\right)$
4. $w_{i,k}^{-} = \frac{\hat{w}_{i,k}^{-}}{\sum_{i=1}^{N} \hat{w}_{i,k}^{-}}$
5. $N_{\text{eff}} = \frac{1}{\sum_{i=1}^{N}\left(w_{i,k}^{-}\right)^2}$
if $N_{\text{eff}} < N_{\text{threshold}}$ **then**
Resample:
6a. $w_{i,k} = \frac{1}{N}$
else
6b. $w_{i,k} = w_{i,k}^{-}$
end if
7. $\hat{\mathbf{x}}_k = \sum_{i=1}^{N} w_{i,k}\, p\left(\hat{\mathbf{x}}_{i,k}^{-}|\hat{\mathbf{x}}_{i,k-1}\right)$
$k = k + 1$
end while

$$w_{i,k} = w_{i,k}^{-} \tag{5.73}$$

Resampling however has a negative influence on the computational load of the filter and therefore it may not be convenient to resample all particles at every time step. Next to that, resampling decreases the diversity of particles, as duplicates are taken instead of unique points. By varying the threshold, it is chosen to only resample the particles when the effective number of particles is lower than a certain value, which reduces the computational load and diversity problem and still prevents the filter to become degenerate.

The final step of the Particle Filter is the estimation of the state, which is done by a summation of the multiplication of every particle with its assigned weight:

$$\hat{\mathbf{x}}_k = \sum_{i=1}^{N} w_{i,k}\, p\left(\hat{\mathbf{x}}_{i,k}^{-}|\hat{\mathbf{x}}_{i,k-1}\right) \tag{5.74}$$

The next iteration is then initiated by particles $p\left(\hat{\mathbf{x}}_{i,k}^{-}|\hat{\mathbf{x}}_{i,k-1}\right)$ and the weights $w_{i,k}$. The governing equations of the PF are summarized in Algorithm 3, which eases the implementation process of the filter. In this algorithm the computations are done for every particle i. Note that in this subsection, only the theory of the PF is described, the implementation, verification and filter tests are discussed in Section 8.2.3. In that section, also the resampling process and its implementation is shown in more detail in Algorithm 4.

5.2 Control

Previously, the navigation and guidance segments of the GNC-system were discussed, leaving the last part to be the control segment. As for the navigation system, the control system can be divided into two parts, namely hardware and software. The hardware of a control system are the actuators, which are the final step in the GNC-loop; they actively adjust the direction or velocity of the vehicle. From the guidance and navigation segment it is now known where the vehicle is and where the vehicle has to go; the software part produces commands for the actuators, due to which the vehicle will actually go to the desired location (by changing its attitude and/or velocity).

Earlier it was discussed that the focus of this thesis is on the navigation segment, and thus it is chosen to use a relatively simple control system. The decision on which actuators were to be used was derived from the selection of the reference vehicle, which was chosen to be the HORUS 2B. Along with this vehicle the

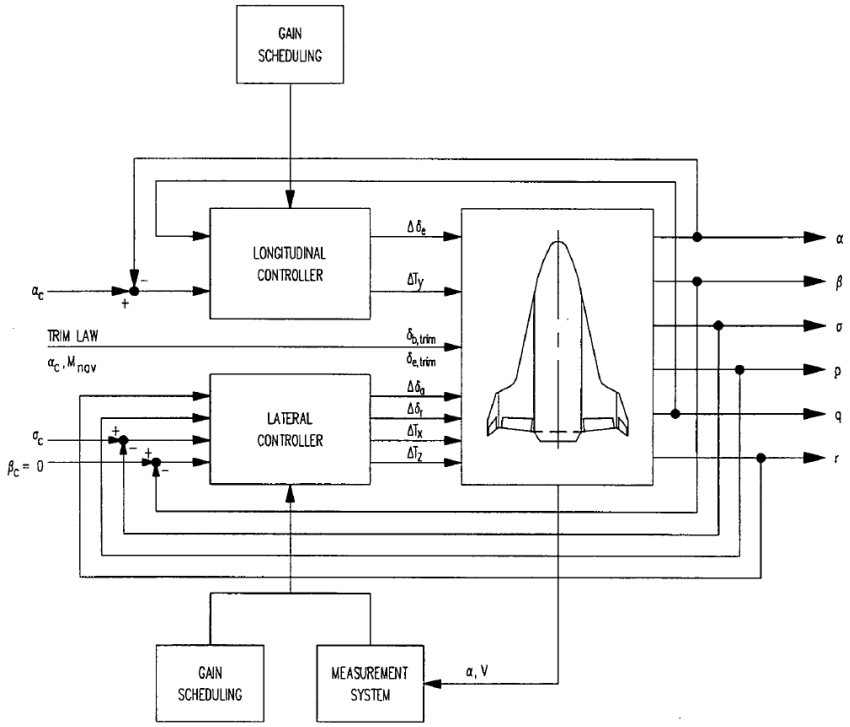


Figure 5.7: Architecture of the HORUS-2B controller (Mooij, 1997). All signals are denoted as arrows, subtitled by their respective variables.

characteristics of the actuators that are used in the control system were found. As for the software it was chosen to use a Linear Quadratic Regulator (LQR), which is based on optimal control theory and was chosen over a PID controller due to the complexity of the re-entry vehicle system dynamics. In this section, first an elaboration on the theory behind the implementation of an LQR is done, after which the layout of the HORUS actuators will be shown.

5.2.1 Linear Quadratic Regulator

As was described in the introduction of this section, it is focused on the navigation segment of the GNC module, and therefore a relatively simple control system is designed. In Mooij (1997) a full design cycle of such a system is described, on which the controller will be based. MBB (1988) then also describes a controller for the reference vehicle that will be used in this research, of which the schematics are shown in Figure 5.7.

In Figure 5.7 it can be seen that the vehicle will be controlled using the output of the guidance system, which in the simulation will be a preset control history containing an angle of attack and a bank angle profile (the angle of sideslip is commanded to be zero at all times). Then, the difference between these commands and the actual values of these variables is computed, which serves as an input to the controller. It is shown that this controller is divided into two parts as it was decided to decouple symmetric and asymmetric motion. This can be done by reasoning that perturbations in longitudinal motion will not induce perturbations in lateral motion and vice versa. Decoupling the system will simplify the design of the controller.

In the two controller blocks the previously described input of the three aerodynamic angles combined with a set of predetermined gains will be used. From these inputs the controllers are able to compute deflections for the aerodynamic control surfaces and moments for the Reaction Control System (RCS), which consists of roll, pitch and yaw thrusters. This output is then used to compute control forces and moments, which are subsequently fed to the dynamics of the re-entry system, resulting in a new state. This new state then functions as an input for the next time step.

According to Srinivasan (2006) the concept of a LQR is to maximize or minimize a predefined cost function, that mostly is defined as a quadratic function of the state and control variables and can be written as:

$$\mathbf{J} = \int_0^\infty \left(\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u}\right) dt \tag{5.75}$$

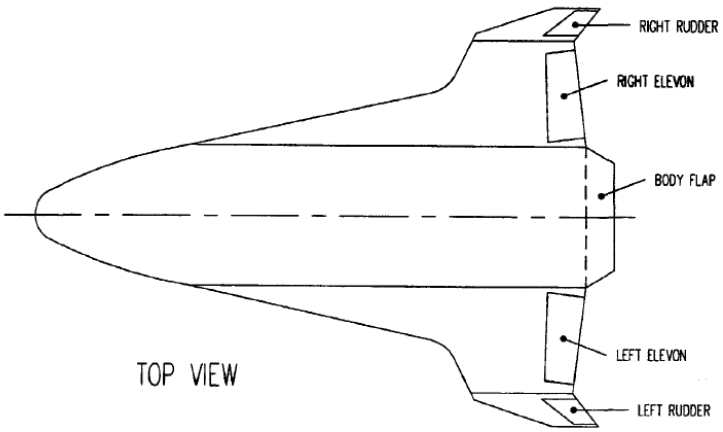


Figure 5.8: Geometry of the aerodynamic control surfaces of the HORUS 2B reference vehicle (MBB, 1988). Next to these aerodynamic surfaces, HORUS also uses a reaction control system (RCS), consisting of roll, pitch and yaw thrusters.

In this function, **x** is the state and **u** are the controls. **Q** and **R** are weighting matrices, with the characteristics that **Q** is real and positive semi-definite, whereas **R** is a real symmetric positive definite matrix. These matrices can be varied to increase or decrease the values of the controls (which is called the control effort) or the speed of the controller's response (which is denoted as control deviation). The implementation of these matrices will be shown in Chapter 7.

The aforementioned cost function was said to be used to determine certain state-feedback gains, which are then used to compute the values of the controls at specific points in time. For this, a reference to the definition of a linearized state space system is made

$$\Delta\dot{\mathbf{x}} = \mathbf{A}\Delta\mathbf{x} + \mathbf{B}\Delta\mathbf{u} \tag{5.76}$$

$$\Delta\mathbf{y} = \mathbf{C}\Delta\mathbf{x} + \mathbf{D}\Delta\mathbf{u} \tag{5.77}$$

where the control law was defined as (Mooij, 1997):

$$\Delta\mathbf{u} = -\mathbf{K}\Delta\mathbf{x} \tag{5.78}$$

The state-feedback gain matrix **K** is defined as

$$\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^{T}\mathbf{P} \tag{5.79}$$

where P is a positive definite matrix and is obtained from the Riccati equation (Gopal, 1993 and Mooij, 1997)

$$\mathbf{A}^{T}\mathbf{P} + \mathbf{P}\mathbf{A} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}\mathbf{P} + \mathbf{Q} = \mathbf{0} \tag{5.80}$$

For finding the solution of the matrix Riccati equation, special MATLAB commands can be used. At this point, the theory of the Linear Quadratic Regulator was discussed, which will be used in Section 7.1 to sustain the implementation. In the following section then the controls that are used in the implementation will be shown, which are based on the HORUS-2B reference vehicle.

5.2.2 HORUS Controls

Previously, the theory of the LQR controller was described, which will be used to design a controller, which subsequently will be implemented in the simulator of the re-entry flight. Earlier in this report it was also stated that the HORUS 2B is used as a reference vehicle, as its characteristics are well-known through the MBB (1988) and Mooij (1995) reports. In this section all properties of the HORUS 2B that are related to the control system that is to be implemented are discussed.

Figure 5.8 shows that five different aerodynamic surfaces are used, together with three types of thrusters that conclude the Reaction Control System (RCS) (MBB, 1988). For roll movement, the vehicle uses an asymmetric deflection of the left and right elevon, which is denoted as aileron deflection, together with the roll thrusters. The aileron deflection is defined as:

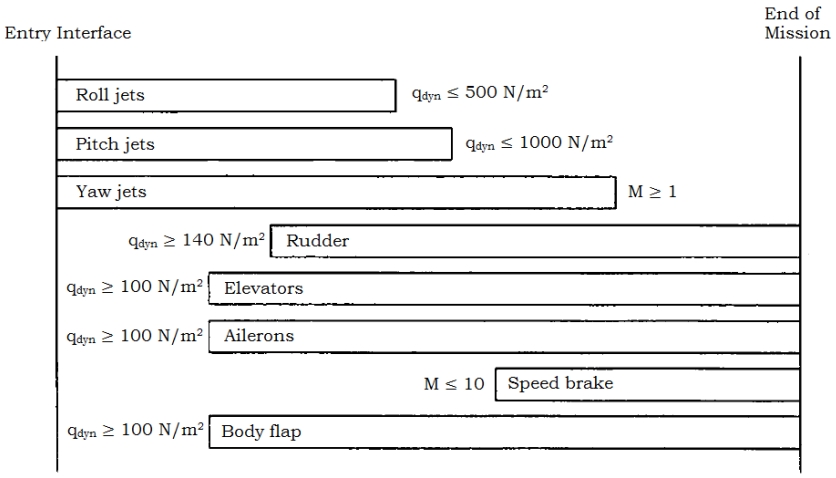


Figure 5.9: Control scheme of the HORUS-2B re-entry mission (based on Cooke, 1982).

$$\delta_a = \frac{\delta_{e,l} - \delta_{e,r}}{2} \tag{5.81}$$

From the control scheme that was derived from the Space-Shuttle scheme, as described by Mooij (1997), it is found that for roll movement the roll jets are used for the first part of the entry flight up until $\bar{q} \geq 100$ N/m^2, where hybrid roll control is initiated. Then, for $\bar{q} > 500$ N/m^2 the roll jets are deactivated, resulting in only aerodynamic control for the last part of the trajectory. For yaw motion, a similar set up is defined, however, the RCS thrusters for this motion are related to the Mach number. The yaw jets are activated up until a Mach number of 1, which is the whole flight in this case, as the TAEM phase is not evaluated. The rudders are activated for $\bar{q} \geq 140$ N/m^2, which differs from the Space-Shuttle scheme, where they were activated at $M = 3.5$. However, as the rudder influence is small for low dynamic pressure, this should not pose any significant differences with respect to the Space-Shuttle control scheme. The rudders can only be deflected outwards, where the deflection of the left rudder is defined as positive.

Finally, longitudinal control is considered, which is decoupled from yaw and roll control. For pitch motion, symmetric deflection of the elevons is used, which is denoted as elevator deflection and defined as:

$$\delta_e = \frac{\delta_{e,l} + \delta_{e,r}}{2} \tag{5.82}$$

Next to that, the body flap can also be used for pitch movement, just as the pitch thruster of the RCS. The latter one again acts from the start of the re-entry flight and is enabled for $\bar{q} \leq 1000$ N/m^2. The elevons are activated for dynamic pressure values that are higher than 100 N/m^2. The body flap then has the same activation constraint as the elevons, but is only used to trim the vehicle and not in corrective control.

Finally, after all the controls that are used for corrective control are described, it rests to mention the trimming process of the vehicle. As was stated, the vehicle uses a predetermined control history as a guidance system, which defines commands for angle of attack and bank angle over the whole re-entry flight. The process of trimming the vehicle is necessary to ensure that the vehicle will not change its orientation when it is in steady flight. Then the function of the aforementioned body flap can be evaluated. The commands from the guidance system are used to compute body flap deflection, which ensures that the vehicle is trimmed. When the maximum deflection of the body flap is reached, a symmetric deflection of the elevons is used to maintain a trimmed vehicle.

In Chapter 6 the simulator will be discussed, along with the results of the nominal trajectory (the flight when the guidance commands are executed perfectly, so the aerodynamical angles are set). Along with these results a trim command profile for both the body flap and the elevons will be shown. This concludes the theory on the control of HORUS 2B. Chapter 7 will contain the implementation of both the longitudinal and lateral controller, along with the computation of the feedback gain matrices and the verification of the controller in terms of step and ramp responses.

6

Simulator Design

Previously, the mission heritage and mission and system requirements were described in Chapter 2. The system requirements were related to the simulator, which will be fully evaluated in this chapter. Then, in Chapter 3, the theory of the flight dynamics of the vehicle was discussed, where the translational and rotational equations of motion were defined. Furthermore, the gravitational field and the atmospheric model that will be used in the simulator were handled in Chapter 4. Subsequently, in Chapter 5, the theory of the navigation sensors and estimators was discussed and LQR control was described. In these chapters also a description was provided of the HORUS-2B reference vehicle, where the geometry, control actuator layout and the aerodynamic characteristics were discussed.

In this chapter, the choices that were made during the design of the re-entry trajectory simulator and their impact on the design itself are listed in Section 6.1. These choices mostly concern the topics that were discussed earlier in this report. Then, the numerical methods that are used in the simulator will be described in Section 6.2. In Section 6.3 the implementation of the different blocks and their integration into the simulator is evaluated. In this section flow charts are provided to show the input and output signals and their trajectories within different blocks. When this is concluded the verification of the simulator is evaluated, for which a distinction is made between module verification and system verification. For doing this the open loop simulation and nominal trajectory simulation are used, which will be described in Section 6.4.

6.1 Simulator Design Choices

As was mentioned earlier, for most (if not all) elements of the simulator there exist multiple designs, each having their own combination of assumptions and design choices. Before the design of the simulator is started these have to be considered, which will be done in this section.

Fist, a decision was made on the software tool that was to be used for programming the simulator, and, for this decision, four options were evaluated: MATLAB/Simulink, TUDAT Aerospace Toolbox, ATA Aerospace Toolkit, and ASTOS. Next to that, the programming languages of Python, MATLAB, C++/C, and Fortran were evaluated and both from the language and the tool options, one was selected for the implementation of the simulator. Houcque (2005) stated that the MATLAB language contained one main advantage over the other conventional computer languages for solving technical problems, which is the fact that the basic data element is an array, due to which no dimensioning is required. Next to that, the MATLAB language was coupled to the MATLAB/Simulink environment, and, from personal communication with Mooij (2017) it was decided to use the verified Simulink Aerospace Blockset, to evaluate the ease of implementation of this tool. However, after the design and implementation of the open-loop simulator, which represents trimmed flight, it was seen that, despite the fact that the implementation and verification is done much faster than for when the different parts of the simulator are programmed manually, the computational load was high in comparison with the simulator designed by Mooij (2017). After system verification on the Aerospace-Blockset simulator was performed, it was chosen to use the simulator from Mooij for the integration of the GNC-system, as this would decrease the time required for performing simulations. This simulator was also programmed in the MATLAB/Simulink environment but contained S-functions, which used the C language. The implementation and integration of the GNC system however was done in the MATLAB language again, which was proved to be a suitable language for this type of problems, as was seen in similar research (Xue and Lu, 2009 and Costa et al., 2002).

The simulator that is described in this chapter was the one that was used for integration of the GNC-system, hence, the one that was provided by Mooij (2017). It is divided into different blocks, each containing their own aspect of re-entry flight. In the top layer of the simulator, six blocks can be distinguished. In the flight dynamics block, the integration of the vehicle state is done, which has the gravitational, aerodynamic, and control forces and moments as input. The gravitational field and atmospheric model are described in the environment block, where some atmospheric properties are used in the computation of the aerodynamic forces and moments, which is done in the aerodynamics block. Then, a guidance, navigation, and control block can be defined, each describing the respective module of the integrated GNC-system. In the guidance block, the trim commands are generated, which partly depend on the guidance profile that in this case consists of angle-of-attack, angle-of-sideslip, and bank-angle commands. In the control system the corrective control commands are computed, which are transformed to control forces and moments. These are then fed to the flight dynamics block as input for the state propagation. With the above described blocks, a 6DoF closed-loop simulation can be performed, however, this would be done with the true state as input for the control system, which is not the case in real life. Therefore a navigation block is added containing a sensor system for generating IMU, GPS, and FADS measurements and a filter for estimating the vehicle's state, which is then fed as input to the guidance and control blocks (as the trim commands are computed with estimated Mach number and dynamic pressure as well).

6.1.1 Environment

As was described earlier, the environment module is used to compute the gravitational forces and the atmospheric properties, using the models that were described in Chapter 4.

- *Atmospheric Model* - In Section 4.2 it was chosen that the US76 Standard Atmosphere was to be used, which is defined for the whole range of altitude the re-entry vehicle will be in.
- *Gravitational Field* - For the gravity that is pulling the vehicle towards the Earth the J_2 term is included.

The vehicle's position and velocity is fed as input to the atmospheric model, which computes the free stream air pressure, density, temperature, speed of sound, Mach number, and dynamic pressure. Then, for the gravity model, the position of the vehicle is used, which is sufficient to compute the gravitational forces.

6.1.2 Aerodynamics

The second block that is evaluated is the aerodynamics block, which provides the other input for the flight dynamics block for uncontrolled flight. Previously, it was described that all characteristics of the reference vehicle were taken from Mooij (1995), and so the aerodynamic properties of the HORUS 2B vehicle can also be found here. In this document, multiple tables can be found containing the aerodynamic coefficients for drag, side force, lift and roll, pitch and yaw moments as function of Mach number, angle of attack and the deflection of several control surfaces. These tables were imported into the Simulink model.

6.1.3 Flight Dynamics

Then, the flight dynamics block can be handled, as the inputs of this block were discussed already. The following design choices are made:

- *Vehicle Characteristics* - The HORUS 2B vehicle is used as a reference. Its geometry elements as well as its mass and inertial properties are taken from Mooij (1995).
- *Flight Dynamics* - For the simulation, a 6DOF model is used, as the translational motion as well as the rotational motion is to be evaluated. The mass of the vehicle is constant during flight and no propulsion forces are present. Finally, it is assumed that there is no wind.

The following nonlinear system is considered

$$\dot{\mathbf{x}} = \mathbf{A}(t, \mathbf{x}, \mathbf{u})\,\mathbf{x} + \mathbf{B}(t, \mathbf{x}, \mathbf{u})\,\mathbf{u} \tag{6.1}$$

where the state vector $\mathbf{x}$ is described by:

$$\mathbf{x} = \left(\dot{x}, \dot{y}, \dot{z}, x, y, z, p, q, r, Q_1, Q_2, Q_3, Q_4\right)^t \tag{6.2}$$

The first six elements describe the translational motion of the vehicle, whereas the latter seven elements describe the rotational motion. For this type of motion it was decided to use quaternions to rule out any possibility of singularities. The control vector $\mathbf{u}$ is defined as

$$\mathbf{u} = \left(\delta_e, \delta_a, \delta_r, M_{T,x}, M_{T,y}, M_{T,z}\right)^T \tag{6.3}$$

where the former three elements describe elevator, aileron, and rudder deflection respectively. The latter three elements are the RCS moments around the x, y, and z axes. This control vector is used to compute the control forces and moments that are fed as input to the flight dynamics block in case of a closed-loop simulation. With the blocks that are evaluated at this point, a free-fall trajectory can be simulated.

6.1.4 Guidance

The guidance block consists of two parts, namely the guidance profile and the trim module. In the former, an attitude profile is provided, which defines commanded values of the angle of attack, angle of sideslip, and bank angle during the whole flight. The rotational rates are commanded to be zero. Note that in real life, an actual guidance system will be available, however, this will be discussed in Chapters 7 and 9. The latter part of the guidance block is the trim module, in which the trim commands are computed. As for the design choices concerning this module, the vehicle is trimmed using the body flap and a symmetric deflection of the elevons if the body flap cannot trim the vehicle by itself. With this block included in the simulator, an open-loop simulation can be performed.

6.1.5 Control

In the control block the guidance commands are used to compute corrective control commands, which are stored in **u** that was described earlier in this section. It is decided that perfect actuators are used, so any effects due to hydraulics or deflection errors are neglected; the actuators have infinite deflection velocity and reach the exact commanded deflection at all times. This simplification can be done as in Mooij (1997) it was stated that the actuator activation velocity is much faster than the eigenmotion of the vehicle. Note that this also holds for the actuators that are used for trimmed flight. Including this block in the simulator will lead to the possibility of a 6DoF closed-loop simulation.

6.1.6 Navigation

Then, finally, in the navigation module three sensors are implemented, which were described in Section 5.1. The IMU is divided into an accelerometer, which measures the accelerations in the body frame, and a gyroscope, which measures the rotational rates of the body frame with respect to the inertial frame. The GPS sensor finds the pseudorange, range rate and carrier phase differences for the available satellites and can select a predefined number of satellites that provide the best GDOP. The FADS sensor then measures pressure over a set of virtual orifices located on the nose of the vehicle. Then, in the navigation filter, a state estimate is computed, which, when this block is integrated in the simulator, will be fed to the guidance and control blocks. Hence, with this block integrated in the 6DoF closed-loop simulation, an error is introduced in the computation of the trim and corrective control commands, that leads to an uncertainty in the input control forces and moments of the flight dynamics module.

6.2 Numerical Methods

Before the simulator can be set up , first the numerical methods that are to be used have to be defined, so that they can be directly implemented in the simulator. In this chapter a discussion is shown on the chosen methods for numerical integration and interpolation in Sections 6.2.1 and 6.2.2 respectively, and subsequently, their application in the simulator is described.

6.2.1 Integration

As was discussed in Chapter 3, in the simulator the (differential) EoM of the vehicle are used. The simulation of the re-entry trajectory is produced by propagating these equations over time. First, the concept of an ordinary differential equation (ODE) (Klees and Dwight, 2013 and Press et al. 1989) is considered

$$\frac{d\mathbf{y}_i(\mathbf{x})}{d\mathbf{x}} = \mathbf{f}_i\left(\mathbf{x}, \mathbf{y}_1, ..., \mathbf{y}_n\right) \tag{6.4}$$

where i goes from 1 to N. Press et al. (1989) then adds that an ODE is not completely specified by the equations only, but that boundary conditions have to be stated in order to describe a solvable problem. These problems can be split into two different sets: initial value problems and two-point boundary problems (Klees and Dwight, 2013). In the former, a specific initial condition is given in the form:

$$\mathbf{y}(\mathbf{x}_0) = \mathbf{y}_0 \tag{6.5}$$

For the two-point boundary problem the boundary conditions are defined in the form:

$$\mathbf{g}\left(\mathbf{y}(\mathbf{x}_0), \mathbf{y}(\mathbf{x}_n)\right) = \mathbf{0} \tag{6.6}$$

When an ODE consists of a linear combination of derivatives of the state (which is defined as $\mathbf{y}$ in the previous equations), it is said to be linear and can be solved analytically (Press et al., 1989). However, in most real-life situations one has to deal with nonlinear ODEs, which have to be solved numerically. Theoretically, the exact solution can be approached by stating

$$\mathbf{y}(\mathbf{x} + \Delta\mathbf{x}) \approx \mathbf{y}(\mathbf{x}) + \dot{\mathbf{y}}\Delta\mathbf{x} \tag{6.7}$$

where decreasing the step size $\Delta\mathbf{x}$ results in a more accurate approximation. As can be seen, Equation (6.7) resembles Euler's method, which is the most easy and straightforward integration method. To determine whether a particular integration method is suited for a specific problem, the following aspects have to be considered:

- **Computational speed** - usually, when the complexity of an integration method increases, its required computational time also increases. One has to trade off the accuracy of the result with the computational speed to make a proper choice.
- **Error** - the error, or accuracy of the integration method can be defined for two points. The local truncation error defines the inaccuracy that is introduced in one single integration step. The final error is constructed by all errors in the individual integration steps.
- **Robustness** - this aspect describes how well the integration method reacts to small changes in initial conditions. When such changes induce large final errors, the method is said to be unstable.

Next to these aspects, it is convenient when the step size of the method is easily adjustable, where a distinction can be made between fixed step size methods and variable step size methods. The advantage of a variable step size method is that the step size is decreased when the error tends to get out of bounds.

Previously, Euler's integration method was stated, and according to Press et al. (1989), this method induces quite a large truncation error, which accumulates into a large final error. However, it is used as a basis for different integration methods, such as the family of Runge-Kutta methods.

Runge-Kutta Methods

The family of Runge-Kutta methods are related to Euler's integration method as the increment of the function value is based on the step size and the value of one or more derivatives over the integration step. The main advantage of this family of integrators is that they are relatively easy to implement and do not require excessive computation time. Next to that, the technique of varying the step size can be implemented, which is done in the Runge-Kutta-Fehlberg (RKF45) method, which has its truncation terms in the fifth order. According to Mathews and Fink (1999) this method computes two solutions at each integration step, which are defined by

$$k_1 = hf\left(t_n, y_n\right) \tag{6.8a}$$
$$k_2 = hf\left(t_n + a_2 h, y_n + b_{21}k_1\right) \tag{6.8b}$$
$$k_3 = hf\left(t_n + a_3 h, y_n + b_{31}k_1 + b_{32}k_2\right) \tag{6.8c}$$
$$k_4 = hf\left(t_n + a_4 h, y_n + b_{41}k_1 + b_{42}k_2 + b_{43}k_3\right) \tag{6.8d}$$
$$k_5 = hf\left(t_n + a_5 h, y_n + b_{51}k_1 + b_{52}k_2 + b_{53}k_3 + b_{54}k_4\right) \tag{6.8e}$$
$$k_6 = hf\left(t_n + a_6 h, y_n + b_{61}k_1 + b_{62}k_2 + b_{63}k_3 + b_{64}k_4 + b_{65}k_5\right) \tag{6.8f}$$

and

$$y_{n+1} = y_n + c_1 k_1 + c_2 k_2 + c_3 k_3 + c_4 k_4 + c_5 k_5 + c_6 k_6 + \mathcal{O}\left(h^6\right) \tag{6.9}$$

which is the fifth order solution and

Table 6.1: Integration parameters found by Dormand and Prince that are used in the adaptive step size RKF45 approach (Dormand and Prince, 1980)

i	a_i	b_{ij}						c_i	c_i^*
1								$\frac{35}{384}$	$\frac{5179}{57600}$
2	$\frac{1}{5}$	$\frac{1}{5}$						0	0
3	$\frac{3}{10}$	$\frac{3}{40}$	$\frac{9}{40}$					$\frac{500}{1113}$	$\frac{7571}{16695}$
4	$\frac{4}{5}$	$\frac{44}{45}$	$-\frac{56}{15}$	$\frac{32}{9}$				$\frac{125}{192}$	$\frac{393}{640}$
5	$\frac{8}{9}$	$\frac{19372}{6561}$	$-\frac{25360}{2187}$	$\frac{64448}{6561}$	$-\frac{212}{729}$			$-\frac{2187}{6784}$	$-\frac{92097}{339200}$
6	1	$\frac{9017}{3168}$	$-\frac{355}{33}$	$\frac{46732}{5247}$	$\frac{49}{176}$	$-\frac{5103}{18656}$		$\frac{11}{84}$	$\frac{187}{2100}$
7	1	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{84}$	0	$\frac{1}{40}$
$j=$		1	2	3	4	5	6		

$$y_{n+1}^* = y_n + c_1^* k_1 + c_2^* k_2 + c_3^* k_3 + c_4^* k_4 + c_5^* k_5 + \mathcal{O}\left(h^5\right) \tag{6.10}$$

which is the so-called embedded fourth order solution.

Mathews and Fink (1999) then describe that the difference between both solutions is used to estimate the local truncation error. When this error does not meet the predefined truncation error constraint, the step size h is decreased and when this error is accurate in more than the predefined significance, the step size is increased. The truncation error estimation is defined as:

$$\Delta_{n+1} = y_{n+1} - y_{n+1}^* = \sum_{i=1}^{6} \left(c_i - c_i^*\right) k_i \tag{6.11}$$

In Chapter 3 the translational and rotational equations of motion were described, and it was seen that these can be classified as differential equations. Propagation of the state of the vehicle then is to be done using numerical integration of the EoM, and thus an integrator is to be chosen. As a complex system is considered, a variable step size integrator would be most convenient, as this type of integrator is able to increase the step size for higher integration speed and to reduce the step size when a higher integration accuracy is required. From the integrated methods in Simulink, the ODE45 routine is chosen, due to better performance than the ODE23 routine on accuracy and speed, as ODE23 is based on a third order Runge-Kutta method, whereas ODE45 represents a fifth order Runge-Kutta method. The ODE113 does not necessarily perform better than ODE45, and as this is a multi-step method, meaning that the solution is based on multiple previous steps, it showed problems with initialization of the problem, as there is only one initial condition available. The integrated ODE45 routine corresponds to the Dormand Prince butcher table that is shown in Table 6.1.

Next to the integration of the EoM, also in the navigation module an integrator is required, to propagate the estimated state using IMU measurements. For doing this, a simpler model is required, as the speed of the integration has high priority. Next to that, it is convenient to use a fixed-stepsize integrator, as the IMU possesses a specific operating frequency, which is fixed as well. The most straightforward method that was discussed in this subsection is Euler integration, and, as this method is fast, it is considered in the implementation of the navigation module. However, as the accuracy of Euler integration is relatively poor, a different method of higher order is evaluated as well. After comparison, a decision can be made on which integrator will be used in the navigation module. For this comparison, it is chosen to use the RK4 method, as this is in almost all cases better than the second-order RK method, as stated by Press et al. (1989). The RK4 method, however, is still easy to implement and describes a fourth-order integrator, whereas Euler integration only is a first-order method. The equations that are used in the RK4 method are:

$$y_{n+1} = y_n + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6} + \mathcal{O}\left(h^5\right) \tag{6.12}$$

The corresponding k parameters are defined by

$$k_1 = f\left(x_n, y_n\right) h \tag{6.13a}$$

$$k_2 = f\left(x_n + \frac{\Delta x}{2}, y_n + \frac{k_1}{2}\right) h \tag{6.13b}$$

$$k_3 = f\left(x_n + \frac{\Delta x}{2}, y_n + \frac{k_2}{2}\right) h \tag{6.13c}$$

$$k_4 = f\left(x_n + \Delta x, y_n + k_3\right) \tag{6.13d}$$

Note that the RK4 and Euler integrators are used for a state prediction in the navigation module. As this prediction is only part of the filters that are used and the final state estimate is done by a measurement update using different sensors, the speed of the integrator may have higher priority than the accuracy. In Section 8.1.1 these two integrators are compared, and depending on their performance of the IMU measurement propagation over the whole entry trajectory, a decision is made on the technique that is used in the final simulator.

6.2.2 Interpolation

In Section 4.3, the aerodynamics of the vehicle during its re-entry were described. Implementation of several aerodynamic coefficients in the simulator is done using aerodynamic tables from the HORUS 2B references in MBB (1988) and Mooij (1995). As these tables only contain values for predefined angles of attack and Mach numbers, interpolation is to be performed between these values to obtain the numbers for all angles of attack and Mach numbers throughout the flight.

According to Press et al. (1989), interpolation should be applied to model the function between two known points by some plausible functional form. Bulirsch and Stoer (1993) mention four approaches for performing interpolation; using polynomials, rational functions, trigonometric functions and splines. It is stated that trigonometric interpolation is mostly used in numerical Fourier analysis, which is not related to the research described in this thesis. Rational interpolation is quite similar to polynomial interpolation with the main difference being that fractions are used, which lead to the possibility of asymptotes in the interpolating functions. Due to that inconvenience only two methods remain. In the previous paragraph it was stated that interpolation is required for the computation of the aerodynamic force and moment coefficients. These coefficients are nonlinear functions of angle of attack and Mach number. According to Bulirsch and Stoer (1993), spline interpolation has one main advantage with respect to polynomial interpolation, being that the interpolating curves are less likely to have the huge oscillations when high-degree polynomials are used.

It was described that interpolation is only used in the computation of the aerodynamic coefficients in the simulator. From the previous discussion two methods remained, which are polynomial interpolation and spline interpolation. The easiest applicable form of spline interpolation is setting the degree of the splines to one, which resembles linear interpolation between the data points. As this induces simple calculations, rather than a higher order polynomial to fit the aerodynamic coefficient curves, the computational load of this method is low. Next to that, the behavior of the coefficients between two data points is unknown, and thus one cannot draw conclusions with respect to the accuracy of the interpolated values. Due to this, the computational load has the highest priority, and therefore it is chosen to use linear spline interpolation. When running the simulator and computing the aerodynamic forces and moments using the calculated coefficients, no problems due to interpolation were encountered.

Mathews and Fink (1999) define a linear spline as:

$$s_k(x) = y_k + \frac{y_{k+1} - y_k}{x_{k+1} - x_k}\left(x - x_k\right) \tag{6.14}$$

where x denotes the input variable, which in the case of the aerodynamic coefficients is Mach number, angle of attack, angle of sideslip or altitude and y represents the tabular data. The subscript k and $k+1$ define the position in the table. Linear spline interpolation is an integrated technique in the Simulink look-up tables, and when increasing the number of dimensions, Simulink first interpolates over the first dimension, then over the second, up until all dimensions are evaluated.

Next to interpolation of the coefficients, some derivatives have to be computed as well. These are defined by $\frac{y_{k+1}-y_k}{x_{k+1}-x_k}$ and basically represent the slope between two table entries. Finally, data may be accessed that are outside of a particular table range, which requires extrapolation. However, as no information is known about the behavior of the aerodynamic coefficients outside the table ranges, the extrapolation method is set to 'clip',

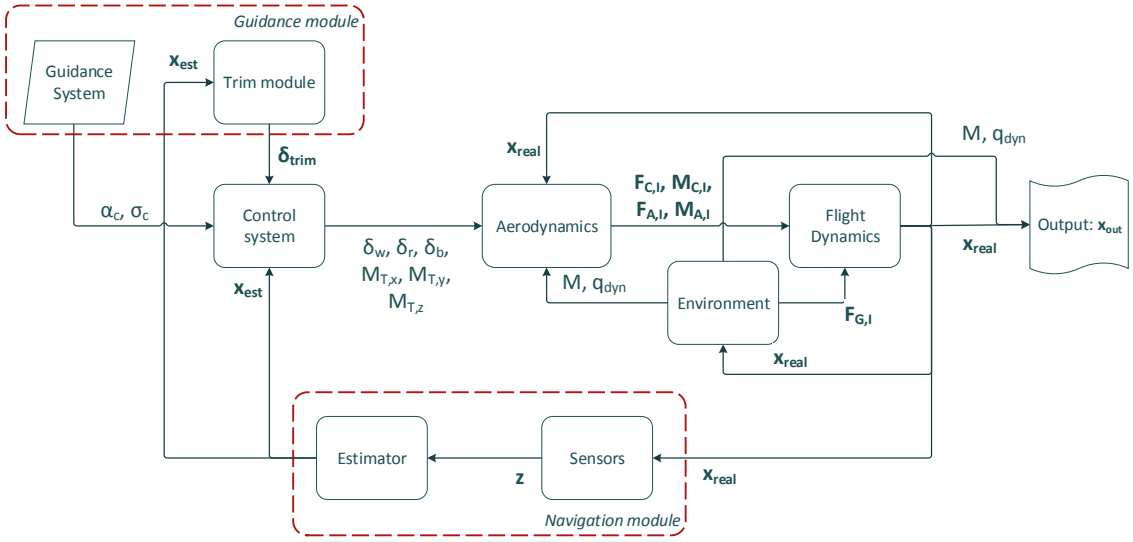


Figure 6.1: Schematic representation of the top layer of the simulator. Note that for convenience, the guidance block is decoupled from the control system and that the control module and navigation module are divided into multiple blocks.

which means that the boundary value of the table is used. In case of a derivative, the slope between the last two table entries in the respective dimension is used.

6.3 Architecture

In the previous sections all design choices that were made in the development of the simulator were discussed. Some of these choices were made before the design process was started, whereas others were made during the development. Next to that, the numerical methods that are to be used in the simulator were evaluated in Section 6.2. As the simulator is the main tool in the research, its architecture is discussed into further detail in this section. The top layer will be described first, after which separate architectures are shown for the state flow of the environment block, the aerodynamics block, the flight dynamics block, the guidance and control module and finally the navigation module. Each of these blocks is described in its own subsection.

6.3.1 Top Layer

It was stated that the simulator was programmed in the Simulink environment, which is suitable for creating a clear overview of different calculation modules. In this environment one can cluster calculations in different blocks and by simply drawing lines between these blocks, the state flow can be activated. Using this, the simulator was divided into multiple levels, with the top layer containing the blocks as seen in Figure 6.1. In this figure it can be seen that the guidance module consists of a system containing the guidance laws, which in this case is reduced to a predetermined attitude profile and a trim module, in which the elevator and body flap commands are computed for trimmed flight. The input of the trim module is the estimated state vector from the navigation module. Then, using the guidance commands, the trim commands and the estimated state, the control system computes the control commands, which are the aerodynamic surface deflections and the RCS thruster moments. The aerodynamics block can be seen as the actuator module, as it computes the aerodynamic coefficients of both the clean configuration and the additions from the control surface deflections. From this, the clean aerodynamic forces and moments $\mathbf{F}_A$ and $\mathbf{M}_A$ are computed, as well as the control forces and moments $\mathbf{F}_C$ and $\mathbf{M}_C$. The control moments consist of the addition of the RCS thruster moments and the induced moments due to the aerodynamic control forces. Finally, the four output force and moment vectors are transformed to the inertial frame, after which they are fed as input to the flight dynamics block.

The only remaining force that is to be computed is the gravitational force, which is done in the environment block. This vector is also transformed to the inertial frame, after which it is sent to the flight dynamics block. In the environment module also the atmospheric properties are computed, depending on the current position and velocity of the vehicle. The Mach number and dynamic pressure are directly used in the aerodynamics block for the computation of the aerodynamic and control forces and moments. Then, after all input components

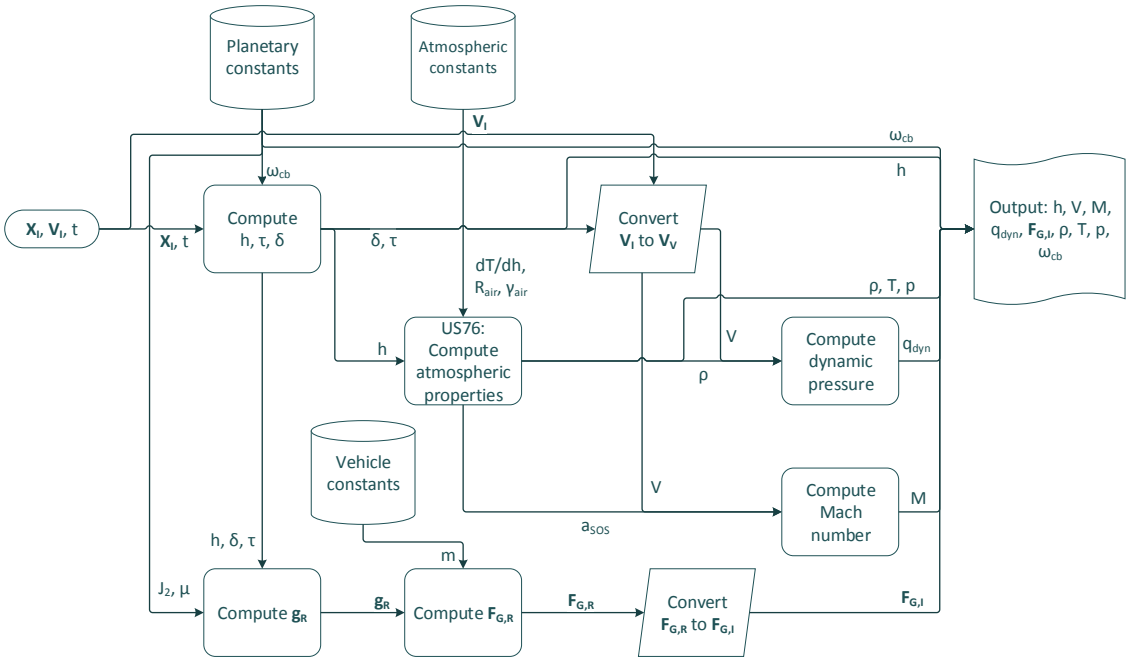


Figure 6.2: Schematic representation of the environment block of the simulator.

of the flight dynamics block are computed, the state of the vehicle is propagated using the ODE45 integrator that was described in Section 6.2.1. The output of this block, which is the true state, is sent to the navigation module, which generates measurements depending on this state and provides a state estimate using a filter, as was described in Section 5.1. The initial state of the vehicle is defined as

$$\mathbf{x}_0 = \left(\mathbf{X}_{I,0}, \mathbf{V}_{I,0}, \mathbf{Q}_0, \boldsymbol{\omega}_0\right)^T \tag{6.15}$$

and contains the initial position and velocity in the inertial frame, the initial attitude expressed in quaternions and the rotational rates of the body frame with respect to the inertial frame. The simulator output is stored in the vector $\mathbf{x}_{\text{out}}$ and contains the appended state of the vehicle at every time step. It is defined as

$$\mathbf{x}_{\text{out}} = \left(t, \mathbf{X}_I, \mathbf{X}_R, \mathbf{V}_I, \mathbf{V}_V, \mathbf{Q}, \phi, \theta, \psi, \alpha, \beta, \sigma, \boldsymbol{\omega}, \delta_{e,r}, \delta_{e,l}, \delta_{e,trim}, \delta_{b,trim}, \delta_{r,r}, \delta_{r,l}, M_{T,x}, M_{T,y}, M_{T,z}, M, \bar{q}\right)^T \tag{6.16}$$

where t is the simulation time, $\mathbf{X}_I$ is the position in the I-frame, expressed in Cartesian components and $\mathbf{X}_R$ is the position in the rotating frame, expressed in altitude h, longitude τ, and latitude δ. The velocity is provided in the inertial frame in Cartesian components ($\mathbf{V}_I$) and in the vertical frame ($\mathbf{V}_V$), expressed in velocity V, flight-path angle γ, and heading angle χ. Then, the output vector contains the attitude in quaternions Q, but also expressed in roll, pitch, and yaw angles (ϕ, θ, and ψ), as well as in angle of attack, angle of sideslip and bank angle (α, β, and σ). The rotational rates $\boldsymbol{\omega}$ are the roll, pitch, and yaw rates p, q, and r. Then, the control surface deflections are provided, where the subscript e denotes the elevon, b denotes the body flap, and r represents the rudder. A difference is made between left and right for the elevons and rudders. The elevon trim deflection is given separately, as well as for the body flap, however, as the latter one is not used in corrective control, $\delta_{b,trim} = \delta_b$. Furthermore, the control moments $M_{T,x}, M_{T,y}, M_{T,z}$ are provided, as well as the Mach number M, and the dynamic pressure $\bar{q}$.

6.3.2 Environment

From Figure 6.4 it is clear that some variables from the environment block are required for computations that have to be made to obtain the state derivative. Figure 6.2 shows the environment module, and it can be seen that the input are the position and velocity of the vehicle, together with the simulation time and thus the environment calculations are dependent on the output of the flight dynamics block. As was described in Chapter 4, two parts can be distinguished that together define the re-entry environment. The first part is

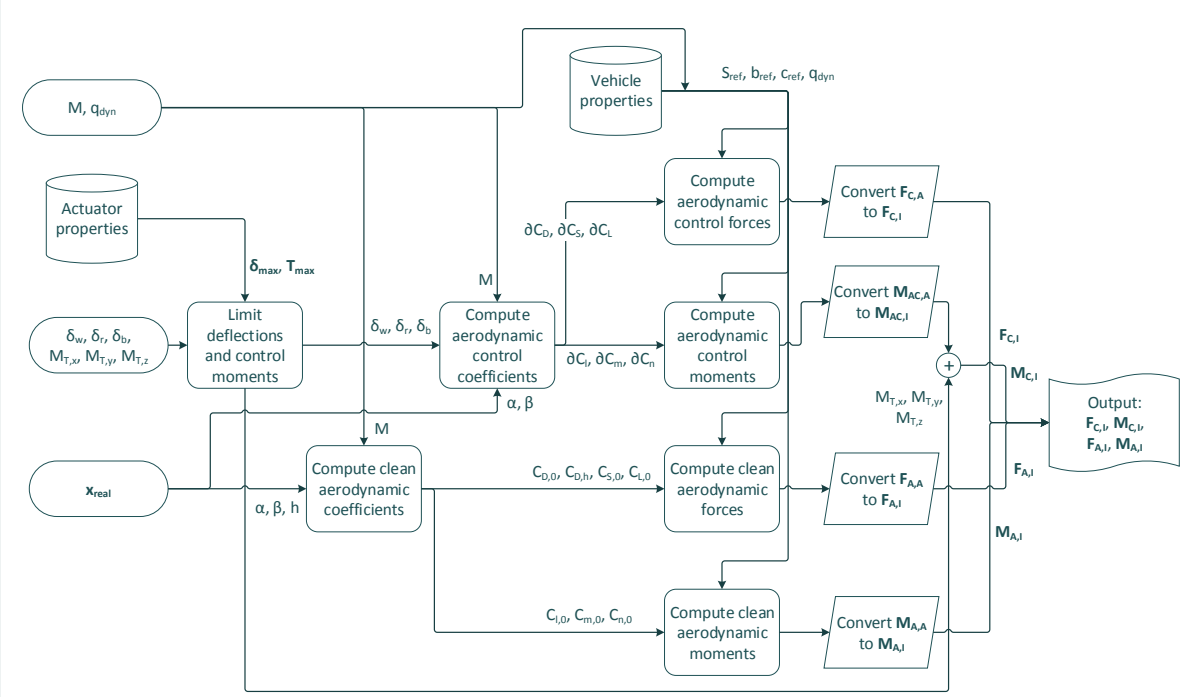


Figure 6.3: Schematic representation of the aerodynamics block of the simulator.

the gravitational field, for which first the position is transformed to the rotating frame. Then, using longitude, latitude and altitude, along with the J_2-constant and the gravitational parameter, the gravitational acceleration is computed. Then, this vector is used to compute the gravitational force, which is subsequently expressed in the inertial frame.

The atmospheric model is the second part that is considered, and in Figure 6.2 it is seen that to compute the atmospheric properties, the position in the R-frame is used, as well as the atmospheric constants. The output of the US76 block contains air pressure, density, temperature and the speed of sound, which are then used to compute the dynamic pressure and the Mach number. For doing this the magnitude of the velocity in the V-frame is used, which requires a transformation of the input velocity of the block. Then, as was seen in Figure 6.1, the Mach number and dynamic pressure are sent to the aerodynamics module, which will be discussed in the following subsection. The gravitational force is fed to the flight dynamics block, which is handled further on in this chapter.

6.3.3 Aerodynamics

The second block that is directly linked to the flight dynamics is the aerodynamics module. Its architecture is seen in Figure 6.3. This figure shows the aerodynamics block, which can be seen as the actuator module of the simualtor. The Mach number and dynamic pressure are obtained from the environment module, whereas the true state is retrieved from the flight dynamics block, which is discussed in the following subsection. Next to that the control surface deflections and RCS thruster moments are provided by the control system. These are then compared to their maximum and minimum, leading to the final control commands that are used for the computation of the aerodynamic control force and moment contributions.

