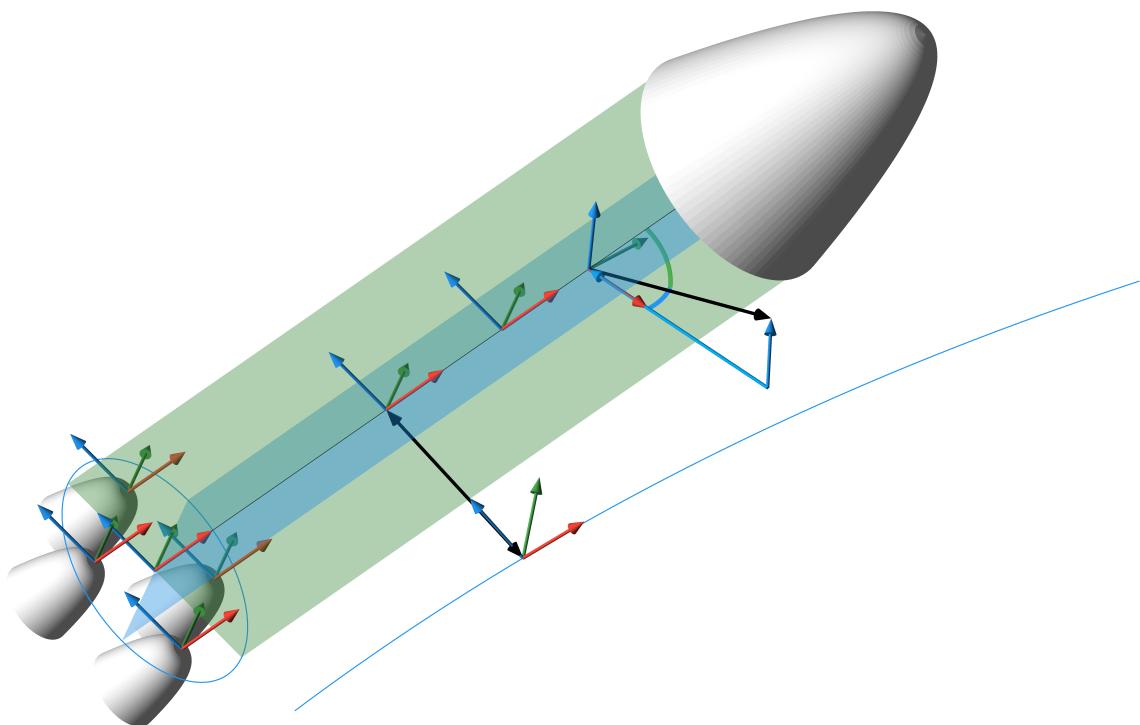


MASTER'S THESIS BY FREDERIK SKYT DÆNCKER RASMUSSEN

Linear Parameter-Varying Control of Modified Vega Launch Vehicle

TENTH SEMESTER OF CONTROL AND AUTOMATION



AALBORG UNIVERSITY

2021



Title:

Linear Parameter-Varying Control of Modified Vega Launch Vehicle

Project:

P10-project

Period of Project:

February 1st 2021 - June 3rd 2021

Project Group:

CA10-1039

Members:

Frederik Skyt Dæncker Rasmussen

Supervisors:

Rafal Wisniewski

ECTS points: 30

Printed exemplary: 0

Page number: 81

Ended: July 3, 2021

Abstract

The goal of this thesis is to design a Linear Parameter-Varying (LPV) controller for a Launch vehicle based on the Vega Launch Vehicle (LV). First a contextual analysis of the launch, flight and landing of LVs is presented. Based on the contextual analysis the objectives are specified to an LPV attitude controller for the gravity turn part of the trajectory. The requirements are set with performance parameters for stability, noise, tracking and disturbance rejection. Subsequently, a state space model of the rigid body LV is created using the dynamic equations and the relevant frame transformations. Using the developed rigid body model, a two dimensional model for the pitch plane is extended with models for bending modes, delay, wind, noise and actuation. This model is then augmented with a set of weights that are based on specified requirements. Next, the model is successfully used for synthesizing a grid based LPV controller. Finally, the system with the controller is simulated in an LPV simulation, which verifies that the controller meets the specified performance requirements.

Nomenclature

Abbreviations

Abbreviation	Definition
AOA	Angle Of Attack
DCM	Direction Cosine Matrix
DRL	Down Range Landing
ECEF	Earth-Centered Earth-Fixed
ECI	Earth-Centered Inertial
ESA	European Space Agency
INS	Inertial Navigation System
IQC	Integral Quadratic Constraints
LFT	Linear Fractional Transform
LPV	Linear Parameter Varying
LLA	Latitude, Longitude and Altitude
LOV	Loss Of Vehicle
LTI	Linear Time Invariant
LTV	Linear Time Varying
LV	Launch Vehicle
MECO	Main Engine Cut Off
RTLS	Return To Launch Site
S	Sensitivity
T	Complementary Sensitivity
TVC	Thrust Vector Control
VTVL	Vertical Takeoff Vertical Landing

Symbols and Notation

- C_{XY} - DCM from frame Y to frame X [-]
 C_L - Aerodynamic lift coefficient [-]
 C_D - Aerodynamic drag coefficient [-]
 R_x, R_y, R_z - Rotation matrices rotating around the respective axis [-]
 Q_d - Dynamic pressure [Pa]
 ρ - Air density [kg/m^3]
 ρ - (From Ch. 5) Scheduling parameter [-]
 $V_{\text{name},X}$ - Velocity vector in frame X [m/s]
 $\dot{V}_{\text{name},X}$ - Acceleration vector in frame X [m/s^2]
 $\Omega_{\text{name},X}$ - Angular velocity vector in frame X [rad/s]
 $F_{\text{name},X}$ - Force vector in frame X [N]
 F_{name} - Force [N]
 $M_{\text{name},X}$ - Vector describing moments in frame X [N m]
 S_{ref} - Reference surface area [m^2]

-
- M - Mach number [–]
 α_{eff} - Effective Angle of Attack [rad]
 α_{aoa} - Angle of Attack [rad]
 α_{sideslip} - Sideslip Angle [rad]
 $P_{\text{name},X}$ - Vector representing a point in frame X [m]
 P_{name} - Vector representing a point [m]
 m_N - Mass of engine nozzle [kg]
 β_X - Actuation angle in frame X [rad]
 J_N - Inertia of engine nozzle [kg/m²]
 \dot{v}_g - Gravitational acceleration [m/s²]
 $v_w(s)$ - Velocity of the modelled wind [m/s]
 $n_w(s)$ - White noise [–]
 $v_{wp}(h)$ - Wind speed profile [m/s]
 L_h - Turbulence length scale [m]
 σ_h - Standard deviation for Dryden filter [–]
 ϵ - Proportional deviation from nominal thrust [–]
 $Q\alpha$ - Load performance parameter [Pa °]
 Θ - Vector describing the orientation of the LV [rad]
 Ω - Vector describing the angular velocity of the LV [rad/s]
 R - Vector describing the position of the LV [m]
 V - Vector describing the velocity of the LV [m/s]
 x_{name} - State vector [–]
 u_{name} - Input vector [–]
 y_{name} - Output vector [–]
 A_{name} - Linear dynamics state matrix [–]
 B_{name} - Linear dynamics input matrix [–]
 C_{name} - Linear dynamics output matrix [–]
 D_{name} - Linear dynamics feedthrough matrix [–]
 I - Identity matrix [–]
 \mathbf{J}_f - Jacobian of function f [–]
 q_i - i 'th bending mode state [–]
 ω_{q_i} - Natural frequency of the i 'th bending mode [rad/s]
 ζ_{q_i} - i 'th bending mode damping ratio[–]
 m_N - Mass of the nozzle [kg]
 l_N - Length from the PVP of the nozzle to its center of gravity [m]
 β - Angle of the actuator in the relevant control plane [rad]
 $\psi_{\text{PVP},i}$ - i 'th bending mode lateral movement coefficient [m]
 $\psi'_{\text{PVP},i}$ - i 'th bending mode rotational coefficient [rad]
 G_{name} - Dynamic model [–]
 m - Mass of LV [kg]
 Δ_{name} - Arbitrary matrix operator[–]
 $K(s)$ - Dynamic controller [–]
 $P(s)$ - Dynamic plant [–]
 $\mathcal{T}_{ed}(s)$ - Interconnection used for controller synthesis [–]
 $\mathcal{T}_{e'd'}(s)$ - Augmented interconnection used for controller synthesis [–]
 W_o - Output weight[–]
-

- W_o - Input weight [-]
 γ^* - Maximum gain of system with optimal controller [-]
 γ - Maximum gain of system with sub optimal controller [-]
 \star - Entry of matrix that will make the matrix symmetric. [-]
 W_{name} - Weight for controller synthesis [-]
 τ - Time delay [s]
 $\epsilon_{\text{NF},i}$ - Depth of the attenuation from the i'th notch filter [-]
 η_i - Width of the i'th notch filter [-]
 $\omega_{\text{BM},i}$ - Center frequency of the i'th notch filter [-]
 $P(\rho)$ - Lyapunov function [-]

Table of Contents

1	Introduction	3
1.1	Motivation	3
2	Contextual Analysis	7
2.1	Launch and Landing	7
2.2	The Vega Launch Vehicle	9
2.3	Definition of Variables and Parameters	10
2.4	Reference frames	12
2.5	Dynamics	15
2.6	Disturbance, Noise and Parameter Uncertainties	18
2.7	Data Used for Parameter Variations	20
3	Objectives and Requirements	21
3.1	Objectives	21
3.2	Requirements	22
4	Modeling of LV	25
4.1	Parameter Overview	25
4.2	Full Dynamic Equations	26
4.3	Dynamic Equations as a Function of states	28
4.4	Jacobian of Dynamic Equations	29
4.5	Inserting Operating Points and Known variables	29
4.6	Rigid-body State Space	30
4.7	Bending Mode Model	32
4.8	Final State Space for LV	33
5	Controller Design	37
5.1	Control Introduction	37
5.2	Control Problem Formulation	43
5.3	Control Sub-Models	44
5.4	Weight Selection	47
5.5	Synthesis of controllers	51
6	Testing	55
6.1	LPV controller	55
6.2	Test Conclusion	64
7	Conclusion	65
7.1	Discussion	65
7.2	Conclusion	67
Bibliography		69

A Appendix	71
A.1 Parameter Trajectories	71

Introduction 1

1.1 Motivation

Going to space is one of humanity's greatest achievements since we took the first small steps into space. We still keep going and have learned a lot. But it is still one of the hardest, most dangerous and expensive things we do. Yet we continue to do so because of all that we can potentially learn and gain from it in the long-term. While there was a small drop in launches around the turn of the century, in recent years there has been a renewed interest in space and a significant increase in launches [1] [2] [3].

The interest can likely be attributed to the technology maturing, and commercial space flight is becoming viable. Thereby accelerating the development of more efficient spaceflight. A prime example of this is SpaceX and its first of a kind reusable rocket boosters, which reuses the otherwise discarded boosters. While there have been attempts at reusability such as the space shuttles, those efforts have been discontinued due to the high cost of making them operational again after flight [4]. Other companies such as Blue origin and Rocket lab are just two more examples out of the many new companies looking to send rockets to space and recover the rockets to greatly reduce the price per launch. Additionally, organisations such as the European Space Agency (ESA) are also developing their first fully reusable rocket stage Themis [5].

1.1.1 Advantages of going to space

There is a wide range of advantages from going to space. The first one is scientific. The Scientific advances from going to space are vast, and can be a part of answering some of the most interesting questions in science. From helping us find the origin of life, and whether there is any other life out there, to a better understanding of physics. Going to space will also help us with astronomy where one of the great advantages in space is having telescopes outside the otherwise obstructing atmosphere, or directly sending missions to the celestial object of interest.

Once the technology matures it could allow us to explore entirely new environments of planets, moons and asteroids. And the difficulties and constraints from going to space will be forcing us to push limits. Which have so far, and hopefully will continue to, spark new ideas and lead to innovations. Innovations that will affect us and help us in everyday life [6].

Most of the technology we use today already rely on the products of space exploration. A prime example of this are the networks and location services countless people use every day: all of them are reliant on satellites. And while it is still very energy intensive to go to space, it can be one of the ways that we can spare our planet from the damage that comes from heavy industry, mining and the likes [7]. Once enough material has been lifted

to space to build a self-sustained base of operation outside Earth's gravity well, getting around space is energy efficient. And production of things such as rockets and fuel in space will have no apparent effect on the Earth's fragile climate, if handled right. And space is full of the resources and energy required for industry. Metal for construction is readily available and otherwise rare and precious metals can be mined and brought back to earth [7]. However, these are long term goals. Challenges such as reliability and cost will have to be solved first.

1.1.2 Price of Going to Space

The cost of going to space has historically been astronomically high. While there was a great reduction in price during the intense years of development during the Space Race, where competition drove development. The cost leveled off after the competition was over and the volume of flights reduced. Only recently, the price has been decreasing because of commercial interests that very much relies on making it economically viable. The cost is now coming down, and it does so fast. This commercialisation of spaceflight, is mainly led by companies such as Blue Origin and right now, especially SpaceX [4].

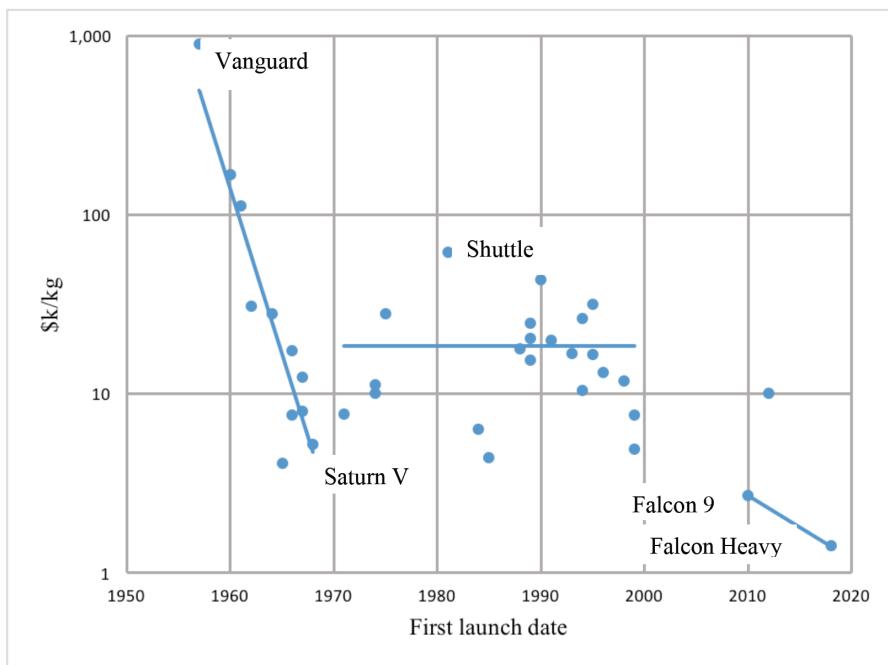


Figure 1.1: The cost over time to get a kg to Low Earth Orbit.[4]

The plot on figure 1.1 shows how the cost of going to space is once again declining in recent years. The sharp decline in recent years from SpaceX is mainly due to its company structure. A structure that is closer to a Silicon Valley company rather than a more bureaucratic government agency according to a paper by NASA [4]. However, they also conclude that in the longer term the main drivers of price reduction will be the reusability of the Launch Vehicles (LV).

1.1.3 Examples of Reusable Launch Vehicles

While SpaceX is currently taking the lead with its progress of reusable LVs there are many examples of launches to space with reusable vehicles in development. They are all with pros and cons, and some more successful than others [8]. A few of the currently most interesting examples will be discussed here.

The first implemented system that is partially reusable is the Space Shuttle as part of the Space Transportation System. It was a project with great potential. I worked by having the main vehicle, the Space Shuttle, that was reusable and could glide to the ground like a plane and thus save fuel. It, however, had a disposable fuel tank, and disposable solid state side boosters. Thus, it was only partially reusable and very expensive to get ready for a new flight. While it was a great success in many ways, it plateaued in price and was eventually discontinued [9].

One of the earliest companies with the vision of reducing the cost of going to space by reusing the rockets was the company Blue Origin. Their long-term goal is to make space-flight cheap and safe, and to move heavy industry and people to space to spare the planet. And while they were the first to successfully land a suborbital booster, they have not yet landed an orbital booster. Currently their reusability is intended for suborbital space tourism, but they have plans to later make reusable orbital LVs [10].

Another example of reusable LVs that is taking a slightly different approach is from the company Virgin Galactic. Their focus is also suborbital space tourism. While not a new idea, the strategy employed by Virgin Galactic is the most unique approach towards reusable LVs in these examples. They use a large carrier plane as the launch platform for their spacecraft. The spacecraft will then use a solid rocket booster to the edge of space. This allows the space tourists to float and experience space for a few minutes. For descent it uses a similar approach to the space shuttle, but at lower speeds. It first aerobrakes, and then slides down and lands like an airplane [11].

And the final example is SpaceX with its Falcon rockets and the upcoming Starship that have so far had the greatest success with reducing costs by reusing their boosters. SpaceX was founded with the long term goal of colonizing the planet Mars. To make that goal feasible, the cost of getting to space had to be reduced significantly. They also decided to aim for reusability to make it more affordable in the long run. Their approach is using both vertical takeoff and landing. SpaceX have successfully landed and reused their falcon 9 rockets multiple times so far. This is one of the reasons SpaceX can get the cost down as far as they have. Their Starship Rocket takes a slightly different approach: While still landing vertically, it uses air braking like some of the other examples. It glides down with the broad side first to get as big of a surface and air resistance as possible. It is controlled during this phase with large control surfaces that can be adjusted for drag. Then just in time it uses the boosters to turn vertical and land. This however, has just recently been successfully demonstrated in a prototype, and is still some time from being used for commercial flights. Because of the promising development and projected low cost of this Starship, SpaceX was recently awarded a contract by NASA to use a modified Starship to

land personal on the moon [12].

For this project the Vertical Takeoff Vertical Landing (VTVL) approach similar to that used by the SpaceX Falcon 9 will be further investigated. This is because it seems to be the most successful one so far and the simplest maneuver. But also because this project is in collaboration with the ESA. ESA has an interest in a similar case for one of their own LVs.

1.1.4 Collaboration With European Space Agency

This thesis is in collaboration with Automatic Control Systems Analyst Finn Ankersen and GNC Systems Engineer Pedro Simplício from ESA. It is a continuation of a project from the 9th semester[13]. ESA has a specific interest in control of a reusable LV. More specifically, ESA is interested in the control of a rocket booster using multiple main engines, how this can be used to control roll and how uncertainties in the thrust of the engines can be accounted for using Linear Parameter Varying control. In this collaboration, ESA will assist with external supervision by Finn Ankersen and Pedro Simplício on the project, that will help with their experience of the subject and help guide the project in a direction that will also be in ESAs interest. ESA has also provided the following for the project:

- Aerodynamic coefficients for the Vega launcher
- Bending mode data for the Vega launcher
- Trajectory for a mission to low earth orbit
- Thrust requirements for engine clusters

1.1.5 Initial Problem formulation

For this project, an initial problem formulation for guiding the problem analysis and requirements:

How can a robust LPV controller for VTVL with a LV similar to the Vega launcher with multiple main engines be designed, while taking into account the parameter uncertainty, and especially the difference in thrust among the multiple engines.

After the problem analysis a final problem formulation will be stated.

Contextual Analysis 2

In this contextual analysis the subjects that are relevant to the initial problem formulation will be investigated. First an introduction to the trajectory of a booster from launch to landing. Then an overview of the Vega Launcher used for the LV model. In the end there will be a few sections describing the definitions, dynamics and other background information used to derive the model in a later chapter.

2.1 Launch and Landing

First this section will investigate the trajectory of a reusable rocket and conclude on what will be used for this project.

2.1.1 Landing site

When landing the LV after ascent there are two main ways to land. The two, each with its own pros and cons. The LV can land on a platform down range close to where the trajectory takes it. This is called Down Range Landing (DRL). Or it can spend fuel on a boostback burn to accelerate back to the launch site. This is called Return To Launch Site (RTLS). Both can be seen on figure 2.1 where the red trajectory illustrates a DRL and the blue a RTLS. DRL has the advantage that it requires less fuel because it does not have to make the boostback burn. This fuel can then be used to carry a larger payload. However, the landing is often on a floating platform like the drone-ship SpaceX is using. And it still requires the LV to be brought back to the Launch site by some other means. It can therefore be advantageous to fly back to the launch site if the payload is small enough

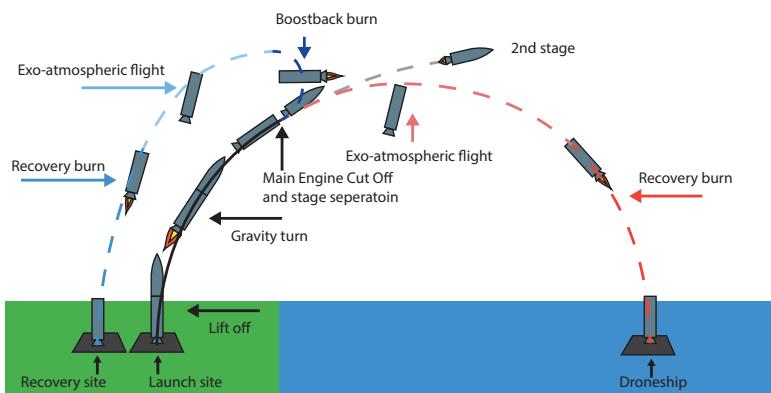


Figure 2.1: Illustration of the different landing trajectories Down Range Landing (red) and Return To Launch Site (blue).

that there is enough fuel leftover to do the boostback burn. However, if you break it up in phases as seen on figure 2.1, the recovery burn is similar in both cases. The only difference for the booster is what happens between Main Engine Cut Off (MECO) and the recovery burn.

In this project the launch and landing will be handled separately. Launch is from takeoff to MECO, and landing is from recovery burn to Landing

2.1.2 Launch trajectory

The launch trajectory mostly follows a gravity turn. A gravity turn is when the LV is turning at a rate such that the Coriolis acceleration equals the gravitational acceleration perpendicular to the body of the LV. Thus, in the body frame there is no acceleration sideways which means that it will keep the Angle Of Attack (AOA) steady. Such a trajectory has been provided by ESA which can be seen on figure 2.2

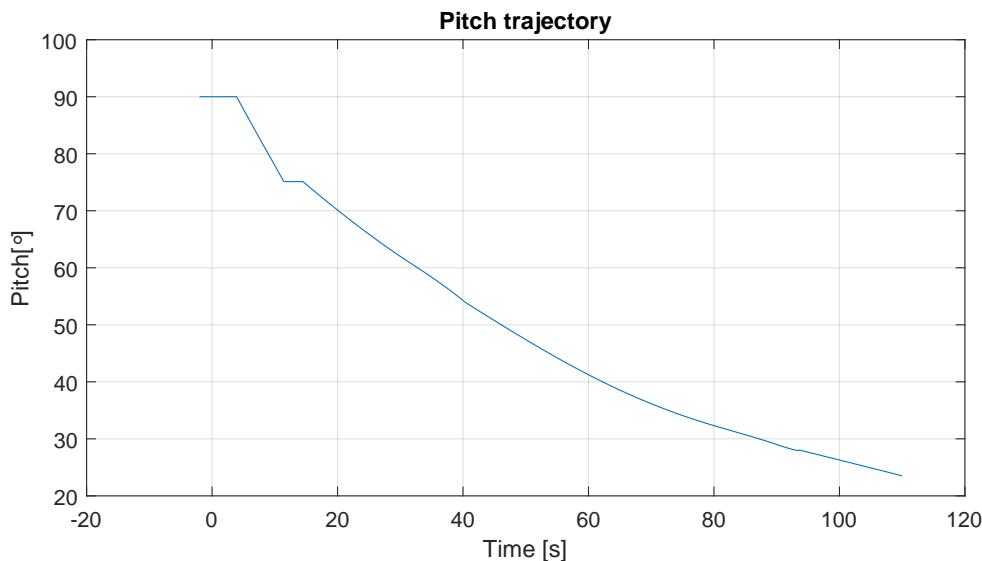


Figure 2.2: The trajectory given as reference by ESA

This is the pitch in the launch frame which has origin on the launchpad, and the pitch plane is aligned with the trajectory which means that for a common launch trajectory the yaw angle will be constant and zero along the entire trajectory. The frames will be explained in section 2.4. The trajectory is used from takeoff to MECO. The first linear piece of the graph, the LV, is flying up straight to clear the launchpad. Next piece with constant pitch rate, it will turn the trajectory away from the launch site in case of accidents. Then after reaching an AOA of zero, it follows the gravity turn. This trajectory is however open-loop guidance as the trajectory will not be adjusted under flight and it will only command the attitude of the rocket during ascent. The reference pitch is predetermined based on a simulation. Therefore there will be dispersions in the path and speed of the trajectory. These dispersions have to be small enough that it can be corrected while flying outside the atmosphere by the other stages of the rocket. The booster is intended to be corrected on the way down. Additionally, the thrust used for ascent in this project is also provided by ESA and can be seen on figure 2.3.

