Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Control Engineering

Stabilization and control of a vertically landing launcher

Master's Thesis

Bc. Richard Bláha

Master programme: Cybernetics and Robotics
Branch of study: Cybernetics and Robotics
Supervisor: Doc. Ing. Martin Hromčík, Ph.D.

Prague, May 2020

Thesis Supervisor:
Doc. Ing. Martin Hromčík, Ph.D.
Department of Control Engineering
Faculty of Electrical Engineering
Czech Technical University in Prague
Karlovo náměstí 13
121 35 Prague 2
Czech Republic

Declaration

I hereby declare I have written this doctoral thesis independently and quoted all the sources of information used in accordance with methodological instructions on ethical principles for writing an academic thesis. Moreover, I state that this thesis has neither been submitted nor accepted for any other degree.

In Prague, May 2020

.....
Bc. Richard Bláha

MASTER'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Bláha Richard** Personal ID number: **406276**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Control Engineering**

Study program: **Cybernetics and Robotics**

Branch of study: **Cybernetics and Robotics**

II. Master's thesis details

Master's thesis title in English:

Stabilization and control of a vertically landing launcher

Master's thesis title in Czech:

Stabilizace a řízení kolmo přistávajícího raketového nosiče

Guidelines:

Stabilization and control of a vertically landing launcher
The goal of the thesis is to design and validate flight control laws for stabilization and guidance of a vertically landing launcher. The topic is motivated by a recent ESA call dedicated to this problematic to which the Department of Control Engineering was invited recently by a European research consortium. The tasks comprise building and validation of the simulation models suitable for control designs and verifications, selection of suitable control architectures and design methodologies, and parametrization of the controllers.
1. Construct simulation mathematical models in SIMULINK environment.
Consider important aerodynamic effects. Perform linearization, validate the model, elaborate the sensitivity of the model to uncertainties in the physical parameters of the launcher.
2. Give specific and quantitative recommendations regarding actuators and sensors placement and parameters. Base your judgments on the simulation model and on realistic assumptions on the parameters of actuating and sensing systems.
3. Propose design methodologies and architectures suitable for stabilization and control of the launcher during the landing phase.
4. Design/parametrize the controller for selected values of physical parameters.
5. Validate performance and robustness of the control laws by means of robust control techniques, and extensive numerical simulations.

Bibliography / sources:

Blakelock, Automatic control of aircraft and missiles, Prentice Hall, 1998 Bryson Jr., Control systems for aicraft and spacecraft, Wiley, 2002

Name and workplace of master's thesis supervisor:

doc. Ing. Martin Hromčík, Ph.D., Department of Control Engineering, FEE

Name and workplace of second master's thesis supervisor or consultant:

Date of master's thesis assignment: **28.08.2019** Deadline for master's thesis submission: **22.05.2020**

Assignment valid until:
by the end of summer semester 2020/2021

doc. Ing. Martin Hromčík, Ph.D.	prof. Ing. Michael Šebek, DrSc.	prof. Mgr. Petr Páta, Ph.D.
Supervisor's signature	Head of department's signature	Dean's signature

III. Assignment receipt

The student acknowledges that the master's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master's thesis, the author must state the names of consultants and include a list of references.

. _____
Date of assignment receipt

_____
Student's signature

Abstract

This thesis is inspired by the open tender for European Space Agency (ESA) regarding the project of developing a Guidance, Navigation and Control (GNC) system framework for re-usable space transportation systems. The Department of Control Engineering at Czech Technical University was invited to participate in this project from the control systems point of view. An overall GNC & Attitude Control System (ACS) architecture divided down into three layers was subsequently proposed. The first layer was focused on mission level GNC, the second on the vessel level ACS algorithms, and finally, the third layer was focused on covering the actuation level control systems.

This thesis's goal is mainly the development of the ACS system with a slight overlap into the third layer, specifically to examine the use and placement of possible actuators and sensors for such a vehicle. In order to accomplish this, a 2D model in SIMULINK is built and later it is enhanced by the suited actuators and control techniques. The final version is then implemented into a 3D, more complex and detailed, SIMULINK model that was already provided but at the end heavily modified.

Keywords: Rocket, launcher, vertical stabilization, landing, TVC, RCS, ACS, Attitude control, PID, LQR

Tato práce byla inspirována otevřeným tendrem pro ESA související s projektem pro vývoj systémového GNC frameworku pro znovu použitelné vermírné transportní prostředky. Katedra řídící techniky na ČVUT byla k tomuto projektu přizvána. Následně byla navržena celková architektura GNC a ACS, následně rozdělena do tří dílčích částí. První byla zaměřena na GNC z pohledu celkové mise, druhá část se soustředila na ACS samotné rakety a na závěr, poslední část se zabývala řídícími systémy pro použité aktuátory.

Cílem této diplomové práce je převážně vývoj ACS systému s lehkým přesahem do třetí části, konkrétně prozkoumat možné umístění použitých aktuátorů a senzorů. Pro splnění tohoto cíle je vytvořen zjednodušený 2D model rakety v prostředí SIMULINK, který je následně vybaven řídícimi systémy a aktuátory. Jejich finální verze je poté implementována do komplexnějšího, již předem poskytnutého, ale v průběhu práce upraveného, 3D modelu.

Keywords: Raketa, nosič, vertikální stabilizace, přistání, TVC, RCS, ACS, řízení polohy, PID, LQR

Acknowledgements

I would love to thank to doc. Ing. Martin Hromčík, Ph.D. for his great advisory and help with this diploma thesis. He has been an excellent supervisor. Beside that, I would like to express gratitude to my family for the support during my whole studies.

List of Tables

List of Figures

List of Acronyms

AC Aerodynamic Center. 11, 22, 52

ACS Attitude Control System. vii, 14, 36, 53, 60

CG Center of Gravity. 10–13, 15, 21–24, 52, 59

CP Center of Pressure. 10–13, 21

DOF Degree Of Freedom. xiii, 16

EDF Electric Ducted Fan. 63, 64, 68

ESA European Space Agency. vii

ESC Electronic Speed Control. 63

GNC Guidance, Navigation and Control. vii, 67, 68

HAT Hardware attached on top. 65

IMU Inertial Measurement Unit. 15–17, 65

ISS International Space Station. 6

LQR Linear Quadratic Regulator. 45–47, 60, 68

LV Launch Vehicle. 1, 7, 9–15, 17, 19–21, 23, 24, 29, 31, 32, 35, 37–40, 49–51, 55, 58, 59, 69

MIMO Multiple-input Multiple-output. 45, 60

PID Proportional-Integral-Derivative. 60, 68

RCS Reaction Control System. 14, 19, 24, 29–31, 37, 48, 52, 53, 56, 60, 64, 67

ROS Robotic Operating System. 65

RPM Rotations per Minute. 63

RTOS Real Time Operating System. 65

SISO Single-input Single-output. 35

SSME Space Shuttle Main Engine. 51

TVC Thrust Vector Control. 14, 19, 23, 29, 30, 36–38, 50–55, 60, 64, 67

List of Symbols

α Angle of attack. 11, 12, 21, 22, 56, 57, 67

θ Pitch. 29, 35–37, 39, 51

q Pitch rate. 29, 35–37, 39

$\delta_{\mathbf{F}}$ Fin deflection. 25

$\delta_{\mathbf{T}}$ TVC deflection. 23–25

Contents

Chapter 1

Introduction

Since the modern age of space travel and science, the rockets have remained mostly the same for several decades. The industry was controlled by few manufacturers and the research and development of new types of rockets stagnated. The industry got itself into the state, where it simply did not make much sense to develop any radically new type of rockets because there was no fierce competition to fight with. All space rockets operators have been national agencies and no privately founded Launch Vehicle (LV) has ever reached the orbit. That was the status quo until billionaire Elon Musk decided to establish the Space Exploration Technologies Corporation, abbreviated and mostly known as SpaceX, in 2002. In 2008, SpaceX's Falcon 1 rocket successfully reached the Earth's orbit and became the first privately developed liquid-fuel rocket to do so. This was the beginning of a new era for space travel. Another significant milestone was achieved on 22nd December 2015.

That day, SpaceX has become the first rocket launcher operator that successfully landed booster stage of its rocket. Until that day, this was considered to be impossible and most of the competitors were looking at SpaceX with laughter and expectations of failure. However, when SpaceX accomplished this outstanding task and was able to significantly lower the cost of a LV by reusing it, a spark was ignited for what later happened to be another great space race. Nowadays, the public can again hear about the plans to go to the Moon, Mars and beyond. Many well-established companies, as well as newly forming startups, want to get a ride, and therefore develop new types of spacecraft and LVs, which will be able to compete with SpaceX's fleet of rockets in the price and performance.

In order to compete with SpaceX's cost for a launch, prominent manufacturers of rockets need to accept the paradigm of reusability for their new and upcoming LVs. That is the only way to lower the price of a heavy-lift rocket significantly. And so far, the only experimentally tested way of doing so is by propulsive vertical landing. That is where this thesis steps in with the goals specified in the next chapter.

Chapter 2

Goals

The goal of the thesis is to design and validate flight control laws for stabilization and guidance of a vertically landing launcher. The topic is motivated by a recent ESA call dedicated to this problem to which the Department of Control Engineering was invited recently by a European research consortium. The tasks comprise building and validation of the simulation models suitable for control designs and verifications, selection of suitable control architectures and design methodologies, and parametrization of the controllers.

The tasks are further specified in greater detail as follows:

1. Construct simulation mathematical models in SIMULINK environment. Consider important aerodynamic effects. Perform linearization, validate the model, elaborate the sensitivity of the model to uncertainties in the physical parameters of the launcher.
2. Give specific and quantitative recommendations regarding actuators and sensors placement and parameters. Base your judgments on the simulation model and on realistic assumptions on the parameters of actuating and sensing systems.
3. Propose design methodologies and architectures suitable for stabilization and control of the launcher during the landing phase.
4. Design/parametrize the controller for selected values of physical parameters.
5. Validate performance and robustness of the control laws by means of robust control techniques, and extensive numerical simulations.

Chapter 3

Introduction to rockets

3.1 History of rockets

Rockets have been with us much longer than most people would expect. To briefly introduce the history of these machines, I will consider a rocket as any machine which utilizes the Third Newton's Law of action and reaction. This law makes the rocket move in the first place as the forward movement of the rocket is a consequence of propellant flowing out of the nozzle at the back of the launcher. [1]

In this manner, the first use of rockets in annals of human history is mentioned 400 BC, when Greek citizen Archytas developed a wooden pigeon propelled by steam. This invention was followed by another Greek inventor Hero of Alexandria, who invented the Hero's engine in the 1st century BC. But the most widely known fact of early use of rockets hold ancient Chinese. They are also the first to use different propellant other than steam. Between 0 to 100 AD, they started to experiment with compounds made from potassium nitrate (KNO_3), realgar (arsenic sulfide, As_4S_4), sulfur (S) and charcoal (C). All of those, except Arsenic Sulfide, were basic ingredients to gunpowder and even today, most of the motors for model rocketry use these same ingredients. [1]

From there, the development has continued with these compounds until the 17^{th} century. In 1696, Robert Anderson suggested that propellants could be mixed into moulds, forming the first step to increase the rate of rockets use, as this was an important step to start building solid-fuel rockets. In 17^{th} century, Isaac Newton was also born and introduced his laws of motion.

The era of modern rocketry has begun at the end of 19^{th} century with two founders of modern rocketry science being born. One of them was Konstantin Tsiolovsky, a Russian teacher, who started to focus on rocket engineering after his retirement. He is famous for introducing the idea of multistage rockets, which can be either serial or parallel. The second founder was Robert Goddard, who is the main person behind the idea of using liquid engines instead of solid ones. He even experimented with this technology himself and launched his first rocket that achieved a height of 12.5 meters. [1]

Since then, the development started to gradually accelerate with the peak between the 1940s

Figure 3.1: Illustration of the Hero's steam engine [2]

to 1970s.

3.2 Modern era of rockets

Humankind has achieved the largest and the most powerful rockets during the Cold war, which among others sparked the Space race between the USA and the Soviet Union. The pinnacle of this race was achieved by setting the human feet on the Moon's surface with the Apollo 11 mission. The mission was possible thanks to the Saturn V rocket, the most powerful rocket to date.