Mooij (1995) described the HORUS-2B vehicle and provided the whole aerodynamic database, which was deduced from MBB (1988). For all vehicle-based aerodynamic coefficients, the relevant tables are loaded into the simulator. Then, using linear interpolation, as described in Section 6.2.2, the aerodynamic force and moment coefficients are found for the corresponding Mach number, angle of attack, angle of sideslip and altitude. In these computations a difference is made between the coefficients for the clean configuration and the contributions due to the control surface deflections. From the former the clean aerodynamic forces and moments are computed, whereas for the latter the control forces and moments are calculated. The final control moment then consists of the addition of the aerodynamic control moment and the RCS thruster moments. Finally, all computed forces and moments are converted to the inertial frame.

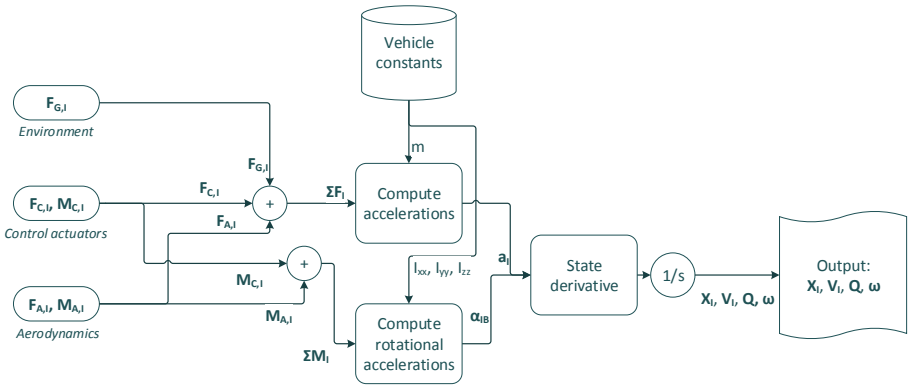


Figure 6.4: Schematic representation of the flight dynamics block of the simulator.

6.3.4 Flight Dynamics

The third block that is described into more detail is the flight dynamics block, which is shown in Figure 6.4. This figure shows the state flow in the flight dynamics module, and three inputs are stated. From the environment the gravitational force is obtained, which was converted to the inertial frame. Then, from the aerodynamics module, both the aerodynamic and control forces and moments are obtained, all expressed in the inertial frame as well. This is done as the state propagation is performed in the inertial state.

In the flight dynamics block all forces and moments are added to ΣF and ΣM. From the total force the accelerations can be computed by simply dividing this vector by the mass of the vehicle, as was stated in Section 3.3. A similar computation is done with the sum of moments; from this vector and the inertia properties of the vehicle, the rotational accelerations are found. Using the EoM for position, velocity, attitude and angular rates the state derivative is found, which is subsequently integrated to obtain the new state, which directly is the output of this module.

6.3.5 Guidance and Control

Then, the guidance and control system implementation are described, which is sufficient for providing controlled flight. In Figure 6.5 these systems are shown, where the guidance module is found in the top-left corner and consists of two blocks. The guidance system contains the guidance laws, which in this case only consist of an attitude profile. The trim module is used to compute the control surface deflections that are required for trimmed flight. Using this guidance module and the previously described blocks, an open-loop simulation can be performed.

For the control module, the input state vector is defined to be an extended version of the estimated state, which is the output of the navigation module that is described in the following subsection. This estimated state contains position, velocity, attitude, angular rates, dynamic pressure and Mach number. In the control block, the estimated attitude and angular rates are compared to the commanded angles from the guidance system. The angles and angular rates that are not defined in the guidance system are commanded to be zero. The attitude and angular rate difference vectors that result from these computations are fed to the controller, which is divided in a segment for longitudinal control and a segment for latitudinal control. In both controllers, the LQR optimal gains are computed via input tables, where the estimated dynamic pressure is used for the breakpoints and where linear interpolation is used to obtain the correct gains for every point of the re-entry trajectory. The controller output is in the form of elevator, aileron and rudder deflection and roll, pitch and yaw control moments. The final computation that is done in the control system software is the conversion from elevator and aileron to elevon deflection. The output of the control software is then a vector containing elevon, rudder and body flap deflection and the RCS moments.

6.3.6 Navigation

At this point the architectures of all blocks that are required to perform a controlled simulation of the re-entry flight of HORUS 2B are described. However, with this simulator one would take the output state from the flight dynamics module as an input for the control system, which cannot be done in real life, as the true state of the vehicle is not known. Therefore, the navigation module is to be included, which uses sensors and an

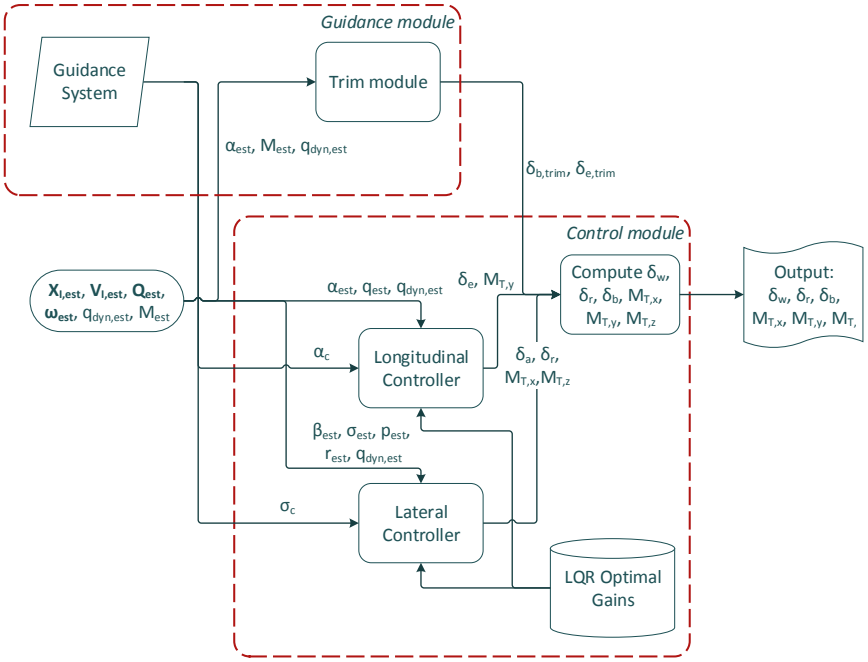


Figure 6.5: Schematic representation of the control module of the simulator.

estimator to approximate the vehicle's state. The navigation system that will be used in the simulator is shown in Figure 6.6.

It can be seen that one of the input blocks contains the actual state of the vehicle, and is fed to three sensors that were described in Section 5.1.1. In the IMU block, the gravitational force is converted to an acceleration vector, using the mass from the vehicle database. From this vector and the total acceleration, the non-gravitational acceleration is computed and modified by applying a bias, scaling error, axis misalignments and sensor noise. Finally, a transformation from inertial to body frame is done, which leads to the first set of measurements, namely the translational acceleration of the vehicle. The second part of the IMU contains a set of gyros, which basically perform similar computations on the input. The rotational rate of the vehicle is fed to this block, which computes the completing set of measurements of the IMU by applying gyro drift, scaling errors, misalignments and noise.

The second block that is evaluated contains the GPS receiver, which takes position, velocity and time as input. A GPS database is loaded, which contains orbital data of all GPS satellites at a specific point in time. These elements are integrated to the actual time, after which they are transformed to Cartesian components, leading to the GPS position and velocity vectors. Then, by comparing the position of all satellites and the vehicle, the visible satellites are selected using a minimum elevation angle. These position and velocity vectors of both the visible GPS satellites and the vehicle, together with the attitude of the vehicle, are then used to compute the pseudorange, range rate and carrier phase differences, including GPS clock bias and drift, and GPS receiver noise for all measurements.

Subsequently, the FADS is addressed, which normally is composed of a sensor and an integrated estimator, which computes angle of attack, angle of sideslip and Mach number. However, as raw data are to be used for the estimator, it was decided to only use the sensor part of the system. From the flight dynamics module and the environment block the velocity and attitude of the vehicle are obtained, next to the Mach number and air pressure. Using the models shown in Section 5.1.1 and the FADS database containing information on the ports that are used the FADS pressure measurements are calculated. Finally, a noise vector is subject to these measurements to create the output of the FADS system.

In Section 5.1.2 the use of the integrated estimator of the navigation module was described. The second input of this module, which is in the bottom of Figure 6.6, is the estimation from the previous time step, which is propagated to the current time step by Euler integration, leading to the prediction of the state. Then, using the measurements from the mentioned sensor combined and a mathematical model that relates these measurements to the state variables, the estimation of the state is completed, thereby obtaining the output of

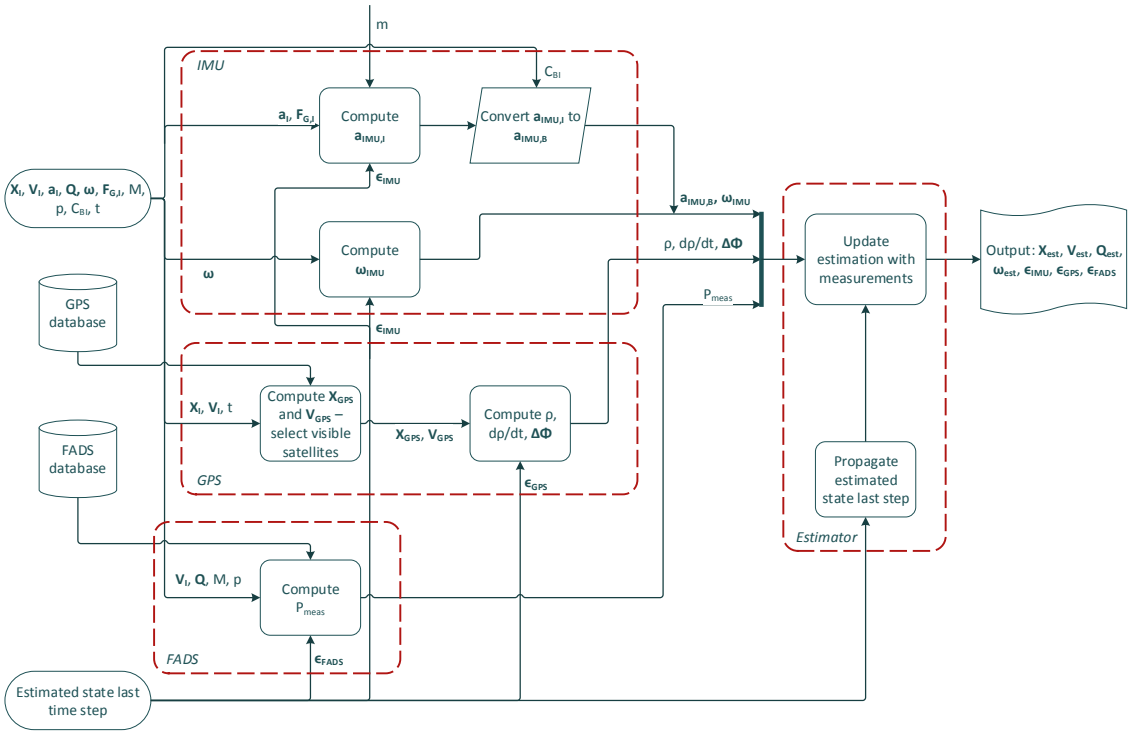


Figure 6.6: Schematic representation of the navigation module of the simulator.

the navigation module. Note that next to position, velocity, attitude and rotational rate, this output vector also contains the selected sensor errors that are estimated during flight.

This concludes the section on the architecture of the simulator, which started by defining the top layer, after which the first level-blocks were viewed. As the simulator is a complex system, a close up of the individual blocks from the first level architectures can be provided, however, only describing this much detail, which is done throughout the report, is sufficient for understanding what happens in these blocks.

6.4 Verification and Simulator Testing

In the previous section the architecture of the simulator was shown and for all top-layer blocks the signal flow was illustrated. This section then will describe the verification methods that were used for the verification of these blocks. For system verification the simulator is tested to fly the nominal trajectory, which is defined by the control history of the guidance system.

6.4.1 Module Verification

Earlier in this chapter it was stated that the simulator was built using the Aerospace Blockset in the MATLAB/Simulink environment. However, it was found that the computational load of the simulator was very high, due to the usage of this blockset. Then, from personal communication with Mooij (2017) it was decided to use the simulator that he developed earlier, which was also programmed in the MATLAB/Simulink environment and contained a fully verified system for the open-loop simulation. However, acceptance tests of the different modules and the system as a whole can still be performed to see whether the system requirements from Chapter 2 are met.

- *Graviational Field* - In terms of the gravitational field that was used in the simulator, the behavior of the gravitational acceleration was evaluated. This vector contained a longitudinal component, which was neglected due to the assumed symmetry of the Earth around its rotational axis. Then, the simulated values were compared to manually computed values for a model not containing the J_2-effect, thereby evaluating the correctness of the simulated magnitude of the individual gravitational field components. A similar approach was taken for the acceptance test of the gravitational acceleration vector in the

inertial frame. Finally, the magnitude of this vector was evaluated over the whole trajectory, and it was seen that it increased for decreasing altitude.

- *Atmospheric Model* - For the acceptance test of the atmospheric model a similar approach was taken as for the gravitational field. The temperature profile that was determined by the output of the US76 block was compared to Figure 4.2, and was found to be similar. Next to that, random pressure and density values for specified altitudes were taken and compared to the tabulated values from NASA (1976), and it was seen that these values were approximated accurately.

- *Aerodynamics* - The aerodynamics module was first tested on the input values that were obtained from the true state (as the navigation module was not applied yet) and the Mach number and dynamic pressure from the environment module. Acceptance tests of the lift, drag and side force were performed by comparison with results from a study on the application of an LQR controller (Mooij, 1997). For the verification of the control forces and moments, first the control module had to be evaluated.

- *Guidance* - As was stated earlier, the guidance system only consisted of an input table with the commanded attitude profile and the trim module. For the former, the only acceptance test that was done was plotting the commanded angles and comparing these to the control history that was seen in Mooij (2015). The trim module was tested on the input, as was also done in the aerodynamics module. It was stated that in this block, aerodynamic coefficients were to be computed, which were obtained from interpolation between tabulated values described by Mooij (1995). Verification of the computed coefficients was done by manually performing the interpolation on a number of randomly selected time steps in the trajectory. The trim commands were compared to the ones that were found in Mooij (1997) and it was seen that these were very similar.

- *Control* - For the control module, a similar approach was taken as for the trim module in the guidance block. Again, on randomly selected points in the trajectory, the aerodynamic coefficient contributions were manually computed and compared to the ones that were calculated in the simulator. Further verification of this module was done by evaluating step and ramp responses, which will be discussed in Section 7.4.

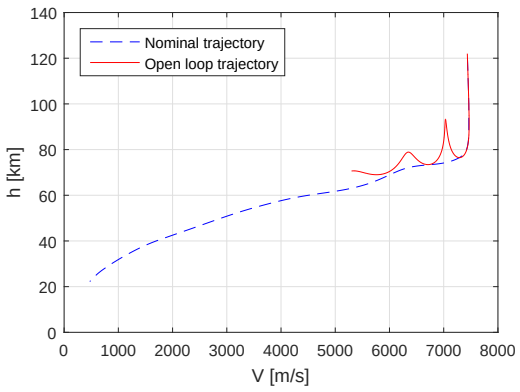
- *Navigation* - The verification of the navigation module was done more thoroughly as this block was programmed during the thesis. As for the sensor module, the IMU was verified as the output acceleration and rotational rates had to be exactly the same as the accelerations and rotational rates during the simulation. The GPS receiver was verified by using sanity checks and calculations by hand. Finally, the FADS sensor was verified by using the estimation methods of the angle of attack, angle of sideslip and Mach number that were described in Section 5.1.1. The verification methods of the sensors, however, are described in more detail in Sections 8.1.1 to 8.1.3. For the estimators, verification was done by manually checking the computations that were performed, as well as by evaluating the convergence of the filter, however, as for the sensor block, the verification of the estimators is discussed in more detail in Sections 8.2.1 to 8.2.3.

- *Flight Dynamics* - Finally, the flight dynamics block was evaluated, which was started with the input forces and moments. These were converted to the I-frame, and for this, transformation matrices were used. These matrices were verified by manually computing the output for a random set of input angles. The acceptance test of the flight dynamics block was then done by manually propagating a simplified system of which the behavior is very straightforward, such as a falling object without aerodynamic drag. This manual propagation was then compared to the same propagation performed by the flight dynamics block.

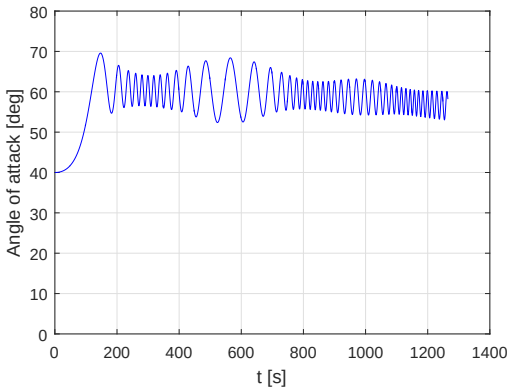
After the module verification and acceptance tests were done, the integrated system was to be verified, which will be described in the following subsection.

6.4.2 Simulator Verification

As was described, the simulator consists of the integration of multiple blocks into one system. It was discussed that these blocks, or modules, were verified individually, which means that the correct output is generated for a correct input. However, integrating all blocks into one system may induce new errors, which can be omitted by performing system verification. In this section the methods for system verification that were used



(a)



(b)

are discussed and the results of the open-loop simulation and closed-loop 6DoF simulation with ideal control are evaluated.

The most reliable method of system verification would be to compare the results of the simulator to an already verified simulator. In Mooij (1997) the design of an LQR is described, in which a re-entry trajectory simulator is used, which is comparable to the simulator that is described in this report. In this paper, both the results of an open-loop simulation and a closed-loop 6DoF simulation are shown and can be used for comparison with the results that are obtained by the simulator that was described in this chapter. The open-loop simulation basically represents a free fall of the vehicle, as no guidance and control is applied. Results of this simulation, however, are hard to verify, as the simulated motion of the vehicle should be compared with the eigenmotions, due to the oscillations that are induced during flight. For system verification of the simulator, a 6DoF closed-loop simulation, is performed as this contains an activated trim module, as well as a guidance and control system to ensure a flight with controlled translational and rotational motion. As the implementation, verification, and testing of the control system are discussed in Chapter 7, it is chosen to simulate the nominal trajectory. This is done by assuming ideal control, hence, the attitude angles are exactly the same as the commanded angles from the guidance system.

When the baseline simulator was fully integrated, it appeared that the process of verification is an iterative process, as the removal of a particular error source may induce a different error elsewhere in the simulator. In the design process, errors were traced and corrected, which ultimately lead to a verified simulator. For finding these errors, a real simple verification method was used, which is the so-called sanity check. With this technique, the possibility of correctness of the value of a variable is evaluated by the programmer, and when this value is deemed incorrect, a lead to an error is found and the tracing process can be started.

6.4.3 Open-Loop Simulation

The first test that was done when the baseline simulator was integrated was the open-loop simulation, in which trimmed flight only is considered. As there is no corrective control on the vehicle, the results from this test are not necessarily used to draw conclusions concerning the behavior of the vehicle, but are compared to open-loop simulations from Mooij (1997), which is used as system verification for the baseline simulator. Before any test was started, the initial conditions were defined from Mooij (2015):

$$h_0 = 119.96 \text{ km}, \tau_0 = -105.97^\circ, \delta_0 = -22.060^\circ,$$
$$V_0 = 7438.1 \text{ m/s}, \gamma_0 = -1.4263^\circ, \chi_0 = 70.437^\circ \tag{6.17a}$$

For the open-loop simulation, the initial conditions for the attitude angles are taken to be: $\alpha_0 = 40^\circ$, $\beta_0 = 0^\circ$, and $\sigma_0 = 0^\circ$ (Mooij, 2015). In Figure 6.7a the altitude is plotted versus the velocity, and this is done for the open-loop simulation and the 6DoF closed-loop simulation that is described in more detail in the following subsection. Figure 6.7b then shows the behavior of the angle of attack. As stated earlier, no conclusions will be drawn from this, however, these results are compared with the open-loop results presented in Mooij (1997) and it is seen that both show similar behavior, hence, using this test system verification is performed. However, to fully evaluate whether the angle of attack behavior is correct, one should evaluate the eigenmotion of the vehicle, and relate this to the oscillations that are shown in Figure 6.7b. As the system verification is also performed by the simulation of the nominal trajectory and as the evaluation of the eigenmodes of the vehicle is

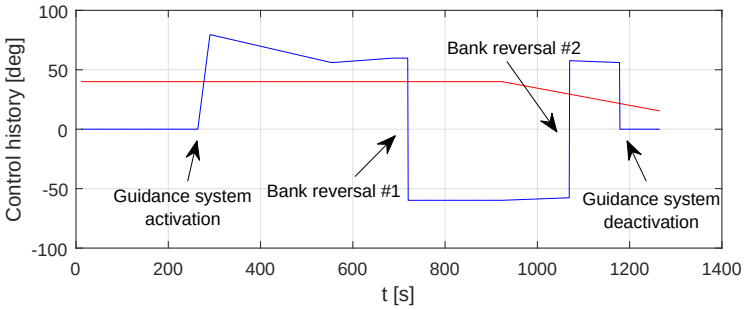
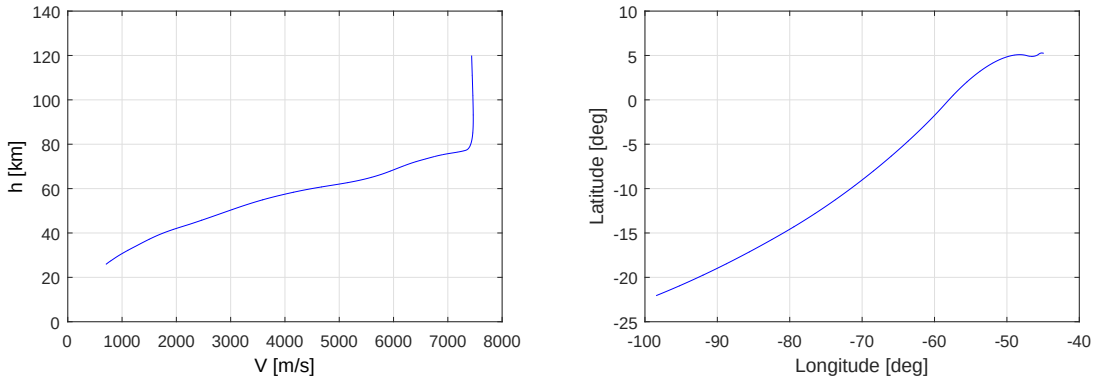


Figure 6.8: Control history of angle of attack and bank angle in degrees.



(a) (b)

Figure 6.9: (**a**) - Altitude versus velocity for the nominal mission. (**b**) - Latitude versus longitude for the nominal mission.

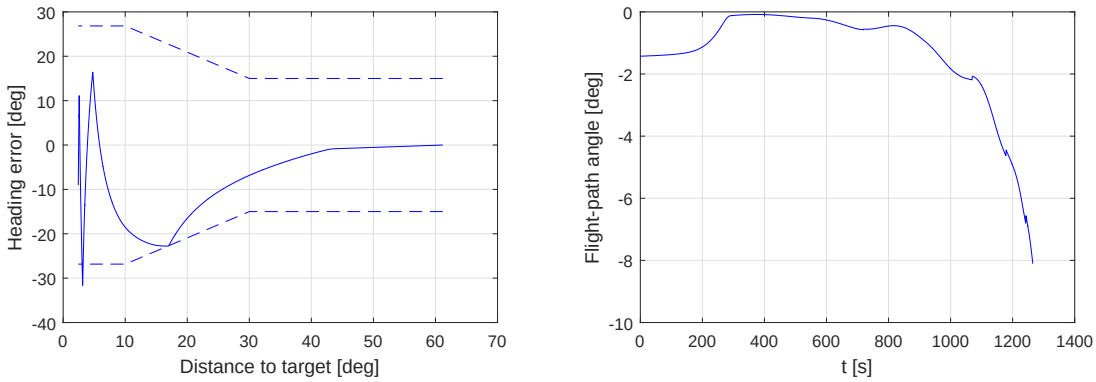
beyond the scope of this thesis, this is not done. The angle of sideslip and bank angle showed similar oscillating behavior, and from Figure 6.7a it can be stated that for trimmed flight only, the descent and deceleration in the simulation time of $t = 1265$ s is much less than for a controlled flight, thereby proving the importance of a guidance logic and a control system.

6.4.4 Nominal Trajectory

The second part of the testing of the baseline simulator was the simulation of the nominal trajectory, in which the configuration from the previous subsection is extended by a guidance system containing the guidance laws, and a control system. However, as this concerns the verification of the baseline simulator, and not of the designed control system (as this is handled in Chapter 7), it is chosen to assume ideal control, meaning that the actual attitude of the vehicle is exactly the same as the commanded attitude from the guidance system. These commands are shown in Figure 6.8.

This figure shows the commanded angle-of-attack and the bank-angle profiles. The angle of sideslip, as well as the roll, pitch, and yaw rates are commanded to be zero at all times. It is seen that at $t \approx 265$ s, the guidance system is activated, as the bank angle becomes nonzero, resulting in a change of heading. Then, at $t \approx 720$ s and $t \approx 1070$ s, two bank reversals can be seen, which is explained further on in this subsection. The control history from Mooij (2015) then contained more bank reversals after $t = 1070$ s, however, due to the absence of a real guidance system the control system was not able to correct for these reversals, and therefore it is chosen to deactivate the bank-angle guidance at $t \approx 1180$, as the scope of this thesis lies on the application of the navigation module to the 6DoF closed-loop simulator.

In Figure 6.9a then the attitude-velocity profile is shown that was already touched upon in Figure 6.7a. It is seen that for ideally controlled flight, the vehicle is able to descent to approximately 25 km altitude, which is much lower than for the uncontrolled flight. Next to that, the velocity at the TAEM interface is reduced to less than 1 km/s, corresponding to a Mach number of approximately 2.5. Figure 6.9b shows the latitude and longitude of the vehicle during the hypersonic descent, where the initial position is in the bottom left corner. The upper right corner then thus shows the position at the end of the trajectory, and it is seen that at the end of



(a) (b)

Figure 6.10: (**a**) - Heading error for the nominal mission. (**b**) - Flight-path angle for the nominal mission.

the flight, the heading of the vehicle changes more than in the starting phase. This is due to the fact that the velocity of the vehicle is much lower at the end of the trajectory, and as the vehicle is closer to its target, the distance that is available to correct for the heading error is smaller as well. The bank reversals in Figure 6.8 are used for these changes in heading, and therefore occur more often at the end of the entry flight.

The heading error of the vehicle is shown in Figure 6.10a, and it is seen that this variable is plotted against the distance to the target. This distance can be computed as a ratio of a circle, and, assuming a spherical Earth, it thus can be expressed in degrees. The heading error is defined as the difference between the true heading and the heading that is to be flown to reach the target, where the latter is defined as (Mooij, 2015)

$$\chi_T = \arctan\left[\frac{\sin(\tau_T-\tau)\sin\left(\frac{\pi}{2}-\delta_T\right)}{\cos\left(\frac{\pi}{2}-\delta_T\right)\sin\left(\frac{\pi}{2}-\delta\right)-\cos(\tau_T-\tau)\cos\left(\frac{\pi}{2}-\delta\right)\sin\left(\frac{\pi}{2}-\delta_T\right)}\right] \tag{6.18}$$

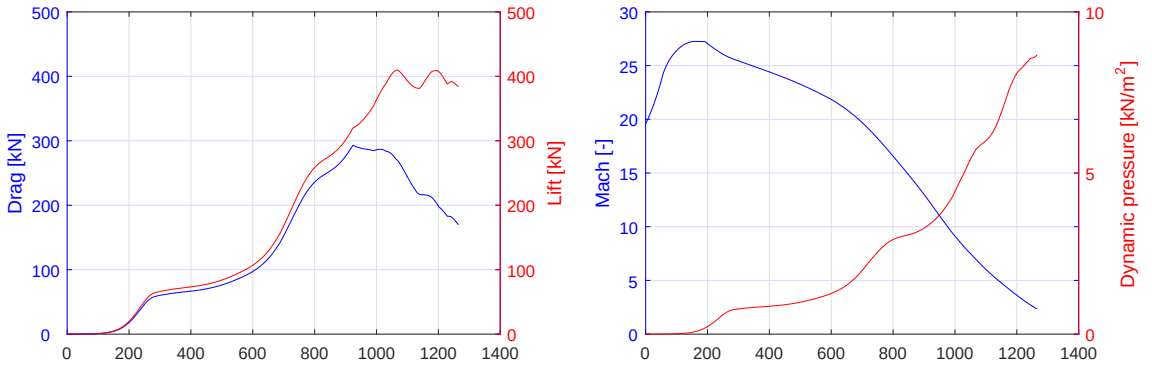
with τ_T and δ_T being the longitude and latitude of the target, which is a runway at Kourou (Mooij, 2015). Using Google Maps, the coordinates of this runway are found to be $\tau_T \approx -52.8°$ and $\delta_T \approx 5.2°$. The distance as described earlier in this paragraph is then obtained via

$$d = \arccos\left[\sin\delta\sin\delta_T + \cos\delta\cos\delta_T\cos(\tau-\tau_T)\right] \tag{6.19}$$

and thus is computed in degrees. In Figure 6.10a it can then be seen that when the vehicle moves closer to the target, the use of bank reversals is applied, which can also be spotted as small irregularities in the flight-path angle, shown in Figure 6.10b. Finally, the heading error dead-band is shown in Figure 6.10a, which denotes the constraint for performing a bank reversal. From experience with the Space Shuttle, dead-band values between 10° and 30° can be used (Mooij, 2015) for a reasonable approach. The boundary of the dead-band defines the maximum allowable heading error, which should be kept relatively small to ensure that the vehicle reaches the TAEM interface. Next to that, a high number of bank reversals is not desired, and therefore the dead-band boundary values should not be very small, as this increases the number of required bank reversals.

In Figure 6.11a the lift and drag forces can be seen, which are similar to the nominal lift and drag found in Mooij (1997). Figure 6.11b then shows the Mach number, which first increases up to a value of roughly 27, after which it is reduced to approximately 2.5 at the TAEM interface. The dynamic pressure increases over time, and as this can be related to the effectiveness of the aerodynamic actuators, the interpolation of the gain values in Chapter 7 is done with this variable as breakpoints. Finally, in Figure 6.12, the trim deflections of the body flap and the elevators are shown. The deflections of the elevators are zero throughout a large part of the flight, and it can be seen that they are actuated when the body flap reaches its maximum deflection, which is only at the very end of the trajectory. For operational purposes it was chosen that the body flap was activated when the dynamic pressure rose above 100 N/m^2, and before that it is chosen that the it maintains a trim angle of 15 degrees.

As was described earlier in this section, the results for both the open-loop and the 6DoF closed-loop (nominal trajectory) simulation, the results were compared to similar research described in Mooij (1997). As the final baseline simulator was retrieved from Mooij (2017), it was already verified, and the results that are shown in this section are used as acceptance tests, as small parts differed from the research described in Mooij



(a) (b)

Figure 6.11: (**a**) - Lift and drag forces for the nominal mission. (**b**) - Mach number and dynamic pressure for the nominal mission.

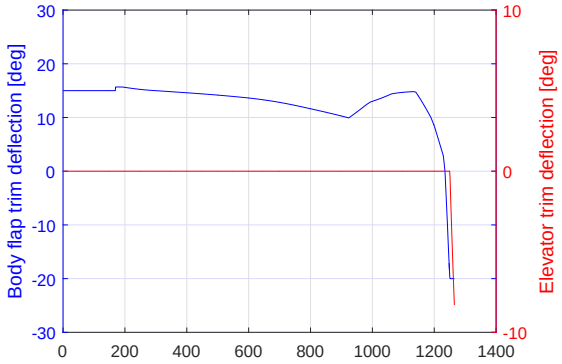


Figure 6.12: Trim angles of the body flap and elevators for the nominal mission.

(1997), such as the fact that the bank-angle guidance is deactivated for this mission. As was noted, this section concerned the baseline simulator, and thus the actual control system and the navigation module were not considered yet. The verification of these systems however is described in Chapters 7 and 9.

7

Implementation and Simulation of the LQR Controller

In the previous chapter the architecture of the simulator, which was built for the research that is described in this thesis, was shown. First the open-loop simulation was tested, using the simulator without control and navigation system. Then, the nominal mission was simulated, using the exact mapping of the guidance profile commands to the aerodynamic angles. In real life this cannot be done, and a control system is required, which was described in Section 5.2. The theoretical background of the LQR was discussed, as well as the actuator system of the reference vehicle, however, no implementation and verification aspects were handled.

This chapter will take the theory from Section 5.2, and state the results of the derivation of the state-space system that is used to obtain the optimal gain matrix (Section 7.1), which contains the linearization process of the EoM. Then, the controller design is discussed in Sections 7.2 and 7.3, and, finally, the verification of the controller, together with its integration in the simulator, is handled in Section 7.4.

7.1 State-Space System

For the design of the LQR controller, the EoM that were derived in Sections 3.5 and 3.6 are used, only for this, they are transformed from the Cartesian system to a spherical system. In this section, the final state-space system is shown that is used for the design of the longitudinal and the lateral controller, after which the eigenmotions of the system are briefly evaluated. In Section 5.2, two important equations were obtained that will be restated here for convenience. First, the matrix Riccati equation is defined as

$$\mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}\mathbf{P} + \mathbf{Q} = \mathbf{0} \tag{7.1}$$

which is used to compute the matrix $\mathbf{P}$ that is used in the computation of the gain matrix:

$$\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{P} \tag{7.2}$$

In these equations, a distinction is made between the gain matrix $\mathbf{K}$, the positive definite matrix $\mathbf{P}$ that was mentioned above, and the four matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{Q}$, $\mathbf{R}$. The former two of these matrices are derived from the EoM, as was stated earlier in this section. The weight matrix $\mathbf{Q}$ then defines the control deviation, in which the maximum deviations from the nominal values are stated for the attitude angles and the rotational rates. Then, in $\mathbf{R}$, the maximum values of the control actuators are used to compute the so-called control effort. For the computation of the control gains, the state-space model of the EoM is to be defined, of which the derivation is shown in Mooij (1997).

This derivation is started by stating the EoM in terms of radius R, longitude τ and latitude δ for the position and velocity V, flight-path angle γ and heading χ for the velocity. The attitude and rotational rates are represented by angle of attack α, angle of sideslip β, bank angle σ, roll rate p, pitch rate q, and yaw rate r. The former six variables define translational motion, whereas the latter six are used for the definition of the rotational motion. As the controller only provides control of the rotational motion of the vehicle, only the equations for the attitude and the rotational rates are used in the design of the LQR.

The derivation of the state-space model that is used for the design of the controller contains two main steps. First, the EoM are simplified by setting a number of assumptions. From Mooij (1997) it is assumed that:

- A non-rotating Earth is considered, hence $\omega_{cb} = 0$ rad/s.
- The vehicle is symmetrical in the OX_BZ_B-plane, hence $I_{xy} = I_{yx} = I_{yz} = I_{zy} = 0$.
- The vehicle is rotationally symmetric, hence $I_{xz} = I_{zx} = 0$.
- The gravitational field is spherical, leading to only a radial component g.
- The influence of asymmetric translational motion is neglected, which is done by assuming that the vehicle does not move in North-South direction, hence $\dot{\delta} = 0$ rad/s. Next to that, a trajectory parallel to the equator is assumed, leading to $\delta = 0°$ and $\chi = 90°$.

Due to the fact that the commanded angle of sideslip is zero throughout the trajectory, the nominal sideforce S_0 is also zero. Furthermore, as the attitude angles α, β, and σ are defined by the control history of the guidance system and are set to be the nominal control, defining $\dot{\alpha} = \dot{\beta} = \dot{\sigma} = 0$ rad/s discards any perturbations of these commanded variables. The second part of the state-space model derivation is linearization of the simplified EoM, which is done by a first-order Taylor approximation

$$\mathbf{x} = \mathbf{x}_0 + \Delta\mathbf{x} \tag{7.3}$$

where $\mathbf{x}_0$ defines the nominal state of the vehicle and $\Delta\mathbf{x}_0$ is a small deviation from this state. The derivation and linearization of the simplified EoM is found in Mooij (1997). As was stated earlier in this section, only rotational motion is considered, which leads to a linearized state vector:

$$\Delta\mathbf{x} = \left(\Delta p, \Delta q, \Delta r, \Delta\alpha, \Delta\beta, \Delta\sigma\right)^T \tag{7.4}$$

The control vector $\mathbf{u}$, which will be used further on in this chapter is defined as:

$$\Delta\mathbf{u} = \left(\Delta\delta_e, \Delta\delta_a, \Delta\delta_r, \Delta M_{T,x}, \Delta M_{T,y}, \Delta M_{T,z}\right)^T \tag{7.5}$$

In this equation, the former three variables denote the elevator, aileron, and rudder deflections, with subscripts e, a, and r respectively. The latter three variables represent the thruster moments over the x, y, and z axes of the vehicle's body frame.

The linearized state-space system was defined by

$$\Delta\dot{\mathbf{x}} = \mathbf{A}\Delta\mathbf{x} + \mathbf{B}\Delta\mathbf{u} \tag{7.6}$$

where $\mathbf{A}$ and $\mathbf{B}$ are reduced to only rotational motion and are given by:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 0 & a_{p\beta} & 0 \\ 0 & 0 & 0 & a_{q\alpha} & 0 & 0 \\ 0 & 0 & 0 & 0 & a_{r\beta} & 0 \\ 0 & a_{\alpha q} & 0 & a_{\alpha\alpha} & 0 & a_{\alpha\sigma} \\ a_{\beta p} & 0 & a_{\beta r} & 0 & a_{\beta\beta} & a_{\beta\sigma} \\ a_{\sigma p} & 0 & a_{\sigma r} & a_{\sigma\alpha} & a_{\sigma\beta} & a_{\sigma\sigma} \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 & b_{pa} & 0 & b_{px} & 0 & 0 \\ b_{qe} & 0 & 0 & 0 & b_{qy} & 0 \\ 0 & b_{ra} & b_{rr} & 0 & 0 & b_{rz} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{7.7}$$

The individual partial derivatives are derived from the linearized EoM (Mooij, 1997) and are:

$$a_{p\beta} = \frac{1}{I_{xx}} \frac{\partial C_l}{\partial \beta} \bar{q}_0 S_{\text{ref}} b_{\text{ref}} \tag{7.8}$$

$$a_{q\alpha} = \frac{1}{I_{yy}} \frac{\partial C_m}{\partial \alpha} \bar{q}_0 S_{\text{ref}} c_{\text{ref}} \tag{7.9}$$

$$a_{r\beta} = \frac{1}{I_{zz}} \frac{\partial C_n}{\partial \beta} \bar{q}_0 S_{\text{ref}} b_{\text{ref}} \tag{7.10}$$

$$a_{\alpha q} = 1 \tag{7.11}$$

$$a_{\alpha\alpha} = -\frac{1}{mV_0} \frac{\partial C_L}{\partial \alpha} \bar{q}_0 S_{\text{ref}} \tag{7.12}$$

$$a_{\alpha\sigma} = -\frac{g_0}{V_0} \cos\gamma_0 \sin\sigma_0 \tag{7.13}$$

$$a_{\beta p} = \sin\alpha_0 \tag{7.14}$$

$$a_{\beta r} = -\cos\alpha_0 \tag{7.15}$$

$$a_{\beta\beta} = -\frac{1}{mV_0} \frac{\partial C_S}{\partial \beta} \bar{q}_0 S_{\text{ref}} \tag{7.16}$$

$$a_{\beta\sigma} = -\frac{g_0}{V_0} \cos\gamma_0 \cos\sigma_0 \tag{7.17}$$

$$a_{\sigma p} = -\cos\alpha_0 \tag{7.18}$$

$$a_{\sigma r} = -\sin\alpha_0 \tag{7.19}$$

$$a_{\sigma\alpha} = \frac{\tan\gamma_0 \sin\sigma_0}{mV_0} \frac{\partial C_L}{\partial \alpha} \bar{q}_0 S_{\text{ref}} \tag{7.20}$$

$$a_{\sigma\beta} = \frac{\tan\gamma_0 \cos\sigma_0}{mV_0} \frac{\partial C_S}{\partial \beta} \bar{q}_0 S_{\text{ref}} - \frac{L_0}{mV_0} + \frac{g_0}{V_0} \cos\gamma_0 \cos\sigma_0 \tag{7.21}$$

$$a_{\sigma\sigma} = \tan\gamma_0 \cos\sigma_0 \frac{L_0}{mV_0} \tag{7.22}$$

$$b_{pa} = \frac{1}{I_{xx}} \frac{\partial C_l}{\partial \delta_a} \bar{q}_0 S_{\text{ref}} b_{\text{ref}} \tag{7.23}$$

$$b_{px} = \frac{1}{I_{xx}} \tag{7.24}$$

$$b_{qe} = \frac{1}{I_{yy}} \frac{\partial C_m}{\partial \delta_e} \bar{q}_0 S_{\text{ref}} c_{\text{ref}} \tag{7.25}$$

$$b_{qy} = \frac{1}{I_{yy}} \tag{7.26}$$

$$b_{ra} = \frac{1}{I_{zz}} \frac{\partial C_n}{\partial \delta_a} \bar{q}_0 S_{\text{ref}} b_{\text{ref}} \tag{7.27}$$

$$b_{rr} = \frac{1}{I_{zz}} \frac{\partial C_n}{\partial \delta_r} \bar{q}_0 S_{\text{ref}} b_{\text{ref}} \tag{7.28}$$

$$b_{rz} = \frac{1}{I_{zz}} \tag{7.29}$$

This state-space model will be used in the design of the controller, which is shown later on in this chapter. In a case study described by Mooij (2015) it was found that the symmetric (pitch) motion of the vehicle induces small deviations in the asymmetric (roll and yaw) motion, and vice versa, which means that both motions are coupled. The eigenvalues of the coupled system were then compared to the eigenvalues for decoupled symmetric and asymmetric motion, and it was found that these values were very similar. From this, Mooij (2015) stated that a decoupled system is a good representation of reality. Due to the fact that decoupling symmetric and asymmetric motion simplifies the design of the controller, it is chosen to do this for the design that is described in this chapter, meaning that the optimal gains for the longitudinal controller (Section 7.2) are computed separately from the gains for the lateral controller (Section 7.3).

7.2 Longitudinal Control

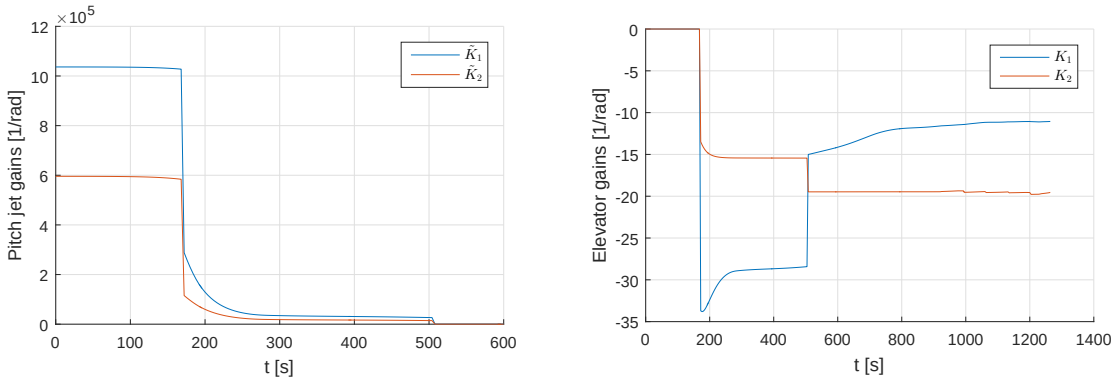
In the previous section it was decided that in the LQR design, symmetric motion was decoupled from asymmetric motion, which means that deviations in roll and yaw do not induce deviations in pitch and vice versa. Due to this assumption, $\Delta\beta$ and $\Delta\sigma$ are set to zero for the design of longitudinal control. The pitch motion is then purely defined by pitch rate q and angle of attack α, resulting in the following state-space model

$$\begin{pmatrix} \Delta\dot{q} \\ \Delta\dot{\alpha} \end{pmatrix} = \mathbf{A} \begin{pmatrix} \Delta q \\ \Delta\alpha \end{pmatrix} + \mathbf{B} \begin{pmatrix} \Delta\delta_e \\ \Delta M_{T,y} \end{pmatrix} \tag{7.30}$$

with

$$\mathbf{A} = \begin{bmatrix} 0 & \frac{1}{I_{yy}} \frac{\partial C_m}{\partial \alpha} \bar{q}_0 S_{\text{ref}} c_{\text{ref}} \\ 1 & -\frac{1}{mV_0} \frac{\partial C_L}{\partial \alpha} \bar{q}_0 S_{\text{ref}} \end{bmatrix} \tag{7.31}$$

$$\mathbf{B} = \begin{bmatrix} \frac{1}{I_{yy}} \frac{\partial C_m}{\partial \delta_e} \bar{q}_0 S_{\text{ref}} c_{\text{ref}} & \frac{1}{I_{yy}} \\ 0 & 0 \end{bmatrix} \tag{7.32}$$



(a) (b)

Figure 7.1: (**a**) - Gain scheduling of pitch jet control. (**b**) - Gain scheduling of elevator control.

Note that in this **B**-matrix, the contributions of the aerodynamic surfaces are included. The state feedback in Equation (5.78) is used to define the controls

$$\frac{\Delta\delta_e}{\delta_{e_{max}}} = -K_1\Delta q - K_2\Delta\alpha \tag{7.33}$$

$$\frac{\Delta M_{T,y}}{M_{T,y_{max}}} = -\tilde{K}_1\Delta q - \tilde{K}_2\Delta\alpha \tag{7.34}$$

In this equation Δq defines the difference between the actual pitch rate and the commanded pitch rate. As the latter is undefined in the guidance system, $\Delta q = q$. For $\Delta\alpha$ it holds that:

$$\Delta\alpha = \alpha - \alpha_c \tag{7.35}$$

Finally, to solve the matrix Riccati equation, the weighting matrices are required, which are simple diagonal matrices containing preset values from Mooij (1997). Normally, the selection of these matrices is done in an iterative manner, however, as they are stated in Mooij (1997) and Mooij (2015), the iterative procedure is beyond the scope of this thesis. This results in

$$\mathbf{Q} = \text{diag}\left(\left[\frac{1}{\Delta q_{max}^2}, \frac{1}{\Delta\alpha_{max}^2}\right]\right) \tag{7.36}$$

$$\mathbf{R} = \text{diag}\left(\left[\frac{1}{\Delta\delta_{e_{max}}^2}, \frac{1}{\Delta M_{T,y_{max}}^2}\right]\right) \tag{7.37}$$

For the definition of the longitudinal control gains, it is used from the HORUS database (Mooij, 1995) that $S_{\text{ref}} = 110$ m^2 and $c_{\text{ref}} = 23$ m. Then, from Mooij (1997) it was found that $\Delta\delta_{e_{max}} = 40°$ and $\Delta T_{y_{max}} = 10400$ Nm. For the definition of $\Delta\alpha_{\max}$ and $\Delta q_{\max}$ the re-entry trajectory was divided in three parts: RCS control, hybrid control and aerodynamic control. The former only uses the thrusters and for the latter only the aerodynamic control surfaces are activated. For hybrid control, both actuators are active, which only occurs in a small part of the trajectory where the air is not thick enough to be efficient for aerodynamic control only, hence the RCS is active as well. For the RCS phase it was defined that $\Delta\alpha_{\max} = 1°$ and $\Delta q_{\max} = 1.5$ deg/s, for hybrid control it was stated that $\Delta\alpha_{\max} = 2.5°$ and $\Delta q_{\max} = 1.5$ deg/s and for aerodynamic control it was found that $\Delta\alpha_{\max} = 2°$ and $\Delta q_{\max} = 4$ deg/s.

Then, after simulating the nominal trajectory and storing its output, the time vector was divided in multiple steps with $dt = 4$ s. For every step in this time vector, the **A** and **B** matrices were computed. Using the control scheme that was described in Section 5.2.2, the available controls were selected. If a particular control method was unavailable, it was discarded from all matrices. The aerodynamic coefficient contributions that appear in **A** and **B** were computed by linear interpolation of the aerodynamic database from Mooij (1995). A derivative was then simply computed by taking the difference of the left and right border and divide this value by the difference of the input variable that corresponds to those two borders. This lead to the following gain schedule for longitudinal control. As can be seen in Figure 7.1a, the pitch jets are functional from the re-entry interface

up until approximately 500 s. The activation of these jets is also found in Figure 7.1b, where the gains show an instant jump. Around $t = 180$ s, the elevators are activated, as can be seen in both figures; in Figure 7.1a the pitch thruster gains decrease instantly, whereas in Figure 7.1b the elevator gains become nonzero. Note that gain scheduling is done before the simulator is run. The gain schemes from Figure 7.1a and Figure 7.1b are tabulated as a function of dynamic pressure and served as an input to the integrated controller in the simulator. Then, for every time step in the simulator, linear integration of these tables is performed to obtain the gains for the controller. It is chosen to use dynamic pressure as the breakpoints of these tables as it is ever increasing over the simulation, as is seen in Figure 6.11b. Next to that, the dynamic pressure is defined as a relation between the atmosphere and the vehicle, and changes depending on the state of the vehicle. Therefore it is much more logical to use this variable instead of time, as time does not contain any information on the state of the vehicle.