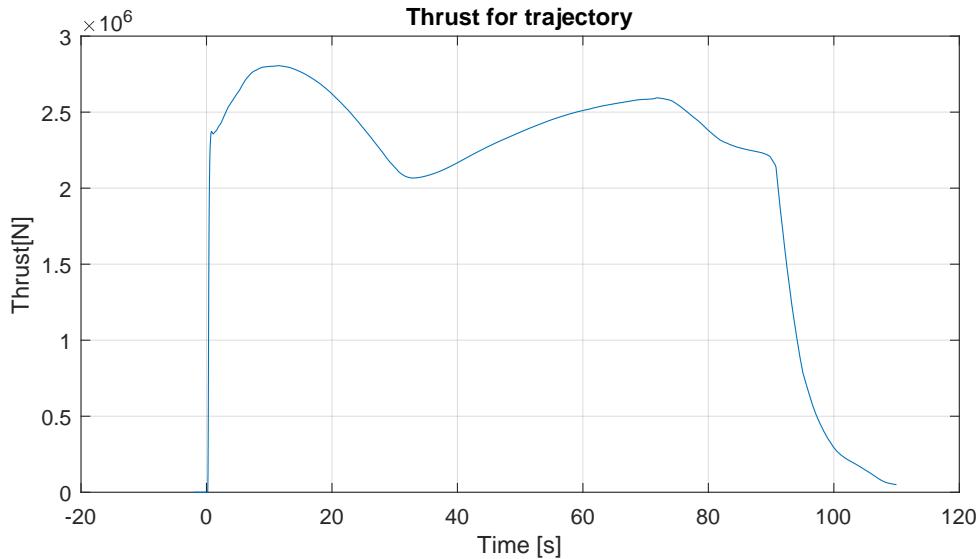


Figure 2.3: The thrust profile for the solid booster on the LV given as reference by ESA

It is taken from a thrust profile of a solid rocket booster. This is simply to have a good reference thrust fitting the LV. Meaning that it has to be a realistic thrust level. Since this controller will not be implemented on a real rocket, the thrust level should be something that would make sense on a real rocket. Since the reference measurements and the thrust is from the same rocket, it will be a good fit. This could be changed for a different main engine system such as liquid fuel engines or similar that could be throttled or restarted for landing. However, for the attitude control for the ascent this will not be important.

2.1.3 Landing trajectory

The landing trajectory in this project will be considered as using closed loop guidance as it is very important that the rocket velocity is zero or is within a margin that the rocket can handle when touching down. It will also have to land within the designated landing area. For a drone ship like the one used by SpaceX for the Falcon booster, it will be less than the 50 meters by 50 meters of the ship. With the dispersion from ascent this problem can be viewed as landing the booster with dispersed initial conditions such as initial position and velocity relative to the landing pad.

2.2 The Vega Launch Vehicle

For this project, the Vega LV seen on figure 2.4 will be used as a reference. It will be used as most of the parameters of the LV are publicly available, and ESA can provide the remaining necessary parameters. The Vega LV has four stages. The first three stages are using solid fuel. The last is a liquid fuel stage that can be restarted up to four times. The Vega LV is designed to carry small payloads up to 2500 kg for missions in Low Earth and Polar orbits [14]. The reference Vega mission is to carry a 1500 kg payload to polar orbit of 700 km altitude. The parameters for the four stages can be seen on table 2.1. As this project will mainly focus on ascent and landing of the booster, the last 3 stages will mainly



Figure 2.4: Overview of the stages of the Vega Launch Vehicle[5]

be used to calculate weight and inertia of the LV. As previously stated, this LV will be

Stage	Stage 1: P80	Stage 2: Zefiro 23	Stage 3: Zefiro 9	Stage4: AVUM
Height	11.20 m	8.39 m	4.12 m	2.04 m
Diameter	3 m	1.9 m	1.9 m	2.18 m
Propellant type	solid	solid	solid	liquid
Propellant mass	87 710 kg	23 814 kg	10 567 kg	577 kg
Total mass	96 243 kg	26 300 kg	12 000 kg	688 kg
Max Thrust	3 015 kN	1 120 kN	317 kN	2.45 kN
Burn time	109.9 s	77.1 s	119.6 s	612.5 s
Specific impulse	280 s	287.5 s	295.5 s	314.6 s

Table 2.1: Parameters for the stages of the Vega LV[14]

used as a reference. However, for this project there will be modifications as per request of ESA and to make it possible to land the vehicle.

The first change is that it will be modelled with multiple engines instead of the one on the reference VEGA LV. This is to control the rotation of the booster with the Thrust Vector Control (TVC), which is not possible with just one engine. For simplicity and for the axial symmetry it will be modelled with 4 engines. The 4 engines will have a similar total thrust to the one on the Vega LV.

Additionally, these engines will have to be reignitable and throttleable to some degree to be able to control during descent and with feedback achieve zero velocity on touchdown. The ascent and landing will be looked at separately and the engines will not be the same. For ascent, the engines will use the thrust profile of the original P80 as it was used to calculate the trajectory, and the landing will require throttling.

2.3 Definition of Variables and Parameters

This section will briefly describe the different parameters in relation to the rocket. It will mainly be described as different reference frames which can be seen on figures 2.5 and 2.6. These will be used to describe the dynamics of the rigid body.

The **Rocket-Frame** is used for defining the relative positions on the rigid rocket model and is static relative to all parts of the rocket. The **CG** is the origin of the body-frame to which all forces and moments are converted for the dynamics. The CG is also be used as the position of the LV in external frames. **CP** is be used as the origin of the velocity frame in which the relative air velocity has a constant direction from along the x-axis. This

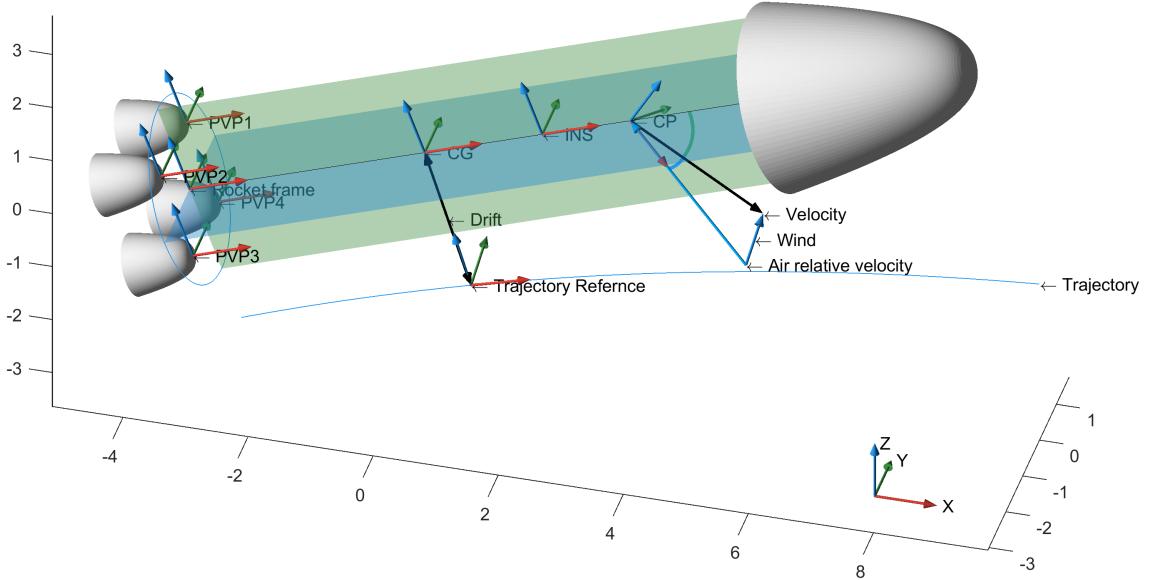


Figure 2.5: Parameters of the LV model seen from a 3D perspective

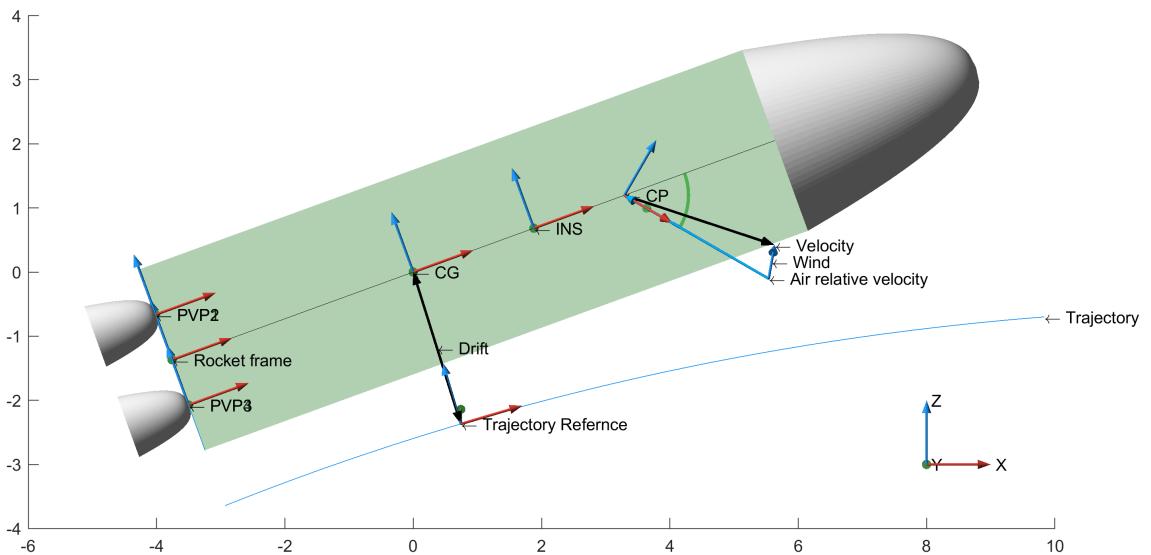


Figure 2.6: Parameters of the LV model seen perpendicular to the Y-plane

makes it easier to calculate the forces in the velocity frame and then afterwards convert to body frame where they are included in the dynamics. The point **INS** is the origin of the measurement-frame which is aligned with the body-frame when the rocket is rigid. The measurement frame describes the frame from where the sensors measure the states of the rocket. The relative movement of the body-frame and the measurement frame is used to model what the sensor will measure. The **Trajectory Reference** is the reference-frame that is used as the reference in which the model is linearized. When the body-frame aligns with the reference frame the error will be zero. Finally, **PVP** points are the origins of the engine-frames. In the engine-frames the thrust has a constant direction along the x-axis. The thrust magnitude can be changed, and the frame can be rotated for control.

2.4 Reference frames

This section will briefly explain the general relationship between arbitrary reference-frames and then afterwards define reference-frames used in this project and their most relevant conversions.

2.4.1 General Description of Change of Reference-Frame

The two quantities that describe the relationship of two reference frames is their relative origin and their relative orientation. A change of reference for a point p_1 from frame B to frame A can be described as:

$$p_{1,A} = C_{AB} \cdot p_{1,B} + p_{B_0,A} = C_{AB} \cdot (p_{1,B} - p_{A_0,B}) \quad (2.1)$$

Where $p_{1,A}$ is the point p_1 in frame A . $p_{1,B}$ is the point p_1 in frame B . C_{AB} is the Direction Cosine Matrix (DCM) describing the rotation from frame B to A . $p_{B_0,A}$ is the origin of frame B in frame A and $p_{A_0,B}$ is the origin of frame A in frame B . Since a vector is just describing a direction and a magnitude, its origin is not needed when changing frames. Thus, a vector v_1 can change from frame B to A as:

$$v_{1,A} = C_{AB} \cdot v_{1,B} \quad (2.2)$$

Where $v_{1,A}$ is the vector v_1 in frame A . $v_{1,B}$ is the vector v_1 in frame B .

With an additional frame C when the transformation from frame C to frame B is known the following transformation can be used from frame C to frame A :

$$p_{1,A} = C_{AB} \cdot (C_{BC} \cdot (p_{1,C} - p_{B_0,C}) - p_{A_0,B}) \quad (2.3)$$

And for a vector from frame C to A can be expressed as:

$$v_{1,A} = C_{AB} \cdot C_{BC} \cdot v_{1,C} \quad (2.4)$$

From this a rotation directly from frame C to A can be expressed as:

$$C_{AC} = C_{AB} \cdot C_{BC} \quad (2.5)$$

It should also be noted that for DCM the transpose is equal to the inverse of the same matrix, which is also equal to the opposite rotation:

$$C_{AB} = C_{BA}^T = C_{BA}^{-1} \quad (2.6)$$

This is because it is an orthonormal basis, and is therefore orthogonal. Which also leads to the following relationship:

$$C_{AB} C_{AB}^T = C_{AB} C_{AB}^{-1} = C_{AB} C_{BA} = I \quad (2.7)$$

Where I is the 3-by-3 identity matrix. These are the equations and the notation that is used in this text when describing frame transformations.

2.4.2 Basic rotations

Three basic rotations will be introduced here. They are the rotation matrices describing a single rotation around a single axis. They will be combined and used to create most of the rotation used in this text. The base rotations:

$$R_x(\theta_x) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_x) & -\sin(\theta_x) \\ 0 & \sin(\theta_x) & \cos(\theta_x) \end{pmatrix} \quad (2.8)$$

$$R_y(\theta_y) = \begin{pmatrix} \cos(\theta_y) & 0 & \sin(\theta_y) \\ 0 & 1 & 0 \\ -\sin(\theta_y) & 0 & \cos(\theta_y) \end{pmatrix} \quad (2.9)$$

$$R_z(\theta_z) = \begin{pmatrix} \cos(\theta_z) & -\sin(\theta_z) & 0 \\ \sin(\theta_z) & \cos(\theta_z) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2.10)$$

These are the functions used when referring to R_x , R_y and R_z . They rotate around the respective axis in the positive mathematical rotational direction.

2.4.3 Earth Reference Frames

When using Earth as a reference frame there are two primary frames used. The Earth-Centered Inertial coordinate frame (ECI), and the Earth-Centered Earth-Fixed (ECEF) coordinate frame. As the names suggest they both have their origin at the center of Earth. The ECI frame is an inertial frame. It is fixed relative to the stars and is useful when describing the motion of satellites, celestial bodies or spacecrafts in orbit. ECEF is a non-inertial frame. However, it is fixed to the rotation of Earth and therefore useful when describing objects on Earth's surface or in the atmosphere. This also means that you can convert directly between ECEF and Latitude, Longitude and Altitude (LLA).

Both ECEF and ECI have the z-axis aligned with the north pole. ECIs x-axis is aligned with the direction of the vernal equinox which is a fixed direction relative to the stars and the solar system. The ECEF x-axis aligned with the line that goes from the center and out through point corresponding to latitude 0° and longitude 0° . Both ECI and ECEF's y-axis complete the right-handed coordinate system. The relationship between the two frames can be described as:

$$C_{E_I}(t) = R_z(t \cdot \omega_E) C_{E_I}(0) \quad (2.11)$$

where ω_E is the angular velocity of the earth. $C_{E_I}(0)$ is the DCM for t=0

2.4.4 Launch Pad and Landing Pad Reference Frame

The reference frames for launch and landing have origin in the center of the launchpad. Their orientation is aligned with the trajectory. A simulation would use this frame to record position and velocity. The position in Launch frame can be calculated from ECI frame as:

$$r_{1,L} = C_{L_E}(C_{E_I}(t)r_{1,I} - r_{L_0,E}) \quad (2.12)$$

The launchpad origin $r_{L_0,E}$ can be converted from LLA coordinates. The simulated mission is flown from the LLA: [5.2°, 52.8°, 0 m]. C_{LE} is the frame with the z-plane tangent to the ground and the x-axis aligned with the trajectory.

2.4.5 Trajectory Reference Frame

The trajectory reference frame is reference for the LV. When the body frame is aligned with this frame the error for the controller will be zero. The orientation of the reference frame is well defined by the trajectory given by ESA and described in section 2.1.2. The yaw and roll is set to zero all along the trajectory, and the pitch is following a gravity turn. The position of the origin is not well defined. It can be described as the path that the LV would follow if it had the reference orientation, reference thrust, and no disturbances. This is not measurable under flight. The error will therefore be approximated as the integral of the velocity perpendicular to the planned flight direction. The velocity will mainly be an integration of an accelerometer. Therefore, it will be a double integration of a measurement prone to noise and offset. This can be somewhat rejected by filtering with some other measurements such as GPS if available. However, this is one of the reasons it is not a high priority control objective. Nonetheless, it will be taken into consideration during modelling of the LV dynamics.

The DCM from Launch pad frame to trajectory frame is given by:

$$C_{T_L}(t) = R_y(\text{pitch}(t)) \quad (2.13)$$

2.4.6 Body Frame

The body reference frame has the origin in the CG and is useful for describing the forces and moments acting on the LV. Because the LV will rotate about CG the moments are easier to define in the frame. Additionally, because of the LVs axial symmetry in the body frame, the moment of inertia is also easier to describe. And because the frame rotates with the LV the inertia is also not changing relative to orientation in the body frame. Which makes it a lot easier to work with. The position of the origin and the orientation will usually be described in relation to the launchpad or landing pad in their respective frames. Or in relation to the trajectory as described in the previous section. The orientation will be described based on three rotations with the angles referred to as pitch yaw and roll. Thus, the DCM from body frame to Launch pad frame can be described as:

$$C_{B_L} = R_z(\text{yaw})R_y(\text{pitch})R_x(\text{roll}) \quad (2.14)$$

2.4.7 Relative Air Velocity Frame

The relative air velocity frame is used when calculating the aerodynamic forces. It has the center in CP. The relative air velocity is the airs relative velocity to the LV. This is a combination of the velocity of the LV and the wind. With the model used in this project all aerodynamic forces are applied at CP. The wind at CP from the rotation of the LV around the CG will be added to the total wind. This is a simplification and is not perfect as the wind induced load is not uniformly distributed[15]. Thus the rotation will have a different average load point from the wind. However, this assumption will be used for this simplified model. When the forces are calculated in this frame the lift and drag can more

easily be calculated. The drag will be expressed along the x-axis in the velocity frame. The x-axis will be parallel with the velocity vector. The lift will be defined perpendicular to the velocity vector; that is, along the y- and z-axis. Ideally when using symmetric LV the velocity frame would be defined using aeroballistic wind coordinates [16]. Aeroballistic wind coordinates use the effective angle of attack α' and the aerodynamic load angle ϕ' . However, the dynamics will have to be linearized around $\alpha' = 0$. This would be a problem as the ϕ' would be undefined at this point [16,p.79]. Therefore, the cartesian incidence angles α and β will be used for the linearization when linearized around zero AoA. Thus, the transformation from body to velocity frame is as follows:

$$C_{V_B} = R_z(\beta)R_y(\alpha) \quad (2.15)$$

And the inverse that is used to get it to body frame from velocity frame:

$$C_{B_V} = C_{V_B}^T \quad (2.16)$$

2.4.8 Nozzle Frame

To keep the method consistent for the model, the forces from the engines are defined in the nozzle frames. The force from the thrust of the engines are applied along the x-axis in the nozzle frames. The force is also applied in the origin of the frames which is placed at the pivot points of the engines. These are the PVP points seen in figure 2.5. The actuation based on the control commands β_y and β_z will apply a rotation to these frames defined as:

$$C_{B_N} = R_z(\beta_z)R_y(\beta_y) \quad (2.17)$$

2.5 Dynamics

This section will describe the most important dynamics that affect the LV during flight. The dynamics will here be explained in their respective frames. This section will give an overview of the dynamics that will later be used to build a model.

2.5.1 Aerodynamic Forces

The aerodynamic forces are some of the biggest forces acting on the LV outside the main engines. But while the main engines are controllable, the aerodynamic forces are not directly controllable. They are usually the main cause of instability as the center of pressure (CP) generally in front of the CG. The aerodynamic forces are also very hard to model directly. Therefore, a method to approximate an aerodynamic model has to be used. For this project, the model is based on measurements or simulations. The result of these estimates is the lift and drag coefficients, C_L and C_D which have been provided by ESA. With good approximations of these, a good model for a fully symmetric LV can be achieved. Then with the dynamic pressure Q_d :

$$Q_d = \frac{1}{2}\rho\|V_{\text{total}}\|^2 \quad (2.18)$$

Where ρ is the density of the air. V_{total} is the total air relative velocity. The force in velocity frame can then be described as:

$$F_{\text{air},V} = -Q_d \cdot S_{\text{ref}} \cdot \begin{pmatrix} C_D(\alpha_{\text{eff}}, M) \\ 0 \\ C_L(\alpha_{\text{eff}}, M) \end{pmatrix} \quad (2.19)$$

Where Q_d is the dynamic pressure. S_{ref} is the relative surface area which is used to scale the forces based on size. C_L and C_D are the lift and drag coefficients that are used to capture the shape of the LV. The coefficients are both dependent on the effective angle of attack which is also used for the aeroballistic coordinate frame. However, this frame is not defined when the angle is zero as mentioned in section 2.4.7. Therefore, the following approximation will be used later when differentiating in exactly zero:

$$F_{\text{air},V} = -Q_d \cdot S_{\text{ref}} \cdot \begin{pmatrix} C_D(\alpha_{\text{aoa}} + \alpha_{\text{sideslip}}, M) \\ -C_L(\alpha_{\text{sideslip}}, M) \\ C_L(\alpha_{\text{aoa}}, M) \end{pmatrix} \quad (2.20)$$

This approximation only holds when differentiating either α_{aoa} or α_{sideslip} one at a time. And it only holds, but is also only necessary, when α_{eff} is exactly zero. Additionally, the lift coefficient should also be zero at this point. Which it will be if full symmetry around the primary axis is assumed. This is the aerodynamic forces expressed in the velocity frame. The coefficients can be seen on figure 2.7

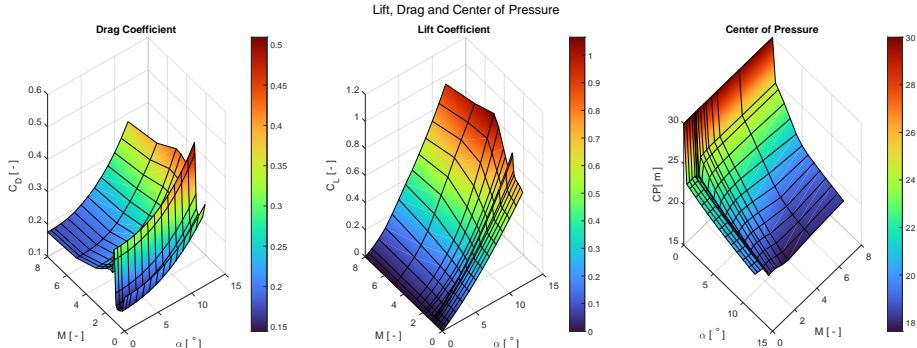


Figure 2.7: Aerodynamic coefficients for Vega LV.

Here it is worth noting that the biggest changes in values is around mach 1. This makes sense as it is when the LV is trans-sonic

2.5.2 Thrust Vector Control Forces

Having already defined the nozzle frame in section 2.4, defining the thrust in the nozzle frame is straight forward. It is simply the thrust along the primary axis and can be described as:

$$F_{\text{TVC},N} = F_{\text{Thrust}} \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad (2.21)$$

Where F_{Thrust} is the magnitude of the thrust. This can either be controlled or follow the thrust profile of the solid rocket booster.

2.5.3 Tail-Wags-Dog Effect

The tail-wags-dog (TWD) effect is from the acceleration of the center of mass of the main engines. This acceleration requires a force, and therefore there will be an equal and opposite force on the LV. In the nozzle frame the TWD force can be expressed as:

$$F_{\text{TWD},N} = P_{\text{CG}_N} \times \ddot{\mathcal{B}}_N \cdot m_N \quad (2.22)$$

Where $F_{\text{TWD},N}$ is the force on the LV in the nozzle frame. P_{CG_N} is the position of the CG of the nozzle, in the nozzle frame. $\ddot{\mathcal{B}}_N$ is the angular acceleration of the nozzle in the nozzle frame. m_N is the mass of the nozzle. Additionally, there is the angular moment required to rotate the nozzle, which also acts equally on the LV. It can be expressed as:

$$M_{\text{TWD},N} = J_N \cdot \ddot{\mathcal{B}}_N \quad (2.23)$$

Where J_N is the inertia of the nozzle. withing $\ddot{\mathcal{B}}_N$ is the acceleration in the nozzle frame.

2.5.4 Gravitational Acceleration

The gravitational acceleration will, for the modelling, be expressed in launch frame as:

$$\dot{V}_{g,L} = \dot{v}_g \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad (2.24)$$

Where \dot{v}_g is the magnitude of the gravitational acceleration. However, for a more accurate representation such as a nonlinear simulation, the roundness of the earth can be taken into consideration. As explained in section 2.1.2 it should be noted that this acceleration will be cancelled by the Coriolis acceleration during the gravity turn. This will however have to be taken into consideration when landing.

2.5.5 Coriolis acceleration

The Coriolis acceleration is the apparent acceleration in a frame that is moving and rotating in an inertial frame. The Coriolis acceleration in the body frame relative to the approximately inertial launch pad frame. Can be expressed as:

$$\dot{V}_{\text{Coriolis},B} = 2V \times \Omega_{\text{LV},B} \quad (2.25)$$

$\dot{V}_{\text{Coriolis},B}$ is the acceleration from Coriolis effect in the body frame. V is the velocity vector. $\Omega_{\text{LV},B}$ is the angular velocity of the LV in the body frame. This will not be considered while doing the gravity turn as it will cancel with gravity during the turn.

2.5.6 Moments From Forces

The moments on the LV from the forces can be calculated, based on the force and the location of the force relative to the point of rotation. Which for the LV is the CG. It can thereby be described as:

$$M_B = (P_{\text{Force},B} - P_{\text{CG},B}) \times F_B \quad (2.26)$$

Where M_B is the moment in the body frame. $P_{\text{Force},B}$ is the point where the force is applied in the body frame. $P_{\text{CG},B}$ is the center of gravity in the body frame. F_B is the force vector in the body frame. This will be used for all the forces when doing the modeling.

2.5.7 Euler Moments

When a reference frame is rotating with respect to the inertial frame, there will seem to be forces acting on objects in the frame. The Euler equations of rotating rigid body describe such moments. [17]

$$M_{\text{euler},B} = J \times \dot{\Omega}_B + \Omega_B \times J\Omega_B \quad (2.27)$$

$M_{\text{euler},B}$ is the euler moment in the body frame. J is the inertia matrix of the LV. $\dot{\Omega}_B$ is the angular acceleration of the LV. Ω_B is the angular velocity of the LV. The first term is resistance to acceleration. Second term is resistance to rotate when already rotating in another direction. When the rotations are described in the body frame these will have to be included in the dynamic equations. And when the reference frame rotates these forces should be added as a disturbance.

