

PROJECT OF GROUP 20GR935

\mathcal{H}_∞ - and LPV Control for a modified VEGA Launch Vehicle in Simulation

NINTH SEMESTER OF CONTROL AND AUTOMATION



AALBORG UNIVERSITY

2020



Title:

\mathcal{H}_∞ - and LPV Control for a modified VEGA Launch Vehicle in Simulation

Project:

P9-project

Period of Project:

September 2nd 2020 - December 21th 2020

Project Group:

20gr935

Members:

Frederik Rasmussen
Jakob Stentoft Poulsen
Mikkel Damgård Hardysøe

Supervisor:

Rafal Wisniewski

External Supervisor:

Finn Ankersen (European Space Agency)

ECTS points: 20

Printed exemplary: 0

Page number: 96

Ended: July 3, 2021

Abstract

The main objective of this report has been to design a robust controller for a Launch Vehicle (LV) from the lift-off phase to the Main Engine Cut-Off (MECO) phase. The controller objective is to satisfy the specified requirements while accounting for varying parameters impacting the LV system. An analysis investigating the most important dynamics on a LV under a launch phase has been carried out. This includes disturbances impacting the LV and a discussion of the best suitable controller to use. Thereafter, a linear state space has been derived using a small signal model. An LPV controller was decided to be ideal to control the highly varying parameters in during the atmospheric phase. However, because of the high similarities was an augmented structured \mathcal{H}_∞ synthesis with weights on the input and output of the closed loop system designed and implemented first. The controller from the \mathcal{H}_∞ synthesis was first tested for stability. Next a nonlinear simulation has been implemented in Simulink for verification of the controller. The results of the test confirm that the controller and simulation work as intended. Future work will have to be carried out to verify the controller through a more thorough analysis and develop a more robust controller through LPV synthesis.

Preface

This report is written by students at Aalborg University, who study the 3nd semester of the Control and Automation Master's degree. The reader of this report is therefore assumed to have some degree of knowledge about the subject of control and automation.

The report contains 9 chapters, beginning with an introduction and ending with a discussion and further developments chapter. The appendix can be found at the end of the report. The main topic of the report is the analysis and development of a \mathcal{H}_{∞} - and LPV Control for a modified VEGA Launch Vehicle tested in a custom simulation also developed in this rapport. All literature and sources used for the report can be found in the bibliography.

The group would like to thanks the European Space Agency (ESA) and namely Finn Ankersen for guidance throughout this project and providing a trajectory together with the aerodynamics of a launch vehicle model.

The group would also like to thank the supervisor Rafal Wisniewski for his help and guidance throughout this project.

Table of Contents

1	Introduction	3
1.1	Introduction	3
1.2	Motivation	3
1.3	Case Description	5
2	Problem Analysis	7
2.1	Vega Launcher	7
2.2	Dynamics of a Launch Vehicle	9
2.3	Variables that is Changing with Time	18
2.4	Disturbances	20
2.5	Trajectory	24
2.6	Controller Choice	25
2.7	Simulation	26
2.8	Conclusion	27
3	Requirement Specification	29
3.1	Requirements Specification	29
4	Modelling and Linearization	33
4.1	Linearized Equation of Motion	34
4.2	Controllability Analysis	40
4.3	TVC Actuator Model	43
4.4	Delay Model	44
4.5	Conclusion	44
5	Control System and Controller Models	47
5.1	LFT Framework	47
5.2	\mathcal{H}_∞ Synthesis	48
5.3	Weighting Function Selection	53
5.4	LPV Modelling	57
5.5	LPV synthesis	57
5.6	Analysis of Varying Parameters	59
5.7	Conclusion	63
6	Implementation of the Simulation	65
6.1	Implementation of Subgroups Inside the Simulation	66
6.2	Variation from the Simulation to Real Tests	71
6.3	Additional Dynamics	72
7	Results and Analysis	73
7.1	Simulations	73

7.2 Results	75
7.3 Result Analysis	79
8 Conclusion	83
9 Discussion and Further Development	85
9.1 Discussion	85
9.2 Further Development	86
Bibliography	87
A Parameters used in MATLAB	91
A.1 Launch vehicle parameters	91

Introduction 1

1.1 Introduction

In these modern times it is hard to imagine living without the advantages that have come from space exploration. From weather forecasts, Global Positioning System (GPS) and almost instantaneously communication around the globe to interplanetary missions to Mars, the Moon and beyond. Space travel will not only help the human race advance to become an interplanetary species but has also connected people across country borders in doing so.

It is clear that space exploration will continue to contribute even more in the coming years. One example of the rather new growing interest in space is the exploitation of asteroids. For the climate and the economy sector there are two main benefits. The first one is to move heavy industry activities into space to spare the environment on Earth. The other benefit is to mine the asteroids for minerals that are otherwise rare on Earth such as platinum group metals. Another resource that will become very important for space exploration in the coming future is water which is valuable for upcoming moon bases not just for hydration but for an efficient radiation shield, and for the production of fuel in space to potentially refuel once out of the earth atmosphere an in orbit. Already these concepts are being pursued by companies such as Planetary Resources and Deep Space Industries [1].

This is just some examples of the many possibilities further space exploration will open for us. Unfortunately, even sending satellites to space in a safe and affordable manner is a difficult task. In the following section, some of the challenges of why space exploration is a difficult task will be elaborated upon.

1.2 Motivation

This project is motivated by making space more accessible for everybody. There are mainly two challenges for making that happen. The first challenge is the success rate of launching rocket into orbit. Compared to the 1 in a 1.000.000 odds for a fatal accident happening on commercial airplane [2], the odds for a critical error for a rocket to classify it is a failure is approximately 1 in 10. However rapidly improving.

These errors include an explosion in the fuel tank to fail to achieve orbit. It should be noted that this statistic is influenced on early launches in the 60's and 70's, where launches had a much more noticeable failure rate. In Table 1.1 the success rate for each decade is listed since the first successful launch of an artificial satellite, namely Sputnik 1.

Period (year)	Success rate (%)
1957-1959	48.70
1960-1969	78.06
1970-1979	93.45
1980-1989	95.31
1990-1999	92.71
2000-2009	94.09
2010-2019	94.82
Total	89.37

Table 1.1: Rocket launch success rate per decade [3].

The period between 1980-1989 has not only the highest success rate of launches but also the most with 1193 launches compared to the 886 launches between 2010-2019. It seems that the success rate is becoming stagnant, this can in part be because of the very conservative space industry still relying strongly on classical control [4].

The first step to get to space is to get from the surface of the Earth to orbit around the Earth. This step is also the hardest part in space travel, since the gravitational acceleration of the Earth is high compared to nearby celestial bodies around the Earth. This is both because the rocket has fought the atmosphere and the strong gravity while trying to get enough vertical speed to stay in orbit.

This is being compensated for by burning more propellant in the lift-off stage. However, this contributes to an impractical loop, because as more fuel is added to the rocket, the gross mass increases and hence more propellant is needed to compensate for that extra propellant mass. This relation is described by the Tsiolkovsky rocket equation explained in subsection 2.3.2. This results in the mass fraction of propellant being substantially high but will change depending on what type of propellant is used. The mass fraction is the ratio of propellant to structural mass and payload that can be put into orbit. In Table 1.2 different propellant types used in rockets can be seen together with their respective mass fraction in a rocket.

Propellant type	Mass propellant percentage for Earth orbit (%)
Solid	96
Kerosene-Oxygen	94
Hypergols	93
Methane-Oxygen	90
Hydrogen-Oxygen	83

Table 1.2: A table of propellant types and their respective ideal mass fraction given in percent [5].

The table shows that hydrogen/oxygen is the best performance-wise propellant, however it is technically more complex. Kerosene offers less performance but gives more simple, robust, and easier to fabricate rocket. These mass fractions have not taken into account of atmospheric drag or incomplete combustion, which means that in practice, the percentages will come closer to the breaking point of 100 %. In fact, if the radius of the Earth was

1.5 times larger, it would not be possible to reach the required velocity to reach an orbit at all with our current types of propellant. However, clever ways of engineering can help compensate for the high mass fraction, such as rocket staging, which reduces the gross mass by removing the dead weight of empty fuel tanks [5].

The second challenge is that spaceflight is too expensive for the time being to have regular spaceflights. In Table 1.3 either relative new rocket models and models yet to come are listed where the oldest model is from the manufacturer Arianespace with the Ariane 5 model from 1996. In the table the maximum cargo capacity, how much it costs to launch and how much it costs per kg are listed respectively. The models are listed from worst to best in terms of cost/kg.

Rocket model	Max. cargo capacity	Cost/launch	Cost/kg
Minotaur VI (TBD)	2.6 ton	\$60 M	\$23,100
Vega (2012)	2.0 ton	\$37 M	\$18,500
Koalizhou 11 (2020) [6]	1.5 ton	\$15 M	\$10,000
Ariane 5 (1996) [7]	20.0 ton	\$178 M	\$8,900
H3 (2021) [8]	10.0 ton	\$50 M	\$5,000
Ariane 6 (2021) [7]	20.0 ton	\$94 M	\$4,700
Soyuz 5 (TBD) [9]	18.0 ton	\$50 M	\$2,800
Falcon 9 (2010)	22.8 ton	\$62 M/\$49 M (reused)	\$2,720/\$2,150 (reused)
Falcon Heavy (2018)	63.8 ton	\$90 M	\$1,400

Table 1.3: Comparison between rocket models ordered from worst to best in terms of cost/kg [10].

From the table the cheapest models so far are the Falcon Heavy model and the Falcon 9 model provided by SpaceX. Even though the rockets are getting cheaper for each model [10], a method to substantially reducing the cost is to reuse the 1st stage just like the Falcon 9- and Falcon Heavy model. By recycling a stage, the cost of building a new stage can be almost removed, and hence a cheaper access to space. However, by making a stage land again adds an extra layer of complexity to an already difficult launch procedure. With that in mind, the following section will conclude the approach for the rest of project.

1.3 Case Description

As mentioned, this project is motivated by making space more accessible. This is done by making space travel safer and less expensive. Seen in Table 1.3, by reusing the 1st stage is making the cost/kg-ratio 570 lower as for the Falcon 9 model. The final objective will be to able to land the 1st stage of a rocket. However, before making the stage land again, the Launch Vehicle (LV) must first be able to ascent. Thus before trying to land a rocket which is the motivation for this project, it has been chosen to first focus on the first part of the flight, from lift-off to the main engine cut-off (MECO) stages of the ascent in this project. This is deemed a good objective to get familiar with the task of controlling a rocket.

As for making space travel safer, a controller will be made to control the LV along a predetermined trajectory from the landing pad to MECO. The controller will be designed to be robust through the ascent. The trajectory will be determined by a typical ascent

profile of a rocket. The controller should thus take into account the things that are the most relevant for a rocket, like the great aerodynamic loads changing all along the trajectory and the big changes in the properties of the rocket as it loses most of its mass for propulsion.

As for this project the Vega rocket will be used as a benchmark. The system dimensions, dynamics and objective will be used as a reference to design a controller. However, the Vega rocket has only one main engine thruster [11]. Since the idea of landing the first stage again requires control of the thrust, it would have to be fitted with new engines this would most likely be liquid fuel engines. And to explore the possibilities of other engine configurations this project will investigate the case of having multiple engines instead of one main engine.

1.3.1 Initial Problem Formulation

This leads to an initial problem formulation, which is stated below to guide the problem analysis in the following chapter, where in the end of Chapter 2, a specific problem formulation can be stated which will describe the final objective of this report:

How can a robust controller be designed to make a launch vehicle follow a predetermined trajectory from lift-off to MECO?

Problem Analysis 2

This chapter is the investigation of the necessary subjects to try and answer the initial problem formulation in section 1.3.1. It will lay out for the reader, the most important subjects explored for this project.

2.1 Vega Launcher

Vega (Vettore Europeo di Generazione Avanzata) is developed under the responsibility of ESA and European Launch Vehicle (ELV)/AVIO as prime contractor. At the time of writing the Vega launcher have had 17 launches where two lift-offs has failed. The Vega launcher is designed to launch small payloads from 300 kg to 2500 kg into either low earth orbit (LEO) or sun-synchronous orbits (SSO). Vega is currently the smallest European launcher with approximately 30m in height, 3m in diameter and weights 137 ton when loaded with fuel [11]. Moreover, the Vega launcher has an asymmetric shape along the longitudinal axis [12], which will be relevant later in subsection 2.2.4, when evaluating the TVC system of the LV.

An overview of the Vega launcher can be seen in Fig. 2.1. Vega is a single body launcher that uses four stages during its ascent. The first stage is the P80, which consists of a Thrust Vector Control (TVC), which has two electromechanical actuators that are gimbaling a movable nozzle [13]. This TVC system stabilizes the rocket during ascent based on measurements from the inertial navigation system (INS). However, with only one engine the Vega LV rely on gas thrusters to stabilize the roll of the rocket. The parameters of the P80 stage are going to be used for the models to design a controller for the TVC, to provide attitude control in the pitch and yaw axis. The parameters of all the stages can be seen in Table 2.1.

Stages	Stage 1: P80	Stage 2: Zefiro 23	Stage 3: Zefiro 9	Stage 4: AVUM
Height [m]	11.7	7.5	3.5	1.7
Diameter [m]	3	1.9	1.9	1.9
Propellant type	Solid	Solid	Solid	Fluid
Propellant mass [t]	88	24	10.5	0.55
Motor dry mass [kg]	7330	1950	915	131
Average thrust [kN]	2200	871	260	2.42
Burn time [s]	110	77	120	667
Specific impulse [s]	280	287.5	296	315.5

Table 2.1: Parameters of the different stages on the Vega launcher [14]

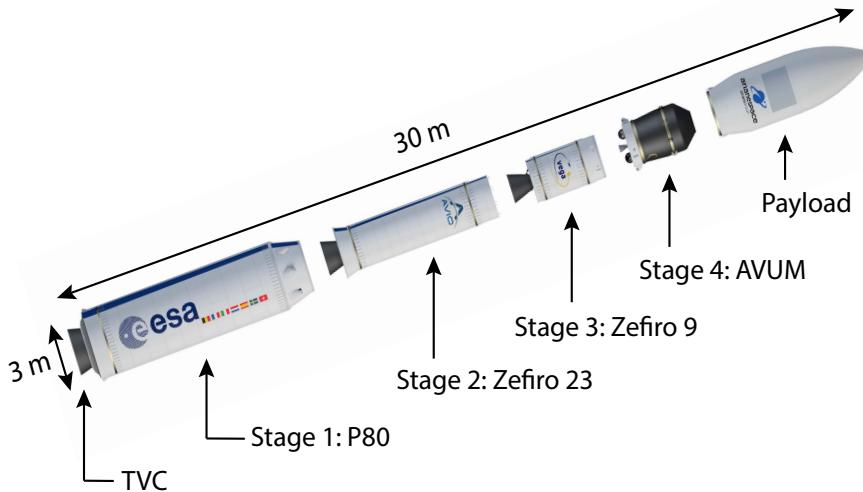


Figure 2.1: Vega launcher overview [15]

2.1.1 Deviations from the Original Vega Launcher

The Vega launcher will be the primary reference for this project. However, some things have been chosen to be different from the original Vega launcher.

Number of Thrusters

The Vega launcher as mentioned before only has one main engine to ascend with [11]. However, in this project it has been chosen to simulate multiple thrusters on the LV. This is intended to create a new challenge to explore the problem of using more than one main engine, which has already explored greatly [4][16].

The multiple thrusters will have the same total thrust. Meaning that the original thrust of the Vega launcher will remain the same for this project. This is done by simply dividing the total thrust with the number of new thrusters chosen. The chosen number of thrusters for the project is four.

The increase in rocket thrusters creates a new challenge. Before, only one thruster could gimbal, however, now there are multiple thrusters which will lead to different ways to control the LV. The dynamics of the Vega launcher are changed by the new number of thrusters and the location of them. However, the dynamics will first be explored for a single thruster and then later the additional thrusters and their locations will be considered. In subsection 2.2.4, the dynamics of the multiple thrusters are elaborated.

Solid Versus Liquid Fuel

The original Vega launcher is equipped with solid fuel. This will not be the case for this project, because of the need to control the engine thrust on the descend. However, in the first phase of the launch it is not required to be able to control the thrust of the engines and therefore this will not be done. The liquid fuel does create a disturbance, which is important to investigate. This can be found in the subsection 2.4.2.

2.2 Dynamics of a Launch Vehicle

This section will cover the most relevant dynamics for controlling a LV. Firstly, the reference frames will be defined for referring to the attitude and position and how to convert between relevant frames. Then, the equations of forces and rotation on a LV will be evaluated, followed by an analysis of the dynamics of the TVC control system. Afterwards the obstacles to overcome such as disturbance and the great change in parameters will be described. And finally, the relevant controllers will be outlined.

2.2.1 Reference Frames

When describing movement in space the reference frame is very important. When on the earth it is convenient to describe the position as fixed relative to the ground, however, when describing motion in space it is far more convenient to describe position on an inertial reference frame. An example is for a satellite in orbit it is more practical to have the Earth's center, or whichever mass it is orbiting, as the center, but align axis with the rest of the universe instead of the Earth. Additionally, it can from a control perspective be practical to use the object that is being controlled as the frame of reference. This project will use the same reference frames used in [16] and will be explained here.

Earth-Centered Reference Frames

The Earth-centered inertial (ECI) frame has the Earth center as the origin and constant in relation to the stars. This is usually used when describing objects in orbit or on their way into orbit. Its basis vectors are $\{i_I, j_I, k_I\}$, where k_I is pointing to the North pole, i_I is pointing to the Sun at vernal equinox in spring and j_I completes the right-handed set. Vernal equinox is when the Sun is directly above the equator of the Earth.

The Earth-Centered Earth-Fixed (ECEF) frame has the center of the Earth as origin just like the ECI. With the basis vectors $\{i_E, j_E, k_E\}$. Compared to ECI it is rotating with the Earth at the velocity of ω_E , while k_E is the same, with i_E is following the Greenwich meridian. Hence the frame rotates with the Earth's angular velocity $\Omega_I = \omega_E k_I$. It is useful for positions on Earth or in the Earth's atmosphere, as it is fixed relative to the ground and the air. Which also means that there is a straightforward conversion from longitude and latitude to ECEF. The rotation of the LV or any free body in space is usually described by quaternions as they don't experience problems such as gimbal lock. And it is described by four elements. In this project, to follow the standards usually used for spaceflight and in the material acquired from ESA, the scalar element will be the first entry of the quaternions.

$$q = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{pmatrix} \quad (2.1)$$

The rotation quaternion $q_E^I(t)$ describes the rotation from ECI to ECEF. And to perform a rotation on a vector the associated Direction Cosine Matrixes (DCM) are used, the DCM,

C_q , and quaternion q has the following relation.

$$C_q = \begin{bmatrix} (q_0^2 + q_1^2 - q_2^2 - q_3^2) & 2(q_1 q_2 + q_0 q_3) & 2(q_1 q_3 - q_0 q_2) \\ 2(q_1 q_2 - q_0 q_3) & (q_0^2 - q_1^2 + q_2^2 - q_3^2) & 2(q_2 q_3 + q_0 q_1) \\ 2(q_1 q_3 + q_0 q_2) & 2(q_2 q_3 - q_0 q_1) & (q_0^2 - q_1^2 - q_2^2 + q_3^2) \end{bmatrix} \quad (2.2)$$

$C_{q_E^I}(t)$, are related to the attitude in ECEF frame C_{q_E} by Eq. (2.3).

$$C_{q_E^I}(t) = \begin{bmatrix} \cos \omega_E t & \sin \omega_E t & 0 \\ -\sin \omega_E t & \cos \omega_E t & 0 \\ 0 & 0 & 1 \end{bmatrix} C_{q_E(t)} \quad (2.3)$$

Launchpad and Body-Fixed Reference Frame

The launchpad reference frame has basis vectors $\{i_L, j_L, k_L\}$, where vector k_L is perpendicular to the ground and i_L in the direction of the launch, while j_L completes the right-hand set. The origin is at the initial position of the LV which is fixed at $r_I(0)$ in the ECI frame. The initial velocity of the LV standing still on the launchpad will then be $v_I(0) = \Omega_I \times r_I(0)$. The transformation between ECEF and the launchpad frame is time invariant, and to determine the position and velocity at the LP frame, the contribution of the Earth's rotation must be accounted for. The position and velocity can be seen in Eq. (2.4) and Eq. (2.5) respectively.

$$r_L(t) = C_{q_L^E} C_{q_E^I}(t) [r_I(t) - r_I(0)] \quad (2.4)$$

$$v_L(t) = C_{q_L^E} C_{q_E^I}(t) [v_I(t) - \Omega_I \times r_I(t)] \quad (2.5)$$

The Body Reference Frame (BRF) frame is the coordinate system from the LV's perspective. The origin is at the Center of Gravity (CG) and has the basis vectors $\{i_B, j_B, k_B\}$. Vector i_B along the longitudinal axis of the rocket and j_B perpendicular to the pitch plane. The orientation of the LV is given by yaw, ψ , pitch, θ and roll, ϕ . Therefore, upon launch the initial orientation of the LV is $\{\psi(0), \theta(0), \phi(0)\} = \{0, \frac{\pi}{2}, 0\}$ radians. The initial orientation of the LV's body is then given in Eq. (2.6). Furthermore, the initial angular velocity of the LV is then $\omega_B(0) = C_{q_B^L}(0) \Omega_I$.

$$C_{q_B^L}(0) = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} C_{q_L^E} C_{q_E^I}(0) \quad (2.6)$$

Later when explaining the aerodynamic characteristics in subsection 2.2.3, the Velocity Reference Frame (VRF) is used. VRF has the basis vectors $\{i_V, j_V, k_V\}$. It also has origin in CG as with BRF but its vector i_V is pointing along the air-relative velocity $v_{air}(t)$ vector. The difference between the BRF and VRF can be described by a rotation matrix using the angle of attack, α and the sideslip β [16]. The angle of attack and sideslip angle will be evaluated further upon in subsection 2.2.3 [16].

$$C_{q_V^B(t)} = \begin{bmatrix} \cos \alpha(t) \cos \beta(t) & \sin \beta(t) & \sin \alpha(t) \cos \beta(t) \\ -\cos \alpha(t) \sin \beta(t) & \cos \beta(t) & -\sin \alpha(t) \sin \beta(t) \\ -\sin \alpha(t) & 0 & \cos \alpha(t) \end{bmatrix} \quad (2.7)$$

An overview of the different frames is provided in Fig. 2.2.

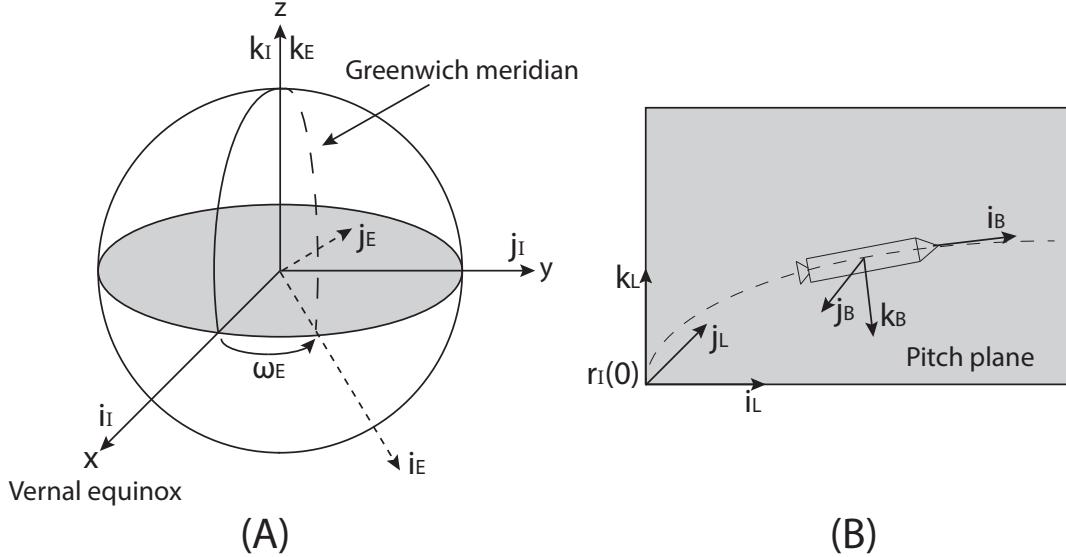


Figure 2.2: Illustrations of different reference frames. (A) shows the ECI- and ECEF frames and (B) shows the LP- and BRF frames.

2.2.2 Equations of Motion

The following dynamic equations describe a position and attitude in space. They have the initial conditions $\{r_I(0), v_I(0), q_B^I(0), \omega_B(0)\}$ that are identified in subsection 2.2.1. Furthermore, the following dynamic equations have not taken moving masses such as tail-wags-dog effect into account, this is however explained in subsection 2.4.2.

The LV's translational motion is described in the ECI frame in Eq. (2.8) [17].

$$\ddot{r}_I(t) = \dot{v}_I(t) = g_I(t) + \frac{1}{m(t)} [F_{aero}(t) + F_{TVC}(t) + F_{fin}(t) + F_{thr,I}(t)] \quad (2.8)$$

where:

$\ddot{r}_I(t)$	is the acceleration.	$[\frac{m}{s^2}]$
$m(t)$	is the mass if the vehicle.	$[kg]$
$F_{aero}(t)$	is the aerodynamic force acting on the vehicle.	$[N]$
$F_{TVC}(t)$	is the thrust vector control acting on the vehicle.	$[N]$
$F_{fin}(t)$	is the force from the fins acting on the vehicle.	$[N]$
$F_{thr,I}(t)$	is the cold gas thruster force acting on the vehicle.	$[N]$

Additionally, the rotational dynamics are described in the body-fixed frame (BRF) is given in Eq. (2.9) [17].

$$\dot{\omega}_B(t) = J^{-1}(t) \left[M_{aero,B}(t) + M_{TVC,B}(t) + M_{fin,B}(t) + M_{thr,B}(t) - \omega_B(t) \times J(t)\omega_B(t) - J(t)\omega_B(t) \right] \quad (2.9)$$

The equations of forces and moment can be augmented to include other effects, such as structural flexibility from high aerodynamic pressure and propellant sloshing. In this chapter however, only the equations of a rigid-body motion are considered

where:

$\omega(t)$	is the angle acceleration.	$\left[\frac{\text{rad}}{\text{s}^2} \right]$
J	is the moment of inertia	$[\text{kgm}^2]$
$M_{aero}(t)$	is the aerodynamic moment acting on the vehicle.	$[\text{Nm}]$
$M_{TVC}(t)$	is the thrust moment acting on the vehicle.	$[\text{Nm}]$
$F_{fin}(t)$	is the moment from the fins acting on the vehicle.	$[\text{Nm}]$
$F_{thr,I}(t)$	is the cold gas thruster moment acting on the vehicle.	$[\text{Nm}]$

The last two terms are due to the Coriolis effect and the rate of change in mass, respectively.