7.3 Lateral Control

The previous section contained the design of the longitudinal controller, which concerns only pitch movement. As symmetric and asymmetric motion are decoupled in the design, this type of movement is assumed not to induce lateral motion. This second controller is designed following the same strategy as for the longitudinal controller, however, it contains a coupled motion between roll and yaw. Similarly to the longitudinal controller, it was now assumed that $\Delta\alpha = 0$ and that there is no pitch rate either. Next to that, as only rotational motion is considered, the translational EoM are discarded. This provides the following system:

$$\begin{pmatrix} \Delta\dot{p} \\ \Delta\dot{r} \\ \Delta\dot{\beta} \\ \Delta\dot{\sigma} \end{pmatrix} = \mathbf{A} \begin{pmatrix} \Delta p \\ \Delta r \\ \Delta\beta \\ \Delta\sigma \end{pmatrix} + \mathbf{B} \begin{pmatrix} \Delta\delta_a \\ \Delta\delta_r \\ \Delta M_{T,x} \\ \Delta M_{T,z} \end{pmatrix} \tag{7.38}$$

with

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & \frac{1}{I_{xx}} \frac{\partial C_l}{\partial\beta} \bar{q}_0 S_{\text{ref}} b_{\text{ref}} & 0 \\ 0 & 0 & \frac{1}{I_{zz}} \frac{\partial C_n}{\partial\beta} \bar{q}_0 S_{\text{ref}} b_{\text{ref}} & 0 \\ \sin\alpha_0 & -\cos\alpha_0 & 0 & -\frac{g_0}{V_0} \cos\gamma_0 \cos\sigma_0 \\ -\cos\alpha_0 & -\sin\alpha_0 & \frac{g_0}{V_0} \cos\gamma_0 \cos\sigma_0 - \frac{L_0}{mV_0} & \frac{L_0}{mV_0} \tan\gamma_0 \cos\sigma_0 \end{bmatrix} \tag{7.39}$$

and

$$\mathbf{B} = \begin{bmatrix} \frac{1}{I_{xx}} \frac{\partial C_l}{\partial\delta_a} \bar{q}_0 S_{\text{ref}} b_{\text{ref}} & 0 & \frac{1}{I_{xx}} & 0 \\ \frac{1}{I_{zz}} & \frac{\partial C_n}{\partial\delta_a} \bar{q}_0 S_{\text{ref}} b_{\text{ref}} & \frac{1}{I_{zz}} \frac{\partial C_n}{\partial\delta_r} \bar{q}_0 S_{\text{ref}} b_{\text{ref}} & 0 & \frac{1}{I_{zz}} \\ 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & \end{bmatrix} \tag{7.40}$$

The control laws are defined similarly to the ones that were used for longitudinal control:

$$\frac{\Delta M_{T,x}}{M_{T,x_{max}}} = -\frac{\Delta\delta_a}{\delta_{a_{max}}} = -K_3\Delta p - K_4\Delta r - K_5\Delta\beta - K_6\Delta\sigma \tag{7.41}$$

$$\frac{\Delta M_{T,z}}{M_{T,z_{max}}} = -\frac{\Delta\delta_r}{\delta_{r_{max}}} = -K_7\Delta p - K_8\Delta r - K_9\Delta\beta - K_{10}\Delta\sigma \tag{7.42}$$

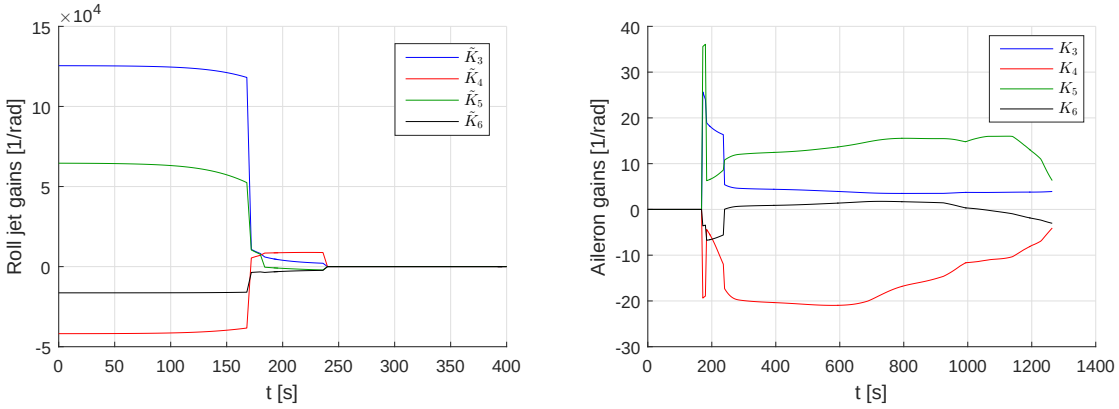
Note that these notations are valid for the aileron and rudder gains; for the RCS moment gains, the $\tilde{K}$ notation is used, however, the equations remain valid. For the lateral control laws, it will be used that the commanded values of the angle of sideslip, the roll and yaw rate are zero, and that only the bank angle has a guidance profile, and thus is defined by $\Delta\sigma = \sigma - \sigma_c$. The **Q** and **R** matrices are set up similarly as well, with

$$\mathbf{Q} = \text{diag}\left(\left[\frac{1}{\Delta p_{max}^2}, \frac{1}{\Delta r_{max}^2}, \frac{1}{\Delta\beta_{max}^2}, \frac{1}{\Delta\sigma_{max}^2}\right]\right) \tag{7.43a}$$

$$\mathbf{R} = \text{diag}\left(\left[\frac{1}{\Delta\delta_{a_{max}}^2}, \frac{1}{\Delta\delta_{r_{max}}^2}, \frac{1}{\Delta M_{T,x_{max}}^2}, \frac{1}{\Delta M_{T,z_{max}}^2}\right]\right) \tag{7.43b}$$

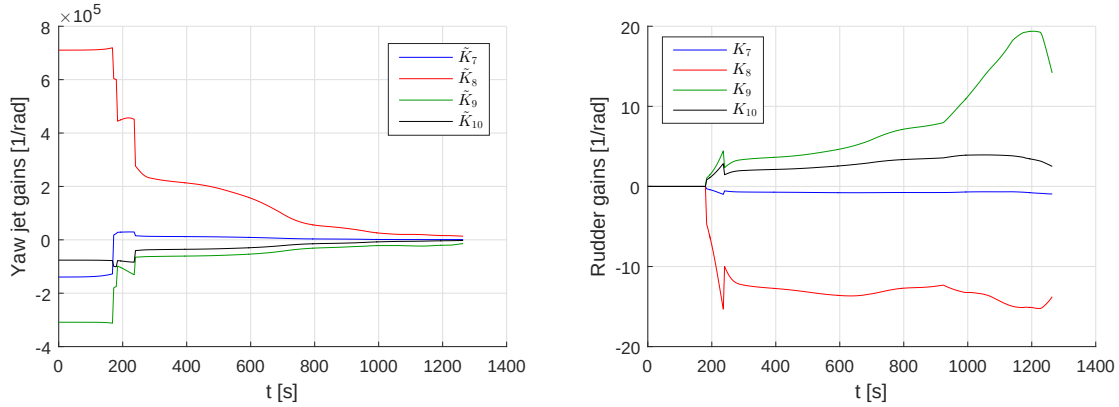
Table 7.1: Scheme for the definition of the weighting matrix of the lateral controller for all three types of control.

	Roll		Yaw	
	Δp_{max} [deg/s]	$\Delta \sigma_{max}$ [deg]	Δr_{max} [deg/s]	$\Delta \beta_{max}$ [deg]
RCS	1.5	4	1.5	1
Hybrid	2.5	4	10	4
Aero	10	10	10	4



(a) (b)

Figure 7.2: (**a**) - Gain scheduling of roll jet control. (**b**) - Gain scheduling of aileron control.



(a) (b)

Figure 7.3: (**a**) - Gain scheduling of yaw jet control. (**b**) - Gain scheduling of rudder control.

where $\Delta\delta_{e_{max}} = 40°$, $\Delta\delta_{r_{max}} = 40°$, $\Delta M_{T,x_{max}} = 1600$ Nm and $\Delta M_{T,z_{max}} = 7600$ Nm. The scheme for the definition of the **Q**-matrix is chosen in the same fashion as was done for the longitudinal controller, where the roll rate is coupled to the bank angle and the yaw rate to the angle of sideslip. The scheme that was used is summarized in Table 7.1. Note that for roll and yaw motion, the activation and deactivation of the actuators is not necessarily to be done simultaneously. Thus, when roll motion is dependent on hybrid control, it does not implicitly mean that yaw motion is also dependent on hybrid control. Using the above defined state-space system in combination with the control laws, weighting matrices and the control scheme from Table 7.1, the following gain profiles are found. Figure 7.2a shows the small part of the trajectory where the roll thrusters are active, which is up until approximately 240 seconds. The jump that occurs around $t \approx 165$ s, is due to the fact that the ailerons are activated. In Figure 7.2b it can be seen that the the roll jets are deactivated around $t \approx 240$ s by the jump that is present in the gains. As lateral motion consists of the coupling of roll and yaw movement, also the activation of the rudders can be spotted, which is shortly after the activation of the ailerons themselves, an instant before $t = 200$ s. Then, Figure 7.3a shows that the yaw jets are used at all times, however, as the dynamic pressure increases and the RCS becomes more and more inefficient, the gains decrease. The

activation of ailerons and rudders (two jumps around $t \approx 165$ s and $t \approx 180$ s) and the deactivation of the roll jets (the irregularity at $t \approx 240$ s) can be seen in the gain profile for the yaw jets. Finally, the rudder gains are defined in Figure 7.3b, where the aforementioned actuator activations and deactivations can again be seen. These gains are, as was done for pitch control, tabulated with dynamic pressure as breakpoints, after which they can be fed to the controller in the simulator.

7.4 Controller Verification and Integration

As both controllers are designed at this point, they can be integrated into the simulator where they will function as one main controller. However, before that can be done it has to be verified whether the gain profiles that were computed earlier in this chapter will actually manage to control the vehicle. This is done by applying step and ramp responses at specific points in the trajectory and evaluate the result that is provided by the controller. For the step responses of the individual controllers two points will be defined along the trajectory, one where only RCS control is available and one where only aerodynamic control is evaluated. For the ramp responses the ramp from the angle of attack profile will be used, which was found in the control history that was shown in Figure 6.8. For the ramp response of the lateral controller the bank reversal at $t \approx 900$ s is chosen, and it is evaluated whether the controller is able to handle two bank reversals in a relatively short time.

For the step responses, two points in time were chosen to evaluate the step responses of both the longitudinal and lateral controller. As the first aerodynamic actuators are activated for $\bar{q} = 150$ N/m^2, it was decided to choose time point 31, where $t = 120$ s and $\bar{q} = 21.9$ N/m^2, as this point lies roughly in the middle of the RCS phase. Then, for aerodynamic control a point was chosen where no change in control variables occurs, but where both control variables are non-zero. This has lead to point 226, where $t = 900$ s and $\bar{q} = 3.29$ kN/m^2. Note that the chosen points are quite arbitrary, with exception of the reasoning that was shown here, however, they are solely used for verification of the controller. When integrated, the controller should be able to follow the guidance profile during the whole trajectory.

In this section the verification process of both the longitudinal and lateral controllers is discussed, after which they are combined into an integrated control system. This final system should be able to use the computed gains to calculate control commands that are able to follow the predefined guidance profile, that is, the control system should be able to keep the true attitude of the vehicle within the predefined error bounds that were used to compute the gains (**Q**-matrix) without exceeding the maximum and minimum actuator values, which were found in **R**. For the bank reversals a different requirement is set, which is that the system should be able to perform the bank reversals with an overshoot that remains within the predefined error bounds and a settling time that is sufficiently short for the vehicle to have the desired attitude before the next bank reversal is performed. The value of this settling time depends on the part of the trajectory, as is seen in Figure 6.8.

Longitudinal Controller

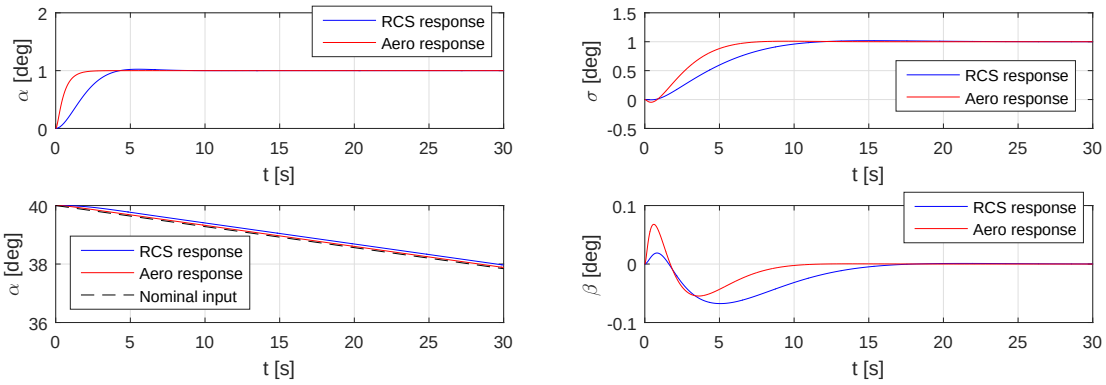
For the longitudinal controller, the ramp response that was used corresponded to the ramp that is present from $t \approx 920$, as can be seen in Figure 6.8. The slope of this part of the angle-of-attack profile is -0.072 deg/s. Figure 7.4a shows the longitudinal controller responses for a step input of $\Delta\alpha = 1^\circ$ and the angle-of-attack ramp input that was described earlier in this section. In this figure it can be seen that the controlled attitude error remains within the bounds of $\Delta\alpha_{max} = 1^\circ$ for RCS control and $\Delta\alpha_{max} = 2^\circ$ for aerodynamic control.

Lateral Controller

In terms of the lateral controller, the step responses for $\Delta\sigma = 1^\circ$ are shown in Figure 7.4b, and also for this test, it is seen that the controlled attitude errors remain within the bounds of $\Delta\beta_{max} = 1^\circ$ and $\Delta\sigma_{max} = 4^\circ$ for RCS control and $\Delta\beta_{max} = 4^\circ$ and $\Delta\sigma_{max} = 10^\circ$ for aerodynamic control. However, as the lateral controller difficulties are located during the bank reversals, the behavior of the controller during these maneuvers is more interesting than the step responses.

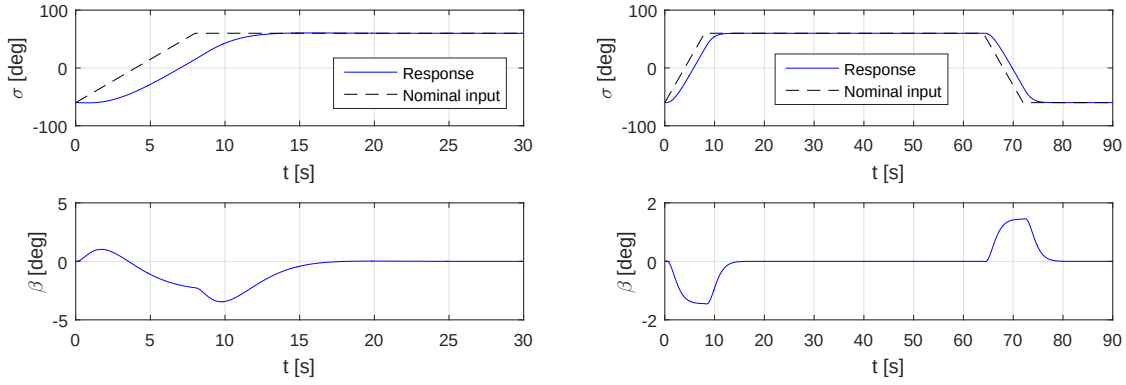
In Figure 6.8, it was seen that multiple bank reversals are performed during the entry flight, and in this control history, these reversals possess an infinite bank angle rate. To ensure that the controller is verified on realistic bank angle rates, the time span of the bank reversals that were defined in the control history is extended up until bank angle rates of $\dot{\sigma} = 10^\circ - 15^\circ$, which may resemble a realistic bank angle rate.

In Figure 7.5a the response of the controller is shown for the bank reversal that occurs at $t \approx 900$, where it is seen that the overshoot is negligible. The induced angle of sideslip remains within the error bounds of $\Delta\beta_{max} = 4^\circ$. Then, in Figure 7.5b two bank reversal responses are evaluated, which were deduced from the original control history. This test is done to evaluate whether the controller is able to keep the settling time



(a) (b)

Figure 7.4: (**a**) - Step and ramp responses of the longitudinal controller for aerodynamic and RCS control. (**b**) - Step responses of the lateral controller for aerodynamic and RCS control.



(a) (b)

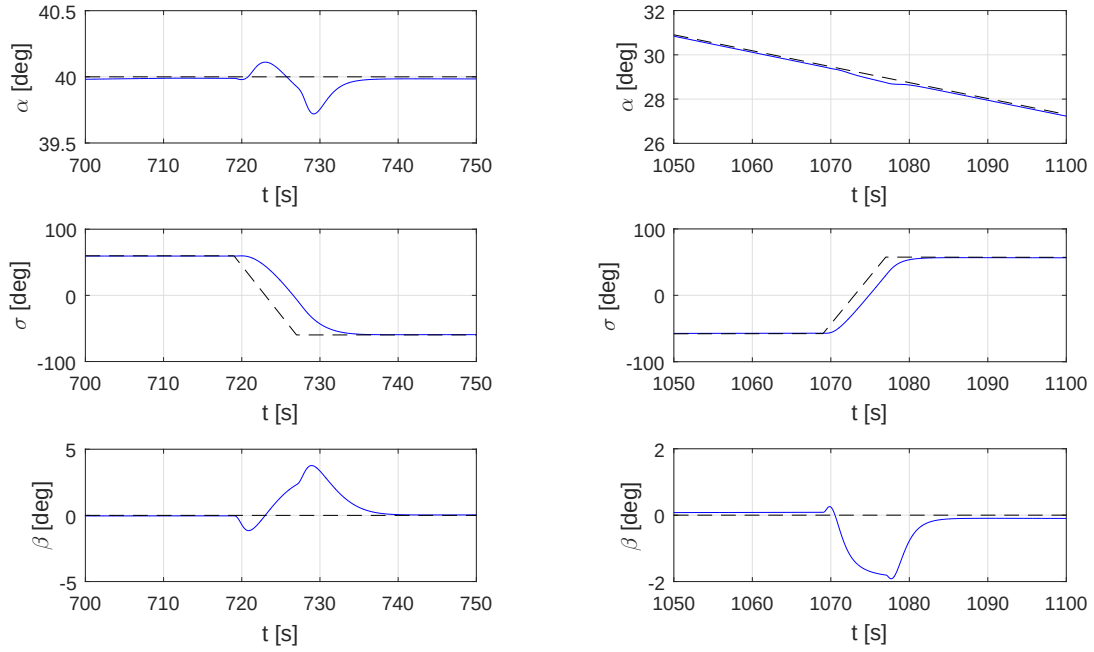
Figure 7.5: (**a**) - Ramp response for the bank reversal that is performed at $t \approx 900$ s. (**b**) - Ramp response for two subsequent bank reversals at the end of the trajectory.

lower than the time between two bank reversals. In this figure it can be seen that the settling time is very small, as the controlled bank angle is damped very well. It is also found that the bank-angle error and the induced angle of sideslip are smaller for a later phase of the trajectory, which can be explained by the fact that the dynamic pressure is higher in this part of the flight, hence the aerodynamic actuators are more efficient.

Integrated Controller

Earlier in this section, the verification of the individual controllers was shown, using a step and ramp response for longitudinal control, and the ramp responses corresponding to the bank reversals for the lateral controller. After the unit verifications were done, the control system was integrated into the simulator, using the true state as input, as no navigation system was available yet. The dynamic pressure was then used for the computation of the LQR gains, which were, together with the difference between the commanded and true attitude, used to compute the actuator commands. After integration, system verification of the controller was done by performing the bank reversals from the control history, as well as the angle-of-attack ramp.

In Figure 7.6a the first bank reversal is evaluated, and it can be seen that the bank angle shows similar behavior as was seen in the verification of the lateral controller. The induced angle of sideslip also has a similar profile, and remains within the predefined error bounds that were described earlier in this section. The main difference between the individual controllers and the integrated system is that a coupling exists between symmetric and asymmetric motion, as an induced angle of attack is seen during the bank reversal. However, as was stated earlier, this perturbation is very small, which validates the decision to decouple the system during the design. Figure 7.6b shows the second bank reversal, in which similar behavior is seen as



(a) (b)

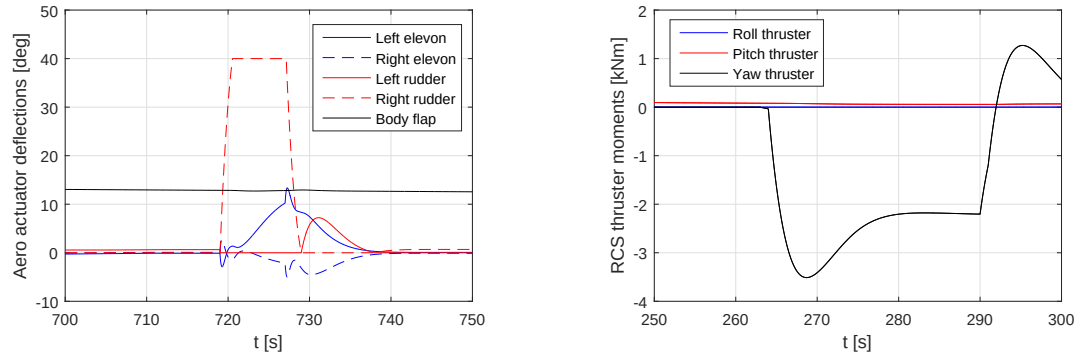
Figure 7.6: (**a**) - Controller results for bank reversal 1 at $t \approx 720$ s. (**b**) - Controller results for bank reversal 2 at $t \approx 1069$ s.

for the first, however, due to the increase in effectiveness of the aerodynamic actuators (from the increased dynamic pressure) the errors between the commanded attitude angles and the true attitude are smaller than for the first bank reversal. Finally, it is seen that the induced angle of attack is corrected quickly after the bank reversal is performed, both for a step and a ramp profile of this angle.

The bank-angle error during the first bank reversal contained a value of approximately 45°, but is recovered within a few seconds. It was discussed that no actual guidance system was used in the simulator, which poses some limitations on the tests that were done and some difficulties with respect to the conclusions that were drawn. It must be noted that the results from this section only provide information concerning the expected behavior of the control system, and its response to certain inputs. In Mooij (1997) similar research was done, however, the simulator that was described there contained a guidance system, and it was seen that the bank-angle error during the first bank reversal still contained a value of roughly 25°. The application of this guidance system explains the difference in the simulated bank-angle errors.

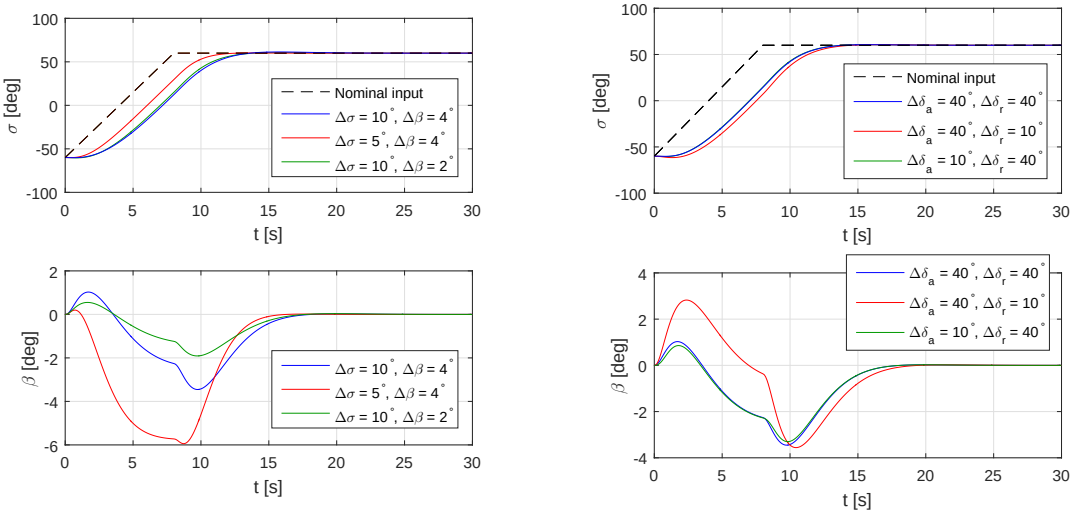
To evaluate the behavior of the actuators, Figure 7.7a and Figure 7.7b are shown. It is chosen to evaluate the first bank reversal for the aerodynamic actuators, as all surfaces are active. The left and right elevons are denoted by the blue lines, and they show asymmetric behavior, as they are a combination of elevator and aileron commands. The body flap is still only used in the trim module, and the left and right rudders are active due to the bank reversal. It is seen that during this maneuver, the right rudder reaches the maximum value, thereby limiting the effectiveness of the control system. During this maneuver, not all thrusters are active, and therefore it is chosen to evaluate the guidance activation at $t \approx 265$ s for the RCS actuators. It is seen that mostly the yaw thrusters are active during this maneuver, as at this point in the trajectory, the aerodynamic surfaces are very inefficient. The roll jet gains are zero, as was seen in Figure 7.2a, and therefore, the roll thrusters are deactivated already.

Finally, the influence of the **Q** and **R** matrices are evaluated, which contain the allowed maximum attitude angle and rotational rate values and the maximum deflections and moments of the actuators respectively. For doing this, the first bank reversal is evaluated for the lateral controller, where in Figure 7.8a the maximum errors for the bank angle and the angle of sideslip are varied. It is seen that for the bank-angle behavior, reducing the maximum bank-angle error has a positive effect, as the controlled bank-angle error reduces. Reducing $\Delta\beta_{max}$ has a negligible influence on the performance of the controller on the bank angle. However,



(a) (b)

Figure 7.7: (**a**) - Aerodynamic actuator results for bank reversal 1 at $t \approx 720$ s. (**b**) - RCS actuator results for guidance activation at $t \approx 265$ s.



(a) (b)

Figure 7.8: (**a**) - Influence of the allowed maximum attitude errors. (**b**) - Influence of the maximum actuator commands.

when the angle-of-sideslip behavior is evaluated, it is seen that the better performance for the small allowed bank-angle error is coupled to a larger induced angle of sideslip, hence, a worse performance for this variable. Lowering $\Delta\beta_{max}$ shows a smaller induced angle of sideslip.

In Figure 7.8b the influence of the maximum actuator commands is shown. It shows that variation in maximum aileron angle has negligible influence, which is explained by Figure 7.7a, as the elevon deflections are far from their maximum during the entry flight. The rudders however, showed maximum deflection during bank reversals, and therefore it can be seen that variation of the maximum allowable rudder deflection influences the performance of the controller. When lowering this value, a worse performance is found in terms of bank-angle control and induced angle of sideslip.

Finally, it is concluded that the performance of the controller can be enhanced by lowering the maximum allowed attitude angles and rates, however, due to this, the controller is 'forced' to control the attitude more swiftly, hence, larger actuator deflections and thruster moments have to be generated. This is seen as a disadvantage, as the ability of the vehicle to perform a sudden maneuver is restricted when the actuators are either fully deflected or not deflected at all. As was stressed earlier, the absence of a real guidance system limits the conclusions that can be drawn from the results that were stated in this chapter. However, despite this, the integrated controller met the requirements that were defined earlier in this chapter.

8

Implementation and Verification of the Navigation Module

In Chapter 5 the theory of the IMU, GPS and FADS sensors was discussed, as well as the EKF, UKF and PF estimators. Using these theories, all six components were implemented to perform the research that was described at the start of this report. Section 8.1 discusses the implementation and verification of the three sensors that were introduced in Section 5.1.1, together with the generated measurements that serve as an input to the estimators. These three mathematical models were also implemented and verified, which is described in Section 8.2.

8.1 Sensors

This section contains the implementation and verification of the three sensors that were described in Section 5.1.1, namely the IMU, GPS and FADS. For every sensor the theory from Section 5.1.1 will be used to find the specific implementation for the simulator. Then, using various methods, verification of the generated measurements from each sensor will be described, after which some related results are shown and discussed.

8.1.1 IMU

The first sensor that was described was the IMU, consisting of an accelerometer for translational motion and a gyroscope for rotational motion. These measurements are used for the state propagation that is done in the module. In this subsection, first the implementation of the IMU sensor in the simulator will be described, after which its measurements will be verified. Next to that, the difference between Euler integration and RK4 integration will be evaluated and the effect of the propagation frequency on the internal measurements of the vehicle will be discussed.

Implementation

Section 5.1.1 contained the general theory of the IMU, together with the measurement equations and error sources. The difference between the simulator and a real flight is that for the simulator the measurements have to be generated using the observation equations, whereas in real life measurements are delivered by the sensor. To ensure that the IMU observations are as realistic as possible, it was chosen to imply several error sources that were described in Section 5.1.1, namely scaling factors, biases and misalignments. As this sensor is used for nominal state propagation, a high operating frequency is favorable, as the integration error will remain small, which is in line with research done by Mooij and Chu (2002) and Gross et al. (2010). The state propagation then is performed using the measured accelerations and rotational rates and the EoM. Then, at the points in time where other measurements are available, a measurement update of the state is done using one of the estimators that were described in Section 5.1.2. During this measurement update, an estimation of the IMU error sources can also be made, as will be explained in this subsection.

The implementation of the IMU is rather straightforward, as it only depends on variables that can be retrieved in the simulator and some predefined constants. To generate the measurements it will be used that

Table 8.1: IMU error sources and their used values, retrieved from Mooij and Chu (2002).

Accelerometer		Gyroscope	
s_x	$2 \cdot 10^{-4}$	s_p	$2 \cdot 10^{-4}$
s_y	$-1.7 \cdot 10^{-4}$	s_q	$-1.7 \cdot 10^{-4}$
s_z	$2.3 \cdot 10^{-4}$	s_r	$2.3 \cdot 10^{-4}$
m_{xy}	10^{-6}	m_{pq}	10^{-6}
m_{xz}	-10^{-6}	m_{pr}	-10^{-6}
m_{yx}	-10^{-6}	m_{qp}	-10^{-6}
m_{yz}	$2 \cdot 10^{-6}$	m_{qr}	$2 \cdot 10^{-6}$
m_{zx}	10^{-6}	m_{rp}	10^{-6}
m_{zy}	$2 \cdot 10^{-6}$	m_{rq}	$2 \cdot 10^{-6}$
b_x	$3 \cdot 10^{-4}$	b_p	$4 \cdot 10^{-6}$
b_y	$-3.5 \cdot 10^{-4}$	b_q	$-5 \cdot 10^{-6}$
b_z	$3 \cdot 10^{-4}$	b_r	$3 \cdot 10^{-6}$

$$\mathbf{a}_m = (\mathbf{I} + \mathbf{S}_a)\,\mathbf{a} + \mathbf{b}_a + \boldsymbol{\nu}_a \tag{8.1a}$$

$$\boldsymbol{\omega}_m = (\mathbf{I} + \mathbf{S}_\omega)\,\boldsymbol{\omega} + \mathbf{b}_\omega + \boldsymbol{\nu}_\omega \tag{8.1b}$$

which were evaluated in Section 5.1.1. Scaling factors and misalignments are predefined and inserted into matrices **S**, whereas the biases have a preset value in vectors **b**. The vectors **a** and $\boldsymbol{\omega}$ are retrieved from the output of the flight dynamics block of the simulator and represent the actual acceleration and rotational rate of the vehicle. The white noise additions are generated by

$$\boldsymbol{\nu_a} = \text{randn} \cdot \sigma_a, \boldsymbol{\nu_\omega} = \text{randn} \cdot \sigma_\omega \tag{8.2}$$

and represent Gaussian white noise. Note that all random generators that are used are seeded, which means that their generated random numbers can be repeated for different runs. Using this seed, results will be reproducable despite their dependence on random numbers. The noise standard deviations that are used are $\sigma_a = 10^{-5}$ m/s^2 and $\sigma_\omega = 10^{-6}$ rad/s. The chosen IMU errors are summarized in Table 8.1. The generated measurements can directly be integrated into the EoM to obtain nominal state propagation. As this propagation only depends on the integration of the EoM, the type of integrator and the operating frequency of the IMU itself are of importance for the accuracy of the propagation results. A lower operating frequency will lead to a larger integration error for a particular step, and, as this error is cumulative, the dependence on the frequency is large for the final error. In Mooij and Chu (2002) the influence of the operational frequency of the IMU is evaluated, with values varying between 25 Hz and 200 Hz. Then, in Smith (2011) and Zhou et al. (2010), frequencies of 100 Hz and 50 Hz are used respectively, applied in similar applications. The operational frequency that is used in the navigation module in the simulator is then chosen after evaluation of the effect of this variable on the nominal state propagation. Next to that, for all frequencies, the difference between the aforementioned Euler and RK4 integration methods is handled, which is described further on in this section.

IMU propagation

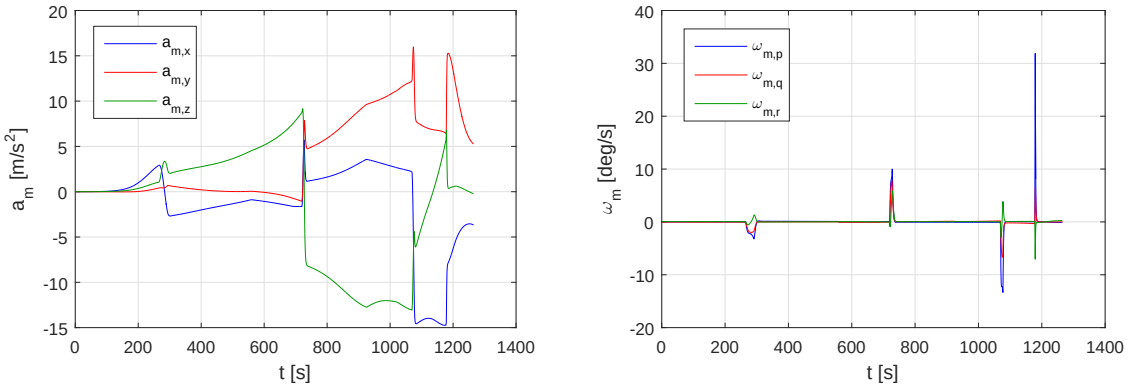
As described, the state estimation process is done by nominal state propagation using the IMU measurements, completed by a measurement update using the GPS/FADS combination. The propagation is based on the EoM expressed in the inertial frame, as the simulator output is also expressed in this frame, and uses:

$$\ddot{x} = a_{I,x_e} + g_x \tag{8.3}$$

$$\ddot{y} = a_{I,y_e} + g_y \tag{8.4}$$

$$\ddot{z} = a_{I,z_e} + g_z \tag{8.5}$$

These are the differential equations for the velocity and it can be seen that they are defined as the total acceleration on the vehicle. As the IMU measures the non-gravitational acceleration, one has to add the g terms to this equation. The non-gravitational accelerations here are estimated using the actual measurements and the estimated error sources, according to (Mooij and Chu, 2002)



(a) (b)

Figure 8.1: Accelerometer and gyroscope measurements over the controlled re-entry flight.

$$\begin{pmatrix} a_{I,x_e} \\ a_{I,y_e} \\ a_{I,z_e} \end{pmatrix} = \mathbf{C_{I,B}} \begin{bmatrix} 1+s_x & m_{xy} & m_{xz} \\ m_{yx} & 1+s_y & m_{yz} \\ m_{zx} & m_{zy} & 1+s_z \end{bmatrix}^{-1} \left\{ \begin{pmatrix} a_{x_m} \\ a_{y_m} \\ a_{z_m} \end{pmatrix} - \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix} \right\} \tag{8.6}$$

where the transformation matrix $\mathbf{C_{I,B}}$ translates the estimated accelerations in the body frame to the estimated accelerations in the inertial frame and is computed using quaternions:

$$\mathbf{C_{I,B}} = \begin{bmatrix} Q_1^2+Q_4^2-Q_2^2-Q_3^2 & 2(Q_1Q_2+Q_3Q_4) & 2(Q_1Q_3-Q_2Q_4) \\ 2(Q_1Q_2-Q_3Q_4) & Q_2^2+Q_4^2-Q_1^2-Q_3^2 & 2(Q_2Q_3+Q_1Q_4) \\ 2(Q_1Q_3+Q_2Q_4) & 2(Q_2Q_3-Q_1Q_4) & Q_3^2+Q_4^2-Q_1^2-Q_2^2 \end{bmatrix} \tag{8.7}$$

The derivatives of the position are simply the velocity components over the respective axes. The use of quaternions is chosen as they rule out any possibility of singularities, but also as they simplify the EoM that are used. As the attitude is part of the state, the derivatives of the quaternions are defined as

$$\dot{\mathbf{Q}}_{I,B} = \frac{1}{2} \begin{bmatrix} -Q_4 & -Q_3 & Q_2 \\ Q_3 & -Q_4 & -Q_1 \\ -Q_2 & Q_1 & -Q_4 \\ Q_1 & Q_2 & Q_3 \end{bmatrix} \begin{pmatrix} p_e \\ q_e \\ r_e \end{pmatrix} \tag{8.8}$$

where the vector on the right-hand side of the equation contains the estimated rotational rates, which are defined by

$$\begin{pmatrix} p_e \\ q_e \\ r_e \end{pmatrix} = \begin{bmatrix} 1+s_p & m_{pq} & m_{pr} \\ m_{qp} & 1+s_q & m_{qr} \\ m_{rp} & m_{rq} & 1+s_r \end{bmatrix}^{-1} \left\{ \begin{pmatrix} p_m \\ q_m \\ r_m \end{pmatrix} - \begin{pmatrix} b_p \\ b_q \\ b_r \end{pmatrix} \right\} \tag{8.9}$$

where p_m, q_m and r_m are the measured rotational rates. As was stated earlier in this report, the estimator will be used to approximate the position, velocity and attitude, but also several error sources, of which the biases are chosen. Firstly, the GPS signals are discussed, where the clock bias and drift are of importance:

$$\dot{b}_c = d_c, \dot{d}_c = 0 \tag{8.10}$$

This equation shows that the derivative of the clock bias is the clock drift, which is seen in the measurement equation for the range rate. The clock drift is assumed to be constant. For the IMU, first the measurements are evaluated, and in Figure 8.1a the measured accelerations are shown, which are expressed in the body frame. Due to the change in attitude during flight, accelerations over all body axes are experienced.

The biases possess small values, as was seen in Table 8.1. Due to this, they can only be estimated when the measured acceleration is also small so that the biases have a relatively large influence on the accelerometer measurements. From Figure 8.1a it can be seen that this is the case for the first phase of flight. The scaling factors then are best estimated for high accelerations, which is due to the fact that their influence increases

for increasing true acceleration. The misalignments are not observable, as is stated by Mooij (2015), however, they are estimated during ground calibration. For the gyroscope, of which the measurements are shown in Figure 8.1b, it is seen that only during guidance activation, the bank reversals, and guidance deactivation, large rotational rates are experienced. Also for this component of the IMU it holds that the biases (which are known as gyroscope drift) can be estimated for small rotational rates, and scaling factors are estimated for high rotational rates.

For the estimation process, it is chosen to estimate the accelerometer biases, together with the x-scaling factor. After evaluation of all scaling factors of the accelerometer, it is chosen to use the ground calibrated values for the y- and z-axes, as their estimations did not fully converge during the final part of the trajectory. For the x-acceleration this is also the case, however, it is kept in the estimation process to show its behavior, which is similar as the behavior of the y- and z-accelerations. For the gyroscope, it was seen that only for a few short moments, the rotational rates were sufficiently large for the scaling factors to be estimated accurately, and therefore, it is chosen to use the ground calibrated values. The gyroscope drift, however, can be approximated very well and is included in the estimation process. In the state propagation, these errors are seen as constants, hence:

$$\dot{s}_x = \dot{b}_x = \dot{b}_y = \dot{b}_z = \dot{b}_p = \dot{b}_q = \dot{b}_r = 0 \tag{8.11}$$

The IMU propagation is done by

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{x}_i \Delta t \tag{8.12}$$

where

$$\mathbf{x} = \left(x, y, z, v_x, v_y, v_z, Q_1, Q_2, Q_3, Q_4, b_c, d_c, s_x, b_x, b_y, b_z, b_p, b_q, b_r\right)^T \tag{8.13}$$

and is the state vector that is chosen for the estimation process. As was described earlier, in the time steps where no GPS or FADS measurements are available, the state estimation is performed by the IMU propagation, which is defined by the equations stated above.

Simulation and Verification

The verification process of the IMU measurements was simple, as they are generated using the actual accelerations and rotational rates of the vehicle. These variables are retrieved directly from the flight dynamics block and contain the exact values. When the observation equations of the accelerometer and the gyroscope from Equation (8.1) are evaluated, and all sensor errors and noise components are set to zero, the sensor output should be equal to the actual acceleration and rotational rates. This was tested for the implemented sensor, and by gaining the correct results and hand checking the application of the error sources, the IMU measurements were verified.

It was described that in the state estimation process, the IMU measurements were propagated, after which a measurement update was done using an estimator. The nominal state propagation was performed by integrating the EoM at a specific frequency. Note that this frequency does not necessarily have to be equal to the operating frequency of any of the sensors. In Section 6.2 two integration methods were described for the state propagation, which were Euler integration and RK4 integration. It was stated that in terms of computational load, Euler integration is the most efficient method that can be used, however, in terms of accuracy, it usually has a poor performance.

In Figure 8.2 the results are shown for state propagation using only an initial state and the IMU measurements. In this figure, a comparison is made for the two aforementioned integration methods and different operating frequencies are evaluated. In terms of position, it can be seen that frequencies of 100 Hz and higher result in a relatively small integration error over the hypersonic descent phase of the re-entry flight. It is also seen that the RK4 method contains a higher accuracy, however, for higher frequencies, the difference between the two integration methods becomes smaller in terms of accuracy, while the Euler integration technique induces a lower computational load. For the velocity, similar results can be seen, where again the 25 Hz case provides a relatively inaccurate outcome compared to the higher frequencies. Next to that, the difference between both integration methods is small for 100 Hz and 200 Hz.

To conclude the evaluation of the nominal state propagation performance, it is chosen to use Euler integration in the navigation module in the simulator, in combination with an IMU operating frequency of 200 Hz. During evaluation, it was found that the implementation of a high frequency in combination with a low-accuracy integration method resulted in a higher estimation accuracy and lower computational time than

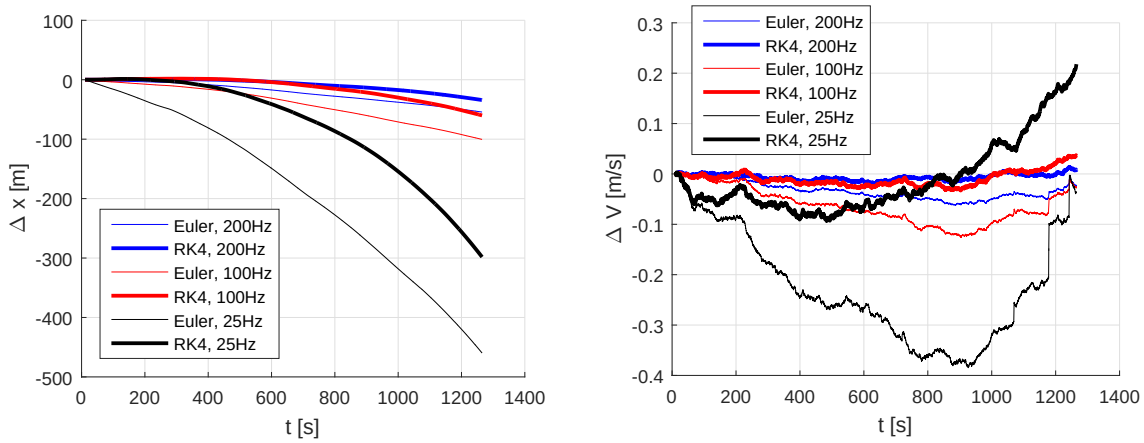


Figure 8.2: Position and velocity results of state propagation using only IMU measurements for operating frequencies of 25 Hz, 100 Hz and 200 Hz and for two different integration methods.

the combination of a low operating frequency and a more accurate integration method. The propagation of the attitude angles was not shown in this section, as the errors that are built up during the estimation process are included in the position and velocity that were presented in Figure 8.2.

8.1.2 GPS

In this subsection the discussion on the implementation of the sensors that are used in the simulator will be continued, and the measurements that are obtained are reviewed as well. Previously, it was seen the application of an IMU to the uncontrolled re-entry flight, in which it was seen that the state of the vehicle can be approximated using only an integrator in combination with IMU measurements. However, the accuracy of this application is not acceptable when it comes to navigation for a re-entry vehicle, hence the integration of an estimator, which is described further on in this chapter. The coupling of IMU with the estimator (which uses measurements from other sensors) can be done in a so-called loosely-coupled system and a tightly-coupled system. Mooij and Chu (2002) describe that the tightly-coupled system has a better performance, as it uses raw measurements from the GPS receiver, such as pseudorange, range rate and carrier phase observations. Therefore it is chosen to use a tightly-coupled INS/GPS system, which will be described in Section 8.2. This subsection is again divided into an implementation part and a verification part, only now concerning GPS instead of IMU.

Implementation

For the GPS receiver, it was chosen to generate the measurements manually instead of using real-time GPS ephemeris, as the focus of this thesis lies on the implementation of the FADS sensor in the navigation module. The generation of these measurements can be done accurately and thus does not pose restrictions on the performance of the final navigation module. It is also done in Mooij and Chu (2002), however, a slightly different approach is used in this thesis. First the orbits of the GPS satellites are to be generated, to determine their position and velocity at the time that is specified in the simulation. As research is done on the system in general it does not matter at which point in time the simulator is set, however, as an initial point of the GPS orbits is specified, the time of execution of the simulator should not differ too much from this point, as the error accumulation would become significant. This would have no influence on the research that is done, only the synchronization of reality and simulation will be off. In Tapley et al. (2004) the orbital elements for 28 GPS satellites on the midnight epoch of January 1st, 2000, or: 1 January 2000, 00 : 00 : 00.000, are found. In Section 3.2 only the Cartesian and spherical systems were described, so a short definition of the orbital elements is provided.

The orbital elements consist of six components that together are used to describe position and velocity of an object that is in orbit around a central body, which in this case is the Earth.

The left side of Figure 8.3 shows the satellite position in two dimensions, with the semi major axis a, the eccentricity e and true anomaly θ. Then, on the right side of this figure the inclination i, argument of pericenter ω and right ascension of ascending node Ω can be found, which conclude the six components of the orbital elements (Mooij, 2015). The initial GPS orbital elements are shown in Table 8.2. In this table six

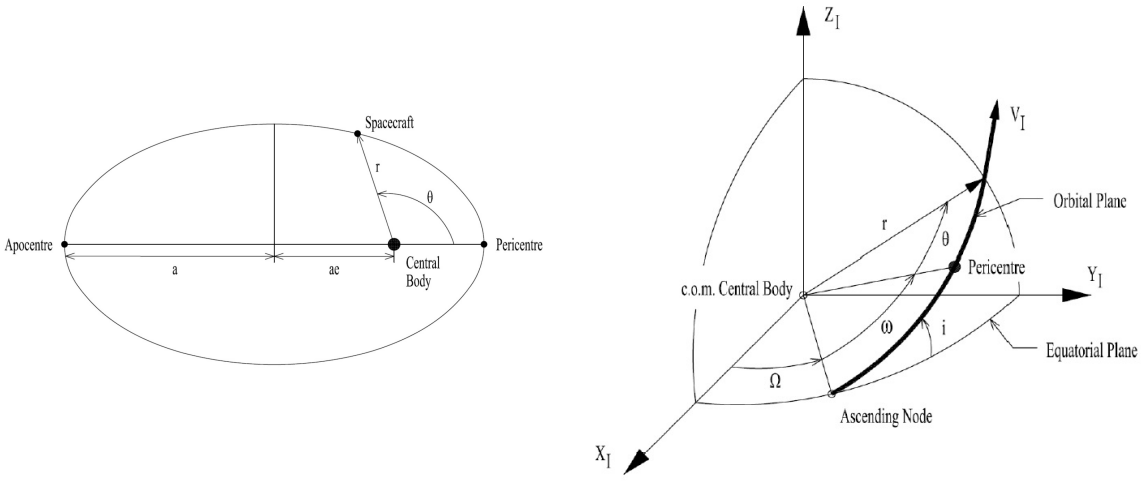


Figure 8.3: Left: two dimensional representation of an elliptic orbit including corresponding orbital parameters. Right: three dimensional representation of (a part of) an elliptic orbit including corresponding orbital parameters (Mooij, 2015).

orbital elements are shown per satellite, along with the identifier (second column) and the orbital plane and position (first column). From the first column it can be stated that GPS satellites orbit in six different orbital planes with either four or five active satellites in each plane.