Since that was considered as the win of the Space race, the push for innovation after such a staggering achievement decreased significantly. The effort was shifted to building new International Space Station and as a part of the program, the Space Shuttle was developed to get materials and crew there. That was truly the first more significant push for the reusability of spacecraft. However, all the additional boosters and fuel tank were still disposed of after each launch, and after the Space Shuttle was retired, all rockets that have been used since then were once again made to be used only once.

3.3 Shift to reusable rockets

From the beginning of their operations, the mentioned SpaceX has been able to use the recovered booster up to 4 times with the theoretical possibility to be used up to 10 times. Besides that, the Dragon spacecraft, which is used to deliver supplies (and in the near future human crew) to the International Space Station (ISS) and the fairings have also been reused at least once. Hence the only part of Falcon 9 rocket, which has not been recovered, remains the second stage. It is worth to mention that in the case of Falcon Heavy, the reusability options are identical. There is a new launch system called Starship under development. That will be the most powerful rocket

ever developed with enough power to deliver payload not only to Earth's orbit but also to the Moon, Mars and beyond. The advantage of this rocket beside its power will be a fully reusable second stage.

Even though SpaceX is rightfully the most known rocket company utilizing reusable parts, it is not the only one. The closest competitor is Jeff Bezos' Blue Origin, which was founded around the same time. They have already tested their LV called the New Shepard with vertical launch and landing and it is expected to be used soon for crewed flights with space tourists.

Another company getting closer to recover their boosters is a young startup from New Zealand called the RocketLab. It is worth to mention that SpaceX lands their rockets either on the landing zone close to the launch pad or on the drone ship downrange on the ballistic trajectory. However, RocketLab just recently revealed their own way to recover the used booster. They tend to slow down the first stage with parachutes, catch the rocket with a helicopter and bring it slowly to land. This is possible mainly due to a very lightweight structure of RocketLab's booster, so the helicopter is able to carry the weight of the stage.

Besides the above-mentioned companies, there are of course many others, which are attempting to come up with their own version of the reusable first stage of the rocket. American United Launch Alliance and European Ariane Group among them.

In general, it can be said with certainty that reusable LVs are the future and they are here to stay since they offer a significant decrease in financial and time demands. That motivates the research of this work as more practical knowledge and experience needs to be gained in this relatively new field of rocketry in order to bring this technology into the mainstream.

Chapter 4

Actuators and sensors

Since the reusability of the rocket's booster is still such a recent and new technique in the space industry, there is no existence of any widely used method. Of course, there is the SpaceX, which has already flight-proven technology. Till this date, SpaceX has successfully landed at least 45 times. Hence it is expected that most of the other manufacturers of launch vehicles will follow the suit with a similar approach.

4.1 Propulsion

Propulsion is the central part of every rocket or even deep-space spacecraft. It propels the craft forward and is essential to get the LV to an orbit and beyond. There are several types of propulsion depending on the fuel and each is suitable for a specific mission.

For propelling a spacecraft deeper into space once it is at stable orbit, the best option is the one that can burn longer. Representatives of this group are, for instance, electric or nuclear rocket engines. They provide less power for a more extended period, which is a better outcome for this type of mission.

For any rocket to leave Earth's gravity, however, much greater force is required. Such a power can be currently provided only by solid, liquid or hybrid rocket engines. Hybrid engines are not commonly used in heavy-lift LVs and therefore will not be considered in this thesis.

4.1.1 Solid Rocket Engines

In the early days of rocketry, only the solid fuel was used. It provides the highest thrust and is still being used these days, however, as an additional and supportive booster. The propellant is known as a *grain* and it is usually a mixture of an oxidizer and the fuel itself. A common solid oxidizer is, for instance, ammonium perchlorate and one of the commonly used fuel is an atomized aluminium powder. [1]

Even though solid rocket motors provide the highest thrust, they bring several major drawbacks. First and foremost, the most significant one is the impossibility of control the thrust and repeated ignition of the motor. Once the motor is ignited, it cannot be turned off and reignited

again. This is a significant disadvantage for achieving a precise trajectory. Therefore, these motors are mostly used only as an additional booster whose primary purpose is to help at the initial phase of liftoff when the greatest thrust is needed.

4.1.2 Liquid Propellant Rocket Engines

The most used technology in modern LVs to provide motion is liquid-propellant engines. As the name suggests, they use the oxidizer and the fuel in liquid form. The oxidizer is usually liquid oxygen, and the fuel is liquid hydrogen or kerosene. These days, liquid methane is becoming more popular as the new SpaceX's Raptor engine and the new Blue Origin's BP-1 engine are powered by liquid methane. The main reason to opt for liquid methane is its possibility to be eventually produced *in situ* on the Moon or Mars.

The main reason for using liquid engines is that they solve the drawbacks of the solid motors. Specifically the possibility to reignite the engines many times and the invaluable option to throttle the engines accordingly. These advantages beat the fact that these engines need significantly more parts, which increases the complexity and the probability of potential failure.

4.2 Attitude control

Next, I will focus on ways how modern rockets are steered during their flight. I will try to describe ways of control both during ascending and descending part of the flight since both of them introduce their own challenges.

4.2.1 Atmospheric stability

For any rocket flying through the atmosphere, it is certain to encounter the aerodynamic forces and moments developed by the atmosphere and the structure of the vehicle. The following sections describe important parameters regarding the atmospheric flight.

Center of Pressure

Every object that moves in a fluid experiences a different velocity and pressure of the fluid at different places along its body. This difference results in a force moving the body from a high-pressure area to low pressure area.

To significantly simplify the calculations, it is common to calculate the average location of the pressure. This position is called the *Center of Pressure (CP)* and it is analogical to the *Center of Gravity (CG)*, which is the average location of the body mass. It can be calculated as:

$$cp = \frac{\int xp(x)dx}{\int p(x)dx} \tag{4.1}$$

where *p(x)* is a function of pressure dependent on the distance from reference line.

The position is highly dependant on the α (Angle-of-attack), which again complicates the calculations. However, there is one specific point of the body, where the atmospheric moment stays almost constant independently on the α. Such a point is called the Aerodynamic Center (AC). It was previously found both empirically and analytically that for most low-speed objects, its position is at 0.25 of the length back from the leading edge of the object. For supersonic objects, the AC moves to 0.5 of the length. [3]

Hence, I will assume that all aerodynamic forces are applied at the AC. For reference, I will also compute the CP as an area average of the centers of the components. This is possible thanks to the assumption that the pressure is nearly constant. Otherwise, there would be a need for Computation Fluid Dynamics simulations to estimate the functions describing the evolution of the pressure around the rocket. [4]

I can, therefore, say that

$$A \cdot cp = (a_b \cdot d_b) + (a_f \cdot d_f) \tag{4.2}$$

Subscript b stands for the body of the LV and f stands for fins (those will be explained later). The variable d is distance of parts' own CPs from the reference line, which in my case will be the bottom of the LV. Variable A stands for projected area and $A = a_b + a_f$. Equation can be rewritten as:

$$A \cdot cp = \left(\pi r^2 \cdot \frac{l_b}{2}\right) + 4(w_f^2 \cdot 0.975 l_b) \tag{4.3}$$

where l_b is length of the first stage and w_f is width of the fin. For purpose of the calculation, let me define $r = 1.85, l_b = 40, w_f = 1$.

$$cp = \frac{(10.75 \cdot 20) + (4 \cdot 39)}{10.75 + 4} \approx 25 \tag{4.4}$$

The static CP is therefore approximately 25 meters from the bottom of the LV. The height of the vehicle is 40 meters. The distance between CP and CG is called the Static Margin. The minimal Static Margin that needs to be maintained to ensure the static stability is equal to the maximal diameter of the LV's body. That is accomplished.

Stability

For the rocket to be aerodynamically stable during its ascend, its CP needs to be below its CG, because any deflection will increase the forces acting on the vehicle. If the CP is behind the CG, the forces have restoring effect. In other case, the forces deflect the vehicle even more. Most of the model rockets follow this rule as well as early powerful rockets. During the second half of the 20^{th} century, however, the responses of computers and autopilot systems became fast enough to steer the rockets no matter how aerodynamically unstable the LV was. [5]

This trend was already apparent with the mighty Saturn V. On the figure below, it is visible that the CG is below the CP most of the time.

There are very small fins at the bottom of the rocket. Their purpose is to only decrease the

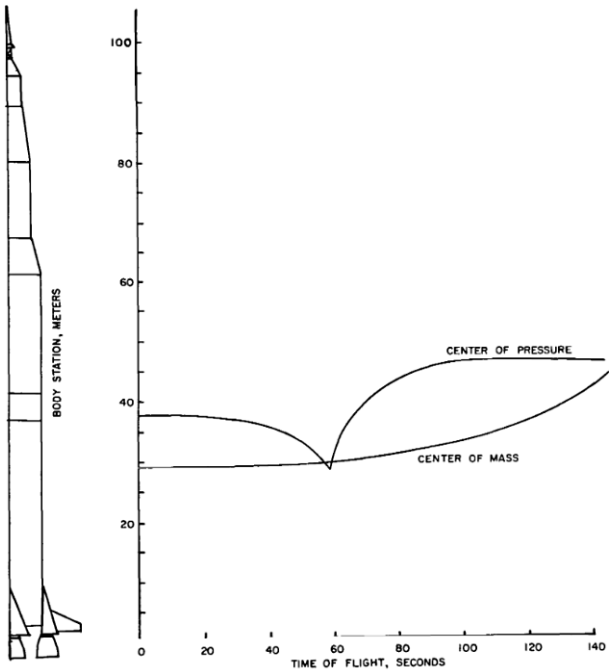

Figure 4.1: Location of Saturn V's CG and CP [6]

instability so in the case of engine gimbal failure, the structural load will not exceed the safe limits and the crew will have enough time to save themselves. [5]

That implies that most of the modern orbital LVs including the Falcon 9 are inherently unstable.

For descend, the situation is the opposite. Now, the CP should be above the CG to be aerodynamically stable. That is far more desirable than in the case of ascending, simply because there are no control surfaces (like fins) neither powerful engine in the upper part of the first stage to control the potential inherent instability by default. The instability could be fatal in such a case.

Therefore, there might be some necessary adjustments required, either in the structure itself to provide inherent stability during the descent or by using active control actuators.

4.2.2 Fins

Fins have been used in atmospheric flight for decades as a primary provider of steering and stability. Most of the time, fins are placed at the tail of the body. By doing so, the natural aerodynamic stability is ensured. With increasing α, however, the CP is moving forward towards the CG and the LV becomes increasingly unstable.

If the fins are controllable, then the rocket can be steered in all three axes (*roll, pitch, yaw*) given the assumption that the LV has multiple fins as is usual.

A different situation occurs in the case of descending flight. Since the heat generated by the friction of booster surface and atmosphere during descent, the part of the LV which can withstand such high temperatures needs to be at the front. It turns out that nozzles are designed precisely

for that. Hence, when the LV is planning to land, it is suitable to be re-entering the atmosphere by the base of the rocket first.

However, then the fins cannot be at the base anymore, as it would mean that CP is still below CG that would destabilize the since there is no actuator at the top to counteract the distorting forces. Therefore, it would be beneficial if the rocket has foldable fins at the top of the first stage which would be folded during the ascend and unfolded during the descend to move the CP further above the CG.

That is precisely what SpaceX is doing with their Falcon 9 and Falcon Heavy boosters and Chinese are trying to do with their Long March-2C rocket. They use so-called grid fins (official title is *Belotserkovskiy's Grid Fin* after the inventor), which are a lattice with small aerodynamic surfaces in a rectangle shape. [7]

Figure 4.2: Grid fins on the Falcon 9 rocket [8]

They are designed specifically for supersonic and hypersonic flights and because of the design, they don't experience as enormous atmospheric forces as they would do if the fins would be planar as usual. Together with a much smaller footprint, this results in significantly smaller hinge moments, therefore much smaller actuators are required. [7] Grid fins are steerable, so they can rotate along its axis in order to stabilize and control the rocket. And as Elon Musk himself stated during his "Ask me Anything" session on Reddit.com, "*The grid fins are super important for landing with precision.*" [9]

To wrap this section up, it seems reasonable to use grid fins during landing. An important question is up to which velocity are they effective. As this thesis focuses on the very last part of the descend, their use is questionable. However, because at the end of my work, I would like to offer validation of recommended actuators together with designed control laws based on real data, I will consider initial conditions to be equal to parameters of LV at the moment of ignition for the final landing burn. Based on observations of data from online SpaceX streams of their launches, the average velocity at that moment seems to be around 290 m/s. [10]

Such velocity is high enough for the LV to be significantly affected by atmospheric forces and therefore it is recommended to use grid fins for both making the vehicle passively stable and maneuverable. The effect of using grid fins is expected to decrease with decreasing speed

and therefore will only introduce marginal help at the end of the landing maneuver.