2.6 Disturbance, Noise and Parameter Uncertainties

This section will describe the main disturbances, noise and uncertainties relevant for this project.

2.6.1 Wind Disturbance

One of the main factors when designing a controller for an LV is the effect of the wind disturbance. The wind can be especially high impact for the first 20 km of altitude, where it is usually modelled[18]. The model used in this source will be used with the data as it matches the mission that this project is concerned with. The data can also be found in [19]. And the model has been used successfully in other places such as [20]. The model is a filter for coloring white noise. It is a filter that will describe the magnitude and frequency content of the wind disturbance. The Dryden filter is on the form [18]:

$$G_w(s) = \frac{v_w(s)}{n_w(s)} = \frac{\sqrt{\frac{2}{\pi} \frac{\|V\| - v_{wp}(h)}{L_h}} \sigma_h^2}{s + \frac{\|V\| - v_{wp}(h)}{L_h}} \quad (2.28)$$

Where $v_w(s)$ is the velocity of the modelled wind. $n_w(s)$ is the white noise used as input. $v_{wp}(h)$ is the wind profile as a function of height. L_h is the turbulence length scale. σ_h is the standard deviation. $\|V\|$ is the speed of the LV. These parameters will just be used for the model and not further explored in this project to limit the scope. However, this model can be added to the synthesis to better capture the frequency content of the disturbance.

2.6.2 Disturbance from Engine Offset

For this project one of the interests is the effect of uncertainties when multiple engines are used. When the thrust varies it will have a direct effect on the dynamic model, as it will have a direct effect of the actuation from the control input. But this effect will be modelled with the other parameter uncertainties as explained in section 5.1.1. However,

when building the model it is assumed that the symmetric placements of the engines will cancel out the moment on the rocket that they individually produce. As the four main engines each have associated uncertainty. They can end up having different thrust levels which will then result in a total moment that will have to be modelled. This will have to be added as an input to the system. To find the magnitude of this moment, the requirement for thrust difference within an engine cluster will be used. This is defined for the transient periods of thrust. That is, when the engines are turning on, and turning off. As this is where the difference is the largest. Therefore the disturbance should be less during the flight than defined for the transient part. For startup the requirement is that the thrust difference is less than 1% and for engine cut off it should be less than 5% based on numbers from ESA. For this project the assumption is then based on these numbers. The assumption is therefore that the thrust difference in the cluster is at worst 1% during flight. The uncertainty will be included as an addition to the magnitude of the force for the individual engines:

$$F_{\text{Thrust}} = F_{\text{Thrust,ref}}(1 + \epsilon) \quad (2.29)$$

F_{Thrust} is the thrust from equation 2.21. $F_{\text{Thrust,ref}}$ will for the ascent, be the thrust on the figure 2.3. And ϵ proportional deviation from nominal thrust.

This moment will enter the system like the other disturbance from the wind.

2.6.3 Noise

The noise introduced in the system is from the sensors. The magnitude of the noise will simply be modelled based on the numbers from [18,p.57]. Which is a PhD in collaboration with ESA about the Vega LV. A further investigation into the noise will not be done for this project. From this, the numbers used as the standard deviation of noise from the INS is 0.02 deg and 0.1 deg/s for the attitude measurement. For the drift measurement it is 0.01 m and 0.001 m/s.

2.6.4 Delay

The delay is from the internal processing time in the different components. The data is taken from [18,p.57] which again uses the same LV model. The following delays are then taken into consideration. The on-board computer adds 12 ms delay, the sensors add 12 ms delay and the TVC actuators add 15 ms delay, which adds to a total of 39 ms second delay. This is with an uncertainty of 10 ms.

2.6.5 Inertial Navigation System

The Inertial Navigation System (INS) is the system that will be providing the measurements to the controller. It uses accelerometers and gyroscopes. The gyroscopes will read the orientation of the body frame directly. However, the accelerometers will read additional translation from the rotation of the LV as it is not placed at the center of rotation which is CG. The additional movement of the INS from the rotation of the LV can be captured by the frame transformation described in 2.4 as:

$$P_{\text{INS},T} = C_{T_B}(P_{\text{INS}} - P_{\text{CG}}) + P_{\text{CG},T} \quad (2.30)$$

The additional velocity can be described as:

$$V_{\text{INS},T} = C_{T_B}(\Omega_B \times (P_{\text{INS}} - P_{\text{CG}})) + V_{\text{CG},T} \quad (2.31)$$

2.7 Data Used for Parameter Variations

The data used for this project, as the parameter varies over time, is from a different project where a nonlinear simulation was implemented[13]. While that simulation was without a lot of the dynamics that make the rocket more difficult to control such as bending modes and disturbance. It should still be a good representation of a nominal trajectory. The data from that simulation has been compared to another paper which also had simulation results from a nominal flight of the Vega LV[21]. After comparing the data, it was concluded that the two were close enough that the data from the implemented simulation could be used. The data can be seen in appendix A.1.

Objectives and Requirements 3

3.1 Objectives

This section describes the objectives for this project in a prioritized manner with argumentation for the objectives and their priority.

3.1.1 Motivational Final Objective

The motivational final objective for this project is to develop a controller that can control a simulated LV similar to that of the VEGA LV during ascent and landing. The controller should be designed for a model of the Vega LV that has multiple main engines. The controller should be designed to take noise, disturbances, parameter variation and uncertainties into consideration.

3.1.2 Prioritization of Objectives

It was decided to prioritize the different steps towards reaching the final motivational objective as it is a complex objective and the time for this project is limited. The following list is a prioritised list of the partial objectives of the project:

1. Controller that stabilize attitude of rigid body model during gravity turn part of ascent
2. Make the control model more detailed
 - a) Include Disturbance
 - b) Include Noise
 - c) Include Delay model
 - d) Include Actuation model
 - e) Include Impact of Multiple Main Engines
 - f) Include Parameter Variation
 - g) Include Uncertainties
3. Controller that stabilize attitude for the full ascent trajectory
4. Controller that stabilize attitude during descent
5. Guidance for the descent using MPC

The first priority **1** is because this part of the trajectory seemed like the best first step and there was a wide range of reference material on a controller for this part. Point **2** is to prioritize covering more control theory, and to make the problem closer to designing a controller for a real rocket, as doing simple control for all stages first might limit the complexity of the control individual controllers. The points **3** and **4** seems like the natural

next steps. Point **5** relies on the implementation of point **4** and is an entirely new topic, and is therefore last.

During the preliminary studies it was decided, in agreement with supervisors, that because of the huge scope of the project, the focus of the thesis should be to work on the first two objectives. Namely, to make a controller for the gravity turn part of the trajectory that includes all the sub objectives of point **2**. With this in mind the project work from here on will be limited to this scope. With these objectives in mind the next section will discuss the requirements for the controller used to verify the success of the objectives.

3.2 Requirements

This section will discuss the requirements for the controllers that will be designed. The requirements will be explained and there will be defined **specific goals** for each in bold. Some of the requirements have opposing goals and thus a balance has to be found.

3.2.1 Stability

The system should be stable. The stability of the system with the controller is the most important requirement, as any other requirement can not be fulfilled if the system is unstable. The main way this requirement will be verified is by the stability of the closed loop system. That is if all the eigenvalues of the system are in the left half plane and simulations of the system. Additionally the controller should by itself be stable. The stability of the system can also be gauged by the classical Phase Margin (PM) and Gain Margin (GM) by looking at some specific signal paths, but can not be directly specified for the system as a whole. They will be indirectly used for tuning of the weights that will be used to essentially define the performance requirements for the system. The Gain and phase margins used as a target will be a **GM of 6 dB and PM of 60°**.

3.2.2 Performance

The performance for this project will be included as weights in the synthesis process of the controllers, see section 5.1.3. The controller synthesis will result in a performance indicator γ . **If $\gamma \leq 1$ the requirements will be met** based on the defined model with weights. Additionally the requirements will be **tested in simulations**.

3.2.3 Attitude tracking

The controller should be able to follow the attitude from the guidance with as high a bandwidth as possible without coupling with the bending modes. The system should **converge to zero tracking error in steady state with no disturbance**. The bandwidth should also be high enough for tracking but low enough to not have flexible coupling with bending modes. This will be indirectly enforced by the error and stability requirements

3.2.4 Load vs. Drift

The aerodynamic loads on the LV can be huge and result in Loss Of Vehicle (LOV). And the drift of the vehicle during the open loop guidance should be small enough that it can be

corrected by the closed loop control later in flight. For the drift the requirements are based on other projects making control for a similar mission[18]. The **maximum drift should be less than 500 m** and **drift rate less than 15 m/s** during the launch trajectory. For the load, the load parameter will be used to define the requirement.

Load Performance Parameter

The requirement for load on the LV is often described in the literature[20] by an envelope of a performance parameter $Q\alpha$ given as:

$$Q\alpha = Q_d \alpha_{\text{eff}} \quad (3.1)$$

This performance parameter is simply given by multiplying the dynamic pressure Q_d with the effective angle of attack α_{eff} which the name suggests. An envelope is then specified for a trajectory as a function of mach number. It is based on what is estimated to be the critical value that would cause an LOV. This is useful as both these parameters can be taken into account when building the model and added as an output that is used when synthesising the controller. As the envelope was not available, and for simplicity, the Q_α requirement have been defined as a constant based on the value at max-Q. Therefore the **maximum value should be $2.6 \times 10^5 \text{ Pa}^\circ$** .

3.2.5 Actuation

The actuators connected to the main engines can only tilt the main engines by a certain amount. And it is desired that the total actuation is limited. The engine used as a reference on the Vega LV can have an **actuation angle of up to 6.5°** which will be the requirement for the maximum actuation deflection angle. Additionally, a bound of the **total integrated actuation is set to 250°** based on [18].

3.2.6 Disturbance rejection

All the other requirements should be met while being disturbed by wind, bending modes, and uncertainties. The wind will be defined based on the Dryden model in section 2.6.1. The bending modes will be included in the model for synthesis 4.7. The uncertainties will also be included in the model using MATLAB and the LFT model, this will be explained in section 5.1.1.

Modeling of LV 4

This chapter will explain the steps to make a model of the LV. The reference frames have been explained in section 2.4. And the forces and moments have been explained in section 2.5. The linear rigid body model has been derived using the following steps that will be elaborated in this chapter.

1. Derive the 3D nonlinear symbolic dynamic equations for the rigid body.
2. Find the symbolic Jacobian of the dynamic equations.
3. Insert values for the operating point to get state space for the 3D model.
4. Make the 2D model from the 3D model by picking the relevant entries.

This results in a rigid body model of the LV that is used for control. The bending mode model is made based on the model derived in [22]. The first section will define all the relevant parameters.

4.1 Parameter Overview

This section will briefly define the parameters and variables that will be used
Angles Θ and angular velocities Ω of the LV:

$$\Theta = \begin{pmatrix} \theta_x \\ \theta_y \\ \theta_z \end{pmatrix}, \Omega = \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \quad (4.1)$$

Position vector R and Velocity vector V of the LV:

$$R = \begin{pmatrix} r_x \\ r_y \\ r_z \end{pmatrix}, V = \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} \quad (4.2)$$

Actuation angles \mathcal{B} and actuation acceleration $\ddot{\mathcal{B}}$ for the main engines.

$$\ddot{\mathcal{B}} = \begin{pmatrix} \ddot{\beta}_x \\ \ddot{\beta}_y \\ \ddot{\beta}_z \end{pmatrix}, \mathcal{B} = \begin{pmatrix} \beta_x \\ \beta_y \\ \beta_z \end{pmatrix} \quad (4.3)$$

Here it is noted that there will be no rotation around the x-axis of the actuators. Thus, the β_x and $\ddot{\beta}_x$ will not be used. This means that they will practically be set to zero during modelling, but they are included here for completeness.

The disturbance from the uncertainty in thrust amongst the engines in the cluster will be

called ϵ_n , where n denotes for which engine the uncertainty models.

The Wind velocity vector W is:

$$W = \begin{pmatrix} w_x \\ w_y \\ w_z \end{pmatrix} \quad (4.4)$$

The point locations P_{CG} , P_{INS} and P_{CP} :

$$P_{CG} = \begin{pmatrix} x_{CG} \\ y_{CG} \\ z_{CG} \end{pmatrix}, P_{INS} = \begin{pmatrix} x_{INS} \\ y_{INS} \\ z_{INS} \end{pmatrix}, P_{CP} = \begin{pmatrix} x_{CP} \\ y_{CP} \\ z_{CP} \end{pmatrix} \quad (4.5)$$

For the engines there is the locations P_{PVP} and P_{CG_N} for each engine:

$$P_{PVP} = \begin{pmatrix} x_{PVP} \\ y_{PVP} \\ z_{PVP} \end{pmatrix}, P_{CG_N} = \begin{pmatrix} x_{CG_N} \\ y_{CG_N} \\ z_{CG_N} \end{pmatrix} \quad (4.6)$$

The equation for a single engine with arbitrary position will be modelled. This equation can then be duplicated for each engine.

4.2 Full Dynamic Equations

This section will combine all the dynamic equations that have been defined so far and will be used for the linearization. The final equations should describe the acceleration of the states as a function of the other states, that is:

$$\dot{x} = f(x, u) \quad (4.7)$$

$$y = h(x, u) \quad (4.8)$$

Where x is the state vector. y is the output vector. f is the function that describes the rate of changes of the states \dot{x} . u is the external input. h is the output function.

Where x , u and y for the full rigid body system with x_{RB} , u_{RB} and y_{RB} are:

$$x_{RB} = \begin{pmatrix} \Theta \\ \Omega \\ R \\ V \end{pmatrix}, u_{RB} = \begin{pmatrix} \mathcal{B} \\ \ddot{\mathcal{B}} \\ W \\ \epsilon_n \end{pmatrix}, y_{RB} = \begin{pmatrix} Q\alpha \\ \Theta_{INS} \\ \Omega_{INS} \\ R_{INS} \\ V_{INS} \end{pmatrix} \quad (4.9)$$

Where the subscript INS means the measurement at the INS.

This, after a linearization, will result in a linear system on the form:

$$\dot{x} = Ax + Bu \quad (4.10)$$

And the measurements can equally be written as:

$$y = Cx + Du \quad (4.11)$$

For this project, the process of linearization was broken down into a number of smaller problems. As an example, the A matrix was partitioned in the following way:

$$A = \begin{pmatrix} A_{\dot{\Theta}}^{\Theta} & A_{\dot{\Theta}}^{\Omega} & A_{\dot{\Theta}}^R & A_{\dot{\Theta}}^V \\ A_{\dot{\Omega}}^{\Theta} & A_{\dot{\Omega}}^{\Omega} & A_{\dot{\Omega}}^R & A_{\dot{\Omega}}^V \\ A_{\dot{R}}^{\Theta} & A_{\dot{R}}^{\Omega} & A_{\dot{R}}^R & A_{\dot{R}}^V \\ A_{\dot{V}}^{\Theta} & A_{\dot{V}}^{\Omega} & A_{\dot{V}}^R & A_{\dot{V}}^V \end{pmatrix} \quad (4.12)$$

Where notation A_b^a is the A sub matrix that describes the linear effect from a on b . And A has the dimensions 12×12 . This matrix can be further simplified with an understanding of the physical system and the dynamic equations to:

$$A = \begin{pmatrix} 0 & I & 0 & 0 \\ A_{\dot{\Omega}}^{\Theta} & A_{\dot{\Omega}}^{\Omega} & 0 & A_{\dot{\Omega}}^V \\ 0 & 0 & 0 & I \\ A_{\dot{V}}^{\Theta} & A_{\dot{V}}^{\Omega} & 0 & A_{\dot{V}}^V \end{pmatrix} \quad (4.13)$$

Where 0 is a zero matrix and I is an identity matrix both of dimension 3×3 . This means that only the equation for linear and angular acceleration \dot{V} and $\dot{\Omega}$ will be needed. Similarly, the B , B_d , C , D and D_d matrices can be written as:

$$B = \begin{pmatrix} 0 & 0 \\ B_{\dot{\Omega}}^B & B_{\dot{V}}^B \\ 0 & 0 \\ B_{\dot{V}}^B & B_{\dot{V}}^B \end{pmatrix}, B_d = \begin{pmatrix} 0 & 0 \\ B_{\dot{\Omega}}^W & B_{\dot{\Omega}}^{\epsilon} \\ 0 & 0 \\ B_{\dot{V}}^W & 0 \end{pmatrix} \quad (4.14)$$

$$C = \begin{pmatrix} C_{Q\alpha}^{\Theta} & C_{Q\alpha}^{\Omega} & 0 & C_{Q\alpha}^V \\ I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ C_R^{\Theta} & 0 & I & 0 \\ 0 & C_V^{\Omega} & 0 & I \end{pmatrix} \quad (4.15)$$

$$D = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}, D_d = \begin{pmatrix} D_{Q\alpha}^W & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \quad (4.16)$$

The total acceleration in body frame is then given by:

$$\dot{V}_B = \dot{V}_{g,B} + \dot{V}_{\text{coriolis},B} + \frac{F_{\text{TWD},B} + F_{\text{air},B} + F_{\text{TVC},B}}{m} \quad (4.17)$$

This acceleration can be transferred to trajectory frame \dot{V}_T . While most examples using a similar model is done in body frame such as [18][23], this project will model the position and velocity of the LV in trajectory frame. This is because these states R_T and V_T have a more intuitive interpretation than R_B and V_B . R_T and V_T are the speed and distance perpendicular to the planned trajectory. And while R_B and V_B are more straightforward

to derive in body frame, they are the just sideways motion of the LV in the body reference frame, and the integration. These, R_B and V_B , does not have obvious interpretations that can be used to set requirements. While limiting the states in the body frame can be an indirect way to limit lateral load, the lateral load can already be fully accounted for by using the angle and the lateral movement in the trajectory frame. Therefore, the acceleration used for this model can be described as:

$$\dot{V}_T = C_{T_B} \dot{V}_B \quad (4.18)$$

And the angular acceleration can be described as:

$$\dot{\Omega} = J^{-1}(M_{\text{TWD},B} + M_{\text{TVC},B} + M_{\text{air},B} + M_{\text{euler},B}) \quad (4.19)$$

Which will only be described in body frame as it would otherwise be difficult to model. But this means that the Euler moments will have to be included as the frame is changing.

With these we have the necessary nonlinear equations. To find the A matrix in equation 4.13 and the B matrix in equation 4.14, we have the nonlinear equations 4.18 and 4.19. For the output matrices C D and D_d in equation 4.15 and 4.16 we have the nonlinear equations 2.30, 2.31 and 3.1.

4.3 Dynamic Equations as a Function of states

The dynamic equations have thus far been defined in terms of the variables that directly affect the system. This is already mostly states. However, the aerodynamics is still dependent on the AOA. None of the states directly represent the angle of attack. To have it represented by states the angle of attack can be written as a function of the states. It is a combination of the angle relative to the nominal trajectory where AOA is assumed zero, plus any relative sideways wind. The total relative wind velocity is given as:

$$V_{\text{total}} = V_{\omega} + W + V \quad (4.20)$$

where V_{ω} is the wind from the rotation of the rocket. Which is given at CP as:

$$V_{\omega} = \Omega \times (P_{\text{CP}} - P_{\text{CG}}) \quad (4.21)$$

The angle of the incoming wind is then expressed as at CP:

$$\alpha_{\text{aoa},v} = \arctan\left(\frac{v_{\text{total},z}}{v_{\text{total},x}}\right) \quad (4.22)$$

$$\alpha_{\text{sideslip},v} = \arcsin\left(\frac{v_{\text{total},y}}{\|V_{\text{total}}\|}\right) \quad (4.23)$$

If we define the states Θ as the error between the reference and the body frame. And we assume that the trajectory follows a gravity turn with AOA of zero, we have the following expression for the total AOA and sideslip that is used in the dynamics.

$$\alpha_{\text{aoa,total}} = \alpha_{\text{aoa},v} + \theta_y \quad (4.24)$$

$$\alpha_{\text{sideslip,total}} = \alpha_{\text{sideslip},v} + \theta_z \quad (4.25)$$

These will be substituted into the dynamics before the next step.

4.4 Jacobian of Dynamic Equations

For the linearization, the Jacobian method will be used. That is, using the Jacobian of the dynamic equations as the linearized dynamics. The Jacobian can be written on the form:

$$\mathbf{J}_f(x_1, \dots, x_n) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \dots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \quad (4.26)$$

Where $\mathbf{J}_f(x_1, \dots, x_n)$ is the Jacobian of the vector function f with respect to the values of x . Where f would be the dynamic equations of the system and x the values about which the system is linearized. The system will in general have to be linearized at an equilibrium. This will be elaborated in the next section.

The Jacobian was first found symbolically for the relevant equations. The dynamic equations that are linearized as described in section 4.2, is for \dot{V} in equation 4.19 and $\dot{\Omega}$ in equation 4.18. And similarly for measurements we have the equation 2.30 and 2.31. And for $Q\alpha$ the equation 3.1. This will give the following Jacobians:

$$\mathbf{J}_{\dot{\Omega}}(x_{\text{RB}}, u_{\text{RB}}), \mathbf{J}_{\dot{V}}(x_{\text{RB}}, u_{\text{RB}}), \mathbf{J}_{Q\alpha}(x_{\text{RB}}, u_{\text{RB}}), \mathbf{J}_{R_{\text{INS}}}(x_{\text{RB}}, u_{\text{RB}}), \mathbf{J}_{V_{\text{INS}}}(x_{\text{RB}}, u_{\text{RB}}), \quad (4.27)$$

With these, the full symbolic state space for the rigid body was assembled. The symbolic equation are too long to put in this thesis, but they can be simplified using the next steps. The next step is to insert the values of the states corresponding to the equilibrium.

4.5 Inserting Operating Points and Known variables

After taking the Jacobians of the full dynamic equations, the symbolic equations are very long. Because they are difficult to read and would just fill out a lot of pages, they will not be written in full. However, after inserting the values for the operating point, most of which are zero, the equations will be much shorter and readable as it will be shown. For an equilibrium the following should be true:

$$f(\bar{x}, \bar{u}) = 0 \quad (4.28)$$

Where f is the dynamic equations. \bar{x} and \bar{u} are the states and inputs that satisfy the equation. With the definitions of the states and for this to be equal to zero, most of the variables will have to be zero except for the forward velocity, as it will be controlled by the guidance.

Because the translational control is done in the trajectory frame none of the states will affect the gravitational nor the Coriolis acceleration. This makes sense as the orientation that is captured by the states Θ and Ω no longer affects the direction which the forces are applied with respect to the translational states R and V . These contributions from gravity and Coriolis are dynamics arising from the rotation of the body frame. If the trajectory does not follow a gravity turn, such as during the initial liftoff and the descent, these forces can be added as a known input to the system in a similar manner to a disturbance. The states that are supposed to be stabilized will have to be at an equilibrium, which will be at zero. That is except the translational position R which in this model has no effects on any

of the dynamics in the model. This will for consistency also be set to zero. As mentioned, the forward velocity of the LV v_x will also not be zero as it will not be controlled by the attitude controller. Some assumptions will also be used to further simplify the equations. The position of CG, CP and INS will all be assumed to be centered along the primary axis of the body frame. That is, the x- and y-coordinates for these will be set to zero. The full set of variables corresponding to states that will be replaced for a linearization at the equilibrium along the trajectory is:

$$V_0 = \begin{pmatrix} v_x \\ 0 \\ 0 \end{pmatrix}, \Theta_0 = 0, \Omega_0 = 0, R_0 = 0, V_0 = 0, \mathcal{B}_0 = 0, \ddot{\mathcal{B}}_0 = 0, W_0 = 0, \epsilon_0 = 0, \quad (4.29)$$

And similarly for the positions of certain frames: Locations P_{CG} , P_{INS} and P_{CP} :

$$P_{\text{CG},0} = \begin{pmatrix} x_{\text{CG}} \\ 0 \\ 0 \end{pmatrix}, P_{\text{INS},0} = \begin{pmatrix} x_{\text{INS}} \\ 0 \\ 0 \end{pmatrix}, P_{\text{CP},0} = \begin{pmatrix} x_{\text{CP}} \\ 0 \\ 0 \end{pmatrix} \quad (4.30)$$

Locations P_{PVP} and P_{CG_N} :

$$P_{\text{PVP},0} = \begin{pmatrix} x_{\text{PVP}} \\ y_{\text{PVP}} \\ z_{\text{PVP}} \end{pmatrix}, P_{\text{CG}_N,0} = \begin{pmatrix} x_{\text{CG}_N} \\ 0 \\ 0 \end{pmatrix} \quad (4.31)$$

Where a vector set to zero means that all entries are set to zero. Once these variables have been replaced in the Jacobians, the state space is far more manageable. However, in that form it still has 12 states. Not all of which is intended to be controlled. More specifically the states r_x and v_x . These will be taken into account by the guidance. Even with the remaining 10 states, the control problem is large to work with. Therefore, the initial control will be in the 3 2-Dimensional planes. These planes will be called the yaw-, pitch- and roll-planes. The name corresponds to the plane whose center rotates around the yaw-, pitch- and roll-axis. Getting these controllers will be explained in the next section using pitch as the example.

4.6 Rigid-body State Space

Now that the full dynamics have been derived and is on the form:

$$\dot{x}_{\text{RB}} = A_{\text{RB}}x_{\text{RB}} + B_{\text{RB}}u_{\text{RB}} \quad (4.32)$$

$$y_{\text{RB}} = C_{\text{RB}}x_{\text{RB}} + D_{\text{RB}}u_{\text{RB}} \quad (4.33)$$

The controllers for the 3 planes can be derived using the relevant entries from the RB matrices. Therefore, the states that are controlled by the 3 controllers control the following states.