However, before this conversion is done, first the mean anomaly of every satellite is propagated over time, so that the anomalies at the desired epoch are known. Then the conversion to Cartesian components is done so that the position and velocity of the vehicle and all GPS satellites in the inertial frame are possessed. Subsequently the real range is computed by subtracting the position matrices from satellite i and the vehicle. This value is divided by the speed of light to compute the time that is required for the signal to travel from the GPS satellite to the receiver. In this small time step both the vehicle and the GPS satellites change their position and velocity, and therefore this is added to the computation of the pseudorange. Then, after the measurements are generated using the equations from Section 5.1.1, noise components were added in a similar manner as was done for the IMU. Crassidis (2005) defines a pseudorange white noise with $\sigma_\rho = 5$ m, the range rate and carrier phase noise components are retrieved from Mooij and Chu (2002), which define $\sigma_{\dot{\rho}} = 0.1$ m/s and $\sigma_{\Delta\Phi} = 0.026$ cycles, which corresponds to approximately 5 mm. The GPS signal frequency that is used is 1575.42 MHz, and the input clock errors are $b_c = 5$ m and $d_c = 0.1$ m/s. The described GPS receiver operating frequency is equal to 10 samples per second, *i.e.*, 10 Hz (Crassidis, 2005 and Mooij and Chu, 2002).

Finally, the last aspect of implementation of the GPS receiver is the part of the entry flight that is called the blackout phase. During this phase, the heat that is generated due to the high velocity of the vehicle, ionized the atmospheric particles located around the vehicle, thereby basically preventing the reception of GPS signals. In the absence of GPS measurements the navigation system will shut down the measurement update of the used estimator and base its position, velocity and attitude estimation solely on IMU measurement propagation. Goodman (2008) describes different flights of the Space Shuttle, for which the GPS outage was measured. The blackout phase for the STS-118 was defined from 268000 ft to 138600 ft, which corresponds to 81686.4 m and 42245.3 m, obtained by post-processing the flight results. For convenience, these values are rounded to the nearest kilometer, by which it is defined that the blackout phase of the vehicle ranges from 82000 m to 42000 m.

Simulation and Verification

At this point, measurements were generated for every GPS satellite, however, only a certain number of them are visible from the vehicle's point of view. Determination of the visibility of a satellite is done using an elevation angle, of which the influence is evaluated further on in this subsection. This means that when the line-of-sight of satellite i makes an angle of more than x degrees with the plane that is perpendicular to the distance vector of the vehicle, the satellite is deemed visible:

$$\epsilon = \text{acos}\left(\frac{\mathbf{r}\cdot\mathbf{r}_{\text{LOS,i}}}{|\mathbf{r}||\mathbf{r}_{\text{LOS,i}}|}\right) \tag{8.14}$$

Table 8.2: GPS satellite constellation at epoch 1 January 2000, 00 : 00 : 00.000, expressed in orbital elements (Tapley et al., 2004)

Orb. Plane & Position	**PRN**	a [km]	e [-]	i [deg]	Ω [deg]	ω [deg]	M [deg]
F-1	1	26568.4	0.0048	55.0	117.9	258.8	76.2
B-3	2	26561.3	0.0194	53.6	232.9	236.7	264.0
C-2	3	26561.2	0.0010	54.1	294.8	72.8	134.7
D-4	4	26562.0	0.0053	56.0	357.8	323.0	37.5
B-4	5	26562.0	0.0019	53.7	234.0	9.5	29.1
C-1	6	26558.9	0.0068	54.5	297.2	222.1	84.3
C-4	7	26559.5	0.0109	54.6	295.4	239.7	195.4
A-5	8	26539.7	0.0086	54.8	178.2	102.1	46.1
A-1	9	26559.3	0.0098	54.0	175.7	32.3	70.7
E-3	10	26557.3	0.0038	55.8	56.5	353.0	263.6
D-5	11	26559.3	0.0029	53.0	355.2	189.8	271.2
F-5	13	26558.7	0.0022	55.2	116.5	322.6	281.5
E-1	14	26562.4	0.0005	56.1	59.3	129.5	261.9
D-2	15	26555.5	0.0073	56.3	0.2	85.8	32.8
E-5	16	26562.3	0.0044	55.9	59.5	19.6	335.9
D-3	17	26558.9	0.0113	56.4	2.5	167.5	57.0
F-3	18	26559.2	0.0076	54.4	114.0	107.0	10.9
A-4	19	26560.0	0.0053	53.1	172.4	203.6	35.2
E-2	21	26559.9	0.0160	55.7	56.9	211.5	278.5
B-1	22	26559.9	0.0129	53.5	233.7	30.5	247.9
E-4	23	26562.3	0.0145	55.9	59.2	249.3	274.3
D-1	24	26561.1	0.0090	56.5	358.4	261.4	64.0
A-2	25	26561.7	0.0078	53.6	173.5	232.9	123.1
F-2	26	26562.3	0.0116	55.2	116.9	2.1	177.9
A-3	27	26559.9	0.0137	53.9	174.7	198.3	13.7
F-4	29	26558.6	0.0073	55.0	115.3	248.2	192.8
B-2	30	26562.8	0.0056	54.1	235.7	83.8	279.7
C-3	31	26561.7	0.0092	54.6	295.4	45.7	129.3

where $\mathbf{r}$ denotes the distance from the center of the Earth to the vehicle and $\mathbf{r}_{\text{LOS,i}}$ represents the line-of-sight from GPS satellite i to the vehicle.

To demonstrate the effect of the number of satellites on the GDOP, situations, where four, five and all satellites are used, are evaluated. When a subset of all visible satellites is taken, first the GDOP values are computed for every possible combination. Then, the constellation with the lowest GDOP is chosen, as this has the most effective spread of satellites across the sky. Figure 8.4a shows the effect of the visible number of satellites on the GDOP. It can be seen that the GDOP decreases when more satellites are used, which is expected. However, for the minimum number of satellites (four) the GDOP still has an acceptable value for performing the mission at all times. For randomly selected points in this figure, the GDOP values were computed by hand, which corresponded to the values that were found in the plots. Note that this only shows the influence of the number of satellites; the behavior of the GDOP may be different for different cases.

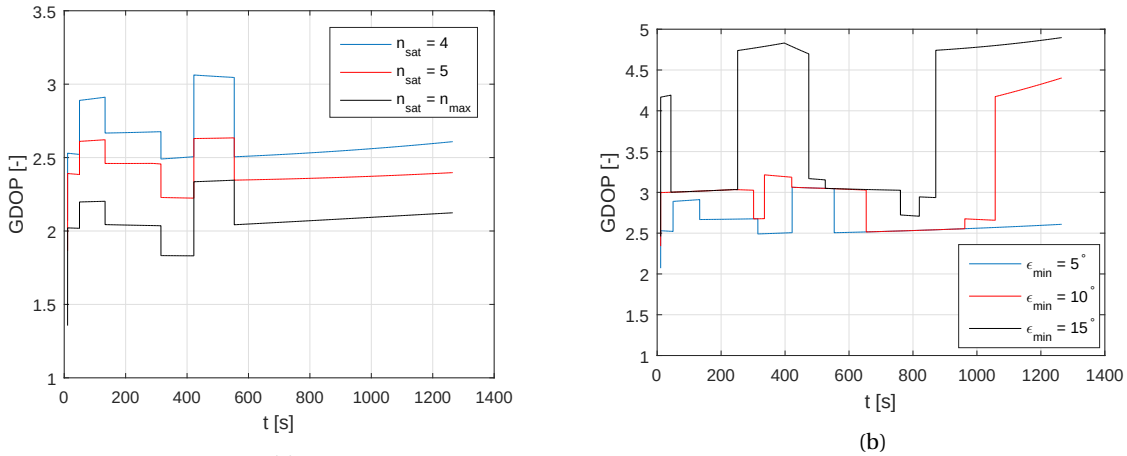


Figure 8.4: (**a**) - Comparison of Geometric Dilution of Precision for different numbers of satellites used. (**b**) - Comparison of Geometric Dilution of Precision for different minimum elevation angles.

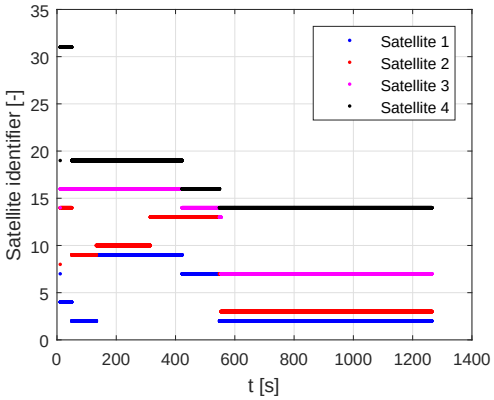


Figure 8.5: Visible satellites for the case where four satellites are used (satellite number corresponds to identifiers shown in Table 8.2).

Figure 8.4b then shows the comparison of the lowest GDOP for a constellation of four satellites as a function of the minimum elevation angle. It can be seen that when this angle is increased the average GDOP value also increases. This is due to the fact that satellites are only visible when they are higher above the horizon, which reduces the spread of the satellites across the sky (a satellite that induces a low GDOP may not be visible for higher minimum elevation angles) and therefore increases the GDOP value. During the simulation however a minimum elevation angle of 5 degrees is assumed. For lower elevation angles, signals from satellites that are located at the edge of the visible sky have to travel a larger distance through the atmosphere, which increases the influence of atmospheric effects on the measurements. Next to that, for very small positive values for ϵ_{min}, interference with other vehicle parts may be experienced. Therefore it is chosen not to decrease the minimum elevation angle even further. To verify the results from Figure 8.4b, random points in the trajectory are selected again, after which the GDOP values are recalculated by hand.

The visible satellites that produce the lowest GDOP when $n_{\text{sat}} = 4$ are shown in Figure 8.5, and it can be seen that the constellation that is used changes multiple times during a whole simulation. During the blackout phase, no external signals can reach the vehicle and thus no GPS is used, however, in this section it is assumed that the blackout phase does not exist. The results that are shown in this section namely are used for understanding of the concept of implementation of a GPS sensor to the vehicle.

Earlier in this paragraph it was stated that for all GPS satellites, the measurements were generated. Then, after the used satellites are selected, all measurements of the non-used ones are set to zero. This is done so that the measurement matrices that are used in the estimators do not change in size when a different number of GPS satellites is used. In this section observations for the minimum required visibility case are discussed,

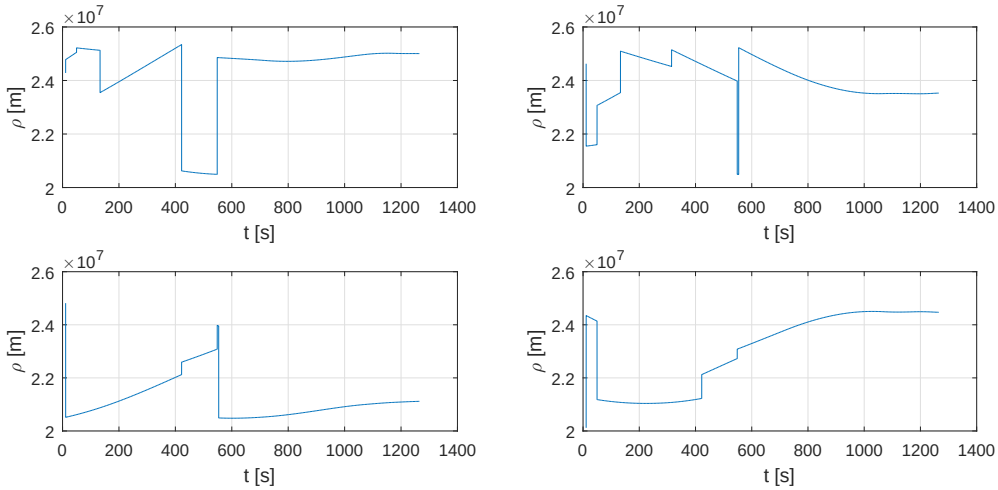


Figure 8.6: Pseudorange measurements for four visible GPS satellites.

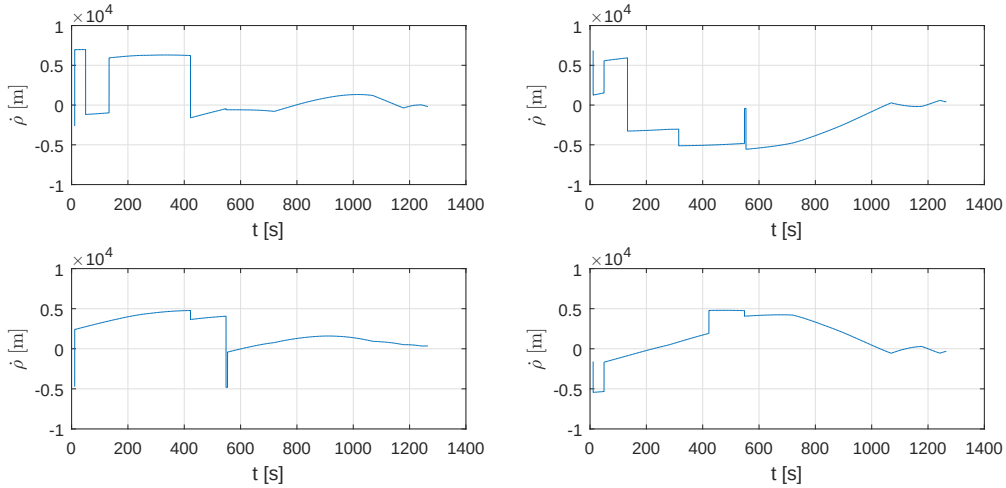


Figure 8.7: Range rate measurements for four visible GPS satellites.

where $n_{\text{sat}} = 4$. This is the minimum number of satellites that is required to obtain the position of the vehicle, as the fourth satellite is used for the time error. For the state estimation in Chapter 9 the worst-case scenario, which means that $n_{\text{sat}} = 4$, is used, to evaluate whether the estimators are able to approximate the state for the case that only four satellites can be used. However, to evaluate the influence of the number of satellites on the estimation accuracy, different values for n_{sat} will be used as well. In Figure 8.6 the pseudorange measurements are shown, whereas in Figure 8.7 the range rate observations are presented.

In both Figures 8.6 and 8.7 jumps can be seen in the measurement values, which are due to the fact that the satellite constellation changes. Furthermore, a peak can be seen around $t = 550$, which is formed by a change of constellation within 5 seconds that has the appearance of a peak due to the compression of the horizontal time axis. The range rate can take both negative and positive values as GPS satellites may move towards and away from the vehicle.

Finally, the carrier phase measurements are discussed, which concludes this paragraph. Figure 8.8 shows the carrier phase difference observations for one GPS satellite, and the jumps that were present in the pseudorange and range rate measurements are also visible in these plots. Normally, the carrier phase difference measurements contain a so-called integer ambiguity. The measurements denote the difference in phase of the signal at two antennas that are located on the vehicle, however, due to the cyclic behavior of the signal, this difference may include a number of whole cycles, which can not be measured. This unknown number of whole cycles is known as the integer ambiguity, and from Mooij and Chu (2002) it can be assumed that this

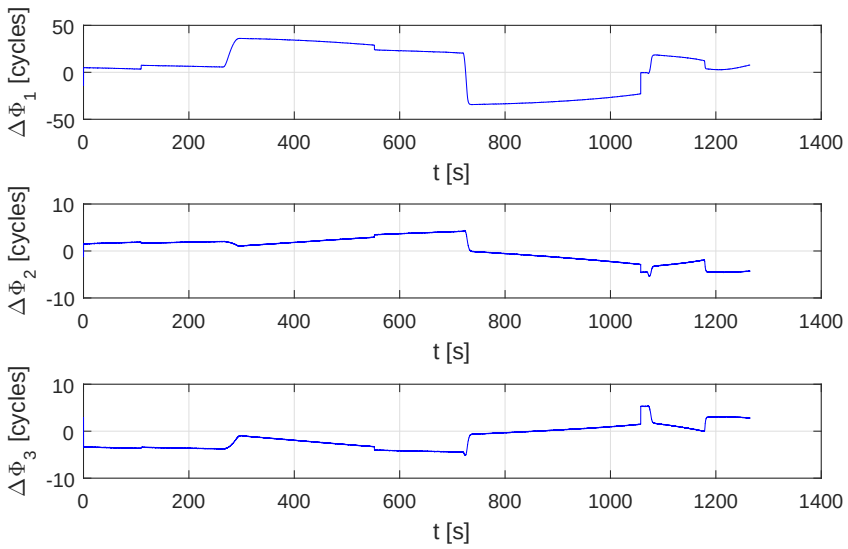


Figure 8.8: Carrier phase difference measurements for one of the visible GPS satellites.

variable can be solved correctly and can be discarded, leading to the measurements being able to go beyond the range of zero and one.

Results from Figures 8.6 to 8.8 show similar measurement values as in Mooij and Chu (2002), where the difference can be deduced to the fact that in this study, a different simulation time is used, hence a different GPS satellite constellation is found. In Figure 8.5 similar results as were shown in Mooij and Chu (2002) are found as well, again with the difference in the time stamp of the results. As the exactness of GPS signals is not crucial for this study, it was evaluated whether the pseudorange, range rate and carrier phase measurements are of the proper magnitude, which was verified by comparison with the measurements described in Mooij and Chu (2002).

For an exact resemblance with reality, one should use real time ephemeris data, as the method that was used during this thesis has one error source that increases over time. As the GPS measurements are based on computed GPS satellite positions, the error in these computations will eventually be added to the observation inaccuracies. The GPS satellite coordinates are computed by a transformation from Cartesian components to orbital elements, a propagation over time using these orbital elements and a transformation back to Cartesian components, *i.e.*, three possible truncation error sources are contained, which are all accumulated in the GPS position error. Therefore it is recommended that for real time simulation, that is, a simulation where the time at which the entry flight is performed is of importance, one should use online ephemeris data, as these data do not contain the above described errors.

The verification of the GPS measurements was limited, however, for the research that is done in this thesis, this does not pose any difficulties, as the focus of this thesis lies on the application of the FADS sensor and the implementation of multiple estimators. Next to that, the error that was introduced due to the aforementioned computation of GPS satellite states does not influence the outcome of the research as well, as it is the same for all estimators that are evaluated.

8.1.3 FADS

From studies such as Mooij and Chu (2002) it was proven that the combination of IMU and GPS measurements was sufficient for an acceptable state estimation. However, whether a FADS system is capable of increasing the accuracy of the estimation process, and what the contribution of this system is to a navigation module, is of interest for the research described in this report. In Section 5.1.1 this sensor was described and it was seen that in this sytem, estimation of several air data parameters was already done, namely the angle of attack, angle of sideslip, Mach number, air pressure and dynamic impact pressure.

As for the IMU and GPS sensors, this subsection will contain the implementation of the theory that was discussed in Section 5.1.1. For the FADS system, this consists of the definition of the sensor layout, the generation of the measurements and the built-in estimator. After the sensor was added to the simulator,

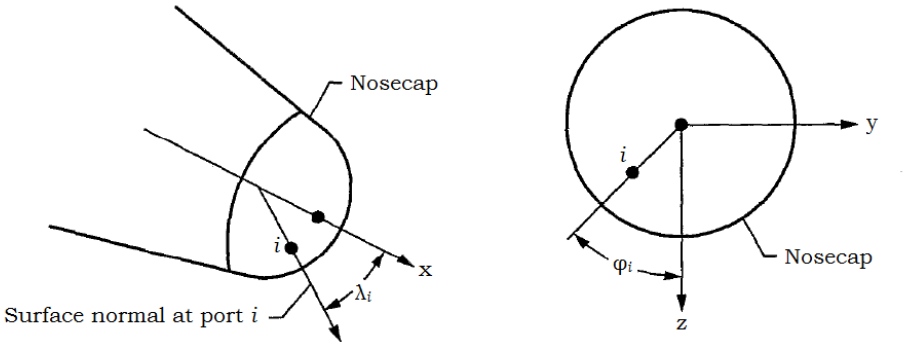


Figure 8.9: Definition of the clock angle λ and the cone angle ϕ that together describe the position of a pressure port of the FADS system (Cobleigh et al., 1999).

measurements were generated and a comparison was done between the estimated air data parameters from the system itself and the results of an estimator.

Implementation

In Section 5.1.1 it was described that the estimation of the air-data parameters in the FADS was done using the so-called triples algorithm. This algorithm was described in Section 5.1.1 and is defined as such, that the measured air-pressure differences of a combination of three ports is sufficient to compute the angle of attack and the angle of sideslip. Using these aerodynamic angles, the incidence angle θ could be computed, which was used in the computation of the other air data parameters, the air pressure, dynamic impact pressure and Mach number, following from the equations stated in Section 5.1.1. Thus, according to this information, a certain form of state estimation could be done by the FADS itself, without need of an estimator.

First of all, the layout of the sensing system had to be defined, as this states which ports are used for which state variables. Ellsworth and Whitmore (2007) and Karlgaard and Schoenenberger (2017) describe two vehicles for which the FADS system was applied. The former one discusses the application of the sensor to a suborbital plane, namely the X-34, whereas the latter one describes the application of the system on a planetary probe performing re-entry on Mars. As the application of Ellsworth and Whitmore (2007) is similar to the case described in this thesis, it is chosen to use the same sensor layout. Figure 8.9 shows the definition of the clock angle λ and cone angle ϕ, which together define the location of a specific pressure port. It is described that eight pressure ports are used, which are divided in such a way that when looking from the front of the nose cap, every quadrant contains at least one pressure port. By taking cone angles of 0 and 180 degrees, one is able to minimize the effect of the angle of sideslip, whereas cone angles of 90 and -90 minimize the effect of the angle of attack.

After implementation of the FADS the occurrence of port misalignments was evaluated, and it was found that for clock and cone angle misalignments up to 1°, the effects of this error were negligible. Whitmore and Moes (1994) used port misalignments up to 0.25°, for which angle-of-attack errors in the magnitude of 0.4° were found. However, as the FADS is added to the INS/GPS combination, any errors due to the FADS measurements may be compensated by the IMU and GPS measurements, which explains why port misalignments up to 1° were found to have a negligible influence on the attitude estimation. As a conclusion, they are included in the model, whereas it was chosen to discard the other errors of the FADS sensor, as was described in Section 5.1.1.

The configuration that is used for the implementation of the FADS is summarized in Table 8.3, and using the definitions from Figure 8.9 it can be seen that every quadrant indeed contains at least one pressure port. As a simulation of the re-entry flight is considered, the FADS measurements have to be generated manually, as was also done for the IMU and GPS. This is done by

$$p_{m,i} = q_c \left[\cos^2\theta_i + \epsilon \sin^2\theta_i\right] + p_\infty + \boldsymbol{\nu}_p \tag{8.15}$$

where p_∞ is taken directly from the US76 model output and represents the air pressure. Calibration parameter ϵ is taken as an input value and to ensure that the dynamic impact pressure and free stream air pressure have similar influence on the measurements, it is taken close to one, so that the term $\cos^2\theta_i + \epsilon \sin^2\theta_i$ approaches one. The calibration parameter is set to 0.9, however, as long as this value is the same in the measurement computations and in the state estimation, it does not influence the results in any way, and as was described in

Table 8.3: Configuration of the pressure ports used in the applied FADS system in the simulator (Ellsworth and Whitmore, 2007).

Port number	λ [deg]	ϕ [deg]
1	16.1	0
2	38.6	0
3	61.1	0
4	6.4	180
5	28.9	180
6	45	90
7	45	-90
8	90	0

Section 5.1.1, it was chosen not to model the FADS errors except the port misalignments in this application, but to leave them as a recommendation for future research. $\boldsymbol{\nu}_p$ represents the sensor noise and is taken as a Gaussian distribution with standard deviation of 1 N/m^2. Subscript i denotes that the incidence angle and air pressure are measured for every port i. Using the direct input of the Mach number the true dynamic impact pressure can be computed via

$$\frac{q_c}{p_\infty} = 166.92 M_\infty^2 \left[\frac{M_\infty^2}{7M_\infty^2 - 1} \right]^{2.5} - 1 \tag{8.16}$$

which is purely done to generate the measurements. Then, using the true angle of attack and angle of sideslip, the true incidence angle for all pressure ports is found. Subsequently, via Equation (8.15) the FADS measurements can be generated. The estimation of the air-data variables is then done using the equations that were described in Section 5.1.1.

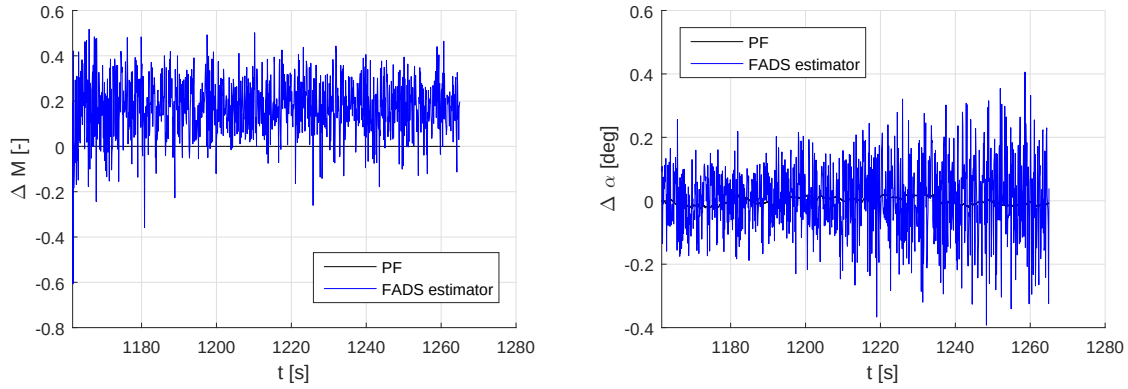
Finally, Whitmore and Moes (1994) describe that the FADS is operational up until a Mach number of approximately 13. In addition, Ellsworth and Whitmore (2007) state that for the FADS to be functional, the pressure measurements are required to have a specific magnitude, which is reach at an altitude of approximately 40 km. When investigating the nominal trajectory, Figure 6.9a and Figure 6.11b show that the constraining variable is the altitude, as the Mach number at a height of 40 km is already way below 13. Ellsworth and Whitmore (2007) also describe that the FADS system is able to generate measurements with a frequency of 25 Hz, however, for the state estimation it is chosen to align the GPS and the FADS to ensure a single measurement update, and thus the lowest frequency is taken as the constraining operating frequency, leading to a FADS frequency of 10 Hz. The evaluation of the effect of the measurement-update frequency of the estimators is done in Chapter 9.

Verification

Earlier in this section, it was stated that the measurements of the FADS were generated using the true values of free-stream air pressure, Mach number, angle of attack and angle of sideslip. Using the former two, the true dynamic impact pressure was computed, after which the air pressures at each FADS port were calculated. These measurements also contained a white noise factor and the included port misalignments in the clock and cone angles, which were taken to be a random number with $\sigma_{\delta\lambda} = \sigma_{\delta\phi} = 0.25°$, which corresponded to the maximum port misalignment that was evaluated by Whitmore and Moes (1994), and thus represented their worst-case scenario.

Verification of the measurements was done by discarding the port misalignments and the sensor noise, which resulted in the true pressures at the FADS ports. By reversing the computations, it was checked whether the calculated true free-stream air pressure, Mach number, angle of attack and angle of sideslip were the same as the true values from the simulator. Next to that, the built-in estimator, which contained these equations, was evaluated. First, the measurements without any errors were fed to the built-in estimator, after which again the true values of the aforementioned air-data variables were found. For the angle of attack, ports 1, 2, 3, 4, 5 and 8 from Table 8.3 were used, as they contained a clock and of either 0° or 180°, thereby discarding any influence of the angle of sideslip. For the angle of sideslip, the triples algorithm was used containing either port 6 or 7, as they contained a clock angle that was not aligned with the z-axis of the vehicle.

Then, the errors were added to the measurements, after which it was evaluated whether the built-in estimator was able to approximate the air-data variables, given that they were not 100% accurate. Finally, an extra verification test was done, which is valid for high Mach numbers. Following from Equation (8.16), it can be stated that the free-stream air pressure is negligible for high Mach numbers, and that the measured pressure



(a) (b)

Figure 8.10: (**a**) - Difference between actual Mach number and estimated Mach number for PF estimation and FADS estimation. The ΔM for the PF estimation is non-zero, but significantly smaller than for the FADS built-in estimator. (**b**) - Difference between actual angle of attack and estimated angle of attack for PF estimation and FADS estimation.

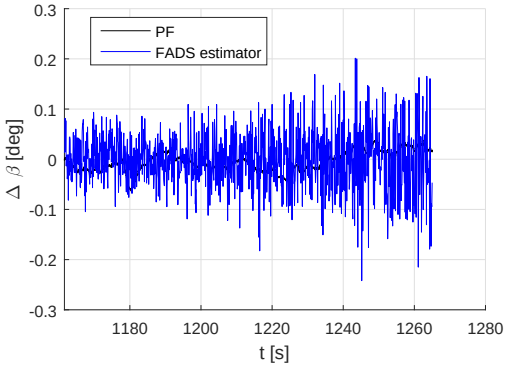


Figure 8.11: Difference between actual angle of sideslip and estimated angle of sideslip for PF estimation and FADS estimation.

is approximately equal to the dynamic pressure. During the verification process, this was tested and found to be valid. Due to this, the FADS measurements were verified, and for the implementation of this sensor in the navigation filter, it was evaluated whether the raw measurements or the approximations from the built-in estimator were to be used.

Simulation

The previous subsection described the verification process of the FADS, however, as the application of this sensor to a re-entry navigation module is new, the measurements are discussed in more detail. It was described that the FADS contains a built-in estimator, capable of approximating the Mach number, air pressure, dynamic impact pressure, angle of attack and angle of sideslip. However, when these estimated parameters are fed to one of the evaluated estimators, white noise becomes colored noise, which will influence the eventual state estimation. Therefore it has to be decided on whether to use the raw pressure measurements of the FADS in combination with an estimation filter or to use the estimated variables from the FADS system, which will be evaluated in this subsection.

To evaluate the effectiveness of the built-in estimator of the FADS system, the air-data state-variable results are compared for the FADS estimator and for the particle filter, which will be described in Section 8.2.3. For this comparison, a noise standard deviation of 1 N/m^2 was used in both cases. Figures 8.10a, 8.10b and 8.11 show that the FADS estimator has a much worse performance for the same noise level than the PF/FADS combination. For the Mach number, a noise standard deviation of 1 N/m^2 induces differences between zero and 0.4 (whereas the difference for the PF/FADS combination is in the order of 10^{-3}), which will eventually have an effect on the control system, as the Mach number serves as an input for this system in terms of control-derivative computation. The angle of attack also shows much worse behavior for the FADS estimator.

However, all variations in Mach number and angle of attack most likely will have a low impact on the actual computation of the control derivatives, as the variations are still small. Similar results can be seen for the angle of sideslip, which also shows a relatively poor performance with respect to the PF/FADS combination.

In all three figures it can be seen that the simulation results are started somewhat after $t = 1160$ s, which is the point where the altitude becomes smaller than 40 km and thus the FADS is initiated. In the FADS estimator, the Mach number is approximated by a summation of coefficients, which will have an absolute error that apparently lies somewhere between 0.15 and 0.2 for high Mach numbers, as can be seen in Figure 8.10a. From this comparison it is decided that it is more efficient to use raw FADS measurements in combination with a filter than the built-in FADS estimator. However, it is decided to use the FADS estimator as a contingency, for the situation where the state estimation is corrupted due to some reason (for example, GPS outage).

The following section will handle the filters that were used for state estimation, and to implement the FADS, the air data variables have to be related to the state variables. When looking at Equation (8.15) and Equation (8.16), it can be seen that none of the air data variables also appear in the state vector from Equation (8.13). For the estimation of the FADS measurements the calculations that have to be performed will therefore have to be defined. First the velocity vector will be transformed from the inertial frame to the rotating frame, for which the correction for the rotation of the Earth has to be computed

$$\mathbf{V}_R = \mathbf{C}_{\mathbf{R,I}} \left(\mathbf{V}_I + \begin{pmatrix} 0 \\ 0 \\ \omega_{cb} \end{pmatrix} \times \mathbf{X}_I \right) \tag{8.17}$$

The norm of this vector is known as the velocity norm with respect to the atmosphere and is used in the computation of the Mach number. To calculate the speed of sound, which is also required for the Mach number, an atmospheric model needs to be implemented, for which the altitude h is used that is extracted from the norm of $\mathbf{X}_I$. The atmospheric model that is implemented was discussed in Section 4.2 and is the same one as was used in the design of the simulator. Following from this model the air temperature and the air pressure are found. The latter one is directly used as p_∞, whereas the former one is used in the computation of the speed of sound via Equation (3.13) and thus in the Mach number via Equation (3.12). Using the air pressure and Mach number in Equation (8.16) the dynamic impact pressure can be computed, which leaves the aerodynamic angles α and β.

In Equation (8.7) the transformation matrix between the inertial frame and the body frame was found, which uses part of the state vector, *i.e.*, the quaternions. This matrix can also be computed via the longitude, latitude, flight-path angle, heading angle, bank angle, angle of attack and angle of sideslip from

$$\begin{aligned} \mathbf{C}_{I,B} &= \mathbf{C}_{R,V}\mathbf{C}_{V,A}\mathbf{C}_{A,B} \\ &= \mathbf{C}_3(-\tilde{\tau})\,\mathbf{C}_2\left(\frac{\pi}{2}+\delta\right)\mathbf{C}_3(-\chi)\,\mathbf{C}_2(-\gamma)\,\mathbf{C}_1(\sigma)\,\mathbf{C}_3(\beta)\,\mathbf{C}_2(-\alpha) \end{aligned} \tag{8.18}$$

where $\tilde{\tau} = \tau + \omega_{cb}(t - t_0)$ compensates for the rotation of the Earth. Using this, and the transformation matrix from the quaternions $\mathbf{C}_{I,B}(\mathbf{Q})$, the following expression is found:

$$\mathbf{C} = \mathbf{C}_1(\sigma)\,\mathbf{C}_3(\beta)\,\mathbf{C}_2(-\alpha) = \mathbf{C}_2(\gamma)\,\mathbf{C}_3(\chi)\,\mathbf{C}_2\left(-\frac{\pi}{2}-\delta\right)\mathbf{C}_3(\tilde{\tau})\,\mathbf{C}_{I,B}(\mathbf{Q}) \tag{8.19}$$

The individual transformation matrices of this expression are found in Appendix A.1. From this equation, it is found that:

$$\alpha = \text{atan2}(\mathbf{C}(1,3), \mathbf{C}(1,1)) \tag{8.20}$$

$$\beta = \text{asin}(\mathbf{C}(1,2)) \tag{8.21}$$

which concludes the transformation of the state variables to the air-data variables. In this section, the implementation and verification of the IMU, GPS, and FADS was discussed. Table 8.4 contains a brief summary of the type of measurements of the sensors, as well as their operational range.

8.2 Estimators

In the previous section elaboration on the sensor theory that was described in Section 5.1.1 was done and the implementation, simulation and verification of the IMU, GPS and FADS were discussed. This section then will

Table 8.4: Overview of the sensor input, output and operational range.

	IMU	**GPS**	**FADS**
Measurements	Acceleration $\mathbf{a}$ Rotational rate $\boldsymbol{\omega}$	Pseudorange $\tilde{\rho}$ Range rate $\dot{\tilde{\rho}}$ Carrier phase differences $\Delta\Phi_1$, $\Delta\Phi_2$, and $\Delta\Phi_3$	Port pressures $p_{m,i}$
Operational range	During whole flight	For $h > 82$ km and $h < 42$ km	For $h < 40$ km and $M < 13$

handle the three estimators that were introduced in Section 5.1.2, and similarly, every estimator is evaluated on its implementation and verification. Simulation of the estimators is done in Sections 9.1 to 9.3, where they are directly compared to each other to evaluate their relative performance.

The estimator implementation is done in a similar fashion for each filter and is disconnected from the simulator itself. As the filter selection is done based on the individual performance of the filters themselves, it is chosen to generate a measurement file using the simulator with integrated sensors. Then, using this measurement file as an input, the three filters are implemented, verified and evaluated on their performance. Finally, after a comparison, a trade-off can be done, after which the chosen filter is integrated in the simulator.

Before the implementation and verification of the estimators is done, the quaternion estimation is addressed, which can be done in multiple ways. In the definition of the quaternion vector $\mathbf{Q} = (Q_1, Q_2, Q_3, Q_4)^T$ it is implied that:

$$\text{norm}(\mathbf{Q}) = \sqrt{Q_1^2 + Q_2^2 + Q_3^2 + Q_4^2} = 1 \tag{8.22}$$

During the state estimation, this constraint is not taken into account, and all quaternions are propagated individually. Choukroun et al. (2002) describes that the quaternion estimate is to be normalized by

$$\overline{\mathbf{Q}}^* = \frac{\overline{\mathbf{Q}}}{|\overline{\mathbf{Q}}|} \tag{8.23}$$

where $\overline{\mathbf{Q}}$ is the estimated quaternion state. The corresponding covariance is updated by:

$$\mathbf{P}^* = \mathbf{P} + \left(1 - \frac{1}{|\overline{\mathbf{Q}}|}\right)^2 \overline{\mathbf{Q}}\overline{\mathbf{Q}}^T \tag{8.24}$$

Using these transformations, the quaternion norm error that is induced in the state estimation is mapped over the three vector components and the scalar component. A different technique is to estimate three quaternions and compute the fourth component of the quaternion vector using Equation (8.22). For this method, the estimation error is mapped over one component only. In Appendix A.2 the conversion from quaternions to Euler angles is shown, and it is seen that not every angle depends on all quaternions. Using latter technique, the estimation error over the calculated roll, pitch, and yaw angles may be misrepresenting the actual estimation error of the attitude, and this is avoided by spreading the error over all quaternion components, as was described by Choukroun et al. (2002).

8.2.1 Extended Kalman Filter

For convenience, this section is structured similarly to Section 5.1.2 and therefore it is started with the implementation and verification of the Extended Kalman Filter (EKF). It was described that this estimator uses a first order Taylor approximation of the nonlinear EoM in combination with the previous estimate to obtain the state estimate of the current time step. As will become clear for all filters that are discussed in this chapter, the state estimation consists of a prediction step and a correction step (measurement update).

In the previous section it was discussed that in a specific part of the atmosphere, the GPS signals are blocked due to ionized particles in the air. In this phase the estimator cannot perform the measurement update, as due to the altitude, no FADS measurements are available either. Due to this the re-entry flight is divided into three phases.

- Phase I: from entry to blackout phase

Table 8.5: Summary of the system dynamics and measurement equations that is used for implementation of the three filters.

Equations of motion	Measurement equations
$\dot{x} = v_x$	$\mathbf{a}_m = (\mathbf{I} + \mathbf{S}_a)\,\mathbf{a} + \mathbf{b}_a$
$\dot{y} = v_y$	$\boldsymbol{\omega}_m = (\mathbf{I} + \mathbf{S}_\omega)\,\boldsymbol{\omega} + \mathbf{b}_\omega$
$\dot{z} = v_z$	$\tilde{\rho}_i = \sqrt{\left(\mathbf{r}_{\text{GPS,i}} - \mathbf{r}\right)^T \left(\mathbf{r}_{\text{GPS,i}} - \mathbf{r}\right)} + c \cdot b_c$
$\dot{v}_x = a_{I,x_e} + g_x$	$\dot{\tilde{\rho}}_i = \frac{(\mathbf{r}_{\text{GPS,i}} - \mathbf{r})^T (\dot{\mathbf{r}}_{\text{GPS,i}} - \dot{\mathbf{r}})}{\sqrt{(\mathbf{r}_{\text{GPS,i}} - \mathbf{r})^T (\mathbf{r}_{\text{GPS,i}} - \mathbf{r})}} + c \cdot d_c$
$\dot{v}_y = a_{I,y_e} + g_y$	$\Delta\Phi_{ij} = -\frac{f_s}{c} \mathbf{b}_{ij_B}^T \mathbf{C}_{B,I} \mathbf{e}_{i,I}$
$\dot{v}_z = a_{I,z_e} + g_z$	$p_{m,k} = q_c \left[\cos^2\theta_k + \epsilon \sin^2\theta_k\right] + p_\infty$
$\dot{\mathbf{Q}}_{I,B} = \frac{1}{2} \begin{bmatrix} -Q_4 & -Q_3 & Q_2 \\ Q_3 & -Q_4 & -Q_1 \\ -Q_2 & Q_1 & -Q_4 \\ Q_1 & Q_2 & Q_3 \end{bmatrix} \begin{pmatrix} p_e \\ q_e \\ r_e \end{pmatrix}$	
$\dot{b}_c = d_c$	
$\dot{d}_c = \dot{s}_x = \dot{b}_x = \dot{b}_y = \dot{b}_z = \dot{b}_p = \dot{b}_q = \dot{b}_r = 0$	

- Phase II: blackout phase
- Phase III: from blackout phase to TAEM interface.

Previously, it was decided that the blackout phase of the mission that is described in this thesis ranges from 82 km to 42 km, based on post-processed flight results of the STS-118 mission described by Goodman (2008). This range, however, depends on the velocity, altitude, and vehicle shape, as the blackout phase is caused by the ionization of atmospheric particles due to the heat that is generated as a result of the high velocity of the vehicle. The heat generation is then dependent on the shape of the vehicle. Due to this division, the filters are each divided into three phases as well, where the second phase, which represents the blackout phase, only contains the nominal propagation of the IMU measurements, and thus is the same for all filters. The first and third phase have different initiations, as the flight conditions at the start of each phase are different as well.

For the implementation of the three filters, the system dynamics and measurement equations are summarized in Table 8.5. In these equations, subscript i denotes the i^{th} GPS satellite, j denotes the j^{th} slave antenna, and k denotes the k^{th} FADS pressure port.

Implementation

As discussed in the introduction of this section, the three filters are tested in a separate environment with the controlled re-entry trajectory output together with IMU, GPS and FADS measurements as an input. For the EKF first the covariance matrix $\boldsymbol{P}$, the process noise matrix $\boldsymbol{Q}$ and the measurement noise matrix $\boldsymbol{R}$ have to be initiated. Next to that, an initial guess is to be defined; for phase I this is an actual initial guess, whereas for phase III this is the final estimate of phase II.

Earlier it was described that once in every 20 steps the chosen filter was activated, which basically updates the propagated state from the IMU. This update is different per filter and is based on the theory that was described in Section 5.1.2; the IMU propagation that was described in Section 8.1.1 is similar for both Kalman filters. In terms of the EKF, a clear prediction and correction step can be found. In the former one, a prediction of the state and the covariance matrix have to be made, which is done by Equation (5.36) and Equation (5.48). The system dynamics matrix $\mathbf{A}_k$ contains the differential EoM that were described in Table 8.5.

$$\mathbf{A} = \left.\frac{\partial \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t)}{\partial \mathbf{x}}\right|_{\mathbf{x}_t, \mathbf{u}_t} \tag{8.25}$$

After the prediction step, which is strongly related to the IMU propagation, the measurement update is to be performed. In this update the estimated measurements $h\left(\bar{\mathbf{x}}\right)$ from the equations stated in Table 8.5, the actual measurements $\mathbf{z}$ and the measurement model matrix $\mathbf{H}\left(\bar{\mathbf{x}}\right)$ are required. The latter one is computed by taking the partial derivatives of the measurement equations with respect to the state variables.

$$\mathbf{H} = \left.\frac{\partial \mathbf{h}\left(\bar{\mathbf{x}}\right)}{\partial \bar{\mathbf{x}}}\right|_{\bar{\mathbf{x}}_t} \tag{8.26}$$

Table 8.6: Input parameters of the process and measurement noise matrices **Q** and **R**, found after tuning the EKF. The process noise for position, velocity, attitude, accelerometer bias b_a, and gyroscope drift b_ω is the same for each of their respective components.

State noise		Measurement noise	
Phase I	Phase III	Phase I	Phase III
$\sigma_X = 30$ m	$\sigma_X = 40$ m	$\sigma_\rho = 30$ m	$\sigma_\rho = 15$ m
$\sigma_V = 0.8$ m/s	$\sigma_V = 1$ m/s	$\sigma_{\dot{\rho}} = 0.2$ m/s	$\sigma_{\dot{\rho}} = 0.05$ m/s
$\sigma_Q = 0.05$	$\sigma_Q = 0.05$	$\sigma_{\Delta\Phi} = 0.05$	$\sigma_{\Delta\Phi} = 0.1$
$\sigma_{b_c} = 30/c$ m	$\sigma_{b_c} = 30/c$ m	$\sigma_a = 2\cdot10^{-4}$ m/s^2	$\sigma_a = 2\cdot10^{-4}$ m/s^2
$\sigma_{d_c} = 0.3/c$ m/s	$\sigma_{d_c} = 0.3/c$ m/s	$\sigma_\omega = 5\cdot10^{-4}$ rad/s	$\sigma_\omega = 5\cdot10^{-4}$ rad/s
$\sigma_{s_x} = 10^{-3}$	$\sigma_{s_x} = 2\cdot10^{-3}$		$\sigma_p = 100$ N/m^2
$\sigma_{b_a} = 10^{-4}$ m/s^2	$\sigma_{b_a} = 10^{-5}$ m/s^2		
$\sigma_{b_\omega} = 5\cdot10^{-4}$ rad/s	$\sigma_{b_\omega} = 5\cdot10^{-4}$ rad/s		

The entries of this matrix are derived in Appendix B. The computations that are done in the measurement update are performed using the predicted state, denoted by $\bar{\mathbf{x}}$. For the raw FADS measurements, dependent on free-stream air pressure, impact pressure, Mach number, angle of attack and angle of sideslip, a lot of partial derivatives were derived, which make the implementation of the FADS system more complex for the EKF. However, the implementation of pressure measurements instead of the processed measurements (for example angle of attack and Mach number) ensure that the filter models the actual FADS noise and not the mapped noise, thereby being more efficient.

After the measurement model matrix and the estimated measurements are computed, the Kalman gain is computed, which is required for the update of the state and covariance matrix. The output state of the filter is then processed to obtain the variables that provide more insight of the estimated state; quaternions are converted to roll, pitch and yaw angle, the altitude and velocity norm are computed from the individual components. As was described, the initialization of the filter is done before phase I and phase III and is summarized in Table 8.6. The process matrix **Q** is defined as:

$$\mathbf{Q} = \text{diag}\left(\left[\sigma_x^2, \sigma_y^2, ..., \sigma_{b_r}^2\right]\right) \cdot \Delta t \tag{8.27}$$

The measurement noise matrix **R** is the diagonal matrix of the measurement variances from Table 8.6, and the initial covariance matrix **P** is the diagonal matrix of the state variables variances from Table 8.6. The values that are shown in Table 8.6 are found after tuning the filter. Note that there is no optimal performance for a filter, as increasing one of the noise values may increase performance for one state variable while decreasing the performance for another one. The state initialization for position, velocity and attitude is defined via the actual initial state, whereas for the sensor errors an initial value is taken independent of the real value. The initial position and attitude are taken from the research performed by Mooij and Chu (2002), whereas the initial velocity error was taken somewhat larger:

$$\Delta X_0 = [30, -30, 35]^T \text{ m} \tag{8.28a}$$

$$\Delta V_0 = [-1, -1, 1]^T \text{ m/s} \tag{8.28b}$$

$$\Delta Q_0 = [0.05, -0.05, -0.05, 0.05]^T \tag{8.28c}$$

Next to that, $b_{c_0} = 30/c$ m, $d_{c_0} = 0.3/c$, $s_{x_0} = 4\cdot10^{-4}$, $b_{x_0} = 2.5\cdot10^{-4}$ m/s^2, $b_{z_0} = 3.5\cdot10^{-4}$ m/s^2, $b_{p_0} = 10^{-3}$ rad/s, $b_{q_0} = -10^{-3}$ rad/s and $b_{r_0} = 10^{-3}$ rad/s. In the previous values and in Table 8.6, the value c refers to the speed of light. The initial conditions of the state variables were chosen such that they are sufficiently large so that convergence of the filter can be visualized easily. Next to that, a filter handling a large initial state error is deemed to have a good performance. As a brief summary, the implementation of the EKF is initiated by the aforementioned initial conditions, after which the equations from Algorithm 1 are used. The system dynamics matrix and the estimated measurements are computed using the equations from Table 8.5 and the measurement model matrix is calculated using the derivatives from Appendix B.

Verification

During the implementation process of the EKF verification had to be done to ensure that the used equations, theories and the combination of all elements were implemented correctly. Both processes were done simultaneously as it was chosen to do the implementation in different steps. First the position and velocity

were estimated, using the true attitude and sensor errors. The obtained results were checked and when they were deemed correct, the next part of the state vector, namely the attitude, was added, after which the same procedure was handled. Subsequently the same was done for the GPS errors, the accelerometer errors and the gyroscope errors.

The EKF posed an extra difficulty on the verification process, as it did not use the nonlinear EoM, but a first order Taylor approximation, due to which a lot of partial derivatives had to be computed. These equations were first derived by hand, after which they were verified using Maple software. The same was done for the measurement equations in the **H** matrix. Finally, a verification test was designed for the multiplication of the Kalman gain with the difference between measurements and estimated measurements, which was denoted as the state measurement update. At the start of the estimation process this vector should contain large values that decrease over time (the update should go to zero), as the difference between the state and the estimation should also approach zero over time, at least when the filter shows correct convergence.