4.2.3 Thrust Vector Control

Modern orbital rockets have very rarely fins at their base these days. As mentioned in the section above, even at the times of the Saturn V, the fins were not required anymore for the steering, but only to reduce the inherent instability. At that time, the computers and control systems became responsive enough that different types of control techniques could be used and eventually, to fully take over the use of fins. This technique is called the thrust vectoring.

The Thrust Vector Control (TVC) manipulates the direction of thrust by several different ways. However, I will consider the one that uses gimbaled engines which allows to steer them (or only nozzle in case of different configurations) accordingly in a specific range of angles. The range varies with each engine but I will assume the maximum range to be between 5° to 15°. This technique is also used by multiple modern fighter jets to increase their maneuverability.

In case of vertically landing a rocket, TVC plays a major role. First, it is able to decelerate the vehicle significantly and second, it has the biggest control authority at the final phase and therefore I assume that this unit needs to be a part of successful vertical landing system.

4.2.4 Reaction Control System

In the last phase of the landing, the speed of LV is significantly slower than at the beginning of the landing burn. At these lower speeds, atmospheric forces are negligible, hence there is a need for another kind of force, which would help the TVC to maintain stability and such an actuator system is called the Reaction Control System (RCS).

Until SpaceX has started to use RCS in their boosters, most of these systems were mostly used on spacecraft as a part of the ACS, where they help to control attitude in a vacuum environment more precisely than the main engine.

The RCS usually represents a set of small thrusters, which are placed along the spacecraft body. Their thrust cannot be throttled and they can be only turned on or turned off. In case of a landing rocket, one dimension can be omitted as there is no need to move rocket up or down since these thrusters are too weak to get such a task done. They only serve as stabilizing actuator. That means it needs thrusters on each side of the upper part of the booster. Jets at the bottom can be omitted thanks to the presence of the main engine (they could be used in the vacuum to turn the rocket backward before its descent, but for the purpose of the final landing phase, they are omitted).

Because the SpaceX is the only company I know about which uses the RCS on their booster (RocketLab has intention to use them as well), I haven't been able to find any exact information on the provided thrust for the RCS in such a vehicle. I will assume that they are at least as powerful as SpaceX's Draco RCS engines, which produces 400 N of thrust [11], or more. Since the booster is a much larger vehicle than the Dragon spacecraft, I believe it can accommodate stronger RCS thrusters. Therefore, I will assume that their thrust is 1 000 N. Their position

should be as high as possible to increase their moment of force on the CG.

4.3 Landing

The last part of the descent is the landing. To not break the engines' nozzles, which are not meant to withstand such a weight, some form of keeping the engine nozzles untouched is necessary.

There are three possible ways. First one is to land on the runway using the same way as Space Shuttle used to do, however our rocket does not have any wings to glide.

The second way is to use deployable legs as the Falcon 9 does. The legs will deploy a few seconds before the final touchdown, therefore will not interfere with atmospheric forces.

Last currently viable option to consider is catching the LV mid-air similarly like RocketLab plans to do. [12] This solution is, however, feasible only for lightweight rockets which can be carried by helicopter.

Based on considerations and the task of the thesis that considers the vertical landing, I will recommend to use the deployable legs. However, I will not include legs into the mathematical model as they don't change the flight dynamics drastically and they are not necessary for completing the task of the thesis.

4.4 Sensing

Besides the flight computer and actuators, there is one more important area of instruments required for a successful mission. And that is sensing.

Sensing is usually focused on the vehicle, its internal parts and its outer environment. I will focus on sensors related to the thesis, therefore mainly about the attitude of the vehicle. And when there is ever mention about measuring the attitude of any vehicle, the Inertial Measurement Unit (IMU) comes to mind.

4.4.1 Inertial Measurement Unit

The IMU is an essential measurement unit in many aerospace vehicles. It is called a unit because it consists of multiple gyroscopes and accelerometers, which allows it to measure both linear acceleration and rotational rate. Those parameters are required for the inertial navigation systems as well as for control systems.

Gyrodynamics

Gyroscopes consist of a wheel spinning at a constant rate and it is mounted in a way that allows it to deflect the axis of rotation in a particular direction called *gimbal*. Within the IMU, we usually find so-called *fully gimbaled gyro*, which means it has three gimbals, the axis of each being perpendicular to the others. This allows to measure rotational dynamics in 3 degrees of freedom.

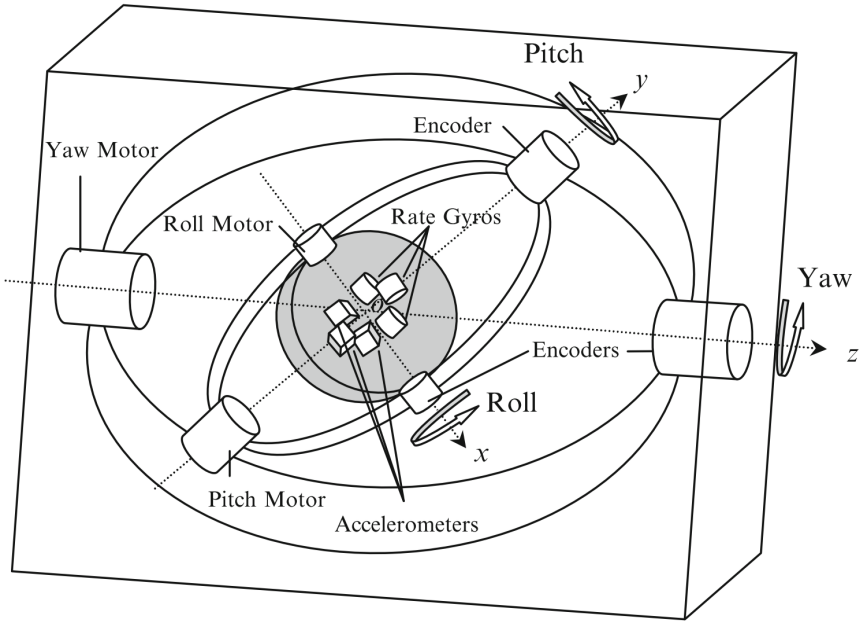

Figure 4.3: 6 DOF IMU with gyros and accelerometers [13]

Because the rotation of the wheel will due to the conservation of angular momentum stay constant, it provides a stable inertial reference which is unaffected by the manoeuvres of the vehicle. As the outer frames rotate, they do so with respect to the rotating wheel. The level of deflection of relative gimbals is measured by sensors such as *encoders*. If there are no sensors specifically measuring the inclination of the vehicle relative to the platform, we can obtain the angular displacement by integrating the angular rate. In such a case, the gyroscope is called *rate-integrating*. If we combine both rate and rate-integrating gyro, we get the full information about the attitude state of the vehicle. [13]

4.4.2 Accelerometers

The second essential part of the IMUs are accelerometers. They can measure the linear acceleration and there are usually at least 3 of them in the IMU to provide measurement in full 3D motion. There are two commonly used types of these sensors.

First types (such as MEMS accelerometers) incorporates internal capacitors that measure changing capacity as the internal parts change their relative inner position.

The second type of accelerometers uses piezoelectric materials which output small electric charge under tension created by the acceleration.

4.4.3 Sensor parameters

There are two properties of every sensor important for the real use that need to be considered when modelling them.

Time delay The time delay uses the fact that in reality it is not possible to get actual state value in real time. There will always be some amount of delay because of the processing time. This time can consist of sensing itself and signal pre-processing such as filtering or converting from analog to digital. The time delay can generally vary, but in this case I will consider it as a constant and I will simulate effect of different values on the control performance later on.

Bandwidth Second parameter to consider is a bandwidth. That tells us at what frequency can the sensor pick up the measured value. Usually, the higher the better. On the other hand, that can cause the control system to be too sensitive which might be a disadvantage in a case of a high frequency noise being present. In such a case, a Low-Pass filter would be put in place to filter out all frequencies higher than the given threshold.

4.4.4 Sensor placement

The last thing to discuss is the sensor placement. In general, it depends on several factors such as location of vehicle bending modes, wiring requirements, interface issues and so on. [14]

For the LV that consists of reusable booster, the location of the control sensors is more or less predetermined by the type of mission. If there would not be the reusable booster, the sensors could be placed in second stage, where it would allow to control both stages with one set of sensors.

For the case of reusable rocket, however, the sensors are required in the first stage as precise measurement is a must to land successfully. Since the most of the part of the booster is made by fuel and oxidizer tanks together with very complex systems for the pumps and engines, it is not suitable environment to place the measurement unit.

Therefore, based on these information, I do recommend to place the IMU on the top of the first stage. First, there is more space to house the sensors and second, the vibrations are much less pronounce than in the lower part closer to the engines. Another benefit of such a placement is smaller offset with respect to center of mass, which is expected to be in the upper part of the first stage due to heavy engines and full tanks.

Chapter 5

2D model

In this chapter I introduce a rigid body model, that serves as a basic benchmark for testing and verifying my assumptions, improving them and it will also help with the design of a proper controller to stabilize and control the rocket.

Because the goal of this thesis is to vertically stabilize the vehicle, I don't need to include controls and stabilization of *roll* which is considered to be stable in this case. Besides, the presence of control fins itself offers benefit of having inherent stabilizing effect in *roll* motion. Hence, I will only need to control *pitch* and *yaw*. The vehicle is axisymmetric about both axes, so the models are the same. This allows me to focus on *pitch* only and once the control laws are set, I can apply them to control the *yaw* as well.

That way, I can begin with modelling a simpler mathematical model positioned only in the 2D world. I can omit *roll* and *yaw* dynamics completely as well as dynamics in the 3$^{\text{rd}}$ dimension, when the assumption that attitude stays close to the considered vertical position is in place. In case of excessive deflections in one of the axis, that could change.

This model includes controllable TVC, RCS and fins that will later help with stabilization and attitude control.

5.1 Subsystems of nonlinear model

For every LV, three main subsystems affect its overall dynamics. Those are atmosphere, gravity and the thrust. Before describing each of those, let me introduce other, no less important parts which also need to be considered.

Coordinate frames

Simulation on the 2D model will consist of two different coordinate systems. The first is a body coordinate frame and it is the main one in terms of attitude control of the LV. The body is however moving in another inertial reference frame that represents the world and forces in it. The transformation between those is needed and its definition has the form of a standard

rotational matrix in 2D:

$$T_{\mathrm{B}}^{\mathrm{IN}} = \begin{bmatrix} \cos(\gamma+\theta) & -\sin(\gamma+\theta) \\ \sin(\gamma+\theta) & \cos(\gamma+\theta) \end{bmatrix}$$

where γ is default rotation between the reference frames set to $\frac{\pi}{2}$.