States $x_{\text{RB,yaw}}$, outputs $y_{\text{RB,yaw}}$ and inputs $u_{\text{RB,yaw}}$ for the yaw controller:

$$x_{\text{RB,yaw}} = \begin{pmatrix} \theta_z \\ \omega_z \\ r_y \\ v_y \end{pmatrix}, u_{\text{RB,yaw}} = \begin{pmatrix} \beta_z \\ \ddot{\beta}_z \\ w_y \\ \epsilon_n \end{pmatrix}, y_{\text{RB,yaw}} = \begin{pmatrix} Q\alpha \\ \theta_{\text{INS},z} \\ \omega_{\text{INS},z} \\ r_{\text{INS},y} \\ v_{\text{INS},y} \end{pmatrix} \quad (4.34)$$

States $x_{\text{RB,pitch}}$, outputs $y_{\text{RB,yaw}}$ and inputs $u_{\text{RB,pitch}}$ for the yaw controller:

$$x_{\text{RB,pitch}} = \begin{pmatrix} \theta_y \\ \omega_y \\ r_z \\ v_z \end{pmatrix}, u_{\text{RB,pitch}} = \begin{pmatrix} \beta_y \\ \ddot{\beta}_y \\ w_z \\ \epsilon_n \end{pmatrix}, y_{\text{RB,pitch}} = \begin{pmatrix} Q\alpha \\ \theta_{\text{INS},y} \\ \omega_{\text{INS},y} \\ r_{\text{INS},z} \\ v_{\text{INS},z} \end{pmatrix} \quad (4.35)$$

States $x_{\text{RB,roll}}$, outputs $y_{\text{RB,yaw}}$ and inputs $u_{\text{RB,roll}}$ for the yaw controller:

$$x_{\text{RB,roll}} = \begin{pmatrix} \theta_x \\ \omega_x \end{pmatrix}, u_{\text{RB,roll}} = \begin{pmatrix} \beta_y \\ \ddot{\beta}_y \\ \beta_z \\ \ddot{\beta}_z \end{pmatrix}, y_{\text{RB,roll}} = \begin{pmatrix} \theta_{\text{INS},x} \\ \omega_{\text{INS},x} \end{pmatrix} \quad (4.36)$$

As an example, the state space matrix for $A_{\text{RB,pitch}}$ is constructed from A_{RB} with the entries corresponding to the states used for the pitch control:

$$A_{\text{RB,pitch}} = \begin{pmatrix} a_{\dot{\theta},y}^{\theta,y} & a_{\dot{\theta},y}^{\omega,y} & a_{\dot{\theta},y}^{r,z} & a_{\dot{\theta},y}^{v,z} \\ a_{\dot{\theta},y}^{\theta,y} & a_{\dot{\omega},y}^{\omega,y} & a_{\dot{\omega},y}^{r,z} & a_{\dot{\omega},y}^{v,z} \\ a_{\dot{\omega},y}^{\theta,y} & a_{\dot{\omega},y}^{\omega,y} & a_{\dot{\omega},y}^{r,z} & a_{\dot{\omega},y}^{v,z} \\ a_{\dot{r},z}^{\theta,y} & a_{\dot{r},z}^{\omega,y} & a_{\dot{r},z}^{r,z} & a_{\dot{r},z}^{v,z} \\ a_{\dot{v},z}^{\theta,y} & a_{\dot{v},z}^{\omega,y} & a_{\dot{v},z}^{r,z} & a_{\dot{v},z}^{v,z} \end{pmatrix} \quad (4.37)$$

Where the notation $a_{\dot{v},z}^{\theta,y}$ is the matrix entry from A_{RB} corresponding to the effect of θ_y on \dot{v}_z . The matrix for just the pitch model will end with a similar structure to the simplified block matrix in 4.13.

$$A_{\text{RB,pitch}} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ a_{\dot{\omega},y}^{\theta,y} & a_{\dot{\omega},y}^{\omega,y} & 0 & a_{\dot{\omega},y}^{v,z} \\ 0 & 0 & 0 & 1 \\ a_{\dot{v},z}^{\theta,y} & a_{\dot{v},z}^{\omega,y} & 0 & a_{\dot{v},z}^{v,z} \end{pmatrix} \quad (4.38)$$

With this method the remaining matrices for the rigid body state spaces were derived. These separate state spaces will model the dynamics in the three control planes. This is assuming that the planes are not interacting. When doing the simplification explained in the section, all the cross terms are not used. These terms could be analyzed to understand how much the cross terms could affect different planes, and this could be taken into consideration. The cross terms could also just be used in a full controller with all the 3 control planes in one. This however will greatly increase the complexity of the controller. And for the purposes of the project, it will not be done.

4.7 Bending Mode Model

The flexible dynamics of the LV will be modelled by the first two bending modes. The model used for the bending modes in this project will be modelled directly in two dimensions for each of the pitch and yaw planes. Which means that there will be no modelled interaction between the different control planes. The model is based on the papers [22] and [20], and is using the same method as in [18]. The model is mainly used because the data from ESA on bending modes and bending shape consists of the coefficients used for this model. Since the model is based on separated 2D-planes there is no direct coupling between the bending modes in different directions. And there is no roll twisting modelled with this model. This model does however capture the most important flexible body dynamics for the control synthesis. While the model can include an arbitrary number of bending modes, for this project only the first 2 bending modes are used. This is because we want to keep the complexity down, and the frequencies of the remaining bending modes is of a much higher frequency than the control. The model consists of a 2nd order model for each bending mode q_i :

$$\ddot{q}_i + 2\zeta_{q_i}\omega_{q_i}\dot{q}_i + \omega_{q_i}^2 q_i = -F_{\text{thrust}}\psi_{\text{PVP},i}\beta - \left(m_N l_N \psi_{\text{PVP},i} - J_N \psi'_{\text{PVP},i}\right) \ddot{\beta} \quad (4.39)$$

Where q_i is the i 'th bending mode state. ω_{q_i} is the natural frequency of the i 'th bending mode. ζ_{q_i} is the i 'th bending mode damping ratio. m_N is the mass of the nozzle. l_N is the length from the PVP of the nozzle to its center of gravity. J_N is the inertia of the nozzle. β is the angle of the actuator in the relevant control plane. $\psi_{\text{PVP},i}$ describes the distance that the point PVP moves laterally based on the state of the bending mode. $\psi'_{\text{PVP},i}$ describes the rotation of the nozzle frame based on the state of the bending mode. The parameters ζ_{q_i} , ω_{q_i} , $\psi_{\text{PVP},i}$, m_N , l_N , J_N and $\psi'_{\text{PVP},i}$ are all parameters provided by ESA.

The distance ψ_{PVP} and the rotation ψ'_{PVP} is illustrated on figure 4.1

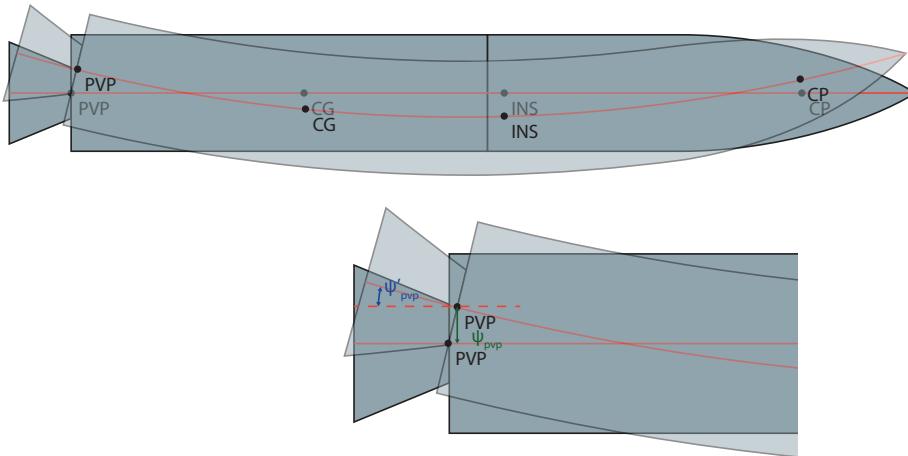


Figure 4.1: Illustration of the effect of the bending and the related parameters for the PVP frame.

Similar parameters are available for the INS, ψ_{INS} and the rotation ψ'_{INS} which will be used for the measurement matrix. With this behavior of the bending states, the following forces and moment can be written for the effect of the bending modes on the LV. First the

Force from the resulting angle of the nozzle:

$$F_{\text{BM},i} = F_{\text{thrust}} \psi'_{\text{PVP},i} q_i \quad (4.40)$$

And the moments:

$$M_{\text{BM},i} = -F_{\text{thrust}} \left(l_{\text{CG}} \psi'_{\text{PVP},i} q_i + \psi_{\text{PVP},i} q_i \right) \quad (4.41)$$

Where $F_{\text{BM},i}$ is the force from the i 'th bending mode on the LV. $M_{\text{BM},i}$ is the moment from the i 'th bending mode on the LV. F_{thrust} is the thrust of the relevant engine, and l_{CG} is the length from the PVP to the CG. Similarly, the contribution from bending modes to the INS can be written as:

$$\theta_{\text{INS}} = \theta_{\text{INS},\text{RB}} + \sum_{i=1}^n \phi'_{\text{INS},i} q_i \quad (4.42)$$

$$\omega_{\text{INS}} = \omega_{\text{INS},\text{RB}} + \sum_{i=1}^n \phi'_{\text{INS},i} \dot{q}_i \quad (4.43)$$

$$r_{\text{INS}} = r_{\text{INS},\text{RB}} + \sum_{i=1}^n \phi_{\text{INS},i} q_i \quad (4.44)$$

$$v_{\text{INS}} = v_{\text{INS},\text{RB}} + \sum_{i=1}^n \phi_{\text{INS},i} \dot{q}_i \quad (4.45)$$

Where for this project $n=2$. With the contribution from the bending modes described, these will be included in the full state space described in the next section.

4.8 Final State Space for LV

With the method for deriving the states space for the rigid body and the bending modes described in the previous sections of this chapter, this section will formulate the full state space used for control synthesis in the pitch plane. The combination of the Rigid Body (RB) and Bending Mode (BM) matrices will have the form:

$$\begin{aligned} \begin{bmatrix} \dot{x}_{\text{RB}} \\ \dot{x}_{\text{BM}} \end{bmatrix} &= \underbrace{\begin{bmatrix} A_{\text{RB}}^{\text{RB}} & A_{\text{RB}}^{\text{BM}} \\ A_{\text{BM}}^{\text{RB}} & A_{\text{BM}}^{\text{BM}} \end{bmatrix}}_{A_{\text{LV}}} \begin{bmatrix} x_{\text{RB}} \\ x_{\text{BM}} \end{bmatrix} + \underbrace{\begin{bmatrix} B_{\text{RB}} \\ B_{\text{BM}} \end{bmatrix}}_{B_{\text{LV}}} u_{\text{pitch}} \\ y_{\text{pitch}} &= \underbrace{\begin{bmatrix} C_{\text{RB}} & C_{\text{BM}} \end{bmatrix}}_{C_{\text{LV}}} \begin{bmatrix} x_{\text{RB}} \\ x_{\text{BM}} \end{bmatrix} + \underbrace{D_{\text{RB}} u_{\text{pitch}}}_{D_{\text{LV}}} \end{aligned} \quad (4.46)$$

Where x_{RB} and x_{BM} are the states for the RB and the BMs. The notation $A_{\text{RB}}^{\text{BM}}$ is the matrix describing the effect from the BM states on the RB states. B_{RB} and B_{BM} describe the linear effect from the control input u_{pitch} on the RB and BM states. C_{RB} and C_{BM} describe the linear effect of the RB and BM states on the measured output y_{pitch} . D_{RB} describe the direct effect of the control input on the measured output y_{pitch} .

Using this form and the modelling described previously in this chapter, it can be written in its full form:

$$\begin{aligned}
 \begin{bmatrix} \dot{\theta}_y \\ \dot{\omega}_y \\ \dot{r}_z \\ \dot{v}_z \\ \dot{q}_1 \\ \dot{q}_2 \\ \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} &= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{\omega,y}^{\theta,y} & a_{\omega,y}^{\omega,y} & 0 & a_{\omega,y}^{v,z} & a_{\omega,y}^{q_1} & a_{\omega,y}^{q_2} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -a_{\dot{v},z}^{\theta,y} & -a_{\dot{v},z}^{\omega,y} & 0 & -a_{\dot{v},z}^{v,z} & -a_{\dot{v},z}^{q_1} & -a_{\dot{v},z}^{q_2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & a_{\dot{q}_1}^{q_1} & 0 & a_{\dot{q}_1}^{q_2} & 0 \\ 0 & 0 & 0 & 0 & 0 & a_{\dot{q}_2}^{q_2} & 0 & a_{\dot{q}_2}^{q_2} \end{bmatrix} \begin{bmatrix} \theta_y \\ \omega_y \\ r_z \\ v_z \\ q_1 \\ q_2 \\ \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \\
 &+ \begin{bmatrix} 0 & 0 & 0 & 0 \\ b_{\dot{\omega},y}^{\beta,y} & b_{\dot{\omega},y}^{\ddot{\beta},y} & b_{\dot{\omega},y}^{w,z} & b_{\dot{\omega},y}^{\epsilon} \\ 0 & 0 & 0 & 0 \\ b_{\dot{v},z}^{\beta,y} & b_{\dot{v},z}^{\ddot{\beta},y} & b_{\dot{v},z}^{w,z} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ b_{\dot{q}_1}^{\beta,y} & b_{\dot{q}_1}^{\ddot{\beta},y} & 0 & 0 \\ b_{\dot{q}_2}^{\beta,y} & b_{\dot{q}_2}^{\ddot{\beta},y} & 0 & 0 \end{bmatrix} \begin{bmatrix} \beta_y \\ \ddot{\beta}_y \\ w_z \\ \epsilon_n \end{bmatrix} \quad (4.47) \\
 \begin{bmatrix} Q\alpha \\ \theta_{INS,y} \\ \omega_{INS,y} \\ r_{INS,z} \\ v_{INS,z} \end{bmatrix} &= \begin{bmatrix} c_{Q\alpha}^{\theta,y} & c_{Q\alpha}^{\omega,y} & 0 & c_{Q\alpha}^{v,z} & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & c_{\theta,y}^{q_1} & c_{\theta,y}^{q_2} & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & c_{\omega,y}^{\dot{q}_1} & c_{\omega,y}^{\dot{q}_2} \\ c_{r,z}^{\theta,y} & 0 & 1 & 0 & c_{r,z}^{q_1} & c_{r,z}^{q_2} & 0 & 0 \\ 0 & c_{v,z}^{\omega,y} & 0 & 1 & 0 & 0 & c_{v,z}^{\dot{q}_1} & c_{v,z}^{\dot{q}_2} \end{bmatrix} \begin{bmatrix} \theta_y \\ \omega_y \\ r_z \\ v_z \\ q_1 \\ q_2 \\ \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \\
 &+ \begin{bmatrix} 0 & 0 & d_{Q\alpha}^{w,y} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \beta_y \\ \ddot{\beta}_y \\ w_z \\ \epsilon_n \end{bmatrix}
 \end{aligned}$$

With the coefficients, first for the A_{LV} :

$$a_{\dot{\omega},y}^{\theta,y} = -\frac{S_{\text{ref}} \rho v_x^2 \left(\left(\frac{\partial}{\partial \theta_y} C_l \right) + C_d \right) (x_{\text{CG}} - x_{\text{CP}})}{2 J_y} \quad (4.48)$$

$$a_{\dot{\omega},y}^{\omega,y} = -\frac{S_{\text{ref}} \rho v_x (x_{\text{CG}} - x_{\text{CP}}) \left(v_x \left(\frac{\partial}{\partial \omega_y} C_l \right) + (x_{\text{CG}} - x_{\text{CP}}) C_d \right)}{2 J_y} \quad (4.49)$$

$$a_{\dot{\omega},y}^{v,z} = -\frac{S_{\text{ref}} \rho v_x (x_{\text{CG}} - x_{\text{CP}}) v_x \left(\frac{\partial}{\partial v_z} C_l \right) + C_d}{2 J_y} \quad (4.50)$$

$$a_{\dot{v},z}^{\theta,y} = -\frac{F_{\text{thrust}} + \frac{S_{\text{ref}} \rho v_x^2 \left(\frac{\partial}{\partial \theta_y} C_l \right)}{2}}{m} \quad (4.51)$$

$$a_{\dot{v},z}^{\omega,y} = -\frac{\frac{S_{\text{ref}} \rho v_x^2 \left(\frac{\partial}{\partial \omega_y} C_l \right)}{2} + \frac{S_{\text{ref}} \rho v_x (x_{\text{CG}} - x_{\text{CP}}) C_d}{2}}{m} \quad (4.52)$$

$$a_{\dot{v},z}^{v,z} = -\frac{S_{\text{ref}} \rho v_x^2 \left(\frac{\partial}{\partial v_z} C_l \right) + C_d}{2 m} \quad (4.53)$$

$$a_{\dot{\omega},y}^{q_i} = \frac{F_{\text{thrust}} (\psi'_{\text{PVP},i} (x_{\text{PVP}} - x_{\text{CG}}) + \psi_{\text{PVP},i})}{J_y} \quad (4.54)$$

$$a_{\dot{v},z}^{q_i} = \frac{F_{\text{thrust}} \psi'_{\text{PVP},i}}{m} \quad (4.55)$$

$$a_{\ddot{q}_i}^{q_i} = -\omega_{q,i}^2 \quad (4.56)$$

$$a_{\ddot{q}_i}^{\dot{q}_i} = -2\zeta_{q,i} \omega_{q,i} \quad (4.57)$$

For the the B_{LV} :

$$b_{\dot{\omega},y}^{\beta,y} = \frac{F_{\text{thrust}} (x_{\text{PVP}} - x_{\text{CG}})}{J_y} \quad (4.58)$$

$$b_{\ddot{\omega},y}^{\ddot{\beta},y} = -\frac{J_{N,y} - x_{\text{CG}} x_{\text{CG,N}} m_N + x_{\text{CG,N}} x_{\text{PVP}} m_N}{J_y} \quad (4.59)$$

$$b_{\dot{\omega},y}^{w,z} = -\frac{S_{\text{ref}} \rho v_x (x_{\text{CG}} - x_{\text{CP}}) C_d}{J_y} \quad b_{\dot{\omega},y}^{\epsilon} = \frac{2 F_{\text{thrust}} z_{\text{PVP}}}{J_y} \quad (4.60)$$

$$b_{\dot{v},z}^{\beta,y} = -\frac{F_{\text{thrust}}}{m} \quad b_{\dot{v},z}^{\ddot{\beta},y} = \frac{x_{\text{CG,N}} m_N}{m} \quad (4.61)$$

$$b_{\dot{v},z}^{w,z} = -\frac{S_{\text{ref}} \rho v_x C_d}{2 m} \quad b_{\ddot{q}_i}^{\beta,y} = F_{\text{thrust}} \psi_{\text{PVP},i} \quad (4.62)$$

$$b_{\ddot{q}_i}^{\ddot{\beta},y} = J_N \psi'_{\text{PVP},i} + m_N (x_{\text{CP}} - x_{\text{CG}}) \quad (4.63)$$

For the the C_{LV} :

$$c_{Q\alpha}^{\theta,y} = \frac{\rho v_x^2}{2} \quad c_{Q\alpha}^{\omega,y} = \frac{\rho (x_{\text{CG}} - x_{\text{CP}}) \sqrt{v_x^2}}{2} \quad c_{Q\alpha}^{v,z} = \frac{\rho \sqrt{v_x^2}}{2} \quad (4.64)$$

$$c_{r,z}^{\theta,y} = x_{\text{CG}} - x_{\text{INS}} \quad c_{v,z}^{\omega,y} = x_{\text{CG}} - x_{\text{INS}} \quad c_{\theta,y}^{q_i} = \psi'_{\text{INS},i} \quad (4.65)$$

$$c_{\dot{\omega},y}^{\dot{q}_i} = \psi'_{\text{INS},i} \quad c_{r,z}^{q_i} = \psi_{\text{INS},i} \quad c_{v,z}^{\dot{q}_i} = \psi_{\text{INS},i} \quad (4.66)$$

And the one coefficient for the D_{LV} :

$$d_{Q\alpha}^{w,y} = \frac{\rho \sqrt{v_x^2}}{2} \quad (4.67)$$

This is the model G_{LV} that will be used for the control synthesis. It will be used to make the individual LTI model for different points of the trajectory. The Evolution of the frequency response and the poles can be seen on figure 4.2 and 4.3.

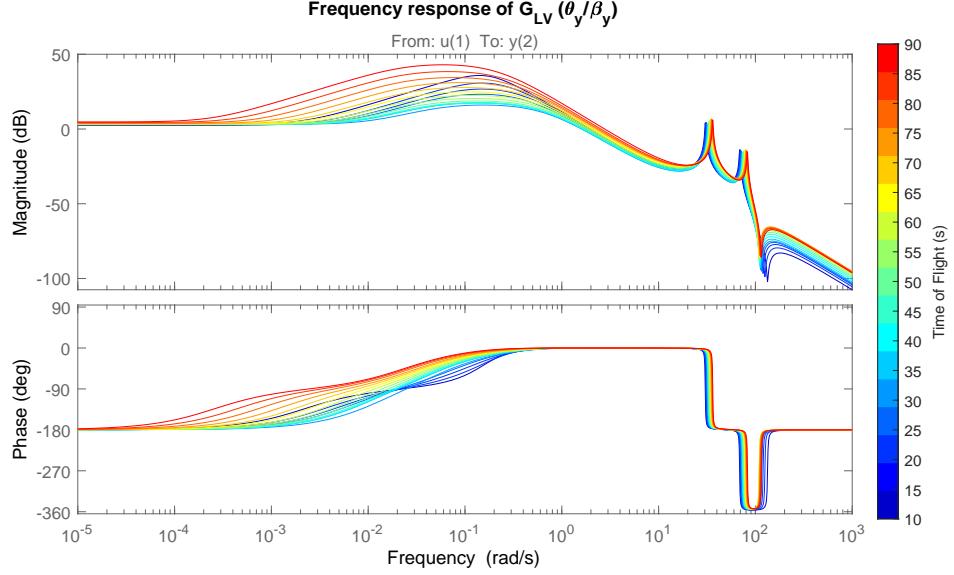


Figure 4.2: Frequency response of $G_{LV} (\theta_y/\beta_y)$ from time 10 s to 90 s

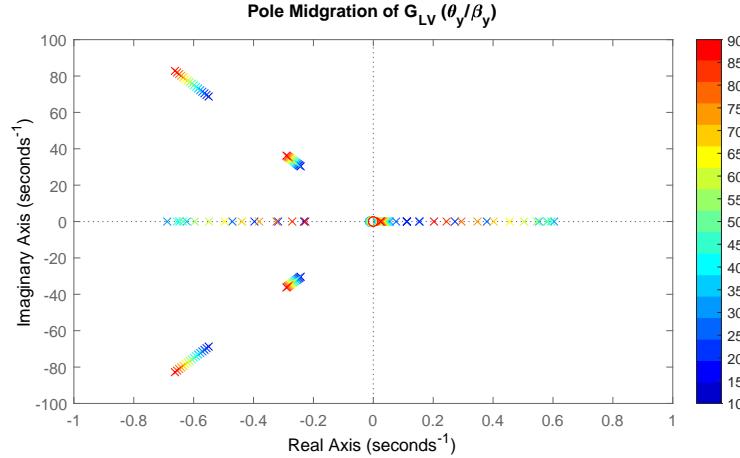


Figure 4.3: Pole Migration of $G_{LV} (\theta_y/\beta_y)$ from time 10 s to 90 s

Controller Design 5

This chapter will outline the design process for the controllers. First a quick summary of the control theory used for the controller design will be presented. Then the proposed design for the pitch controller will be presented. Followed by explanations of the design choices and procedures. The figures in this section are color coded for readability. The inputs will be coloured blue and the outputs red. The names of the internal signals used to connect the sub models will be in green.

5.1 Control Introduction

This section will briefly introduce some of the core control theory used for the design of the controller in this project. First a summary of the Linear Fractional Transforms (LFT) and the notation that will be used. Then a quick introduction for the H_∞ controller that will be used as a reference and as a natural step towards building the LPV controller. And then finally the LPV controller, that is the final step that will take the time varying aspect of the LV into account.

5.1.1 Linear Fractional Transform

The LFTs can generally be used for modeling dynamic systems and will here primarily be used for uncertainties and for connecting the controller to the system.

The LFT can be used to model a feedback interconnection of two matrix operators. Let $M \in \mathbb{C}^{(n_d+n_u) \times (n_e+n_y)}$ and $\Delta \in \mathbb{C}^{n_y \times n_u}$. With M partitioned as:

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \quad (5.1)$$

Then the two versions of the LFT can be written as:

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

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

And the corresponding interconnections can be seen on figures 5.1b and 5.1a

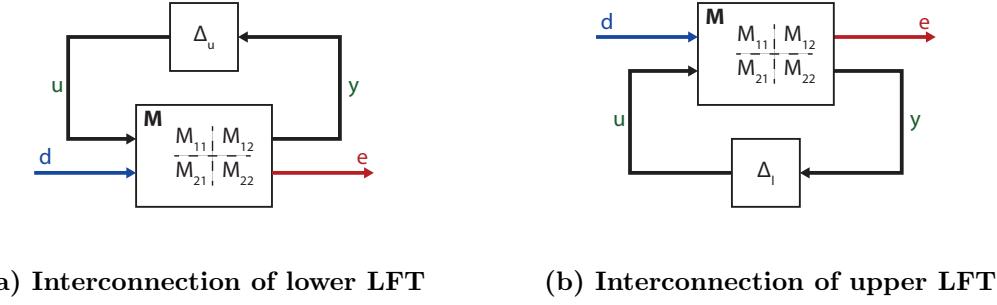


Figure 5.1: The two common LFT interconnections

The upper LFT is commonly used in control theory to model uncertainties. For this project the uncertainties will be variations of real numbers and no complex uncertainties. Uncertainties for a real parameter x can be modelled as:

$$x = x_0 + \sigma_x \delta_x \quad (5.4)$$

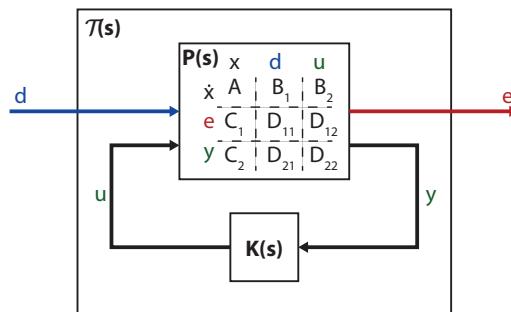
Where x_0 is the nominal value of x . σ_x is the magnitude of the uncertainty and δ_x is a norm bounded uncertainty flag with $\|\delta_x\| < 1$. This is convenient as the M_{22} in 5.2 can represent the nominal system and the remaining terms can represent $\sigma_x \delta_x$ for any rational parameter variation. These uncertainties in this project will be taken care of by MATLAB using the Robust Control Toolbox, but it is important to know the structure and its limitations. The Lower LFT is most commonly used to model the interconnection of a controller connected to a system via feedback. A more in-depth explanation can be found in [24].