Lastly, the quaternion-based kinematics equation is given in Eq. (2.10), where the orientation of the LV's body axes is propagated.

$$\dot{q}_B^I(t) = \frac{1}{2} \begin{bmatrix} q_4(t) & -q_3(t) & q_2(t) \\ q_3(t) & q_4(t) & -q_1(t) \\ -q_2(t) & q_1(t) & q_4(t) \\ -q_1(t) & -q_2(t) & -q_3(t) \end{bmatrix} \omega_B(t) \quad (2.10)$$

The quaternion is given by $q_B^I(t) = [q_1(t) \ q_2(t) \ q_3(t) \ q_4(t)]^T$ and is defined to contain the scalar part, $q_1(t)$ as its first component. The associated DCM, $C_{q_B^I}$, are also important for the computation of the forces and moments given in Eq. (2.8) and Eq. (2.9) [16].

2.2.3 Aerodynamic Characteristics

The aerodynamic forces and moments on the LV's main body depend not only on the instantaneous dynamic pressure, direction, and shape of LV.

To understand the dynamic pressure on the LV's body, first assume windless conditions. That way the velocity of the LV can instead be interpreted as velocity of wind flow impacting the LV. Then the one-dimensional version of the conservation of linear momentum for a fluid is given by [18]:

$$-\frac{\Delta Q}{\Delta x} = \rho V \frac{\Delta u}{\Delta x} \quad (2.11)$$

where:

Q	is pressure	$[\text{Pa}]$
x	is the direction of the flow	$[\text{m}]$
V	is velocity of the wind in the x-direction.	$[\frac{\text{m}}{\text{s}}]$
ρ	is density of the fluid	$[\frac{\text{kg}}{\text{m}^3}]$

Rearranging Eq. (2.11):

$$\begin{aligned} \frac{dQ}{dx} + \rho \cdot V \frac{dV}{dx} &= 0 \\ \frac{dQ}{dx} + \frac{d}{dx} 0.5 \cdot \rho V^2 &= 0 \\ \frac{d}{dx} (Q + 0.5 \rho V^2) &= 0 \\ Q_s + \underbrace{0.5 \rho V^2}_{Q_d} &= Q_t \end{aligned} \quad (2.12)$$

where the first term, Q_s , is the static pressure, the second term, Q_d , is dynamic pressure and Q_t is the total pressure.

$$Q_d = \frac{1}{2} \rho V^2 \quad (2.13)$$

The dynamic pressure is a defined property of a moving flow of gas, it is the pressure experienced by an object with relative velocity to a liquid or gas. The dynamic pressure is squarely proportional to the relative wind speed of an object [19]. Note that Eq. (2.13) is assuming no wind, the speed of any wind relative to the object will be added to the speed.

Setting the velocity as the air-relative velocity vector at the LV's CG, $V(t)$ can then be written as: $V(t) = \|v_{air}(t)\|$ and $v_{air}(t) = [v_{air,x}(t) \ v_{air,y}(t) \ v_{air,z}(t)]^T$. This vector written in the body-fixed frame from subsection 2.2.1 and taking wind disturbances and Earth's rotation into account $v_{air}(t)$ can be written as:

$$v_{air}(t) = C_{q_B^I}(t) \left[v_I(t) - \Omega_I \times r_I(t) - C_{q_I^E}(t) w_E(t) \right] \quad (2.14)$$

The local velocity at an arbitrary location along the LV's body can also be represented by $P_j(t)$ relative to the CG position of the LV. The position P_{CG} is represented as $P_{CG}(t) = [x_{CG}(t) \ y_{CG}(t) \ z_{CG}(t)]^T$. Where $y_{CG}(t)$ and $z_{CG}(t)$ is assumed to be zero.

$$v_{air}^j(t) = v_{air}(t) + \omega_B(t) \times [P_j(t) - P_{CG}(t)] \quad (2.15)$$

The place where the aerodynamic forces that are applied to the LV is at the center of pressure (CP), i.e. at $P_j(t) = P_{CP}(t)$. In the CP two aerodynamic angles can then be defined as followed:

$$\alpha^{CP}(t) = \arctan_2 \frac{v_{air,z}^{CP}(t)}{v_{air,x}^{CP}(t)} \quad (2.16)$$

$$\beta^{CP}(t) = \arcsin \frac{v_{air,y}^{CP}(t)}{\|v_{air}^{CP}(t)\|} \quad (2.17)$$

where α^{CP} is the angle of attack (AoA) between the longitudinal axis and local airflow vector at the CP in the pitch plane and β^{CP} is the sideslip angle between the longitudinal axis and local airflow vector at the CP in the yaw plane. In Fig. 2.3 an illustration of how the AoA and sideslip angle are related where the blue and green arrow is the projected longitudinal axis on the pitch and yaw plane and projected airflow vector on the pitch and yaw plane respectively. Note that the airflow vector is assumed to be the same as the velocity vector, V , of the LV just in the opposite direction.

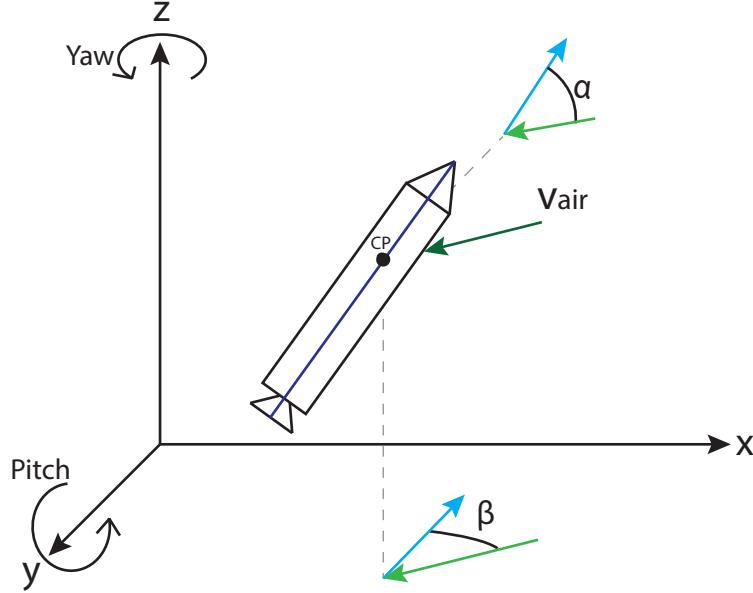


Figure 2.3: Illustration of how the angle of attack and sideslip angle form. The blue and green arrow is the projected longitudinal axis on the pitch and yaw plane and projected airflow vector on the pitch and yaw plane respectively.

As mentioned in Section 2.1 the Vega launcher has an axisymmetric shape around the longitudinal axis. With this knowledge the aerodynamic forces can then be expressed in the air-relative velocity reference frame.

$$F_{aero,V}(t) = -Q(t)S_{ref} \begin{bmatrix} C_D(\alpha_{eff}(t), M(t)) \\ 0 \\ C_L(\alpha_{eff}(t), M(t)) \end{bmatrix} \quad (2.18)$$

where C_D and C_L are dimensionless quantities of the drag- and lift coefficient and are given by Eq. (2.19) and Eq. (2.20) respectively.

$$C_D = \frac{D}{Q_d S} \quad (2.19)$$

$$C_L = \frac{L}{Q_d S} \quad (2.20)$$

where:

Q_d	is the dynamic pressure	[Pa]
S	is the reference area	[m ²]
D	is the drag force.	[N]
L	is the lift force	[N]

Note that the drag force D is parallel and in the opposite direction of the velocity vector, while the lift force L is normal to the velocity vector of the LV w.r.t. the air. Furthermore, both the coefficients are measured and functions of the Mach number and the effective angle of attack [20], which are given by Eq. (2.21) and Eq. (2.22) respectively.

$$M = \frac{V}{v_{\text{sonic}}(t)} \quad (2.21)$$

$$\alpha_{\text{eff}} = \arccos \left(\cos \alpha^{CP}(t) \cos \beta^{CP}(t) \right) \quad (2.22)$$

Where $v_{\text{sonic}}(t)$ is the speed of sound. Eq. (2.18) can be written into ECI frame given in Eq. (2.23).

$$F_{\text{aero},I}(t) = C_{q_I^B}(t) C_{q_V^B}(t) F_{\text{aero},V}(t) \quad (2.23)$$

Additionally, an aerodynamic moment around the vehicle's CG is produced, due to the CP being offset by the CG. In the BRF this momentum is expressed as Eq. (2.24).

$$M_{\text{aero},B} = [P_{CP}(t) - P_{CG}(t)] \times C_{q_V^B}(t) F_{\text{aero},V}(t) \quad (2.24)$$

Similarly with P_{CG} , P_{CP} is the point is represented as $P_{CP}(t) = [x_{CP}(t) \ 0 \ 0]^T$, where it is assumed that the center of pressure is in the center along the x-axis of the LV.

2.2.4 Thrust Vector Control System

For this section the one main engine of the rocket will be evaluated first and next there will be evaluated when the main engine is split into four thrusters.

The rocket engine generates the thrust needed for ascending into the atmosphere. By adjusting the magnitude and direction of the thrust, the LV's trajectory can be controlled. As mentioned in Section 2.1, this adjustment is achieved by two TVC electromechanical actuators that operate the movable nozzle by $\{\beta_{\text{TVC},y}(t), \beta_{\text{TVC},z}(t)\}$ along the BRF axes j_B and k_B respectively. The two actuators are assumed to be attached 90° apart, one to rotate around the y-axis and then the z-axis. By assuming the gimbal ring, the rotation around the thrusters' pitch and yaw axes can be described by a rotation matrix. With the BRF in mind the TVC generated force is given in Eq. (2.25) [16].

$$F_{\text{TVC},B}(t) = F_{\text{TVC},\text{ref}}(t) \begin{bmatrix} \cos \beta_{\text{TVC},y}(t) \cos \beta_{\text{TVC},z}(t) \\ \cos \beta_{\text{TVC},y}(t) \sin \beta_{\text{TVC},z}(t) \\ -\sin \beta_{\text{TVC},y}(t) \end{bmatrix} \quad (2.25)$$

where $T_{\text{TVC},\text{ref}}$ is the required thrust magnitude which is commanded by the guidance system. A visual representation of the TVC vector and the associated angles can be seen

in Fig. 2.4. The coordinate system is in the BRF and the green vector represents the pitch angle while the red vector represents the yaw angle. It is assumed that the thruster is located at the LV's bottom center.

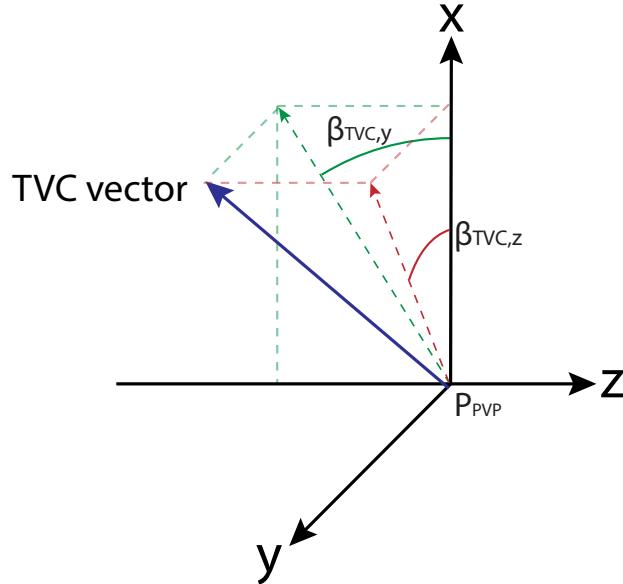


Figure 2.4: Representation of a TVC vector and how the angles are related to Eq. (2.25). The green vector represents the pitch angle while the red vector represents the yaw angle.

The TVC generated force in the ECI frame can be computed by multiplying it with the associated DCM and is given in Eq. (2.26).

$$F_{TVC,I}(t) = C_{q_I^B}(t) F_{TVC,B}(t) \quad (2.26)$$

The moment around the CG of the LV is given by:

$$M_{TVC,B}(t) = [P_{PVP} - P_{CG}(t)] \times F_{TVC,B}(t) \quad (2.27)$$

where $P_{PVP} = [P_{PVP} \ 0 \ 0]^T$ is representing the TVC pivot position.

Multiple Thruster Dynamics

Back in subsection 2.1.1, it was mentioned that the LV model in this report have four thrusters instead of one, and that they each have 1/4 of the total thruster force from the original Vega launcher.

The four thrusters are placed in a square formation, such that they are located in a symmetrical manner around the y- and z-axes. In Fig. 2.5A the formation is visualized from the bottom of the LV.

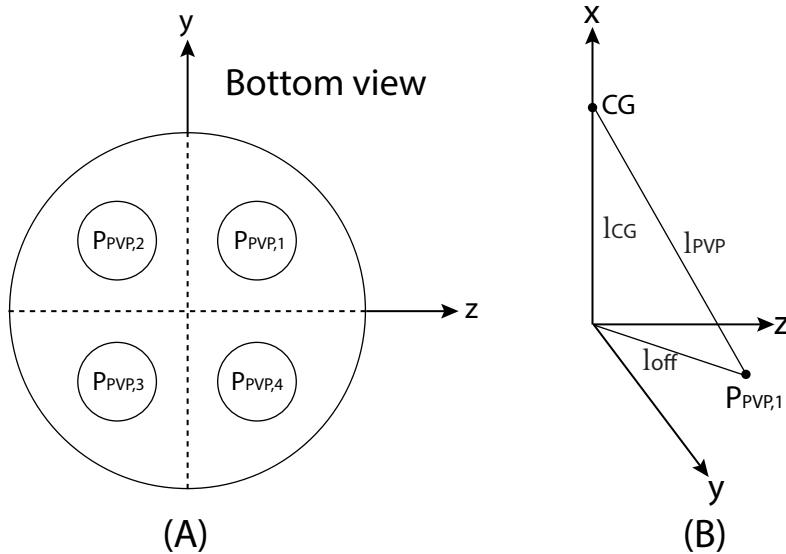


Figure 2.5: (A) is showing the formation of thrusters from the bottom of the LV. (B) is showing the geometric relations of the pivot point of the thrusters, when they are offset from the origin.

The offset the four thrusters have from the origin in the yz-plane, is changing the distance from the center of gravity to the pivot point, P_{PVP} . The new distance is visualized in Fig. 2.5B, and is given by:

$$l_{PVP} = \sqrt{l_{CG}^2 + l_{off}^2} \quad (2.28)$$

where:

l_{PVP}	is the distance from the CG to the pivot point of the thruster.	[m]
l_{CG}	is the distance along the x-axis from the CG to the bottom of the LV.	[m]
l_{off}	is the distance from the bottom of the LV to the pivot point.	[m]

2.2.5 Fins and Cold Gas Thrusters

Planar fins are used on the LV to ensure sufficient attitude control under low or none TVC effectiveness. Traditionally, two pair of fins are assigned to pitch- and yaw motion control, where $\{\beta_{fin,1}(t), \beta_{fin,2}(t)\}$ is assigned to the pitch control and $\{\beta_{fin,3}(t), \beta_{fin,4}(t)\}$ are assigned to yaw control.

Cold gas thrusters are also included for controllability under low or zero TVC effectiveness but for low fin effectiveness as well. Much like the fins, two pairs of thrusters are assigned to pitch and yaw control.

However, as mentioned in Section 1.3 the project only considers the trajectory from liftoff to MECO. During the ascent between the stages, the effectiveness from the TVC system is relatively high, resulting the planar fins and cold gas thrusters will not be needed to be controlled. That is why in this report the fins and cold gas thrusters are not considered when modelling the LV. However, for future work when landing the first stage again, the

cold gas thrusters and fins are needed for attitude control to orientate towards the landing pad during the descent stage. During that stage, the effectiveness of the TVC system will be substantially lower, so that the fins and cold gas thrusters are no longer redundant to the control system of the LV.

2.3 Variables that is Changing with Time

When dealing with LV ascents, some parameters that is either impacting with- or within the LV can change at given times during the ascent. The relevant parameters that for an ascent is listed below.

2.3.1 Changes in Gravitational Acceleration

The gravitational acceleration of the Earth changes at different altitudes. The gravity depending on the altitude is given by Eq. (2.29) [21].

$$g_h = g_0 \cdot \left(\frac{r_0}{r_0 + h} \right)^2 \quad (2.29)$$

where:

g_0	is the gravitational acceleration at Earth's surface.	$\left[\frac{\text{m}}{\text{s}^2} \right]$
r_0	is the radius of the Earth.	$[\text{m}]$
h	is the altitude above the surface of the Earth.	$[\text{m}]$
g_h	is the gravitational acceleration at a given altitude.	$\left[\frac{\text{m}}{\text{s}^2} \right]$

2.3.2 Changes in Vehicle Mass

As for the mass of the LV, the fuel mass is reduced when the engines are producing thrust. As for calculating the rate of change in mass, the rocket equation describes this phenomenon. Different forms to calculate the rate of change in mass exists, one is given in Eq. (2.30) [22].

$$\dot{m} = -\frac{F_{\text{TVC}}}{I_{sp} \cdot g_h} \quad (2.30)$$

where:

\dot{m}	is the mass flow rate.	$[\text{kg}]$
F_{TVC}	is the thrust force.	$[\text{N}]$
I_{sp}	is the specific impulse.	$[\text{s}]$
g_h	is the gravitational acceleration at an given altitude.	$\left[\frac{\text{m}}{\text{s}^2} \right]$

The reason for I_{sp} is measured in seconds is to compare engines with each other in a meaningful way. It is the number of seconds a unit weight of fuel will last if the engine will imply a unit force. That is if an engine can be scaled proportionately [22].

2.3.3 Change of CG position

The change in CG can be done using a simple function. First is the CG found for the entire rocket except of the first stage fuel. This is the weight of all the stages, payloads, and the maximum fuel on the LV. When this initial CG is found can a function be created using the percentage fuel left together with its mass to find the dynamic CG.

$$CG_{\text{total}} = \frac{m_r \cdot CG_r + m_f \cdot CG_f(m_f)}{m_r + m_f} \quad (2.31)$$

where:

CG_{total}	the mass of the rocket.	[kg]
m_r	the mass of the rocket.	[kg]
m_f	the total mass of the fuel left.	[kg]
CG_r	the distance from the engines to the CG for the launch vehicle.	[m]
$CG_f(m_f)$	outputs the distance from the engines to the CG for the fuel.	[m]

The function $CG_f(m_f)$ uses the fuel left as an input to calculate the current CG for the remaining fuel. This can be seen in equation 2.32:

$$CG_f(m_f) = \frac{l_f \cdot \frac{m_f}{m_{f,\max}}}{2} \quad (2.32)$$

where:

$CG_f(m_f)$	outputs the distance from the engines to the CG for the fuel.	[m]
l_f	the length of the fuel tank.	[m]
m_f	the mass of the current fuel.	[kg]
$m_{f,\max}$	the maximum mass of the fuel.	[kg]

2.3.4 Change of Inertia

The inertia is calculated by first finding the inertia of the fuel since this is the varying part. Then the inertia of the fuel can simply be summed together with the varying inertia of the fuel.

The fuel inertia in the x rotation is done by the following function:

$$J_{f,x} = \frac{1}{2} \cdot m_{\text{fuel}} \cdot r_f^2 \quad (2.33)$$

where:

$J_{f,x}$	is the fuel inertia in the x rotation.	[kg m ²]
m_{fuel}	is the mass of the current fuel.	[kg]
r_f	is the radius of the fuel tank.	[m]

The inertia of the fuel in the y and z plane is the same because of symmetry and can both be calculated together. This is done by the following method:

$$J_{f,yz} = \frac{1}{4} \cdot m_{fuel} \cdot r_f^2 + \frac{1}{12} \cdot m_{fuel} \cdot l_{fuel}^2 + m_{fuel} \cdot cg_f(f)^2 \quad (2.34)$$

where:

$J_{f,yz}$	is the fuel inertia in the y and z rotation.	$[\text{kg m}^2]$
m_{fuel}	is the mass of the current fuel.	$[\text{kg}]$
r_f	is the radius of the fuel tank.	$[\text{m}]$
l_{fuel}	is the length of the current fuel.	$[\text{m}]$
$cg_f(m_f)$	is the distance from the engines to the CG for the fuel.	$[\text{m}]$

Now that the inertia of the fuel has been calculated, and the inertia of the rest of the LV is given, the two inertia can be summed together. The output is given as a matrix with the inertia in the diagonals.

$$J_{\text{total}} = \text{diag}([J_{f,x}, J_{f,y}, J_{f,z}]) + \text{diag}([J_{r,x}, J_{r,y}, J_{r,z}]) \quad (2.35)$$

where:

J_{total}	is the total inertia for the LV.	$[\text{kg m}^2]$
$J_{f,xyz}$	is the inertia of the fuel for the chosen rotation.	$[\text{kg m}^2]$
$J_{r,xyz}$	is the inertia of the LV without fuel for the chosen rotation.	$[\text{kg m}^2]$

2.4 Disturbances

In order to send the rocket to space, it must travel through the atmosphere of the Earth. The atmosphere has been divided into multiple layers depending on their general characteristics [23]. Here, it is important that the rocket can withstand the different and changing effects of atmospheric conditions and winds throughout all the layers of the atmosphere. This is the main disturbance however there is also other disturbances introduced in the system through unwanted dynamics, uncertainties, and forces. Therefore, this section will discuss the different disturbances and how to deal with them.

2.4.1 On launch pad

Already on the launch pad will the rocket experience disturbances. These disturbances are mainly wind shears and wind gusts. However, while the rocket is on the launchpad we will not have to control it.

2.4.2 In flight

The most frequent external disturbances that occur in flight is either wind shear or gusts, alongside with pressure differences. Other disturbances are also occurring because of the structure of the design of the launch vehicle. These are disturbances such as fuel sloshing, bending modes, tail-wag-dog effect, and parameter uncertainty.

Wind shear

Wind shear is the difference between wind velocities measured at two heights divided by the height interval [24]. Wind shear can occur at multiple different weathers. The most dangerous would be in thunderstorms and similar. However, this is not a realistic example to be looking at because a launch would never be allowed in such a rough weather. Wind shear can nonetheless still happen at more quiet weathers whenever there is not a completely uniform wind. Wind shear is mostly looked upon when the launch vehicle is still on the ground and mounted to the launch pad. Here the objective is to keep the rocket on the launch pad and analyses the structural load happening on the rocket [25]. And while it also happens during flight the disturbance that will be considered this project is a uniformly distributed wind.

Wind

Wind shear along with gusts and turbulence disturbs the orientation and the trajectory of the rocket. In order to take it into account is it needed to make some choices, and some decision about the modelling of the winds and the control method.

Different weather conditions have different effects on the system. Strong winds can be quite a challenge however it is usually not relevant as a rocket launch will simply be cancelled if such conditions are present. However, in more calm weather it also only affects the LV with a smaller amount, and this is what we will initially consider. That is, we will assume wild weather [25].

Pressure gradient

The atmospheric pressure changes along the altitude. However, the pressure experienced by the launch vehicle is not only dependent on the different atmospheres along the trajectory but it is also dependent on the velocity of the rocket and the density of the air[26]. It is however still possible to estimate the maximum pressure that the rocket is exposed to since the velocity is already estimated along with the trajectory through all the phases of the launch. Other disturbances are mostly the wind shear and gusts which also increase the pressure and change along the altitude.

A way of modeling the static pressure and the density of the atmosphere along the altitude can be estimated with an exponential model [27]:

$$\rho = \rho_0 \cdot e^{-\frac{h}{H}} \quad (2.36)$$

Where ρ_0 is the base density of the medium. H is the scale height that depends upon the selected average isothermal temperature. ρ and H are tweaked to obtain the best possible match for the standard atmosphere in a certain range of altitudes.

It is possible to estimate the static pressure in the atmosphere along the altitude. For an illustration of this gradient, there has been created an example showing the static pressure on a logarithmic scale on the y-axis in pascal and the altitude on the x-axis in meters. Figure 2.6 shows pressure gradient:

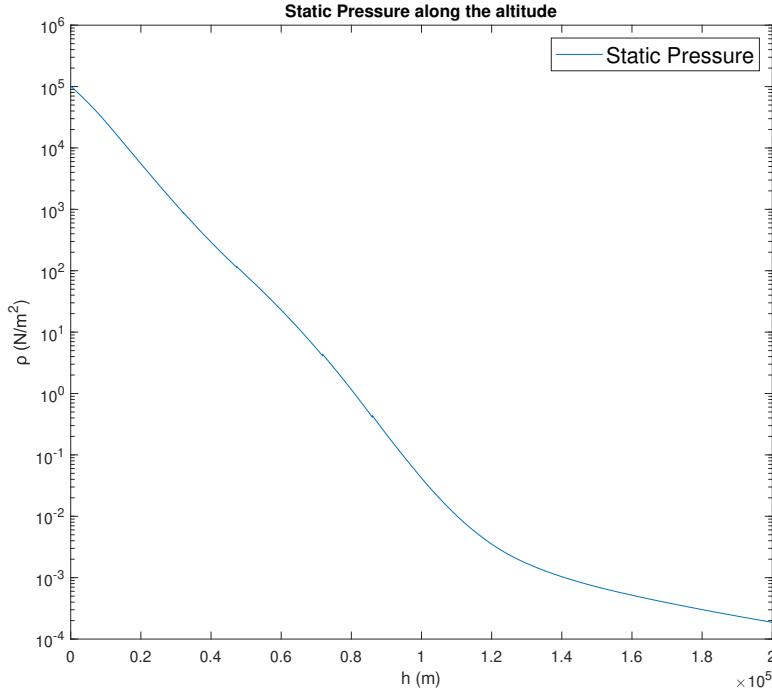


Figure 2.6: An example showing the static pressure on a logarithmic scale [27].