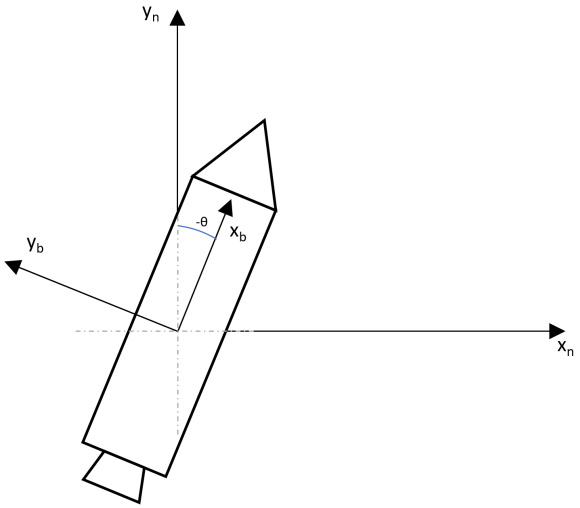

Figure 5.1: Overview of reference frames

Rotational Dynamics

To describe the rotational behaviour of the LV, let start with the angular momentum L, which is defined as

$$L = J \cdot \omega \tag{5.1}$$

Its change is determined as the net torque (moment of force) based on the 2$^{\mathrm{nd}}$ Newton-Euler Law

$$M = \dot{L} \tag{5.2}$$

I can express this inertial derivative by using Transport theorem, which yields

$$M = \dot{L} + (\omega \times L) = (\dot{J}\omega + J\dot{\omega}) + (\omega \times J\omega) \tag{5.3}$$

I can separate $\dot{\omega}$ to get the equation for angular acceleration

$$\dot{\omega} = [M - \dot{J}\omega - (\omega \times J\omega)]J^{-1} \tag{5.4}$$

Since the moment of inertia for my 2D rocket is considered constant, its derivative will be zero, hence

$$\dot{\omega} = [M - (\omega \times J\omega)]J^{-1} \tag{5.5}$$

Because of the 2D case, $\omega = (0, 0, \omega)$ and $J_{zz}\omega = (0, 0, J_{zz}\omega)$. Their respective cross product is

equal to zero. That results in final formula

$$\dot{\omega} = MJ^{-1} \tag{5.6}$$

Moment of inertia for cylinder rotating around its center is calculated as

$$J = \frac{1}{12}ml^2 + \frac{1}{4}md^2 \tag{5.7}$$

where l is height of the LV and d is its diameter. [15]

5.1.1 Atmosphere

As the LV reaches the velocity up to several thousand meters per second, the atmospheric forces play a significant role in the behaviour of the rocket. This behaviour is among others significantly defined by the relative position of the CP and the CG as it has been explained in the section 4.2.1.

In terms of atmospheric forces, the most prominent ones are certainly *drag* and *lift*. Drag is the force acting in the opposite direction of the airflow velocity. Lift force is perpendicular to the relative airflow. These two forces are especially used with dynamics of airplanes.

For the rocket, it is welcomed to measure all the forces in the body coordinate frame. Hence I will use different set of forces, called *axial* and *normal*. These forces are defined such that *axial* force F_A acts in the opposite direction to the longitudinal motion of the rocket and *normal* force F_N is perpendicular to it. (It is possible to use transformation matrix from 5.1 (without the considered γ) to transform F_D, F_L to F_A, F_N and vice versa.)
The axial force is defined as

$$F_{\mathrm{A}} = C_A \frac{1}{2}\rho S v_{air}^2 \tag{5.8}$$

The normal force similarly as

$$F_{\mathrm{N}} = C_N \frac{1}{2}\rho S v_{air}^2 \tag{5.9}$$

where C_A is the axial atmospheric coefficient, C_N is the normal atmospheric coefficient, ρ is the air mass pressure at current altitude, S is the frontal area and v_{air} is the velocity of the surrounding air relative to the body.

Quantity $\rho\frac{v^2}{2}$ is in aerospace industry well known as *dynamic pressure* and is usually denoted with q. The term can be for instance heard during commentary of space rocket launch when technicians mention that the vehicle is passing through the *"Max-Q"* at the moment when the structure of the vehicle is exposed to the greatest atmospheric forces during its flight. [1]
Both C_A and C_N are dependent on the α (Angle of Attack) as follows:

$$C_{\mathrm{A}} = C_{A_0} + \alpha C_{A_\alpha} \tag{5.10}$$

$$C_{\mathrm{N}} = C_{N_0} + \alpha C_{N_\alpha} \tag{5.11}$$

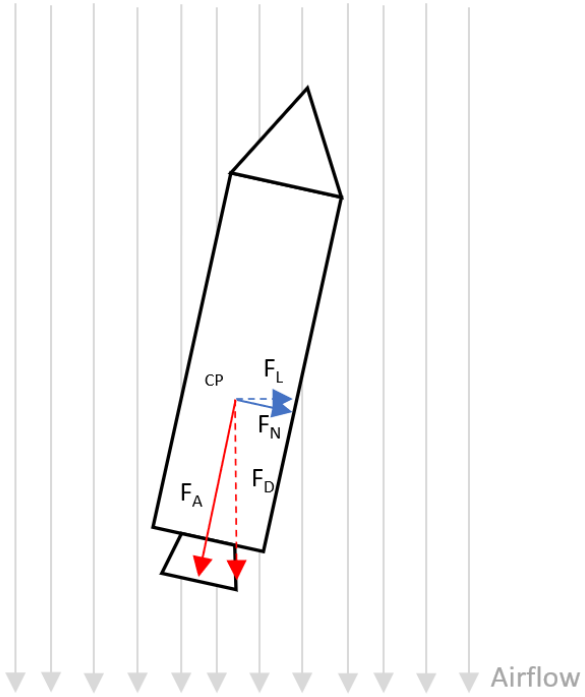

Figure 5.2: Diagram of forces acting on the rocket

Elements with index 0 are coefficients for zero α. C_{N0} is always zero for zero α. The coefficients are also highly dependent on many other variables such as the Mach number, dynamic pressure, Reynolds number and others. Therefore they are usually observed by experiments or through Computational Fluid Dynamics simulations. I have used coefficient values inspired by values considered for the velocity of 200 m/s from a script related to [16] and throughout the thesis, they are treated as constant. For the $\vec{r}_{\mathrm{AC}}$ being location of the aerodynamic center, the moment created by aerodynamic forces can be defined as

$$\vec{M}_{\mathrm{atm}} = r_{\mathrm{AC}} \times \vec{F}_{\mathrm{atm}} \tag{5.12}$$

However, I will follow the aerodynamic convention and will define the pitching moment in a similar fashion like F_{A} and F_{N}

$$M = C_{\mathrm{M}} \frac{1}{2} \rho S v_{air}^2 l_{\mathrm{AC}} \tag{5.13}$$

where l_{AC} is the distance between AC and CG. That changes based on airspeed as explained in 4.2.1. Similarly as coefficients of atmospheric forces, also moment coefficient is defined as

$$C_{\mathrm{M}} = C_{M_0} + \alpha C_{M_\alpha} \tag{5.14}$$

Finally, α is defined as an angle between oncoming airflow and the longitudinal axis of the vehicle. Because the velocity of the airflow can be neglected with respect to high speeds of the

rocket, the calculation can be done as:

$$\alpha = \tan^{-1} \frac{v_y}{v_x} \tag{5.15}$$

where parameters are respective velocities in the body frame.

Actual values are derived from information provided in [16], [17] and [18] and are following:

C_{A_0}	0.8
C_{A_α}	0.01
C_{N_0}	0
C_{N_α}	0.05
C_{M_0}	0
C_{M_α}	0.02

Table 5.1: Aerodynamic coefficients for 200 m/s

5.1.2 Gravity

The gravitational force is applied in the CG. To define the magnitude of the force in the body frame, θ needs to be taken into account.

$$\vec{F_g} = \begin{bmatrix} F_{g_x} \\ F_{g_y} \end{bmatrix} = \begin{bmatrix} -mg\cos(\theta) \\ mg\sin(\theta) \end{bmatrix} \tag{5.16}$$

Since the CG is a point of application of the force, and it is at the origin of the body coordinate frame, there is no torque created.

5.1.3 Thrust

The thrust is another essential part of the dynamics of the LV as it is the thrust which makes the rocket to move from the launch pad in the first place. By default, the nozzle is aligned with the CG and the longitudinal axis of the rocket. Hence in an ideal case, the resulting thrust is aligned as well and total moment exerted on the rocket is zero.

In such a case, the rocket would move only forward. Hence two main ways to control and steer the rocket became popular over time. The first one is to use control fins in similar way control surfaces are used with aircraft. With four of them, we can control the *roll*, *pitch* and *yaw*. Control fins can also be used only for their stabilizing function.

Nowadays, however, most of the LVs are steered by the TVC. The TVC unit controls the angle of the nozzle respectively to the body in two axes. Therefore it creates a moment, which subsequently steers the rocket in one or another direction. The mathematical expression of thrust in 2D is then defined as follows:

$$F_{\mathrm{thr}x} = F_{thr} \cdot \cos\delta_{\mathrm{T}} \tag{5.17}$$

$$F_{\text{thr}y} = F_{thr} \cdot \sin \delta_{\text{T}} \tag{5.18}$$

where δ_{T} is the angle of the nozzle.

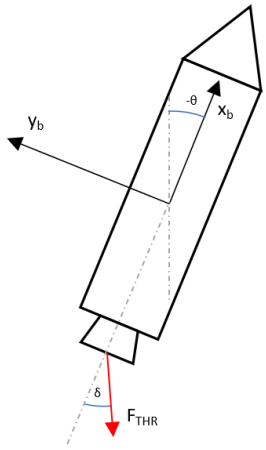

Figure 5.3: Thrust Vectoring Control

Moment of force is defined as

$$\vec{M}_{\text{thr}} = R_{nozzle} \times \vec{F}_{thr} \tag{5.19}$$

5.1.4 RCS

The function and explanation of RCS were presented in further detail in section 4.2.4. The aim of this paragraph is to introduce the mathematical definition of those already explained mechanisms.

The RCS thrusters point perpendicularly to the LV's body and $\vec{F}_{\text{RCS}}$ is independent to an angular position of the rocket. It can be only either fully turned on or completely turned off.

Moment of force for the RCS is equal to

$$\vec{M}_{\text{thr}} = R_{\text{RCS}} \times \vec{F}_{\text{RCS}} \tag{5.20}$$

where R_{RCS} is position of RCS thrusters relative to the CG.

5.1.5 Fins

As said earlier, the fins are very important during the atmospheric part of the descend. In this thesis, I will assume use of so-called *Grid fins*, which are especially designed for the supersonic regime. It is common to use 4 fins, so the control in all axes is possible. Since my model is only 2D, I will consider only two of these in a pair at z axis opposite to each other. For the calculation of the final torque, the common aerospace equations to calculate both force and torque created

by the control surfaces are used.

$$M_{fin} = C_{M_F} q S_{fin} l_{AC} \tag{5.21}$$

where q is the dynamic pressure and C_{M_F} is a pitch moment for the fin.

$$C_{M_F} = C_{M_0} + \delta_{\mathrm{F}} C_{M_{\delta_{\mathrm{F}}}} \tag{5.22}$$

where the coefficient for zero δ_{F} is zero. Assumed value for the moment coefficient at 10 degrees is 3.75. [19] At the end, the values are multiplied by 2 since the 2 fins are used.

The deflection of fins has however one more effect on the vehicle. It produces a drag calculated as

$$F_{F_A} = C_{A_F} q S_{fin} \tag{5.23}$$

5.2 Implementation of the model in SIMULINK environment

Matlab and SIMULINK environment are chosen tools to simulate the system. The implementation itself can be seen on the images below.

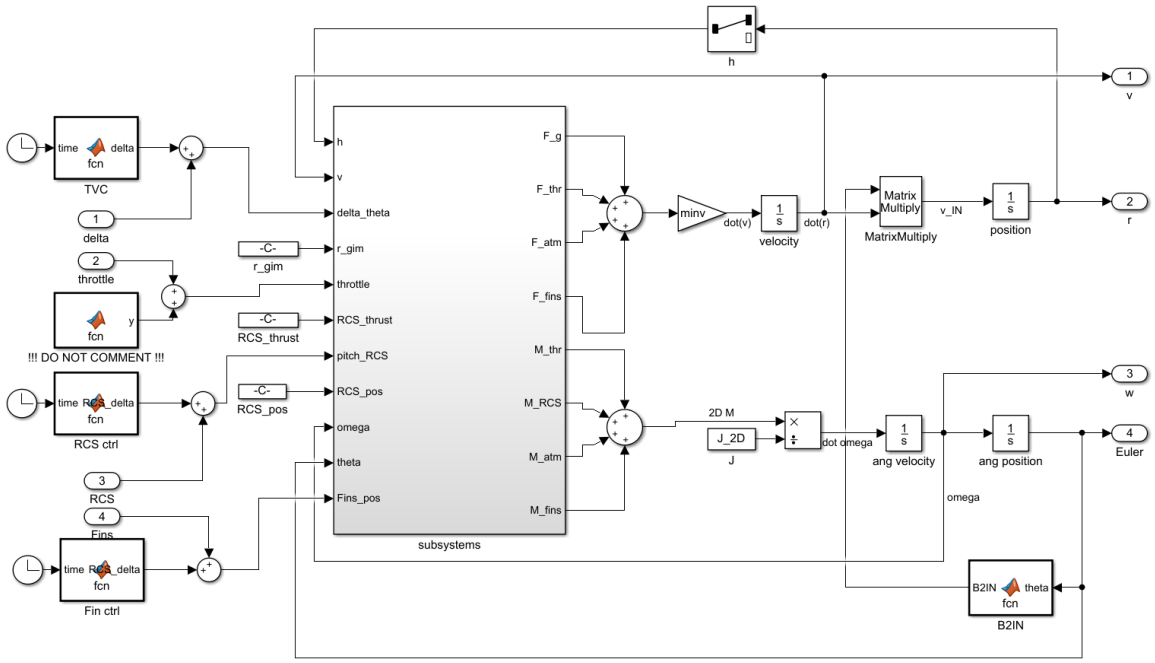

Figure 5.4: SIMULINK implementation of 2D model

Input variables are $u = \begin{bmatrix} \delta_{\mathrm{T}} & F_{\mathrm{thr}} & F_{\mathrm{RCS}} & \delta_{\mathrm{F}} \end{bmatrix}^T$

To finalize the definition of this model, remaining relevant parameters are set accordingly based on a real reference mission.