Finally, as system verification, the differences between the actual state and the chosen state variables were evaluated after which the EKF was said to be verified. Note that these results contain the estimations of the variables that were shown in Equation (8.13) and are not necessarily the variables of interest. The estimated state of the navigation module is fed to the control system (and for other systems it may be fed to the guidance module), for which transformation between state variables has to be done. This however will be discussed in Sections 9.1 to 9.4. To wrap up the implementation and verification process of the EKF, both filters for phase I and phase III were tuned to increase their performance on the variables of interest for the control system.

8.2.2 Unscented Kalman Filter

The second filter that was discussed in Section 5.1.2 was the Unscented Kalman Filter (UKF), which is based on the use of so-called sigma points to compute the state estimate. Beforehand it was stated that the main difference between the UKF and the EKF is that the former one uses the nonlinear EoM, whereas the latter one uses the first order Taylor approximation as was discussed in the previous subsection.

Implementation

Similar as for the EKF, the implementation of the UKF is started by defining the initial covariance matrix, the process noise matrix and the measurement noise matrix. Again, **P** contains the initial state variances, **Q** is a diagonal matrix with the multiplications of the state variances and the differential Δt and **R** is a diagonal matrix containing the measurement variances. For a fair comparison, the initial state vector of the UKF is taken equal to the initial state vector of the EKF. Next to the initialization of these matrices and vectors, also the UKF properties have to be defined. In Section 5.1.2 the number of state variables L were found, which is equal to 18. The scaling parameter α is set to 0.5 and the extra degree of freedom β was stated to be 2, following from a similar study by Gross et al. (2010). Then, following from Crassidis (2005) it is found that a common value of scaling parameter $\kappa = 3 - L$, which thus leads to a value of -15. Using these parameters, it can be computed that

$$\lambda = \alpha^2 (L + \kappa) - L = 0.25(18 + 3 - 18) - 18 = -17.25 \tag{8.29}$$

Then, using these parameters, the output state and covariance matrix of the previous time step and the process noise matrix, 37 sigma points can be computed, following from Equation (5.50), where Cholesky decomposition is used to calculate the square root of the sum of the covariance matrix and the process noise matrix. Using Equation (5.51) then the corresponding weights are computed. The principle of the UKF is that $2L + 1$ possible states are defined, which are fed to the EoM that are defined in Table 8.5. From this UT output, which is a new set of $2L + 1$ states, the weighted sum is computed to be the predicted state and the weighted covariance is calculated similarly to $\bar{\mathbf{P}}$ from the EKF.

In the measurement update the transformed sigma points from the prediction step are used to estimate the measurement sigma points, using the measurement equations from Table 8.5. The measurement model matrix **H** is omitted, which discards the probability of encountering an error in the derivation of the partial derivatives or in the calculation of this matrix. Using the weighted sum of the new sigma points, the measurement and the observation covariance can be estimated. Next to that the cross-covariance is computed, which leads to the Kalman gain. The Kalman gain is then required to compute the state estimate and the updated covariance matrix, all following from the equations that were shown in Algorithm 2.

Similar to the EKF, the UKF is initiated for phase I and III only, due to the absence of measurements in phase II. Using this, it can also be said that the state estimation in phase II is exactly the same for EKF and UKF (and PF). The initial state vector was mentioned in Section 8.2.1, the noise components of the UKF filter are summarized in Table 8.7.

Table 8.7: Input parameters of the process and measurement noise matrices **Q** and **R**, found after tuning the UKF. The process noise for position, velocity, attitude, accelerometer bias b_a, and gyroscope drift b_ω is the same for each of their respective components.

State noise		Measurement noise	
Phase I	Phase III	Phase I	Phase III
$\sigma_X = 5$ m	$\sigma_X = 5$ m	$\sigma_\rho = 30$ m	$\sigma_\rho = 10$ m
$\sigma_V = 0.05$ m/s	$\sigma_V = 0.1$ m/s	$\sigma_{\dot{\rho}} = 0.2$ m/s	$\sigma_{\dot{\rho}} = 0.5$ m/s
$\sigma_Q = 0.005$	$\sigma_Q = 0.001$	$\sigma_{\Delta\Phi} = 0.1$	$\sigma_{\Delta\Phi} = 0.03$
$\sigma_{b_c} = 8/c$ m	$\sigma_{b_c} = 30/c$ m m/s	$\sigma_a = 2 \cdot 10^{-4}$ m/s^2	$\sigma_a = 2 \cdot 10^{-4}$ m/s^2
$\sigma_{d_c} = 0.03/c$ m/s	$\sigma_{d_c} = 0.3/c$ m/s	$\sigma_\omega = 5 \cdot 10^{-4}$ rad/s	$\sigma_\omega = 5 \cdot 10^{-4}$ rad/s
$\sigma_{s_x} = 10^{-2}$	$\sigma_{s_x} = 10^{-3}$		$\sigma_p = 10$ N/m^2
$\sigma_{b_a} = 10^{-4}$ m/s^2	$\sigma_{b_a} = 10^{-5}$ m/s^2		
$\sigma_{b_\omega} = 5 \cdot 10^{-4}$ rad/s	$\sigma_{b_\omega} = 5 \cdot 10^{-4}$ rad/s		

It is seen that the noise components of the UKF differ for several variables with those that were found during the EKF implementation. This is caused by the fact that both filters require different noise components for optimal performance of the state variables that have priority.

Concluding the implementation of the UKF, the initialization of the filter is done similarly as for the EKF, and the equations from Algorithm 2 are used for this filter. The sigma points that are computed are transformed and used to calculate the state prediction using the EoM from Table 8.5. These transformed sigma points are then used to approximate the observations using the measurement equations from Table 8.5, which are subsequently required for the computation of the state update.

Verification

In terms of verification, it is found that the processes of the UKF and EKF are similar, if not equal. Again the state vector is estimated in parts, where it is started with taking only position and velocity. Due to the fact that both described filters up to this point belong to the Kalman filter family, most computations and algorithms are similar, which eases the verification of the UKF, as this process was already done for the EKF.

8.2.3 Particle Filter

The third estimator that was evaluated was the Particle Filter (PF), as was seen in Section 5.1.2. Whereas the other two filters are part of the Kalman filter family and thus are similar, the PF still uses a prediction and correction step but is based on a different principle. Again, for the PF, a distinction will be made between the implementation and verification in their respective subsections. Note that the IMU propagation is not done by simply performing the Euler integration, but still uses the particle clouds, only in combination with IMU measurements.

Implementation

For the Kalman filters a covariance matrix, process noise matrix, measurement noise matrix and an initial state were defined, which are all done differently for the PF. The initiation of this filter, which again is done twice (once for phase I and once for phase III) is started by the definition of a cloud of N particles and their respective weights. Using the theory from Section 5.1.2 it is found that the main principle of the PF is that a number of N particles is generated, each representing a possible state. These particles are assigned a weight, after which they are all propagated using the EoM from Table 8.5, which is similar to the propagation of the sigma points in the UKF. In the measurement update, the measurements are estimated and compared to the actual observations for each particle, resulting in a cloud with observation differences. Using a Probability Density Function (PDF) the differences that are small are assigned a large value, whereas the differences that are large are assigned a small value according to the properties of this PDF. Using these values the particle weights are modified, which, together with the particles from the predicted state lead to the state estimate after the process of resampling. The new set of particles and weights is used as the input for the next state.

Due to the fact that multiple types of measurements are used, the probability distributions for one state variable may be different for specific measurements. For example, the probability distribution of a quaternion that is based on a carrier phase measurement is not necessarily similar to the distribution of a position component that is based on pseudorange and range rate measurements. To avoid that the state variables are estimated on the wrong measurements, the state vector is divided into particles X for position, V for velocity,

Algorithm 4 Sampling Importance Resampling

if $N_{\text{eff}} < \eta_{\text{SIR}} \cdot N$ **then**
  $A = \text{cumsum}\left(x_k^-\right)$
  $u = \left[\left(0, \frac{1}{N}, ..., 1 - \frac{1}{N}\right) + \frac{\text{rand}(1)}{N}, 1\right]$
  $i = 1$
  $j = 1$
  while $i \leq N$ **do**
    if $u(i) < A(j)$ **then**
      $c_x(i) = j$
      $i = i + 1$
    else
      $j = j + 1$
    end if
  end while
  $X_k^+ = X_k^-(:, c_x)$
  $x_k^+ = \frac{1}{N}$
else
  $X_k^+ = X_k^-$
  $x_k^+ = x_k^-$
end if

Y for attitude, U for gyroscope errors, W for accelerometer errors and Z for GPS clock errors. Their assigned weights are denoted with their respective lower-case letter.

For the initiation of phase I the actual initial state is taken with a random distribution to one side of the state variable. For example, for the x-coordinate N particles are taken that are distributed between x_0 and $x_0 + \Delta x$ following a uniform distribution. The same is one for all state variables, and it is ensured that similar magnitudes are taken as for the initial state differences of the EKF and UKF for a fair comparison. The initial weights are taken to be $1/N$.

The prediction step of the PF is done using the EoM from Table 8.5, and every particle is fed to these equations. Note that even though six different particle clouds were defined, every first particle belongs to state 1, every second particle belongs to state 2, up until the N^{th} particle. Using the attitude particles Y the transformation matrix between the inertial frame and the body frame can be computed, and using position particles X the gravitational acceleration can be estimated. Then, the state prediction is done, where for every state variable a normally distributed noise component is added, with the initially stated noise standard deviation of that state variable. Note that this step is very similar to the IMU propagation for the Kalman filters.

The measurement update was already briefly touched upon, and is based on the probability distribution of the difference between the actual measurements and the estimated measurements, following from the observation equations in Table 8.5. This observation estimation is done for all N states, each adding a measurement noise component with an initially defined standard deviation. Using the real observations a vector of N differences can be obtained for all measurements. If a particular difference is small, it means that the state variables that were used to estimate that measurement apparently are close to the actual state, whereas a large difference defines a state that apparently is further from the truth. In other words, the likelihood of the state is evaluated using the measurements, and it is chosen to use a normally distributed PDF to define this likelihood. Subsequently, the weights of the six particle clouds are multiplied by the obtained likelihoods of the assigned measurements in an element-wise manner.

The updated weights are then normalized, after which they are used to compute the effective number of particles. This entity defines the number of particles that are sufficiently close to the truth so that they can be used for state estimation. When the particles are propagated over time, it will become clear that from the initial cloud only a few particles actually approach the truth, whereas the rest of the particles are assigned a negligible weight and basically can be discarded. This phenomenon is denoted as filter degeneration and can be avoided by the process of resampling. In Gross et al. (2010) and Fay and Speyer (2008) the technique of Sampling Importance Resampling (SIR) is described and it is chosen to use this method in the PF state estimation.

As seen in Algorithm 4, the SIR method takes the state particles as an input, in this case denoted by X_k^-,

Table 8.8: Input parameters of the process and measurement noise matrices **Q** and **R**, found after tuning the PF. The process noise for position, velocity, attitude, accelerometer bias b_a, and gyroscope drift b_ω is the same for each of their respective components.

State noise		Measurement noise	
Phase I	Phase III	Phase I	Phase III
$\sigma_X = 0.1$ m	$\sigma_X = 0.3$ m	$\sigma_\rho = 20$ m	$\sigma_\rho = 10$ m
$\sigma_V = 0.008$ m/s	$\sigma_V = 0.003$ m/s	$\sigma_{\dot{\rho}} = 0.5$ m/s	$\sigma_{\dot{\rho}} = 1$ m/s
$\sigma_Q = 0.005$	$\sigma_Q = 0.005$	$\sigma_{\Delta\Phi} = 0.03$	$\sigma_{\Delta\Phi} = 0.1$
$\sigma_{b_c} = 0.3/c$ m	$\sigma_{b_c} = 0.5/c$ m	$\sigma_a = 2 \cdot 10^{-3}$ m/s^2	$\sigma_a = 2 \cdot 10^{-3}$ m/s^2
$\sigma_{d_c} = 0.005/c$ m/s	$\sigma_{d_c} = 0.01/c$ m/s	$\sigma_\omega = 2 \cdot 10^{-5}$ rad/s	$\sigma_\omega = 2 \cdot 10^{-5}$ rad/s
$\sigma_{s_x} = 10^{-6}$	$\sigma_{s_x} = 10^{-6}$		$\sigma_p = 10$ N/m^2
$\sigma_{b_a} = 10^{-6}$ m/s^2	$\sigma_{b_a} = 10^{-4}$ m/s^2		
$\sigma_{b_\omega} = 5 \cdot 10^{-7}$ rad/s	$\sigma_{b_\omega} = 10^{-3}$ rad/s		

however, this parameter may be interchanged by any of the other particle vectors. The corresponding input weights are represented by x_k^-. Then, N is the real number of particles, whereas N_{eff} is the effective number of particles. Finally, η_{SIR} is an input constant that defines the activation of the SIR algorithm, and is a number between zero and one. For $\eta_{\text{SIR}} = 0$, no resampling is done whatsoever, whereas for $\eta_{\text{SIR}} = 1$, resampling is done in every step of the PF.

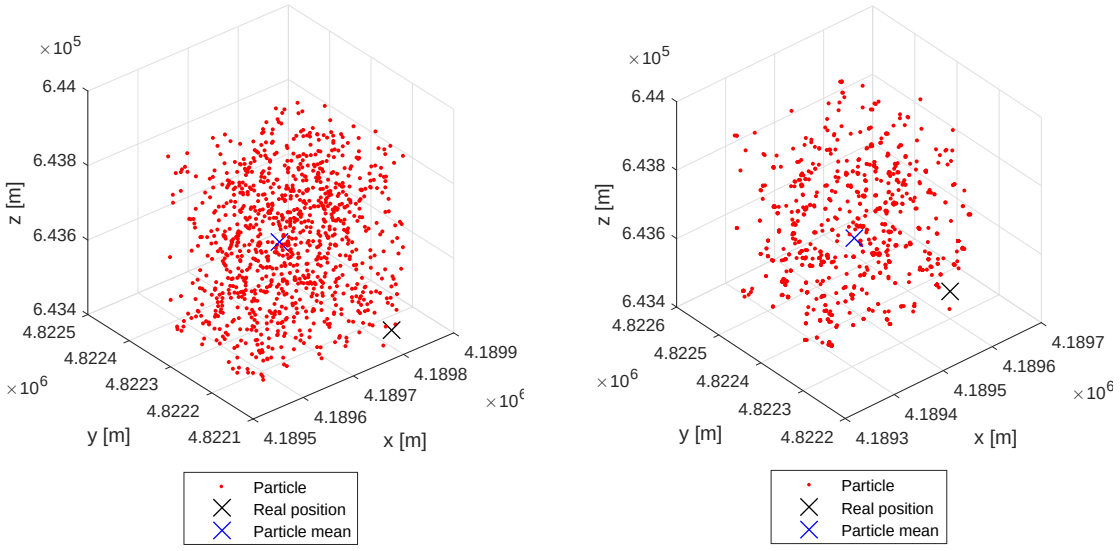
The SIR algorithm is then uses A, which contains the cumulative sum of the input weights, and u, which represents a vector containing random numbers between zero and one. Then, a loop is started, where it is evaluated whether the sample from vector u lies within the interval of the corresponding real weight vector A. If this is the case the assigned weight is deemed too small to be kept and thus is replaced by the previous weight that had sufficient magnitude. In the other case, the weight's significance is apparently sufficient and thus the weight can be kept. By looping through the whole vectors A and u the vector c_x is completed, which contains the particle indexes that have a sufficient weight, where a number of these particles is duplicated. The output particle cloud X_k^+ then contains the particles corresponding to the indexes from c_x. The new weights are then restored to a value of $1/N$. In the case that the number of effective particles is not less than the predefined limit, the SIR step is skipped. The final step of the PF is the computation of the state estimates, which is done by

$$\bar{\mathbf{x}} = \sum_{k=1}^{N} x_k^+ X_k^+ \tag{8.30}$$

Previously it was described that not all measurements were used for all particle states, and due to this not every state is estimated in every step. It was described that the GPS/FADS measurement update was performed once every 20 steps, due to the availability of the measurement data. In the step where all measurements are available, it is chosen to only use the GPS and FADS data. Ren and Ke (2010) describe that this is done to avoid any difficulties with the inertial sensor drift due to the fact that the integrated time is larger. During the IMU propagation therefore the accelerometer measurements are used to estimate position X, velocity V and accelerometer errors W, whereas the gyroscope observations are used to estimate attitude Y and gyro errors U. In the measurement update, the pseudorange is used for position, velocity and GPS errors Z, range rate is used for velocity and GPS errors and the carrier phase differences are only used for attitude estimation. The FADS measurements are applied to position, velocity and attitude.

This subsection described the implementation of the PF, where a distinction was made between the IMU propagation and the measurement update, which is similar to the other two filters that were discussed. In both steps the noise components were defined, which are also similar to the EKF and UKF. In Table 8.8 it can be seen that for some state variables the noise components differ drastically with respect to the EKF and UKF, which highlights the principal difference between the PF and the Kalman filter family.

As a brief summary of the implementation of the PF, the initialization of this filter is done by the definition of six particle clouds for position, velocity, attitude, GPS errors, accelerometer errors, and gyroscope errors and their assigned weights. Then, the equations from Algorithm 3 are used in combination with the EoM from Table 8.5 to obtain the state prediction. The measurement update is performed using the observation equations from Table 8.5 and by computing the probability density function for the difference between true and estimated measurements. Then, the new particle cloud is computed, after which the SIR technique from Algorithm 4 is applied to avoid degeneration of the filter.



(a) (b)

Figure 8.12: (**a**) - Particle distribution, mean and true value of the position vector X at the initiation of phase III. (**b**) - Particle distribution, mean and true value of the position vector X at 0.1 s after initiation of phase III.

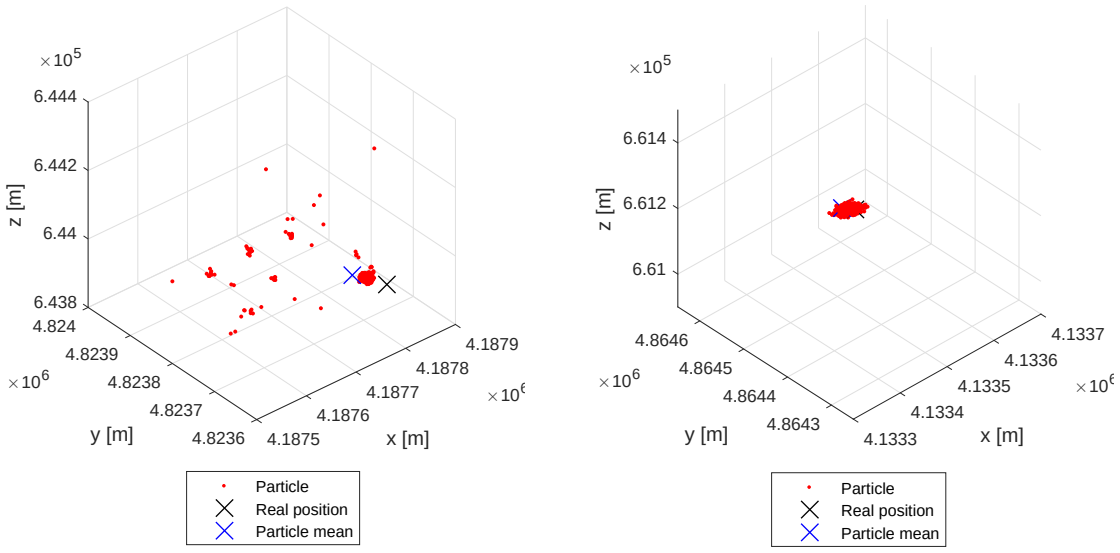
Verification

During the implementation of the PF, simultaneously the process of verification was executed, as was also done for the EKF and UKF. As the main principle of the PF is different, the verification process included some different steps as well. First of all, as the PF requires a lot of matrix computations, the sizes of the individual matrices were checked regularly, starting with the input matrices of the particles and their assigned weights. Then, the IMU propagation step was verified using a number of particles N of 1, that resembled the actual state exactly. When only propagating this particle, without any measurement updates it should be showing similar behavior as for the IMU propagation in the Kalman filters. After this, a similar approach was done for the measurement update, and again, by tuning the noise components in such a way, the singular particle should behave similarly as was seen in the Kalman filters.

After verification with $N = 1$, the number of particles was increased to $N = 10$ to check whether the matrix calculations were performed correctly; the initial particle cloud was given the uniform random distribution, and the PF functioned correctly, although 10 particles was deemed significantly too low to estimate the state accurately. Then, by increasing the number of particles and subsequently tuning the filter, the state error approximated zero, after which the PF was said to be verified.

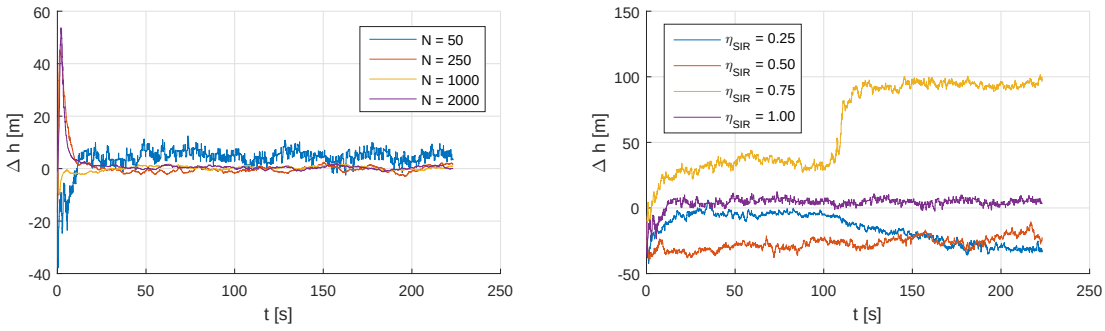
For the resampling technique a different verification method was handled, where the resampling frequency was set equal to the operating frequency of the IMU, *i.e.*, in every step the SIR algorithm was activated. When plotting the particles for a specific state, the particle cloud should become denser around the true state with time. By showing this, not only the convergence of the filter, but also the SIR algorithm was verified. This is shown in Figures 8.12a, 8.12b, 8.13a and 8.13b. These figures represent the particle cloud of position vector X in phase III of the entry flight. Figure 8.12a contains the initial particle cloud, with $N = 1000$. It is seen that the particles are randomly distributed over a specific volume, with the blue cross being the estimated mean of all particles. The black cross represents the true state, and it is seen that the estimated mean lies quite far from the truth. It is chosen to use phase III for this representation, as the input error for this phase is larger than for phase I due to the fact that the input for phase I is defined manually, whereas the input for phase III is the output of the blackout phase, in which state estimation cannot be done very accurately, as will be shown in Section 9.2. In Figure 8.12b the propagation of the particles after one measurement update, *i.e.*, 0.1 s, is seen. The particles are still randomly distributed over the volume, but are now clustered in groups, which highlights the principle of resampling. These groups represent particle duplicates, and it can be clearly seen that the PF filters out all particles with negligible weight.

Then, when looking at 1 second after initiation of the filter, which is shown in Figure 8.13a, a large cluster



(a) (b)

Figure 8.13: (**a**) - Particle distribution, mean and true value of the position vector X at 1 s after initiation of phase III. (**b**) - Particle distribution, mean and true value of the position vector X at 30 s after initiation of phase III.

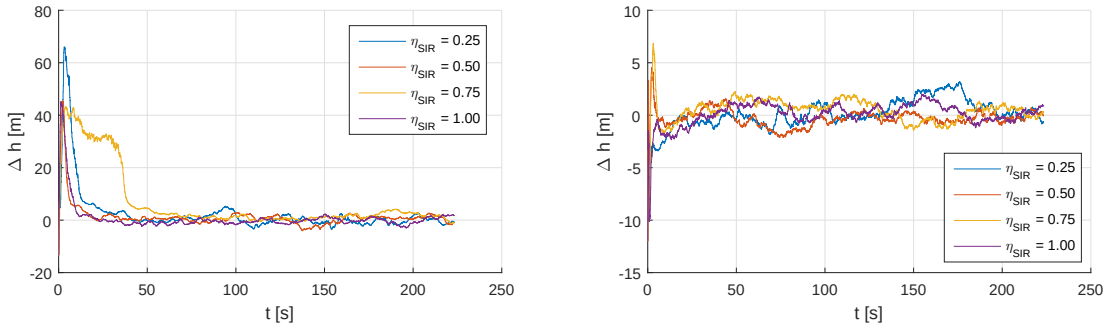


(a) (b)

Figure 8.14: (**a**) - Results for the variation of number of particles N for the PF in phase I. (**b**) - Variation of resampling constraint η_{SIR} for $N = 50$ in phase I.

of particles can be seen that is located near the true state. Due to the number of particles that lie in this cluster, the estimated mean also moves closer to the truth model, and the difference in position error can be clearly seen between Figure 8.13a, and Figures 8.12a and 8.12b. Finally, Figure 8.13b shows the particles at 30 seconds after initiation of the PF. It is seen that at this point they are all located near the truth, thereby taking the estimated mean very close to the true state as well. For convenience, the axes on all figures are scaled so that they show a similar volume of space. This way it can be seen that the particle distribution after 30 seconds is very accurate compared to the distribution at 0 and 0.1 s. Duplicating the particles every step could decrease the diversity of the filter, however, by applying process noise, this is avoided.

After verification and the evaluation of the SIR sequence, the effect of the number of particles and the resampling constraints may be interesting. Increasing N should also increase accuracy of the estimation, however, the computational loads will most likely go up as well. For the resampling constraint η_{SIR} a value near one means that the algorithm is activated a lot, which increases computational loads, but avoids degeneration of the filter. Figures 8.14a, 8.14b, 8.15a and 8.15b show the results of the variations of these variables. In Figure 8.14a the altitude estimation results of the PF for phase I are shown, where it is varied between $N = 50$ up until $N = 2000$ with $\eta_{\text{SIR}} = 1.00$. It can be stated that by increasing N, the accuracy of the estimation increases,



(a) (b)

Figure 8.15: (**a**) - Variation of resampling constraint η_{SIR} for $N = 250$ in phase I. (**b**) - Variation of resampling constraint η_{SIR} for $N = 1000$ in phase I.

Table 8.9: Estimations of the computation time for different number of particles and resampling constraints, averaged over 10 independent simulations.

N	t_{sim}	η_{SIM}	t_{sim}
50	30.0 s	0.25	68.5 s
250	40.3 s	0.50	68.7 s
1000	69.4 s	0.75	69.3 s
2000	151.4 s	1.00	69.4 s

up until a certain point, as the difference between $N = 1000$ and $N = 2000$ is small.

Figure 8.14b shows the variation in the resampling constraint for 50 particles. Previously, it was seen that the state estimation for this amount of particles is relatively inaccurate, and it can be seen that when resampling is not done in every time step, the filter does not converge during phase I. At initiation of the PF, N possible states are defined, and when this number is as small as 50, the probability of one of these possible states being close to the truth is small. When resampling is only done once every few steps, most of the time state estimation is performed using inaccurate states, leading to a filter for which the state error does not converge to zero, as is seen in Figure 8.14b. Then, by increasing the number of particles to 250, it can be seen that the filter is able to converge the error towards zero, for all values of η_{SIR}. For an even higher number of particles $N = 1000$ it is seen that for low resampling constraints the filter performs somewhat worse, but still provides an excellent state estimation. It was seen that from $N = 250$ the filter converges, however, the state error at the start of the phase becomes lower when increasing the number of particles to for example 1000. Finally, an unexpected phenomenon can be seen in Figure 8.15a, where the state estimation for a relatively high resampling constraint takes a long time to converge. It seems that the PF has multiple clusters of particles after some time and assumes the wrong cluster to be the truth up until $t \approx 40$ s, where the state error still goes towards zero.

Lastly, the difference in computation time for the different number of particles and for the different resampling constraint values for $N = 1000$ was evaluated. Following from these results, it was decided to use $N = 1000$, as this provides an accurate state estimation, whilst keeping the computation time under half of the flight time. Increasing to $N = 2000$ would not add that much accuracy to the results, whereas the computation time increases to almost the flight time. In terms of the resampling constraint, the difference in computational load is negligible. However, when taking a high resampling constraint, the filter will try to cluster the particles near the true state on every time step, which aids in decreasing the convergence time. Figure 8.15b shows that a resampling constraint of 1 induces the fastest convergence, although the differences again are small. For the implementation of the PF it is decided to use $\eta_{\text{SIR}} = 1$, as this has a small advantage of a lower convergence time for $N = 1000$.

9

Simulation of the Navigation Module

Previously, the sensors and estimators of which the theory was described in Sections 5.1.1 and 5.1.2 was used in Chapter 8 for the implementation of the IMU, GPS, and FADS, and the EKF, UKF, and PF. In this chapter, the results of the filters and sensor combinations are discussed for the three phases that were defined in Section 8.2.1. Phase I, from entry to the blackout phase, is discussed in Section 9.1, in which a comparison is made between the three filters. Then, in Section 9.2, the blackout phase is described, in which the state propagation is purely done by propagation of the IMU measurements and the EoM, and thus is exactly the same for all filters. Earlier in this report, it was chosen that the blackout phase ranges from 82 km to 42 km altitude. Subsequently, the phase from blackout to the TAEM interface is handled in Section 9.3. Finally, a trade-off is done between the three filters, after which the chosen filter is integrated in the simulator, of which the results are discussed in Section 9.4.

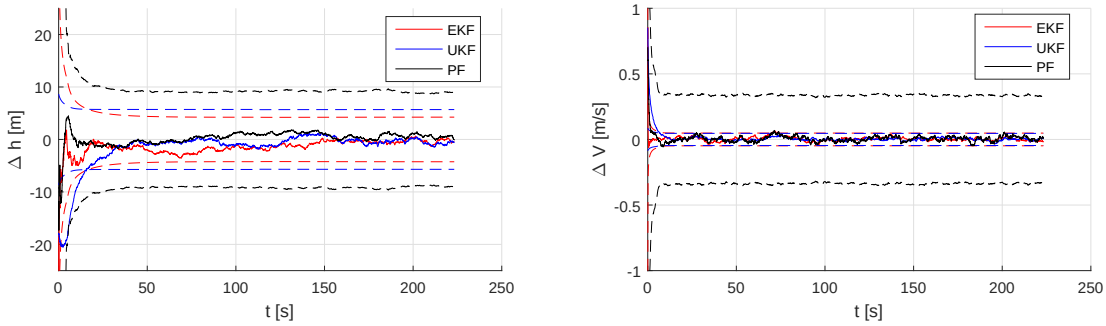
9.1 Phase I: From Entry to Blackout Phase

Previously, in Section 8.1, the implementation of the IMU sensor, the GPS receiver and the FADS system was seen. All three sensors were integrated in the simulator after which the verification was done as described in the respective subsections. Using the simulator, with integrated control system from Chapter 7 and these three sensors, a measurement file was created containing the trajectory of HORUS with corresponding acceleration and rotational rate measurements, pseudoranges, range rates, carrier phase differences and the location of the GPS satellites and finally a set of pressure measurements from the FADS. In separate files the estimators from Section 8.2 were tested, where the measurement file was used as an input. This approach ensured that the input data for all estimators were the same, so that a fair comparison could be made.

This section, together with Section 9.2 and Section 9.3 will describe the results that were obtained for the individual phases, where a comparison is made between the EKF, UKF and PF. The variables that are evaluated in each of these sections are:

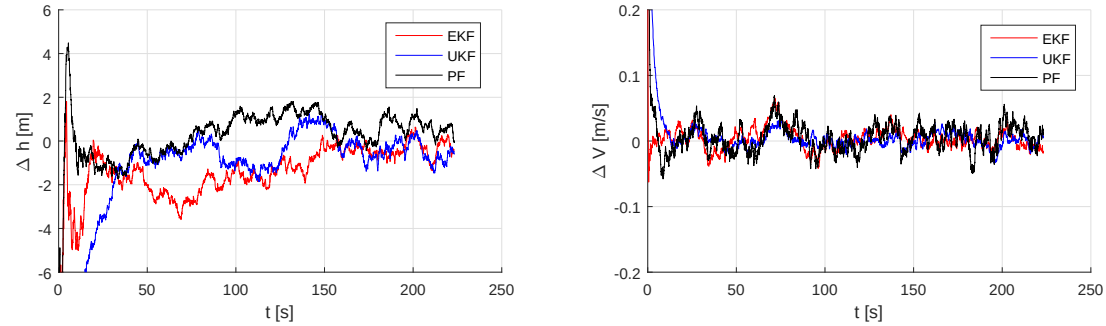
- Altitude h
- Velocity norm with respect to the atmosphere V
- Angle of attack α
- Angle of sideslip β
- Bank angle σ
- Clock bias b_c
- Clock drift d_c
- Scaling factor s_x
- Accelerometer bias b_z

Note that for all these variables, the difference between the estimated value and the true value is shown. For all estimators, the individual position and velocity components have shown converging results and therefore it is decided to provide the altitude and velocity, as these variables combine the individual components. As for the



(a) (b)

Figure 9.1: (**a**) - Comparison on altitude estimation error for phase I with covariance lines. (**b**) - Comparison on velocity norm estimation error for phase I with covariance lines.



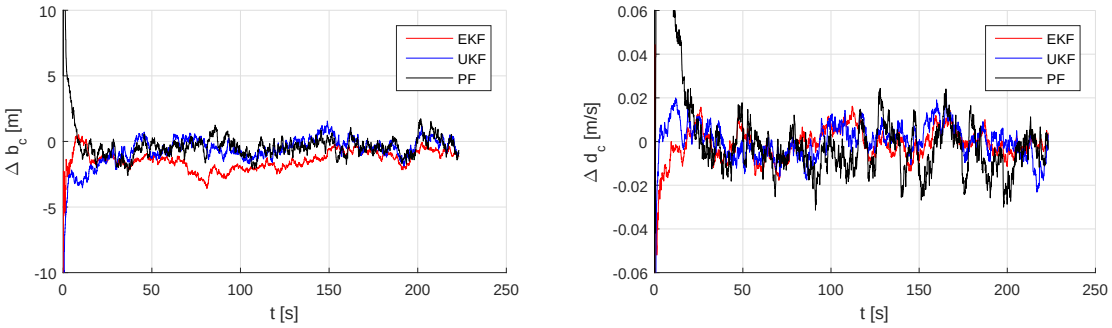
(a) (b)

Figure 9.2: (**a**) - Comparison on altitude estimation error for phase I. (**b**) - Comparison on velocity norm estimation error for phase I.

attitude, the angle of attack, angle of sideslip, and bank angle are chosen, as they provide more insight than quaternions. The use of roll, pitch, and yaw angles was also considered, however, as the aerodynamic angles appear in the input of the control system, it is chosen to show these angles. As the roll, pitch, and yaw angles are simply obtained by transforming the aerodynamic angles, the same conclusions should be drawn from both. The GPS errors are evaluated, as they appear to have a significant impact on the altitude and velocity. The accelerometer biases and gyroscope drifts showed similar behavior, whereas the scaling factor over the x-axis behaved different, and therefore a small selection is made, resulting in s_x and b_z. Note that the state variables that are not stated in this list are still included in the estimation process.

Finally, in all filters the use of random numbers was encountered, which influenced the estimation results. To ensure that the results are reproducable, a seed was defined, so that the random number generators generate the same numbers every simulation. For all simulations, `seed = 0` was used (also for the other two phases and the integrated system). Together with all estimations the covariance lines can be plotted, however, for the estimation of all three filters, the plots get chaotic. For the state estimation in the first phase, it was seen that position and velocity estimates converged, as well as the clock bias and drift. To show this converging behavior, the covariance lines are shown for the altitude and velocity, but show similar behavior for the GPS errors. The attitude estimation errors also converged, and, as was stated, the covariance lines are not shown as the plots get chaotic, but they behave similarly as the covariance lines in the altitude and velocity plots. Finally, for the IMU errors, the covariance lines are shown, as they show different behavior, which will be handled further on in this section.

Figures 9.1a, 9.1b, 9.2a and 9.2b show the comparison of the filters concerning altitude and velocity. For the former two figures, the covariance lines are shown, and it can be seen that all filters converge. In the latter two figures the actual estimated errors are shown somewhat clearer. Concerning the altitude error estimation, it can be concluded that all filters perform similarly, which is in line with the results from Gross et al. (2010), who showed that the EKF and UKF are very alike. The main difference is that the PF has a better performance in this study, which can be explained by the fact that a higher number of particles is used. From



(a) (b)

Figure 9.3: (**a**) - Comparison on clock bias estimation error for phase I. (**b**) - Comparison on clock drift estimation error for phase I.

100 simulations, the mean altitude error for the EKF was 1.79 m, compared to 0.85 m for the UKF and 0.58 m for the PF, thereby showing that the PF outperforms the Kalman filters on position estimation. In terms of velocity, it was found that the EKF has a mean error of 0.0085 m/s, compared to 0.0079 m/s for the UKF and 0.0140 m/s for the PF. Note that by tuning the filters, accuracy on a particular state variable may be increased, whereas the accuracy on a different variable may be decreased. The position of the vehicle was prioritized over the velocity, therefore the estimation of the PF seems worse for the velocity, however, it can be tuned in such a way that this performance is increased, which simultaneously decreases accuracy on the altitude estimation.

It can also be stated that the covariance lines for EKF and UKF are similar, as they are estimated in every measurement update step as well. The covariance lines for the PF are computed by estimating the variance of the particle distribution and thus has a different origin than the covariance for the Kalman filters. This can be seen in Figures 9.1a and 9.1b, where the covariance lines for the PF are also influenced by noise and have a higher magnitude. It was expected that the EKF would perform less accurate. This may be explained by the fact that in phase I, the non-gravitational accelerations are small with respect to the gravitational acceleration, thereby reducing the nonlinearity of the system dynamics. However, as will be shown in Section 9.3, when the nonlinearity of the system increases due to higher non-gravitational accelerations, the performance of the EKF will be less.

In Figures 9.3a and 9.3b the GPS clock error estimations are shown, which are similar to the altitude and velocity estimations. From multiple simulations clock bias estimation errors of 1.63 m for the EKF, 0.93 m for the UKF and 1.11 m for the PF were found. In terms of clock drift, estimations of 0.0072 m/s for the EKF, 0.0104 m/s for the UKF, and 0.0136 m/s for the PF were computed.

In Figure 9.4 the estimations of the angle of attack, angle of sideslip, and bank angle are shown. From multiple simulations, the EKF errors are computed to be 0.012°, 0.022°, and 0.009° for the angle of attack, angle of sideslip and bank angle respectively. The UKF provides estimation errors of 0.015° for α, 0.026° for β, and 0.013° for σ, whereas the errors of 0.020°, 0.050°, and 0.011° are found for the PF, for angle of attack, angle of sideslip, and bank angle respectively. All attitude angle converge, and it is seen that the behavior is very similar for all filters.

Previously it was shown that the position and velocity estimation process was successful, however, to illustrate the accelerometer errors, the scaling factor over the x_B-axis and the bias over the z_B-axis in Figures 9.5a and 9.5b were plotted, together with their covariance lines. In terms of the scaling factor, it was seen in Figure 8.1a that the acceleration over the x-axis increases over this part in the trajectory, and, as expected, the filters converge only after a longer time than for the other state variables that were seen in this section, thereby proving that the scaling factor can only be estimated for higher accelerations. The accelerometer bias then first shows converging behavior for all filters, which is in the part of phase I where the accelerations are small. When these accelerations increase, the covariance lines of the EKF and PF show that the filters diverge, albeit that the z-bias estimation is still done accurately during this phase. The covariance lines of the UKF appear to remain constant after a few seconds, however, it is found that also these lines very slowly diverge, with similar behavior as the covariance lines of the UKF.

The results in this paragraph show that all filters can be used for phase I of the re-entry flight. However, this is deemed the least interesting phase of the three, as the blackout phase is still to be covered, which will be discussed in Section 9.2. The final values for the state variables of all filters are saved to initialization files, which are then fed as an input to the phase II MATLAB files. To research the spread of these final values, in

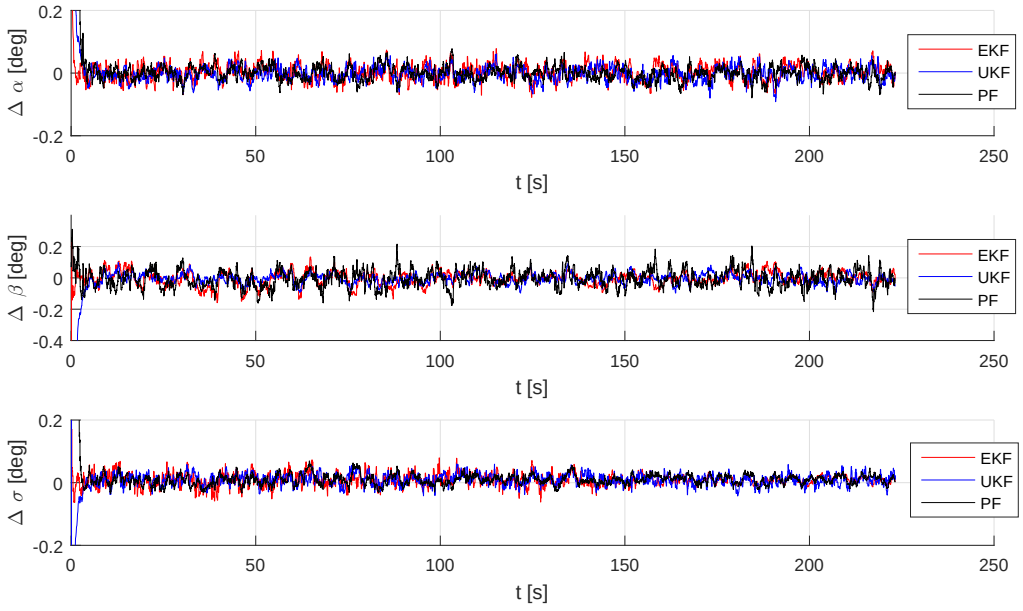
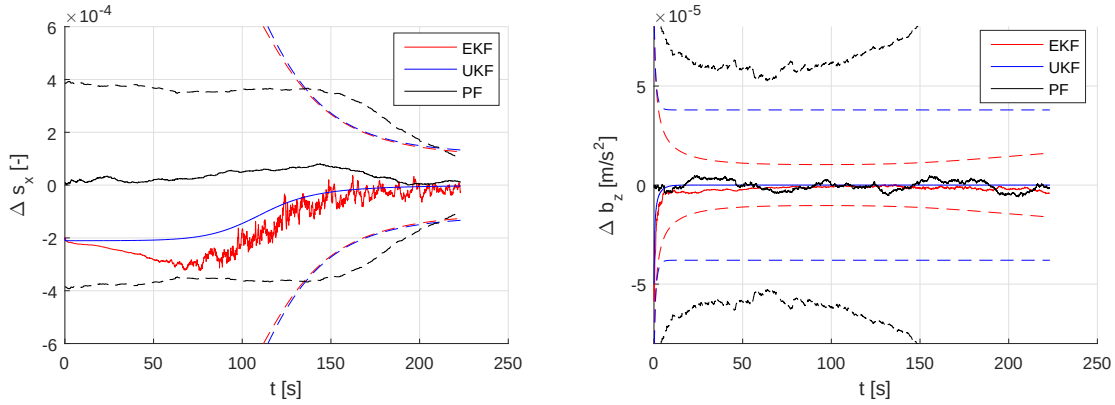


Figure 9.4: Comparison on aerodynamic-angle errors for phase I.



(a) (b)

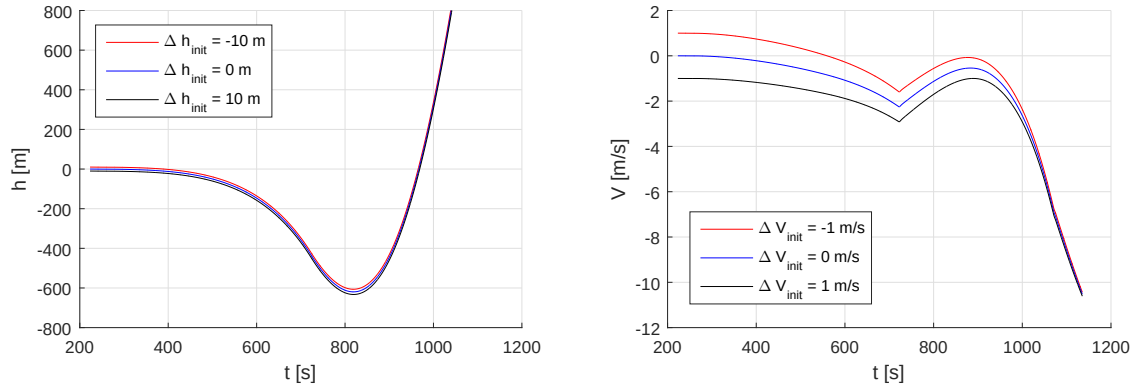
Figure 9.5: (**a**) - Comparison on scaling factor s_x estimation error for phase I. (**b**) - Comparison on accelerometer bias b_z estimation error for phase I.

Table 9.1 the mean and standard deviation is provided for the three filters for all three phases, where phase III is divided into a part with the FADS and a part without the FADS. As stated, the outcome for all estimators is dependent on random numbers, which causes the results to differ when no seed is applied. By taking 100 simulations, an approximation of the mean final error and standard deviation for all state variables can be made, which aids in the filter trade-off described in Section 9.4.

9.2 Phase II: Blackout Phase

Earlier in this report, it was described that from 82 km to 42 km, the vehicle goes through the blackout phase, which was based on post-processed results of the STS-118 flight, described by Goodman (2008). In this phase, no GPS measurements can be used, and as the altitude is above 40 km, no FADS measurements are available either.

Using only internal measurements, the vehicle's navigation system should be able to estimate part of the state using the IMU propagation that was described in Section 8.1.1. Mooij and Chu (2002) describe a similar



(a) (b)

Figure 9.6: (**a**) - Phase II altitude estimation error for different input values. (**b**) - Phase II velocity estimation error for different input values.

procedure and after application, it is seen that the final error of this state is not crucial for the mission, as it is relatively easily compensated for by the reactivation of the filter in phase III. As the state estimation during the blackout phase is purely based on IMU propagation, the results are the same for all filters. During this part of the trajectory the following variables are of importance:

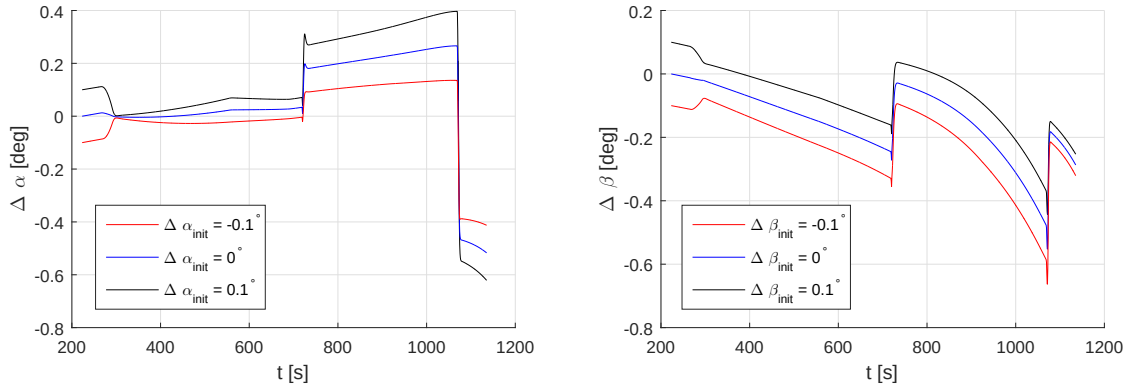
- Altitude h
- Velocity norm with respect to the atmosphere V
- Aerodynamic angles α, β, and σ

The IMU biases were successfully estimated during phase I, and the final values at the end of that phase are treated as a constant for phase II. In terms of the GPS errors, the same can be stated for the clock drift. The clock bias however shows a linear increase or decrease, as this variable is computed by the integration of the clock drift over time. As the state estimation behavior is exactly the same for all filters, the output of phase II mainly depends on the input, which is the final state estimation of phase I. In this section, the influence of the input error is evaluated for the aforementioned state variables of interest.

In Figures 9.6a and 9.6b the propagation of the altitude and velocity are shown, and it is seen that the initial error has a negligible influence on the results. In terms of altitude, it is seen that for the controlled flight, the state estimation becomes inaccurate after $t = 400$ s. However, an initial offset of 10 m results in 19.5 m extra output error, which is, when looking at the magnitude of the state estimation error, a relatively small difference. Also in terms of velocity, the initial error has almost no influence on the output of phase II. For an initial offset of 1 m/s, a final estimation error difference of only 0.087 m/s is found, which is, again looking at the magnitude of the nominal velocity state estimation, negligible. It is noted that the errors tend to get large during the blackout phase, however, by reactivating the filters in phase III, these estimation errors can be compensated for easily, as will be seen in Section 9.3.