Fuel Sloshing

Fuel Sloshing is a disturbance which only occurs with the use of liquid fuel. It is the forces that happens when the fuel is slamming back and forth inside the fuel tanks because of the maneuvers done during the launch. If the launch rocket is at straight upward direction and is acceleration in the same direction does the fuel does not move a lot from side to side. This is because both the gravity and the acceleration force are pushing the fuel down to the bottom of the tanks. However, this can still produce pendulum like effect that can make the LV unstable if exited at its resonance frequency.

This can be modelled by mass-spring-damper system to simulate the back and forth sloshing behavior. With this method it is possible to simulate the same effect of a mass sloshing inside the launch vehicle [28].

Bending modes

The high pressures affecting the launch vehicle combined with the size of the rocket makes the launch vehicle able to bend. This is not wanted and can be seen as a disturbance, and if left unchecked the control system can in the worst-case scenario excite these oscillations making the system unstable. Therefore, is it also important to investigate these bending modes of the launch vehicle such that the control system can avoid amplifying these oscillations. The bending modes of the launch vehicle can be implemented into the model of the system [26].

Tail-wag-dog effect

Tail-Wag-Dog effect is a phenomenon that occurs when the engines are angled in the bottom of the launch vehicle. This moving of the mass of the engines creates a reaction torque that acts on the launch vehicle. Which is because the inertia of the engines. This should be considered in order to ensure that the vibrations do not become excited since this could introduce instability. However, it can also be considered in the dynamic model of the launch vehicle just like the bending modes [26].

Parameter uncertainty in control system

Parameter uncertainty is always a problem in the plant model. Even with the precise manufacturing often applied when building rockets there will be some uncertainties. In a system in such extreme conditions this must be considered. This is a major part of robust control.

2.4.3 Wind disturbance model

For this project the main disturbance that will be considered is the varying wind. Before starting to model the wind, there are some decisions that should be made regarding the precision of the model and the time limit. The modeling is determined by the specific phase of the launch. For instance, the first phase of the launch could use more simplified mathematical models of the physical system while the other phases could use models with increasing detail and accuracy [25]. When modeling wind and when testing the durability of the launch vehicle, stochastic models are used because wind is hard to predictable and therefore, the stochastic models is the best way to capture the uncertainties in the models. Since this project will not concern itself with making disturbance models for wind, we have decided to use an already existing model and implement this into our control loop.

A wind model can be created using a Dryden model which is a filter for white noise. The Dryden model is, frequently used by NASA among others when modeling wind turbulence [29], and there is a MATLAB tool called "Dryden Wind Turbulence Model" [30]. Therefore, this will be the chosen method of handling the simulation of wind disturbance. This model of the filter has the following mathematical expression [16]:

$$G_{\text{wind}}(s) = \frac{v_w(s)}{n_w(s)} = \frac{\sqrt{\frac{2}{\pi} \frac{V_w(h)}{l_w(h)} \sigma_w^2(h)}}{s + \frac{V_w(h)}{l_w(h)}} \quad (2.37)$$

The Dryden model is a way of modeling the wind disturbances as a transfer function where, $v_w(s)$ is the velocity of the wind disturbances and $n_w(s)$ is white noise. In order to create this transfer function, it is necessary to have the turbulence length scale, $l_w(h)$, and the standard deviation, $\sigma_w(h)$, as a function of altitude. It is also important to have the launch vehicle's vertical airspeed in steady state ($V_w(h)$) [16].

Once the Dryden model has been used to create a filter, it can be implemented into the system model. However, a filter is needed to be created for each level of severity of wind, etc. "light", "moderate", and "severe" wind turbulence. The MATLAB blockset

"Aerospace Blockset" contains a block called "Dryden Wind Turbulence Model" [31] which helps exactly with this and can simulate wind gusts at different levels.

With this filter, it is possible to model the wind disturbance, and therefore the next section will be regarding the trajectory of launch vehicles.

2.5 Trajectory

As mentioned in Section 1.3, even though Vega typical launches satellites at an altitude from 600- to 800 km, the altitude that will be focused on is from the liftoff stage to the MECO stage.

In order to optimally enter orbit around the Earth, the LV initiates a relatively small pitch over maneuver, and then starts what is called a gravity turn. The gravity turn should be started while the vertical velocity is still low to avoid large aerodynamic loads on the vehicle during the maneuver. The gravity turn consists first turning the LV a bit to not be vertical in the direction that the turn should take place. After the pitch over maneuver the engines are reset to point parallel to the axis of the LV again [32]. This slight tilt of the LV causes the LV to gain vertical speed while keeping the angle of attack close to zero due the gravitational force reducing the vertical acceleration but not the horizontal. The velocity vector eventually settles parallel to the direction of total force acting on the LV, however with the direction of total acceleration always changing due to gravitational drag, the velocity vector always lags behind the acceleration. On Fig. 2.7 this concept is illustrated.

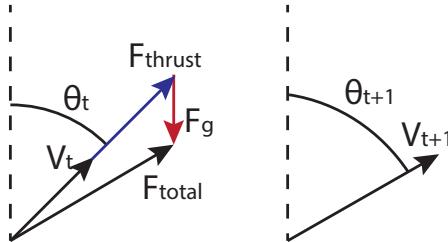


Figure 2.7: Illustration of how the slight tilt of the rocket causes the rocket to tilt even further from time t to time $t+1$.

During the atmospheric stage, the guidance is performed in an open-loop configuration following a preprogrammed trajectory based on attitude as a function of either time or velocity. The reason why it is an open-loop configuration, is because a closed loop configuration would conflict with the main control task of the atmospheric stage, namely load alleviation. The control system must keep the angle of attack low in the presence of atmosphere and wind disturbances. The trade-off of using an open-loop configuration will then naturally be deviations from the reference trajectory. Any final resulting deviations from the nominal trajectory subsequently have to be corrected during the upper stages using closed-loop guidance [26].

The gravity turn helps the guidance system to reduce the AoA by accelerating the LV horizontally, and thus reducing the lateral aerodynamic loads that would otherwise be needed to accelerate vertically.

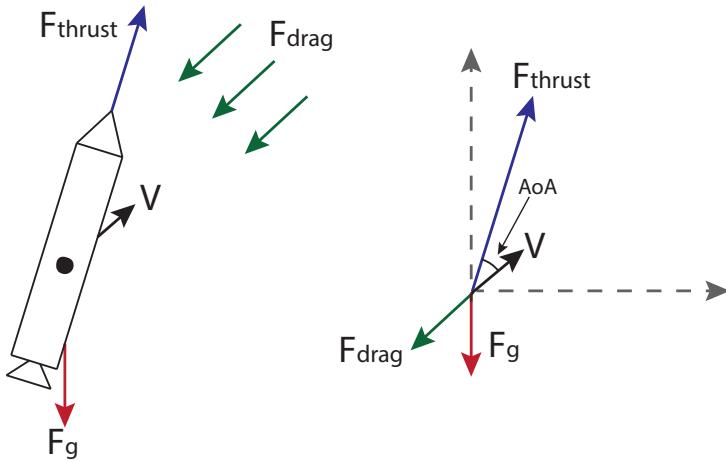


Figure 2.8: Illustration of how the AoA comes to be. The extra load from the gravitational force on the vertical thrust force results in the total force on the LV to become more horizontal.

As for the ascent of the Vega launcher, the liftoff stage begins by igniting the P80 stage which then burns for 107 seconds. The gravity turn initiates after four seconds of vertical flight, just enough time to avoid any possible collisions with the launch pad [26]. By the time of MECO the Vega launcher should have reached an altitude of 44 km, a downrange of 64 km due to the gravity turn and a relative velocity of 1825 m/s [33].

2.6 Controller Choice

The controller choice is important since it will determine what controller will be focused on in this project. Therefore, will this section go into detail about the pros and cons for two types of robust controllers, namely the \mathcal{H}_∞ controller or Linear Parameter Varying (LPV) control.

2.6.1 Linear Parameter Varying Control

The primary reason for the LPV to be chosen is because of the characteristics for the system of a rocket launch. The system is non-linear and have multiple highly varying parameters. LPV, as the name is suggesting, is useful for dynamic systems where the parameters are varying. Along with it, it is also useful for non-linear systems with linearization. Another reason is that LPV is currently taught in the course called "Non-linear Control Systems" in this semester and is therefore a fitting choice since it already matches the characteristics for the system.

In order to create an LPV controller is it needed to have a state-space representation of the given system [4]. However instead of a standard state-space are the matrices continuous dependent on a vector of time-varying parameters [4].

The requirements for an LPV controller system can be expressed in the terms of weighting functions which are selected based on the requirements that the controller might have [4]. This is a good way of implementing the prioritization of the different requirements above each other in the controller.

Another thing to note is that in order to design an LPV controller, is it needed to setup the same design framework as for the \mathcal{H}_∞ synthesis. Therefore, when designing an LPV it is convenient to have the framework for a \mathcal{H}_∞ controller already been made.

2.6.2 \mathcal{H}_∞ Control

As mentioned before, the \mathcal{H}_∞ design framework is the same used in LPV but with static matrices. Therefore, the groundwork for the \mathcal{H}_∞ can be used when developing the LPV. The weights are still used in the \mathcal{H}_∞ and can still be used to express the prioritization of the different requirements. The main difference between LPV and the \mathcal{H}_∞ is just the parameter variation. The \mathcal{H}_∞ can be considered a simpler version of the LPV design. The \mathcal{H}_∞ does not consider for parameter variation by default but it can be implemented as a disturbance, which however does not capture the variation over time that the LPV does.

Therefore, the first controller developed for the LV system in this report, is a \mathcal{H}_∞ controller followed by an LPV.

2.6.3 Uncertain Modelling

When designing a robust controller, either a \mathcal{H}_∞ controller or an LPV controller, they can each take uncertainties into account. These uncertainties are where the actual value of either a measured or predicted value is dispersed within a certain margin. These are not to be mistaken by varying parameters from LPV control. The uncertainties are usually modelled as a noise in the model as a Gaussian distribution with a nominal value and a variance. A robust controller is then designed to make the worst-case dispersion of the uncertainties stable in the closed loop system.

However, in this report the uncertainties are not prioritized when designing the controllers later in Chapter 5. How the uncertainties are modelled will only be examined, but not implemented in the final model, since this has not been prioritized in this project. Modelling uncertainties is something that can be done in future work.

Sensor noise

Sensor noise in the system is usually modeled as an additive uncertainty considered as opposed to the multiplicative uncertainty from parameters and can be added to the model by an additional input. The noise of the sensor will be implemented in the feedback in the closed loop controller on the states that are desired to be controlled.

2.7 Simulation

Once a controller is synthesised it is desired to validate the controller in a real scenario as it can sometimes be hard to see based on the numbers whether the controller will work as has been experienced many times during this project. It has therefore been chosen to develop a high-fidelity simulation in order to validate the final controller for the LV's trajectory.

There are multiple different ways to implement a simulation suitable for this project. However, since the university have access to MATLAB and the group have some earlier

experience with the tool Simulink from MATLAB, and since it is a very intuitive way to implement a model, it has chosen as the platform for this project.

In Simulink is it possible to model and simulate the system [34]. It is also possible to implement other author's toolboxes as blocks inside the system. This could help with implementation complex models such as aerodynamics and wind disturbances. Another advantage with Simulink is that the system can be separated into blocks with clear cut interfaces between each other.

Launches of real LVs is costly and can be a disaster if failed and is not a viable way to test the controller in this project 1.3. Therefore has it been chosen to make the simulation as high fidelity as possible in order to test the controller as close to real life situation as possible so that this simulation can be used to validate the controller .

2.8 Conclusion

Throughout Chapter 2, dynamics needed to construct a LV model have been explained in section 2.2. The reference frames, and both the translational- and rotational dynamics have been evaluated. The disturbance that affects a LV during launch have been analysed for in section 2.4 and lastly a controller has been chosen for development to stabilize the LV during the planned trajectory in section 2.6. However, it is needed to test the controller's effectiveness, and durability. This has been chosen to be done by full nonlinear simulation in section 2.7.

Therefore, looking forward will this project contain the development of three 2-D state space representations of the LV. One for each rotational axis: roll, yaw, and pitch.

The controller will be an LPV controller for the pitch, yaw, z drift, y drift and roll. But in order to create an LPV is it needed to first have the correct system structure. This is best obtained through a \mathcal{H}_∞ analyses and therefore a \mathcal{H}_∞ controller will first be developed with focus on later creating an LPV controller.

As mentioned in the case description 1.3 has it been decided to have multiple engines on the LV. The dynamics of this can be seen in the subsection 2.2.4, however, it is needed to analyses the controllability of the different engine setups. This will be later in chapter 4.2.

2.8.1 The Problem formulation

With these choices is it possible to specify the final problem formulation based on the problem analysis and the initial problem formulation in subsection 1.3.1 for this project: *How can a structured \mathcal{H}_∞ and an LPV controller be designed with appropriate weights to make a launch vehicle follow a predetermined trajectory from lift-off to MECO inside a high-fidelity simulation developed in Simulink.*

Requirement Specification

3

This Chapter will outline the goals and requirements for this project. First the functional requirements will describe the overall goals of the controller, and then afterwards testable technical requirements will be specified

3.1 Requirements Specification

In order to ensure that the LV meets certain expectations, functional and technical requirements are made which will have to be fulfilled by the final controller. The functional requirements will evaluate how the design of the LV will work entirely from a functional perspective. These requirements are evaluated subjectively and are fulfilled if the corresponding technical requirements are met. The technical requirements have been specified with values and can therefore be measured and determined if the requirements have been met or not.

The requirements are derived from the problem formulation 2.8.1 and will therefore contain requirements for a \mathcal{H}_∞ and an LPV controller.

3.1.1 Functional Requirements

The functional requirements are mostly made from the desired functions of the controller. What kind of disturbances it should be able to handle? The functional requirements can be seen in the table 3.1 with a priority ranking followed with an explanation about why they are chosen.

Priority	Functional Requirements
1	Follow the predetermined trajectory
2	Follow the trajectory with disturbances
3	Follow the trajectory with disturbances and parameter uncertainty

Table 3.1: The functional requirements for the system.

The launch vehicle needs to be able to follow the predetermined trajectory. This requirement is set to require that the control system is can correctly follow the trajectory without any loss of vehicle. The requirement comes from the trajectory section 2.5.

The second functional requirement to make the control system robust and able to handle disturbances. Such as, wind gusts, wind shear and parameter inaccuracy/error. The amount of wind the LV has been chosen to withstand are "light" weather conditions.

These light weather conditions are assessed to be from 0 m/s to ≈ 2.1 m/s [31] using the same method as found in section 2.4.

The third requirement includes the parameter uncertainties, where the controller takes the uncertain parameters into account. This can both be considered in the \mathcal{H}_∞ and the LPV with an upper fractional transform and a delta block.

3.1.2 Technical Requirements

In the list below, a set of well-known and relevant challenges for a flight control system are stated [35]:

1. *Stability Indicators:*

The launcher shall be actively stabilized while coping with parameters that have a highly dynamical variation and uncertainty level due to its aerodynamically unstable motion. In addition, prespecified Gain Margins (GMs) and Phase Margins (PMs) shall be complied with, as shown in Table 3.2.

2. *Attitude Tracking:*

In a steady state the attitude tracking error shall converge to zero. Moreover, the transient response shall be constrained in terms of maximum rate and overshoot.

3. *Load and Drift Management:*

An optimized balance between load vs drift trade-off must be achieved. This is because the launcher lateral drift from its reference trajectory shall be limited, but in order to do so, it has to contradict minimizing the aerodynamic load by pitching away from the wind field and thereby increasing the AoA.

4. *Actuation Minimization:*

The demanded TVC actuation during the flight shall never reach its deflection and bandwidth limits.

5. *Disturbance Rejection:*

Rejecting disturbances such as wind gusts and internal dynamics.

These requirements can contradict each other, and hence involves a control design which exploits the underlying trade-offs while operating as close to the limits as possible. For example, when minimising the load, it involves keeping the AoA small by pitching into the wind field, which causes the attitude errors to grow. The achievable performance is determined by the degree of sophistication of the controller and by the physical limitations of the system [16].

Trajectory to follow

ESA has provided a trajectory which will be used as the target trajectory of the LV during flight. This trajectory will be used as the reference when doing the tests to determine the success of the tests done.

The trajectory is split into two parts. The first part is the required pitch for the trajectory, the second part is the thrust given from the engines

The pitch trajectory can be seen on figure 3.1:

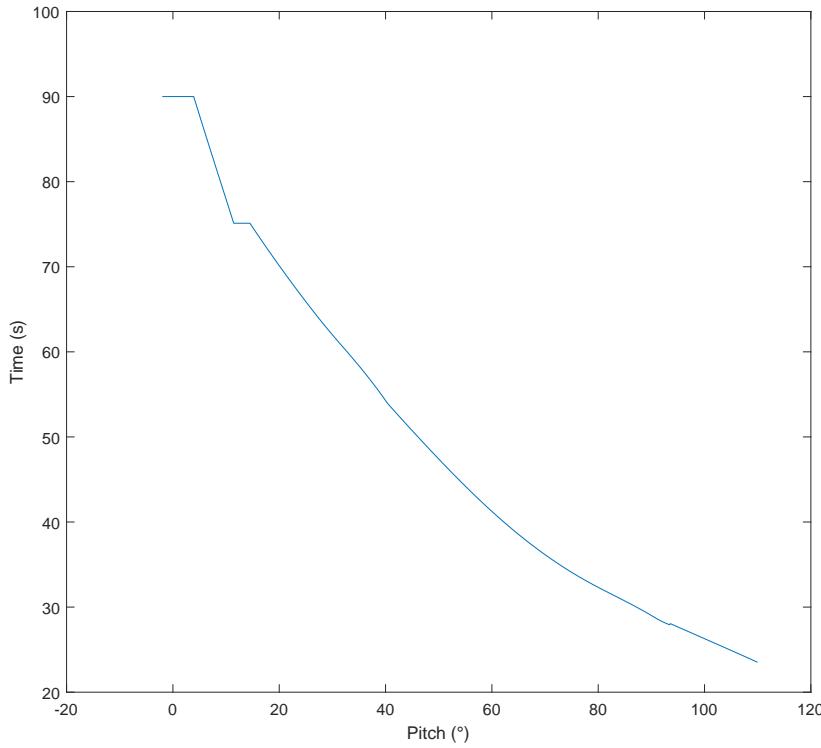


Figure 3.1: The pitch trajectory dependent on time given from ESA.

The thrust trajectory can be seen on figure 3.2:

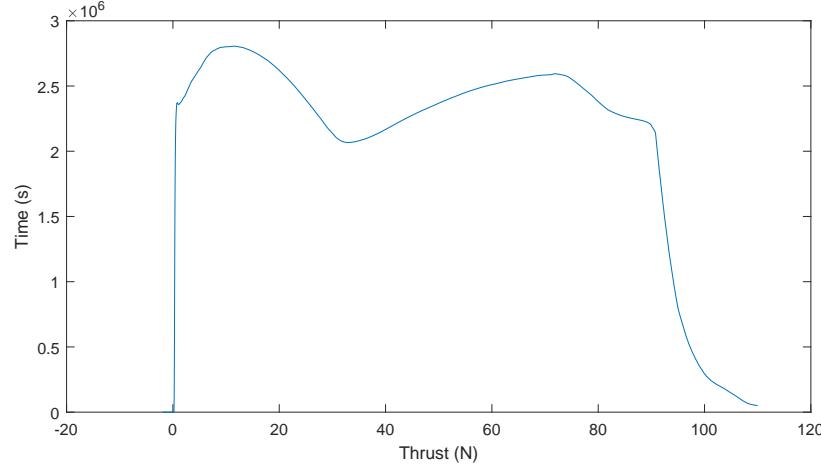


Figure 3.2: The thrust trajectory dependent on time given from ESA.

Stability Requirements

The stability of the LV TVC system is analysed in the frequency domain in terms of classical stability margins stated in Table 3.2. The stability requirements are defined for nominal conditions. Only three rigid body-body margins are considered, namely Low-Frequency Gain Margin (LF-GM), Delay Margin (DM) and High-Frequency Gain Margin

(HF-GM). Note that at the margin that is computed, the Phase Margin (PM) is expressed as the equivalent delay. It is important to note that the stability specifications are to be verified assuming low roll rate of the LV. Thereby, the test can be performed in either the pitch- or yaw axis [26].

Metrics		Bounds
LF-GM	Nominal	$\geq 6 \text{ dB}$
DM	Nominal	$\geq 100 \text{ ms}$
HF-GM	Nominal	$\leq -6 \text{ dB}$

Table 3.2: Stability requirements for the LV TVC system

Performance Requirements

One of the most important requirements during the atmospheric flight is to keep the structural load within a certain range. This can be ensured with an envelope of a performance parameter Q_α . This envelope is based on the Mach number of the LV. This Q_α is the product between the dynamic pressure and the angle of attack. This makes the requirement sensitive to wind disturbances. Moreover, the guidance is in open loop, meaning the TVC system must also limit the drift with respect to the reference trajectory 2.5. That is for both the position (y, z) and the velocity (\dot{y}, \dot{z}). In addition, to reduce the TVC consumption and avoid saturation of the actuators, it is important to limit the actuation effort of the TVC actuators. These different performance metrics can be seen in Table 3.3, and must remain under their given bounds even with dispersion from their nominal values coming from disturbances such as noise and wind.

Requirements	Metrics	Bounds
Load performance	Q_α	$< 2.75e5 \text{ Pa}$
Lateral control performance	Position (y, z)	$< 500 \text{ m}$
	Velocity (\dot{y}, \dot{z})	$< 15 \text{ m/s}$
Actuation performance	β	$< 6.5^\circ$
	Integrated β	$< 250^\circ$

Table 3.3: Performance requirements for the TVC system [26] [4]

The Q_α contains two variables, the maximum pressure and the AoA, which needs to fulfill the total requirement of $2.75e5$ [4]. When performing tests on the controllers is it possible to simulate the pressure which the LV is exposed to along the trajectory. If no other disturbances, like wind or deviance in the AoA, are added should it be possible to estimate this number and not deviate too much from test to test. With this it is possible to determine the maximum pressure happening on the LV during the given trajectory. Now that the maximum pressure that the LV is exposed to during the trajectory is determined can the maximum of the other part of the Q_α be determined, namely AoA.

With this is it possible to set a maximum on the allowed error of the AoA. However, this also means that the requirement cannot be set before the first test or simulation of the trajectory with the given LV. The maximum error on the AoA will be determined in the results chapter 7.

Modelling and Linearization 4

This chapter is going to present the linearization of the nonlinear equations of motion of the rigid LV body presented back in subsection 2.2.2. First, the translational dynamics will be linearized with necessary linear approximations presented along the process. The linearization is done for an arbitrary thruster. The actual thruster dimensions and parameters can then be set for the individual thruster later. Secondly, the rotational dynamics will be linearized as well, using most of the approximations used in the linearization of the translational dynamics. Next, the linearized equations of motion will be expanded to include four thrusters located as seen back in Fig. 2.5A. A state space representation is set up for the whole linearized system. Next, a controllability comparison analysis is presented of two thruster configurations. Finally, a TVC actuator model and a delay models will be evaluated.

In Fig. 4.1, a model of the LV with all the relevant parameters is visualized in the BRF defined back in subsection 2.2.1. The sketch is only showing two thrusters for simplicity, as they are considered sufficient to show the dynamics.

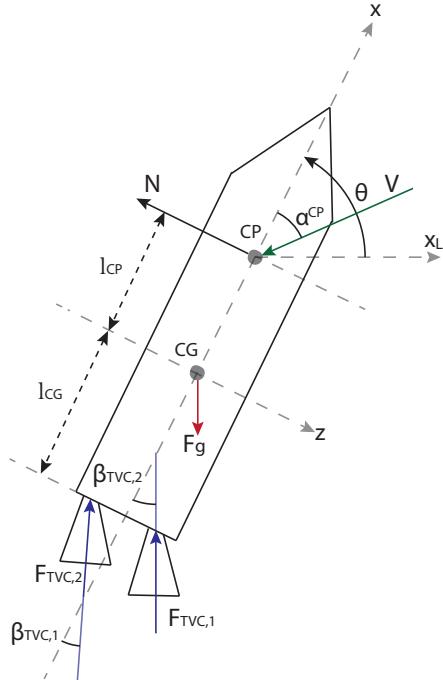


Figure 4.1: Illustration of relevant forces and measurements that acts upon a LV during an ascent

4.1 Linearized Equation of Motion

In this section the linearized equations of motion will be derived. Largely following the derivation in [16]. The nonlinear equations that are analyzed, are Eq. (2.8) and Eq. (2.9) rewritten into the axes in the BRF. That is the control problem have been split into 3 2D problems instead of one 3D problem to simplify the controls. This is considered sufficient approximation as the control interference between the axes is assumed small [16]. They will be derived w.r.t. to drift z and drift rate \dot{z} as well as pitch θ and $\dot{\theta}$ from subsection 3.1.2 respectively.

4.1.1 Linearizing Translational Dynamics

First, the translational motion of the vehicle is evaluated for one thruster, parallel to the x-axis in the body frame. The expression from subsection 2.2.2 is restated below:

$$\ddot{r}_I(t) = \dot{v}_I(t) = g_I(t) + \frac{1}{m(t)} [F_{aero}(t) + F_{TVC}(t)] \quad (4.1)$$

Referring to Fig. 4.1, the sum of forces in the z-axis are given in Eq. (4.2).

$$m(\ddot{z} - \dot{\theta}\dot{x}) = -Q_d S C_N(\alpha^{CP}) - \underbrace{F_{TVC} \sin \beta_{TVC}}_{F_{TVC,z}} + \underbrace{F_g \cos \theta}_{F_{g,z}} \quad (4.2)$$

where θ represents the pitch angle, z represents the lateral drift and β_{TVC} represent the angle of how much the thruster is gimballing compared to the longitudinal axis. Note here that the dependence of time is omitted due to improved readability, as well as the subscript "y" for pitch plane TVC deflections.

On the right-hand side of the equations the thruster force and gravitational force relative to the z-axis stated. The thruster force relative to the z-axis is given by Eq. (2.25). A visual representation on why the gravitational force in the z-axis are given with a cosine, can be seen in Fig. 4.2.