5.1.2 H_∞ and structured H_∞ Control

The H_∞ control problem is usually formulated in the following way. Using the interconnection of the plant $P(s)$:

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

And the controller $K(s)$ as seen on figure 5.2

Figure 5.2: Standard H_∞ interconnection.

Where the goal is to minimize

$$\min_{K(s)} \|\mathcal{T}_{ed}(s)\|_\infty = \min_{K(s)} \max_{\omega \in \mathbb{R}} \bar{\sigma}(\mathcal{T}_{ed}(j\omega)) = \gamma^* \quad (5.6)$$

This can be done in multiple ways. The two main ways this is done is either by directly solving a two-Riccati formulae [25], or solving the same Riccati equation set up as LMIs [26]. Using a numerical method it is solved iteratively, as solving directly for optimal solution γ^* is hard. Instead, it is easier to solve for any solution that is better than some γ . That is some $\gamma^* \leq \gamma$. Then keep solving for a lower γ until it is no longer solvable. This will result in a high order controller that is of the same order as the system. This can be undesirable.

A major advantage of the H_∞ synthesis is that it can also take uncertainties into consideration when modelled with LFTs by using the small gain theorem [27]. This is usually done in an iterative algorithm called DK-iteration where there are two steps. First the synthesis of the H_∞ controller, then another step called μ -analysis. That is a robustness analysis that will give some scaling matrices that can be used to synthesise a more robust controller, and then with this controller the analysis can be done again and so on [27]. The result is that this process converges towards a more robust controller. This can also be done for the structured H_∞ controller.

The structured H_∞ control synthesis is a similar problem to the H_∞ synthesis but has its own pros and cons. The main advantage is the option to decide the controller structure. This allows for better understanding of the inner workings and structure of the controller which can be very useful, it also allows for the design of less complex controllers which can be very useful when implementing. However, the main problem with the structured H_∞ controller is that it requires Bilinear Matrix Inequalities which makes the optimization problem non-convex [28]. This means that the optimization is not guaranteed to find a global optimal controller. However, with a combination of random initial starting conditions and an understanding of the expected controller behavior, a good controller can usually be found.

5.1.3 Weighting Functions

Weighting the inputs and outputs of the system is a common way to make the design process simpler[27]. For the designs based on norms it is essential that the input output relations are normalized as the synthesis will assume a normalized system. This weighing will define what is the normalized inputs and outputs. That is, the input weights, $W_i(s)$ defines what is the expected maximum input, and the output weights, $W_o(s)$, define the maximum outputs that fulfill the performance requirements. The system with these weights is called the augmented system and the interconnection can be seen on figure 5.3

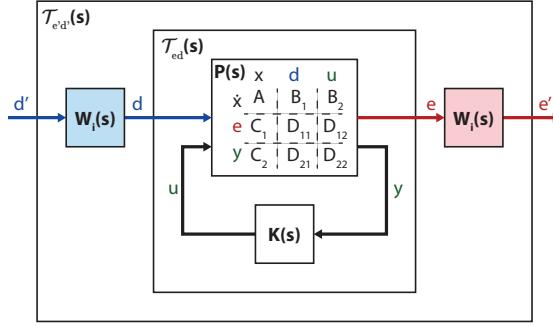


Figure 5.3: Augmented H_∞ interconnection.

Where the new plant to be optimized is $\mathcal{T}_{e'd'}(s)$ with input output relation:

$$\mathbf{e}' = \underbrace{W_o \mathcal{F}_l(P, K) W_i}_{\mathcal{T}_{e'd'}(s)} \mathbf{d}' \quad (5.7)$$

With these weights the performance parameter indicates how well the performance requirements are fulfilled with $\gamma \leq 1$ is fulfilled. Additionally, with the LFT representation of uncertainties and all the uncertainties pulled out in Δ the robust problem can be set up as:

$$\min_{K(s)} \max_{\Delta \in \Delta_{\mathbb{R}}} \|\mathcal{T}_{e'd'}(s, \Delta)\|_\infty \quad (5.8)$$

With the interconnection as seen on figure 5.4

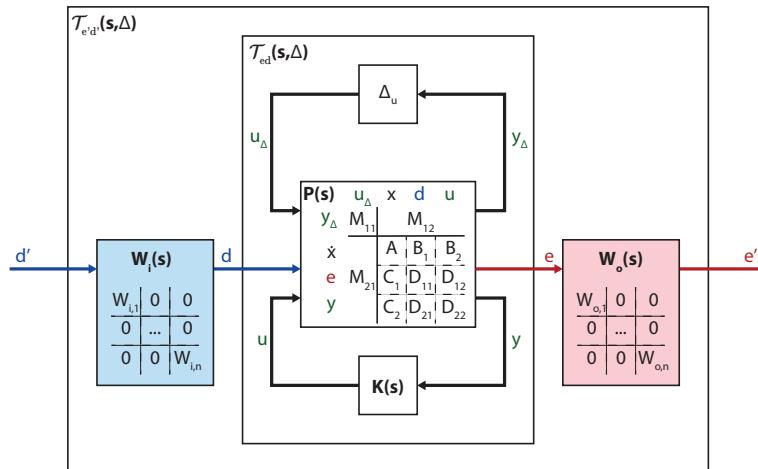


Figure 5.4: Augmented robust H_∞ interconnection.

These interconnections can be used for both H_∞ and structured H_∞ . This structure is also what will be used for the LPV synthesis, the only difference being that the matrices are no longer constant, but depend on some parameter ρ .

5.1.4 Linear Parameter Varying Control

For this project, the dynamics of the system varies significantly over time. Mainly the velocity, the air density and the weight of the rocket. The velocity and air density greatly affects the aerodynamic effects, and the weight influences most of the dynamics. The previously presented controller syntheses have been for Linear Time Invariant (LTI) systems. These methods can be used with gain scheduling by switching or interpolating and can usually, in practice, give a good controller the parameters that vary slowly. However, such gain scheduling does not guarantee performance nor stability when the parameters are varying in time. This is why the LPV controllers are used as they can guarantee stability and performance when the parameters are varying in time. LPV is also a type of gain scheduling. However, the synthesis takes the effect from changing the variables into account based on some measurable parameters ρ . The LPV controller can guarantee stability and performance for linear combination of the gridded points and is generated in a single step. The LPV control synthesis can be seen as a generalization of the H_∞ control. It uses the same framework as the H_∞ control, which is why both have been explored during this project. The LPV systems is very similar to Linear Time Varying (LTV) systems. In fact, the first iteration developed for this project is made using time as the scheduling parameter and is thus technically an LTV system [29]. This is because the controller is developed for the planned part of the trajectory. Where the system has predictable dynamics at each time instance. Thus, the dynamics will develop in a predictable manner with respect to time during this period, and time can be used as a scheduling parameter. For a controller for the descent or any other part where the trajectory is not predictable, another scheduling parameter is needed, as the dynamics will not develop in a predictable manner. For the ascent, the controller can also be scheduled based on the velocity. Using velocity for scheduling will therefore be more robust towards dispersions in velocity which is one of the main sources of uncertainty for the LV dynamics [21] .

While LPV synthesis does not yet have any official support in MATLAB, there is a third party toolbox called *LPVTools* that can do some LPV synthesis[30]. The *LPVTools* toolbox will be used for the final synthesis of the LPV controller in this project.

LPV summary

An LPV system is in many ways similar to an LTI system. The only difference is that the matrices used to describe the system can depend on some parameters over time and can be written as:

$$\begin{bmatrix} \dot{x}(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} A(\rho(t)) & B(\rho(t)) \\ C(\rho(t)) & D(\rho(t)) \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix}, \quad \underline{\rho} \leq \rho \leq \bar{\rho} \quad (5.9)$$

Where $\rho(t)$ is a vector of time varying parameters $\rho(t) = [\rho_1(t), \dots, \rho_{n_\rho}(t)]^T$ that can take any value within \mathcal{P} . $\dot{\rho}$ is the imposed rate of change of ρ and can change within the defined limits $\underline{\nu} \leq \dot{\rho} \leq \bar{\nu}$. These constraints on ρ capture a set of admissible parameter trajectories of ρ that the synthesis will consider. The time dependence will be omitted for readability.

There are three main ways to work with LPV systems[31], polytopic, LFT based, and grid based. Polytopic description relies on affine parameter dependence on ρ in the matrices. LFT based LPV use LFT theory to capture the variation of the parameters. Finally, the grid based LPV is similar to traditional gain scheduling as it uses a series of LTI models gridded according to the scheduling parameter and is straightforward to use even with complex systems. The last two methods are implemented in the *LPVTools* toolbox and the grid based method will be used for this project.

The synthesis of the LPV controller uses a similar interconnection to H_∞ controllers but with parameter dependent dynamics. The standard interconnection is used as seen on figure 5.5.

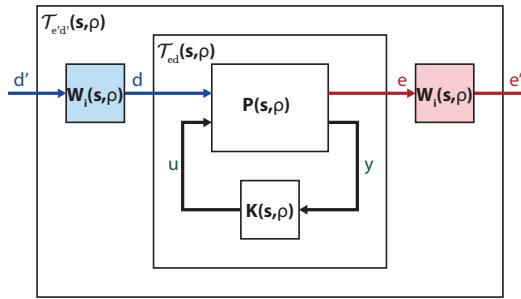


Figure 5.5: LPV interconnection used for synthesis.

Were the plant $P(\rho)$ has the dynamics described by:

$$\begin{bmatrix} \dot{x} \\ e \\ y \end{bmatrix} = \underbrace{\begin{bmatrix} A(\rho) & B_1(\rho) & B_2(\rho) \\ C_1(\rho) & D_{11}(\rho) & D_{12}(\rho) \\ C_2(\rho) & D_{21}(\rho) & D_{22}(\rho) \end{bmatrix}}_{P(\rho)} \begin{bmatrix} x \\ d \\ u \end{bmatrix} \quad (5.10)$$

And the controller $K(\rho)$ described by:

$$\begin{bmatrix} \dot{x}_K \\ u \end{bmatrix} = \underbrace{\begin{bmatrix} A_K(\rho) & B_K(\rho) \\ C_K(\rho) & D_K(\rho) \end{bmatrix}}_{K(\rho)} \begin{bmatrix} x_K \\ y \end{bmatrix} \quad (5.11)$$

Closing the loop with the controller will result in the following system:

$$\begin{bmatrix} \dot{x}_{cl} \\ e \end{bmatrix} = \underbrace{\begin{bmatrix} A_{cl}(\rho) & B_{cl}(\rho) \\ C_{cl}(\rho) & D_{cl}(\rho) \end{bmatrix}}_{T_{ed}(\rho) = \mathcal{F}_l(P(\rho), K(\rho))} \begin{bmatrix} x_{cl} \\ d \end{bmatrix} \quad (5.12)$$

The objective of the synthesis is then to minimize the \mathcal{L}_2 gain of $T_{ed}(\rho)$ with $K(\rho)$:

$$\min_{K(\rho)} \|T_{ed}(\rho)\|_{\mathcal{L}_2 \rightarrow \mathcal{L}_2}; \text{ subject to } \begin{array}{c} \rho \in \mathcal{P} \\ \underline{\nu} \leq \dot{\rho} \leq \bar{\nu} \end{array} \quad (5.13)$$

Where the \mathcal{L}_2 gain is a more general version of the H_∞ norm. For an LTI system the \mathcal{L}_2 gain and H_∞ norm is the same. However, the H_∞ norm is not defined for LPV systems. The \mathcal{L}_2 gain can be used as a performance metric that describes the maximum amplification of the energy from inputs d to outputs e of \mathcal{T}_{ed} in equation 5.12 with the defined limits for the trajectory of ρ . It can be expressed as for an LPV system[18][31][30]:

$$\|\mathcal{T}_{ed}(\rho)\|_{\mathcal{L}_2 \rightarrow \mathcal{L}_2} = \sup_{\substack{\rho \in \mathcal{P} \\ \underline{\nu} \leq \dot{\rho} \leq \bar{\nu}}} \sup_{\substack{d \in \mathcal{L}_2 \\ \|d\|_{\mathcal{L}_2} \neq 0}} \frac{\|e\|_{\mathcal{L}_2}}{\|d\|_{\mathcal{L}_2}} \quad (5.14)$$

An LPV system can be either quadratically stable or robustly stable. Quadratic stability is very conservative and relies on finding a single Lyapunov function P that proves stability for all values of ρ . The system is said to be quadratically stable if it satisfies the following LMI:

$$\begin{bmatrix} A_{cl}(\rho)^T P + PA_{cl}(\rho) & PB_{cl}(\rho) & C_{cl}(\rho)^T \\ * & -\gamma I & D_{cl}(\rho)^T \\ * & * & -\gamma I \end{bmatrix} < 0 \quad (5.15)$$

Where the matrices are from 5.12. And the $*$ represents matrices that will make the LMI symmetric. γ here represents an upper bound of the L_2 -norm from equation 5.14:

$$\|\mathcal{T}_{ed}(\rho)\|_{\mathcal{L}_2 \rightarrow \mathcal{L}_2} \leq \gamma \quad (5.16)$$

The robust stability allows for the Lyapunov function $P(\rho)$ to change with the scheduling parameter ρ and can be far less conservative. The inclusion of the dependence of ρ on the Lyapunov function increases the complexity of the synthesis. This is in part because one has to consider the derivative of the Lyapunov function with respect to the parameter ρ . The system is said to robustly stable if it satisfies the following LMI:

$$\begin{bmatrix} A_{cl}(\rho)^T P(\rho) + P(\rho)A_{cl}(\rho) + \sum \dot{\rho}_i \frac{\partial P}{\partial \rho_i}(\rho) & P(\rho)B_{cl}(\rho) & C_{cl}(\rho)^T \\ * & -\gamma I & D_{cl}(\rho)^T \\ * & * & -\gamma I \end{bmatrix} < 0 \quad (5.17)$$

The derivative can be taken into consideration in a linear system by taking the worst cases into consideration. That is, the limits defined for $\dot{\rho}$, $\underline{\nu}$ and $\bar{\nu}$. The process and the LMI's are described in [32]. The specific synthesis used by the *LPVTools* toolbox in this project is for a robust dynamic output feedback controller using gridding. Note that the robustness of the LPV refers to the variable Lyapunov function and not robustness to uncertainties. Synthesis taking uncertainties into account are not yet a part of the toolbox, however a method is discussed here [33]. It uses Integral Quadratic Constraints (IQC) and has a similar procedure to the DK-iteration for LTI systems. However, this is out of scope for the project.

5.2 Control Problem Formulation

On figure 5.6 an overview of the controller structure used for synthesis is seen.

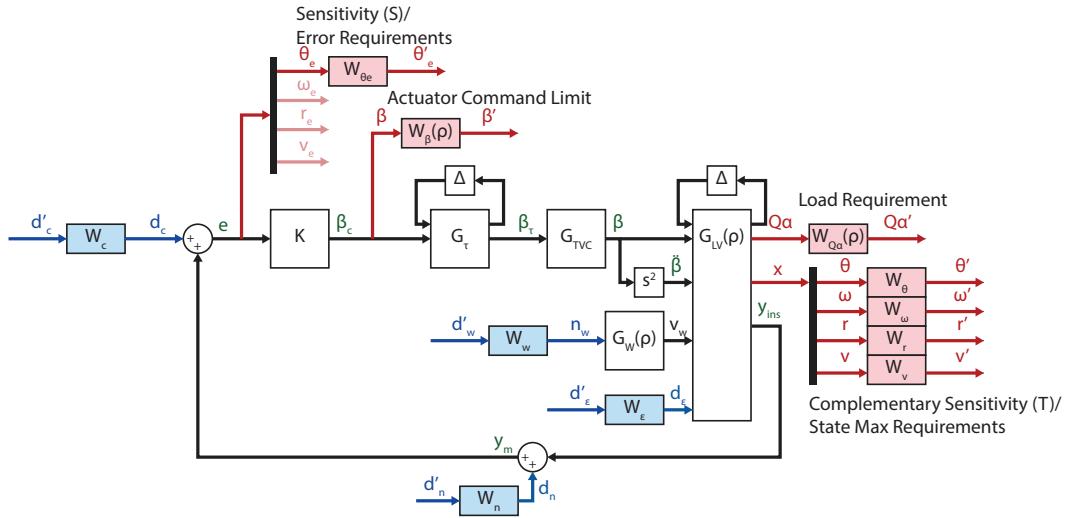


Figure 5.6: Proposed controller structure for synthesis

Besides the main model of the LV, G_{LV} , the controller system also includes models for the actuation of the main engines, G_{TVC} , and for delay, G_{τ} . There are also input and output weights included. These include weights to limit the sensitivity of the pitch channel, $W_{\theta,e}$. The weight $W_{\beta}(\rho)$ to limit the actuation command. The weight $W_{Qa}(\rho)$ to limit the load. The set of weights W_{θ} , W_{ω} , W_r and W_v is to limit the maximum deviation of the states for which there were requirements set. The W_{θ} weight also is used to limit the complementary sensitivity of the θ channel. For the input weights the W_c is based on the expected commands. The input weight W_w defines the expected wind level into the $G_w(\rho)$ that defines the wind shape for the different instances of the flight. W_{ϵ} defines the expected uncertainty in the thrust. The weight W_n defines the magnitude of the noise in the different channels. The K block represents the controller in the system. And finally, the δ blocks represent uncertainties modelled in the connected blocks. These blocks will all be explained in the following subsections. The system will be rearranged using MATLAB to the general form needed for synthesis. For the structured H_{∞} synthesis it will be structured as illustrated on figure 5.4. And for the LPV synthesis it will be arranged as seen on figure 5.5.

5.3 Control Sub-Models

This section will describe the different models used to construct the full model interconnection.

5.3.1 LV Model

The LV model G_{LV} was derived in chapter 4. Uncertainties were added to the model using the LFT framework in MATLAB. The uncertainties of the parameters is defined in table 5.1

Variables	Uncertainty
Q_d, C_d, C_l	20 %
v_x, F_{TVC}, x_{CP}	10 %
J_y, m, x_{CG}	2 %
$\omega_{BM,1}$	20%
$\omega_{BM,2}$	25%
$\psi_{BM,1}, \psi'_{BM,1}$	40%
$\psi_{BM,2}, \psi'_{BM,2}$	50%

Table 5.1: Uncertainties for G_{LV}

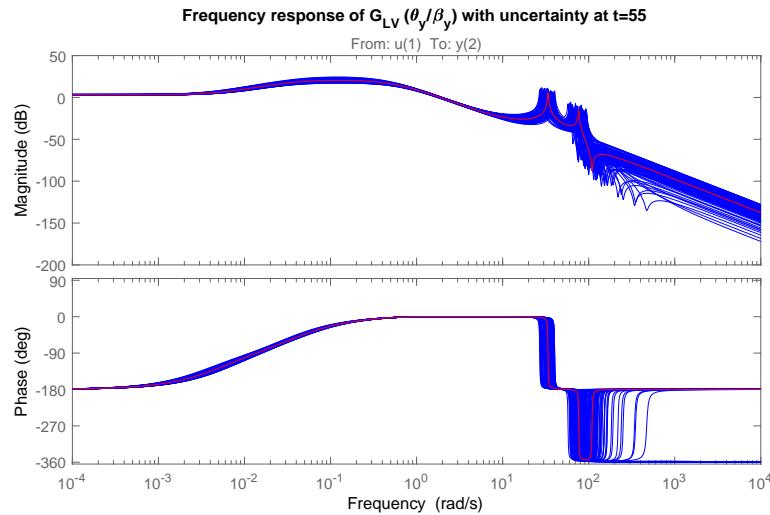
The uncertainties for the RB model are based on [18] where the same launcher was modelled in cooperation with ESA. For the bending modes the numbers were provided by ESA. The frequency response of the θ_y/β_y channel of the model at $t=55$ with random samples of the uncertainty parameters is seen on figure 5.7. The time $t=55$ was chosen as it is at the time of the highest dynamic pressure and the most critical instance of the model.

5.3.2 Delay model

The delay model is a Pade-approximation with an uncertain parameter τ specifying the time delay. The state space used is:

$$\begin{aligned} \dot{x}_\tau &= \begin{bmatrix} 0 & 1 \\ -12/\tau^2 & -6/\tau \end{bmatrix} x_\tau + \begin{bmatrix} 0 \\ -12/\tau \end{bmatrix} u_\tau \\ y_\tau &= \begin{bmatrix} 0 & 1 \end{bmatrix} x_\tau + \begin{bmatrix} 1 \end{bmatrix} u_\tau \end{aligned} \quad (5.18)$$

Where τ is the delay that the model approximates. The nominal value of τ is 40 ms and the uncertainty is 40 % based on numbers from [18] as discussed in section 2.6.4.

**Figure 5.7:** Frequency response of the model G_{LV} at $t=55$ with random samples of the uncertainty parameters. Red response is the nominal response.

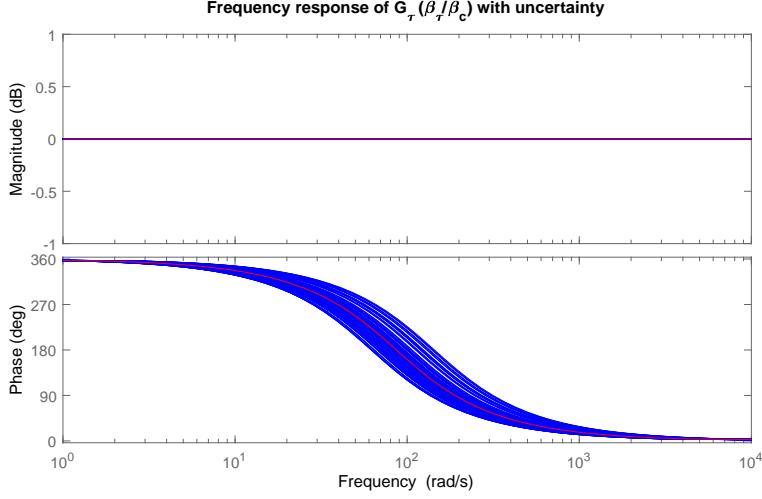


Figure 5.8: Frequency response of the model G_r with random samples of the uncertainty parameter. Red response is the nominal response.

The frequency response is plotted on figure 5.8. The time delay does not vary with the scheduling parameter.

5.3.3 TVC Actuator Model

The TVC actuator model for this project is modelled as a low pass filter with a bandwidth similar to a model used in [20] for the main engine of the Vega LV. While the frequency response was available but not the model, therefore an approximation was made. The model is a 4th order Butterworth-lowpass-filter. The model should capture the most important dynamic of the TVC system which is the bandwidth limitation. The TVC model can easily be replaced in the full model if a better model becomes available. It was decided that building a more complex model was not prioritized. The response can be seen on figure 5.9.

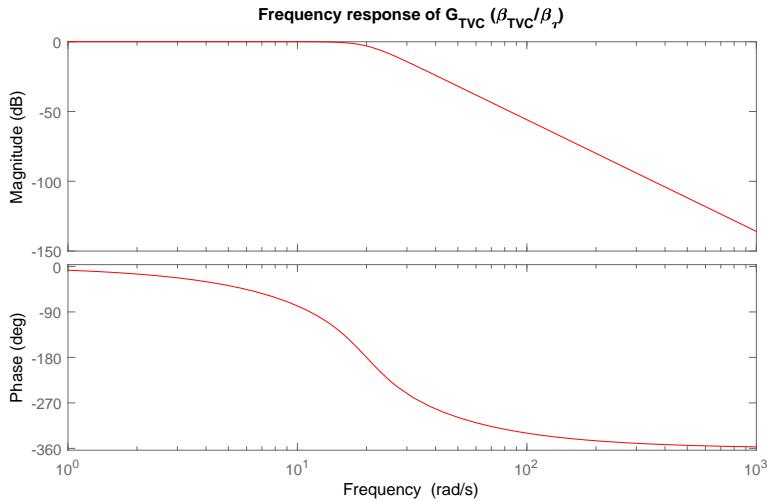


Figure 5.9: Frequency response of the model G_{TVC} .

5.3.4 Dryden wind model

The Dryden wind model described in section 2.6.1 has also been implemented and also changes along the trajectory. The model has been plotted for every 5 seconds of the trajectory and can be seen on figure 5.10.