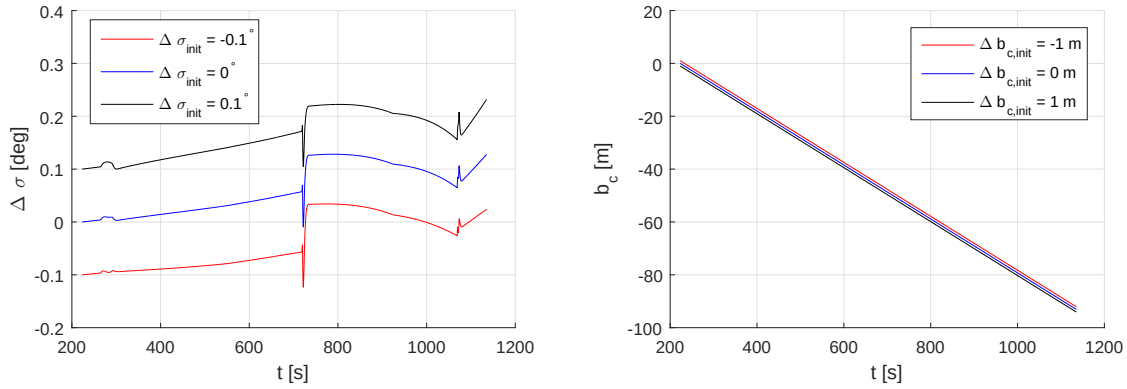
Figures 9.7a, 9.7b and 9.8a then show the behavior for the angle of attack, angle of sideslip, and bank angle respectively, for initial errors of $-0.1°$, $0°$, and $0.1°$. It is seen that the estimation-error behavior is almost independent of the input error that is applied. For the angle of attack and angle of sideslip, the activation of the guidance system at $t \approx 265$ decreases the estimation-error offsets with respect to the nominal error behavior. For the bank angle, the behavior of the estimation error is almost fully independent of the initial offset. In terms of performance, the angle of attack maps an initial error of $0.1°$ to $0.104°$, however, as seen in Figure 9.7a, the magnitude and sign of this mapping vary during flight. The magnitude of the angle-of-sideslip error decreases at the end of the blackout phase, where an initial offset of $0.1°$ is mapped to $0.034°$. As discussed, the bank angle remains almost independent of the initial error, and a $0.104°$ error is found for initial offset of $0.1°$.

At $t \approx 720$ s and $t \approx 1070$ s, the bank reversals are executed, which can be seen in the state estimation of the aerodynamic angles. It can be said that the navigation module has a small delay, which causes the peaks occurring at these points. These delays can be explained by the fact that the IMU only measures with a finite sampling frequency. During the bank reversals, a high bank angle rate is found, and due to the delays that were described, the estimation error during these reversals tends to get large. In real life, a guidance system



(a) (b)

Figure 9.7: (**a**) - Phase II angle-of-attack estimation error for different input values. (**b**) - Phase II angle-of-sideslip estimation error for different input values.



(a) (b)

Figure 9.8: (**a**) - Phase II bank-angle estimation error for different input values. (**b**) - Phase II clock bias estimation error for different input values.

would compensate for the extremely high bank reversal rates, and, depending on the performance of the guidance system, the peaks that are found in the state estimation would be decreased. Furthermore, in the aforementioned figures, it is seen that after a bank reversal, the error 'jumps', after which the behavior is slightly different. These jumps can be explained by the fact that the state estimation is done using quaterions, which can be directly transformed to roll, pitch, and yaw angles, following equations from Appendix A.2. Then, using a nonlinear transformation, the aerodynamic angles can be computed (using equations from Appendix A.1). A small change in quaternion behavior is found after a bank reversal is performed, which leads, due to the nonlinear transformations, to the 'jumps' that are seen in Figures 9.7a, 9.7b and 9.8a.

Finally, the clock bias is briefly estimated, which was said to experience linear behavior. In Figure 9.8b it can be seen that this is true, and by looking at numerical solutions, it is found that an initial offset of 1 m is also mapped to 1 m output error. Note that in this section, the errors that are described are compared to the nominal error behavior during the blackout phase, which is done to evaluate the influence of an initial offset, *i.e.*, a non-ideal state estimation during phase I. As seen in the aforementioned figures, the actual final error of the state variables diverges, which again defines the need for a filter.

9.3 Phase III: From Blackout Phase to TAEM Interface

At the end of the blackout phase, the estimation errors have drifted a lot since the results from phase I, as was seen in the previous section. When the altitude of 42 km is reached, phase III is initiated, using the exact final state estimation of phase II as an input for the EKF and UKF. For the PF particle clouds are created that are centered around this final estimation. It is chosen not to keep the particle clouds from phase I, as particles that

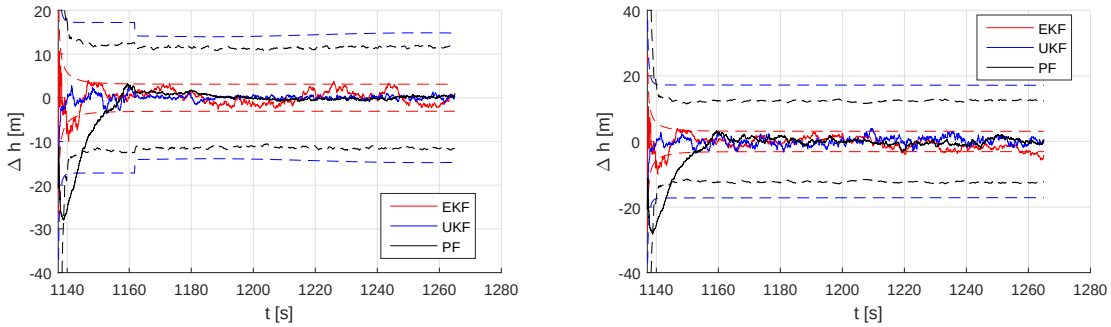


Figure 9.9: (**a**) - Comparison on altitude estimation error for phase III using IMU, GPS, and FADS. (**b**) - Comparison on altitude estimation error for phase III, using only IMU and GPS.

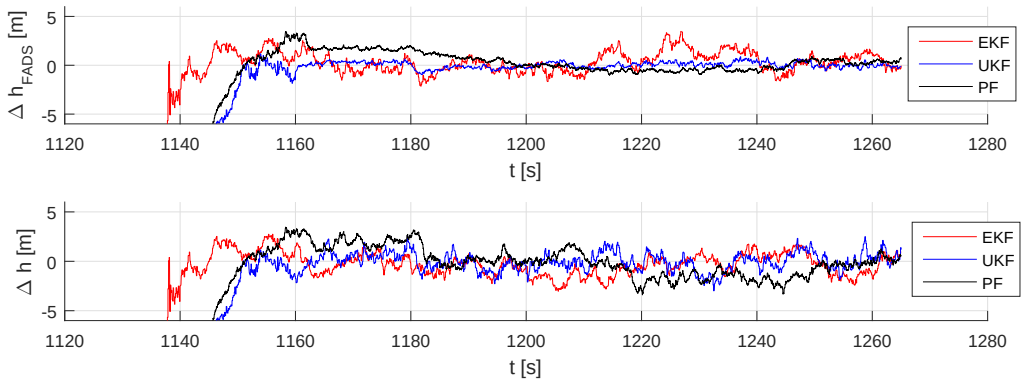


Figure 9.10: Comparison on altitude estimation error for phase III, for an active FADS (upper plot) and inactive FADS (lower plot).

showed high likelihood during this phase may be drifted too far from the true state. As the final steps of the PF in phase I contained mostly a clustered particle cloud around the true state, this cloud may be inefficient for phase III, as the particle drift during phase II may have changed this cloud in such a way that no particles are close to the truth any more. By setting a new cloud, similarly to the method that was handled for phase I, a fast(er) convergence of the filter can be realized. For the case where the clouds from phase I are kept, no convergence may be achieved at all.

Phase III is defined from the end of the blackout phase to the TAEM interface, but due to the fact that the provided guidance system only provides commands up until $t = 1265$ s, it is chosen to have phase III also end at this time. This leads to the fact that phase III is the shortest part of the trajectory, and can be deemed the most important, as the final state estimation should be sufficient for the vehicle to successfully find its way to the target cylinder of the TAEM interface. However, in this part of the trajectory GPS signals are available again and, under an altitude of 40 km (Ellsworth and Whitmore, 2007) the FADS becomes active as well. This section contains the results for the three estimators with and without the FADS, so that research can be done with respect to the addition of this sensor to the state estimation process. For a fair comparison, the initial state of phase III is taken to be the same for all filters.

First, the altitude will be evaluated, which is directly related to the input of the control system and thus its estimation accuracy is of great importance. Figures 9.9a and 9.9b show the altitude error profiles for phase III, together with the covariance lines, and for both nonlinear filters, the UKF and PF, the use of the FADS can clearly be seen in these covariance lines. Around $t \approx 1161$ s, extra information is introduced in the form of FADS measurements, which realizes a more accurate error estimation. Then, when a close-up is made, Figure 9.10 is obtained, where the effect of the FADS on the UKF and PF can be seen. From 100 simlulations, the EKF final altitude estimate decreases from 1.44 m to 1.13 m when activating the FADS, however, the improvement in performance is difficult to see in Figure 9.10. For the UKF, a reduction from 0.66 m to 0.19 m is found, whereas the PF experiences a performance increase as the altitude error estimate decreases from 0.93 m to 0.62 when

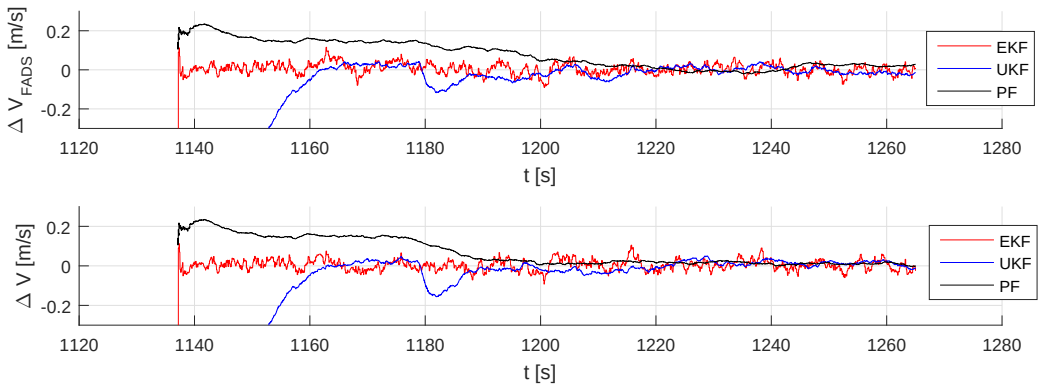


Figure 9.11: Comparison on velocity estimation error for phase III, for an active FADS (upper plot) and inactive FADS (lower plot).

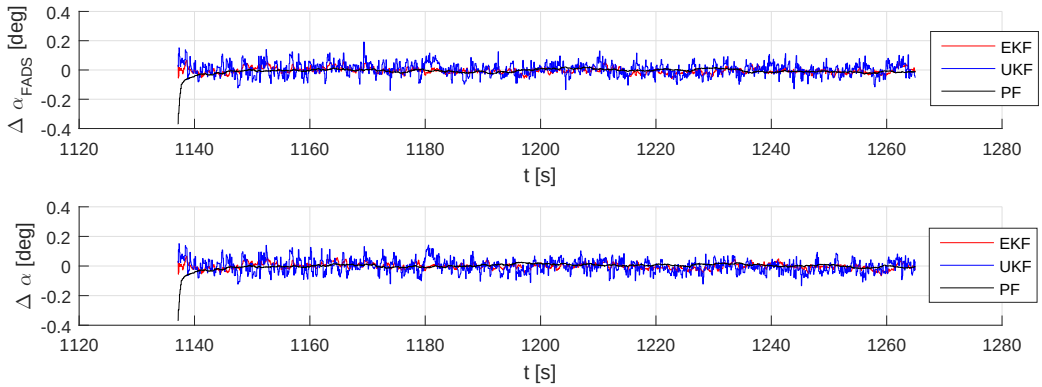


Figure 9.12: Comparison on angle-of-attack estimation error for phase III, for an active FADS (upper plot) and inactive FADS (lower plot).

activating the FADS.

For the EKF, the filter convergence was limited more and more by narrowing the covariance band, which explains why the FADS activation cannot clearly be seen in the covariance lines and in the close-up figure. The narrowing of this band was done to increase the accuracy of the EKF results. The altitude results for phase III show that for a highly nonlinear phase (as the effects of aerodynamics are large with respect to gravitational effects), the UKF and PF outperform the EKF significantly.

Then, the velocity is evaluated, which is shown in Figure 9.11. This state variable also converges, however, to clearly show the error behavior, the covariance lines are not shown here. For the velocity-error profile, it is seen that the nonlinear filters have a longer settling time, however, they are less noisy than the EKF. FADS activation did not show any significant improvement, which can be explained by the fact that the pressure measurements are strongly dependent on the altitude of the vehicle, and much less on the velocity. EKF showed a final velocity estimation error of 0.016 both with and without the FADS being active. For the UKF, 0.008 m/s was found wihtout the FADS and 0.010 m/s was found with the FADS. From this, it cannot be stated that activation of the FADS has a negative effect on velocity estimation; due to the fact that the altitude estimation process was prioritized, the filters were tuned for an optimal performance for position estimation. Apparently, during the tuning process of the UKF, the accuracy of the velocity estimate was slightly reduced. Finally, the PF experienced final errors of 0.021 m/s, both for an inactive and an active FADS.

Then, in Figures 9.12 to 9.14 the behavior of the attitude-angle error estimates is shown. From these figures, it directly becomes clear that the UKF induces a relatively large error during the bank reversal at $t \approx 1180$ s, for both the angle of sideslip and the bank angle. In the velocity plots, these peaks were also found for the UKF and are induced by the attitude estimation. It was found that when the noise components of the respective angles were increased for both the UKF and PF, these peaks were eliminated, however, the overall results of the attitude estimation became more noisy as well, which basically means that this error was not concentrated at the bank reversal point, but was spread across the whole phase.

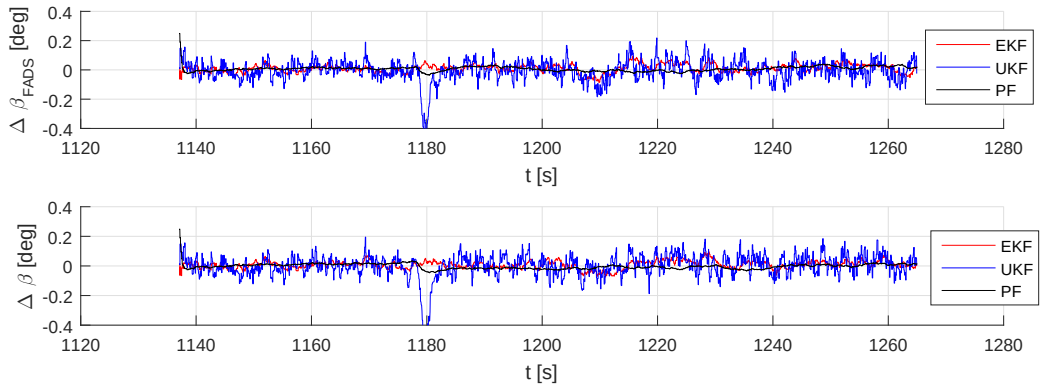


Figure 9.13: Comparison on angle-of-sideslip estimation error for phase III, for an active FADS (upper plot) and inactive FADS (lower plot).

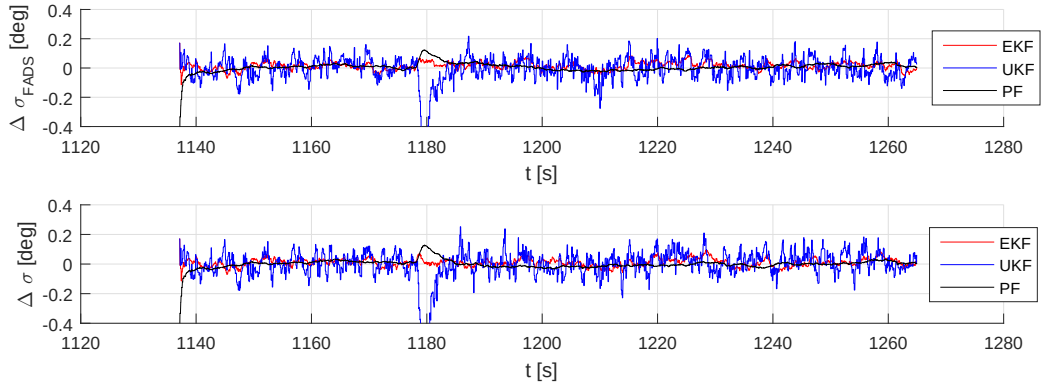


Figure 9.14: Comparison on bank-angle estimation error for phase III, for an active FADS (upper plot) and inactive FADS (lower plot).

For the results that are shown in these figures, application of the FADS showed negligible effect on the final attitude-estimation error. For the EKF, attitude errors between $0.010° - 0.020°$ were found, whereas for the UKF they had a magnitude between $0.020° - 0.065°$. Finally, the PF showed results between $0.005° - 0.020°$. These results were found after performing 100 simulations, and the errors per attitude angle are summarized in **??**.

The original bank reversals were defined such that the reversal was performed in one second, thereby leading bank-angle rates up to $120°/s$, which is not realizable. These results, however, contain restricted bank angle rates of $15°/s$. These restrictions are normally coupled with the guidance system. In the EKF, a linear model of the attitude estimation is used from a point where the attitude state was estimated well (directly before the bank reversal is executed). Using this linear model, the also linear bank reversal is approximated well for that one point, thereby showing no peak. For the UKF, the approximation during the bank reversals are executed using the nonlinear models, with a few sigma points for every state. The middle sigma point represents the state prediction and thus includes an error, whereas the two outer sigma points include an additional error. Due to the nonlinear conversion of the attitude sigma points into the carrier phase difference estimations, a different error is introduced for the different sigma points, resulting in a mean that differs from the middle sigma point, *i.e.*, a larger estimation error. Following this line of thought, the PF, which estimates the state using multiple points as well, also experiences small peaks for angle of sideslip and bank angle. Only, due to the large number of particles that is used compared to the sigma points in the UKF, the effect of this peak is much less. By increasing the noise components for the UKF, the $\mathbf{Q}$-matrix becomes more significant in the computation of the sigma points, and apparently reduces the 'displacement' of the mean, thereby reducing the peak, but inherently increases the noise in the estimation results.

To avoid the aforementioned peaks, two solutions can be defined. First, the speed of the bank reversals can be limited (which is normally done using a guidance system). Due to this, the change in measured carrier-phase differences will be smaller for each time step, thereby reducing the error that is formed by the nonlinear mapping of the attitude angles. Next to that, an increase in measurements could increase the accuracy of the

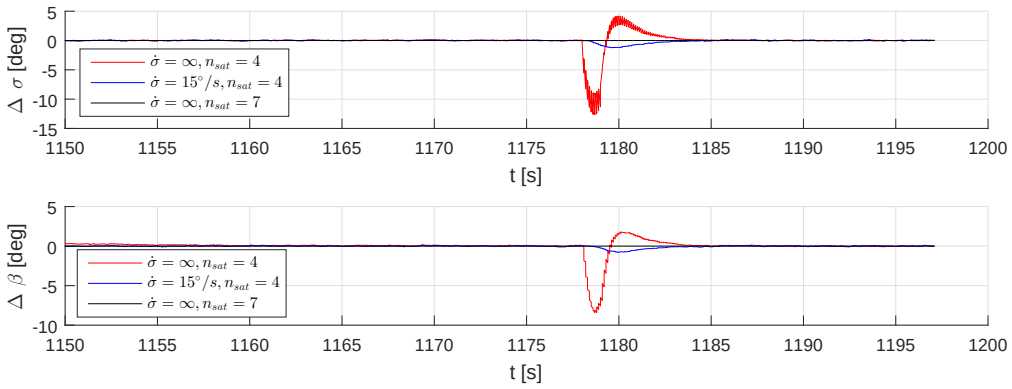


Figure 9.15: Bank-angle estimation error (upper plot) and angle-of-sideslip estimation error (lower plot) for different bank-reversal rates and number of used GPS satellites.

estimation process of the attitude. This can be done by using more than the minimum required number of GPS satellites. Using the implementation of the GPS receiver from Section 8.1.2, it was found that for phase III, the minimum number of visible satellites is seven at all times. Using these possible solutions, three situations are evaluated, which are shown in Figure 9.15.

This figure shows the UKF estimates of the bank angle and the angle of sideslip, as the peaks had the largest magnitude for this filter. First, the initial case is evaluated, where the bank reversals were performed in one time step, leading to unrestricted bank-angle rates, in this case up to 120°/s. However, when decreasing the time step of the control history, this value is increased, and therefore it is denoted by the ∞-sign in Figure 9.15. For both angles it is seen that a high peak is encountered, which is very noisy, and has a maximum bank-angle error of more than 10°. Then, by restricting the bank-angle rate to 15°/s, the results are obtained, which were also shown in Figures 9.13 and 9.14 and where smaller peaks are observed, with maximum bank-angle error of approximately 0.5°, hence, it can be concluded that reducing the bank-angle rates also reduces the magnitude of the peaks. Then, the number of GPS satellites was increased from four to seven, thereby increasing the number of carrier-phase measurements from 12 to 21 (as each satellite provided three measurements). For the unrestricted bank angle rates, the magnitude of the peaks was reduced to $\sim 0.001°$, thereby proving that an increase in number of measurements severely reduces the magnitude of the peaks. The application of the number of satellites will be further evaluated in Section 9.4.

For the results of the GPS and IMU errors, no difference was made between the systems with and without FADS, as the FADS has no direct effect on these variables. In Figure 9.16 the results are shown for the clock bias, clock drift, and scaling factor s_x. The former two represent the performance of the filters on the GPS errors, and it is seen that the filters converge for these variables.

Finally, the scaling factor s_x is evaluated. The covariance lines in Figure 9.16 all show similar behavior as for the UKF, which means that all filters diverge for the estimation of this state variable. In a close-up figure in Figure 9.17 this can be confirmed, as the initial error is very close to zero and the estimates drift away from this value. However, from $t \approx 1250$ onwards, the estimates seem to stop drifting. In Figure 8.1a it was seen that at the end of the trajectory, the deceleration over the x_B axis is rather small, which explains why the filters diverge for this phase. Earlier it was seen that convergence was achieved for this variable, which was found in a phase where the deceleration over the x_B axis was larger. From this, it is concluded that the scaling factors indeed can be estimated accurately if the corresponding acceleration is sufficiently large.

In terms of accelerometer bias and gyroscope drift, it can be stated that the accelerations are too high to accurately estimate b_x, b_y, and b_z, as the filters diverge for these variables in this phase. Therefore it is chosen to use the best estimate from phase I (where the biases were estimated accurately) for these variables. For the gyroscope drifts, the filters converge when the rotational rates remain small. Figure 6.8 showed that after the first bank reversal, the magnitude of the bank angle slowly decreases, whereas the angle of attack starts decreasing after $t \approx 900$. Using this, it is concluded that the smallest rotational rates are measured in phase I, and therefore the gyroscope drifts can best be estimated during that phase, hence, the best estimates of the gyroscope drifts from phase I are used in phase III.

At this point, it was seen that all filters show acceptable performance for phase III of the re-entry flight, with the PF being the most accurate. The implementation of the FADS system aided in better altitude and

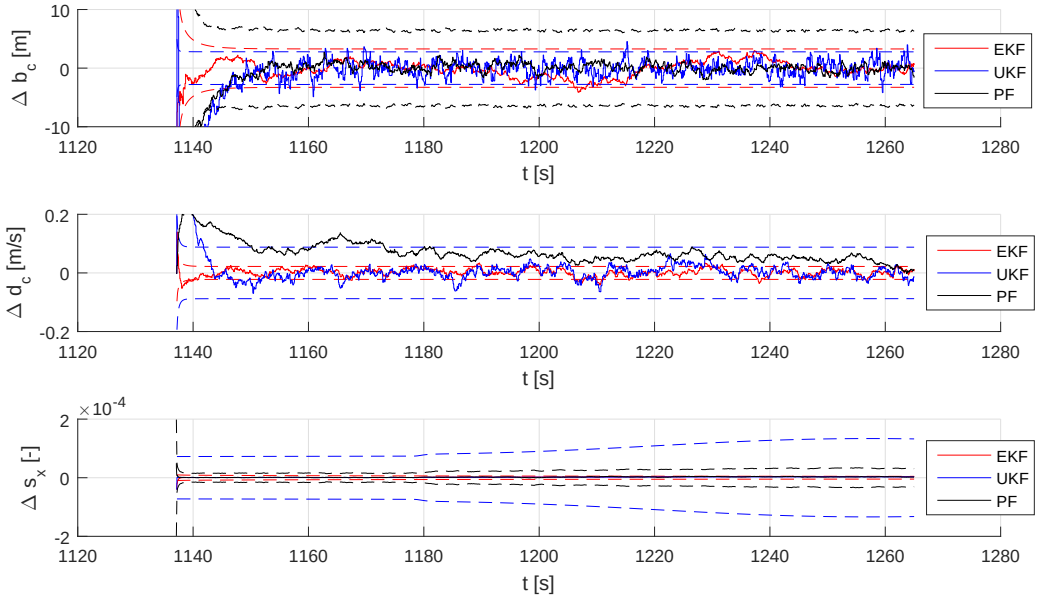


Figure 9.16: Comparison on estimation errors of clock bias (upper plot), clock drift (middle plot), and scaling factor over the x_B-axis (lower plot) for phase III.

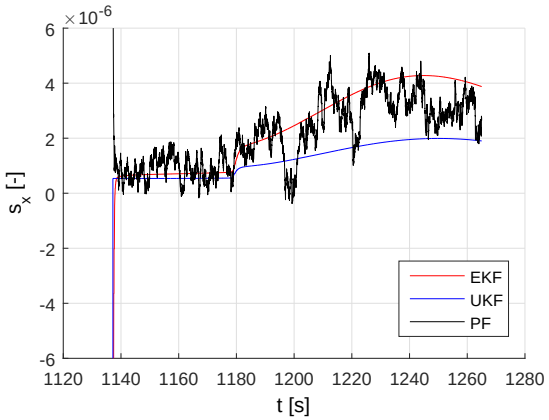
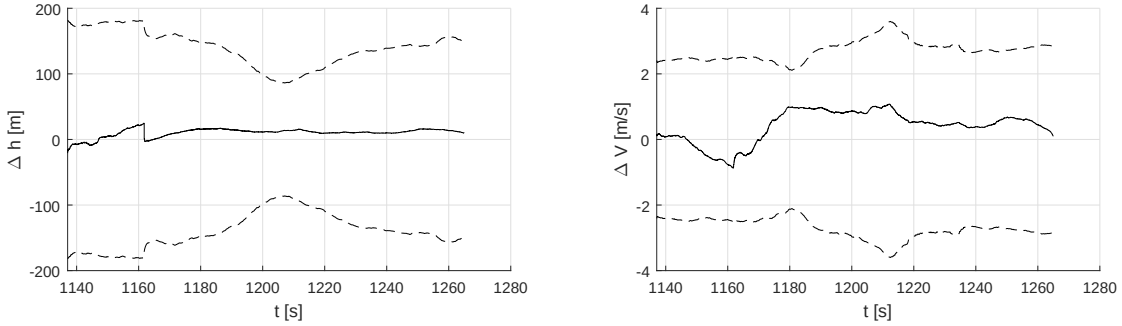


Figure 9.17: Close-up of the estimation-error behavior of the scaling factor over the x_B-axis for phase III.

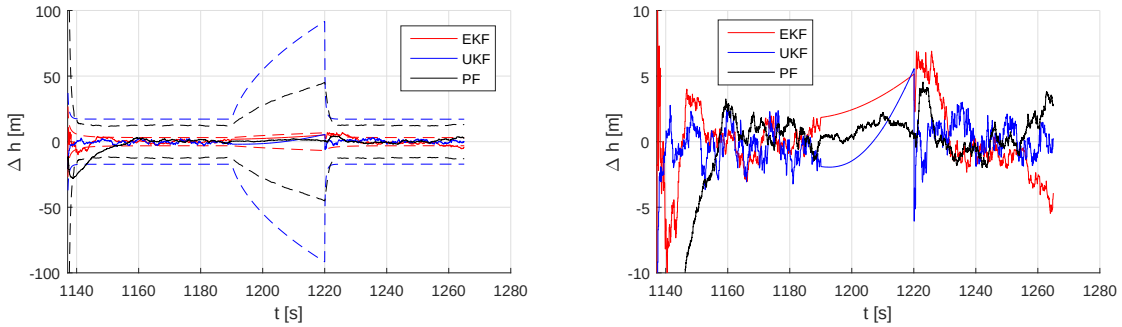
velocity estimations, although the effect on the former variable was much more significant. It was explained in Section 5.1.1 and Section 8.1.3 that the FADS produces pressure measurements, which are strongly dependent on the altitude of the vehicle, due to their relation in the atmospheric model. The dependency of the measured pressures on the velocity and attitude are negligible, hence the addition of the FADS to the filters was only clearly visible in the altitude estimation results. The atmospheric model is only defined in radial direction and therefore is independent of latitude and longitude in the simulator. Due to this, applying only the FADS and IMU to the estimation process creates a filter that does not converge, as is seen in Figures 9.18a and 9.18b. At $t \approx 1161$ s the FADS is activated, resulting in a jump in the altitude profile and covariance lines. For the velocity, no significant difference can be seen again, which was explained earlier in this paragraph. In both figures it is seen that the filter does not converge, and it was found that the filter is unstable, as the final altitude results may vary between a few meters and a few kilometers.

Using these results, it can be concluded that the FADS is incapable of performing the estimation process as a stand-alone external measurement device. However, earlier in this section is was seen that the FADS does function as a useful addition to the INS/GPS integrated system, increasing the accuracy of mostly the altitude measurements. Next to that, it was discussed that in case of, for example, carrier-phase measurement failure,



(a) (b)

Figure 9.18: (**a**) - PF estimation of altitude using only FADS. (**b**) - PF estimation of velocity norm using only FADS.



(a) (b)

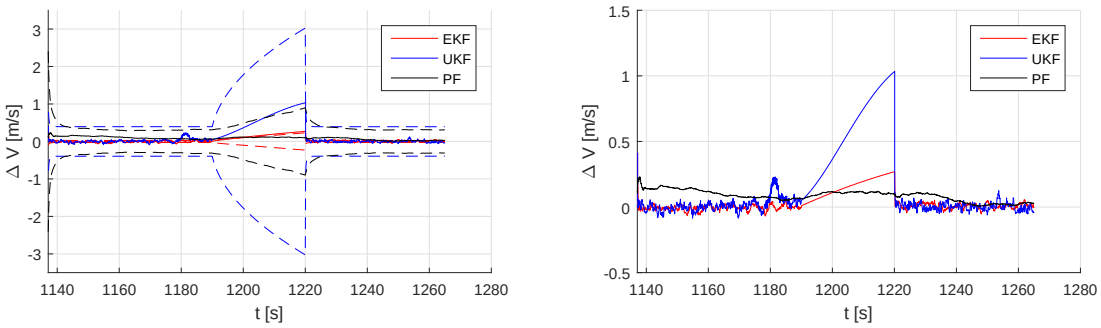
Figure 9.19: (**a**) - Comparison on altitude estimation error for phase III with GPS outage between $t = 1190$ s and $t = 1220$ s. (**b**) - Comparison on altitude estimation error for phase III with GPS outage between $t = 1190$ s and $t = 1220$ s, zoomed in.

the built-in FADS estimator that was implemented in the sensing system itself could serve as a contingency. The position and velocity estimation however cannot be done without GPS in this case.

To illustrate the system performance on a sudden GPS outage, the incoming GPS signals are deactivated from $t = 1190$ s to $t = 1220$ s. This is done to evaluate how well the filters respond to failure of the GPS receiver for a limited amount of time. For a full GPS receiver failure, similar results are obtained as for the system where no GPS was applied at all. Figure 9.19a shows the altitude estimation, together with the covariance lines, and it can be seen that the 1-σ band expands drastically during the GPS outage, with the increase of the band being proportional to the covariance before this step, therefore the EKF covariance expands much less than for the other two filters. In Figure 9.19b a close-up is shown where it is seen that during the GPS outage, the filters diverge as they are driven by IMU and FADS measurements. After the GPS outage, the filter converges again to errors less than 5 m, albeit with some settling time, especially for PF. For the PF, it is found that the particle propagation is based on FADS and IMU measurements for 30 seconds, thereby probably deleting particles which show high likelihood with the (absent) GPS measurements. Due to this, the filter requires some time to converge. The EKF and UKF directly fall back to the accuracy that was also seen earlier in this section.

Finally, it is seen that during a GPS outage of only 30 seconds, the altitude error drifts up to roughly 5 m for the Kalman filters, where it remains close to 1 m for the PF. Specifically the UKF shows diverging behavior during this phase, and for longer GPS outages the error will grow swiftly. However, as was seen earlier, a large error can be compensated for when the GPS outage is over, which is similar to the reactivation of the filters after phase II. The altitude error thus can be reduced when GPS signals are available again, however, the closer the vehicle is to the TAEM interface, the more crucial it gets that the state estimation is performed accurately, and thus a late GPS outage may corrupt the mission's success.

For the behavior of the velocity profile, similar results are experienced. The 1-σ covariance band expands during the GPS outage, as seen in Figure 9.20a. The velocity norm estimation from Figure 9.20b also shows that the IMU/FADS system is incapable of accurately estimating the velocity during GPS outage. The results for the PF however seem reasonable, but this varies depending on the seed that is used for the random number



(a) (b)

Figure 9.20: (**a**) - Comparison on velocity norm estimation error for phase III with GPS outage between $t = 1190$ s and $t = 1220$ s. (**b**) - Comparison on velocity norm estimation error for phase III with GPS outage between $t = 1190$ s and $t = 1220$ s, zoomed in.

generator (all simulations in this chapter are done with `seed = 0`). After the GPS is reactivated, also the velocity norm converges directly, towards final errors less than 0.1 m/s. In terms of attitude, the errors drift less during the GPS outage, and for a 30 second window, their estimates remain smaller than 0.02°. The clock drift is again treated as a constant for this period, whereas the clock bias shows linear behavior, due to the integration of the clock drift. Finally, the IMU error estimation process is continued as the filters are still activated and IMU measurements are available.

In this section it was seen that the FADS increases the navigation module performance, especially for altitude estimations. All three filters have an acceptable performance, however, it can be stated that the EKF is not able to achieve the same accuracy as the nonlinear filters. It was also seen that the FADS cannot be used as a stand-alone external measurement device, which provides the need for GPS in the estimation process. However, after a sudden GPS outage, the system is able to recover from the drift of the state variables.

9.4 Simulation Results of the Integrated GNC System

At this point, the simulation results of the three filters were discussed for phase I, II, and III, where in the latter the FADS was applied to research the effect of this sensor on the state estimation. In Sections 9.1 to 9.3 the simulation results however were shown for one seed, and due to the dependency on the random number generators, different seeds will produce different results, albeit that the filters converge for all seeds. To research the actual accuracy of these estimators, they are subject to a larger number of simulations, where, as an outcome, the mean and standard deviation of the final state errors are computed. Using the results from these computations and the conclusions that were drawn in the previous three sections, a trade-off can be made between the EKF, UKF, and PF, which will be described in this section. Then, when one filter is chosen, it is integrated in the simulator and coupled to the control system and guidance profile, which concludes the implementation of the integrated GNC-system.

As discussed, first the estimators are evaluated for a large number of simulations, to investigate the spread of the results. For doing this, phase I, phase III without FADS, and phase III with FADS are evaluated on the final state estimation. It was chosen to use 100 simulations for each case, as this was proven to be sufficient for an accurate estimation of the mean state error. The outcome of these computations is summarized in Table 9.1, where the best estimates of the position, velocity, and attitude are highlighted, as they are prioritized over the sensor errors. Phase II was not evaluated, as the state estimation was purely done by IMU propagation, which was the same for all filters. Evaluation of this phase would therefore not add anything to the performance evaluation of the filters.

First, phase I is discussed, of which the simulation results are shown in columns two to four of Table 9.1. In the leftmost column the mean μ and standard deviation σ for some state variables are defined, where the subscript denotes the corresponding variable. In terms of altitude, it is seen that the PF has the smallest mean error, but its spread is somewhat larger than for the UKF. The EKF appeared to have a good performance, but significantly worse than the nonlinear filters, as was expected. For the velocity norm, the EKF performs similar to the UKF, whereas the PF has a somewhat poorer performance due to the fact that position estimation was prioritized. The second block contains the attitude, and it is seen that all filters are able to estimate the aerodynamic angles well, with the EKF even performing better than the UKF and PF. Then, looking further, the

Table 9.1: Mean and standard deviation of the final state error estimations of phase I and phase II with and without the FADS being activated, computed over 100 simulations.

	Phase I			Phase III (IMU/GPS/FADS)			Phase III (IMU/GPS)		
	EKF	UKF	PF	EKF	UKF	PF	EKF	UKF	PF
μ_h [m]	1.78	0.85	**0.58**	1.13	**0.19**	0.62	1.44	**0.66**	0.93
σ_h [m]	0.93	0.37	0.45	0.92	0.16	0.27	1.12	0.49	0.71
μ_V [m/s]	0.009	**0.008**	0.014	0.016	**0.010**	0.021	0.016	**0.008**	0.022
σ_V [m/s]	0.006	0.006	0.011	0.013	0.008	0.020	0.013	0.007	0.015
μ_α [deg]	**0.012**	0.015	0.020	0.010	0.021	**0.007**	0.011	0.027	**0.007**
σ_α [deg]	0.010	0.011	0.012	0.007	0.016	0.005	0.008	0.019	0.005
μ_β [deg]	**0.022**	0.026	0.050	0.017	0.036	**0.011**	0.019	0.038	**0.010**
σ_β [deg]	0.016	0.019	0.030	0.013	0.026	0.008	0.013	0.031	0.008
μ_σ [deg]	**0.009**	0.013	0.011	0.020	0.056	**0.015**	0.018	0.064	**0.016**
σ_σ [deg]	0.006	0.009	0.007	0.013	0.040	0.011	0.014	0.046	0.012
μ_{b_c} [m]	1.63	0.93	1.11	0.92	1.17	0.59	0.88	1.16	0.63
σ_{b_c} [m]	0.70	0.41	0.41	0.67	0.87	0.45	0.63	0.83	0.44
μ_{d_c} [m/s]	0.007	0.010	0.014	0.011	0.010	0.025	0.009	0.010	0.022
σ_{d_c} [m/s]	0.004	0.005	0.009	0.007	0.008	0.072	0.006	0.007	0.015
μ_{s_x} [-]	$3.8\cdot10^{-5}$	$3.5\cdot10^{-6}$	$1.3\cdot10^{-5}$	$3.9\cdot10^{-6}$	$2.9\cdot10^{-6}$	$3.3\cdot10^{-6}$	$3.9\cdot10^{-6}$	$2.9\cdot10^{-6}$	$3.3\cdot10^{-6}$
σ_{s_x} [-]	$2.0\cdot10^{-5}$	$<10^{-10}$	$9.6\cdot10^{-6}$	$<10^{-10}$	$<10^{-10}$	$1.5\cdot10^{-6}$	$<10^{-10}$	$<10^{-10}$	$1.5\cdot10^{-6}$
μ_{b_z} [m/s^2]	$3.2\cdot10^{-6}$	$<10^{-10}$	$2.2\cdot10^{-6}$	$2.0\cdot10^{-6}$	$6.2\cdot10^{-9}$	$5.5\cdot10^{-6}$	$2.0\cdot10^{-6}$	$6.2\cdot10^{-9}$	$5.5\cdot10^{-6}$
σ_{b_z} [m/s^2]	$5.6\cdot10^{-7}$	$<10^{-10}$	$1.6\cdot10^{-3}$	$<10^{-10}$	$<10^{-10}$	$3.8\cdot10^{-6}$	$<10^{-10}$	$<10^{-10}$	$3.8\cdot10^{-6}$
t_{sim} [s]	15.2	25.1	81.4	8.0	10.6	41.9	8.0	10.9	40.3
	1.00	1.65	5.34	1.00	1.32	5.21	1.00	1.36	5.02

clock errors are evaluated, and it is seen that the UKF and PF outperform the EKF on clock bias, whereas for the clock drift, the results are similar. In terms of the IMU errors, the accuracy of the PF results is significantly lower than for the EKF and UKF. However, overlooking all results of phase I, it can be concluded that all filters provide sufficient state estimation accuracy.

Lastly, the computational load of the filters can be discussed, which is shown in the last two rows of Table 9.1. The upper row states the mean time, whereas the lower row defines the relative time with respect to the EKF simulation. It is seen that for phase I, where the filters are active, the EKF has the lowest computational load. Due to the use of sigma points, this load is increased for the UKF to approximately 1.65 times larger than the EKF simulation time. As the PF uses $N = 1000$, the number of calculations that is to be done in the filter is also large, thereby leading to a large simulation time, which is approximately 5.34 times the EKF simulation time. However, as the real time of phase I is more than 200 s, it can be stated that all three filters can be used for the state estimation during this phase.

Table 9.1 then also displays the results from phase III, which was denoted as the most important phase in the state estimation process, as the error from the blackout phase is to be corrected by the navigation module. Comparing the velocity outcomes, the results are similar, albeit that the difference between UKF and EKF becomes clear as the UKF outperforms the EKF at this point. The main difference between the filters can be seen in the altitude results as the UKF is significantly more accurate than the PF, which in its turn is also significantly more accurate than the EKF. By comparing the FADS integrated system and the IMU/GPS system, it can be stated that especially for the altitude, the accuracy of the state estimation is increased when the FADS is activated, due to the high dependency of the FADS measurements on the altitude of the vehicle. From this it is concluded that the FADS adds significance to the state estimation process and will be included in the chosen filter.

The attitude estimation posed a slight problem for the UKF, as was seen in Section 9.3. The peaks that showed during the bank reversals can be corrected by addition of GPS satellites or restricting the bank-angle rate, however, the PF and EKF performance also improve due to these solutions, and thus the UKF will not perform as efficiently as the PF and EKF. It is found that the PF has the best performance with respect to the attitude of the vehicle. In terms of the GPS and IMU errors, it can be stated that all filters are capable of estimating these errors during flight. The resulting final errors are in the order of only a few percent of the calibrated ground value, and therefore it is chosen to only show the scaling factor and one bias. It can also be noted that the FADS measurements have a negligible influence on the GPS and IMU errors, as these are only dependent on their respective measurements. Only the clock bias and drift differ slightly, which is due to the

fact that in the estimation of these variables, position and velocity of the vehicle is also included.

Again, finally the simulation time is evaluated, and similar relative values are found as was seen in phase I in Table 9.1. From this, the same conclusion can be drawn; as the simulation time is significantly lower than the real flight time of phase III, this poses no restrictions on the trade-off of the filter.

Chapter 6 described that the estimated state of the navigation module was served as an input for the control system. This system was already integrated in the simulator, and its results were described in Chapter 7. However, in the research on the control system, it was stated that the true state of the vehicle was used as input, which is not the case in real life. By integrating the navigation module an estimated state is computed and fed to the control system, thereby introducing an additional error. The accuracy of the state estimation thus should be sufficient for the control system to compute and execute the commands that will make the vehicle fly according to the guidance system. In the simulator that was described in Chapter 6, the guidance system is simply comprised of an aerodynamic angle model and thus requires no input, which is different for real life applications where the guidance system may also require some state variables as an input. As in this thesis the navigation module only provides input for the control system, the variables of importance are limited to:

- **Mach number** M - this variable is used for the computation of the aerodynamic coefficients and derivatives. Following from the definition shown in Equation (3.12), it can be stated that the Mach number is dependent on the velocity norm and the speed of sound, which is directly linked to the altitude of the vehicle. The important state variables that are required for the Mach number thus are the position and velocity norms.
- **Dynamic pressure** $\overline{q}$ - the dynamic pressure is comprised of the velocity norm and the density of the atmosphere, thereby having a direct dependence on the altitude of the vehicle as well. This variable is required for the selection of the control gains and thus is to be computed accurately, thereby again selecting the position and velocity norms as important state variables.
- **Aerodynamic angles** α**,** β **and** σ - the control commands are computed using the difference between these aerodynamic angles and there commanded values from the guidance system. As the actual control forces and moments are directly related to these inputs, the attitude estimation is deemed as important as well.
- **Rotational velocities** p**,** q**, and** r - part of the control commands is computed using the rotational velocities that are found from the gyroscope measurements. This leads to an increased importance for the gyroscope drifts.

In this list, it is described that the altitude and velocity are highly prioritized, together with the attitude of the vehicle. From the results shown in Table 9.1 it was seen that the EKF has acceptable attitude and velocity estimation, but lacks the ability to approximate the altitude as accurate as the other filters. The UKF then shows the best velocity and altitude estimations and thus should be a better choice than the EKF, however, it was stated that the attitude estimation of the UKF is worse than for the PF and EKF. The PF then performs slightly worse than the UKF, and better than the EKF for phase III on altitude and velocity. It was seen that the PF provides an accurate attitude estimation, without the difficulty of having high estimation errors during required maneuvers such as bank reversals. The main disadvantage of this filter would be the computation time, however, as the values found in Table 9.1 do not approximate the actual flight time, this variable can be neglected. Based on the results presented in this section and the outcome of the simulations described in Sections 9.1 to 9.3, together with the conclusions that are drawn in this paragraph, it is chosen to integrate the PF in the simulator. Earlier it was also stated that the FADS will be implemented, next to the GPS and IMU, to achieve the highest state error estimation accuracy.

For the integration of the PF in the simulator, the measurements from the IMU, GPS and FADS were merged into one vector, containing zeros for when particular measurements were not available. This vector was fed to the navigation module, together with a matrix containing the GPS satellite positions and velocities. Then, the PF was initiated by setting the initial particle clouds with assigned weights and the implementation of Section 8.2.3 was done, using an `IF`-statement to determine whether the vehicle was in phase I, II, or III. For every time step, an output vector was constructed containing the state variables that were shown in Equation (8.13). To make this estimated state compatible with the input requirements of the control system, the Mach number and dynamic pressure were computed from the position and velocity components and the roll, pitch, and yaw angles were converted to angle of attack, angle of sideslip, and bank angle. Finally, the roll, pitch, and yaw rates were computed using the IMU gyroscope measurements corrected by the estimated gyroscope drifts b_p, b_q, and b_r. System verification was done by separately running the different phases, and

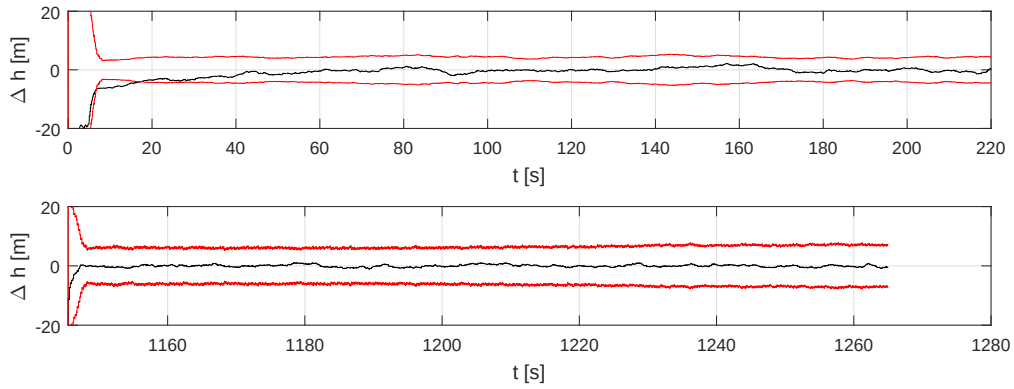


Figure 9.21: Altitude estimation of the integrated system for phase I (upper plot) and phase III (lower plot).

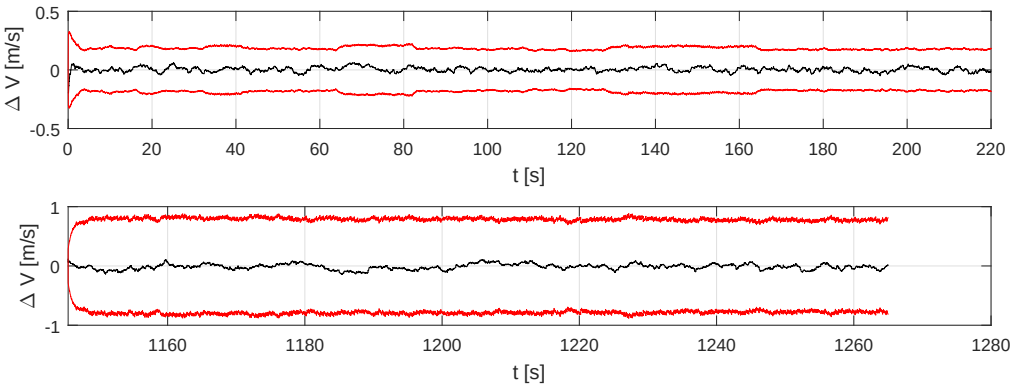


Figure 9.22: Velocity estimation of the integrated system for phase I (upper plot) and phase III (lower plot).

comparing the outcome of the estimated state with the results from Sections 9.1 to 9.3, also for the transformed states M, $\overline{q}$, α, β, σ, p, q, and r. Finally, after all phases were verified, the navigation module was coupled to the control system. By running this configuration, the final results were obtained where the state is estimated by the navigation module, compared to the guidance commands, and fed to the control system, hence, the integration of the GNC system with minimal guidance.