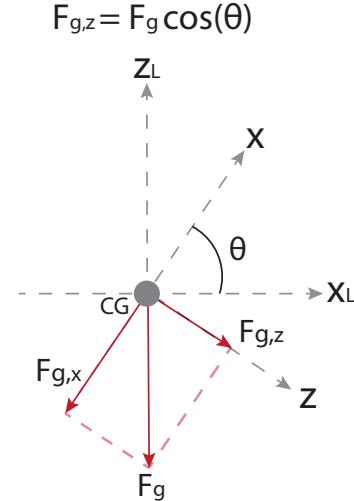


Figure 4.2: Illustration of how the gravitational force is given relative to the z-axis.

Furthermore, C_N is the normal aerodynamic LV force coefficient as is given by:

$$C_N(\alpha(t), M(t)) = C_L(\alpha(t), M(t)) \cos \alpha(t) + C_D(\alpha(t), M(t)) \sin \alpha(t) \quad (4.3)$$

The acceleration of the lateral drift is also affected by how fast the LV is changing the pitch angle and how fast its going in the x-direction. This is because the velocity stays the same when the LV rotate, but will get a lateral component in the body frame. This will be considered a drift along the z-axis.

Steady State Solutions

To find an operating point a steady state is desired, certain parameters must therefore be found to make the lateral velocity constant. The steady state solution to Eq. (4.2) consists in the set of AoA, pitch and TVC angles, which α_0 , θ_0 and $\beta_{TVC,0}$ respectively.

The angular velocity of the pitch then becomes the constant pitch rate, denoted q_0 , and $V \cos \alpha_0$ becomes the constant longitudinal velocity, (i.e. along the x-axis). Since the trajectory is following a gravity turn these are all assumed to be zero in steady state. Eq. (4.4) is then given when $\ddot{z} = 0$.

$$-mq_0V \cos \alpha_0 = -Q_d S_{ref} C_N \left(\alpha_0^{CP} \right) - F_{TVC} \sin \beta_{TVC,0} + F_{g,z} \cos \theta_0 \quad (4.4)$$

Perturbed Equations of Motion

To linearize the translational motion, each term is replaced by the corresponding steady-state and a perturbation in the form of $\nu = \nu_0 + \delta\nu$, where ν_0 is a chosen operating point and $\delta\nu$ is the variation. Applying this to Eq. (4.2) and combining it with Eq. (4.4) yields:

$$m \left(\delta \ddot{z} - q_0 V \cos \alpha_0 - q_0 \delta \dot{x} - V \cos \alpha_0 \delta \dot{\theta} - \delta \dot{\theta} \delta \dot{x} \right) = -Q_d S C_N (\alpha_0^{CP} + \delta \alpha^{CP}) - F_{TVC} \sin (\beta_{TVC,0} + \delta \beta_{TVC}) + F_g \cos (\theta_0 + \delta \theta) \quad (4.5)$$

where the left-hand side of the equation comes from the term $(q_0 + \delta \dot{\theta})(V \cos \alpha_0 + \delta \dot{x})$.

Aerodynamics Linearization

The aerodynamic forces are linearized around the steady-state AoA using the corresponding force gradient coefficient. The normal force, applied at its Centre of Pressure (CP), is then approximated by:

$$C_N (\alpha_0^{CP} + \delta \alpha^{CP}) \approx C_N (\alpha_0^{CP}) + C_{N_\alpha} \delta \alpha^{CP} \quad (4.6)$$

Small Angle Approximations

Below are some well-known approximations for functions with small perturbations $\delta \nu$, that will be used next.

$$\delta \nu \delta \nu \approx 0 \quad (4.7)$$

$$\cos \delta \nu \approx 1 \quad (4.8)$$

$$\sin \delta \nu \approx \delta \nu \quad (4.9)$$

$$\cos (\nu_0 + \delta \nu) \approx \cos \nu_0 - \sin (\nu_0) \delta \nu \quad (4.10)$$

$$\sin (\nu_0 + \delta \nu) \approx \sin \nu_0 + \cos (\nu_0) \delta \nu \quad (4.11)$$

In Eq. (4.10) and Eq. (4.11) the sinusoidal functions are approximated with a first order Taylor approximation.

Using Eq. (4.5) as a baseline, the translational dynamics can be rewritten by using the approximations in Eq. (4.6), Eq. (4.7), Eq. (4.10) and Eq. (4.11). Furthermore, Eq. (4.4) is used to remove the steady-state terms. In [16] it is assumed that the steady state constants are cancelling each other out when summed together, thus making it linear and not affine.

$$m \left(\delta \ddot{z} - q_0 V \cos \alpha_0 - q_0 \delta \dot{x} - V \cos \alpha_0 \delta \dot{\theta} - \delta \dot{\theta} \delta \dot{x} \right) = -Q_d S \left(C_N (\alpha_0^{CP}) + C_{N_\alpha} \delta \alpha^{CP} \right) - F_{TVC} \left(\sin \beta_{TVC,0} + \cos (\beta_{TVC,0}) \delta \beta_{TVC} \right) + mg (\cos \theta_0 - \sin (\theta_0) \delta \theta) \quad (4.12)$$

Angle of Attack Linearization

Furthermore, the AoA is affected by the pitch $\delta \theta$ and drift variation $\delta \dot{z}$, as well as by wind gusts with velocity v_w . Taking the inverse tangent of Eq. (2.16), AoA perturbations at the LV's CG are given in Eq. (4.13).

$$\delta\alpha \approx \delta\theta + \frac{\delta\dot{z} - v_w}{V \cos \alpha_0} \quad (4.13)$$

At other locations beside from the CG, the local velocity variations also depend on the vehicle's rotation this includes also for the CP. Hence the small perturbation AoA at the LV's CP is given by Eq. (4.14).

$$\delta\alpha^{CP} \approx \delta\alpha - \frac{l_{CP}}{V \cos \alpha_0} \delta\dot{\theta} \quad (4.14)$$

where l_{CP} is the distance from the CG to CP: $l_{CP} = x_{CP} - x_{CG}$.

Inserting Eq. (4.13) into Eq. (4.14) and inserting that into Eq. (4.12) yields:

$$m \left(\delta\ddot{z} - q_0 \delta\dot{x} - V \cos \alpha_0 \delta\dot{\theta} \right) = -Q_d S C_{N_\alpha} \underbrace{\left(\delta\theta + \frac{\delta\dot{z}}{V \cos \alpha_0} - \frac{v_w}{V \cos \alpha_0} - \frac{l_{CP}}{V \cos \alpha_0} \delta\dot{\theta} \right)}_{\delta\alpha^{CP}} - F_{TVC} (\cos(\beta_{TVC,0}) \delta\beta_{TVC}) + F_{g,z} (-\sin(\theta_0) \delta\theta) \quad (4.15)$$

Final Assumptions

Finally, the following assumption is applied:

$$q_0 \delta\dot{x} \approx 0 \quad (4.16)$$

This assumption is considered realistic, because both q_0 and $\delta\dot{x}$ are very small compared to $\delta\dot{z}$. This allows to uncouple the longitudinal and the lateral dynamics. Additionally, according to other works of ascent flight control, it is assumed that the trim angles α_0 and $\beta_{TVC,0}$ are close to zero, meaning $\cos\alpha_0 \approx \cos\beta_{TVC,0} \approx 1$ [16]. From the case description, this report only focuses on the ascent flight control, and thereby it has been chosen to implement this approximation. However, for future work when working with descent flight control, this approximation is not considered adaptable.

Finally, the linear equation of translational motion can be reorganised as the following:

$$m(\delta\ddot{z} - V \delta\dot{\theta}) = - (N_\alpha + F_g \sin(\theta_0)) \delta\theta - \frac{N_\alpha}{V} \delta\dot{z} + \frac{N_\alpha}{V} v_w + \frac{l_{CP} N_\alpha}{V} \delta\dot{\theta} - F_{TVC} \delta\beta_{TVC} \quad (4.17)$$

where $N_\alpha = Q_d S C_{N_\alpha}$.

4.1.2 Linearizing Rotational Dynamics

Now that the translational motion of a LV has been linearized, the rotational dynamics will be linearized likewise. The rotational dynamics from Eq. (2.9) will be the starting point, and is also stated below:

$$\dot{\omega}_B(t) = J^{-1}(t) \left[M_{aero,B}(t) + M_{TVC,B}(t) + M_{fin,B}(t) + M_{thr,B}(t) - \omega_B(t) \times J(t)\omega_B(t) - \dot{J}(t)\omega_B(t) \right] \quad (4.18)$$

Referring to Fig. 4.1, the sum of lateral moments corresponds to

$$J_N \ddot{\theta} = l_{CP} Q_d S C_N (\alpha^{CP}) - l_{PVP} F_{TVC,z} \quad (4.19)$$

where, l_{PVP} is the distance from the CG and the pivot point of the thrusters. In the case of the only one thruster the x_{PVP} is located directly under the CG, $l_{PVP} = x_{PVP} - x_{CG}$.

The steady state solution of the rotational dynamics such that the acceleration of the pitch is to be cancelled out, are parameters within the set of AoA and TVC angles. Below is the steady state solution in terms of $\ddot{\theta} = 0$.

$$0 = l_{CP} Q_d S C_N (\alpha_0^{CP}) - l_{CG} F_{TVC} \sin \beta_{TVC,0} \quad (4.20)$$

Applying the steady states with perturbations on each term on Eq. (4.19) yields:

$$J_N \ddot{\theta} = l_{CP} Q_d S (\alpha_0^{CP} + \delta \alpha^{CP}) - l_{PVP} F_{TVC} \sin (\beta_{TVC,0} + \delta \beta_{TVC}) \quad (4.21)$$

Using the approximation in Eq. (4.6), the small angle approximation from Eq. (4.11) and Eq. (4.20). It is assumed that the sum of steady state constants is cancelling each other out, thus making it linear and not affine.

$$\begin{aligned} J_N \delta \ddot{\theta} &= l_{CP} Q_d S \left(\underline{C_N(\alpha_0^{CP})} + C_{N_\alpha} \delta \alpha^{CP} \right) \\ &\quad - l_{PVP} F_{TVC} \left(\underline{\sin(\beta_{TVC,0})} + \cos(\beta_{TVC,0}) \delta \beta_{TVC} \right) \end{aligned} \quad (4.22)$$

Replacing using the approximation in Eq. (4.14), yields:

$$\begin{aligned} J_N \delta \ddot{\theta} &= l_{CP} Q_d S C_{N_\alpha} \overbrace{\left(\delta \theta + \frac{\delta \dot{z}}{V \cos \alpha_0} - \frac{v_w}{V \cos \alpha_0} - \frac{l_{CP}}{V \cos \alpha_0} \delta \dot{\theta} \right)}^{\delta \alpha^{CP}} \\ &\quad - l_{PVP} F_{TVC} \cos(\beta_{TVC,0}) \delta \beta_{TVC} \end{aligned} \quad (4.23)$$

Finally, using the assumption in Eq. (4.16) and that the trim angles α_0 and $\beta_{TVC,0}$ are close to zero, yields:

$$\delta \ddot{\theta} = \mu_\alpha \delta \theta + \frac{\mu_\alpha}{V} \delta \dot{z} - \frac{\mu_\alpha}{V} v_w - \frac{l_{CP} \mu_\alpha}{V} \delta \dot{\theta} - \mu_{CG} \delta \beta_{TVC} \quad (4.24)$$

where:

$$\mu_\alpha = \frac{N_\alpha}{J_N} l_{CP}, \quad \mu_{CG} = \frac{F_{TVC}}{J_N} l_{CG}$$

are the moment proneness coefficients.

4.1.3 Implementing multiple thrusters

Before putting the translational and rotational dynamics into a state space representation, the equation is expanded into having four thrusters, which each have an offset to the origin in the yz -plane instead of only one thruster in origin. Each thruster will be set to have 1/4 of the produced thrust from the single engine of the Vega rocket, meaning $F_{TVC,1} = F_{TVC,2} = F_{TVC,3} = F_{TVC,4} = \frac{1}{4}F_{TVC}$. By adding a thruster away from the origin will result in an offset in the β_{TVC} angle. The offset angle can be calculated with:

$$\beta_{\text{off}} = \arctan \frac{l_{\text{off}}}{l_{\text{CG}}(t)} \quad (4.25)$$

Keep in mind the length of $l_{\text{CG}}(t)$ is varying with time, because the CG will change due to fuel combustion. This is evaluated back in subsection 2.3.3.

With the thruster offset, the thrust vector will impact the CG relative to l_{PVP} instead of l_{CG} . This causes the offset angle that must be considered. Furthermore, due to the offset, $F_{TVC,z}$ no longer becomes the normal force on the CG, so a new force is introduced, $F_{TVC,N}$, thus changing the moment. In Fig. 4.3A a visualization of the β_{off} and $F_{TVC,N}$ has been made. Since the modelling is done with respect to the z -axis, the projection from P_{PVP} onto the z -axis is applied, and $l_{\text{PVP},z}$ and $l_{\text{off},z}$ is introduced. This is visualized in Fig. 4.3B. Making the $\beta_{\text{off},z}$ be computed as:

$$\beta_{\text{off},z} = \arctan \frac{l_{\text{off},z}}{l_{\text{CG}}(t)} \quad (4.26)$$

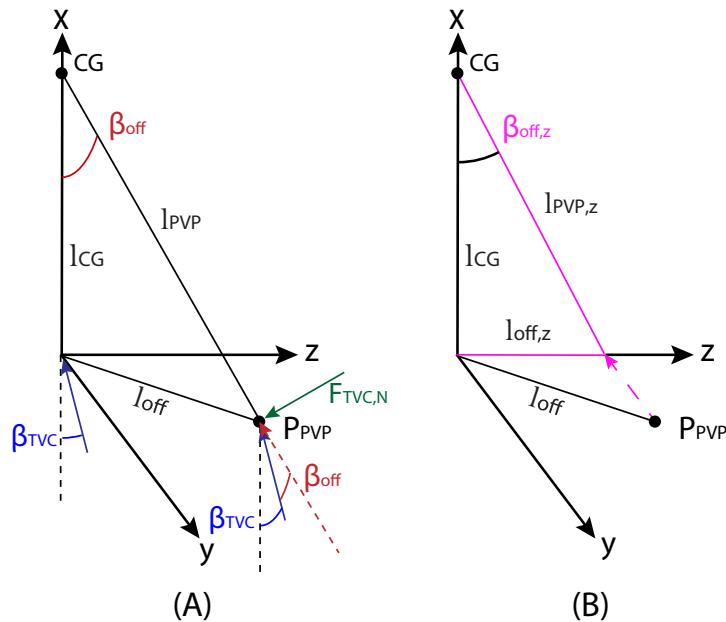


Figure 4.3: Illustration of how the offset angle and the normal TVC force come to be, due to moving the thrusters away from origin of the yz -plane.

With the introduction of the offset angle, the offset will be either added to- or subtracted from the trim angle $\cos(\beta_{TVC,0} \pm \beta_{off})$ when considering rotation and is therefore no longer considered to be close to 1. From the definition in Fig. 2.5A, the off angle is added in P_{PVP1} and P_{PVP4} , while in P_{PVP2} and P_{PVP3} is subtracted. However, while the trim angle $T_{TVC,0}$ is considered to be close to zero, the relation $\cos(-x) = \cos(x)$ is used.

It is considered that the input β_{TVC} is the same for all four thrusters, resulting in them turning synchronized.

The resulting equations for the translational and the rotational dynamics will then respectively become.

$$m(\delta\ddot{z} - V\delta\dot{\theta}) = -(N_\alpha + F_g \sin(\theta_0))\delta\theta - \frac{N_\alpha}{V}\delta\dot{z} + \frac{N_\alpha}{V}v_w + \frac{l_{CP}N_\alpha}{V}\delta\dot{\theta} - F_{TVC}\delta\beta_{TVC} \quad (4.27)$$

$$\delta\ddot{\theta} = \mu_\alpha\delta\theta + \frac{\mu_\alpha}{V}\delta\dot{z} - \frac{\mu_\alpha}{V}v_w - \frac{l_{CP}\mu_\alpha}{V}\delta\dot{\theta} - \mu_{PVP} \cos(\beta_{off,z})\delta\beta_{TVC} \quad (4.28)$$

where $\mu_{PVP} = \frac{F_{TVC}}{J_N}l_{PVP,z}$.

4.1.4 State Space Representation

Now that the linear dynamic equations have been derived in Eq. (4.27) and Eq. (4.28) they are organised into a state space representation in Eq. (4.29) on the form $\dot{x} = Ax + Bu + B_d v_w$.

$$\begin{bmatrix} \delta\dot{\theta} \\ \delta\ddot{\theta} \\ \delta\dot{z} \\ \delta\ddot{z} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \mu_\alpha & -\frac{l_{CP}\mu_\alpha}{V} & 0 & \frac{\mu_\alpha}{V} \\ 0 & 0 & 0 & 1 \\ -\frac{N_\alpha + F_g \sin \theta_0}{m} & \frac{l_{CP}N_\alpha}{mV} + V & 0 & -\frac{N_\alpha}{mV} \end{bmatrix} \begin{bmatrix} \delta\theta \\ \delta\dot{\theta} \\ \delta z \\ \delta\dot{z} \end{bmatrix} + \begin{bmatrix} 0 \\ -\mu_{PVP} \cos \beta_{off,z} \\ 0 \\ -\frac{F_{TVC}}{m} \end{bmatrix} \delta\beta_{TVC} \quad (4.29)$$

$$\begin{bmatrix} 0 \\ -\frac{\mu_\alpha}{V} \\ 0 \\ \frac{N_\alpha}{mV} \end{bmatrix} v_w \quad (4.30)$$

4.2 Controllability Analysis

This section will explore the controllability of the system. The system is defined as a rocket with four main engines that should be controlled with TVC using those four engines. The control will be split into 3 different, perpendicular planes, namely the pitch, yaw, and roll plane. This section will be concerned with mainly the pitch plane, but because of the symmetry of the rocket, it will also apply to the yaw plane.

4.2.1 Engine Configurations

With four engines there are two main configurations of the four engines relative to the pitch and yaw plane. They are considered main configuration because of the axial symmetry.

The configurations A and B can be seen on figure 4.4.

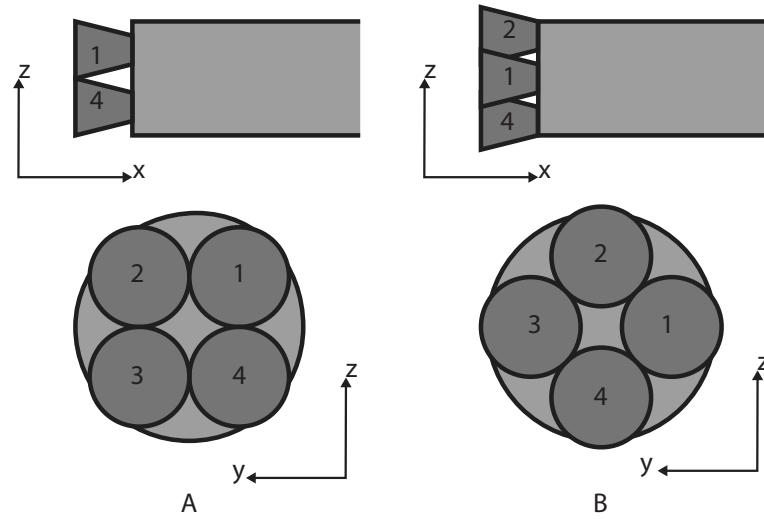


Figure 4.4: The two main engine configurations A and B.

The main difference between the configurations, seen from the controlled two-dimensional planes, are that the effect from pivoting the engines have the different effects on the rocket for some engines, and similar effects on some, assuming same thrust on all engines. In configuration A the engines 1 and 2, and engines 3 and 4 will have similar effects in the xz-plane (because of symmetry this will also be true in xy-plane). Furthermore, they are symmetric around the center and thus the magnitude of their effects will also be similar. Meanwhile, in configuration B, there are 3 different points from which the engines pivot with 3 different effects in the xz-plane. However, as opposed to configuration A, there are 2 different distances from the center in the xz-plane, engine 1 and 3, and engine 2 and 4. This section will therefore explore what effect this can have on the control.

4.2.2 Dynamic Equations for Engines

To analyze the effect the differences of position in the xz-plane the dynamic equations for the engines will be specified. The relevant variables are illustrated in Fig. 4.5. The forces acting on the rocket is simply described by the force vector F . Since we have decided to control for lateral drift, we are interested in the lateral control force F_z which is given by 4.31

$$F_z = F \cdot \sin \beta_{TVC} \quad (4.31)$$

Where F is the force produced by the engine, and β_{TVC} is the angle that is caused by the actuator. This is independent from engine placement so will not be investigated further. The other controlled parameter is pitch, which is the rotation in this case around the CG in the xz-plane. The force contributing to the rotation is the force perpendicular to the line from where the force is applied, P_{PVP} , to P_{CG} . This force is illustrated as F_M on

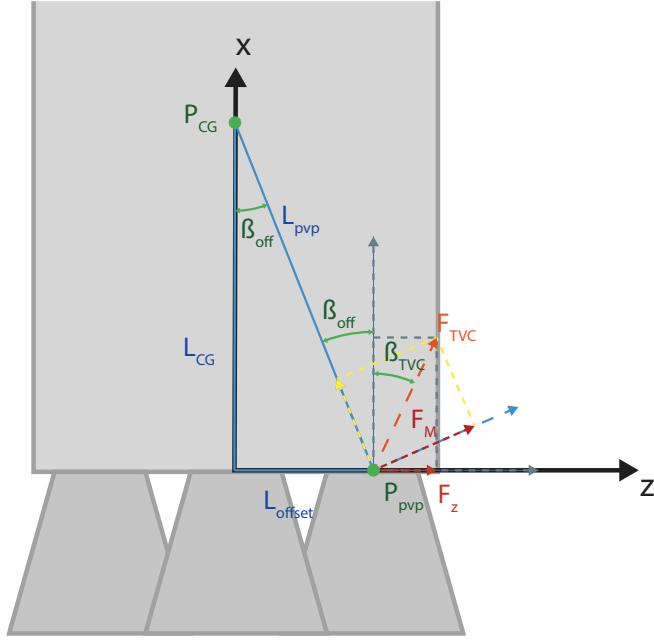


Figure 4.5: Variables used to determine effect of engine offset.

figure 4.5. The force is given by

$$F_M = F \cdot \sin(\beta_{TVC} + \beta_{off}) \quad (4.32)$$

Where β_{off} is the angle caused by the offset of the engine. And the moment around CG, M_F is given by

$$M_F = L_{pvp} \cdot F_M \quad (4.33)$$

The length L_{pvp} changes the further away from the center the engine is placed. This length can be calculated in many ways based on known geometric relations, in 4.34 two are shown, the first is based on the known and easily measured parameters, the second will be used here and is interesting for reasons that will be clear later.

$$L_{pvp} = \sqrt{L_{CG}^2 + L_{offset}^2} = \frac{L_{CG}}{\cos \beta_{off}} \quad (4.34)$$

where L_{CG} is the distance from the bottom of the rocket to the CG along the x-axis. L_{CG} is the lateral from center of the rocket to the engine in the xz-plane. When combining 4.34 and 4.32 into 4.33 we get 4.35

$$M_F = \frac{L_{CG}}{\cos \beta_{off}} \cdot F \cdot \sin(\beta_{TVC} + \beta_{off}) \quad (4.35)$$

When this is linearized as in Section 4.1, we get the following linear relation to the change in actuation angle $\delta\beta_{TVC}$

$$\delta M_F = \left(\frac{L_{CG}}{\cos \beta_{off}} \cdot F \cdot \cos(\beta_{TVC,0} + \beta_{off}) \right) \cdot \delta\beta_{TVC} \quad (4.36)$$

And when the linearization point $\beta_{TVC,0}$ is at 0 which it will be assumed for this project we get 4.37

$$\delta M_F = \left(\frac{L_{CG}}{\cos \beta_{off}} \cdot F \cdot \cos(\beta_{off}) \right) \cdot \delta\beta_{TVC} \quad (4.37)$$

where the effect of the offset captured fully by the angle β_{off} can be canceled out and we get.

$$\delta M_F = L_{\text{CG}} \cdot F \cdot \delta\beta_{\text{TVC}} \quad (4.38)$$

Therefore, with the linearization used for the statespace that is used to design the controller, the placement has no effect. However, it is important to note that there is an effect from this offset as soon as the linearization point is not around zero. This will not be further explored in this project. However, it was shortly looked into and there seemed to be small ways to utilize this difference to optimize actuation effort, but the difference is cancelled by the symmetry as soon as the engines mirrored in the x-axis move with the same control signal. While this will not be further investigated here, it could be worth looking into in future work.

Thus, to conclude this section. While there is a difference in engine placement, it will not have any effect on the linearized control when linearized around zero as terms cancel in that specific instance. Therefore, in this project all engines will be controlled using just one control signal.

4.3 TVC Actuator Model

The TVC control becomes activated in the propelled phase of the atmospheric flight. gimbal the nozzle around two axes via two electromechanical actuators. The TVC input will be from the output the controller and with a small delay with a model described in the next section, Section 4.4. The output from the actuator model will be fed into the LV model in Eq. (4.29). The dynamics of the TVC actuator model has been obtained from [35]. For a detailed description of this model, the reader is referred to consult to that paper. The model can be described as an upper LFT model to include uncertainties: $\mathcal{F}_u(G_{\text{TVC}}(s), \Delta_{\text{TVC}})$, where $\Delta_{\text{TVC}} \in \Delta_{\text{TVC}}$. Theory of LFT will be evaluated later in Section 5.1. The bode plot of the TVC actuator model is given in Fig. 4.6.