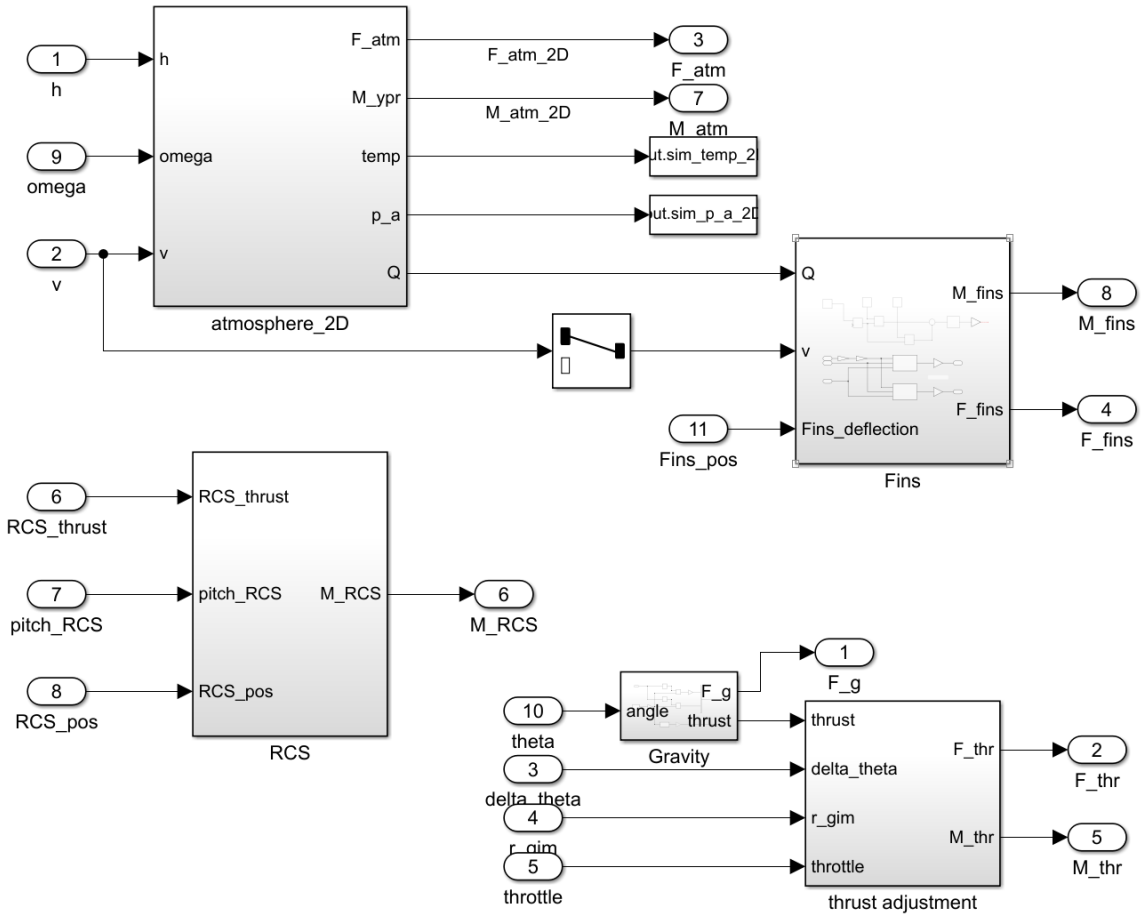

Figure 5.5: SIMULINK implementation of subsystems in 2D model

m	30,000	[kg]
d	3.7	[m]
l	40	[m]
J	4034225	$[\mathrm{kg} \cdot \mathrm{m}^2]$

Table 5.2: Physical parameters of the LV

By default, the thrust of the rocket engine in this thesis is set to be equal to $\vec{F}_g$, so rocket hovers at the same altitude. In reality, however, even a single engine of the Falcon 9 rocket, which I use as a reference, is not able to throttle down to such an extent.

5.3 Final equations of 2D motion

Now, when all the important physical relationships have been described, I will put them all together to one system of equations of motion. Note that they are written in the common state-space description.

$$\dot{v}_x = \frac{1}{m}(F_{thrx} + F_A + F_{g_x} + 2F_{F_A}) \tag{5.24}$$

$$\dot{v}_y = \frac{1}{m}F_{g_y} \tag{5.25}$$

$$\dot{r}_x = T_B^{IN} v_x \tag{5.26}$$

$$\dot{r}_y = T_B^{IN} v_y \tag{5.27}$$

$$\dot{q} = \frac{M_{atm} + M_{thr} + M_{RCS} + 2M_{Fins}}{J} \tag{5.28}$$

$$\dot{\theta} = q \tag{5.29}$$

5.4 Model analysis

5.4.1 Linearization

The first step for quick analysis is to linearize the nonlinear model. Linearization is a linear approximation around a certain operating point. This point can be chosen arbitrarily but in most cases, the *equilibrium* point is chosen. That is the point, in which the dynamic system will stay unless it is externally forced out of it. To linearize the system, I need to differentiate all the equations of motion with respect to individual states.

$$x = \begin{bmatrix} v_x & v_y & r_x & r_y & q & \theta \end{bmatrix}^T \tag{5.30}$$

The result will be linear equations describing the dynamics of the system in the close neighbourhood of the operating point. I will provide both symbolical and numerical result of the linearization.

The state matrix A, which describes the internal dynamics of the system, is obtained as

Jacobian matrix of the equations of motion. Result of symbolic linearization is shown below

$$\begin{bmatrix} \frac{C_A\, S\, \rho\, v_x + 4\, C_{D_F}\, S\, S_f\, \rho\, v_x}{m} & 0 & 0\ 0\ 0 & g\, \sin(\theta) \\ 0 & 0 & 0\ 0\ 0 & g\, \cos(\theta) \\ \cos(\psi+\theta) - \sin(\psi+\theta) & 0 & 0\ 0\ 0 & -v_x\, (\cos(\psi+\theta) + \sin(\psi+\theta)) \\ 0 & \cos(\psi+\theta) + \sin(\psi+\theta) & 0\ 0\ 0 & v_y\, (\cos(\psi+\theta) - \sin(\psi+\theta)) \\ \frac{2\, S_f\, l\, \rho\, v_x \left(\frac{C_{M_F}}{2} + \frac{\mathrm{CM}_{\delta_f}\, \delta_f}{2}\right) + C_M\, S\, l\, \rho\, v_x}{J} & 0 & 0\ 0\ 0 & 0 \\ 0 & 0 & 0\ 0\ 1 & 0 \end{bmatrix} \tag{5.31}$$

Input matrix B tells us how the inputs affect the individual states.

$$\begin{bmatrix} -\frac{\mathrm{F_{thr}}\, \sin(\delta)}{m} & \frac{\cos(\delta)}{m} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{\mathrm{nozzle_{dist}}\, \mathrm{F_{thr}}\, \cos(\delta)}{J} & \frac{\mathrm{nozzle_{dist}}\, \sin(\delta)}{J} & \frac{\mathrm{RCS_{dist}}}{J} & \frac{\mathrm{CM}_{\delta_f}\, S_f\, l\, \rho\, {v_x}^2}{2\, J} \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{5.32}$$

For numerical linearization, I have used ***linmod*** command to linearize SIMULINK model at operating point of state vector $x = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$. That translates to vertically positioned hovering rocket. Result is shown below.

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 9.814 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{5.33}$$

As mentioned, the representation of the system will follow the state-space form

$$\dot{\vec{x}} = A\vec{x} + B\vec{u} \tag{5.34}$$

$$\vec{y} = C\vec{x} + D\vec{u} \tag{5.35}$$

The state equation for earlier stated LV's parameters looks as follows:

$$\dot{\vec{x}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 9.814 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \vec{x} + \begin{bmatrix} 0 & 9.8136 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1.0764 & 0 & 0.0064 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \vec{u} \tag{5.36}$$

Output matrix C is the identity matrix and feed-forward matrix D is zero. For the current configuration of the selected operating point, the influence of the fins is set to zero as they don't produce any force in the hovering state.

5.4.2 Stability, controllability and observability

Simple criterion for accomplishing linear system's stability it that none of the poles have a positive real part, or there cannot be repeated poles with zero real part. Since state matrix A represents the state dynamics itself, I can compute the poles as eigenvalues of matrix A and get

$$eig(A) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

which means that the system is unstable as there are multiple eigenvalues placed at the origin.

Rank test reveals that the system is fully controllable and observable. Therefore, it is possible to move the initial state to any other state within a finite time via external input, which is essential for designing the control laws later in the work. The observable system means that all internal states are observable out of the system via output vector y.

5.4.3 Pitch dynamics

This subsection will provide analysis of internal dynamics related to the angular position and angular velocity.

The transfer functions between the TVC and θ are

$$H(s)^{\theta}_{\text{TVC}} = \frac{-1.076}{s} \tag{5.37}$$

and TVC and q

$$H(s)^{q}_{\text{TVC}} = \frac{-1.076}{s^2} \tag{5.38}$$

Transfer function for the RCS control looks the same, just with different gain. Values can be seen in 5.36.

5.4.4 Validation of model's behaviour

Before I can proceed further, I need to make sure that the model is designed well and it behaves as expected, especially in terms of applied control. Results of these scenarios are used as verification of the linearization process. As long as the inner state won't be too different from the operating point, the linearized model should copy the response of the original nonlinear model with high enough accuracy.

Control of the TVC

The model responses to the TVC control input as expected and more importantly, the response of both systems is identical:

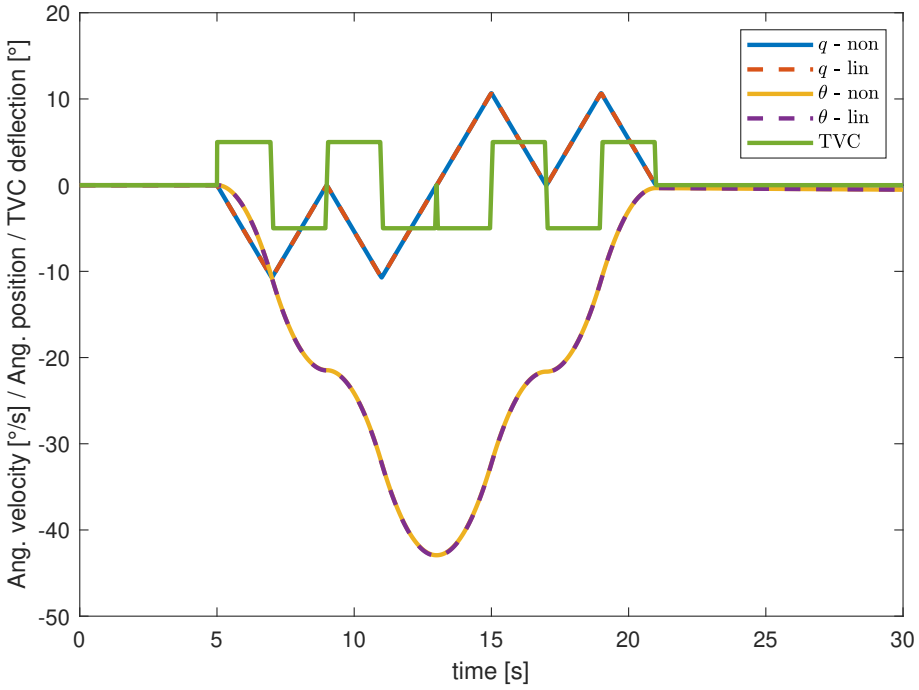

Figure 5.6: Comparison of linear and nonlinear system for TVC control inputs

Control of RCS

Linear response to the RCS control input again follows the original nonlinear version as shown below. Since the response for the fins control behaves the same in terms of no difference between the linear model and nonlinear one, the figure is skipped to keep the section less cluttered.