In Figure 9.21 the position estimation is shown in terms of altitude, for phase I (upper plot) and phase III (lower plot). Figure 9.22 then shows the velocity estimation, also for phase I and III, and finally, the attitude estimation errors are plotted in terms of aerodynamic angles, as seen in Figure 9.23. In these figures, the black lines define the state estimates, whereas the red lines represent the covariance of the respective state variable. It is seen that for the integrated GNC-system, the PF converges for the variables that are shown. The GPS and IMU error estimates are used in the pseudorange, range-rate, acceleration, and rotational-rate estimations and thus have a direct influence on the variables that are shown in the aforementioned figures. For the integrated system, they show similar performance as was described in Sections 9.1 and 9.3.

Originally, a measurement-update frequency of 10 Hz was used in the simulator, which showed good performance, as the PF converged for the state variables that were shown earlier in this section. However, to decrease computational load, situations with update frequencies of 5 Hz and 1 Hz were evaluated as well. For this, the altitude, dynamic pressure, Mach number, and angle of attack are shown. In the case that a real guidance system is used, the altitude is of importance, as this usually is used as input for the system, and therefore it is evaluated in this section. The dynamic pressure, Mach number, and angle of attack were used as input for the control system, and so they are evaluated as well.

Figures 9.24 and 9.25 show that a decrease of update frequency also decreases the accuracy of the results, as was expected. This can be explained by the fact that the state estimation cannot be performed using IMU measurements solely, as was seen in Section 9.2. Increasing the update frequency using the GPS and FADS measurements reduces the time for the state estimate to drift from the zero-error line, thereby improving the

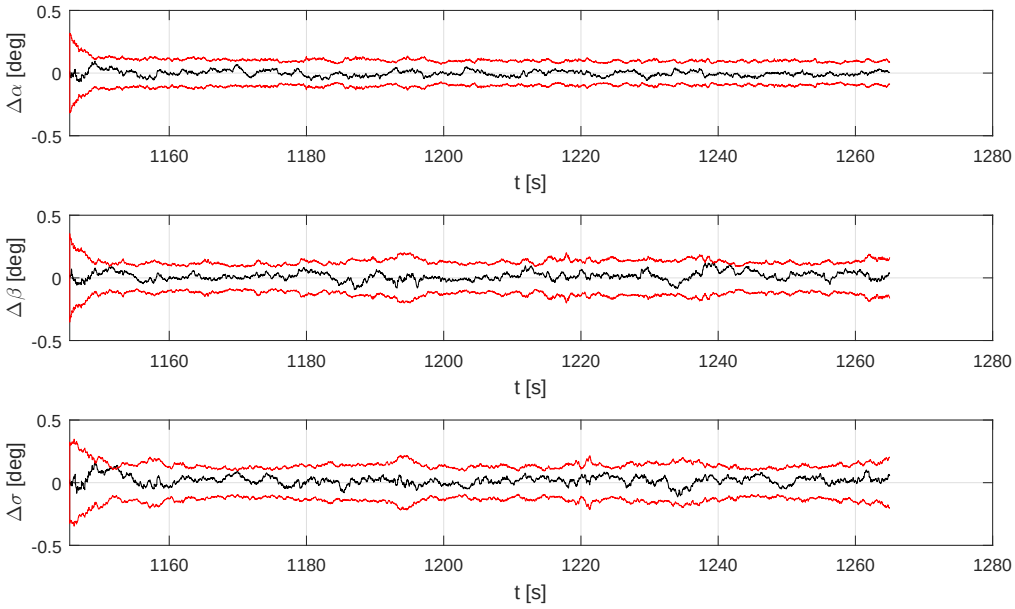


Figure 9.23: Attitude estimation of the integrated system for phase III.

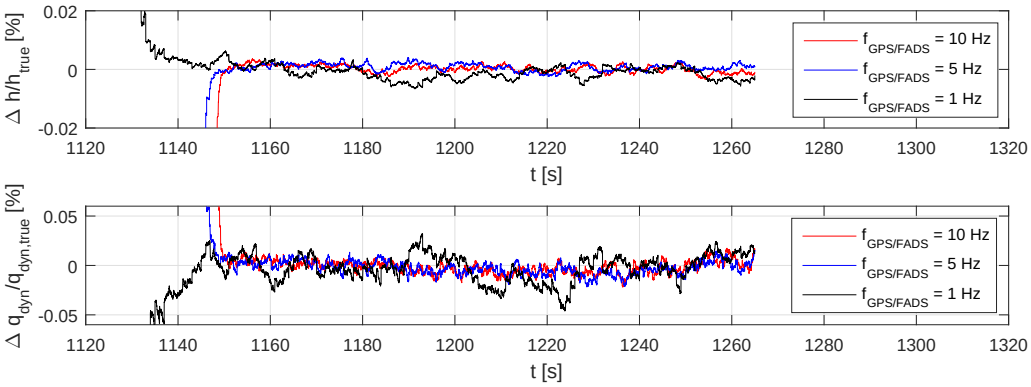


Figure 9.24: Phase III altitude and dynamic pressure estimates for different measurement update frequencies.

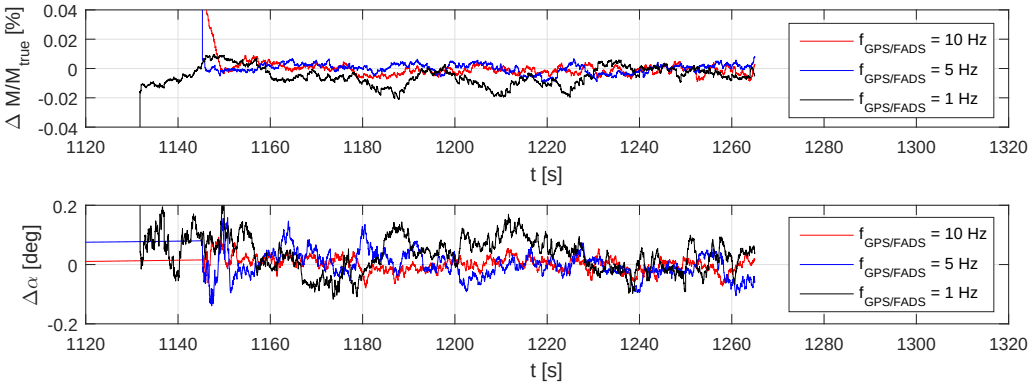
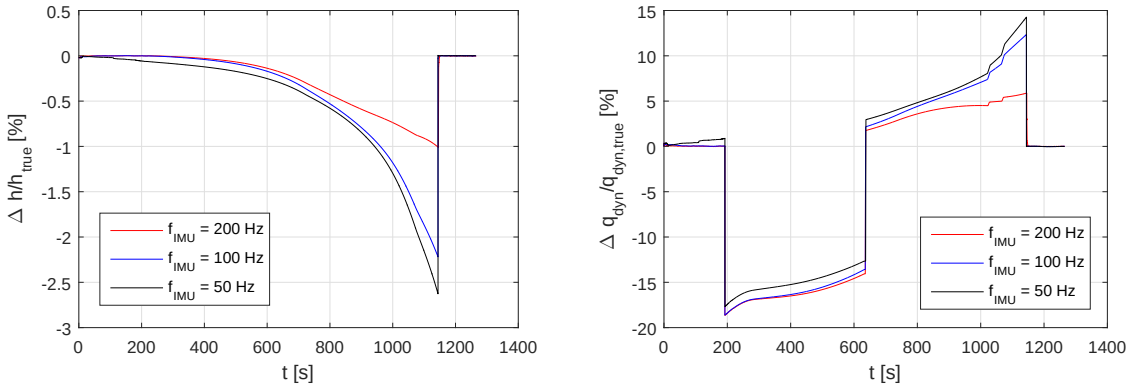
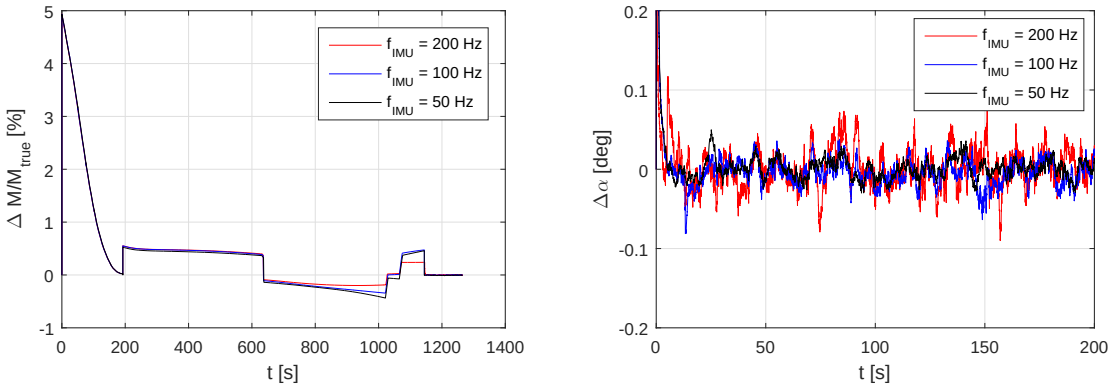


Figure 9.25: Phase III Mach-number and angle-of-attack estimates for different measurement update frequencies.



(a) (b)

Figure 9.26: (**a**) - Altitude estimate for varying IMU frequencies. (**b**) - Dynamic-pressure estimate for varying IMU frequencies



(a) (b)

Figure 9.27: (**a**) - Mach-number estimate for varying IMU frequencies. (**b**) - Angle-of-attack estimate for varying IMU frequencies.

accuracy of the filter. In these figures, the ratio of altitude, dynamic-pressure, and Mach-number errors with respect to their true values are evaluated, and it is seen that when the filter is activated (phase I and phase III), these ratios remain between 0.05% of their true values. Finally, it is seen that the 1 Hz-filter is activated earlier in the trajectory than the other two filters. Due to the coupling between navigation and control system, the state error is included in the input of the latter system. For the 1 Hz-filter, the state error in the blackout phase grows larger than for the other filters, which influences the computed control commands such that a slightly different trajectory is flown. In this trajectory the altitude of 42 km is reached earlier, and therefore the 1 Hz-filter is activated earlier as well.

In terms of computation time, decreasing the update frequency had a small influence, as the computational load of the filter is driven by the frequency of the state propagation using the IMU. This value was decided to be 200 Hz, as discussed earlier in this report. Figures 9.26a, 9.26b, 9.27a and 9.27b show the effect of decreasing the state-propagation frequency, again on the state estimates of the altitude, dynamic pressure, Mach number, and angle of attack.

In these figures the altitude, dynamic-pressure, and Mach-number ratios are plotted for the whole simulation, and from the altitude estimates it can be seen that a low state-propagation frequency decreases accuracy of the estimation process during the blackout phase. As the dynamic pressure and Mach number are dependent on the position and velocity of the vehicle, this is also observed for these variables, especially around $t \approx 1100$ s, which is at the end of the blackout phase. The jumps that are seen in the results for dynamic pressure and Mach number denote boundaries of the atmospheric model, where the behavior of the atmospheric properties, on which the dynamic pressure and Mach number are dependent, changes. It was found that computations at higher altitudes differ for the estimated atmospheric properties and the simulated atmospheric properties, and, as is seen, especially for the 200 Hz-state propagation, the estimates are more

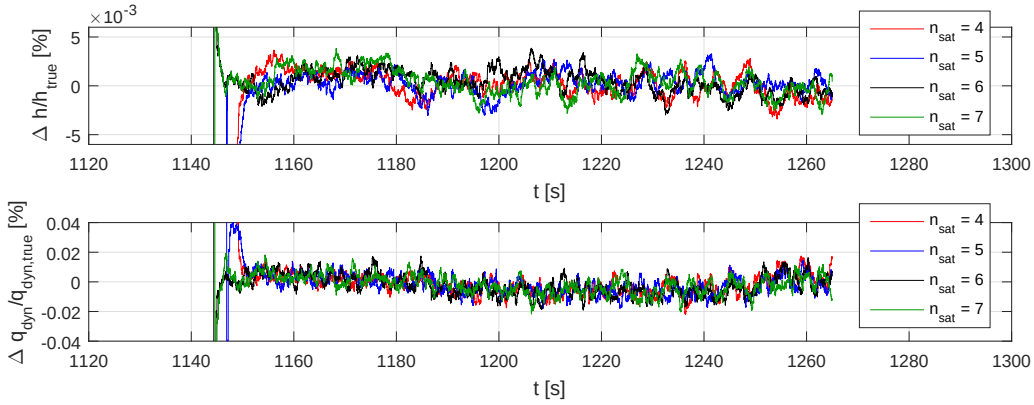


Figure 9.28: Phase III altitude and dynamic pressure estimates for different number of GPS satellites.

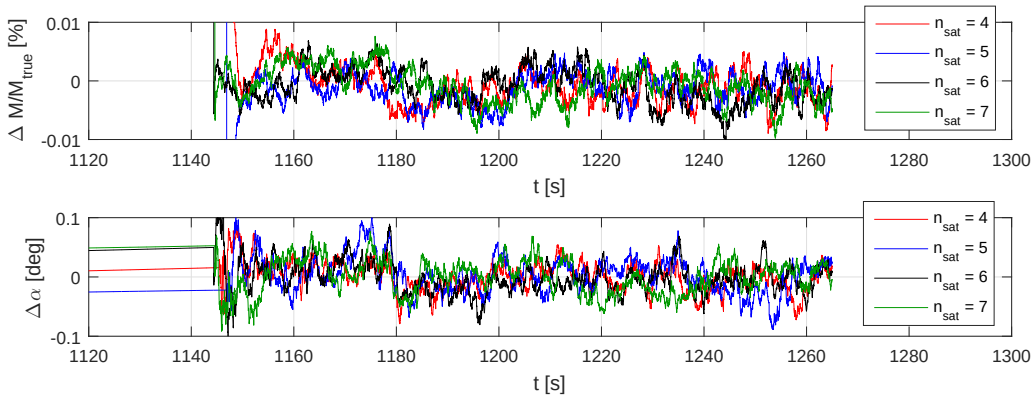


Figure 9.29: Phase III Mach-number and angle-of-attack estimates for different number of GPS satellites.

accurate towards the end of the trajectory, where the altitude of the vehicle is lower. The dynamic-pressure estimates between $t \approx 200$ s and $t \approx 630$ s appear to be much worse than at the end of the blackout phase, however, due to the fact that the true dynamic pressure is smaller at this point in the trajectory, the error is relatively large.

Finally, the angle of attack was evaluated, which is done for phase I. It is seen that for lower state-propagation frequencies, the estimate with activated filter becomes more accurate. This behavior is also found for the other state variables and can be explained by the fact that for a high frequency, the particle cloud of a specific variable is propagated with IMU measurements for a large number of steps, after which it is reshaped in the measurement update. The IMU measurements are not sufficient for accurate state estimation, and when they have a large influence on the particle clouds (*i.e.*, for high state-propagation frequencies), the accuracy of the state estimates decreases.

Then, in Figures 9.28 and 9.29 the influence of the number of satellites on the state estimation process is shown for phase III. Again it is seen that the state estimates are very accurate, as they remain within 0.02% of their true values. Furthermore, it seems that there is negligible influence of the number of used satellites on the accuracy of the state estimate. However, by numerically evaluating the mean magnitude of the error during the third phase, it is found that the mean altitude-ratio error decreases from 0.0042% for four satellites to 0.0012% for seven satellites. The mean ratio of the dynamic pressure then experiences a reduction from 0.0244% to 0.0057%, and, finally, the Mach-number ratio is decreased from 0.0034% to 0.0026%, again for four and seven satellites respectively. The mean error of the angle of attack during phase III remains almost constant when the number of GPS satellites is increased. As a conclusion, it can be stated that, when more than four satellites are visible, they can best be used, as they improve the accuracy of the state estimates. Next to that, the additional computational time they induce is negligible with respect to the total computation time.

At this point, the variation of the state-propagation frequency, measurement-update frequency, and the number of GPS satellites is evaluated. For the following simulations, the original values of the aforementioned

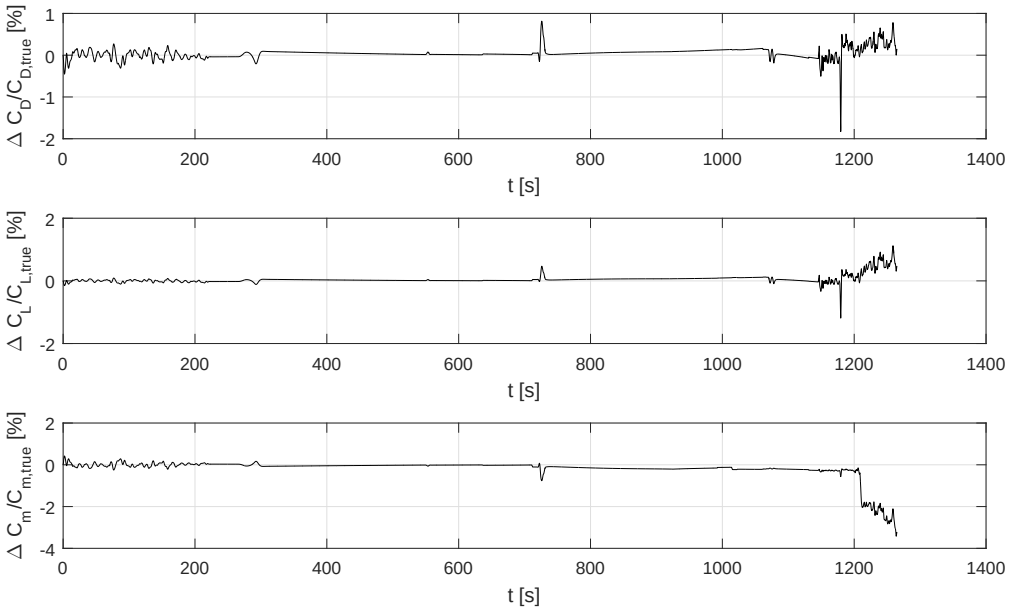


Figure 9.30: Influence of the state-estimation errors on the drag-, lift-, and pitch-moment-coefficients.

parameters are kept, hence, a measurement-update frequency of 10 Hz and a state-propagation frequency of 200 Hz are used, as well as the minimum number of four satellites. It was seen that the Mach number and dynamic pressure are estimated very accurately during phase I and phase III, however, during the blackout phase, the dynamic-pressure estimate was found to be relatively large with respect to its true value. Earlier it was discussed that the control forces and moments that are computed are dependent on the Mach number, dynamic pressure and attitude of the vehicle. The state-estimation error of these variables then also induces an error in these forces and moments. The influence of the state-estimation errors is first reflected on the computation of the aerodynamic coefficients, of which ratio of the drag, lift, and pitch moment coefficients with respect to their true values is shown. In this case, the true values are the coefficients that were computed for the ideally controlled vehicle, *i.e.*, where the estimated state was assumed to be the true state.

In Figure 9.30 these drag, lift, and pitch-moment coefficient ratios are shown, and it is seen that the state-estimation errors of the Mach number and attitude angles induce errors of less than 0.5% for the computed coefficients during phase I. Then, in the blackout phase, the aerodynamic-coefficient errors drift slightly from their true values, after which they are relatively noisy in phase III. This can be related to the fact that the actuator deflections oscillate for this phase, as is seen in Figure 9.31. These oscillations will be explained further on in this section. Next to that, the angle-of-attack estimate becomes more significant, as its error increases relative to the true value. The peaks that are found at $t \approx 265$ s, $t \approx 720$ s, $t \approx 1070$ s, and $t \approx 1180$ s, are due to the bank reversals, and the activation and deactivation of the guidance system. The estimates of the angle of attack, and, to a lesser extent, the Mach number also showed peaks at these points. Due to the fact that the coefficients are computed via linear interpolation, using the aforementioned two variables as input, these peaks are induced for the aerodynamic coefficients as well. Then, at the end of the trajectory, the accuracy of the coefficients decreases, which is caused by the deflection oscillations of the aerodynamic surfaces. The trim commands are also computed using the estimated Mach number and attitude, and the inaccuracy of the pitch-moment coefficient is also found in the body-flap deflection difference in Figure 9.31.

The aerodynamic coefficients are then used in the computation of the actuator commands, and therefore these commands are evaluated as well. In Figure 9.31, the difference between the elevon, rudder, and body flap deflections of the ideally controlled vehicle and the system with integrated navigation module are shown, and it is seen that relatively small errors in Mach number, dynamic pressure, angle of attack lead to significant differences in the actuator commands. It is seen that in the first part of the trajectory, the aerodynamic actuators are deactivated. Then, when they are active, the deflections start to oscillate, which can be explained by the fact that for an underestimation of a specific attitude angle, the angle may be corrected in a specific

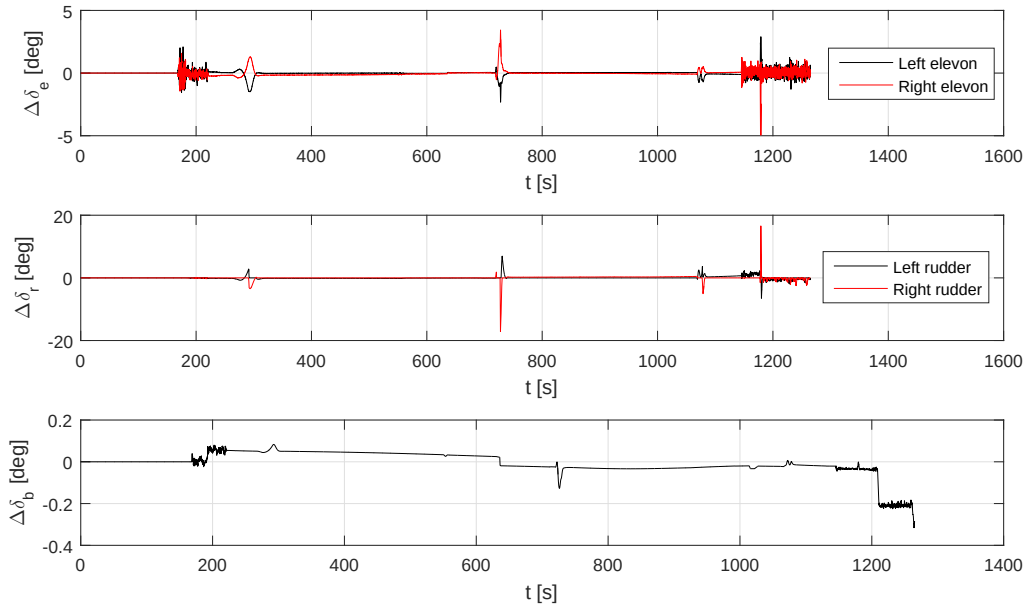


Figure 9.31: Influence of the state-estimation errors on the difference between the actuator deflections for the ideally controlled vehicle (where the estimated state is the true state) and the vehicle with integrated navigation system.

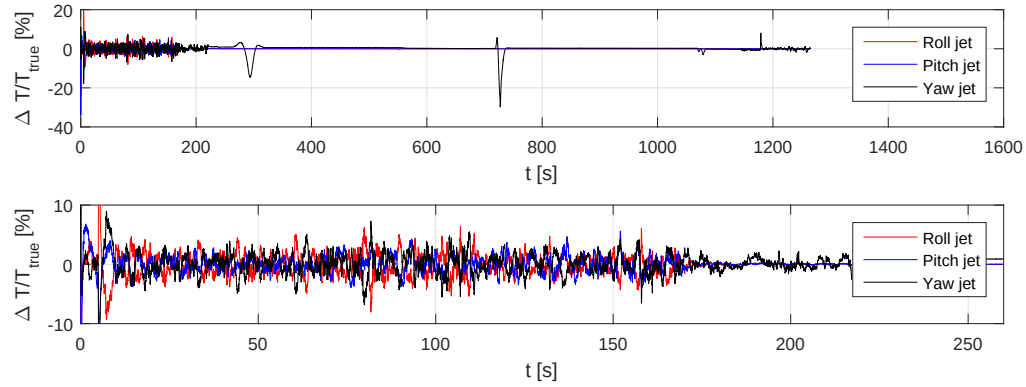


Figure 9.32: Influence of the state-estimation errors on the difference between the thruster moments for the ideally controlled vehicle (where the estimated state is the true state) and the vehicle with integrated navigation system.

direction, after which the estimation rises to an overestimation, thereby correcting the controlled angle in the opposite direction. The oscillations that form due to this were also seen in the aerodynamic-coefficient ratios in Figure 9.30 and are only present when the filter is activated, as the angle corrections are induced by the measurement update in the filter.

Subsequently, the peaks due to activation and deactivation of the guidance system can be seen, as well as the peaks that were induced by the bank reversals. When performing a banking maneuver (which is done for all peaks), the control command that is computed to perform that maneuver contains an error for the system with the integrated navigation module. Due to this, the vehicle's correction of the attitude also contains an error, which subsequently induces a new error in the newly computed control commands. This process is continued up until the maneuver is stopped; after this, the error is reduced again. The behavior that is described in this paragraph explains why there are differences between the ideally controlled vehicle and the vehicle with integrated navigation module, which result in the peaks at the aforementioned maneuvers.

Finally, the thruster commands are handled, which conclude the discussion on the effect of the state-estimation error on the control commands. The results in Figure 9.32 contain the behavior of the commands

over the whole trajectory, where it can be seen that the roll and pitch jets are deactivated somewhere during phase I. The part of the flight where all jets are active can be seen in the lower graph, where ratios are seen that are mostly less than 5%. The oscillations in this phase again can be coupled to the behavior that was described for the aerodynamic control surfaces, and, the same can be stated for the peaks that occur during the bank reversals and guidance deactivation.

This concludes the chapter on the simulation and results of the navigation module. The three phases were evaluated individually, and the behavior of the state estimates was discussed. Then, it was found that all estimators performed acceptable, but, depending on the variables of interest for the control system, the PF was chosen to be the most well-suited filter, despite the larger computational load. This filter was then integrated and coupled to the control system, and together with the guidance profile, a fully integrated GNC-system was implemented. After running the simulator, it was shown that the integrated system was able to estimate the state accurately and control the vehicle, despite the introduced error from the navigation module. The integrated filter was evaluated, and similar performance was seen as for the individual phases. Finally, the effect of operating frequencies and number of GPS satellites on the error was discussed, as well as the influence of the error on the aerodynamic coefficients and control commands.

10

Conclusions and Recommendations

In this thesis, the implementation, verification, and simulation of an integrated GNC-system was described, where the FADS was evaluated on its influence on the state-estimation process. From the results that were found and discussed throughout this report, conclusions could be drawn, which are described in Section 10.1. Finally, recommendations for future research are stated in Section 10.2.

10.1 Conclusions

Starting this thesis, the relation between the research that was to be done in this thesis and the corresponding field of interest was discussed, after which the goal of the research was stated. Then, the so-called research question was formulated as:

"To what extent can the performance of an estimator-based navigation module for an unpowered, winged re-entry vehicle be optimized between the atmospheric entry point and the TAEM interface?"

and it was divided into four groups, all addressing a specific aspect of the research that was to be done. First, the design and development of a re-entry flight simulator was handled, which provided the baseline of the research that was to be done. Then, the development and simulation of both a control system and a navigation module were discussed, which together provided an NC-system due to the absence of a realistic guidance module. For both these subtopics, individual research on performance optimization was done, which lead to a final integrated NC-system on the baseline simulator. This system was then evaluated on performance as well, concluding the four aspects that were to be handled to provide an answer to the aforementioned research question.

Design and development of the re-entry flight simulator

The first phase of this thesis was dedicated to the development of the baseline simulator, which contained the flight dynamics that were described in Chapter 3. The US76 atmospheric model was used, which was related to the HORUS-2B aerodynamics to compute any aerodynamic forces and moments. In this aerodynamics module, any values outside the angle of attack, angle of sideslip and Mach number breakpoints were clipped to the last existing value due to the uncertainty of the behavior of the aerodynamic coefficients outside these breakpoints. For the gravitational field the J_2-effect was taken into account.

The baseline simulator was taken from Mooij (2017), as the Aerospace Blockset significantly increased the computational loads due to additional calculations that were required to obtain the preferred state variables and the fact that the blocks from this blockset themselves were computationally intensive. For verification, the baseline simulator nominal trajectory and open loop simulation results were compared to previous research that was described in Mooij (1997). Finally, the inertial frame was chosen as a reference, thereby inducing a small time uncertainty, however, due to the nature of the research that was done, this uncertainty was deemed insignificant.

Development and simulation of the control system

Then, after the baseline simulator was completed, the control and guidance system were addressed. A simple

guidance profile consisting of angle-of-attack and bank-angle commands was used, and the angle of sideslip was commanded to be zero at all times. In this profile, a guidance activation, several bank reversals, and guidance deactivation were included, which would show insight into the GNC-system behavior for these maneuvers. The LQR controller was chosen and implemented, where it was concluded that even though this controller is quite basic, it was able to compute the control commands such that the vehicle was able to follow the aforementioned guidance profile. The control system was fed using the true state of the vehicle, due to the absence of a navigation system, and thus an additional error was to be included for the integrated system.

The controller behavior was evaluated and a small delay between the guidance profile and the actual aerodynamic angles was observed, which can be explained by the fact that the system is discrete and no rotational rates were commanded. For the design of the controller, longitudinal and lateral motion were decoupled, as the eigenvalues of the decoupled and the coupled system were very similar. However, after integration of the controller, it was seen that both types of motion are coupled, and that small induced angles of attack are found at the bank-angle maneuvers. Due to these induced angle-of-attack differences, the vertical lift component differed from the commanded value, which was corrected by the vehicle using a bank angle. Controller tests were done for bank maneuvers with a rates in the order of 15°/s, and relatively large attitude errors were found. It was concluded that these errors can be greatly reduced by application of a real guidance system. Finally, it was stated that the performance of the controller can be enhanced by lowering the maximum allowed attitude angles and rates, which inevitably cause larger actuator deflections and thruster moments. However, due to the absence of a real guidance system, the conclusions that were drawn from the controller are limited.

Development and simulation of the navigation module

Subsequently, the design of the navigation module was handled, which was divided into the sensor and estimator systems. As was described, the navigation module was the driving system for major part of the research and thus was evaluated more thoroughly than the control system.

In terms of sensors, it was concluded that the IMU/GPS combination was sufficient for accurate state estimation. For the state propagation, the Euler method was chosen to minimize the computational load of this step. Then, addition of the FADS was evaluated, and the built-in estimator of this sensor was deemed unsuccessful as a stand-alone estimator of the air-data variables, as it showed large oscillations in Mach number. It was stated that, as a contingency, this estimator could be used, in case of, for example, a GPS outage for a longer time. From the FADS sensor, it was concluded that the layout was defined such that the angle of attack and the angle of sideslip could both be estimated accurately.

Then, for the estimators, the EKF, UKF, and PF were applied, and the entry trajectory was divided into three phases: (I) from entry to blackout, (II) the blackout phase itself, and (III) from blackout to the TAEM interface. For phase I, it was seen that all filters converge for almost all state variables. The accelerometer biases can only be estimated when the accelerations are still very small, which is restricted to the start of the trajectory only. The accelerometer scaling factors converged after the accelerations became sufficiently large. The altitude of the vehicle was prioritized in the state estimation, and it was seen that the PF outperformed the other filters.

During phase II, the filters were deactivated and it was concluded that the final estimate of this phase was restricted to the input. It was found that the behavior of the state estimates is mainly dependent on the behavior of the IMU measurements. Then, phase III was evaluated, where the UKF outperformed the other two filters on position and velocity, and the PF provided the most accurate state estimates in terms of attitude. The UKF, however, showed peaks in attitude estimation during guidance deactivation, which were explained by the fact that nonlinear mapping of the sigma points is done in the estimation of the carrier-phase difference measurements. The PF also experienced these peaks, however, these were greatly reduced due to the fact that a large number of particles is used in this filter. It was concluded that this behavior can be avoided by limiting the allowable maximum bank-angle rate and by increasing the number of used GPS satellites, where the latter was deemed most efficient. For this phase, all filters converged, except for the scaling factors, which are dependent on the magnitude of the accelerations.

Furthermore, it was found that the filters diverge for an IMU/FADS combination. However, in case of a brief GPS outage, the filters are able to recover quickly when the GPS measurements become available again. After comparison of the individual filters, the EKF was proven to be the least efficient, especially for highly nonlinear parts of the trajectory. The UKF performed slightly better than the PF, however, the latter filter was chosen for integrated due to the negligible the attitude-estimation peaks during guidance activation for an unlimited bank-angle rate and minimum number of GPS satellites available, hence, the worst-case scenario. Evaluation of the PF showed that the number of particles enhances the accuracy of the estimates, however, it

also increases computational load. Then, a high resampling frequency proved to have a positive effect on the accuracy of the state-estimation process as well. The addition of the FADS improved the position estimation, as the FADS measurements are greatly dependent on the altitude of the vehicle, however, it showed negligible effect on the velocity and attitude estimation. The Mach-number and aerodynamic-angle estimates were found to be much more accurate than the results from the built-in estimator of the FADS.

Simulation of the integrated system
Finally, when the trade-off was completed and the FADS integrated PF-based navigation module was integrated into the simulator, several tests were run to verify the system as a whole. Subsequently, the navigation module was coupled to the control system, resulting in the fully integrated GNC-system. Simulations proved that the control system is still able to ensure a fully controlled flight driven by the guidance commands, even though the state input contained an estimation error. Or, in other words, the navigation module was proven to provide a sufficient state estimation for the control system to still perform as desired.

After various tests using the integrated system, it was concluded that a high measurement-update frequency induces a more accurate state estimate. A reduction of the state-propagation frequency then also improves the state-estimation process when the filter is active, however, during the blackout phase, the error grows larger than for high state-propagation frequencies. In terms of the number of satellites that are used, it was found that by increasing this number (depending on the number of visible satellites), the state estimate becomes slightly more accurate. The effect of the state-estimation errors was found to be small with respect to the computed aerodynamic coefficients, and resulted in only large differences in actuator commands during the bank reversals and guidance activation and deactivation.

To relate the research that was done and the conclusions that were drawn in this chapter to the predefined research question, it can be said that in the flight between atmospheric entry and the TAEM interface, implementation of the FADS increased the performance of the estimator-based navigation module. By using the PF instead of a baseline EKF that was used in Mooij and Chu (2002), performance enhancement was achieved as well. These two conclusions state that the performance was increased, if not optimized for the specific configuration. However, the recommendations shown in the following subsection state that this optimization process is not ended here, as new points of improvement and new questions will always be found.

10.2 Future Recommendations

Even after completion of a large project, there is always room for more research, and new questions always arise. Therefore, the following recommendations for future work are defined as:

- The design and integration of a real guidance system, where not only a simple attitude profile is provided, but where, using guidance laws, it is defined what the vehicle has to do at a specific point in the trajectory.
- The extension of the flight into TAEM interface. In the simulator that was described in this thesis, only a small part of the flight contains FADS measurements, however, in lower parts of the atmosphere the air becomes denser, and the behavior of the FADS in combination with the state estimator for these regimes is interesting for the performance of the system during the TAEM interface.
- The addition of a FADS error model in the estimator, so that the errors with highest significance can be estimated as well. It would be interesting to research the improvement of the state estimation due to the addition of these error models.
- The optimization of the hypersonic-descent trajectory in terms of G-loads and heat flux, which represent two variables that are of importance for the so-called entry corridor, in which the vehicle has to be at all times during re-entry.
- The evaluation of multiple resampling methods for the PF. The principle of SIR was used in this thesis, however, the performance of the PF may be optimized when using a different approach.
- The evaluation of different filters, such as the alpha-beta filter. This will aid the optimization of the state-estimation process of the navigation filter, as the most well-suited filter can be chosen (this does not necessarily have to be the PF that was evaluated in this thesis).
- The design and integration of a nonlinear controller, such as an NDI-controller. Doing this may decrease the attitude-angle differences between the true and the commanded state of the vehicle. These differences were found to be significant, especially during the bank reversals and guidance activation and

deactivation. Reducing the peaks that occurred at these points can be deemed an improvement of the control system.

- The transformation of the state propagation from the inertial frame to the rotating frame, to avoid the time error that is induced. By doing this, the influence of this error on the state-estimation process can be evaluated.
- The implementation of an extended atmosphere and gravitational field model, which estimates the air data and the gravitational acceleration with higher accuracy. This will increase the representativity of the simulator with respect to real life.
- The addition of a wind model to the simulator, as winds during re-entry flight may have a severe impact on the vehicle dynamics, and thus also on the navigation and control systems.

Bibliography

Anderson Jr., J.D. (2011). *Fundamentals of Aerodynamics.* McGraw-Hill, New York, 5th edition.

Anderson Jr., J.D. (2012). *Introduction to Flight.* McGraw-Hill, New York, 7th edition.

Bistrovs, V. and Kluga, A. (2011). "MEMS INS/GPS Data Fusion using Particle Filter". Riga Technical University, ISSN 1392-1215, No. 6.

Choukroun, D. Bar-Itzhack, I.Y. and Oshman, Y. (2002). "A Novel Quaternion Kalman Filter". *AIAA Guidance, Navigation, and Control Conference and Exhibit* - AIAA 2002-4460.

Cobleigh, B.R., Whitmore, S.A., Hearing Jr., E.A., Borrer, J. and Roback, V.E. (1999). "Flush Airdata Sensing (FADS) System Calibration Procedures and Results for Blunt Forebodies". NASA/TP-1999-209012.

Cooke D.R. (1982). "Space Shuttle Stability and Control Test Plan". *9th Atmospheric Flight Mechanics Conference* - AIAA-82-1315.

Costa, R.R., Silva, J.A., Wu, S.F., Chu, Q.P. and Mulder, J.A. (2002). "Atmospheric Re-Entry Modeling and Simulation". *Journal of Spacecraft and Rockets*, Vol. 39, No. 4, pp. 636-639.

Crassidis, J.L. (2005). "Sigma-Point Kalman Filtering for Integrated GPS and Inertial Navigation". *AIAA Guidance, Navigation, and Control Conference and Exhibit* - AIAA 2005-6052.

Crassidis, J.L. and Markley, F.L. (1996). "Attitude Estimation Using Modified Rodrigues Parameters". *Proceedings of the Flight Mechanics/Estimation Theory Symposium* - NASA/CP-1996-3333, pp. 71-83.

Dean, A.S. and Mena, A.L. (1983). "The Calibration and Flight Test Performance of the Space Shuttle Orbiter Air Data System". *NASA, Langley Research Center Shuttle Performance: Lessons Learned, Pt. 2* - pp 1187-1212.

Dormand, J.R. and Prince, P.J. (1980). *A Family of Embedded Runge-Kutta Formulae. Journal of Computational and Applied Mathematics*, Vol. 6, No. 1.

Ellsworth, J.C. and Whitmore, S.A. (2007). "Re-entry Air Data System for a Sub-orbital Spacecraft Based on X-34 Design". *AIAA Aerospace Sciences Meeting and Exhibit* - AIAA 2007-1200.

Fay, G. and Speyer, J. (2008). "GPS Code Tracking Using a Sampling Importance Resampling Particle Filter". *AIAA Guidance, Navigation and Control Conference and Exhibit* - AIAA 2008-7473.

Goodman, J.L. (2011). *History of Space Shuttle Rendezvous.* Lyndon B. Johnason Space Center/NASA, STI-DAA, 3rd revision.

Goodman, J.L. and Propst, C.A. (2008). "Operational Use of GPS Navigation for Space Shuttle Entry". *IEEE/ION PLANS Conference* - 20080014095.

Gopal, M. (1993). *Modern Control System Theory.* New Age International (P) Ltd., 2nd edition.

Grewal, M.S., Weill, L.R. and Andrews, A.P. (2001). *Global Positioning Systems, Inertial Navigation, and Integration.* John Wiley & Sons, Inc.

Gross, J., Gu, Y., Gururajan, S., Seanor, B. and Napolitano, M.R. (2010). "A Comparison of Extended Kalman Filter, Sigma Point Kalman Filter, and Particle Filter in GPS/INS Sensor Fusion". *AIAA Guidance, Navigation, and Control Conference* - AIAA 2010-8332.

Gustafsson, F. (2009). "Particle Filter Theory and Practice with Positioning Applications". *IEEE A&E Systems Magazine*, Vol. 25, No. 7.

Higham, D.J. (2004). *An Introduction to Financial Option Valuation.* Cambridge University Press, 1st edition.

Hillje, E.R. and Nelson, R.L. (1981). "Postflight Analysis of the Space Shuttle Ascent Air Data System". *AIAA 1st Flight Testing Conference* - AIAA 81-2457.

Houcque, D. (2005). "Introduction to MATLAB for Engineering Students". Northwestern University, v1.2.

Julier, S.J. and Uhlmann, J.K. (1997). "A New Extension of the Kalman Filter to Nonlinear Systems". *University of Oxford*.

Karlgaard, C.D. and Schoenenberger, M. (2017). "Planetary Probe Entry Atmosphere Estimation Using Synthetic Air Data System". *AIAA Atmospheric Flight Mechanics Conference* - AIAA 2017-0470.

Klees, R. and Dwight, R.P. (2013). Applied Numerical Analysis - Course AE2220I. Delft University of Technology, Faculty of Aerospace Engineering.

Kuipers, J.B. (1999). *Quaternions and Rotation Sequences - a Primer with Applications to Orbits, Aerospace, and Virtual Reality*. Pinceton University Press.

Lissauer, J.J. and De Pater, I. (2013). *Fundamental Planetary Science - Physics, Chemistry and Habitability*. Cambridge Universtiy Press, New York, 1st edition.

Mathews, J.H. and Fink, K.D. (1999). *Numerical Methods Using Matlab*. Prentice-Hall, 3th edition.

MBB (1988). "Study on Re-Entry Guidance and Control". Final report; ESA report reference: ESA cr (p) 2652, MBB Space Communication and Propulsion Systems Division.

Montez, M.N. and Madden, M.F. (1982). "Space Shuttle Orbiter Descent Navigation". *Guidance and Control Conference*, pp. 361-366, AIAA 82-1563.

Mooij, E. (1994). "The Motion of a Vehicle in a Planetary Atmosphere". Delft University of Technology, Faculty of Aerospace Engineering, Report LR-768, ISBN: 90-5623-003-4.

Mooij, E. (1995). "The HORUS-2B Reference Vehicle". Delt University of Technology, Faculty of Aerospace Engineering, Report M-692.

Mooij, E. (1997). "Linear Quadratic Regulator Design for an Unpowered Winged Re-entry Vehicle". Delft University of Technology, Faculty of Aerospace Engineering, Report LR-806, ISBN: 90-5623-040-9.

Mooij, E. (1998). *Aerospace-Plane Flight Dynamics - Analysis of Guidance and Control Concepts*. Ph.D. Thesis, Delft University of Technology.

Mooij, E. (2015). *Re-entry Systems (AE4870B Lecture Notes)*. Delft University of Technology, Faculty of Aerospace Engineering.

Mooij, E. (2017). Personal Communication.

Mooij, E. and Chu, Q.P. (2002). "Tightly-coupled IMU/GPS Re-entry Navigation System". *AIAA Guidance, Navigation, and Control Conference and Exhibit*, AIAA-2002-5005.

Mooij, E. and Ellenbroek, M. (2007). "Multi-Functional Guidance, Navigation and Control Simulation Environment - Rapid Prototyping of Space Simulations". *Modeling and Simulation Technologies Conference*, REF: AIAA 2007-6887.

Moran, M.J. and Shapiro, H.N. (2006). *Fundamentals of Engineering Thermodynamics*. John Wiley & Sons, Inc., 5th edition.

Mulder, J.A., Chu, Q.P., Sridha, J.K., Breeman, J.H. and Laban, M. (1999). "Non-linear Aircraft Flight Path Reconstruction Review and New Advances". *Progress in Aerospace Sciences*, Elsevier, **35**, pp. 673-726.

Mulder, J.A., Van Staveren, W.H.J.J., Van der Vaart, J.C., De Weerdt, E., De Visser, C.C., In 't Veld, A.C. and Mooij, E. (2013). *Flight Dynamics (Lecture Notes AE3202)*. TU Delft.

NASA (1976). *U.S. Standard Atmosphere 1976*. U.S. Government Printing Office, Washington D.C.

Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P. (1989). *Numerical Recipes in C - The Art of Scientific Computing*. Cambridge Univeristy Press, 2nd ed.

Ragsdale, A. (1985). "Space Shuttle Navigation Validation". *Computers in Aerospace Conference, 5th*, pp 408-415 - AIAA 85-6025.

Regan, F.J. and Anandakrishnan, S.M. (1993). *Dynamics of Atmospheric Re-Entry*. AIAA Education Series, 1st edition.

Ren, Y. and Ke, X. (2010). "Particle Filter Data Fusion Enhancements for MEMS IMU/GPS". *Intelligent Information Management, Scientific Research*, 417-421.

Siemers III, P.M., Wolf, H. and Flanagan, P.F. (1983). "Shuttle Entry Air Data System Concepts Applied to Space Shuttle Orbiter Flight Pressure Data to Determine Air Data - STS 1-4". AIAA-83-0118.

Smith, A.L. (2011). "An Integrated Navigation System Model for Coupled GNSS-Inertial Navigation Research". *AIAA Modeling and Simulation Technologies Conference* - AIAA 2011-6330.

Spilker, J.J. (1994). Satellite Constellation and Geometric Dilution of Precision. *Global Positioning System: Theory and Applications*, Vol. 1, pp.177-208.

Srinivasan, K. (2006). "Control System Design Using State-Space Methods". *Mechanical Engineers' Handbook: Instrumentation, Systems, Controls and MEMS, Volume 2*, 3rd ed., pp. 757-788, John Wiley & Sons, Inc.

Stein, J.M., Machin, R.A., Wolf, D.F. and Hillebrandt, F.D. (2007). "Low Velocity Airdrop Tests of an X-38 Backup Parachute Design". *18th AIAA Aerodynamic Decelerator Systems Technology Conference* - 2007-0026138.

Stoer, J. and Bulirsch, R. (1993). *Introduction to Numerical Analysis*. Springer-Verlag, New York, 2nd edition.

Tapley, B.D., Schutz, B.E. and Born, G.H. (2004). *Statistical Orbit Determination*. Elsevier Academic Press, 1st edition.

Van der Merwe, R. (2004). "Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models". Oregon Health & Science University, *Scholar Archive*, Paper 8.

Wakker, K.F. (2010). *Astrodynamics-I (Lecture Notes AE4874-I)*. Delft University of Technology, Faculty of Aerospace Engineering.

Wang, L. and Bochsler, D. (1987). "Space Shuttle Onboard Navigation Console Expert/Trainer System". *First Annual Workshop on Space Operations Automation and Robotics (SOAR 87)*, pp 11-14.

Welch, G. and Bishop, G. (2006). "An Introduction to the Kalman Filter". University of North Carolina, Department of Computer Science, Report TR 95-041.

Wendel, J., Maier, A., Metzger, J. and Trommer, G.F. (2005). "Comparison of Extended and Sigma-Point Kalman Filters for Tightly Coupled GPS/INS Integration". *AIAA Guidance, Navigation, and Control Conference and Exhibit* - AIAA 2005-6055.

Wertz, J.R. (2009). *Orbit & Constellation Design & Management*. Microcosm Inc., 2nd edition.

Whitmore, S.A. and Moes, T.R. (1994). "Measurement Uncertainty and Feasibility Study of a Flush Airdata System for a Hypersonic Flight Experiment". *NASA Technical Memorandum 4627*, pp. 1-18.

Xue, S. and Lu, P. (2009). "Constrained Predictor-Corrector Entry Guidance". *AIAA Guidance, Navigation, and Control Conference*, REF: AIAA 2009-5767.

Zhang, M., Lu, B. and Song, W.J. (1999). "A Method for Dual-Frequency Ionospheric Time-Delay Correcting Using a C/A Code GPS Receiver". *Journal of Electronics (China)* - Vol. 16, No. 1.

Zhao, Y. (2015). "Optimized Filter Design for Non-Differential GPS/IMU Integrated Navigation". Technische Universität Darmstadt, ISBN: 978-3-935631-35-8.

Zhou, J., Knedlik, S. and Loffeld, O. (2010). "Sequential Processing of Integrated Measurements in Tightly-Coupled INS/GPS Integrated Navigation System". *AIAA Guidance, Navigation, and Control Conference* - AIAA 2010-8190.

Appendices

A

Variable Transformations and HORUS Characteristics

In this appendix the reference frame transformations that are used throughout the simulation are shown (Appendix A.1). Note that in the transformation matrices, the trigonometric functions are abbreviated by 'c' (cos) and 's' (sin) for simplification. Next to that, the conversion between several state variables is shown in Appendix A.2. Finally, the HORUS 2B main characteristics can be found in Appendix A.3.

A.1 Reference Frame Transformation Matrices

This section contains all transformation matrices that may be used in the simulation, together with the conversion between classical Euler angles and aerodynamic angles.

Rotating planetocentric to inertial planetocentric frame

This transformation is done between the two reference frames that have their origin in the c.o.m. of the main body, which in this case is the Earth. From the definition of both frames it can be stated that they coincide at the $J2000$ epoch, at $t = 0$. The transformation matrix then is defined as

$$\mathbf{C_{I,R}} = \mathbf{C_3}\left(-\omega_{cb}t\right) = \begin{bmatrix} \mathrm{c}\omega_{cb}t & -\mathrm{s}\omega_{cb}t & 0 \\ \mathrm{s}\omega_{cb}t & \mathrm{c}\omega_{cb}t & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{A.1}$$

where ω_{cb} is the rotational velocity of the main body in rad/s and t is the time from the $J2000$ epoch in s.