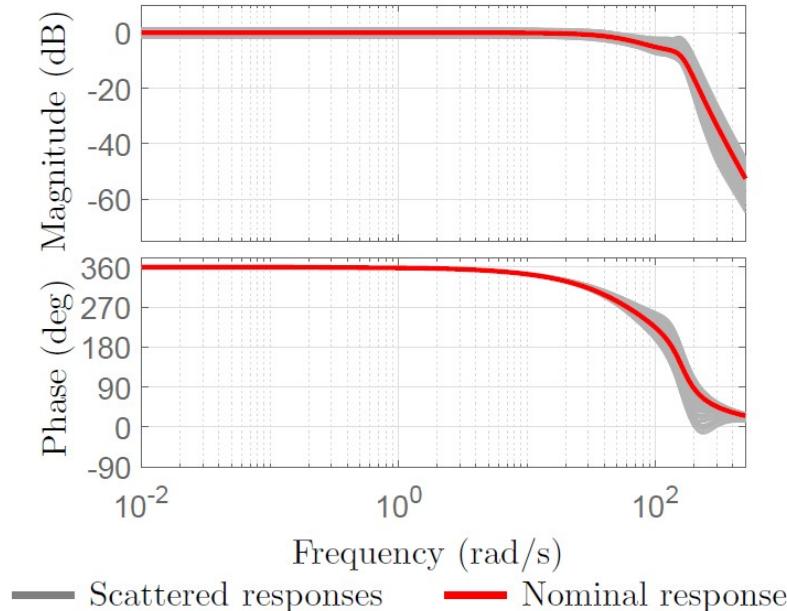


Figure 4.6: Bodeplot of LFT TVC actuator model ($\beta_\theta(s)/\beta_\tau(s)$) [4]

Fig. 4.6 shows the nominal response and scattered responses. However, back in section 2.6.3 it was decided not to model the uncertainties. In this report there will only be focus on the nominal (red) response.

4.4 Delay Model

The delay model models the characteristics of the different delays there are in the system between controller and actuator. In the different delays originate from TVC actuators, sensors, and digital processing from computers on board of the LV. From [4], the delays from these systems are 15 ms, SI12ms and 12 ms respectively. This delay model can also be represented as an upper LFT model, with uncertainties on the delay times. However as before, this has not been considered. The delay model is modelled with a second order Padé approximation and can be seen in Eq. (4.39).

$$G_\tau(s) = \frac{\beta_\tau(s)}{\beta_c(s)} = \frac{0.002s^2 - 0.234s + 12}{0.002s^2 + 0.234s + 12} \quad (4.39)$$

The bode plot from the delay model can be seen in Fig. 4.7.

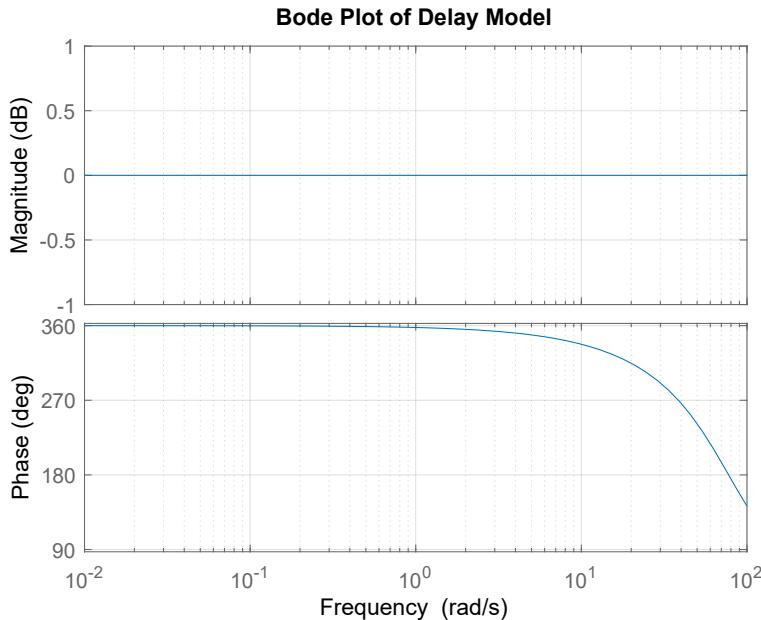


Figure 4.7: Bodeplot of delay model of the LV system ($\beta_\tau(s)/\beta_c(s)$)

The bode plot shows a frequency range of from 0.01 rad/Hz to 100 rad/Hz, which is what launcher systems generally covers. Frequencies above 100 rad/Hz are not in interest, hence the approximation is only a good approximation until that point [4].

4.5 Conclusion

This chapter presented a linearization of the LV and has been transformed into state space form with disturbance. Moreover, the TVC actuator model of the system along the

actuator model has been evaluated. These models will be used in high-fidelity simulation environment and for analysis and design of a controller.

Control System and Controller Models

5

The robust control framework provides a set of techniques that allow to manage issues such as specification of MIMO requirements in systematic manner and impact of uncertainties. Recall back in subsection 2.6.3, it was decided not to include uncertainties when designing a controller but was something that can be done in future work. However, the theory of how to model uncertainties will be explained. The first step in the robust control framework is modelling. Here the objective is to capture possible uncertainties in a methodical manner. This is e.g. a way to model the way parameters are varying in Linear Parameter Varying (LPV) systems. After the modelling has been done, the control synthesis comes after. During this phase there can either a standard \mathcal{H}_∞ - , a structured \mathcal{H}_∞ - or LPV-synthesis can be done [16]. After the synthesis, an analysis whether the designed controller is working as intended should be done. An acceptable controller is usually a result of an iterative cycle between the control synthesis and analysis, to ensure that the design requirements are fulfilled.

This chapter will first introduce the LFT representation in Section 5.1, then the synthesis of the \mathcal{H}_∞ is explained in Section 5.2, next how the LV system will be implemented together and how with an augmented configuration in subsection 5.2.1. Then an augmented \mathcal{H}_∞ system will be evaluated for where weights on the input and out of the LFT framework will be selected. Lastly, an LPV synthesis will be made along with an analysis of the parameters which affect the LV system the most in Section 5.5 and Section 5.6 respectively.

5.1 LFT Framework

Typically, in a robust control context, a mathematical representation known as Linear Fractional Transformation (LFT) is employed. The LFT formulation has been chosen because it is a well-established and suitable approach to model the unknowns of a system. In that way, the uncertainties can convenient be implemented in the model in future work.

The LFT representation is defined by a feedback interconnection between two matrices, $P \in \mathbb{C}^{(n_d+n_u) \times (n_e+n_y)}$ and $\Delta \in \mathbb{C}^{n_y \times n_u}$, where P is traditionally partitioned into four submatrices [4]:

$$\begin{bmatrix} e \\ y \end{bmatrix} = P(s) \begin{bmatrix} d \\ u \end{bmatrix} = \begin{bmatrix} P_{11}(s) & P_{12}(s) \\ P_{21}(s) & P_{22}(s) \end{bmatrix} \begin{bmatrix} d \\ u \end{bmatrix} \quad (5.1)$$

where:

- | | |
|-----|--|
| u | is the control variables. |
| y | is the measured variables. |
| d | is the exogenous signals such as disturbances d and commands r . |
| e | is the so-called error signals which are to be minimized in some sense to meet the control objectives. |

There are two types of LFT interconnections. Both are showed in Fig. 5.1, where (A) is showing the upper interconnection and (B) showing the lower interconnection. Notice that that the 's' has been omitted for better readability.

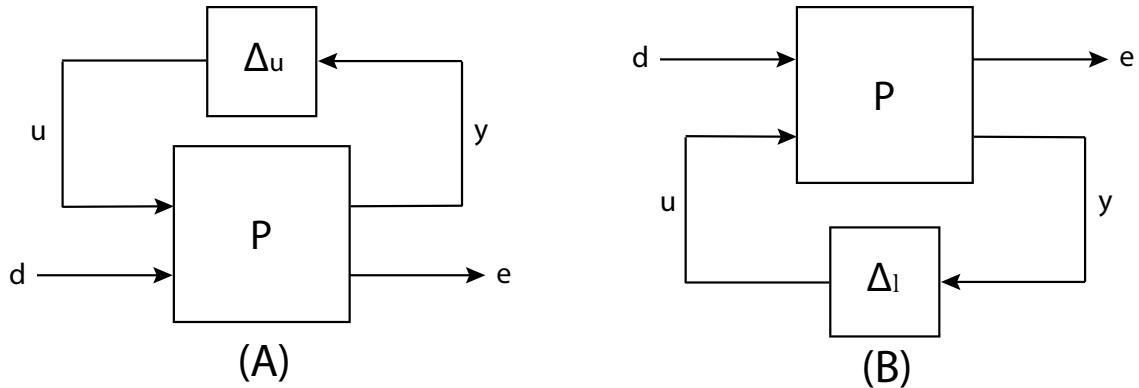


Figure 5.1: (A) showing the upper LFT interconnection and (B) showing the lower LFT interconnection

The upper LFT describes the relation between P and Δ_u . The operator \mathcal{F}_u defines the closed loop from the input signal d to the output signal e as follows:

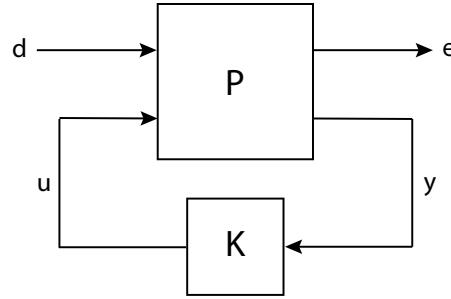
$$\mathcal{F}_u(P, \Delta_u) = P_{22} + P_{21}\Delta_u(I - P_{11})^{-1}P_{12} \quad (5.2)$$

where, P_{22} represents the nominal plant and P_{11} , P_{12} and P_{21} describe how the nominal plant is affected by the perturbation of Δ_u . Therefore, this representation can be used to model plants subject to uncertainties [4] and will be used for future work when implementing the uncertainties. As with the upper LFT, the lower LFT describes the relation between P and Δ_l . The interconnection here is defined by the operator \mathcal{F}_l and the relation is shown in Eq. (5.3).

$$\mathcal{F}_l(P, \Delta_l) = P_{11} + P_{12}\Delta_l(I - P_{22})^{-1}P_{21} \quad (5.3)$$

5.2 \mathcal{H}_∞ Synthesis

For the robust control synthesis, the objective is to find a linear controller $K(s)$, such that all the desired requirements of a MIMO plant $P(s)$ are fulfilled. The \mathcal{H}_∞ control problem is traditionally formulated using the lower LFT configuration shown in Fig. 5.1B, this time with the controller in the feedback loop.

**Figure 5.2: General control configuration**

where the closed loop of the system is represented by Eq. (5.4) [4]:

$$e = \mathcal{F}_l(P, K)d \quad (5.4)$$

For the generalized plant P is partitioned as in Eq. (5.5).

$$\begin{bmatrix} \dot{x} \\ e \\ y \end{bmatrix} = \underbrace{\begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{bmatrix}}_{P(s)} \begin{bmatrix} x \\ d \\ u \end{bmatrix} \quad (5.5)$$

where x is the state vector of P .

The following general assumptions for \mathcal{H}_∞ synthesis control problems are typically made:

1. (A, B_2, C_2) are stabilizable and detectable.
2. D_{12} and D_{21} have full rank.
3. $\begin{bmatrix} A - j\omega I & B_2 \\ C_1 & D_{12} \end{bmatrix}$ has full column rank for all ω .
4. $\begin{bmatrix} A - j\omega I & B_1 \\ C_2 & D_{21} \end{bmatrix}$ has full row rank for all ω .

where:

1. ensures existence of stabilizing controllers K .
2. is sufficient to ensure the controllers are proper and hence realizable.
3. and 4. ensure that the optimal controller does not try to cancel poles or zeros on the imaginary axis which will result in closed loop instability [36].

The control problem then becomes finding a controller $K(s)$ that is stabilizing the plant $P(s)$ and minimizes the \mathcal{H}_∞ norm of the closed loop system:

$$\min_{K(s)} \|\mathcal{F}_l(P, K)\|_\infty = \min_{K(s)} \max_{\omega \in \mathbb{R}} \bar{\sigma}(\mathcal{F}_l(P, K)(s)) \quad (5.6)$$

where the maximum singular value of P is defined as $\bar{\sigma}(P(s))$ [4]. Note that in practice it is considered too computationally demanding to find an optimal solution to the \mathcal{H}_∞ synthesis problem. Instead, it is simpler to design a sub-optimal controller. Let γ_{min}

be the minimum value of $\|\mathcal{F}_l(P, K)\|_\infty$ over all stabilizing controllers K . Then the \mathcal{H}_∞ sub-optimal control problem is given by finding a stabilizing controller K such that:

$$\|\mathcal{F}_l(P, K)\|_\infty < \gamma \quad (5.7)$$

where $\gamma > \gamma_{min}$ is given [36]. For achieving a good controller, one should strive for designing γ as small as possible. This can be done through an iterative process until an optimal solution is approached.

For the general control configuration shown in Fig. 5.2 and with the general assumptions mentioned above, there exists a controller $K(s)$ such that $\|\mathcal{F}_l(P, K)\|_\infty < \gamma$ if and only if:

(i) $X_\infty \geq 0$ is a solution to the Riccati equation:

$$A^T X_\infty + X_\infty A + C_1^T C_1 + X_\infty \left(\gamma^{-2} B_1 B_1^T - B_2 B_2^T \right) X_\infty = 0 \quad (5.8)$$

such that $\text{Re } \lambda_i \left[A + \left(\gamma^{-2} B_1 B_1^T - B_2 B_2^T \right) X_\infty \right] < 0, \forall i$.

(ii) $Y_\infty \geq 0$ is a solution to the Riccati equation:

$$A Y_\infty + Y_\infty A^T + B_1 B_1^T + Y_\infty \left(\gamma^{-2} C_1^T C_1 - C_2^T C_2 \right) Y_\infty = 0 \quad (5.9)$$

such that $\text{Re } \lambda_i \left[A + Y_\infty \left(\gamma^{-2} C_1^T C_1 - C_2^T C_2 \right) \right] < 0, \forall i$.

(iii) $\rho(X_\infty Y_\infty) < \gamma^2$

The general controller is then given as [36]:

$$K(s) = -Z_\infty L_\infty (sI - A_\infty)^{-1} F_\infty \quad (5.10)$$

where:

$$\begin{aligned} F_\infty &= -B_2^\top X_\infty \\ L_\infty &= -Y_\infty C_2^\top \\ Z_\infty &= \left(I - \frac{1}{\gamma^2} Y_\infty X_\infty \right)^{-1} \\ A_\infty &= A + \frac{1}{\gamma^2} B_1 B_1^\top X_\infty + B_2 F_\infty + Z_\infty L_\infty C_2 \end{aligned}$$

The lower LFT interconnection of Fig. 5.2 can be augmented by taking different weighted transfer functions into account via the construction of P . These weights are implemented on the input side $W_i(s)$ and output side $W_o(s)$ of the lower LFT interconnection. In Fig. 5.3 this configuration can be seen.

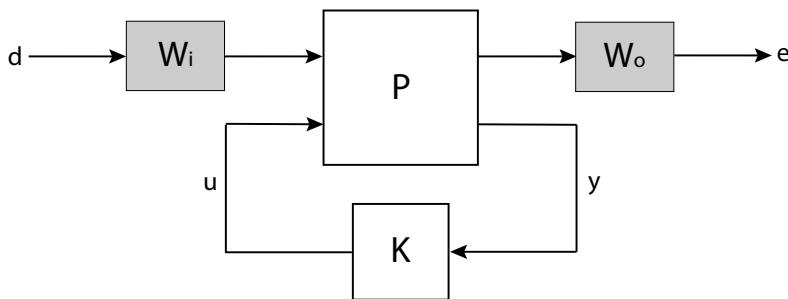


Figure 5.3: Augmented control interconnection

The weighting function are selected to impose the desired closed loops performance of the system in the frequency domain and to normalize the control problem. This means that if there exists a stabilizable controller that fulfills all the requirements given for the problem, then the infinity norm of the augmented closed loop system will be $\|\mathcal{F}_l(P, K)\|_\infty < 1$, where the total augmented system is given in Eq. (5.11) [4].

$$e = W_o \|\mathcal{F}_l(P, K)\|_\infty W_i d \quad (5.11)$$

5.2.1 \mathcal{H}_∞ Design

The closed loop diagram used for design is presented in Fig. 5.4.

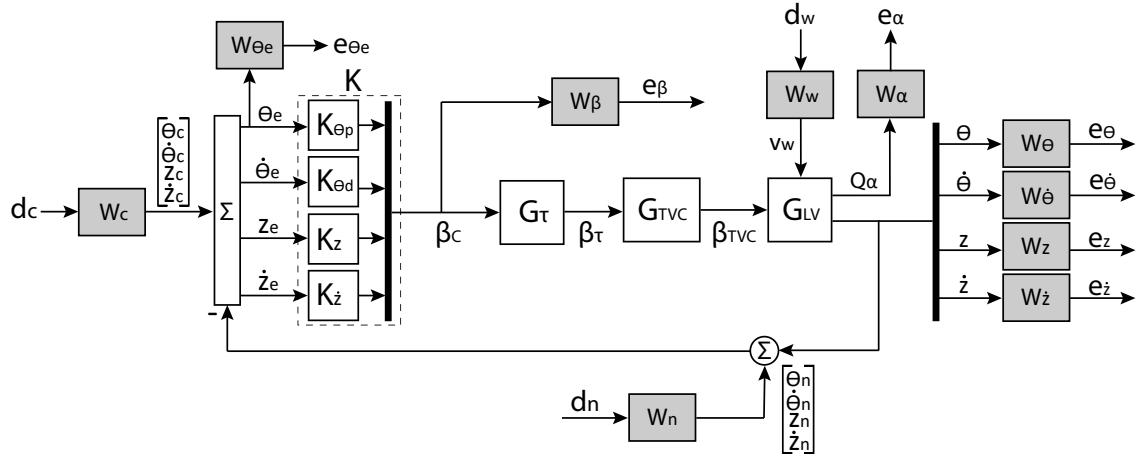


Figure 5.4: Closed loop diagram for the LV-system

This closed loop diagram consists of four main blocks: The controller K , the total delay of the system, G_τ , the actuator dynamics G_{TVC} and the launch vehicle dynamics, G_{LV} . All the dynamic blocks have been described back in subsection 4.1.4, Section 4.3 and Section 4.4, and are all nominal, meaning no uncertainties will be evaluated for. The controller block, K , is composed of four gains, which can be tuned, namely $K_{\theta,p}$, $K_{\theta,d}$, K_z and $K_{\dot{z}}$, which each represent the states that we want to control. The different weights are chosen with the technical requirements in mind, which were described back in subsection 3.1.2. There have been placed 10 weights at different locations in the closed loop diagram. These weights include the command signal, W_c , which describes how the states should behave. Moreover, a weight has been placed on the input disturbance, W_w , which is the wind acting on the LV. The last weight placed at an input to the system is W_n , which describes the impact of the noise onto the sensors. Recall back to subsection 2.6.3, how it was decided to implement sensor noise on the feedback on the states that are going to be controlled. From subsection 3.1.2 a stability requirement was given on the GM and PM. This will be incorporated in the $W_{\dot{\theta}}$. Furthermore, from the requirement specification, a requirement was set on the actuation, therefore W_β has been included in the closed loop diagram as well for the dynamic pressure that the LV can be exposed

to. The final four weights, W_θ , $W_{\dot{\theta}}$, W_z and $W_{\dot{z}}$ will also be chosen directly from the performance requirements stated back in subsection 3.1.2.

Fig. 5.4 can be formulated as a robust control problem using augmented \mathcal{H}_∞ interconnection. This setup can be seen in Fig. 5.5. By this representation, the plant $P(s)$ and the design specifications represented by the input- and output weights, W_i and W_o respectively, have their dynamics gathered. Note that it has been chosen to partition the plant and W_i and W_o into a block M , such that the closed loop system is given by:

$$e = \mathcal{F}_l(M, K)d \quad (5.12)$$

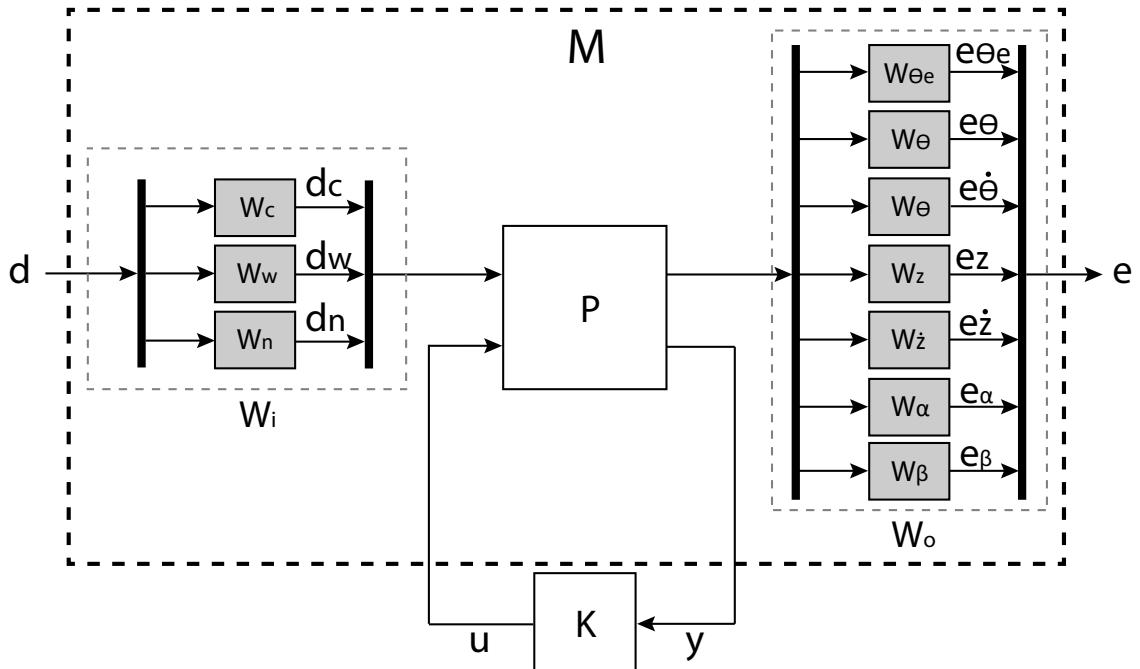


Figure 5.5: Augmented \mathcal{H}_∞ interconnection for the system

With this new representation, the plant, P have a set of input weights W_i in the form by commands, wind disturbance and sensor noise. The input is partitioned in vector form: $d = [d_c \ d_w \ d_n]^T$. Likewise, a set of outputs from the plant has also been partitioned: $e = [e_{\theta,e} \ e_\theta \ e_z \ e_{\dot{z}} \ e_\alpha \ e_\beta]^T$. The input weighting functions are tuned to scale the closed loop dynamics w.r.t. their expected maximum variations. For example, for disturbances d should be scaled as shown in Eq. (5.13).

$$d = \frac{\hat{d}}{\hat{d}_{\max}} \quad (5.13)$$

where \hat{d} is the unscaled variable of disturbance and \hat{d}_{\max} is the largest expected change in disturbance [36]. while the output weighting functions are tuned to specify the closed loop performance. The input- and output weights are partitioned as:

$$W_i = \begin{bmatrix} W_c & 0 & 0 \\ 0 & W_w & 0 \\ 0 & 0 & W_n \end{bmatrix} \quad (5.14)$$

$$W_o = \begin{bmatrix} W_{\theta,e} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & W_\theta & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & W_{\dot{\theta}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & W_z & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & W_{\dot{z}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & W_\alpha & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & W_\beta \end{bmatrix} \quad (5.15)$$

Note also that the controller K have been pulled out of the generalized plant. By doing this, a scalar $u = \beta_c$ is representing the controller output and the output from the plant is represented by $y = [\theta_e \dot{\theta}_e z_e \dot{z}_e]^T$.

Finally, the structured \mathcal{H}_∞ optimization will be used. What this does compared to the standard \mathcal{H}_∞ optimization, is that the order of the controller K can be chosen freely and the gains which are to control the closed loop system can be chosen. In this project there has been chosen gains of each state as indicated in Fig. 5.4 and the order of each gain is 0.

5.3 Weighting Function Selection

In this section each of the mentioned weights will be designed up against the requirements from subsection 3.1.2. Generally, the design is an iterative process, where the weights will be tuned until an adequate stability and performance is achieved. Thus, initial conditions for the weights will be described, but their parameters can be changed under the tuning process. During the designing process, the order of the weighting functions is kept low for simplicity and ease of tuning. It is also important to use the same units on both sides of the system. This is done to improve the conditioning of the design problem for the scaled variables [36] [37]. In this design it has been chosen to have degrees for angles on both sides as this is how the requirements have been defined. While internally the system is using radians, thus the conversion is used for the weights related to angles.

5.3.1 Input Weight Selection

Starting with the command input weight, there have been put a weight on each signal, θ_c , $\dot{\theta}_c$, z_c and \dot{z} . These are each partitioned like shown in Eq. (5.16).

$$W_c = \begin{bmatrix} \frac{\pi}{180} W_{\theta_c} & 0 & 0 & 0 \\ 0 & \frac{\pi}{180} W_{\dot{\theta}_c} & 0 & 0 \\ 0 & 0 & W_{z_c} & 0 \\ 0 & 0 & 0 & W_{\dot{z}} \end{bmatrix} \quad (5.16)$$

It has been decided to set the expected maximum attitude angle command to 1 degree. W_{θ_c} will then be tuned accordingly to balance the attitude and attitude rate channels to

a point where they have the same level of magnitude. In [4], the way to balance this was to take ratio between the controller gains $K_{\theta,p}$ and $K_{\theta,d}$:

$$W_{\dot{\theta}c} = \frac{K_{\theta,p}}{K_{\theta,d}} \quad (5.17)$$

To help finding $K_{\theta,p}$ and $K_{\theta,d}$, analyzing the sensitivity function, $S_\theta(s)$ and the complementary sensitivity function $T_\theta(s)$ of the pitch is a good start for an initial guess. These are important when studying the robustness and performance of the system as they can give details of how the system behaves to disturbances. The sensitivity function represents the transfer function between the error of the attitude command θ_e and the output θ , while the complementary sensitivity function represents the transfer function between the attitude command θ_c and the output θ . The relationship between them is: $S_\theta(s) + T_\theta(s) = 1$. According to [4], from $S_\theta(s)$ and $T_\theta(s)$, the gains $K_{\theta,p}$ and $K_{\theta,d}$ can be derived from the relevant state space entries, see [4]. The derived gains can be seen below in Eq. (5.18) and Eq. (5.19).

$$K_{\theta,p} = \left(1 + \frac{\omega_n^2}{\mu_\alpha}\right) \frac{\mu_\alpha}{\mu_{PVP}} \quad (5.18)$$

$$K_{\theta,d} = \frac{2\zeta\omega_n}{\mu_{PVP}} \quad (5.19)$$

where μ_α and μ_{PVP} are the load proneness of the angle of attack and TVC respectively. These proneness coefficients were derived from the state space representation back in subsection 4.1.4. As for the dampening of the system, $\zeta = 0.7$ has been chosen since this value generally gives a good rise time of a system, while keeping the overshoot at a minimum. The natural frequency ω_n has been set to 30 Hz which is the frequency response from the TVC actuator designed back in Section 4.3.