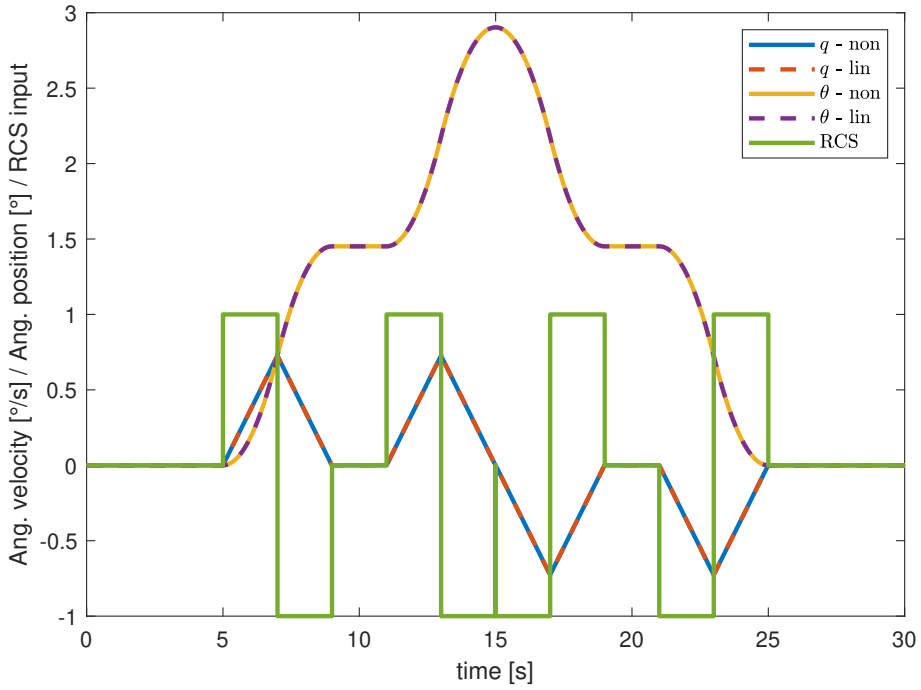

Figure 5.7: Comparison of linear and nonlinear system for the RCS control inputs

5.4.5 Sensitivity to uncertainties in the physical parameters

Because the physical parameters used in this thesis are mostly based on the assumption that comes from available information online or in textbooks, some information related to rockets is still very scarce to get hands-on in comparison to, for instance airplanes. The aerodynamic parameters are unique to each vehicle and since the SpaceX does not provide detailed parameters for their Falcon 9 LV, they had to be largely assumed with first initial assumption taken from [16] and than scaled to my model. The same applies to the thrust of the RCS because no details for any other landing booster with similar configuration are publicly available.

Although I would be able to get all parameters correct, there would still be a difference between my model and real rocket because it is effectively impossible to count and model every single part of the real system precisely. This is the reason why after modelling a physical system, it is common to conduct *Sensitivity analysis*. A goal of such analysis is to point out how the dynamics and model itself can change based on varying parameters.

I have used a toolbox available for SIMULINK which provides the Sensitivity analysis tool. With its help, I have generated random 50 samples for multiple parameters. Each parameter had a specific range for the generation, but generally, it has been $\pm 10\%$ of the originally assumed value. The output of these simulations was the angular deviation. I have conducted three different analysis, each for only one active system of actuators. For each simulation, predefined actuation was set. Conducted sensitivity analysis then measured how does the angular deflection change with different physical parameters. Results are shown in the figure below.

It is clear that the aerodynamic moment coefficients are the one that have the biggest influence. Second comes the Moment of Inertia followed by the frontal area of the rocket.

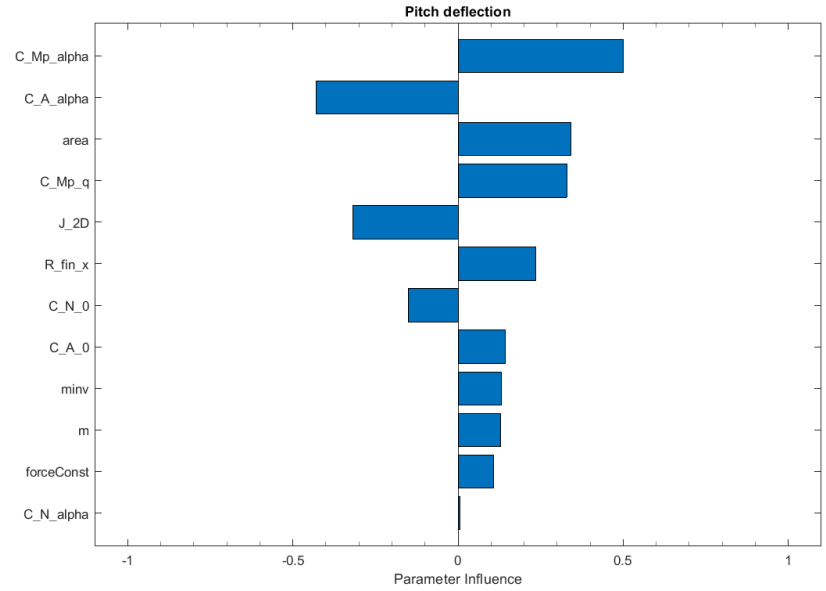

Figure 5.8: Sensitivity analysis for fins

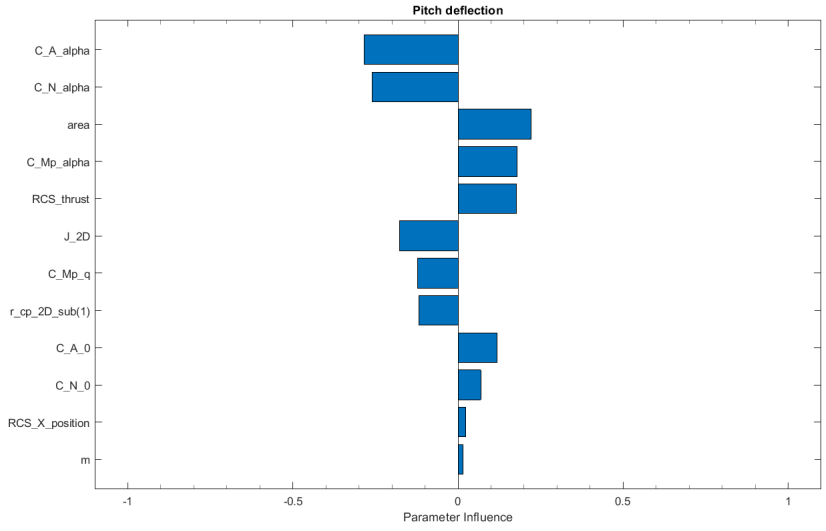

Figure 5.9: Sensitivity analysis for the RCS

5.5 Visualization

Although every simulation produces many plots with all important data, sometimes it is still difficult to understand or visualize, what is happening right away. Therefore I have created additional subsystem in the SIMULINK that focuses on showing the results of simulations in a form of 2D animation of the LV in an environment.

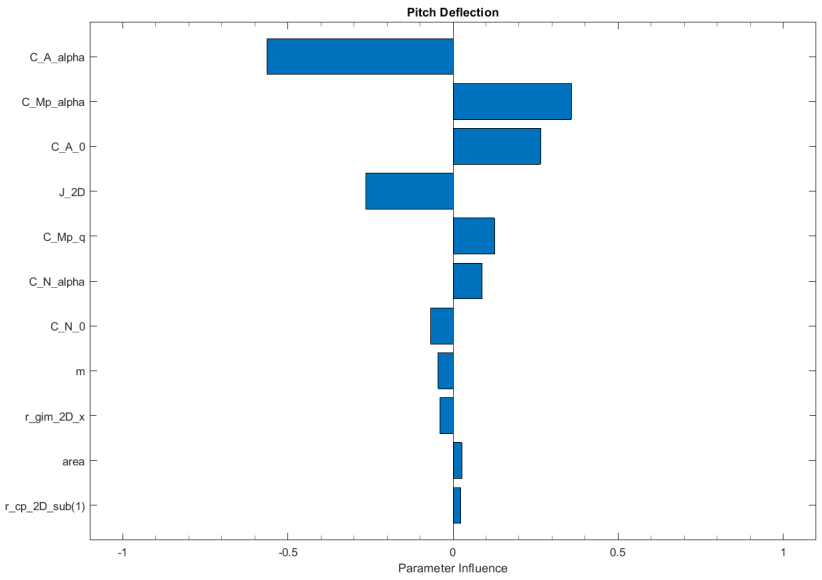

Figure 5.10: Sensitivity analysis for the TVC

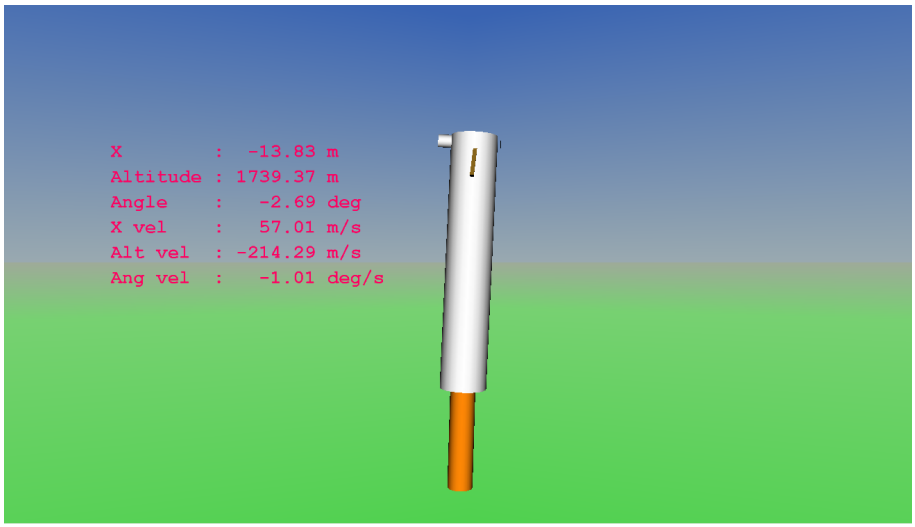

Figure 5.11: Showcase of visualization for 2D model

Chapter 6

Designing control laws

Until now, the model's available actuators have been used only for validating expected behaviour. But there was no way in which to deliberately stabilize the rocket in case of deflection from desired attitude. As modern rockets travel with astonishing speeds and possess extremely powerful engines, even the slightest control input has a huge impact on subsequent values of internal states of the LV. The rocket can either end up somewhere completely else or it can even be destroyed as great atmospheric forces during supersonic flight overcome the structural integrity if the rocket doesn't fly with the right attitude.

For this task, I will choose PID controller with following mathematical formula:

$$u = k_p e(t) + k_i \int_t e(\tau) d\tau + k_d \frac{de(t)}{dt}$$

Each of the constants is tunable based on controller designer needs. As derivative foresee the future state, it can help to reduce overshooting. On the other side, the integral part is responsible for mitigating small errors over time.

The disadvantage of using PID is it can be used only for Single-input Single-output (SISO) loops. That means, that I need to use multiple loops, each loop for one parameter. Loops can be either set parallel or in cascade, in which case, the inner loops usually control more rapidly changing parameter.

6.1 Pitch control

For vertically stabilizing the considered 2D model of a rocket, the pitch control is essential.

6.1.1 TVC control

Even though the pitch angle (θ) is at the end the main attitude parameter to be controlled, there is a pitch rate (q) to consider as well. Taking the q into consideration will help to keep the rotational rates in safe boundaries. Besides, the loop for pitch rate serves like the derivative part of the PID controller for the overall θ control.