Vertical to rotating planetocentric frame

This transformation matrix is defined as

$$\mathbf{C_{R,V}} = \mathbf{C_3}\left(-\tau\right)\mathbf{C_2}\left(\frac{\pi}{2}+\delta\right) = \begin{bmatrix} -\mathrm{c}\tau\mathrm{s}\delta & -\mathrm{s}\tau & -\mathrm{c}\tau\mathrm{c}\delta \\ -\mathrm{s}\tau\mathrm{s}\delta & \mathrm{c}\tau & -\mathrm{s}\tau\mathrm{c}\delta \\ \mathrm{c}\delta & 0 & -\mathrm{s}\delta \end{bmatrix} \tag{A.2}$$

where τ is the longitude and δ is the latitude, both expressed in radians.

Vertical to inertial frame

This transformation matrix is defined as

$$\mathbf{C_{I,V}} = \mathbf{C_3}\left(-\tilde{\tau}\right)\mathbf{C_2}\left(\frac{\pi}{2}+\delta\right) = \begin{bmatrix} -\mathrm{c}\tilde{\tau}\mathrm{s}\delta & -\mathrm{s}\tilde{\tau} & -\mathrm{c}\tilde{\tau}\mathrm{c}\delta \\ -\mathrm{s}\tilde{\tau}\mathrm{s}\delta & \mathrm{c}\tilde{\tau} & -\mathrm{s}\tilde{\tau}\mathrm{c}\delta \\ \mathrm{c}\delta & 0 & -\mathrm{s}\delta \end{bmatrix} \tag{A.3}$$

where $\tilde{\tau} = \tau + \omega_{cb}t$ is the celestial longitude and δ is the latitude, both expressed in radians.

Trajectory to vertical frame

This transformation matrix is defined as

$$\mathbf{C_{V,T}} = \mathbf{C_3}\left(-\chi\right)\mathbf{C_2}\left(-\gamma\right) = \begin{bmatrix} \mathrm{c}\chi\mathrm{c}\gamma & -\mathrm{s}\chi & \mathrm{c}\chi\mathrm{s}\gamma \\ \mathrm{s}\chi\mathrm{c}\gamma & \mathrm{c}\chi & \mathrm{s}\chi\mathrm{s}\gamma \\ -\mathrm{s}\gamma & 0 & \mathrm{c}\gamma \end{bmatrix} \tag{A.4}$$

where χ is the heading angle and γ is the flight path angle, both expressed in radians. These angles are subscripted with an a for the airspeed based frame and with a g for the groundspeed based frame.

Aerodynamic to trajectory frame

This transformation matrix is defined as

$$\mathbf{C_{T,A}} = \mathbf{C_1}\left(\sigma\right) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \mathrm{c}\sigma & \mathrm{s}\sigma \\ 0 & -\mathrm{s}\sigma & \mathrm{c}\sigma \end{bmatrix} \tag{A.5}$$

where σ is the bank angle in radians. Note that this transformation is either airspeed based or groundspeed based, the bank angle can again be subscripted by an a or a g.

Body to aerodynamic frame

This transformation matrix is defined as

$$\mathbf{C_{A,B}} = \mathbf{C_3}\left(\beta\right)\mathbf{C_2}\left(-\alpha\right) = \begin{bmatrix} \mathrm{c}\alpha\mathrm{c}\beta & \mathrm{s}\beta & \mathrm{s}\alpha\mathrm{c}\beta \\ -\mathrm{c}\alpha\mathrm{s}\beta & \mathrm{c}\beta & -\mathrm{s}\alpha\mathrm{s}\beta \\ -\mathrm{s}\alpha & 0 & \mathrm{c}\alpha \end{bmatrix} \tag{A.6}$$

where α is the angle of attack and β is the angle of sideslip, both expressed in radians. Note that this transformation is either airspeed based or groundspeed based, both angles can again be subscripted by an a or a g.

Body to vertical frame

This transformation matrix is defined as

$$\mathbf{C_{V,B}} = \mathbf{C_3}\left(-\psi\right)\mathbf{C_2}\left(-\theta\right)\mathbf{C_1}\left(-\phi\right) = \begin{bmatrix} \mathrm{c}\theta\mathrm{c}\psi & \mathrm{s}\phi\mathrm{s}\theta\mathrm{c}\psi - \mathrm{c}\phi\mathrm{s}\psi & \mathrm{c}\phi\mathrm{s}\theta\mathrm{c}\psi + \mathrm{s}\phi\mathrm{s}\psi \\ \mathrm{c}\theta\mathrm{s}\psi & \mathrm{s}\phi\mathrm{s}\theta\mathrm{s}\psi + \mathrm{c}\phi\mathrm{c}\psi & \mathrm{c}\phi\mathrm{s}\theta\mathrm{s}\psi - \mathrm{s}\phi\mathrm{c}\psi \\ -\mathrm{s}\theta & \mathrm{s}\phi\mathrm{c}\theta & \mathrm{c}\phi\mathrm{c}\theta \end{bmatrix} \tag{A.7}$$

where ϕ, θ and ψ are the roll, pitch and yaw angles respectively, all expressed in radians.

Additional frame transformations

This paragraph contains the transformations that are constructed by multiplication of the previously stated matrices.

$$\begin{aligned} \mathbf{C_{V,A}} &= \mathbf{C_{V,T}}\mathbf{C_{T,A}} \\ &= \mathbf{C_3}\left(-\chi\right)\mathbf{C_2}\left(-\gamma\right)\mathbf{C_1}\left(\sigma\right) \end{aligned} \tag{A.8}$$

$$\begin{aligned} \mathbf{C_{R,A}} &= \mathbf{C_{R,V}}\mathbf{C_{V,A}} \\ &= \mathbf{C_3}\left(-\tau\right)\mathbf{C_2}\left(\frac{\pi}{2}+\delta\right)\mathbf{C_3}\left(-\chi\right)\mathbf{C_2}\left(-\gamma\right)\mathbf{C_1}\left(\sigma\right) \end{aligned} \tag{A.9}$$

$$\begin{aligned} \mathbf{C_{R,B}} &= \mathbf{C_{R,V}}\mathbf{C_{V,A}}\mathbf{C_{A,B}} \\ &= \mathbf{C_3}\left(-\tau\right)\mathbf{C_2}\left(\frac{\pi}{2}+\delta\right)\mathbf{C_3}\left(-\chi\right)\mathbf{C_2}\left(-\gamma\right)\mathbf{C_1}\left(\sigma\right)\mathbf{C_3}\left(\beta\right)\mathbf{C_2}\left(-\alpha\right) \end{aligned} \tag{A.10}$$

$$\begin{aligned} \mathbf{C_{I,B}} &= \mathbf{C_{R,V}}\mathbf{C_{V,A}}\mathbf{C_{A,B}} \\ &= \mathbf{C_3}\left(-\tilde{\tau}\right)\mathbf{C_2}\left(\frac{\pi}{2}+\delta\right)\mathbf{C_3}\left(-\chi\right)\mathbf{C_2}\left(-\gamma\right)\mathbf{C_1}\left(\sigma\right)\mathbf{C_3}\left(\beta\right)\mathbf{C_2}\left(-\alpha\right) \end{aligned} \tag{A.11}$$

Aerodynamic angles to classical Euler angles

For the conversion from aerodynamic angles to classical Euler angles, it is assumed that the aerodynamic angles α, β and σ, as well as the heading and flight path angles χ and γ are known. Then, two similar transformation matrices are defined that represent the transformation from body to vertical frame, which will be denoted by $\mathbf{C}$ in this paragraph:

$$\mathbf{C_3}\left(-\psi\right)\mathbf{C_2}\left(-\theta\right)\mathbf{C_1}\left(-\phi\right) = \mathbf{C_3}\left(-\chi\right)\mathbf{C_2}\left(-\gamma\right)\mathbf{C_1}\left(\sigma\right)\mathbf{C_3}\left(\beta\right)\mathbf{C_2}\left(-\alpha\right) \tag{A.12}$$

The left-hand side of this equation was defined in Equation (A.7) and from this equation it can be stated that:

$$\phi = \text{atan2}\left(\mathbf{C}\left(3,2\right),\mathbf{C}\left(3,3\right)\right) \tag{A.13a}$$
$$\theta = \text{asin}\left(-\mathbf{C}\left(3,1\right)\right) \tag{A.13b}$$
$$\psi = \text{atan2}\left(\mathbf{C}\left(2,1\right),\mathbf{C}\left(1,1\right)\right) \tag{A.13c}$$

The transformation matrix entries that are stated in Equation (A.13) are computed via the right-hand side of Equation (A.12).

$$\mathbf{C}\left(1,1\right) = \text{c}\chi\left[\text{c}\gamma\text{c}\alpha\text{c}\beta + \text{s}\gamma\left(\text{c}\alpha\text{s}\beta\text{s}\sigma - \text{s}\alpha\text{c}\sigma\right)\right] + \text{s}\chi\left(\text{c}\alpha\text{s}\beta\text{c}\sigma + \text{s}\alpha\text{s}\sigma\right) \tag{A.14a}$$
$$\mathbf{C}\left(2,1\right) = \text{s}\chi\left[\text{c}\gamma\text{c}\alpha\text{c}\beta + \text{s}\gamma\left(\text{c}\alpha\text{s}\beta\text{s}\sigma - \text{s}\alpha\text{c}\sigma\right)\right] - \text{c}\chi\left(\text{c}\alpha\text{s}\beta\text{c}\sigma + \text{s}\alpha\text{s}\sigma\right) \tag{A.14b}$$
$$\mathbf{C}\left(3,1\right) = -\text{s}\gamma\text{c}\alpha\text{c}\beta + \text{c}\gamma\left(\text{c}\alpha\text{s}\beta\text{s}\sigma - \text{s}\alpha\text{c}\sigma\right) \tag{A.14c}$$
$$\mathbf{C}\left(3,2\right) = -\text{s}\gamma\text{s}\beta - \text{c}\gamma\text{c}\beta\text{s}\sigma \tag{A.14d}$$
$$\mathbf{C}\left(3,3\right) = -\text{s}\gamma\text{s}\alpha\text{c}\beta + \text{c}\gamma\left(\text{s}\alpha\text{s}\beta\text{s}\sigma + \text{c}\alpha\text{c}\sigma\right) \tag{A.14e}$$

Classical Euler angles to aerodynamic angles

Similarly as for the conversion from aerodynamic angles to classical Euler angles the conversion from classical Euler angles to aerodynamic angles can be defined. Now it is assumed that the heading and flight path angles χ and γ are known, as well as the roll, pitch and yaw angles (ϕ, θ and ψ respectively). A relation similar to Equation (A.12) can be defined:

$$\mathbf{C_1}\left(\sigma\right)\mathbf{C_3}\left(\beta\right)\mathbf{C_2}\left(-\alpha\right) = \mathbf{C_2}\left(\gamma\right)\mathbf{C_3}\left(\chi\right)\mathbf{C_3}\left(-\psi\right)\mathbf{C_2}\left(-\theta\right)\mathbf{C_1}\left(-\phi\right) \tag{A.15}$$

The left-hand side of Equation (A.15) can be written as

$$\mathbf{C} = \begin{bmatrix} \text{c}\alpha\text{c}\beta & \text{s}\beta & \text{s}\alpha\text{c}\beta \\ -\text{c}\alpha\text{s}\beta\text{c}\sigma - \text{s}\alpha\text{s}\sigma & \text{c}\beta\text{c}\sigma & -\text{s}\alpha\text{s}\beta\text{c}\sigma + \text{c}\alpha\text{s}\sigma \\ \text{c}\alpha\text{s}\beta\text{s}\sigma - \text{s}\alpha\text{c}\sigma & -\text{c}\beta\text{s}\sigma & \text{s}\alpha\text{s}\beta\text{s}\sigma + \text{c}\alpha\text{c}\sigma \end{bmatrix} \tag{A.16}$$

which leads to:

$$\alpha = \text{atan2}\left(\mathbf{C}\left(1,3\right),\mathbf{C}\left(1,1\right)\right) \tag{A.17a}$$
$$\beta = \text{asin}\left(-\mathbf{C}\left(1,2\right)\right) \tag{A.17b}$$
$$\sigma = \text{atan2}\left(\mathbf{C}\left(3,2\right),\mathbf{C}\left(2,2\right)\right) \tag{A.17c}$$

The transformation matrix entries that are stated in Equation (A.17) are computed via the right-hand side of Equation (A.15).

$$\mathbf{C}\left(1,1\right) = \text{c}\gamma\left(\text{c}\chi\text{c}\psi\text{c}\theta + \text{s}\chi\text{s}\psi\text{c}\theta\right) + \text{s}\gamma\text{s}\theta \tag{A.18a}$$
$$\mathbf{C}\left(1,2\right) = \text{c}\gamma\left[\text{c}\chi\left(\text{c}\psi\text{s}\theta\text{s}\phi - \text{s}\psi\text{c}\phi\right) + \text{s}\chi\left(\text{s}\psi\text{s}\theta\text{s}\phi + \text{c}\psi\text{c}\phi\right)\right] - \text{s}\gamma\text{c}\theta\text{s}\phi \tag{A.18b}$$
$$\mathbf{C}\left(1,3\right) = \text{c}\gamma\left[\text{c}\chi\left(\text{c}\psi\text{s}\theta\text{c}\phi + \text{s}\psi\text{s}\phi\right) + \text{s}\chi\left(\text{s}\psi\text{s}\theta\text{c}\phi - \text{c}\psi\text{s}\phi\right)\right] - \text{s}\gamma\text{c}\theta\text{c}\phi \tag{A.18c}$$
$$\mathbf{C}\left(2,2\right) = \text{s}\chi\left(\text{c}\psi\text{s}\theta\text{s}\phi - \text{s}\psi\text{c}\phi\right) + \text{c}\chi\left(\text{s}\psi\text{s}\theta\text{s}\phi + \text{c}\psi\text{c}\phi\right) \tag{A.18d}$$
$$\mathbf{C}\left(3,2\right) = \text{s}\gamma\left[\text{c}\chi\left(\text{c}\psi\text{s}\theta\text{s}\phi - \text{s}\psi\text{c}\phi\right) + \text{s}\chi\left(\text{s}\psi\text{s}\theta\text{s}\phi + \text{c}\psi\text{c}\phi\right)\right] + \text{c}\gamma\text{c}\theta\text{s}\phi \tag{A.18e}$$

A.2 State Variable Transformations

In this section the transformation from the Cartesian system to the spherical system and vice versa is defined, as well as the transformations between quaternions and Euler angles.

Cartesian to spherical components

This transformation is shown using the expressions for the position vector. For the velocity vector, the transformation is similar.

$$R = \sqrt{x_R^2 + y_R^2 + z_R^2} \tag{A.19a}$$

$$\tau = \text{atan}\left(\frac{y_R}{x_R}\right) \tag{A.19b}$$

$$\delta = \text{atan}\left(\frac{z_R}{R}\right) \tag{A.19c}$$

Spherical to Cartesian components

This transformation is shown using the expressions for the position vector. For the velocity vector, the transformation is similar.

$$x_R = R\cos\delta\cos\tau \tag{A.20a}$$

$$y_R = R\cos\delta\sin\tau \tag{A.20b}$$

$$z_R = R\sin\delta \tag{A.20c}$$

Euler angles to quaternions

The conversion from Euler angles to quaternions is found by comparing the $\mathbf{C}_{I,B}$ matrices, which can be defined by both. Quaternions can then be computed by

$$\mathbf{Q} = \begin{pmatrix} \text{s}\frac{\phi}{2}\text{c}\frac{\theta}{2}\text{c}\frac{\psi}{2} - \text{c}\frac{\phi}{2}\text{s}\frac{\theta}{2}\text{s}\frac{\psi}{2} \\ \text{c}\frac{\phi}{2}\text{s}\frac{\theta}{2}\text{c}\frac{\psi}{2} + \text{s}\frac{\phi}{2}\text{c}\frac{\theta}{2}\text{s}\frac{\psi}{2} \\ \text{c}\frac{\phi}{2}\text{c}\frac{\theta}{2}\text{s}\frac{\psi}{2} - \text{s}\frac{\phi}{2}\text{s}\frac{\theta}{2}\text{c}\frac{\psi}{2} \\ \text{c}\frac{\phi}{2}\text{c}\frac{\theta}{2}\text{c}\frac{\psi}{2} + \text{s}\frac{\phi}{2}\text{s}\frac{\theta}{2}\text{s}\frac{\psi}{2} \end{pmatrix} \tag{A.21}$$

where the fourth element defines the scalar component.

Quaternions to Euler angles

By comparison of the transformation matrices between the inertial and the body frame, also the conversion from quaternions to Euler angles can be found:

$$\begin{pmatrix} \phi \\ \theta \\ \psi \end{pmatrix} = \begin{pmatrix} \text{atan2}\left(2\left[Q_1Q_4 + Q_2Q_3\right], 1 - 2\left[Q_1^2 + Q_2^2\right]\right) \\ \text{asin}\left(2\left[Q_2Q_4 - Q_1Q_3\right]\right) \\ \text{atan2}\left(2\left[Q_1Q_2 + Q_3Q_4\right], 1 - 2\left[Q_2^2 + Q_3^2\right]\right) \end{pmatrix} \tag{A.22}$$

A.3 HORUS 2B Characteristics

In this section the main characteristics of the HORUS 2B reference vehicle are mentioned. It is split in different tables for the engine, mass and inertia, and geometry properties.

Table A.1: Engine properties of the HORUS 2B reference vehicle (MBB, 1988 and Mooij, 1995)

Parameter	**Value**	**Unit**
Main propulsion engine (HM60) (vacuum)	1055	kN
Injection/deorbit-burn engines	$2 \cdot 4$	kN
Attitude control/OCS engines	$14 \cdot 400$	N
Fine attitude control/RCS engines	$18 \cdot 10$	N

Table A.2: Mass and inertia properties of the HORUS 2B reference vehicle (MBB, 1988 and Mooij, 1995)

Parameter	**Value**	**Unit**
Structure	13140	kg
Propulsion	2148	kg
Equipment	2775	kg
Margin	331	kg
Empty mass	18394	kg
Residual propellant	305	kg
Payload (100%)	7000	kg
Reserve propellant	330	kg
Landing mass	26029	kg
Propellants	30167	kg
Gross mass	56196	kg
I_{xx}	119605	kg m^2
I_{yy}	769000	kg m^2
I_{zz}	805395	kg m^2
I_{xy}	0	kg m^2
I_{xz}	-20372	kg m^2
I_{yz}	0	kg m^2

Table A.3: Geometry properties of the HORUS 2B reference vehicle (MBB, 1988 and Mooij, 1995)

Parameter	**Value**	**Unit**
Total length	25.0	m
Fuselage length	23.0	m
Wing span	13.0	m
Height	4.5	m
Exposed wing area	43.0	m^2
Projected fuselage area	101.0	m^2
Total plan view area	151.0	m^2
Reference area	110.0	m^2
Reference length pitch	23.0	m
Reference length roll/yaw	13.0	m
Winglet area	$2 \cdot 12.0$	m^2
Rudder area	$2 \cdot 3.1$	m^2
Elevon area	$2 \cdot 4.6$	m^2
Body flap area	8.5	m^2
Body maximum width	5.4	m
Body maximum height	4.5	m
Longitudinal location of CoM (from nose)	14	m
Coordinates of RCS thrusters		
Roll (vertical line z-line of thrust)		
2 forward thrusters	$x = 20.0$	m
	$y = 2.0$	m
2 rear thrusters	$x = 20.0$	m
	$y = -2.0$	m
Pitch (vertical line z-line of thrust)		
2 forward thrusters	$x = 1.0$	m
4 rear thrusters	$x = 20.0$	m
Yaw (horizontal y-line of thrust)		
2 forward thrusters	$x = 1.0$	m
	$z = 0.3$	m
2 rear thrusters	$x = 20.0$	m
	0.0	m

B

Partial Derivatives

This appendix contains the elements of the measurement matrix $\mathbf{H}_k$ that is used in the implementation of the EKF. These elements consist of the partial derivatives of the measurement equations with respect to the state variables that were used:

$$\mathbf{H}_k = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \tag{B.1}$$

This matrix can be written down as:

$\mathbf{H}_k =$

$$\begin{bmatrix}
\frac{\partial \tilde{\rho}_i}{\partial x} & \frac{\partial \tilde{\rho}_i}{\partial y} & \frac{\partial \tilde{\rho}_i}{\partial z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial \tilde{\rho}_i}{\partial b_c} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{\partial \dot{\tilde{\rho}}_i}{\partial x} & \frac{\partial \dot{\tilde{\rho}}_i}{\partial y} & \frac{\partial \dot{\tilde{\rho}}_i}{\partial z} & \frac{\partial \dot{\tilde{\rho}}_i}{\partial \dot{x}} & \frac{\partial \dot{\tilde{\rho}}_i}{\partial \dot{y}} & \frac{\partial \dot{\tilde{\rho}}_i}{\partial \dot{z}} & 0 & 0 & 0 & 0 & 0 & \frac{\partial \dot{\tilde{\rho}}_i}{\partial d_c} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{\partial \Delta\Phi_{i1}}{\partial x} & \frac{\partial \Delta\Phi_{i1}}{\partial y} & \frac{\partial \Delta\Phi_{i1}}{\partial z} & 0 & 0 & 0 & \frac{\partial \Delta\Phi_{i1}}{\partial Q_1} & \frac{\partial \Delta\Phi_{i1}}{\partial Q_2} & \frac{\partial \Delta\Phi_{i1}}{\partial Q_3} & \frac{\partial \Delta\Phi_{i1}}{\partial Q_4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{\partial \Delta\Phi_{i2}}{\partial x} & \frac{\partial \Delta\Phi_{i2}}{\partial y} & \frac{\partial \Delta\Phi_{i2}}{\partial z} & 0 & 0 & 0 & \frac{\partial \Delta\Phi_{i2}}{\partial Q_1} & \frac{\partial \Delta\Phi_{i2}}{\partial Q_2} & \frac{\partial \Delta\Phi_{i2}}{\partial Q_3} & \frac{\partial \Delta\Phi_{i2}}{\partial Q_4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{\partial \Delta\Phi_{i3}}{\partial x} & \frac{\partial \Delta\Phi_{i3}}{\partial y} & \frac{\partial \Delta\Phi_{i3}}{\partial z} & 0 & 0 & 0 & \frac{\partial \Delta\Phi_{i3}}{\partial Q_1} & \frac{\partial \Delta\Phi_{i3}}{\partial Q_2} & \frac{\partial \Delta\Phi_{i3}}{\partial Q_3} & \frac{\partial \Delta\Phi_{i3}}{\partial Q_4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\vdots & & & & & & & & & & & & & & & & & & \vdots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial a_x}{\partial s_x} & \frac{\partial a_x}{\partial b_x} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial a_y}{\partial b_y} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial a_z}{\partial b_z} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial \omega_p}{\partial b_p} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial \omega_q}{\partial b_q} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial \omega_r}{\partial b_r} \\
\frac{\partial p_k}{\partial x} & \frac{\partial p_k}{\partial y} & \frac{\partial p_k}{\partial z} & \frac{\partial p_k}{\partial \dot{x}} & \frac{\partial p_k}{\partial \dot{y}} & \frac{\partial p_k}{\partial \dot{z}} & \frac{\partial p_k}{\partial Q_1} & \frac{\partial p_k}{\partial Q_2} & \frac{\partial p_k}{\partial Q_3} & \frac{\partial p_k}{\partial Q_4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\vdots & & & & & & & & & & & & & & & & & & \vdots
\end{bmatrix} \tag{B.2}$$

Then, in this appendix, the components x, y, and z refer to the position elements in the I-frame, and their time derivatives $\dot{x}$, $\dot{y}$, and $\dot{z}$ denote velocity in the inertial frame. In the derivation of the FADS partial derivatives the NED-notation (North-East- Down notation) is used, which is computed by transforming the inertial velocities to the V-frame and correcting for the rotation of the Earth.

Pseudorange

This paragraph contains the partial derivatives of the pseudorange measurements from the GPS.

$$\frac{\partial \tilde{\rho}}{\partial x} = \frac{x - x_{\text{GPS,i}}}{\tilde{\rho}} \tag{B.3}$$

$$\frac{\partial \tilde{\rho}}{\partial y} = \frac{y - y_{\text{GPS,i}}}{\tilde{\rho}} \tag{B.4}$$

$$\frac{\partial \tilde{\rho}}{\partial z} = \frac{z - z_{\text{GPS,i}}}{\tilde{\rho}} \tag{B.5}$$

$$\frac{\partial \tilde{\rho}}{\partial b_c} = c \tag{B.6}$$

All other partial derivatives of the pseudorange are zero.

Range Rate

This paragraph contains the partial derivatives of the range rate measurements from the GPS. The actual distance between the vehicle and GPS satellite i is:

$$\rho = \sqrt{\left(x_{\text{GPS,i}} - x\right)^2 + \left(y_{\text{GPS,i}} - y\right)^2 + \left(z_{\text{GPS,i}} - z\right)^2} \tag{B.7}$$

Then, it is found that:

$$\frac{\partial \dot{\tilde{\rho}}}{\partial x} = \frac{\dot{x} - \dot{x}_{\text{GPS,i}}}{\rho} + \frac{\left(x_{\text{GPS,i}} - x\right)\left[\left(x_{\text{GPS,i}} - x\right)\left(\dot{x}_{\text{GPS,i}} - \dot{x}\right) + \left(y_{\text{GPS,i}} - y\right)\left(\dot{y}_{\text{GPS,i}} - \dot{y}\right) + \left(z_{\text{GPS,i}} - z\right)\left(\dot{z}_{\text{GPS,i}} - \dot{z}\right)\right]}{\rho^3} \tag{B.8}$$

$$\frac{\partial \dot{\tilde{\rho}}}{\partial y} = \frac{\dot{y} - \dot{y}_{\text{GPS,i}}}{\rho} + \frac{\left(y_{\text{GPS,i}} - y\right)\left[\left(x_{\text{GPS,i}} - x\right)\left(\dot{x}_{\text{GPS,i}} - \dot{x}\right) + \left(y_{\text{GPS,i}} - y\right)\left(\dot{y}_{\text{GPS,i}} - \dot{y}\right) + \left(z_{\text{GPS,i}} - z\right)\left(\dot{z}_{\text{GPS,i}} - \dot{z}\right)\right]}{\rho^3} \tag{B.9}$$

$$\frac{\partial \dot{\tilde{\rho}}}{\partial z} = \frac{\dot{z} - \dot{z}_{\text{GPS,i}}}{\rho} + \frac{\left(z_{\text{GPS,i}} - z\right)\left[\left(x_{\text{GPS,i}} - x\right)\left(\dot{x}_{\text{GPS,i}} - \dot{x}\right) + \left(y_{\text{GPS,i}} - y\right)\left(\dot{y}_{\text{GPS,i}} - \dot{y}\right) + \left(z_{\text{GPS,i}} - z\right)\left(\dot{z}_{\text{GPS,i}} - \dot{z}\right)\right]}{\rho^3} \tag{B.10}$$

$$\frac{\partial \dot{\tilde{\rho}}}{\partial \dot{x}} = \frac{x - x_{\text{GPS,i}}}{\rho} \tag{B.11}$$

$$\frac{\partial \dot{\tilde{\rho}}}{\partial \dot{y}} = \frac{y - y_{\text{GPS,i}}}{\rho} \tag{B.12}$$

$$\frac{\partial \dot{\tilde{\rho}}}{\partial \dot{z}} = \frac{z - z_{\text{GPS,i}}}{\rho} \tag{B.13}$$

$$\frac{\partial \dot{\tilde{\rho}}}{\partial d_c} = c \tag{B.14}$$

All other partial derivatives of the range rate are zero.

Carrier Phase Difference

This paragraph contains the partial derivatives of the carrier phase difference measurements from the GPS. For the calculation of these derivatives it was defined that $\mathbf{b}_{ij}$ was the baseline vector and defines the distance between slave antenna j and the master antenna. Next to that $\mathbf{C}_{B,I}$ defines the transformation matrix from the inertial to the body frame, and $\mathbf{e}_{i,I}$ defines the line-of-sight unit vector of the vehicle with respect to GPS satellite i. Then, the following partial derivatives can be computed:

$$\frac{\partial \Delta\Phi_{ij}}{\partial x} = -\frac{f_s}{c}\mathbf{b}_{ij}^T\mathbf{C}_{B,I}\left\{\begin{pmatrix} -\frac{1}{\rho} \\ 0 \\ 0 \end{pmatrix} + \frac{x_{\text{GPS,i}} - x}{\rho^3}\begin{pmatrix} x_{\text{GPS,i}} - x \\ y_{\text{GPS,i}} - y \\ z_{\text{GPS,i}} - z \end{pmatrix}\right\} \tag{B.15}$$

$$\frac{\partial \Delta\Phi_{ij}}{\partial y} = -\frac{f_s}{c}\mathbf{b}_{ij}^T\mathbf{C}_{B,I}\left\{\begin{pmatrix} 0 \\ -\frac{1}{\rho} \\ 0 \end{pmatrix} + \frac{y_{\text{GPS,i}} - y}{\rho^3}\begin{pmatrix} x_{\text{GPS,i}} - x \\ y_{\text{GPS,i}} - y \\ z_{\text{GPS,i}} - z \end{pmatrix}\right\} \tag{B.16}$$

$$\frac{\partial \Delta\Phi_{ij}}{\partial z} = -\frac{f_s}{c}\mathbf{b}_{ij}^T\mathbf{C}_{B,I}\left\{\begin{pmatrix} 0 \\ 0 \\ -\frac{1}{\rho} \end{pmatrix} + \frac{z_{\text{GPS,i}} - z}{\rho^3}\begin{pmatrix} x_{\text{GPS,i}} - x \\ y_{\text{GPS,i}} - y \\ z_{\text{GPS,i}} - z \end{pmatrix}\right\} \tag{B.17}$$

$$\frac{\partial \Delta\Phi_{ij}}{\partial Q_1} = -\frac{2f_s}{c}\mathbf{b}_{ij}^T\begin{bmatrix} Q_1 & Q_2 & Q_3 \\ Q_2 & -Q_1 & -Q_4 \\ Q_3 & Q_4 & -Q_1 \end{bmatrix}\mathbf{e}_{i,I} \tag{B.18}$$

$$\frac{\partial \Delta\Phi_{ij}}{\partial Q_2} = -\frac{2f_s}{c}\mathbf{b}_{ij}^T\begin{bmatrix} -Q_2 & Q_1 & Q_4 \\ Q_1 & Q_2 & Q_3 \\ -Q_4 & Q_3 & -Q_2 \end{bmatrix}\mathbf{e}_{i,I} \tag{B.19}$$

$$\frac{\partial \Delta\Phi_{ij}}{\partial Q_3} = -\frac{2f_s}{c}\mathbf{b}_{ij}^T \begin{bmatrix} -Q_3 & -Q_4 & Q_1 \\ Q_4 & -Q_3 & Q_2 \\ Q_1 & Q_2 & Q_3 \end{bmatrix} \mathbf{e}_{i,I} \tag{B.20}$$

$$\frac{\partial \Delta\Phi_{ij}}{\partial Q_4} = -\frac{2f_s}{c}\mathbf{b}_{ij}^T \begin{bmatrix} Q_4 & -Q_3 & Q_2 \\ Q_3 & Q_4 & -Q_1 \\ -Q_2 & Q_1 & Q_4 \end{bmatrix} \mathbf{e}_{i,I} \tag{B.21}$$

All other partial derivatives of the carrier phase differences are zero.

Acceleration

This paragraph contains the partial derivatives of the acceleration measurements from the IMU.

$$\frac{\partial \mathbf{a}_m}{\partial s_x} = \begin{pmatrix} a_x \\ 0 \\ 0 \end{pmatrix}, \frac{\partial \mathbf{a}_m}{\partial b_x} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \frac{\partial \mathbf{a}_m}{\partial b_y} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \frac{\partial \mathbf{a}_m}{\partial b_z} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \tag{B.22}$$

All other partial derivatives of the accelerations are zero.

Rotational Rate

This paragraph contains the partial derivatives of the rotational rate measurements from the IMU.

$$\frac{\partial \boldsymbol{\omega}_m}{\partial b_p} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \frac{\partial \boldsymbol{\omega}_m}{\partial b_q} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \frac{\partial \boldsymbol{\omega}_m}{\partial b_r} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \tag{B.23}$$

All other partial derivatives of the rotational rates are zero.

Pressure

This paragraph contains the partial derivatives of the pressure measurements from the FADS. It was seen that these measurements depend on the dynamic impact pressure q_c, the incidence angle θ, and the free stream air pressure p_∞. For convenience, this section is divided into three parts, each describing the partial derivatives of the aforementioned variables. The partial derivatives of the FADS pressure are then computed by

$$\frac{\partial p_m}{\partial x} = \frac{\partial q_c}{\partial x}\left[\cos^2\theta + \epsilon \sin^2\theta\right] + q_c \frac{\partial \theta}{\partial x}\left[(2\epsilon - 2)\cos\theta \sin\theta\right] + \frac{\partial p_\infty}{\partial x} \tag{B.24}$$

which is similar for all state variables.

Free Stream Air Pressure

The free stream air pressure p_∞ depends on the atmospheric region, as was described in Section 4.2. The region in which the vehicle is located is only related to the altitude of the vehicle, hence the partial derivatives of p_∞ with respect to velocity, attitude, and GPS and IMU errors are zero. The altitude is defined as

$$h = \sqrt{x^2 + y^2 + z^2} - R_E \tag{B.25}$$

with

$$\frac{\partial h}{\partial x} = \frac{x}{h} \tag{B.26}$$

which is similar for y and z. The following derivatives are then calculated with respect to altitude, with:

$$\frac{\partial p_\infty}{\partial x} = \frac{\partial p_\infty}{\partial h}\frac{\partial h}{\partial x} \tag{B.27}$$

For regions with $\partial T/\partial h = 0$ it is then derived that

$$\frac{\partial p_\infty}{\partial h} = -p_0 \frac{g}{RT_0}\frac{R_E}{R_E - h} e^{-\frac{g}{RT_0}\left[\frac{R_E^2(h-h_0)}{(R_E-h)(R_E-h_0)}\right]} \tag{B.28}$$

and for regions where $\partial T/\partial h \neq 0$ it is obtained that

$$\frac{\partial p_\infty}{\partial h} = -p_0 \frac{g}{RT_\infty} \frac{R_E}{R_E - h} \left(\frac{T_\infty}{T_0} \right)^{-\frac{g}{R\partial T/\partial h}} \tag{B.29}$$

where all variables with subscript 0 denote the boundary values of that particular atmospheric region, which can be found in NASA (1976). The gravitational acceleration is assumed to be constant, as was also done in the atmospheric model that was implemented in the simulator. As the partial derivative of the radial gravitational acceleration with respect to the altitude is sufficiently smaller than the free stream air pressure derivative calculated above, it can be neglected.

Dynamic Impact Pressure

For the dynamic impact pressure, the partial derivatives are computed to be

$$\begin{aligned} \frac{\partial q_c}{\partial x} = &\left(166.92 M_\infty^2 \frac{\partial p_\infty}{\partial x} + 333.84 p_\infty M_\infty \frac{\partial M_\infty}{\partial x} \right) \left[\frac{M_\infty^2}{7M_\infty^2 - 1} \right]^{2.5} \\ &+ 417.3 p_\infty M_\infty^2 \frac{\partial M_\infty}{\partial x} \left(\frac{2M_\infty}{7N_\infty^2 - 1} - \frac{14M_\infty^3}{\left(7M_\infty^2 - 1\right)^2} \right) \left[\frac{M_\infty^2}{7M_\infty^2 - 1} \right]^{1.5} - \frac{\partial p_\infty}{\partial x} \end{aligned} \tag{B.30}$$

which is similar for all state variables, although some partial derivatives found in this equation may be found to be zero. The equations for the free stream pressure were handled in the previous subsection. The Mach number then can be differentiated to:

$$\frac{\partial M_\infty}{\partial x} = \frac{1}{a} \frac{\partial V}{\partial x} - \frac{V}{a^2} \frac{\partial a}{\partial x} \tag{B.31}$$

which is also similar for all state variables. The rightmost term containing the partial derivative of the speed of sound is only dependent on altitude, and thus on position:

$$\frac{\partial a}{\partial x} = \frac{1}{2a} \frac{\partial T_\infty}{\partial x} \tag{B.32}$$

The temperature derivative with respect to a position coordinate is then derived to be:

$$\frac{\partial T_\infty}{\partial x} = \partial T / \partial h \frac{x R_E}{h (R_E - h)} \tag{B.33}$$

where $\partial T/\partial h$ depends on the current atmospheric region and is found in Table 4.2. The velocity norm consists of a position term and a velocity term, which are transformed from the I-frame to the V-frame. It is then found that:

$$\frac{\partial V}{\partial x} = \frac{\omega_{cb}^2 x + \omega_{cb} \dot{y}}{V} \tag{B.34}$$

$$\frac{\partial V}{\partial y} = \frac{\omega_{cb}^2 y - \omega_{cb} \dot{x}}{V} \tag{B.35}$$

The component in the z-direction is zero, as this axis is oriented perpendicular to the plane of rotation. The partial derivatives with respect to the individual velocity components are:

$$\frac{\partial V}{\partial \dot{x}} = \frac{\dot{x} - \omega_{cb} y}{V} \tag{B.36}$$

$$\frac{\partial V}{\partial \dot{y}} = \frac{\dot{y} + \omega_{cb} x}{V} \tag{B.37}$$

$$\frac{\partial V}{\partial \dot{z}} = \frac{\dot{z}}{V} \tag{B.38}$$

The velocity norm was corrected for the rotation of the Earth and is defined as:

$$V = \sqrt{\left(\dot{x} - \omega_{cb} y \right)^2 + \left(\dot{y} + \omega_{cb} x \right)^2 + \dot{z}^2} \tag{B.39}$$

Using these equations the partial derivatives of the Mach number can be computed, which are then substituted in Equation (B.30).

Incidence Angle

The incidence angle will be broken down in multiple partial derivatives, similar to the dynamic impact pressure. It is found that

$$\frac{\partial\theta}{\partial x} = \frac{\frac{\partial\alpha}{\partial x}\left(\mathrm{s}\alpha\mathrm{c}\beta\mathrm{c}\lambda - \mathrm{c}\alpha\mathrm{c}\beta\mathrm{c}\phi\mathrm{s}\lambda\right) + \frac{\partial\beta}{\partial x}\left(\mathrm{c}\alpha\mathrm{s}\beta\mathrm{c}\lambda - \mathrm{c}\beta\mathrm{s}\phi\mathrm{s}\lambda + \mathrm{s}\alpha\mathrm{s}\beta\mathrm{c}\phi\mathrm{s}\lambda\right)}{\sqrt{1-\left(\mathrm{c}\alpha\mathrm{c}\beta\mathrm{c}\lambda + \mathrm{s}\beta\mathrm{s}\phi\mathrm{s}\lambda + \mathrm{s}\alpha\mathrm{c}\beta\mathrm{c}\phi\mathrm{s}\lambda\right)^2}} \tag{B.40}$$

which contains the clock and cone angles ϕ and λ, and is similar for all state variables. The angle of attack and angle of sideslip are computed via a transformation matrix $\mathbf{C}$ that is defined as:

$$\mathbf{C} = \mathbf{C}_2\left(\gamma\right)\mathbf{C}_3\left(\chi\right)\mathbf{C}_2\left(-\frac{\pi}{2}-\delta\right)\mathbf{C}_3\left(\tilde{\tau}\right)\mathbf{C}_{B,I} \tag{B.41}$$

The partial derivatives of α and β are then defined as:

$$\frac{\partial\alpha}{\partial x} = \frac{\mathbf{C}_{11}^2}{\mathbf{C}_{13}^2+\mathbf{C}_{11}^2}\left(\frac{1}{\mathbf{C}_{11}}\frac{\partial\mathbf{C}_{13}}{\partial x} - \frac{\mathbf{C}_{13}}{\mathbf{C}_{11}^2}\frac{\partial\mathbf{C}_{11}}{\partial x}\right) \tag{B.42}$$

$$\frac{\partial\beta}{\partial x} = \frac{1}{\sqrt{1-\mathbf{C}_{12}^2}}\frac{\partial\mathbf{C}_{12}}{\partial x} \tag{B.43}$$

where the subscript of the transformation matrix $\mathbf{C}$ denotes the element of the matrix. Again, these two equations are the same for all state variables. Finally, the partial derivatives of $\mathbf{C}$ are to be defined. In the carrier phase difference equations, the derivatives of $\mathbf{C}_{B,I}$ were already used, so they will not be stated in this section. The derivatives of the remaining four transformation matrices however still have to be derived

$$\begin{aligned}\frac{\partial\mathbf{C}}{\partial x} = {} & \frac{\partial\mathbf{C}_2\left(\gamma\right)}{\partial x}\mathbf{C}_3\left(\chi\right)\mathbf{C}_2\left(-\frac{\pi}{2}-\delta\right)\mathbf{C}_3\left(\tilde{\tau}\right)\mathbf{C}_{B,I} + \mathbf{C}_2\left(\gamma\right)\frac{\partial\mathbf{C}_3\left(\chi\right)}{\partial x}\mathbf{C}_2\left(-\frac{\pi}{2}-\delta\right)\mathbf{C}_3\left(\tilde{\tau}\right)\mathbf{C}_{B,I} \\ & + \mathbf{C}_2\left(\gamma\right)\mathbf{C}_3\left(\chi\right)\frac{\partial\mathbf{C}_2\left(-\frac{\pi}{2}-\delta\right)}{\partial x}\mathbf{C}_3\left(\tilde{\tau}\right)\mathbf{C}_{B,I} + \mathbf{C}_2\left(\gamma\right)\mathbf{C}_3\left(\chi\right)\mathbf{C}_2\left(-\frac{\pi}{2}-\delta\right)\frac{\partial\mathbf{C}_3\left(\tilde{\tau}\right)}{\partial x}\mathbf{C}_{B,I} \\ & + \mathbf{C}_2\left(\gamma\right)\mathbf{C}_3\left(\chi\right)\mathbf{C}_2\left(-\frac{\pi}{2}-\delta\right)\mathbf{C}_3\left(\tilde{\tau}\right)\frac{\partial\mathbf{C}_{B,I}}{\partial x}\end{aligned} \tag{B.44}$$

with:

$$\frac{\partial\mathbf{C}_2\left(\gamma\right)}{\partial x} = \frac{\partial\gamma}{\partial x}\begin{bmatrix}-\mathrm{s}\gamma & 0 & -\mathrm{c}\gamma\\ 0 & 0 & 0\\ \mathrm{c}\gamma & 0 & -\mathrm{s}\gamma\end{bmatrix} \tag{B.45}$$

$$\frac{\partial\mathbf{C}_3\left(\chi\right)}{\partial x} = \frac{\partial\chi}{\partial x}\begin{bmatrix}-\mathrm{s}\chi & \mathrm{c}\chi & 0\\ -\mathrm{c}\chi & -\mathrm{s}\chi & 0\\ 0 & 0 & 0\end{bmatrix} \tag{B.46}$$

$$\frac{\partial\mathbf{C}_2\left(-\frac{\pi}{2}-\delta\right)}{\partial x} = \frac{\partial\delta}{\partial x}\begin{bmatrix}-\mathrm{c}\delta & 0 & -\mathrm{s}\delta\\ 0 & 0 & 0\\ \mathrm{s}\delta & 0 & -\mathrm{c}\delta\end{bmatrix} \tag{B.47}$$

$$\frac{\partial\mathbf{C}_3\left(\tilde{\tau}\right)}{\partial x} = \frac{\partial\tilde{\tau}}{\partial x}\begin{bmatrix}-\mathrm{s}\tilde{\tau} & \mathrm{c}\tilde{\tau} & 0\\ -\mathrm{c}\tilde{\tau} & -\mathrm{s}\tilde{\tau} & 0\\ 0 & 0 & 0\end{bmatrix} \tag{B.48}$$

Previously, it was seen that the partial derivatives of the transformation from body frame to inertial frame were dependent on the attitude of the vehicle. The four matrices that are described above depend on either position and velocity or on position only. The flight-path angle is differentiated to:

$$\frac{\partial\gamma}{\partial x} = \omega_{cb}\mathrm{s}\tilde{\tau}\mathrm{s}\delta\frac{V_N V_D}{V^2\sqrt{V^2-V_D^2}} - \omega_{cb}\mathrm{c}\tilde{\tau}\frac{V_E V_D}{V^2\sqrt{V^2-V_D^2}} + \omega_{cb}\mathrm{s}\tilde{\tau}\mathrm{c}\delta\frac{V_D^2-V^2}{V^2\sqrt{V^2-V_D^2}} \tag{B.49}$$

$$\frac{\partial \gamma}{\partial y} = -\omega_{cb} \mathrm{c}\tilde{\tau} \mathrm{s}\delta \frac{V_N V_D}{V^2 \sqrt{V^2 - V_D^2}} - \omega_{cb} \mathrm{s}\tilde{\tau} \frac{V_E V_D}{V^2 \sqrt{V^2 - V_D^2}} - \omega_{cb} \mathrm{c}\tilde{\tau} \mathrm{c}\delta \frac{V_D^2 - V^2}{V^2 \sqrt{V^2 - V_D^2}} \tag{B.50}$$

$$\frac{\partial \gamma}{\partial z} = 0 \tag{B.51}$$

$$\frac{\partial \gamma}{\partial \dot{x}} = -\mathrm{c}\tilde{\tau} \mathrm{s}\delta \frac{V_N V_D}{V^2 \sqrt{V^2 - V_D^2}} - \mathrm{s}\tilde{\tau} \frac{V_E V_D}{V^2 \sqrt{V^2 - V_D^2}} - \mathrm{c}\tilde{\tau} \mathrm{c}\delta \frac{V_D^2 - V^2}{V^2 \sqrt{V^2 - V_D^2}} \tag{B.52}$$

$$\frac{\partial \gamma}{\partial \dot{y}} = -\mathrm{s}\tilde{\tau} \mathrm{s}\delta \frac{V_N V_D}{V^2 \sqrt{V^2 - V_D^2}} + \mathrm{c}\tilde{\tau} \frac{V_E V_D}{V^2 \sqrt{V^2 - V_D^2}} - \mathrm{s}\tilde{\tau} \mathrm{c}\delta \frac{V_D^2 - V^2}{V^2 \sqrt{V^2 - V_D^2}} \tag{B.53}$$

$$\frac{\partial \gamma}{\partial \dot{z}} = \mathrm{c}\delta \frac{V_N V_D}{V^2 \sqrt{V^2 - V_D^2}} - \mathrm{s}\delta \frac{V_D^2 - V^2}{V^2 \sqrt{V^2 - V_D^2}} \tag{B.54}$$

Then, for the heading angle, it is found that:

$$\frac{\partial \chi}{\partial x} = -\omega_{cb} \mathrm{s}\tilde{\tau} \mathrm{s}\delta \frac{V_E}{V_N^2 + V_E^2} - \omega_{cb} \mathrm{c}\tilde{\tau} \frac{V_N}{V_N^2 + V_E^2} \tag{B.55}$$

$$\frac{\partial \chi}{\partial y} = \omega_{cb} \mathrm{c}\tilde{\tau} \mathrm{s}\delta \frac{V_E}{V_N^2 + V_E^2} - \omega_{cb} \mathrm{s}\tilde{\tau} \frac{V_N}{V_N^2 + V_E^2} \tag{B.56}$$

$$\frac{\partial \chi}{\partial z} = 0 \tag{B.57}$$

$$\frac{\partial \chi}{\partial \dot{x}} = \mathrm{c}\tilde{\tau} sx\delta \frac{V_E}{V_N^2 + V_E^2} - \mathrm{s}\tilde{\tau} \frac{V_N}{V_N^2 + V_E^2} \tag{B.58}$$

$$\frac{\partial \chi}{\partial \dot{y}} = \mathrm{s}\tilde{\tau} \mathrm{s}\delta \frac{V_E}{V_N^2 + V_E^2} + \mathrm{c}\tilde{\tau} \frac{V_N}{V_N^2 + V_E^2} \tag{B.59}$$

$$\frac{\partial \chi}{\partial \dot{z}} = -\mathrm{c}\delta \frac{V_E}{V_N^2 + V_E^2} \tag{B.60}$$

The longitude and latitude are a function of position only:

$$\frac{\partial \tilde{\tau}}{\partial x} = -\frac{y}{x^2 + y^2} \tag{B.61}$$

$$\frac{\partial \tilde{\tau}}{\partial y} = \frac{x}{x^2 + y^2} \tag{B.62}$$

$$\frac{\partial \tilde{\tau}}{\partial z} = 0 \tag{B.63}$$

$$\frac{\partial \delta}{\partial x} = -\frac{xz}{r^2 \sqrt{r^2 - z^2}} \tag{B.64}$$

$$\frac{\partial \delta}{\partial y} = -\frac{yz}{r^2 \sqrt{r^2 - z^2}} \tag{B.65}$$

$$\frac{\partial \delta}{\partial z} = \frac{zr - z^2}{r\,(r - R_E) \sqrt{r^2 - z^2}} \tag{B.66}$$

In these equations, it is found that:

$$r = \sqrt{x^2 + y^2 + z^2} \tag{B.67}$$

By substituting the aforementioned equations all the way back to Equation (B.40), the partial derivatives of the incidence angle are computed. Note that this is done for every pressure port, and thus, when using 8 pressure ports, 8 partial derivatives are computed. Finally, the partial derivatives of the free stream air pressure, dynamic impact pressure and incidence angle can be substituted into Equation (B.24) to compute the pressure derivatives that are used in the **H** matrix.