W_z will be fixed to 1, while W_z is balancing the drift- and drift rate channel[4]. The wind disturbance weight W_w is scaled according to the expected maximum wind velocity, which is going to be 3, which is mentioned back in Chapter 3. However, normally the disturbance is a low frequency signal it will be rejected if that the maximum singular value is made small over the same frequencies that of the disturbance. Therefore, a low-pass filter would have been ideal for disturbance rejection [36], but for simplicity the weight will be kept constant across all frequencies. As for the sensor noise, the noise on the pitch is set to 0.02°, the pitch rate is set to 0.1 °/s, the noise on the drift is set to 0.01 m and lastly the noise on the drift rate is set to 0.001 m/s [37].

The final W_i can be seen in Eq. (5.20).

$$W_i = \text{diag} \left(\frac{\pi}{180}, \frac{\pi}{180} 7, 15, 1, 3, \frac{\pi}{180} 0.02, \frac{\pi}{180} 0.1, 0.01, 0.001 \right) \quad (5.20)$$

5.3.2 Output Weight Selection

The output weights are selected for designing a controller with adequate performance w.r.t. the requirements stated back in subsection 3.1.2. While the input weight is simply

a scaling of the normalized input to the expected input. The output is a weight that when the maximum desired value is put through the weight it should be normalized to 1. Thus, to get the output weight, it will be the inverse of the maximum desired value of that signal.

W_{θ_e} and W_θ enforce the stability and tracking requirements. To design these weights, the gain of the sensitivity function is used. The maximum peak of the infinity norm of the sensitivity function is a way to analyze closed-loop robustness. $\|S_\theta(s)\|_\infty$ yields a lower bound for the classical stability gain margin and phase margin [36]. These relations can be seen in Eq. (5.21) and Eq. (5.22) respectively.

$$GM \geq \frac{\|S_\psi(s)\|_\infty}{\|S_\psi(s)\|_\infty - 1} \quad (5.21)$$

$$PM \geq 2 \arcsin \left(\frac{1}{2 \|S_\psi(s)\|_\infty} \right) \quad (5.22)$$

Moreover, the inverse of W_{θ_e} and W_θ impose an upper bound on the classical sensitivity and complementary sensitivity functions of the attitude channel. So, to ensure good stability margins a constant weight on the tracking error W_{θ_e} dependent on the sensitivity function is applied as [4]:

$$W_{\theta_e} = \left(\frac{\pi}{180} \|S_\theta(s)\|_\infty \right)^{-1} \quad (5.23)$$

The tracking error weight W_{θ_e} is then designed to impose an upper bound on 10 dB on the sensitivity function $\|S_\theta(s)\|_\infty$:

$$W_{\theta_e} = \left(\frac{\pi}{180} 3.16 \right)^{-1} \quad (5.24)$$

The inverse of W_θ is then shaped like a low-pass filter to limit the complementary sensitivity function $T_\theta(s)$, and is given by:

$$W_\theta(s) = \left(\frac{\pi}{180} \frac{h_\theta s + \omega_\theta}{s + \frac{\omega_\theta}{l_\theta}} \right)^{-1} \quad (5.25)$$

where, h_θ and l_θ are the high- and low asymptotes of the low-pass filter respectively and ω_θ is the bandwidth of the filter. In [4] the l_θ has been set to the value of $\|S_\theta(s)\|_\infty = 10$ dB to reduce the number of weight parameters that needs to be tuned later in the optimization process. The h_θ is set to a gain of -40 dB to reduce noise contribution. Finally, the crossover frequency is set to 10 rad/s in order to limit the tracking bandwidth.

$$W_\theta(s) = \left(\frac{\pi}{180} \frac{0.01s + 10}{s + \frac{10}{3.16}} \right)^{-1} \quad (5.26)$$

The inverse of the rate of pitch weight $W_{\dot{\theta}}$ put to 0, meaning there will be no upper limit for the pitch rate. W_z and $W_{\dot{z}}$ represent the lateral control requirements. Their respective inverses correspond to their requirements from subsection 3.1.2, namely their maximum expected drift and drift rate. Thus, making them both be defined as constants functions.

$$W_z = (z_{max})^{-1} \quad (5.27)$$

$$W_{\dot{z}} = (\dot{z}_{max})^{-1} \quad (5.28)$$

However, it should be noted that lowering the allowed drift and drift rate deviations will reduce the wind disturbance effect on the drift-rate channel and, in turn, relieve the structural load, Q_α , which is impacted by the lateral drift-rate. The trade-off by restricting the allowed drift and drift rate deviations will however restrict the tracking control, since that allows for lateral deviations. Nonetheless the inverse weights will correspond to their respective requirements:

$$W_z = (500)^{-1} \quad (5.29)$$

$$W_{\dot{z}} = (15)^{-1} \quad (5.30)$$

The inverse load requirement W_{Q_α} will be bounded through the maximum angle of attack α_{max} . Recall back in 3.1.2 that $Q_\alpha = Q_d \alpha$, where Q_d is a predetermined dynamic pressure calculated for the specific flight. W_{Q_α} will be fixed throughout the whole atmospheric flight phase, and the maximum angle of attack is set to be 3° . This is done because the actual maximum angle of attack cannot be determined before after the trajectory have been tested or simulated as mentioned in 3.1.2.

$$W_{Q_\alpha} = \left(\frac{\pi}{180} Q_d \alpha_{max} \right)^{-1} = \left(\frac{\pi}{180} 3Q_d \right)^{-1} \quad (5.31)$$

The last output weight to consider is the actuator weight W_β , which the inverse of that refers to the maximum actuator deflection, which from subsection 3.1.2 is 6.5° [37].

$$W_\beta = \left(\frac{\pi}{180} 6.5 \right)^{-1} \quad (5.32)$$

All of the mentioned output weights are put into $W_o(s)$ as shown:

$$W_o(s) = diag \left(\left(\frac{\pi}{180} 3.16 \right)^{-1}, \left(\frac{\pi}{180} \frac{h_\theta s + \omega_\theta}{s + \frac{\omega_\theta}{l_\theta}} \right)^{-1}, 0, (500)^{-1}, (15)^{-1}, \left(\frac{\pi}{180} 3Q_d \right)^{-1}, \left(\frac{\pi}{180} 6.5 \right)^{-1} \right) \quad (5.33)$$

Now that the input weights and the output weights have been designed, the \mathcal{H}_∞ controller will be evaluated later in Section 7.1. Next the LPV modelling will be explained.

5.4 LPV Modelling

This section is going to present the LPV framework and synthesis. LPV systems is a way to estimate nonlinear systems using a multiple linear system by embedding nonlinear terms such as changing the trajectory in the time varying parameter. As mentioned back in Section 2.6, the LPV framework can be considered as an augmentation of the \mathcal{H}_∞ framework. This means the same interconnection is used as the \mathcal{H}_∞ approach with the same number of weights. As a starting condition, the same weights from Section 5.3 is used to as weight selection.

5.4.1 LPV Framework

LPV is represented as a state-space system, however the linear matrices are continuous functions of a time varying vector $\rho(t)$. The general LPV system can be described by Eq. (5.34).

$$\begin{bmatrix} \dot{\mathbf{x}}(t) \\ \mathbf{y}(t) \end{bmatrix} = \begin{bmatrix} A(\rho(t)) & B(\rho(t)) \\ C(\rho(t)) & D\rho(t) \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{bmatrix}, \quad \rho \in \mathcal{P}, \quad \underline{\nu} \leq \dot{\rho} \leq \bar{\nu} \quad (5.34)$$

here $x(t)$ is the state vector, $y(t)$ is the output vector, and $u(t)$ is the input vector. $\mathcal{P} \subset \mathbb{R}^{n_\rho}$ is a known compact set which ρ belongs to. A, B, C, and D are the continuous state-space matrices.

The time-varying vector $\rho(t)$ is defined as multiple parameters of the desired system that are changing and are unknown but measurable in real time (causal). The vector $\rho(t)$ may also have bounds on rate of variation depending on the stability of the LPV. This bound is defined by the decreasing rate $\underline{\nu}$ and the increasing rate $\bar{\nu}$.

There are several modelling approaches to choose from, e.g. a grid-based LPV model, which is obtained from different operating points chosen from the changing parameters values. An LFT-based modelling method that captures the time-varying behavior of the given system. Lastly there is a polytopic modelling approach, that is making the state space matrices that depends on the scheduling parameter ρ affine [4]. In this report, the grid-based modelling method will be used.

5.5 LPV synthesis

As mentioned before with the structured \mathcal{H}_∞ problem, the LPV design approach also uses the same interconnection in the form of a lower LFT representation. This time the plant P and controller is K is dependent on ρ , as indicated in Fig. 5.6.

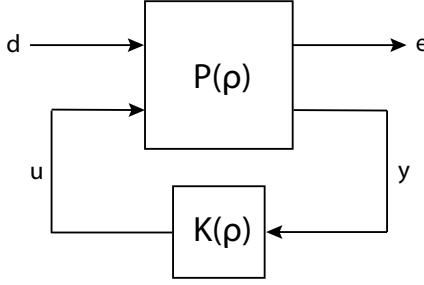


Figure 5.6: LPV control problem formulation

Likewise, the plant $P(\rho)$ is partitioned as shown below:

$$\begin{bmatrix} \dot{x} \\ e \\ y \end{bmatrix} = \underbrace{\begin{bmatrix} A(\rho) & B_1(\rho) & B_2(\rho) \\ C_1(\rho) & D_{11}(\rho) & D_{12}(\rho) \\ C_2(\rho) & D_{21}(\rho) & D_{22}(\rho) \end{bmatrix}}_{P(\rho)} \begin{bmatrix} x \\ d \\ u \end{bmatrix} \quad (5.35)$$

where $x \in \mathbb{R}^{n_x}$, $u \in \mathbb{R}^{n_y}$, $d \in \mathbb{R}^{n_d}$, $e \in \mathbb{R}^{n_e}$ and the state space are in their appropriate dimensions [4]. The LPV synthesis problem then becomes finding a stabilizing controller $K(\rho)$ which minimizes the induced \mathcal{L}_2 norm of the cost function stated in Eq. (5.36).

$$\min_{K(\rho)} \left\| \mathcal{F}_l(P(\rho), K(\rho)) \right\|_{\mathcal{L}_2 \rightarrow \mathcal{L}_2}; \quad \text{subject to } \begin{array}{c} \rho \in \mathcal{P} \\ \underline{\nu} \leq \dot{\rho} \leq \bar{\nu} \end{array} \quad (5.36)$$

As it can be seen, the Eq. (5.36) is compared to Eq. (5.6) a generalization of the \mathcal{H}_∞ norm for LPV system.

The control synthesis can be described as an LMI problem instead of Eq. (5.36). If the rate of change of the parameters are unbounded meaning the decreasing rate $\underline{\nu}$ and the increasing rate $\bar{\nu}$ are $-\infty$ and ∞ respectively, the synthesis can be performed by a single quadratic Lyapunov function [4].

When considering the control LPV system in Eq. (5.35) with $x(0) = x_0$ it is quadratically stabilizable by a state feedback $u = K(\rho)x \Leftrightarrow \exists X \in S_{>0}^n$ and a function $Y : \Delta_p \rightarrow \mathbb{R}^{m \times n}$ such that the LMI:

$$\begin{bmatrix} He(A(\rho)X + B(\rho)Y(\rho)) & E(\rho) & (C(\rho)X + D(\rho)Y(\rho))^T \\ \star & -\gamma I & F(\rho)^T \\ \star & \star & -\gamma I \end{bmatrix} < 0 \quad (5.37)$$

holds for all $\rho \in \Delta_\rho$ [38]. Where in the above matrix $He(M) = M + M^T$ and \star means the transposed of its symmetric counterpart. From this notation the state feedback is given by:

$$u = \underbrace{Y(\rho)X^{-1}}_{K(\rho)} x \quad (5.38)$$

and performance is given by [38]:

$$|z|_2 \leq \gamma |w|_2 + \sqrt{\gamma x_0^T X^{-1} x_0} \quad \forall (w \in L_2, \rho \in \mathcal{P}) \quad (5.39)$$

As for the augmented lower LFT interconnection for the LPV system, the setup will be the same as in Section 5.2:

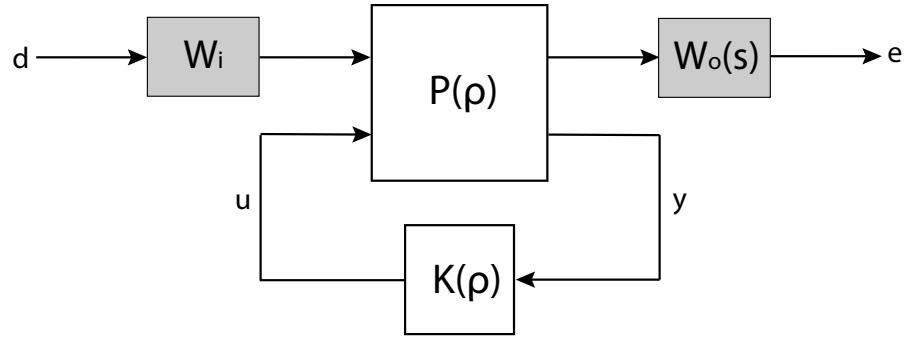


Figure 5.7: Augmented LPV control configuration

Thus, making the augmented closed loop system:

$$e = W_o(s) \|\mathcal{F}_l(P(\rho), K(\rho))\|_\infty W_i d \quad (5.40)$$

Where W_i and $W_o(s)$ are given in Eq. (5.20) and Eq. (5.33) respectively.

Now that the synthesis and weight selection for the augmented LFT model for LPV has been accounted for, choosing which parameters from Eq. (4.29) to grid is the next step in design of an LPV controller. This will be done in the next section.

5.6 Analysis of Varying Parameters

There are many parameters that vary during the flight of the LV. To take all of them into account when designing the controller would add a lot of complexity to the problem and increase the compute time exponentially. Thus, if less parameters can be considered it would greatly simplify the design. Therefore, this analysis will be done to investigate which variation of parameters have the biggest effect on the state-space model. Then the most impact full parameters can be used. Some parameters also vary similarly or dependent on each other and can be expressed as a function of a single variable which could further reduce the complexity of the optimization problem. The parameters we will be looking at and their extreme values is listed in Table 5.2. The values are taken from a simulation with a controller that was stable but hadn't been optimized. Thus, might not be exactly what will be final values with an optimized controller. However, it is considered to be close enough to give an idea of the magnitude of the variation of the parameters.

Parameter	Min value	Max Value	Unit
Velocity	1	1900	m/s
Air density	5.58e-5	1.2	kg/m ³
Inertia	3.155e+6	4.658e+6	kg/m ²
Center of Gravity	8.508	14.05	m
Center of Pressure	28	30	m
Gravity	9.61	9.78	m/s ²
Mach number	0	5.7	—
Pitch	26	90	°

Table 5.1: List of varying parameters with their respective minimum value and maximum value

These values were then used in all possible combinations to make every extreme state space matrices, specifically the system matrices. The main way the data was interpreted was by looking at the singular values of all the matrices.

5.6.1 Change in Singular Values

The singular values were chosen as they give an overview of the gain of signals through the system. Initially, the most interesting singular values are the largest ones as they affect the dynamics in the system the most. However, as it will be discussed later, this might not be true as some of the states have different units and different importance. Nonetheless, it was considered a good way to get a sense of the effect of the different parameters had on the system. It should be noted that it is by no means a robust analysis with clear answers, but simply a somewhat systematic way to look at the effect of the parameter variations. In Fig. 5.8 the relative change in singular values are compared. The comparison was done for each combination of parameters where only one parameter is changed one at a time from their respective minimum value to the maximum value. What is plotted is how many times larger the larger $\bar{\sigma}$ is compared to the smaller $\bar{\sigma}$ of the two. This is also done for the 2nd, 3rd, and 4th σ . By doing this, the largest possible effect a parameter has on the LV system is found.

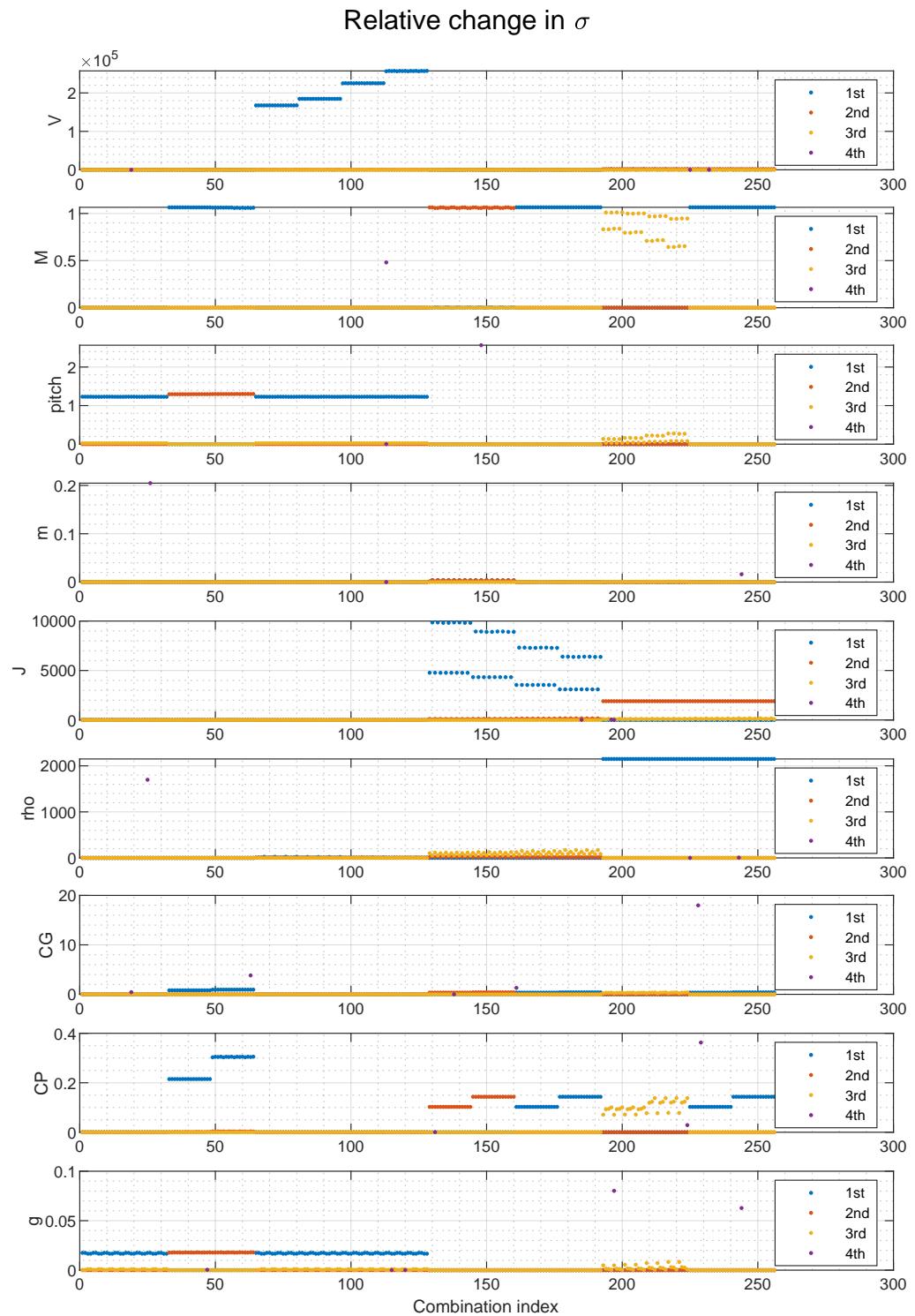


Figure 5.8: Relative change in singular values, when changing one variable between smallest and largest value

The main thing that was investigated, was the maximum variation from changing the parameter. Notice how the scale of the y-axis on of the graphs vary for each parameter. This gives the impression of the order of magnitude each parameter affects the system. Meanwhile, the x-axis has no apparent interpretation as it is just an index.

The next thing to take note of, is the frequency of which the parameter change has a visible effect. As an example, the CG has only one value that seems to cause the high scale on the y-axis. Moreover, the largest singular values, marked with blue, should be inspected, as they are the ones with the biggest impact on the dynamics.

A last thing to keep in mind when looking at these plots is that some of the combinations will not actually happen at the same time. Most of the parameters consistently has their peaks and minimums at certain point of the trajectory and thus will not be combined in all the ways as Fig. 5.8 indicates. Furthermore, some parameters are directly depended on each other or vary in a similar manner, e.g. mass and inertia, or velocity and Mach number. All these combinations that could or would never happen during flight should be removed for a further analysis.

To conclude on these considerations, the apparent affect from the parameter variation has been listed from highest impact to lowest in Table 5.2.

Parameter	Maximum Relative effect on σ (without single outlier)	Nr
Velocity	250e3	1
Inertia	10e3	2
Air density	2e3	3
Center of Gravity	19 (1.5)	4
Pitch	2.5(1.3)	5
Mach number	1	6
Center of Pressure	0.35	7
Mass	0.2(0.01)	8
Gravity	0.1(0.02)	9

Table 5.2: Parameters which are affecting the LV system listed from most to least

Thus, based on this analysis, it can be concluded that the most important parameter variations to consider is the velocity, inertia, and air density. It should be noted that some parameters are interdependent such as inertia, mass and center of mass and will only attain a certain set of combinations.

5.6.2 Future Work on LPV Control

As stated in the previous section, the parameters that have the largest impact on the LV system are listed in Table 5.2. Therefore, it makes sense to design a controller with a grid based LPV synthesis by including the ranked parameters as they are ordered in the table one at a time. As one parameter is included, a tuning process will be done on the selected weights back in Section 5.5 until an adequate controller, which complies with the requirements stated in subsection 3.1.2, has been designed. Then the parameter with the

next largest impact will be implemented, and the tuning of the weights start again. This goes on until either all the parameters have been included in the LPV control problem or the effect the parameters have on the system are considered too insignificant.

5.7 Conclusion

Throughout this chapter, an overview of the full system is given in Fig. 5.4. The LFT framework has been introduced back in Section 5.1 and \mathcal{H}_∞ synthesis has been evaluated in Section 5.2. Thereafter a selection of input weights and outputs weight have been evaluated in Section 5.3. Finally, an LPV synthesis has been outlined as well as an analysis of the parameter affecting the system the most in Section 5.4 and

Implementation of the Simulation 6

This chapter deals with the implementation of the simulation along with how the controller is implemented into the simulation. The simulation has been implemented in Simulink. The Simulink environment allows the simulation structure to be split into subgroups with well-defined interfaces between each other. Each subgroup implemented in Simulink can be seen in the figure 6.1.

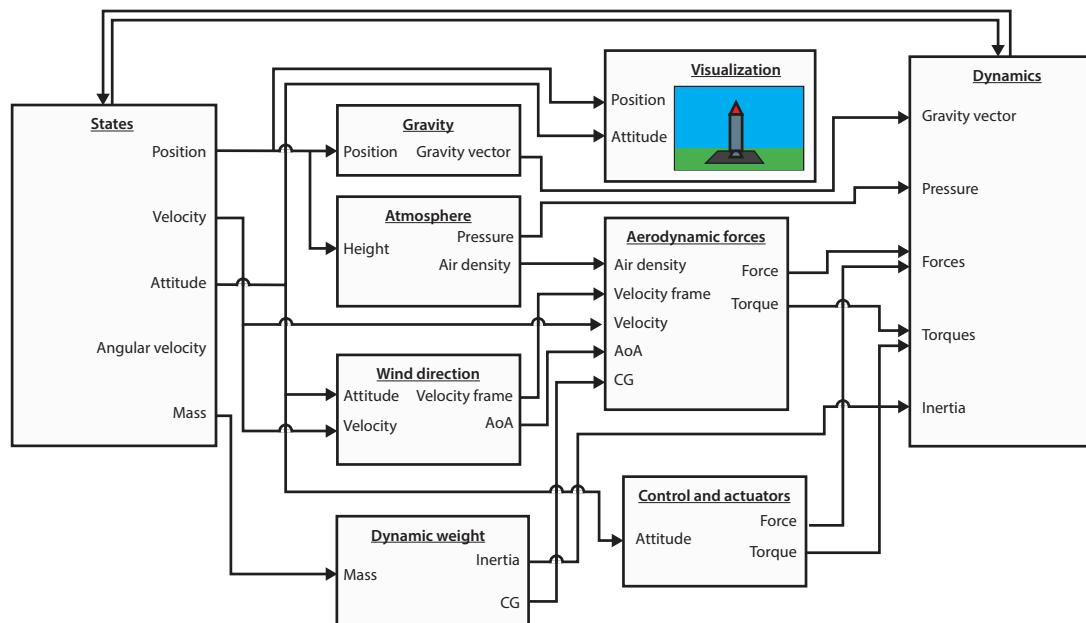


Figure 6.1: Showing the different subgroups in Simulink to create an overview

The first section 6.1 will go into details about the interface of each subgroup. The function or task of each of the subgroups will also be listed along with a flowchart showing the subgroup.

The second section 6.2 will go into details about how the simulation differs from the real world. This is done to clarify the weaknesses and the strengths of the simulation.

Additional can a table of all the initial dimensions of the LV be found in appendix A to give an overview of the parameters used in the simulation.

6.1 Implementation of Subgroups Inside the Simulation

The implementation is done by splitting the work up in subgroups. Each subgroup has a defined interface which will be specified in the associated subsection. The simulation contains the following subgroups: The Dynamics, the Air forces, the Control input, the States, the Center of gravity, the Gravity together with the Atmosphere, and the Animation. These are also seen in the figure 6.1.

6.1.1 States

The subgroup states contain the states sent from the dynamics. The main function of this subgroup is to separate the states and convert them to the correct format to match the interfaces of the other subgroups. All the states and how to find them can be found in chapter 4.