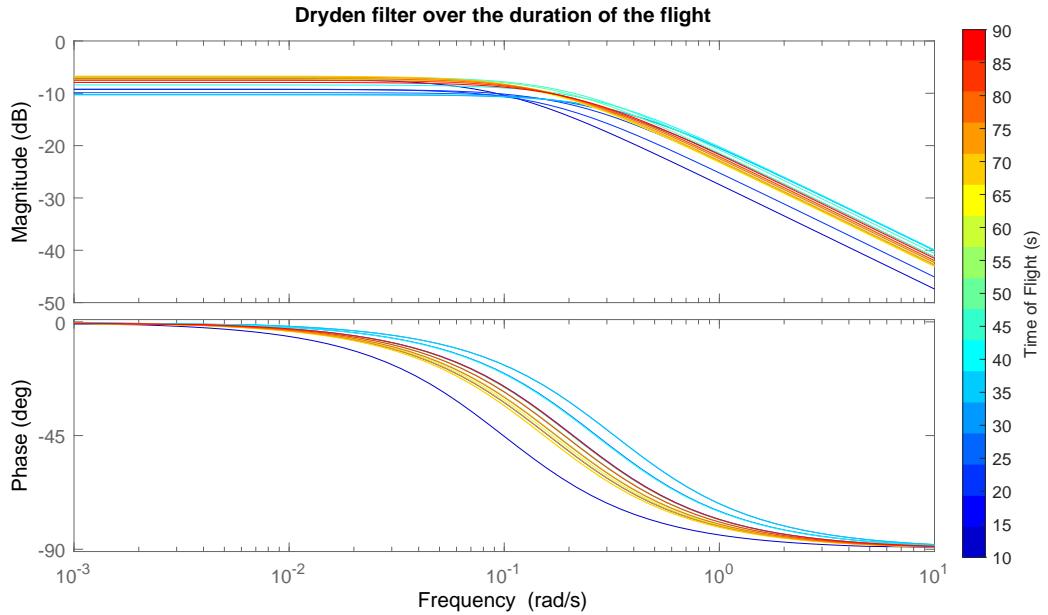


Figure 5.10: Dryden filter along the flight.

This model will help the synthesis prioritize the response to the wind along the trajectory.

5.4 Weight Selection

This section will discuss the choices and designs of the input weights. When choosing weights there is a trade off to be made. While it is possible to add frequency dependent weights, it comes at a cost. Frequency dependency requires more states which will be added to the synthesis model. The controller will have as many states as the synthesis model. This means adding more complex frequency information with the weights will result in a more complex controller and greater compute time both during synthesis but also when implemented. Therefore frequency dependent weights have been kept at a minimum, and only used where it was considered significant.

5.4.1 Command Input Weights

The command input weights are used to scale the feedback channels. For the synthesis they correspond to the maximum command expected. However, for the gravity turn the command signal will be set to zero. The initial weights used are based on [18] which use

the same LV. The Weights are as follows:

$$W_c = \begin{bmatrix} W_{\theta,c} & 0 & 0 & 0 \\ 0 & W_{\omega,c} & 0 & 0 \\ 0 & 0 & W_{r,c} & 0 \\ 0 & 0 & 0 & W_{v,c} \end{bmatrix} = \begin{bmatrix} \frac{\pi}{180} 1 & 0 & 0 & 0 \\ 0 & \frac{\pi}{180} 2.6 & 0 & 0 \\ 0 & 0 & 15 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.19)$$

The weights are not necessary for this part of the trajectory, and the input could be removed entirely. But after talking to our supervisors from ESA, it was decided to use the weights as they can help with the conditioning for the solver when synthesizing.

5.4.2 Wind Input weights

The wind input filter scales the wind disturbance. As the G_W is designed to scale an normalized input to the wind content within one standard deviation, the weight W_w can scale the standard deviation of wind speeds included. The weight will be set to 2 for the purposes of this project which will include about 95% of wind speeds at the severity defined in the filter:

$$W_w = 2 \quad (5.20)$$

5.4.3 Noise Input Weights

The noise input weight is based on the numbers described in section 2.6.3. The number are constant and W_n will therefore be:

$$W_n = \begin{bmatrix} W_{\theta_n} & 0 & 0 & 0 \\ 0 & W_{\theta_n} & 0 & 0 \\ 0 & 0 & W_{rn} & 0 \\ 0 & 0 & 0 & W_{vn} \end{bmatrix} = \begin{bmatrix} \frac{\pi}{180} 0.02 & 0 & 0 & 0 \\ 0 & \frac{\pi}{180} 0.1 & 0 & 0 \\ 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0.001 \end{bmatrix} \quad (5.21)$$

5.4.4 Thrust Uncertainty Input Weight

The thrust uncertainty is based on section 2.6.2 and the value will correspond to a maximum uncertainty in thrust amongst the main engines of 1 %:

$$W_\epsilon = 0.01 \quad (5.22)$$

5.4.5 State Output Weights

The state output weights W_θ , W_ω , W_r and W_v are used to limit the maximum deviation of the states from the equilibrium. These weights are first of all used to capture the requirements for maximum deviation from the trajectory. They are also used to indirectly affect the complementary sensitivity (T) of the system, and because of the inverse relationship with sensitivity it will also affect the sensitivity (S). The goal is to get a low S at lower frequencies where the disturbances influence the system and a low T at higher frequencies where the noise affects the system. Therefore, a filter is placed at the output W_θ to shape T as desired. The cut off is set at 20 rad/s to not limit the bandwidth of the control. The response can be seen of figure 5.11:

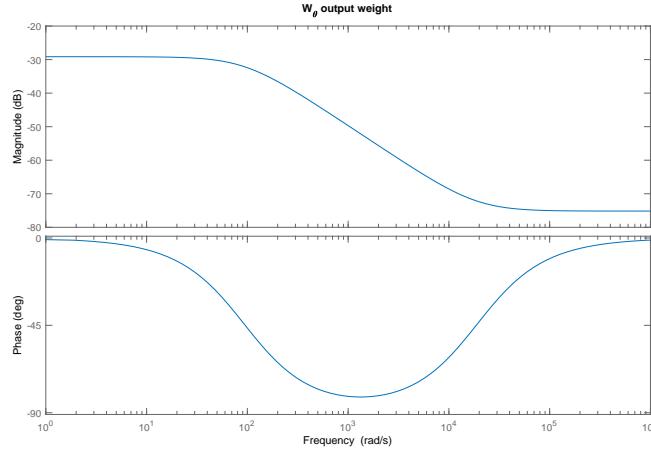


Figure 5.11: Frequency response of W_θ

With this filter the final weight on the states will be:

$$W_n = \begin{bmatrix} W_\theta & 0 & 0 & 0 \\ 0 & W_\omega & 0 & 0 \\ 0 & 0 & W_r & 0 \\ 0 & 0 & 0 & W_v \end{bmatrix} = \begin{bmatrix} \left(\frac{\pi}{180} 2 \frac{0.01s+20}{s+\frac{20}{2}}\right)^{-1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & (500)^{-1} & 0 \\ 0 & 0 & 0 & (15)^{-1} \end{bmatrix} \quad (5.23)$$

Here it should also be noted that the weight on ω is set to zero which will result in no limit for the rate of rotation. This is because there is no apparent reason to limit it that is not already taken into account by some other weight.

5.4.6 Error Output Weights

The output weight $W_{\theta,e}$ is used to put an indirect limit on the sensitivity. The maximum peak of the sensitivity has the following relationship with the GM and PM using classical control theory [27].

$$GM \geq \frac{\|S_\theta(s)\|_\infty}{\|S_\theta(s)\|_\infty - 1} \quad (5.24)$$

$$PM \geq 2 \arcsin \left(\frac{1}{2 \|S_\theta(s)\|_\infty} \right) \quad (5.25)$$

With this, we can indirectly limit the PM and GM of the most important control channel $e_\theta/d_{c,\theta}$. Solving for a gain that will fulfill the targets that was set in the requirements in section 3 will give a gain of 2. With the input to the channel being set to 1° and a gain of $\|S_\theta(s)\|_\infty = 2$ the weight will be:

$$W_{\theta,e} = \left(\frac{\pi}{180} 2 \right)^{-1} \quad (5.26)$$

5.4.7 Load Performance Weight

The weight $W_{Q\alpha}$ is used to limit the structural load on the LV. It is based on the performance parameter described in section 3.1. For simplicity it is held constant along

the trajectory, and the value used is the limit at the time of highest dynamic pressure. The value is therefore selected to be:

$$W_{Q\alpha} = (2.6 \cdot 10^5)^{-1} \quad (5.27)$$

For future work the weight should change along the trajectory as the maximum value for $Q\alpha$ changes a lot. However, it was decided that it should just fulfill the requirement at max-Q for simplicity.

5.4.8 Command Output Weight

The weight W_β will be used to limit the actuation signal. It will have multiple functions, first it is to limit the maximum actuation to 6.5° . Secondly it is used to make the controller itself stable and to reduce the actuators coupling with the bending modes. This was achieved by adding a low-pass-filter and two notch filters to the output. The bandwidth of the low-pass filter was set to start at the first bending mode as well as the frequency of the first notch filter, the second notch filter was placed at the second bending frequency. The weight will then be a combination of those requirements:

The filter changes over the duration of the flight. The response of the weight can be seen on figure 5.12.

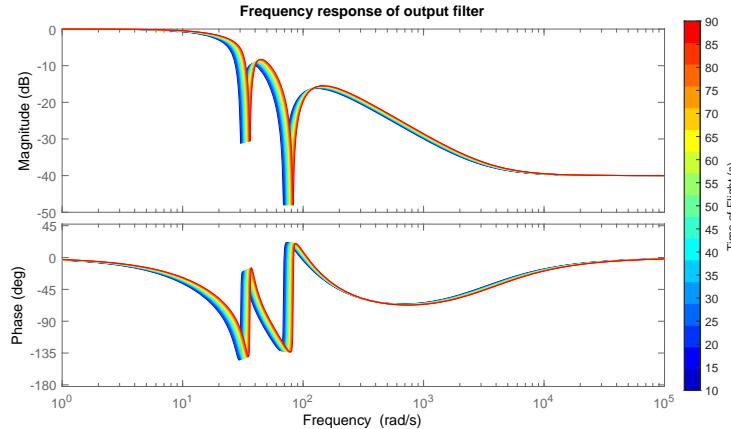


Figure 5.12: Frequency response of output filter

$$W_\beta(s) = \left(\frac{\pi}{180} 6.5 \frac{s^2 + \eta_1 s + (\omega_{BM,1})^2}{s^2 + \frac{\eta_1}{\epsilon_{NF,1}} s + (\omega_{BM,1})^2} \cdot \frac{s^2 + \eta_2 s + (\omega_{BM,2})^2}{s^2 + \frac{\eta_2}{\epsilon_{NF,2}} s + (\omega_{BM,2})^2} \cdot \left(\frac{0.01s + \omega_{BM,1}}{s + \frac{\omega_{BM,1}}{6.5}} \right) \right)^{-1} \quad (5.28)$$

Where $\epsilon_{NF,i}$ is depth of the attenuation from the i 'th notch filter and η_i is the width of the notch filter. $\omega_{BM,i}$ is the center frequency of the filter. It should be noted that for the structured H_∞ synthesis W_β is simply $\frac{\pi}{180} 6.5$ and the filters are instead included as bending filters.

5.5 Synthesis of controllers

This section will go through the process of synthesizing the controllers. The method of gain scheduling will also briefly be explained.

5.5.1 Scheduling of the parameters

The scheduling of parameters used in this project will for simplicity be time, from 10 seconds into the flight until 90 second into the flight. With instances at an interval of 10 seconds. This means there will be a total 9 controllers to interpolate between, using gain scheduling. This interpolation will mean that all systems in between will be estimated. To illustrate this estimation, two examples of the estimated LTI models and real the LTI model, for two time instances, have been chosen. The first is for the time $t=45$. The model is a linear interpolation between the model at $t=40$ and $t=50$. This is around the time of max-Q which is an important part of the trajectory and was therefore selected as an example. The comparison can be seen on figure 5.13:

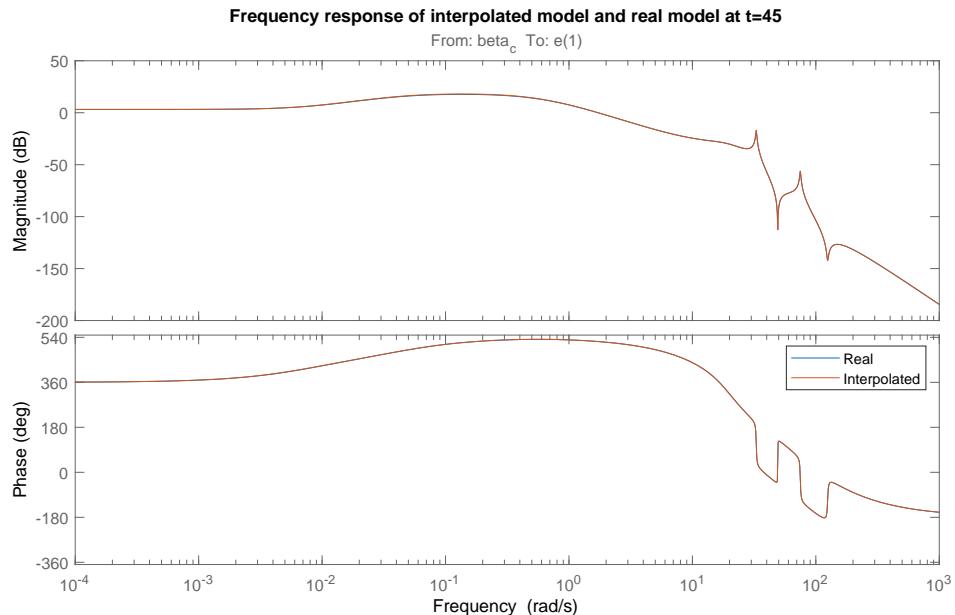


Figure 5.13: Comparison of interpolated and real model at $t=45$ where the LV is under max-Q

This is very good and the worst gain difference is less than 0.5 dB and phase difference less than 0.5°. The second example is at $t=25$. This is around the time where the vehicle is transonic. As described in section 2.5.1, the aerodynamic parameters change the most around this time. The comparison can be seen on figure 5.14:

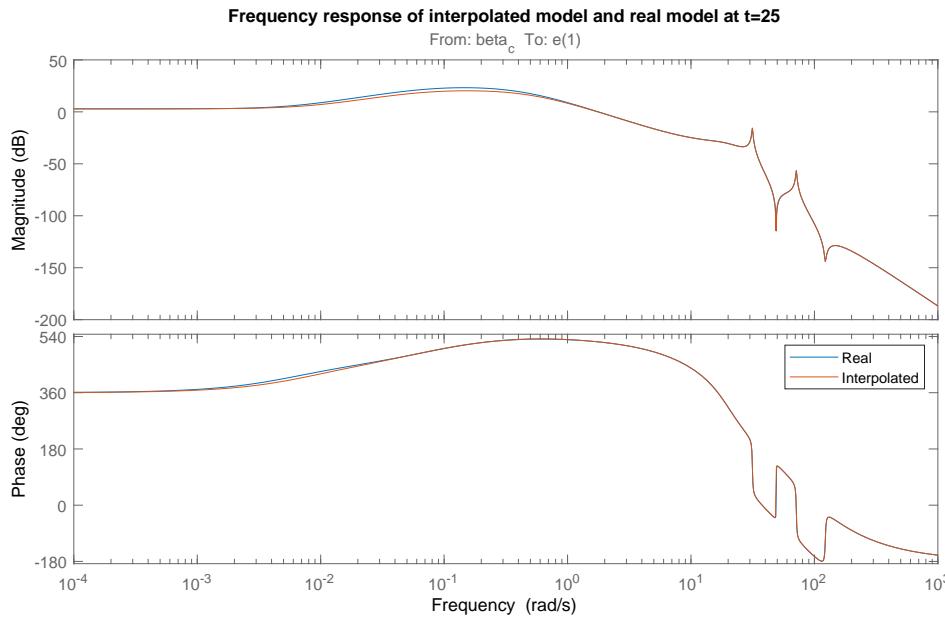


Figure 5.14: Comparison of interpolated and real model at $t=25$ where the LV is transonic.

As expected, the interpolation error is greatest around this time when the vehicle is transonic. The maximum gain difference is more than 3 dB and the phase more than 10°. Therefore, for further development a more careful selection of grid points, like a more dense grid around the time the LV is transonic, should be done. This selection is in itself a large subject and will be considered out of scope for this thesis. For better performance, one could also: Use a different parameter for scheduling such as velocity, have more parameters, have more instances or be more selective about what instances are used for scheduling. This can all potentially improve performance of the controller.

The time period from $t=10$ to $t=90$ is approximately the time of the gravity turn and will be used for this thesis.

5.5.2 Synthesis of the Structured H_∞ Controller

The Structured H_∞ problem is set up as explained in section 5.1.2. Using the model explained in section 5.2. To better find the local solutions of the non-convex problem of the structured H_∞ controller, the problem is solved for every second along the trajectory. This high resolution grid will make it easier to make an initial guess for a controller instead of using random initial starting conditions for every controller. With a higher resolution, controllers for adjacent time instances can be used as an initial guess. An example is using the solution of a controller found for $t=55$ as the initial guess when solving for the controller at $t=54$. The closer these systems are, the greater is the chance that the adjacent controller is a good guess. Here, one second steps was found to give a good trade off between higher resolution grid and using more random starts. Starting with controller at the time of max-Q, the controller is found with 20 random starts from a uniform distribution within the specified limits for each parameter, to increase the chance of finding the optimal controller. Then that controller is used as the initial controller when finding the adjacent controllers.

Those controllers will then be used as the initial guess for their adjacent systems in each direction away from the initial controller. With this iterative method the controllers that are found are usually better suited for gain scheduling by interpolation and more likely the optimal controllers. This method will often not give the best controllers at every instance in the first attempt. But for the cases that seem like outliers, some ad hoc methods can be used. They are identified by looking at the controllers that seem to be significantly worse than the rest or break a pattern. Then one can try other adjacent controllers as initial guesses or try to do more random starts for those specific instances until all the controllers seem to fit a pattern that will make them suitable for gain-scheduling. When doing random starts one will also quickly get an idea of what is a reasonable range for the gain. This can be used to specify a range for the random starts that will greatly improve the chance of finding the optimal controller. However, making these ranges too small might result in excluding the optimal controller and should be done with care. All this extra work will not even guarantee to find a set of controllers that can be gain scheduled and is one of the disadvantages of the approach. It should be noted that the H_∞ control synthesis was mainly used as a stepping stone to developing a LPV controller and to familiarize with the weight augmentation structure and procedure. And a gain scheduled H_∞ controller for the final system was not implemented.

5.5.3 Synthesis of the LPV Controller

The control synthesis for the LPV uses the model explained in section 5.2 rearranged to the standard setup explained in section 5.1.4. The model consists of a gridded structure of the 9 LTI systems according to 9 time-instances. The 9 time instances are from $t=10$ to $t=90$ and every 10 seconds in between. For the synthesis the *LPVTools*[30] toolbox for MATLAB will be used. For the synthesis, a basis function for the Lyapunov function is chosen. The more complex Lyapunov function can make a less conservative controller but will also significantly increase the computing time for the controller. Since the main dynamics of the system such as the velocity enters the system via a quadratic relation to the aerodynamic forces. A good guess for a base function is therefore a second order polynomial as the base function. That is the function of the Lyapunov $P(\rho)$ function with respect to ρ will be:

$$P(\rho) = X_1 + \rho X_2 + \rho^2 X_3 \quad (5.29)$$

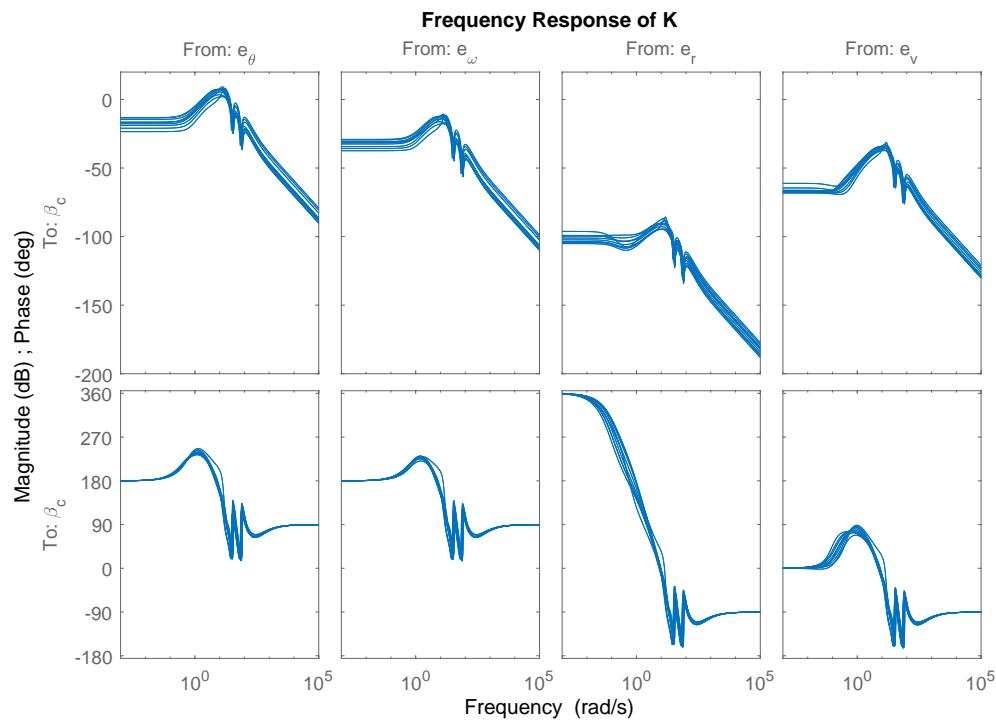
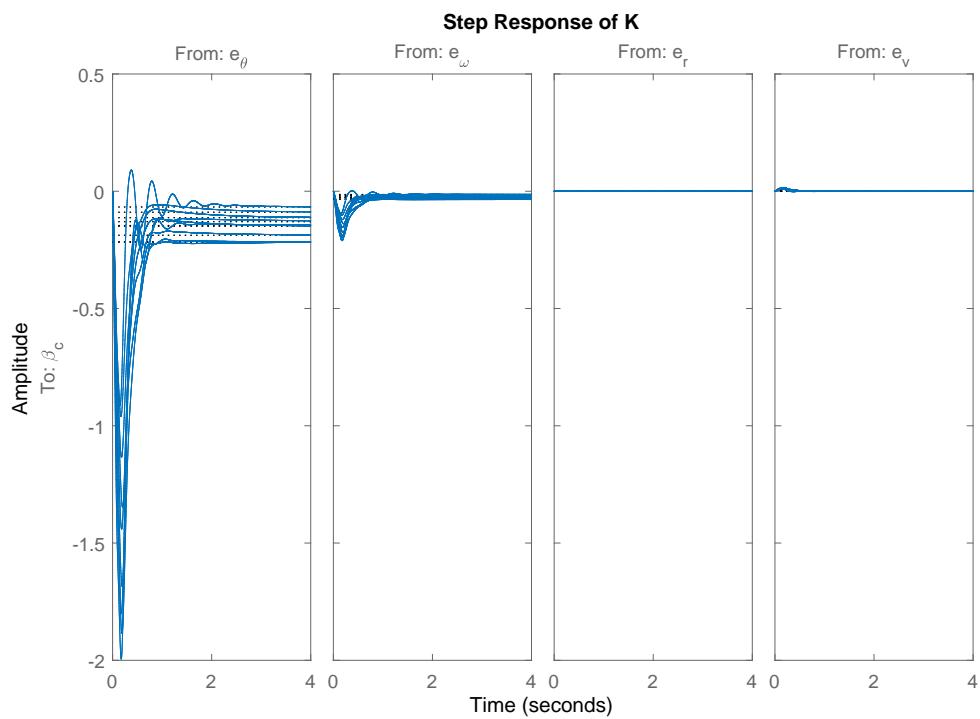
Where X_1 , X_2 and X_3 are matrices with the coefficients of the Lyapunov function that the solver will solve for. The system will be a system of the same order as the weighted system. For the system designed in this project the controller will therefore have 22 states.

Resulting Controller

The resulting controller has 21 states and is stable. Where the pole closest to the imaginary axis is -0.07. The γ from the synthesis is 1.09. For a sense of the controller the following responses will be shown:

The frequency response of the LTI models can be seen on figure 5.15

The step response of the LTI models can be seen on figure 5.16

Figure 5.15: Frequency response of LTI systems from G_K Figure 5.16: Step response of LTI systems from G_K

Testing 6

This chapter will present the tests and results to verify the performance of the controllers.

6.1 LPV controller

This section will present the results from tests of the LPV controller.

6.1.1 Stability

There are multiple ways to check for stability. The first to look at the upper bound γ from equation 5.16, of the L_2 -norm from the synthesis in equation 5.14. With the weights used for this project, the resulting γ is 1.09. This γ implies that beyond being stable, that the controller ensures that the L_2 -gain of the closed-loop system is less than γ for all the allowed parameter trajectories of ρ . To also verify that the frozen LTI instances of the CL is stable the rightmost eigenvalue of all the LTI systems was found to be -0.019. Finally the stability was also verified in an LPV simulation, with the system following a parameter trajectory and with simulated inputs. This will be explained next.

6.1.2 Performance

The first way to gauge the performance of the system is again to look at the γ from the synthesis. With the weights added according to the requirements, the γ should be less than 1 to fulfill the requirements defined by the weights. If the weights and model are good enough the requirements they are based on should also be fulfilled. While this was not entirely achieved it should be noted that it is very close, and loosening the requirements a bit will bring the γ below the 1. Another way to test the requirements is to directly simulate the system with the controller. Ideally, when a real rocket can not be used, a full nonlinear 6 degrees of freedom simulation including all the dynamics should be used. However, this would be a project in itself. Therefore, an LPV simulation available in the *LPVTools* toolbox[30], *LPVSim*, will be used. The *LPVSim* function lets one specify the parameter trajectory, initial states and the inputs to the system. For the inputs, the command signals of d_c will be set to zero, as we are concerned with the gravity turn where all the commands will be zero. The noise will be random samples with zero mean and variance according to section 2.6.3. For input to the Dryden filter d_w the signal will either be white noise with variance of 2, or it will be a constant of 2. This will simulate the very unlikely case that the wind is constant at the maximum magnitude. This constant wind is a typical worst case[18], that the synthesis should have designed for. Same inputs will be used for the engine cluster uncertainty disturbance d_ϵ , but with the magnitude defined in 2.6.2. Finally the trajectory of ρ which will simply be the current time. The

simulation will be carried out with a fixed step-size of 0.001 s. There will be multiple tests for different cases and scenarios to determine the performance. For all of these simulations the requirements in section 3 should be satisfied.