Since the state-space representation of the model is available, I can easily obtain transfer functions of all parts of its inner dynamic via ***tf*** command in MATLAB. Now it is possible to see the effect of main engine TVC on the θ and q

$$H(s)^{\theta}_{\text{TVC}} = \frac{-1.076}{s},\ H(s)^{q}_{\text{TVC}} = \frac{-1.076}{s^2}$$

I will start with implementation of a simple P controller. To get better results, I've opted for Control Systems toolbox for MATLAB and SIMULINK and used it to autotune the P gain accordingly based on my preferences with respect to time constant and overshoot of the response. The computed value is

$$k_P = 0.4708$$

Implementation of pitch stabilization in SIMULINK is shown below. You can see, that I've implemented an inner feedback loop specifically for pitch rate, which is a common technique when implementing pitch damper for aircraft. Outer loop takes care of the deflection angle itself and makes sure it is stabilized.

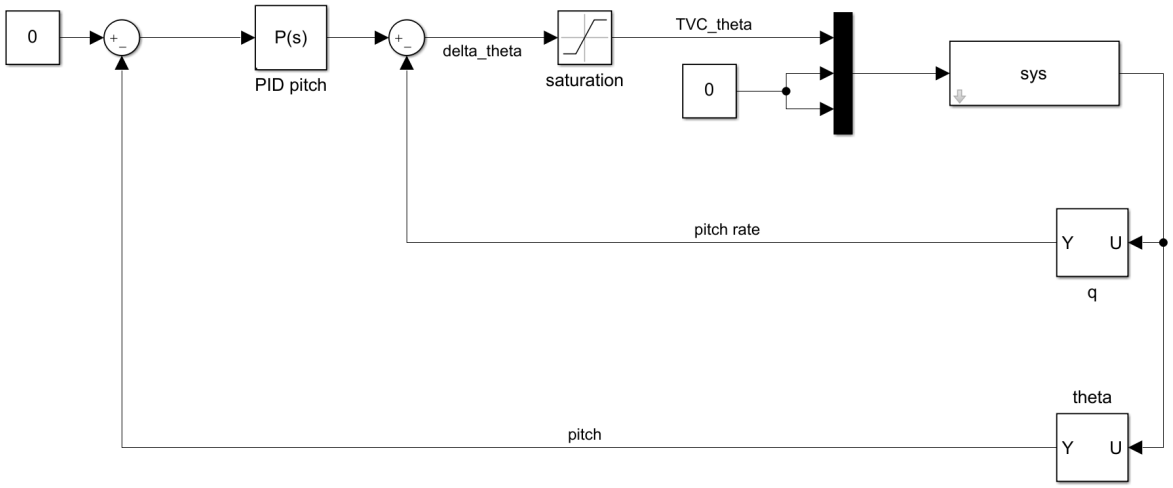

Figure 6.1: Implementation of P controller for pitch angle

When I set initial conditions of $\theta = 5$ degree and $q = 2$ degree/sec, the controller successfully stabilizes the system and it will return to its equilibrium fast enough to stabilize the rocket during its final landing burn which usually lasts between 30 to 35 seconds.

In the atmospheric flight, the attitude deflection of 5 degree could be lethal, as the atmospheric forces would destroy the structure. The ACS tries to keep θ as small as possible. During the considered final landing maneuver, however, the absolute angular position with respect to the landing pad can be indeed much larger. Therefore, throughout the thesis, the absolute angular deflection with respect to vertical axis is considered.

An integral part of the PID control does not have to be implemented as the integrator in the system itself will ensure zero steady-state error.

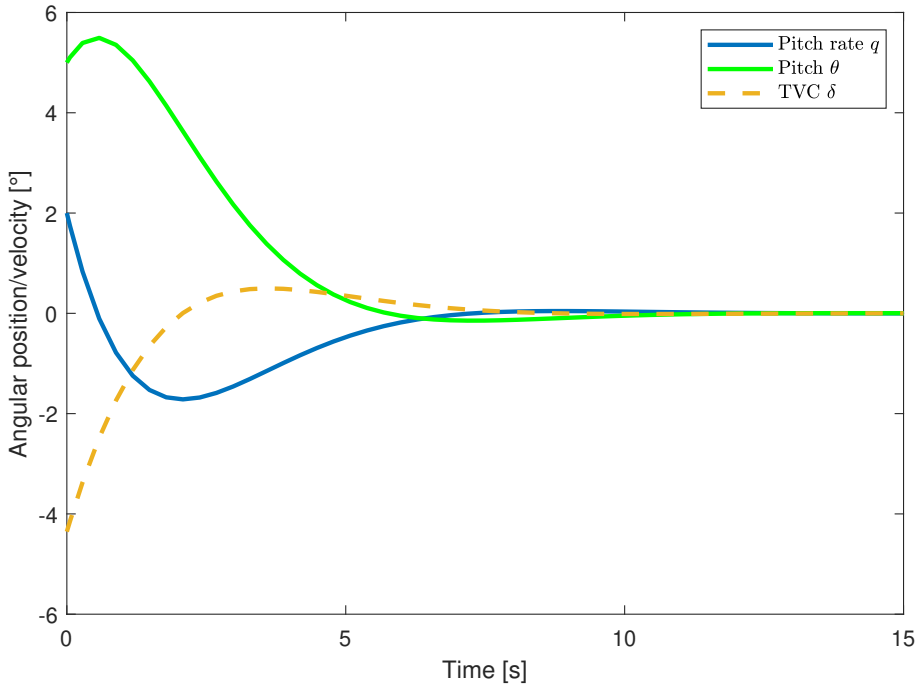

Figure 6.2: Response of linearized system with implemented P controller

6.1.2 RCS control

As much as the TVC is an important part to successfully land the rocket, the RCS still provides additional help to stabilizing the LV, especially in the very last phase of the descend, when the speed is much lower.

The transfer functions from the RCS to θ and to q are:

$$H(s)^{\theta}_{RCS} = \frac{0.006337}{s^2},\ H(s)^{q}_{RCS} = \frac{0.006337}{s}$$

The input is significantly smaller than what the TVC is capable of. Dynamics are unstable again and the control needs to be added. I have created a separated feedback loop (parallel to the TVC loop), which is controlled by the PD compensator with following gains.

$$k_{PRCS} = -423.2936,\ k_{DRCS} = -1970.7396$$

The comparison of the system's response in three situations is presented below. First one is by using only the TVC, second by using only the RCS and the last one shows the response for the case when both systems are used together.

Stabilization with only the RCS is expectedly much slower than by using the TVC since it provides less thrust. However, because the real cold gas thrusters mostly work as ON-OFF jets, I've decided to implement them as well in the ON-OFF manner. Specifically with the implementation of the Schmidt trigger, which partially prevents the inherent oscillatory behaviour of typical default ON-OFF switch by adding a dead-zone into it.

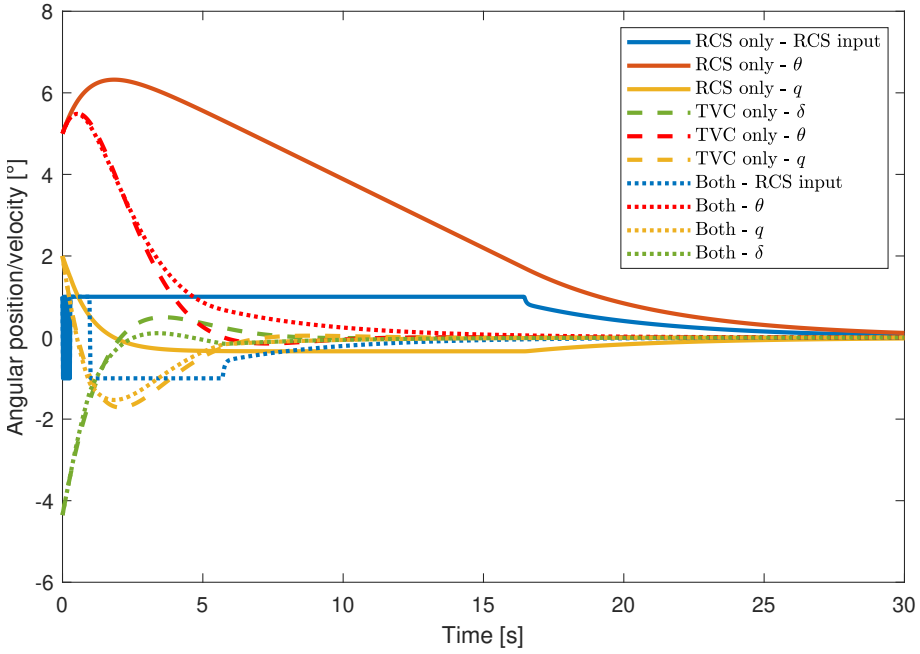

Figure 6.3: Comparison of response of linearized system with implemented controllers for RCS and TVC

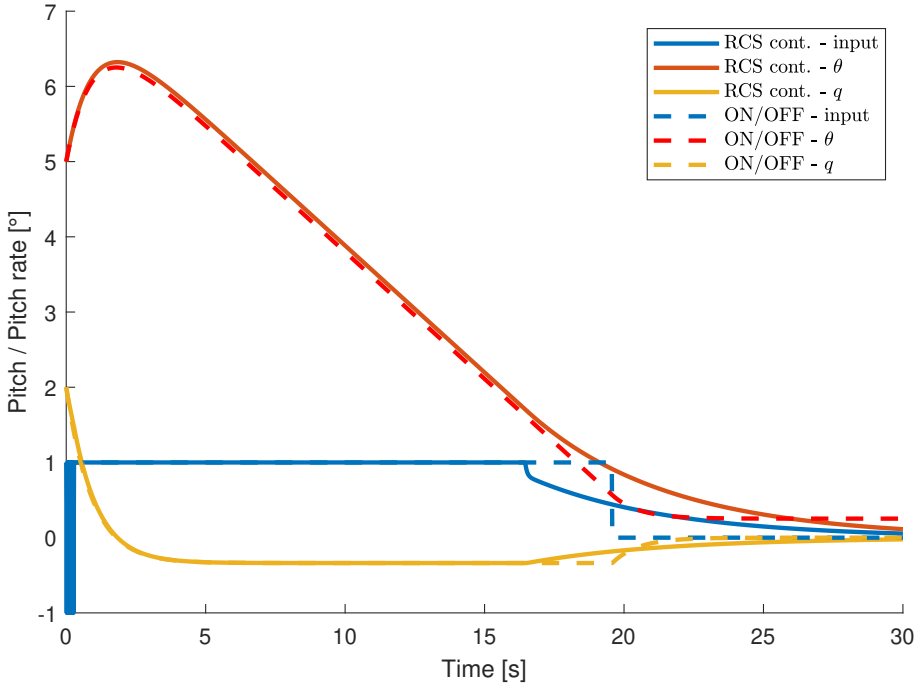

Figure 6.4: Response comparison for continuous or ON/OFF RCS controller

6.1.3 Fins control

Until the last 5 to 20 seconds of the flight, the LV's velocity is great enough for the vehicle to be prone to large aerodynamic forces. They can easily deflect the vehicle from desired trajectory when the TVC is not in used which is the case for most of the descent. In this phase of descent, the fins are the only way to control the attitude and that is why I also include them into my model and control architecture.