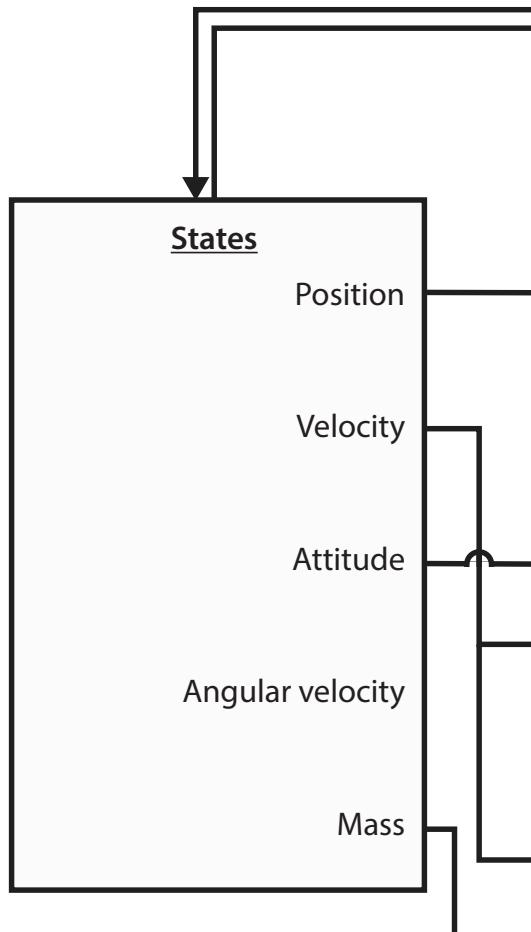


Figure 6.2: The subgroup States block diagram

The states are initiated in the Dynamic's subgroup (6.1.7) and sent to the State's subgroup. Here are the states separated and integrated to get the current values. The states are the mass, the position, the velocity, the angles, and the angle velocities. The position r is sent to the Gravity. Another part of the position is converted from ECI to ECEF and then again from ECEF to LLA for compatibility further on in the other subgroups. The LLA position

is sent to Gravity for calculation and the altitude is used in the Atmosphere. The velocity v is also converted from ECI to ECEF for compatibility and sent to the Atmosphere, the Air forces, and the Wind direction. The angles q is represented with quaternions and is converted to a rotation matrix. The rotation matrix is sent to the Air forces, and the Control. The rotation matrix is also converted again from ECI to ECEF and sent to Wind directions.

6.1.2 Atmosphere and Gravity

The two small subgroups called "Atmosphere" and "Gravity" are calculating the changes in the environment happening along the altitude. As the launch vehicle climbs the atmosphere will the static pressure from the atmosphere change alongside the gravity. For further explanation on the calculation of pressure look in section 2.4.

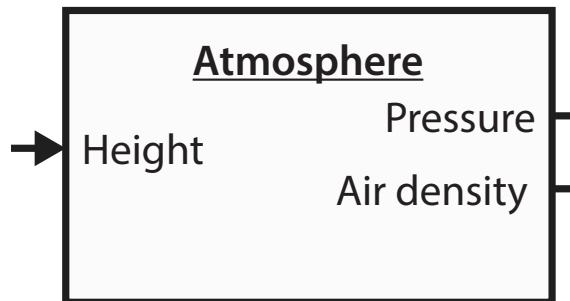


Figure 6.3: The subgroup Atmosphere block diagram

The implementation of the gravity is done with a function that takes in the position of the launch vehicle in LLA form. This is the latitude, longitude, and the altitude which is needed in order to find the given gravity in a lookup table. Along with the position in LLA does it also take in the position in ECI form. This is because the ECI also contains the orientation of the launch vehicle and can therefore set a vector for which direction the gravity is pulling. The gravity with the vector is then sent into the Dynamic's subgroup.

The function in charge of calculating the atmosphere takes in the altitude, and the velocity. With these two parameters it calculates the Mach number of the launch vehicle, the static pressure of the atmosphere, and the air density. The Mach number and the air density are used to calculate the air forces and the static pressure is sent to the dynamics.

6.1.3 Wind Direction

The input to the subgroup Wind direction is the velocity and the orientation of the launch vehicle. The subgroup is tasked with the calculation of the angle of attack, and the orientation in body frame. In subsection 2.2.3 are the aerodynamics explained in further details.

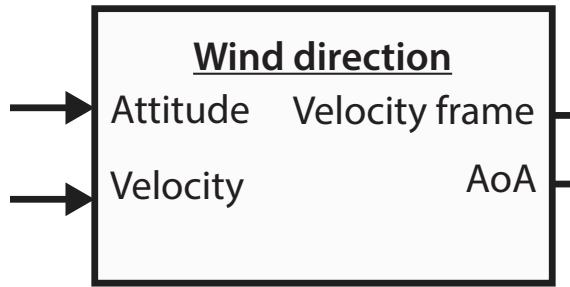


Figure 6.4: The subgroup Wind direction block diagram

Both outputs go into the Air forces subgroup. The angle of attack is used to calculate the aerodynamic forces happening on the launch vehicle, while the orientation in body frame is used to pinpoint the force vector and the direction of it on the launch vehicle.

6.1.4 Air Forces

The air forces such as drag, lift, and the center of pressure is found in the air forces subgroup. More information on the aerodynamics can be found in the section 2.2.3.

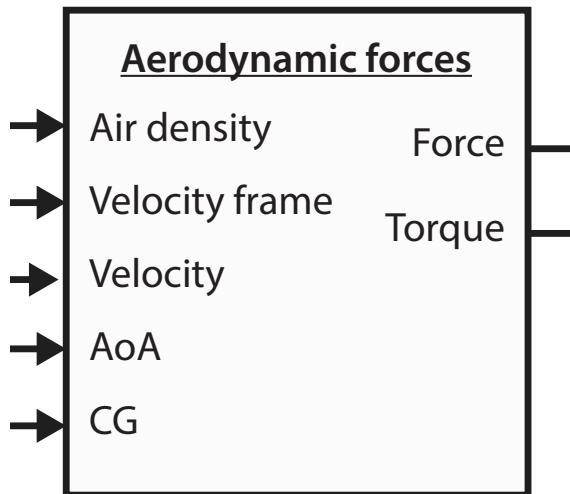


Figure 6.5: The subgroup Air forces block diagram

The subgroup Air forces consists of multiple lookup functions. These functions use the angle of attack and the Mach number in order to find the drag, lift, and the center of pressure in lookup graphs. The drag and lift are then converted to the total force affecting the launch vehicle. The center of pressure is collected with the center of gravity to calculate the air torque on the launch vehicle and is sent to the dynamics.

6.1.5 Center of Gravity

The Center of gravity subgroup is tasked with calculating the changing center of gravity and the inertia of the launch vehicle. Deeper explanation of how these parameters are

calculated can be found in section 2.3.

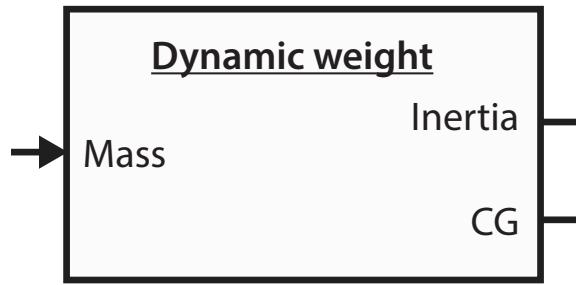


Figure 6.6: The subgroup Dynamic weight block diagram

The Center of gravity only takes in the mass of the fuel on board of the launch vehicle. It then together with all the different constants calculates the changing center of gravity and the inertia that changes with it. The center of gravity is then sent to the Air forces subgroup and the Control in order to calculate the thrust moment on the launch vehicle.

6.1.6 Control

The control subgroup is tasked with controlling the LV and setting the positions of the engines based on the implemented controller. As seen on figure 6.7 the inputs to the Control subgroup are the orientation of the LV, the velocity, the total mass, the angle of attack, and the side slip. The input for the controller is however taken directly from the calculation without any sensor disturbances etc. which is explained in the section 6.2.

More information on what controllers are used and a deeper understanding of the controller implementation can be found in chapter 5

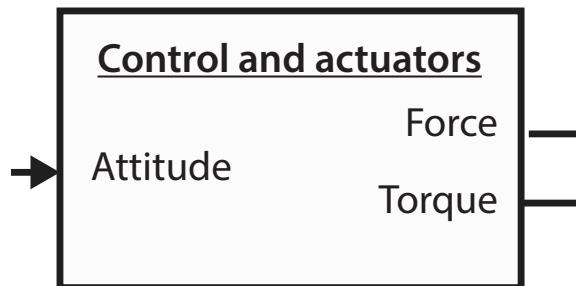


Figure 6.7: The subgroup Control and actuators block diagram

Inside the Control subgroup is the controller handling the inputs to the subgroup and outputting the position of the engines. This output is then used inside a block called thruster dynamics. Here is the moment and the total thrust calculated this is combined

with the orientation of the LV and sent to the Dynamics subgroup in order to calculate all the new states for the system.

6.1.7 Dynamics

The Dynamics subgroup is the last subgroup in the simulation. Here are all the different forces acting on the LV along with all the states taken in. These are the gravity, static pressure, jet dampening, the thrust force, the mass depletion, and the inertia matrix derivative as seen on the figure 6.8.

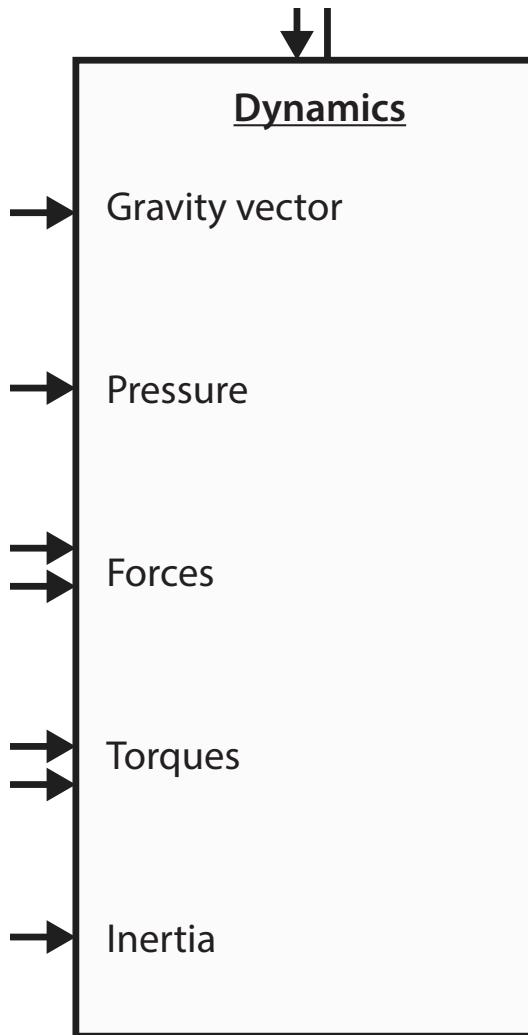


Figure 6.8: The subgroup Dynamics block diagram

When the simulation starts are the states initiated inside the dynamics subgroup and sent back to the States subgroup. However, when the simulation has run one loop is the new states calculated again and sent back to the States subgroup again. In other words, the output of the Dynamics subgroup is how the states of the LV behave according to all the dynamics in the environment. The states that are sent back are also projected on graphs

in order to create an overview of how the launch vehicle behaves. This is the simulation and the movement of the LV can also be seen animated inside the Animation subgroup.

6.1.8 Animation

The subgroup Animation take in the current orientation and the current position of the LV as seen on figure 6.9.

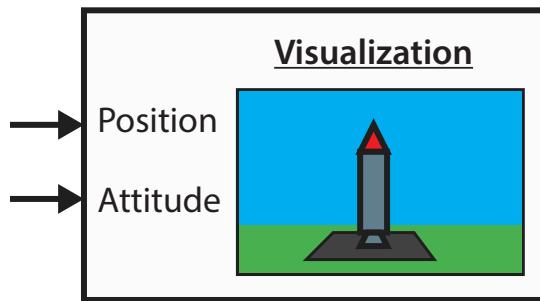


Figure 6.9: Block representing the visualization code with inputs

The current orientation and position are differentiated in order to get the movements of the LV. These movements are then inserted into a special featured function inside the Simulink in order to output a visual simulation of the LV in real time. An example of the visuals can be seen on figure 6.10 and as time passes will the rocket inside the figure move accordingly to the movement of the LV.

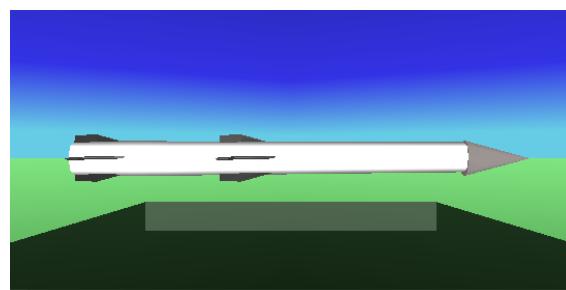


Figure 6.10: Visual output of the LV from the simulation.

6.2 Variation from the Simulation to Real Tests

There are multiple differences from how the system behaves in the simulation from how it would behave in a real test. However, this section will not go into details about how the simulation is missing some few minor dynamics and that some parameters are rounded off. This section is focused on what differences there exists that have an impact on the controller and its behavior.

6.2.1 Inputs into the Controller

The inputs into the controller do not contain any sensor disturbances since they are taken directly from the source of calculation. Therefore, there has not been any implementation

of sensor noise reduction. However, the currently AoA is feed directly into the controller. This is not something done easy if not impossible in real life and is normally done by following a long list of calculations and estimations of the trajectory [39]. Other inputs into the controller which is also taken directly from the source is the pitch, and the drift of the LV. The drift of the LV is calculated by integrating the sideways movement of the LV.

6.2.2 Gravity Turn

There are also other differences from the simulation to real tests. The gravity turn is normally done by following the trajectory 2.5. However, since the trajectory is not optimised for the specific LV used in the simulation will this increase the load on the LV. Since the actual angle of attack is obtainable in real time can the load on the LV be lowered by decreasing the AoA and deviate from the trajectory.

Both ways of doing the gravity turn have been implemented into the simulation.

6.3 Additional Dynamics

Due to time limit and the focus of this project being the development of a controller for the LV were the simulation not fully implemented with all dynamics. Some of these dynamics were the jet dampening, and the full aerodynamic coefficients. This will be discussed further in the chapter 9.

Results and Analysis

7

In order to verify the functionality of the controller an analysis of the closed loop response have been made and tests using the simulation have been conducted. The tests are to ensure that the controller behaves as expected in a nonlinear environment. The first section 7.1 will go into detail about how the tests were conducted. Here will it also be stated why the specific test was chosen. The next section 7.2 will show the results of the tests. There will be no analysis of the results here as it is just the raw data. The last section 7.3 will go into details about what the results are showing and analyses them.

7.1 Simulations

This section will go through the different tests which was conducted and the intentions of them. In all the different tests the same controller used as described in 5.2.1.

The final gains used in the simulation and the tests was from the structured H_∞ synthesis and ended with a gamma of approximately 4, and final gains:

$$K = \begin{pmatrix} 6.5525 & 0.8003 & -0.0001 & -0.0073 \end{pmatrix} \quad (7.1)$$

These gains were found by first simulating the trajectory. Here is the parameter Q focused upon and the highest values are found. Here can the extremes of the different parameters be noted down and used for synthesis of the H_∞ . The parameters used for the H_∞ controller were the values given in the table 7.1.

Parameter	Value	Unit
Velocity	570	m/s
Air density	0.24	kg/m ³
Inertia	2.32e6	kg/m ²
Center of Gravity	8.5	m
Center of Pressure	28	m
Gravity	9.6950	m/s ²
Mach number	2.8500	—
Pitch	67	°

Table 7.1: Extremes of the simulation parameters taken at the highest value of Q in the simulation of the trajectory

Maximum Q value test

The first test is done in order to estimate the maximum values of pressure Q affecting the LV throughout the trajectory. This is done in order to obtain the maximum controller

error allowed on the AoA as mentioned in requirement specification chapter 3. Equation 7.2 shows how this is done:

$$\alpha_{max} = \frac{Q_\alpha}{Q_{\max}} \quad (7.2)$$

Closed Loop Stability

In order to ensure the stability of the system have another test been carried out with the intentions of check for the stability of the closed loop. This is done by computing the eigenvalues of the closed loop system. These eigenvalues must be in the left half-plane of the S plane. This is because any poles in the right half-plane would create positive feedback and will therefore introduce instability.

Step Response

A step response has been generated from the controller gains and the state-space matrices. This test does not directly come from a requirement but give an insight in the performance. These performances characteristics checked for are the following:

- Rise time
- Settling time
- Overshoot (if any)
- Undershoot (if any)
- Peak
- Peak time

These characteristics will be the results from the step response and will be analysed in the end of this chapter in section 7.3.

Simulation Test

In order to test the full performance of the controller will the last test be to implement the controller into the simulation. Here it will be seen if the controller can fulfill the first requirement in table 3.1 in functional requirements "Follow the predetermined trajectory". All the different parameters will be saved throughout the simulation and can be found in results. These results will then be analysed in the end to see if the controller fulfill the first requirement.

The defining parameters can be found in the technical requirements 3.1.2. These defining parameters are the actuation done, the AoA error and therefore also the Q_α , the pitch error, and the lateral drift rate together with the total lateral drift.

The second and third functional requirement concerns with the introduction of disturbances and parameter variations. This has however not been tested for because the controller has not yet been developed for this.

7.2 Results

This section will display all the results found in the tests. None of the results will be analysed in this section, as this will be examined in the section 7.3.

Closed Loop Stability

Eigenvalues

The eigenvalues of the closed loop system are found to be the following:

$$\lambda = \begin{pmatrix} -3.4832 + 2.7407i \\ -3.4832 - 2.7407i \\ -0.0180 + 0.0066i \\ -0.0180 - 0.0066i \end{pmatrix} \quad (7.3)$$

There are two complex eigenvalue pairs. All the eigenvalues are on the left half-plane of the S plane.

Step Response

In figure 7.1 can the closed loop step response be seen.

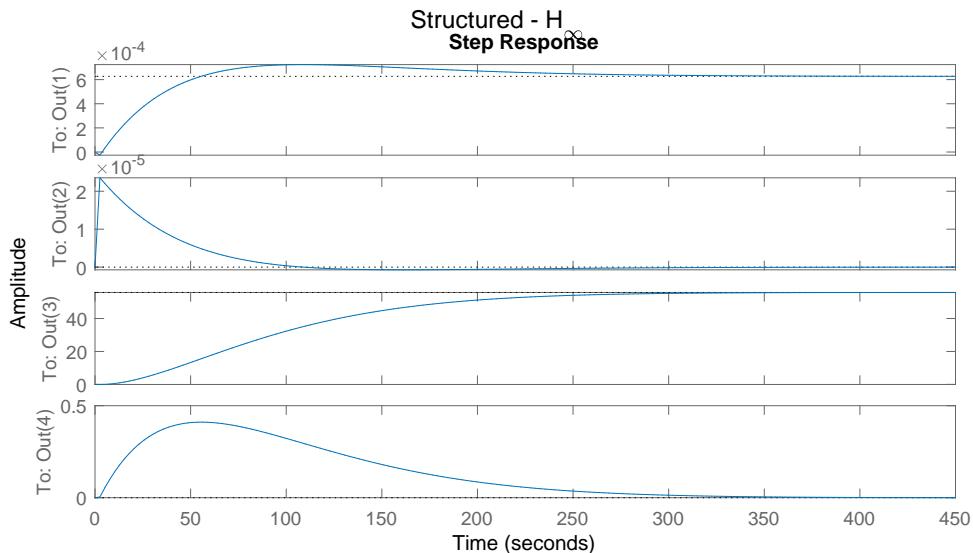


Figure 7.1: The step response of the closed loop system

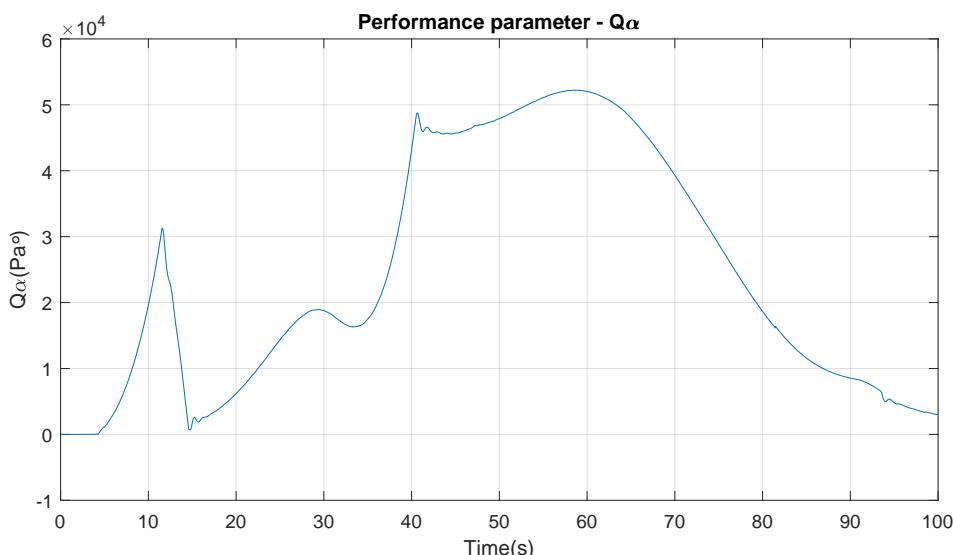
The characteristics of the step response are the seen in the table 7.2

Table 7.2: Step response results showing the characteristics

Step Response Characteristic	θ Step Response Values	z Step Response Values
Rise time	38,28	159,57
Settling time	279,23	269,95
Overshoot	15,53	0,02
Undershoot	4,18	0,08
Peak	7,24e -4	55,82
Peak time	107,50	476,05

Dynamic Pressure times Angle of Attack (Q_α)

The variation of Q_α throughout the whole trajectory can be seen on the figure 7.2.

**Figure 7.2: The performance parameter dynamic pressure times AoA, Q_α , over the duration of the flight.****Max Q value**

The max value of Q can be seen on the graph 7.3.

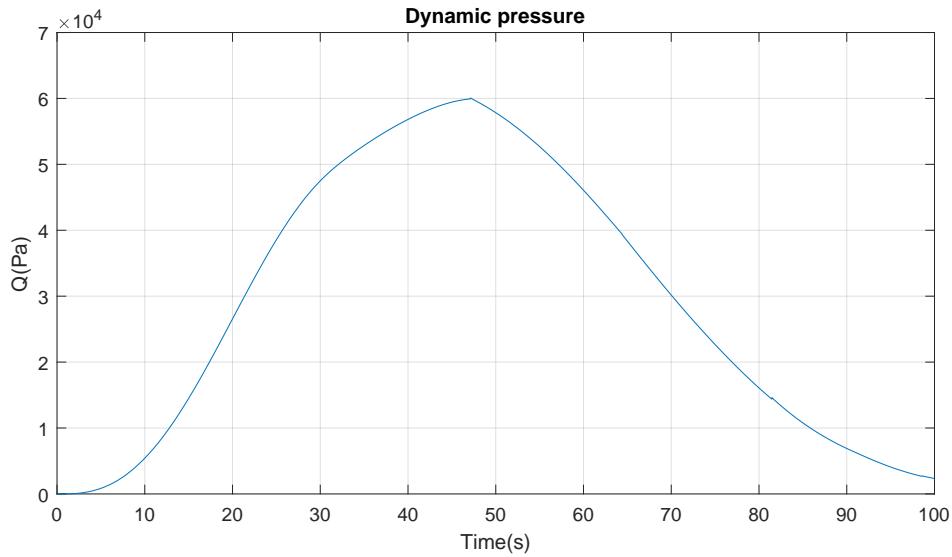


Figure 7.3: The Q parameter over the duration of the flight.

The maximal Q value reached on the graph is $6e4$.

7.2.1 Total Actuation

The total actuation done by the controller in the simulation is shown in the figure 7.4.

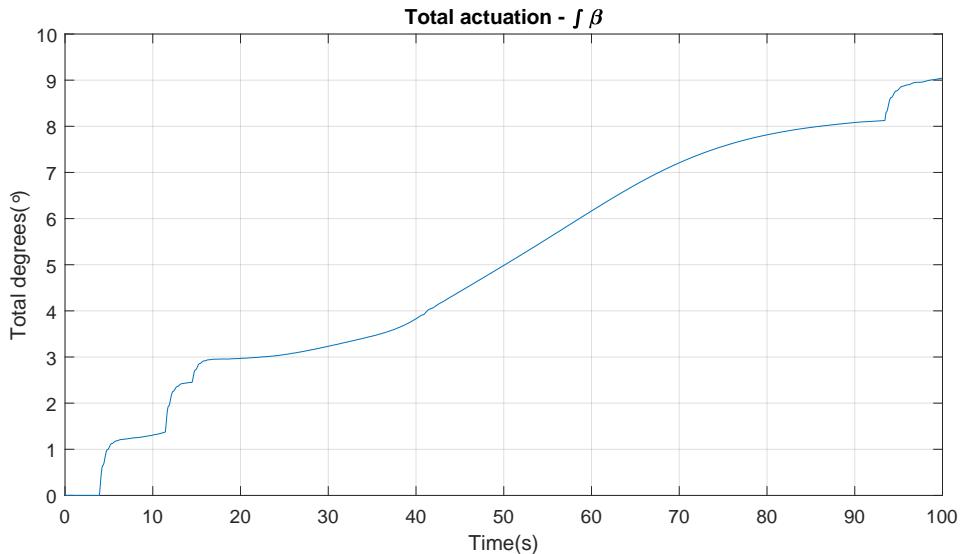


Figure 7.4: Total actuation integrated over the duration of the flight.

7.2.2 Pitch Error

The pitch divergence from the trajectory, or the pitch error in the controller can be seen on the graph 7.5.