6.1.3 LPV Simulation with Mixed Disturbance and Initial error

This simulation is used as a scenario that shows the system's behavior in a likely scenario. It has an initial error on the pitch channel to show the response of that channel. It has white noise as the input to the wind channel to show the effect of a random wind disturbance. It has a constant error on the engine as it is likely that one engine is simply under performing and therefore constant. The inputs is seen on figure 6.1. Note that while the input to the Dryden filter is white noise, the input to the rest of the system will be filtered.

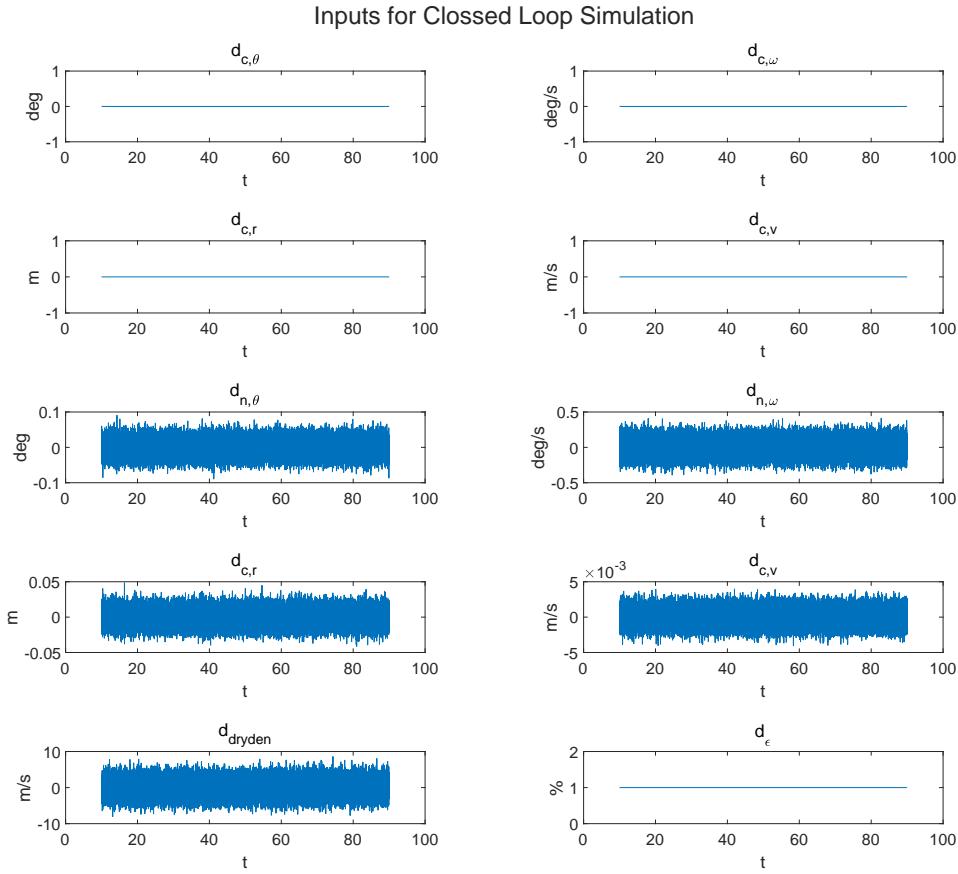


Figure 6.1: Graphs of the input to the closed loop system. The command signals $d_{c,\theta}$, $d_{c,\omega}$, $d_{c,r}$ and $d_{c,v}$ are kept constant and at zero. There is white noise on the four noise channels $d_{n,\theta}$, $d_{n,\omega}$, $d_{n,r}$ and $d_{n,v}$. There is white noise with variance of 2 at the input to the Dryden filter d_{dryden} . The input to the offset disturbance, d_ϵ , simulate a constant engine cluster difference of 1 %

With these inputs results in the outputs seen on figure 6.2

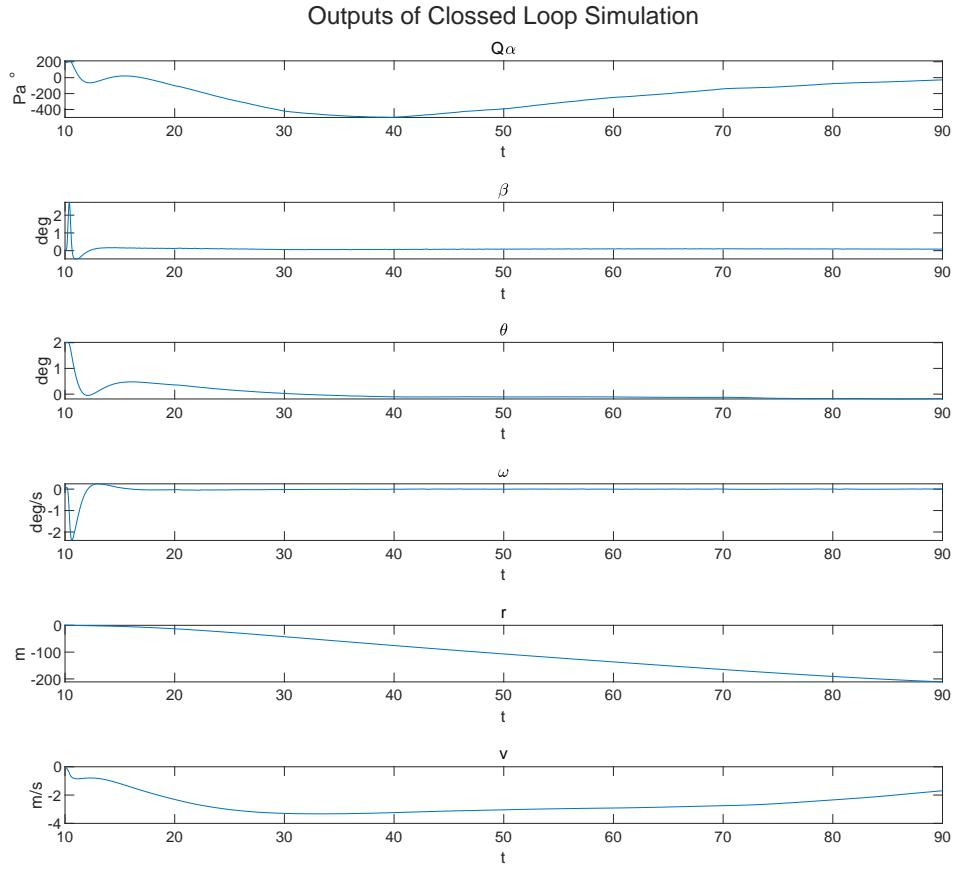


Figure 6.2: Graphs of the outputs from the simulation of the closed loop system with mixed disturbance and an initial error. It shows the performance parameter Q_α . The actuation angle β . The pitch error θ . The angular velocity error ω . The drift r . The drift velocity v .

This simulation satisfies the following requirements:

Requirement	Max	Result	Satisfied
Max actuation	6.5°	2.7°	✓
Integrated actuation	250°	8°	✓
Max Q_α	$2.6 \times 10^5 \text{ Pa}^\circ$	498 Pa°	✓
Max drift	500 m	211 m	✓
Max drift rate	15 m/s	3.3 m/s	✓

Table 6.1: Requirements check for simulation with mixed disturbance and initial error

This simulation shows that under the specified conditions the controller meets the requirements.

6.1.4 LPV Simulation with No Disturbance and Initial error

This simulation is used to check if the error will converge to zero if there is no disturbance as it was a requirement set in section 3. It has an initial error of the pitch channel to show the response of that channel. It has no input to the wind channel and no constant error on the engine. The inputs is seen on figure 6.3.

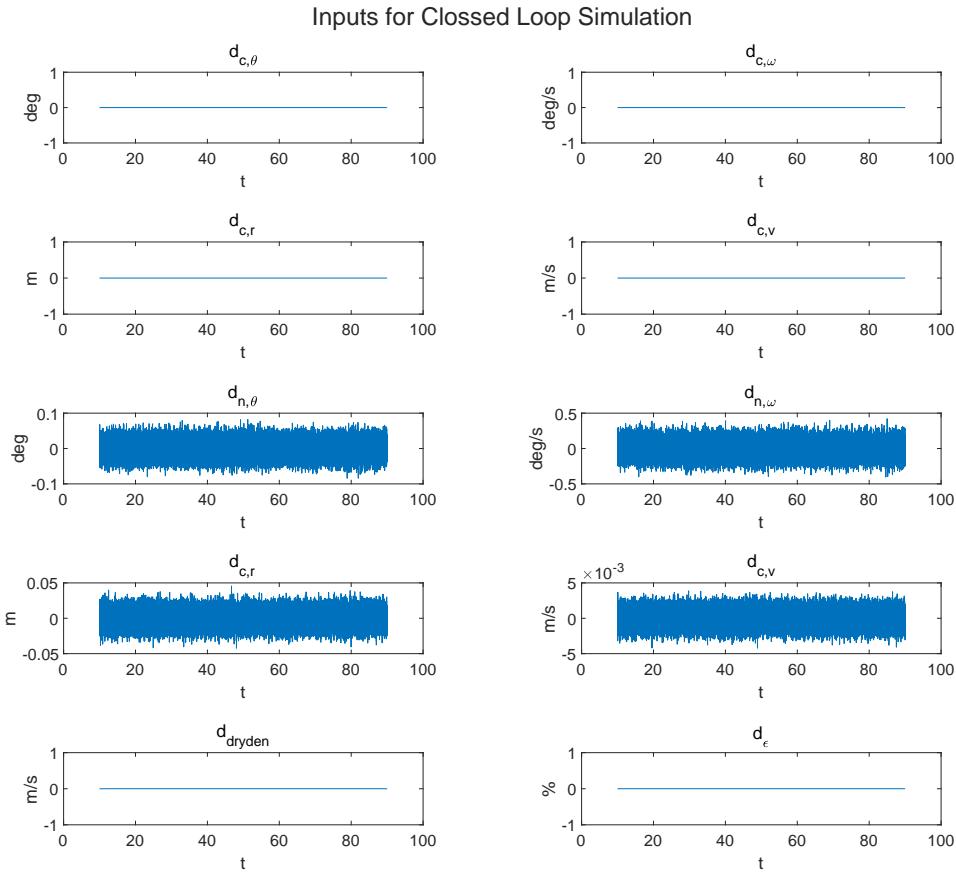


Figure 6.3: Graphs of the input to the closed loop system. The command signals $d_{c,\theta}$, $d_{c,\omega}$, $d_{c,r}$ and $d_{c,v}$ are kept constant and at zero. There is white noise on the four noise channels $d_{n,\theta}$, $d_{n,\omega}$, $d_{n,r}$ and $d_{n,v}$. There is no input to the Dryden filter d_{dryden} . The input to the offset disturbance, d_ϵ , is also set to a constant of zero for this simulation.

With these inputs, and an initial pitch of 2° and drift of -400 m, the resulting outputs can be seen on figure 6.4.

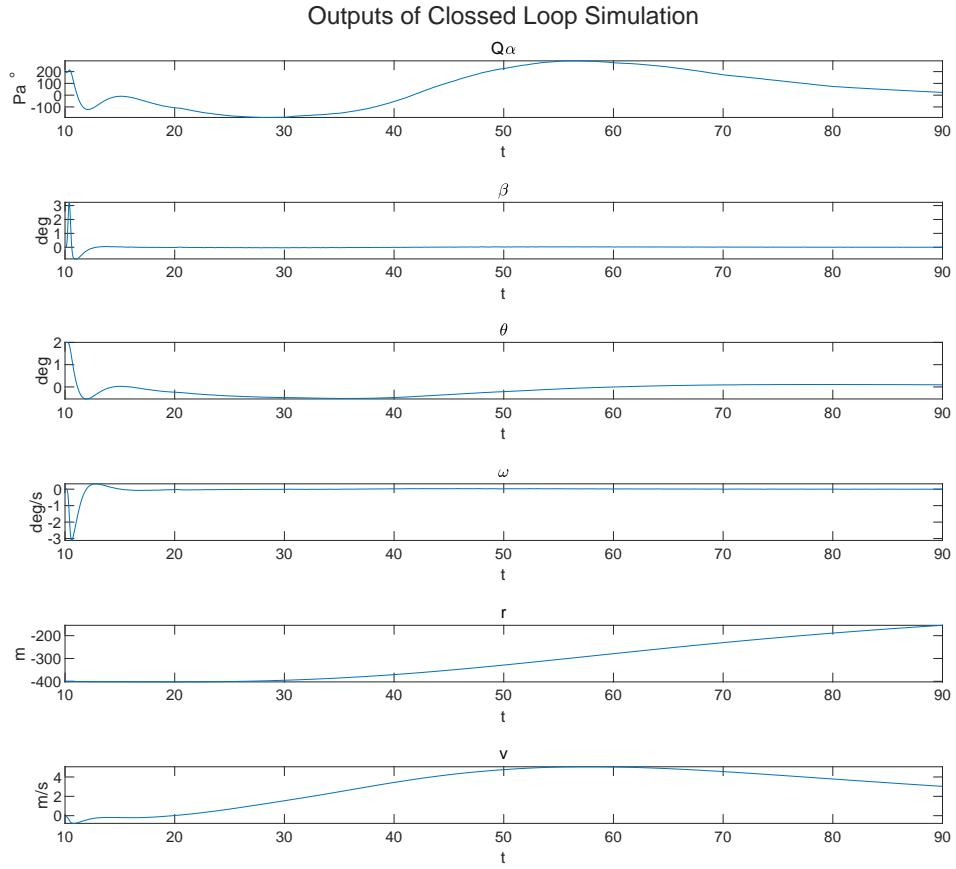


Figure 6.4: Graphs of the outputs from the simulation of the closed loop system with no disturbance and an initial error on both the pitch and drift. It shows the performance parameter $Q\alpha$. The actuation β . The pitch error θ . The angular velocity error ω . The drift r . The drift velocity v .

This simulation satisfies the following requirements:

Requirement	Max	Result	Satisfied
Max actuation	6.5°	3.2°	✓
Integrated actuation	250°	2.9°	✓
Max $Q\alpha$	$2.6 \times 10^5 \text{ Pa}^\circ$	291 Pa°	✓
Max drift	500 m	402 m	✓
Max drift rate	15 m/s	5 m/s	✓

Table 6.2: Requirements check for simulation with no disturbance and initial errors

This also shows that the states of the system do converge towards zero error with no disturbance.

6.1.5 LPV simulation with random disturbance

This simulation is to test the tracking during random disturbance. It has the random inputs seen on figure 6.5. Note that while the input to the Dryden filter is white noise, the input to the rest of the system will be filtered.

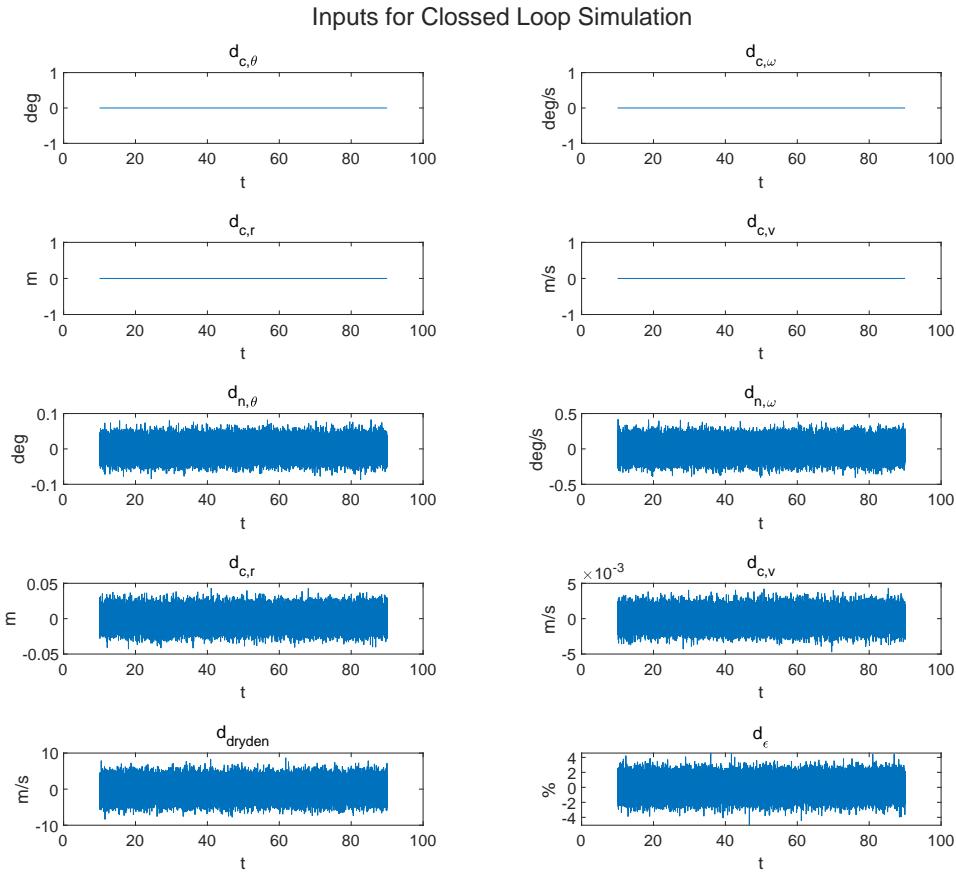


Figure 6.5: Graphs of the input to the closed loop system. The command signals $d_{c,\theta}$, $d_{c,\omega}$, $d_{c,r}$ and $d_{c,v}$ are kept constant and at zero. There is white noise on the four noise channels $d_{n,\theta}$, $d_{n,\omega}$, $d_{n,r}$ and $d_{n,v}$. There is white noise with variance of 2 at the input to the Dryden filter d_{dryden} . The input to the offset disturbance, d_{ϵ} , simulates a white noise engine cluster difference with mean of 0% and variance of 1 %.

With these inputs and all zero initial states, the results of the outputs can be seen on figure 6.6

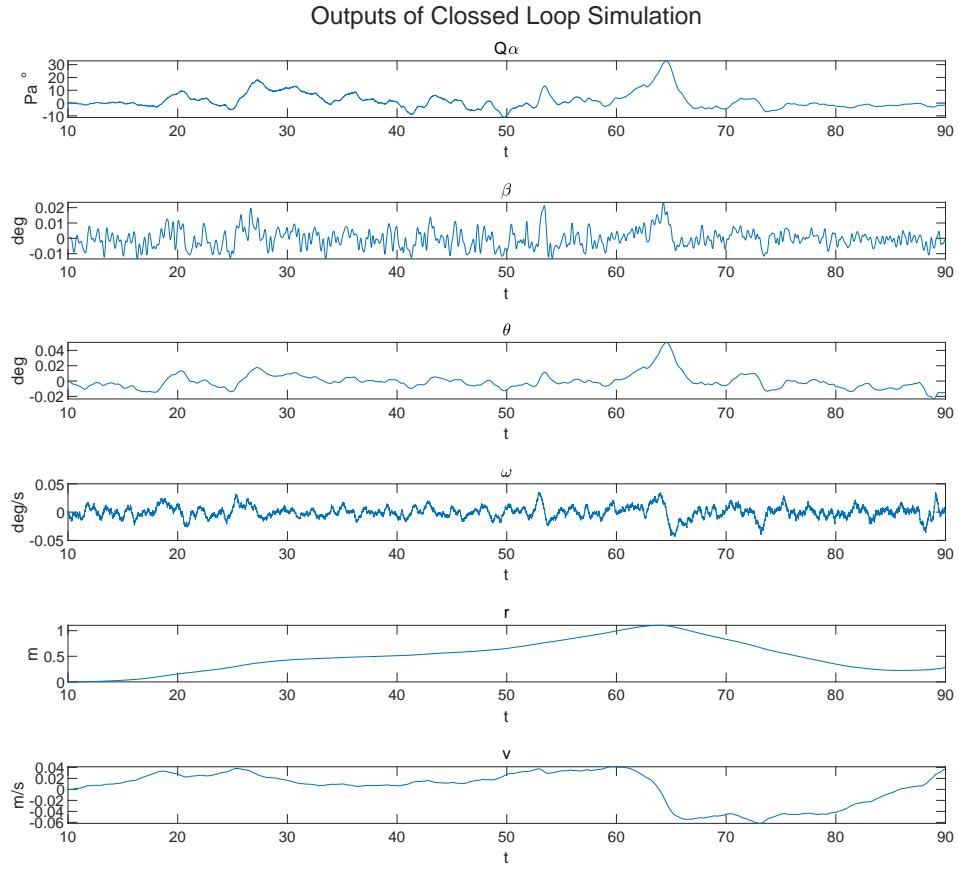


Figure 6.6: Graphs of the outputs from the simulation of the closed loop system with random disturbance and no initial errors. It shows the performance parameter Q_α . The actuation β . The pitch error θ . The angular velocity error ω . The drift r . The drift velocity v .

This simulation satisfies the following requirements:

Requirement	Max	Result	Satisfied
Max actuation	6.5°	0.02°	✓
Integrated actuation	250°	0.35°	✓
Max Q_α	$2.6 \times 10^5 \text{ Pa } ^\circ$	$32 \text{ Pa } ^\circ$	✓
Max drift	500 m	1.1 m	✓
Max drift rate	15 m/s	0.06 m/s	✓

Table 6.3: Requirements check for simulation with random disturbance

This simulation verifies that with these random disturbances, the system satisfies all the set requirements.

6.1.6 LPV simulation with constant disturbance

This simulation with constant disturbance has inputs is seen on figure 6.7. This simulation is done because a likely worst case is when the wind is constant. When the wind is random

it will average out the effect on the LV. When it is constant it will keep pushing it off the trajectory the whole way through. And since it is a part of the scenarios the synthesis takes into account, it is worth simulating.

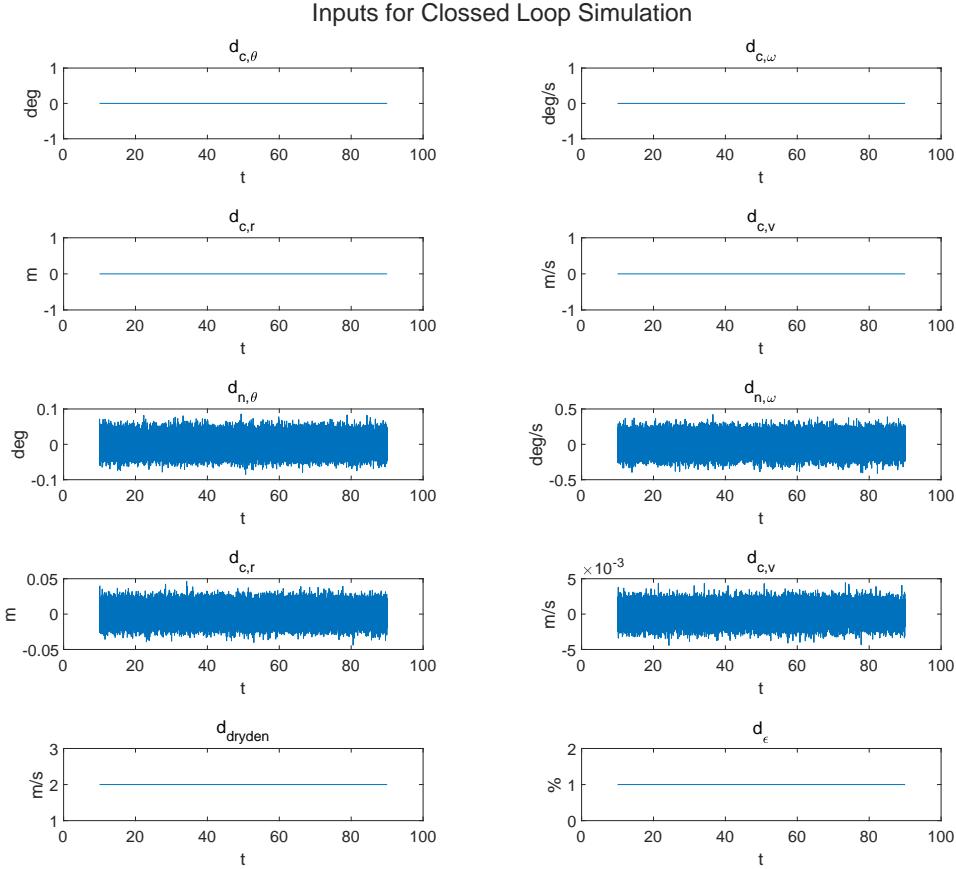


Figure 6.7: Graphs of the input to the closed loop system. The command signals $d_{c,\theta}$, $d_{c,\omega}$, $d_{c,r}$ and $d_{c,v}$ are kept constant and at zero. There is white noise on the four noise channels $d_{n,\theta}$, $d_{n,\omega}$, $d_{n,r}$ and $d_{n,v}$. There is a constant input of 2 at the input to the Dryden filter d_{dryden} . The input to the offset disturbance, d_ϵ , simulates constant engine cluster difference of 1%.

With these inputs and all states initialized at zero, the results of the outputs is seen on figure 6.8.