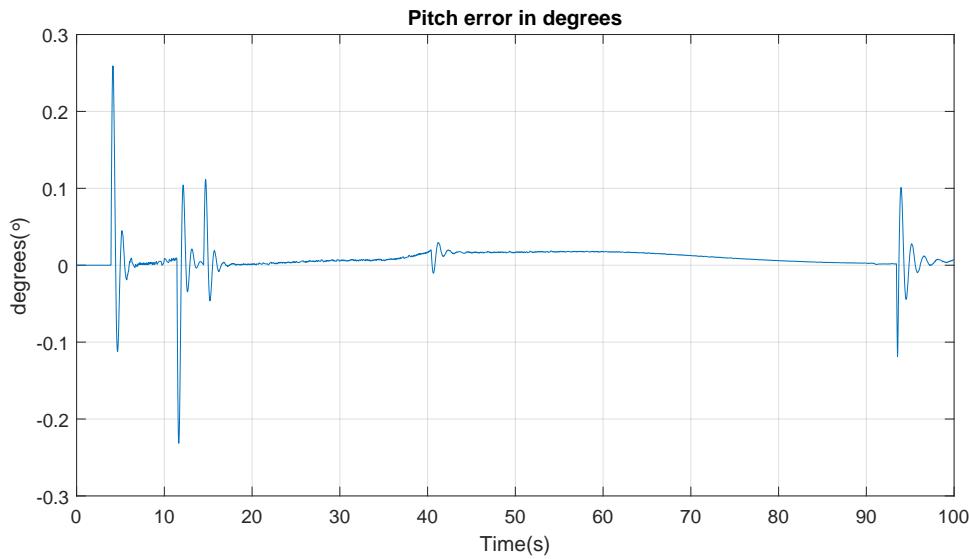


Figure 7.5: Pitch error over the duration of the flight.

7.2.3 Angle of Attack

The AoA is shown in the graph 7.6.

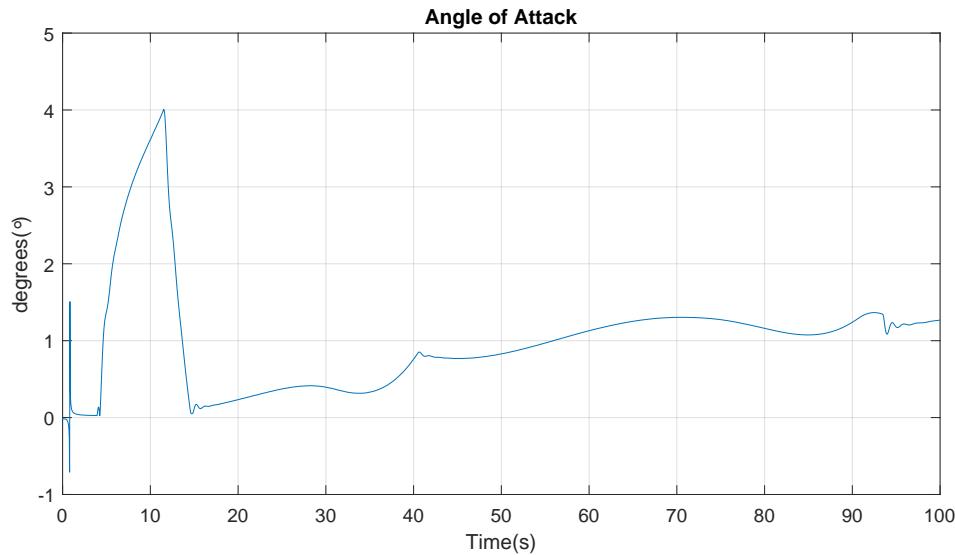


Figure 7.6: Angle of Attack over the duration of the flight.

7.2.4 Lateral Drift

The lateral drift away from the trajectory can be seen in graph 7.7.

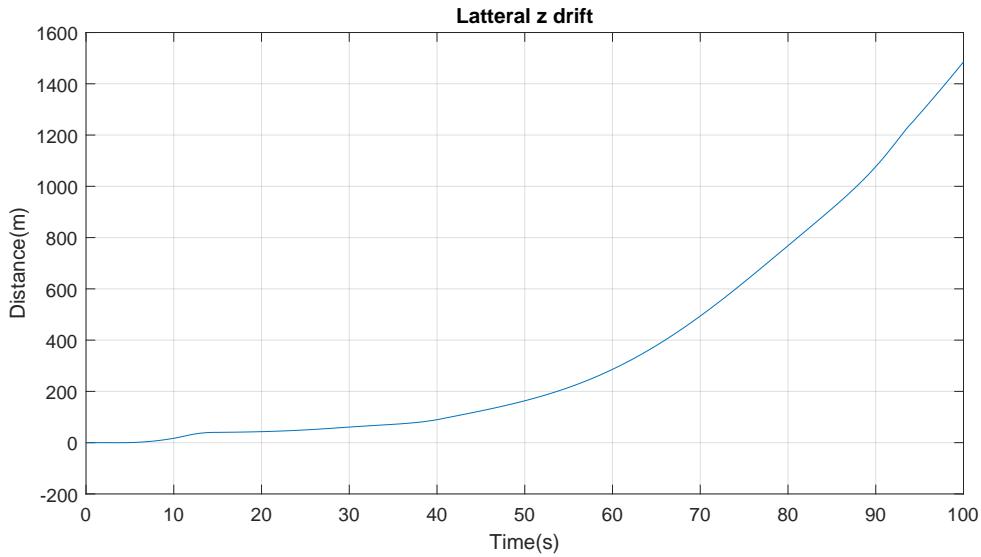


Figure 7.7: Lateral drift in the z direction over the duration of the flight.

7.2.5 Lateral Drift Rate

The rate of change in the lateral drift is shown on graph 7.8.

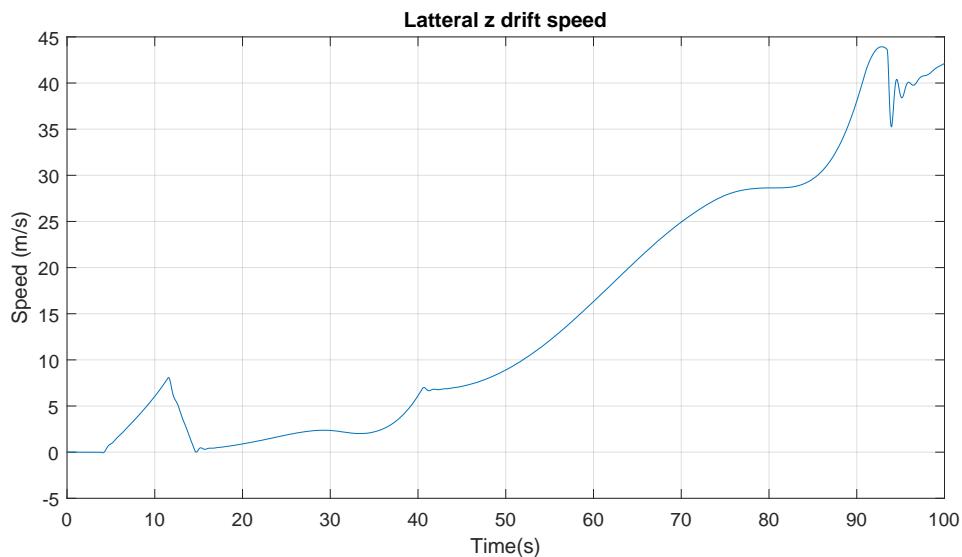


Figure 7.8: Lateral drift Rate in the z direction over the duration of the flight.

7.3 Result Analysis

Here will the different results from the tests be analysed. There will also be investigated if the requirements made from the chapter 3 are fulfilled. At last will there also be identified what needs to be discussed for further development.

Maximum Q value test

The test to find the maximal pressure acting on the LV has been done. With this it is possible to investigate what the max tolerance on the AoA error should be for the controller. The highest value that Q reached was $6e4$ and with this can we calculate the maximum allowed error on AoA.

$$\frac{2.75e5}{6e4} = 4.296 \quad (7.4)$$

Meaning that the maximum allowed AoA error is approximately 4.3 degrees. This will give an estimate about the quality of the performance made in the simulation test where the whole system with the controller is tested in the simulation.

Closed Loop Stability

The results from the closed loop stability shows the eigenvalues of the closed loop system. It can be seen that all the eigenvalues are negative and therefore are all of them also located in the left half-plane of the S-plane. This is good since it ensures stability in the controller if there are no parameter variations.

However, one of the complex pairs (namely the $0.018 \pm 0.0066i$) are very close to the imaginary axis in the S-plane. This is not good for stability in case of parameter variations. Because if the eigenvalues reach the imaginary axis and turns out to be poles in the system will this create oscillations and even worse can create unstable behavior if they reach the right half-plane of the S-plane.

This means that the close loop stability of the system is good for the first requirement in the functional requirements 3.1. But it cannot be implemented onto a real system since that will have parameter variations of some sort. Therefore, will this needed to be investigated for further development in chapter 9.

Step Response

The step responses for both the pitch angle and the lateral drift is again just like the eigenvalues ensuring that the stability of the closed loop, showing that the settling time is given within finite time. However, this is also mostly of what can be said for them.

The Step response for the pitch angle have a very high rise time of approximated 38 seconds together with a high settling time of approximated 280 seconds and these values are even higher for the lateral drift. This is not good performances for the controller. The reason for this is because the time period that the controller is supposed to be used for only take approximated 100 seconds which is from engine start on the launch platform until main engine cutoff. If the LV is expose to a disturbance of any kind will the controller not have corrected for the error before the trajectory is done. There is also an overshoot of 15% on the pitch variable. This is also not desirable, however is not as bad as the response time of the controller.

Simulation test

The simulation test results are good and seems to fulfill the first requirement of the functional requirements "Follow the predetermined trajectory" 3.1.

The controller is stable and there are no continues oscillations in the errors (pitch, AoA and lateral drift). As seen in the graph of Q_α 7.2 does this by far not exceed the requirement of 2.75e5 from the technical requirements 3.1.2. However, the maximum AoA calculated from the Maximum Q value test figure 7.1 comes very close to the maximum of 4,3 degrees with the error being 4,0 at its max. The reason why this is not a total loss is because the maximum allowed AoA is calculated from the worst-case scenario. On figure 7.3 can it be seen that the pressure Q which is affecting the LV starts out low and will first reach 50% of the maximum value after 20 seconds. The fluctuation in the AoA graph 7.6 happens in the beginning of the simulation before the 20 second, and therefore will the total pressure applied to the LV not be close to the critical maximum.

The requirement for the integrated β was set on 250 degrees in the technical requirements and this is also not violated as can be seen in the graph 7.4. The pitch error also looks good. There is no direct requirement on the maximum value of the pitch error, however, the maximum simulated error is 0,26 degrees. This is not a high value.

The requirement for the lateral control performance on the other hand is not satisfied. The maximum allowed drift in position is 500 meter and the maximum velocity is 15 meters per second 3.1.2. The results show that the lateral position drift ends by being at 1486 meters and does not seem to stop. This is almost 3 times more than allowed. The maximum lateral drift rate is measure at 44 meters per second. This is also almost 3 times more than allowed. The controller has failed this requirement, however, the reason for this can be discussed.

The reason for the lateral drift position and rate could be explained by the fact that the trajectory being followed in the simulation is not optimised for the specific LV used in this project. This creates a constant AoA which is coursing the drift. There was also not implemented a feed back in the controller for the lateral drift.

Test Conclusion

Looking at the controller and the overall compliance with the requirements are only one technical requirement broken for the first functional requirement. This is the lateral drift requirement. In order to fulfill this requirement will further development be needed in order to optimise a new trajectory for the specific LV which is used in this project. However, since it was not intended to develop a new optimised trajectory for the LV and doing this would make the controller fulfill all the requirements. Can it be said that the first functional requirement is fulfilled.

When this is said are there still a lot of left for further development. There was not enough time to develop an LPV controller and therefore was this not tested. The second and third functional requirement have not yet been implemented and this will need to be done in further development and the hand in of this project. This is mostly due to the time pressure of the hand in date of the project. The second functional requirement is

including disturbances such as wind and other non-deterministic dynamics affecting the LV creating this disturbance. The third functional requirement is about variable variation in the components of the LV. All of these will be discussed in chapter 9.

Conclusion 8

The objective of this project was to explore the control of a rocket during launch. Therefore, it was decided to build a robust controller in collaboration with ESA. The VEGA LV was used as reference for the design and it was decided to explore the control using multiple engines instead of VEGA's one engine. The dynamics of a rocket was explored in chapter 2. A linear state space containing the rigid body dynamics was developed in chapter 4.

The nonlinear rigid body simulation was implemented in Simulink in chapter 6. The simulation was implemented with clear interfaces between smaller subgroups. This made it compatible with future implementations, new dynamics, and controllers. The simulation was used to verify legitimacy of the augmented \mathcal{H}_∞ controller. However, not all the dynamics were implemented into the simulation due to the time limit and the prioritisation being on the development of the controller.

As mentioned in the problem formulation in section 2.8.1, the intention was to build an LPV controller as it is ideal for the highly changing parameters of the rocket during flight. It was then decided to use the weighing system used for augmented \mathcal{H}_∞ and it was decided to make an augmented structured \mathcal{H}_∞ controller first. This was because the \mathcal{H}_∞ uses the same structure as the LPV, the same weights, and the same structure of the plant. They were intended to be used for the LPV design, but they were not implemented due to time constraints. Therefore, only an augmented structured \mathcal{H}_∞ controller was developed for the most extreme point of the trajectory in time.

In chapter 7, all the tests were conducted in order to test the requirements set in chapter 3. Here, it was found that the tests of the implemented controller were promising as they fulfilled the highest prioritised functional requirement, "Follow the predetermined trajectory". The tests showed stable closed loop eigenvalues for the controller. Also, the step responses were stable but slow. Good tracking of the pitch error in the simulation and low actuation were present in the tests. As mentioned in the test conclusion 7.3, the lateral drift requirement were the only technical requirements not fulfilled. This is, however, most likely due to the fact that the trajectory is not made for the specific LV made in this project. The tests looking at the highest prioritised functional requirement can be concluded as good but are missing a new trajectory for the modified LV, and the further development should continue.

The rest of the functional requirements were not met. This is due to the fact that the LPV was not developed within the time limit. In chapter 9, this will be discussed in relation to further development.

To summarise, the simulation was developed to estimate all the dynamics affecting the LV

during launch. The simulation succeeded in simulating most of the dynamics of the LV along the trajectory. The simulation was also compatible with the implementation of the \mathcal{H}_∞ controller. The augmented \mathcal{H}_∞ controller managed to control the LV and follow the trajectory. This was without any disturbances and variable variations. The controller was promising even though it did not take the varying of the parameters into account which an LPV does. Overall the group is satisfied with the progress, learning, implemented controller synthesis and developed simulation that was the product of this project.

Discussion and Further Development 9

In this chapter will the conclusion made in chapter 8 be discussed. Also, the outcome from the analysis section 7.3 in the test chapter will be discussed and the suggested further development will be explained in further details here.

9.1 Discussion

This section will discuss the things that did not go as planned in the project. A better solution and alternative for future projects will also be proposed.

The prioritisation of Simulation and Controller Development

During the early stages of the project was the high-fidelity simulation developed instead of focusing on designing the controller. This was done to understand the dynamics better by testing and visualizing them, thereby getting an intuitive understanding. Moreover, an argument for having a nonlinear high-fidelity simulation was to verify that the designed controller was working on the derived linear state space representation of the whole system. Worst case could have been that the linearized model deviated too far from the nonlinear dynamics, such that the LV would crash. Therefore, it was important to verify that derived model was a valid linearized system. The upside was a full understanding the mechanics included in the simulation and became more experienced in Simulink. The downside of this decision to make a high-fidelity simulation, was that it consumed a majority of the time that was available for the project. In hindsight, it might have been better to spend that time on designing a robust controller more thoroughly. There are also still some dynamics which has not yet been included. This is because it was decided to stop development on the simulation and focus on the controllers. And because the simulation got to a point where it had the most important dynamics.

From the high-fidelity simulation it was experienced that the LV seemed more stable than expected. This could be a lack of dynamics such as disturbance in the form of wind, noise, and general parameter uncertainties.

Another Method for Linearizing the LV Model

Seen in hindsight, the method used for linearizing the LV dynamics done in Chapter 4, had too many gimmicks and "tricks" to simplify the equation. Instead, using Jacobian linearization might have been better, since that is a well-known systematic method for linearization.

9.2 Further Development

In the further development section will the different features and tasks that did not make the time limit be discussed. The discussion will include the specifics that did not get developed in time and what is proposed to do for further development.

Further Development for the Simulation

As mentioned in the conclusion in Chapter 8, the simulation did not get implemented with all the desired features. These features need further development in order to be implemented and creating the high-fidelity simulation. The dynamics missing are the jet dampening, the fuel sloshing inside the fuel tanks, the elastic bending modes, and the tail-wag-dog effect.

There are features missing from the simulation. The features are the ability to implement parameter uncertainties, placement of sensor positions, implementation of sensor noise, introduction of disturbance such as winds etc., and lastly implementation of delay into the system inside the simulation.

Another thing to work on, is to verify the simulation itself. Even though the simulation might have the correct dynamics implemented, controlled tests for each dynamic can be carried out to inspect if the behavior of the dynamics are indeed correct.

9.2.1 Further Development for the Controllers

Due to time limit was the LPV controller not developed. However, the \mathcal{H}_∞ was developed with the state-space of the LV. This means that the weights method can be reused for the development of the LPV since it is the same requirements setup for the two controllers. The state-space would, however, needed to be modified into a parameter varying state-space.

The \mathcal{H}_∞ controller's performance on moderate disturbances and moderate variable variations will not be investigated. As it would most likely fail and not succeed with the current since it was not built to handle this. Instead of making the \mathcal{H}_∞ more robust and introducing a disturbance delta into the control system, will the further development focus on the development of the LPV controller for the system. This is because it was the intention of creating such a controller from the beginning of the project, and it is estimated that it will perform better than the \mathcal{H}_∞ because of its properties with parameter variances.

The LPV should be developed with the desire to handle multiple parameter variances, uncertainties, delay, sensor noise, and disturbances.

Bending Modes

This entire project the rocket has been assumed to be a rigid body, however in reality this is far from the truth. Therefore, for future work a state space model for bending modes should be further studied and derived.

Bibliography

- [1] The asteroid trillionaires. <https://physicsworld.com/a/the-asteroid-trillionaires/>. Accessed: 2020-18-09.
- [2] Plane crash and fear of flying statistics. <https://flyfright.com/statistics/>. Accessed: 2020-27-11.
- [3] Worldwide orbital launch summary by year. <http://www.spacelaunchreport.com/logyear.html>. Accessed: 2020-19-09.
- [4] D. Navarro-Tapia. *Robust and Adaptive TVC Control Design Approaches for the VEGA Launcher*. University of Bristol, 2019.
- [5] The tyranny of the rocket equation. https://www.nasa.gov/mission_pages/station/expeditions/expedition30/tryanny.html. Accessed: 2020-22-09.
- [6] First launch of chinese kuaizhou-11 rocket ends in failure. <https://spacenews.com/first-launch-of-chinese-kuaizhou-11-rocket-ends-in-failure/>. Accessed: 2020-17-09.
- [7] Ariane 6. https://www.esa.int/Enabling_Support/Space_Transportation/Launch_vehicles/Ariane_6. Accessed: 2020-17-09.
- [8] First h3 launch slips to 2021. <https://spacenews.com/first-h3-launch-slips-to-2021/>. Accessed: 2020-17-09.
- [9] Soyuz-5 rocket to enter service in mid-2020s. <https://spacenews.com/soyuz-5-rocket-to-enter-service-in-mid-2020s/>. Accessed: 2020-17-09.
- [10] The annual compendium of commercial space transportation: 2018. https://www.faa.gov/about/office_org/headquarters_offices/ast/media/2018_AST_Comp.pdf. Accessed: 2020-17-09.
- [11] Arianespace. Vega - user's manual. https://www.arianespace.com/wp-content/uploads/2015/09/Vega-Users-Manual_Issue-04_April-2014.pdf. Accessed: 2020-25-09.
- [12] Stefano Bianchi. Vega, the european small launcher: Development status, future perspectives, and applications. *Acta Astronautica*, 63(1):416 – 427, 2008. Touching Humanity - Space for Improving Quality of Life. Selected Proceedings of the 58th International Astronautical Federation Congress, Hyderabad, India, 24-28 September 2007.
- [13] Astronautix. P80. <http://www.astronautix.com/p/p80.html>. Accessed: 2020-29-09.

- [14] Avio. Vega satellite launcher. https://web.archive.org/web/20141101022341/http://www.avio.com/files/catalog/pdf/avum_78.pdf. Accessed: 2020-01-10.
- [15] ESA. Vega-c elements. https://www.esa.int/ESA_Multimedia/Images/2020/04/Vega-C_elements. Accessed: 2020-29-09.
- [16] P.V.M. Simplicio. *Guidance and Control Elements for Improved Access to Space: From Planetary Landers to Reusable Launchers*. University of Bristol, 2019.
- [17] Arthur L Greensite. *Analysis and design of space vehicle flight control systems*, volume 11. Spartan books New York, 1970.
- [18] NASA. Conservation of momentum. <https://www.grc.nasa.gov/WWW/K-12/airplane/conmo.html>. Accessed: 2020-12-10.
- [19] NASA. Dynamic pressure. <https://www.grc.nasa.gov/WWW/K-12/airplane/dynpress.html>. Accessed: 2020-12-10.
- [20] Peter H Zipfel. *Modeling and simulation of aerospace vehicle dynamics*. American Institute of Aeronautics and Astronautics, 2007.
- [21] Mansfield Connecticut. Gravity changes with altitude. <https://www.mansfieldct.org/Schools/MMS/staff/hand/lawsgravaltitude.htm>. Accessed: 2020-11-10.
- [22] Fandom. Specific impulse. https://engineering.fandom.com/wiki/Specific_impulse. Accessed: 2020-11-10.
- [23] Robert A. Braeunig. Atmospheric models. <http://www.braeunig.us/space/atmmodel.htm>. Accessed: 2020-20-09.
- [24] G. F. McDONOUGH and et al. Wind effects on launch vehicles. <https://www.sto.nato.int/publications/AGARD/AGARD-AG-115/AGARD0G115.pdf>. Accessed: 2020-30-09.
- [25] Dale L. Johnson and William W. Vaughan. The wind environment interactions relative to launch vehicle design. https://www.scielo.br/scielo.php?pid=S2175-91462020000100301&script=sci_arttext#aff2. Accessed: 2020-20-09.
- [26] Diego Navarro Tapia. Robust and adaptive tvc control design approaches for the vega launcher. Accessed: 2020-28-09.
- [27] Ashish Tewari. Atmospheric and space flight dynamics: Modeling and simulation with matlab and simulink. Accessed: 2020-28-09.
- [28] Tiziano Pulecchi, Andres Marcos, Christophe Roux, Valerio Mantini, and Samir Bennani. Thrust vector control robustness of axial-symmetric launch vehicles with fuel slosh. https://www.tasc-group.com/uploads/6/1/1/9/61196919/pulecchimarcos_robanalysis_aiaagnc12.pdf. Accessed: 2020-07-10.
- [29] Yeager and Jessie C. Implementation and testing of turbulence models for the f18-harv simulation. <https://ntrs.nasa.gov/citations/19980028448>. Accessed: 2020-21-10.

- [30] Inc The MathWorks. Dryden wind turbulence model (continuous). <https://www.mathworks.com/help/aeroblks/drydenwindturbulencemodelcontinuous.html>. Accessed: 2020-21-10.
- [31] Inc. The MathWorks. Dryden wind turbulence model (continuous). <https://ntrs.nasa.gov/citations/19980028448>. Accessed: 2020-02-12.
- [32] Samuel Glasstone. *Sourcebook on the space sciences*. 1965.
- [33] ESA. The european small launcher. <https://www.esa.int/esapub/br/br257/br257.pdf>. Accessed: 2020-05-10.
- [34] Inc. The MathWorks. Simulink. <https://se.mathworks.com/products/simulink.html>. Accessed: 2020-12-10.
- [35] Pedro Simplicio, Samir Bennani, Andrés Marcos, Christophe Roux, and Xavier Lefort. Structured singular-value analysis of the vega launcher in atmospheric flight. *Journal of Guidance, Control, and Dynamics*, 39:1342–1355, 06 2016.
- [36] Sigurd Skogestad and Ian Postlethwaite. *Multivariable feedback control: analysis and design*, volume 2. Wiley New York, 2005.
- [37] Diego Navarro-Tapia, Andres Marcos, Samir Bennani, and Christophe Roux. Structured h-infinity and linear parameter varying control design for the vega launch vehicle. September 2017. Eucass - 7th European Conference for Aeronautics and Space Sciences ; Conference date: 03-07-2017 Through 06-07-2017.
- [38] Corentin Briat. *Linear Parameter-Varying and Time-Delay Systems "Analysis, Observation, Filtering Control"*, volume 3. Springer-Verlag Berlin Heidelberg, 2015.
- [39] Jack Kevin Ly. Angle of attack determination using inertial navigation system data from flight tests. https://trace.tennessee.edu/cgi/viewcontent.cgi?referer=https://www.google.com/&httpsredir=1&article=5851&context=utk_gradthes. Accessed: 2020-10-12.

Parameters used in MATLAB A

A.1 Launch vehicle parameters

The table A.1 lists all of the launch vehicle's parameters that is used throughout this project with their correct SI units.

Table A.1: Launch vehicle parameters

Launch vehicle parameters	N/A	SI unit
Total mass without fuel	47014	<i>kg</i>
Mass of the fuel	88400	<i>kg</i>
Engine nozzle radius	0,94	<i>m</i>
Engine nozzle area	2,76	<i>m</i> ²
Engine placement radius	1	<i>m</i>
Initial first state thrust	2200	<i>M · N</i>
Payload radius	0,75	<i>m</i>
Payload length	5	<i>m</i>
Fairing radius	1,3	<i>m</i>
Fairing length	7,88	<i>m</i>
Payload adapter radius	1,09	<i>m</i>
Payload adapter length	1,46	<i>m</i>
Upper stage radius	1,09	<i>m</i>
Upper stage length	2,04	<i>m</i>
3rd stage radius	0,95	<i>m</i>
3rd stage length	4,12	<i>m</i>
2nd stage radius	0,95	<i>m</i>
2nd stage length	8,39	<i>m</i>
1st stage radius	1,5	<i>m</i>
1st stage length	11,2	<i>m</i>
Total length	29.93	<i>m</i>
Initial center of gravity	8.72	<i>m</i>
Without fuel inertia x	35,74	<i>M · kg · m</i> ²
Without fuel inertia y and z	2,80	<i>G · kg · m</i> ²
Total inertia x	135,19	<i>M · kg · m</i> ²
Total inertia y and z	4,64	<i>G · kg · m</i> ²