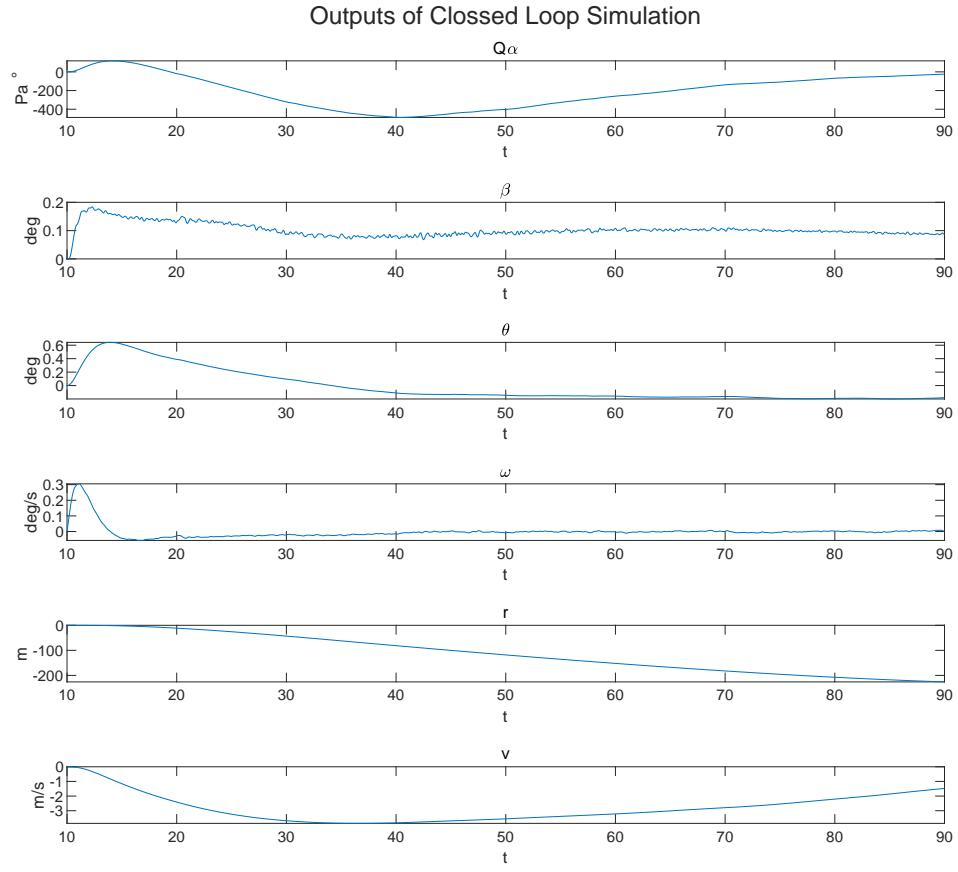


Figure 6.8: Graphs of the outputs from the simulation of the closed loop system with constant disturbance and no initial errors. It shows the performance parameter Q_α . The actuation angle β . The pitch error θ . The angular velocity error ω . The drift r . The drift velocity v .

This simulation satisfies the following requirements:

Requirement	Max	Result	Satisfied
Max actuation	6.5°	0.18°	✓
Integrated actuation	250°	8.2°	✓
Max Q_α	$2.6 \times 10^5 \text{ Pa}^\circ$	488 Pa°	✓
Max drift	500 m	226 m	✓
Max drift rate	15 m/s	3.8 m/s	✓

Table 6.4: Requirements check for simulation with constant disturbance

Additionally this simulation verifies that even though a constant disturbance has a larger effect on the system, it still satisfies all the requirements.

6.2 Test Conclusion

All the tests were successful and satisfied the set requirements. The first test shows that even if there is a small error when starting the gravity turn the controller can correct and follow the gravity turn while under disturbance. The second test shows that with initial error and no disturbance the LV will converge to zero. While the drift distance converges slowly it is not as important as it doesn't affect the dynamics significantly, but should just be kept within the 500 meters specified by the requirements. The disturbance tests show that it can keep the output channels within the specified requirements, both with random disturbance, and with constant disturbances.

Conclusion 7

7.1 Discussion

This section will discuss some of the results, decisions and circumstances developed in this thesis.

7.1.1 ESA Collaboration and Multiple engine Control

This thesis is in collaboration with ESA, due to ESA having specific interests in the subjects of this thesis. Specifically they are interested in the allocation of the TVC system when using multiple engines. This is mainly because most other subjects covered in this thesis have already been explored in previous work[18][23]. The effect of multiple main engines is therefore explored while developing the modeling. The conclusion is that for the level of control applied in this thesis, the fact that there are multiple engines does not change the linear dynamics for the pitch and yaw plane. This can be attributed to the linear control input for each engine being the same no matter the position on the bottom of the rocket in the linear model. Thus there is no difference from just having one engine in terms of how the control signal enters the linear system. This is also concluded with a different method in a previous project[13]. The disturbance from difference in thrust is however taken into account. The other potential subjects to explore regarding multiple engines is not explored in this thesis. The primary subject to be explored would be the control of the roll of the rocket. This should be straightforward with the model that is build as it included the states that would be controlled, and the individual engines' effect on these states. While the next steps are outlined clearly in theory, implementing this went beyond the scope of this thesis. In regards to the collaboration, assistance from Automatic Control Systems Analyst Finn Ankersen and GNC Systems Engineer Pedro Simplício is greatly appreciated. Both with providing data and material on the relevant subjects, while also sharing their experience and insights on the subject.

7.1.2 Uncertainties are not Used for Control Synthesis

The uncertainties were modeled using LFTs and were simulated with random samples. However, it is not possible to make a synthesis using just the small gain theorem that the LFT modeling is used for. To include uncertainties in a less conservative manner would require IQC theory that is decided to be out of scope for this project. But even the early tests using H_∞ control already made clear that the compute time quickly reached impractical levels using the D-K iteration for large LTI systems. Therefore, there is no real measure of the robustness of the system.

7.1.3 Prioritization of Objectives

This thesis prioritized including more dynamics into the control, instead of designing a simpler controller for the whole duration of the flight. This is decided in agreement with ESA and supervisor, resulting in a more realistic design process, but for a smaller problem. Therefore, even though the motivation for the project is the design of a controller used for landing a rocket booster, it is not developed. Instead, a more realistic and wide set control theory and complex model building is explored successfully.

7.1.4 Simulation

Initially it was planned to develop a nonlinear 6 DOF simulation to verify the controller. Such a simulation with some of the basic dynamics has already been developed in a previous project[13]. However, it was advised against spending too much time on this activity. A simulation would be hard to verify even if it was developed. As with any other implementation it is also very hard to estimate the development time. Since a simulation would take time from developing a controller, it was decided to not attempt to develop it and instead use linear simulations to verify the controller. Using a linear simulation will not tell if the linear estimates used for the controller synthesis will hold. Simulating the controller with the same model the controller is synthesised from will also not necessarily show if there are any mistakes in the modeling because of the nature of the synthesis. It is therefore to be expected that if the closed loop model is stable, even if the model is wrong, the controller will perform well. However, this is the nature of working with a subject like this, there is no way to just test it on a real rocket, and most simulators with the fidelity needed to test the system are not readily available.

7.1.5 Future Work

Already after the initial problem analysis it became apparent that designing a controller for a LV is a difficult and complex task. So while a plan with a path to make a controller for landing a booster was made, it is clearly not possible to realize this within the scope of this thesis. The implementation is therefore made in a way that changing it or including the necessary parts for the remaining flight, have been planned in the design. While the next objective on the list would be to make the controller robust to uncertainties, this would be an entirely new subject, and seems rather complex with some difficulties already apparent. The more natural steps would be to first make all the controllers for pitch, yaw and roll. This should be straightforward as the models are already made, although it might be time consuming to implement. It should also be straightforward to use the model containing the states of the pitch, yaw and roll controller, to implement a model with the states of all three planes in one. This would be instead of designing controllers that would control the individual planes. This might however result in exponentially longer synthesis times. For the nonlinear simulation, the core functionality is already implemented, and a plan for the remaining dynamics was made at the start of the semester. But again this could end up being very time consuming. Finally, the remaining parts of the trajectory should be analysed. It should be considered how the dynamics and the scheduling should be modified when the assumptions that were used during the gravity turn no longer apply.

7.2 Conclusion

This thesis sought to develop an LPV controller for the altitude control of a rocket booster. First, an analysis of rocket dynamics and the control problem were done. Based on this analysis, a prioritized set of objectives was formulated. The objectives were prioritized towards a controller with more dynamics, but for the gravity turn only.

A rigid-body model was developed using vectors and reference frames that could be used for all three control planes. This modeling framework makes modifying or adding dynamics easier, and it is ready for use in a controller that controls the pitch, yaw and roll planes individually or all at once. The pitch model was first augmented with bending modes. Sub models for actuation, delay and wind were added to this model. It was then connected with the relevant inputs for command, noise and disturbances. Then the weights were added to specify the requirements for the synthesis. While adding weights and more sub models to the system, the structured H_∞ was used to synthesise LTI controllers using the interconnection as a stepping stone to the LPV. The LPV controller was then synthesized using a gridded set. The controller was at first internally unstable, to solve this issue, a weight was added to the output that made the controller internally stable.

The controller was finally tested in an LPV simulation and was proven to be stable. All the requirements were met in simulation, but the L_2 -norm upper bound γ was slightly above one at 1.09. This means that there might be some configuration that was not tested that could break the requirements a bit. However, with the test results the controller fulfills the requirements. And thus, an LPV controller for the gravity turn was successfully developed and tested.

Bibliography

- [1] Worldwide commercial space launches. <https://www.bts.gov/content/worldwide-commercial-space-launches>. Accessed: 2021-02-03.
- [2] Space launch report. <https://www.spacelaunchreport.com/>. Accessed: 2021-02-03.
- [3] Timeline of spaceflight. https://en.wikipedia.org/wiki/Timeline_of_spaceflight#cite_ref-Yomiuri20200913_1-0. Accessed: 2021-02-03.
- [4] H. Jones. The recent large reduction in space launch cost. 48th International Conference on Environmental Systems, 2018.
- [5] Esa website. [www.http://www.esa.int/](http://www.esa.int/). Accessed: 2021-06-04.
- [6] Nasa spinoff technologies. https://en.wikipedia.org/wiki/NASA_spinoff_technologies. Accessed: 2021-08-03.
- [7] Team ASTRA. Asteroid mining, technologies roadmap and applications, 2010.
- [8] A. Humbert S. Heinrich and R. Amiel. Greenspace: Recovery reusability scenarios for launcher industry, 2018.
- [9] Wikipedia. Space transportation system. https://en.wikipedia.org/wiki/Space_Transportation_System. Accessed: 2021-02-04.
- [10] Wikipedia. Blue origin. https://en.wikipedia.org/wiki/Blue_Origin. Accessed: 2021-02-04.
- [11] Wikipedia. Virgin galactic. https://en.wikipedia.org/wiki/Virgin_Galactic. Accessed: 2021-02-04.
- [12] Wikipedia. SpaceX. <https://en.wikipedia.org/wiki/SpaceX>. Accessed: 2021-02-04.
- [13] Frederik Rasmussen, Jakob Stentoft Poulsen and Mikkel Damgård Hardysøe. $\mathcal{H}\infty$ - and lpv control for a modified vega launch vehicle in simulation, 2020.
- [14] Arianespace. *Vega User's Manual Issue 4 - Revision 0*. Arianespace, Boulevard de l'Europe, France, 2014.
- [15] R. F. Vause Vause and Brett R. Starr Starr. Trajectory-based loads for the ares i-x test flight vehicle. 2011.
- [16] Peter H. Zipfel. *Modeling and Simulation of Aerospace Vehicle Dynamics, Second Edition*. American Institute of Aeronautics and Astronautics, 1 2007.
- [17] Wikipedia. Euler's equations (rigid body dynamics). [https://en.wikipedia.org/wiki/Euler%27s_equations_\(rigid_body_dynamics\)](https://en.wikipedia.org/wiki/Euler%27s_equations_(rigid_body_dynamics)). Accessed: 2021-27-04.

- [18] D. Navarro-Tapia. *Robust and Adaptive TVC Control Design Approaches for the VEGA Launcher*. University of Bristol, 2019.
- [19] D. L. Johnson. *Terrestrial Environment (Climatic) Criteria Guidelines for Use in Aerospace Vehicle Development*. George C. Marshall Space Flight Center Marshall Space Flight Center, Alabama, 1993.
- [20] Pedro V M Simplicio, Samir Bennani, Xavier Lefort, Andres Marcos, and Christophe Roux. Structured singular-value analysis of the vega launcher in atmospheric flight. *Journal of Guidance, Control, and Dynamics*, 39(6):1342–1355, June 2016.
- [21] Christophe Roux and I Cruciani. Scheduling schemes and control law robustness in atmospheric flight of vega launcher. 05 2021.
- [22] Jeb S. Orr, Matthew D. Johnson, Jonathan D. Wetherbee, and James H. McDuffie. State space implementation of linear perturbation dynamics equations for flexible launch vehicles. American Institute of Aeronautics and Astronautics Inc., 2009.
- [23] P.V.M. Simplicio. *Guidance and Control Elements for Improved Access to Space: From Planetary Landers to Reusable Launchers*. University of Bristol, 2019.
- [24] J. Doyle, A. Packard, and K. Zhou. Review of lfts, lmis, and mu. In */1991/ Proceedings of the 30th IEEE Conference on Decision and Control*, pages 1227–1232 vol.2, 1991.
- [25] J.C. Doyle, K. Glover, P.P. Khargonekar, and B.A. Francis. State-space solutions to standard h_2 and h_∞ control problems. *IEEE Transactions on Automatic Control*, 34(8):831–847, 1989.
- [26] Pascal Gahinet and Pierre Apkarian. A linear matrix inequality approach to h control. *International Journal of Robust and Nonlinear Control*, 4(4):421–448, 1994.
- [27] Sigurd Skogestad and Ian Postlethwaite. *Multivariable Feedback Control: Analysis and Design*. John Wiley Sons, Inc., Hoboken, NJ, USA, 2005.
- [28] P. Apkarian and D. Noll. Nonsmooth h synthesis. *IEEE Transactions on Automatic Control*, 51(1):71–86, 2006.
- [29] Janardhanan Sivaramakrishnan. Linear time-variant systems - introduction, 10 2013.
- [30] Arnar Hjartarson, Peter Seiler, and Andrew Packard. Lpvtools: A toolbox for modeling, analysis, and synthesis of parameter varying control systems. *IFAC-PapersOnLine*, 48(26):139–145, 2015. 1st IFAC Workshop on Linear Parameter Varying Systems LPVS 2015.
- [31] Christian Hoffmann and Herbert Werner. A survey of linear parameter-varying control applications validated by experiments or high-fidelity simulations. *IEEE Transactions on Control Systems Technology*, 23(2):416–433, 2015.
- [32] Corentin Briat. *Linear Parameter-Varying and Time-Delay Systems*, volume 3. 2015.
- [33] Shu Wang, Harald Pfifer, and Peter Seiler. Robust synthesis for linear parameter varying systems using integral quadratic constraints. *Automatica*, 68:111–118, 2016.

Appendix A

A.1 Parameter Trajectories

This appendix contains the graphs of the varying parameters of the model.

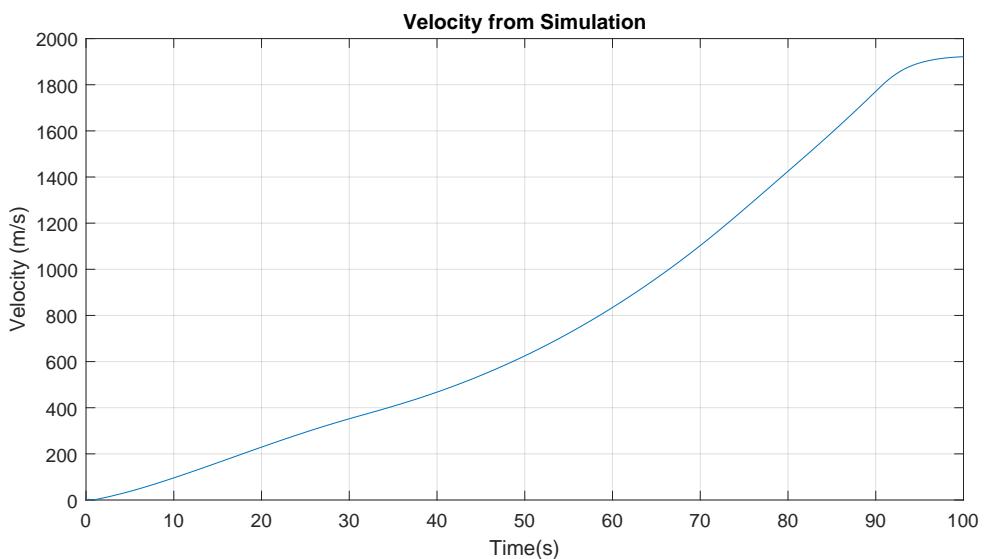


Figure A.1: Velocity parameter trajectory from simulation.

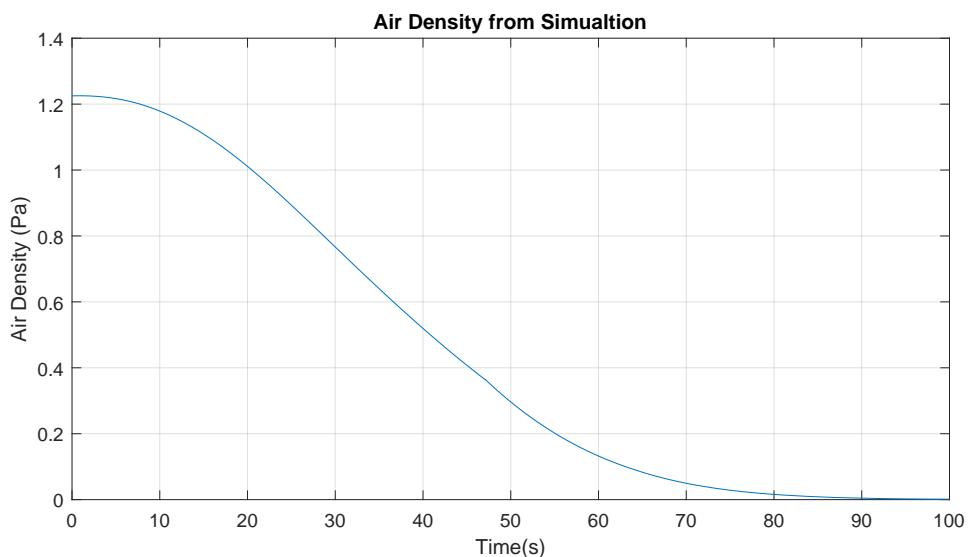


Figure A.2: Air Density parameter trajectory from simulation.

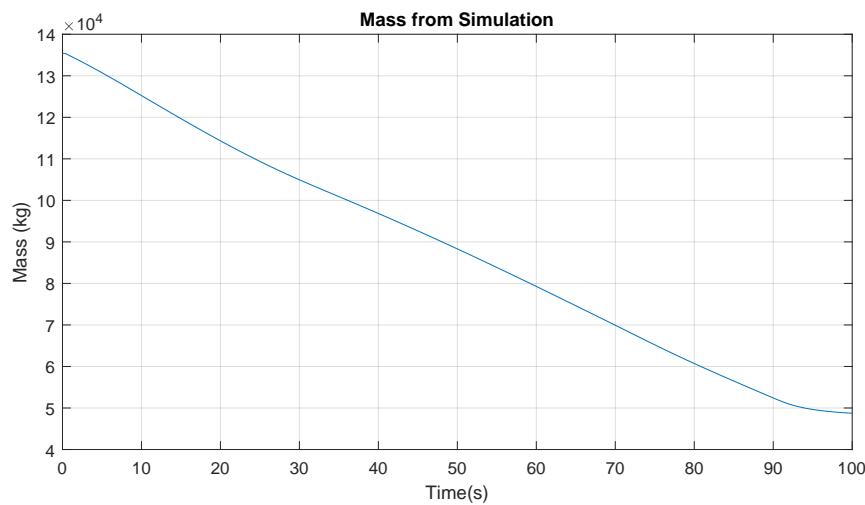


Figure A.3: Mass parameter trajectory from simulation.

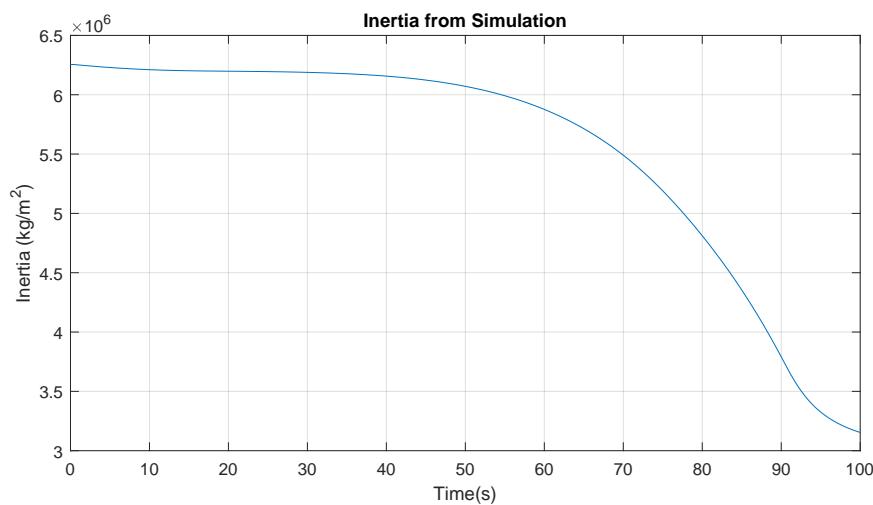


Figure A.4: Moment of Inertia parameter trajectory from simulation.

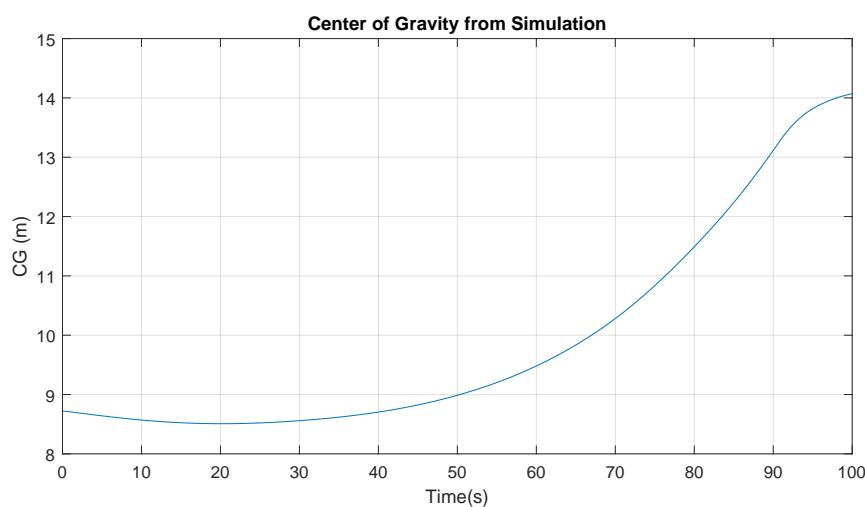


Figure A.5: Center of Gravity parameter trajectory from simulation.

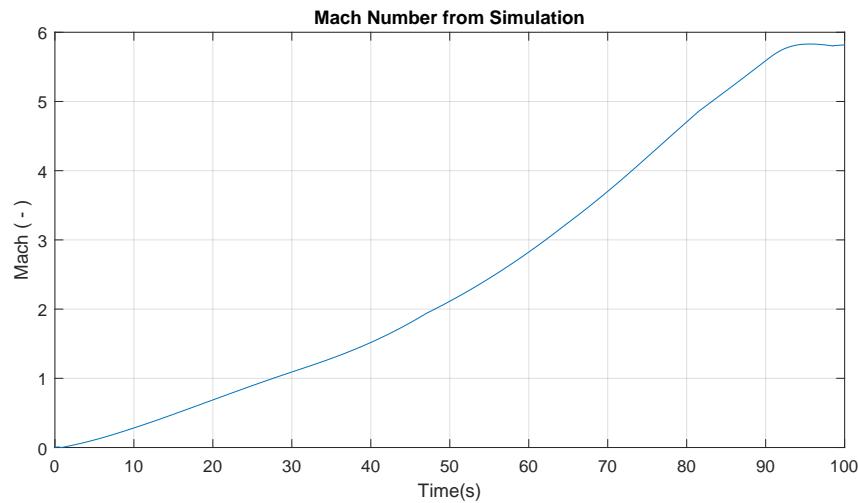


Figure A.6: Mach parameter trajectory from simulation.

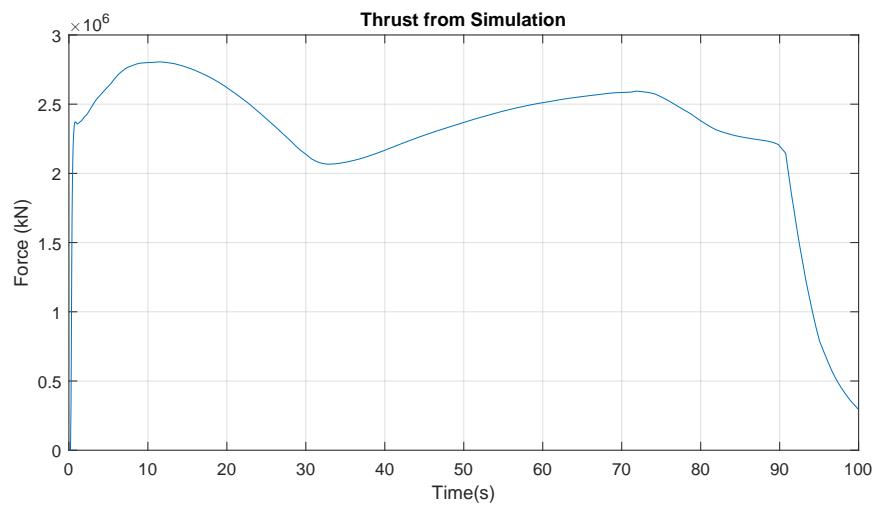


Figure A.7: Thrust parameter trajectory from simulation.