The transfer functions from fins to θ and to q at velocity of 200 m/s are:

$$H(s)_{\mathrm{F}}^{\theta} = \frac{-0.3134}{s^2 + 0.8044},\ H(s)_{\mathrm{F}}^{q} = \frac{-0.3134}{s + 0.8044}$$

I once again opted for the proportional controller. Since the fins have the greatest control authority in the mid-flight phase when the highest velocities are present, model was for this cause linearized for operation point in mid-air flight. Different initial conditions for the test are then set to: $x_0 = \begin{bmatrix} -200\,m/s & 0\,m/s & 0\,m & 4000\,m & 2\,deg/s & 5\,deg \end{bmatrix}^T$

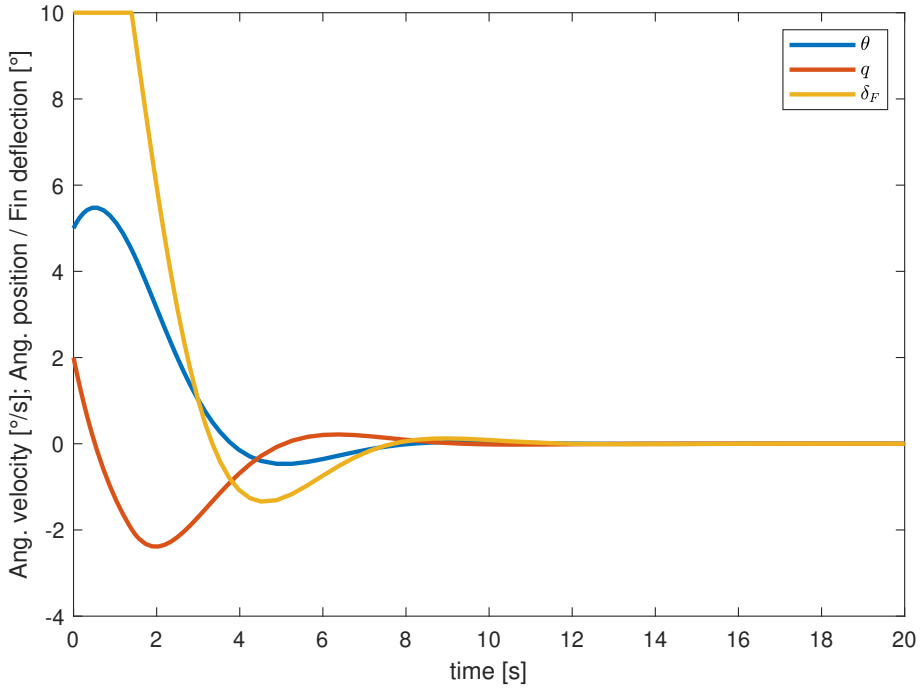

Figure 6.5: Behaviour of the system with grid fins in use

6.2 Altitude control

After stabilizing the attitude, another step is the altitude control achieved by throttling the thrust of the main engine accordingly. Hovering at the same altitude for a longer period of time is not desired as the real engines have too much thrust to hover the LV. The goal is to land by gradually decelerating the vehicle. Transfer functions from thrust to vertical velocity and altitude are:

$$H(s)_{\mathrm{thr}}^{v_y} = \frac{9.814}{s},\ H(s)_{\mathrm{thr}}^{r_y} = \frac{9.814}{s^2}$$

PID controller with large k_D gain is chosen as the LV needs to land smoothly without any overshoot in the simulations responses.

$$k_{Pthr} = 1.6,\ k_{Pthr} = 0.0002,\ k_{Dthr} = 7$$

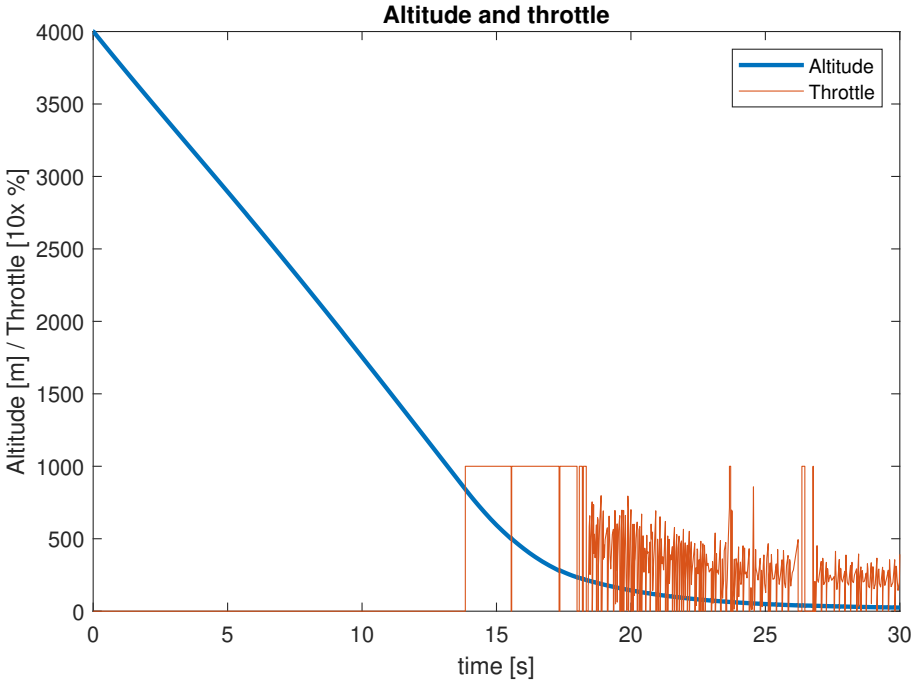

Figure 6.6: The progress of altitude and throttle with PD altitude controller in use

Initial conditions for simulation were $x_0 = \begin{bmatrix} -200\,m/s & 0\,m/s & 0\,m & 4000\,m & 0\,deg/s & 0\,deg \end{bmatrix}^T$.

The system is expected to achieve the landing pad in 30 to 35 seconds. The LV will stop at 20 metres, which is the altitude of the center of the rocket at the time of landing. The controller was tuned experimentally in the way that it decreases the altitude faster than by auto-tuning.

6.3 Implemented controls

In previous sections, I have described the process of designing classical control laws for essential attitude parameters of the landing rocket. First and foremost, the vertical stability is ensured. Secondly, the final altitude (and hence the velocity) is under control as well.

Next figure then shows the complete system's response for both initial condition and external disturbance, which could be present in the form of wind gust for instance. The test was run on the nonlinear model and the responses are almost identical to the ones for the linear model.

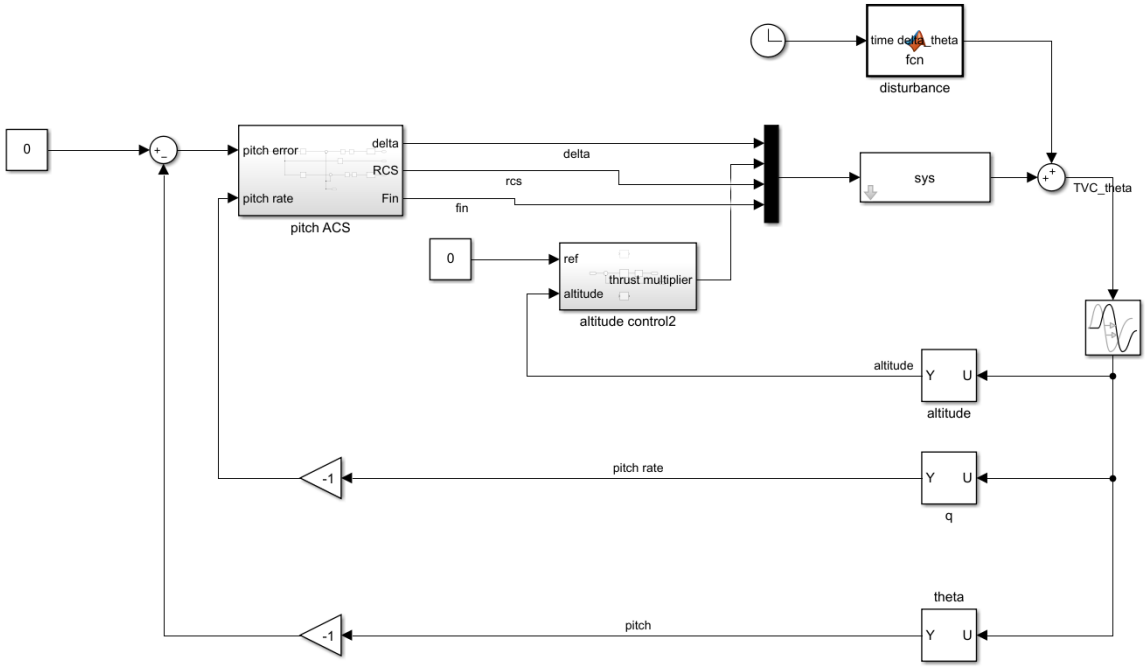

Figure 6.7: Entire control system designed in SIMULINK

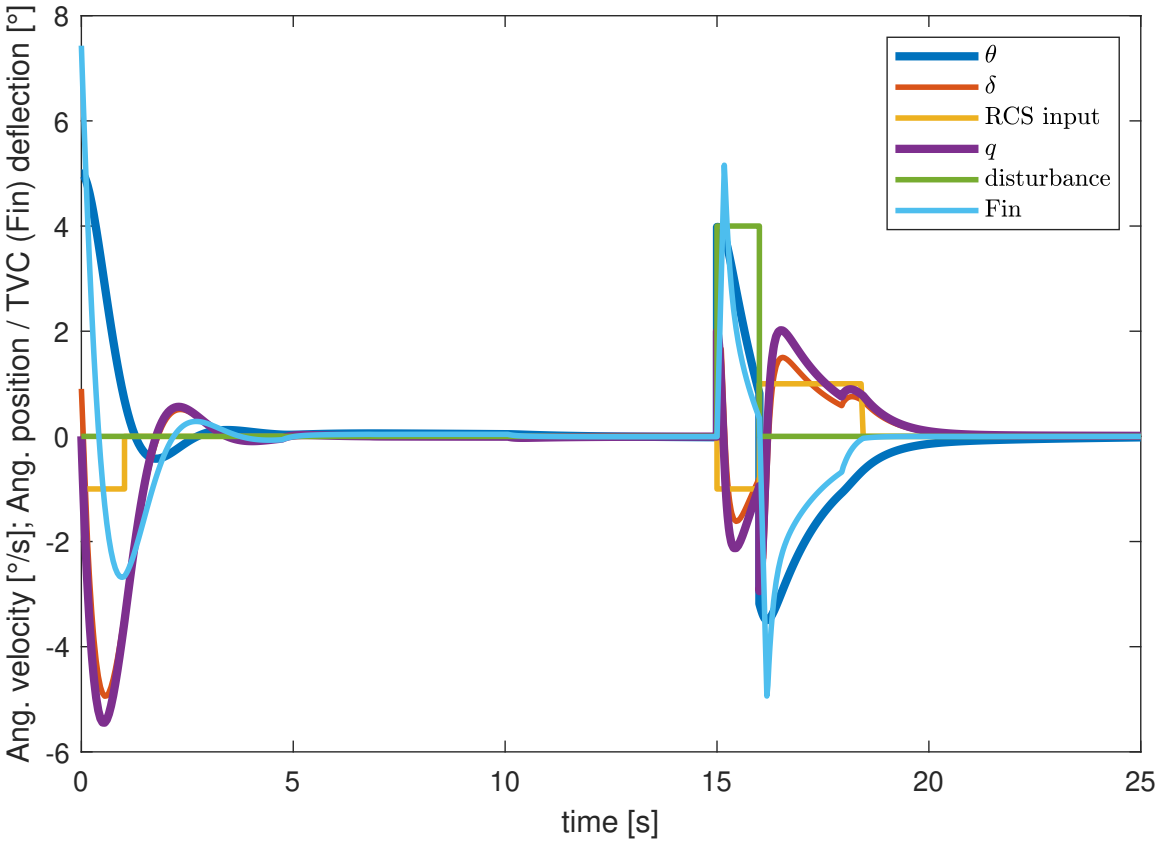

Figure 6.8: Full system's response to initial conditions and external disturbance

6.3.1 Analysis of stabilized system

Once the closed-loop feedback is implemented, it is common to verify stability and robustness of the system. Robustness is an important parameter regarding the performance of the control system which tells us what the system can withstand before it will become unstable. For instance the gain margin let us by which factor it is possible to increase the gain in open loop before the closed-loop loose its stability. Another parameter is the phase margin which refers to the amount of phase which can be increased or decreased before making the system unstable. Decrease of the phase can be for instance achieved by adding a transport delay. In fact, the phase margin is often taken as a protection against the uncertainty of the transport delay. It is worth to mention that in aerospace applications, these margins can be greater than in other industries as for commercial airlines the stability and robustness is extremely important.

Stability margins

To test the margins, the Bode plot is usually the tool to choose. It is common that Bode plot shows both Gain and Phase margin as displayed below. Although these metrics take into consideration stability of the closed-loop, they refer to applied changes in open loop. Therefore, to be able to perform the measurement, I need to disconnect the designed closed-loops and make them open before they can be analysed.

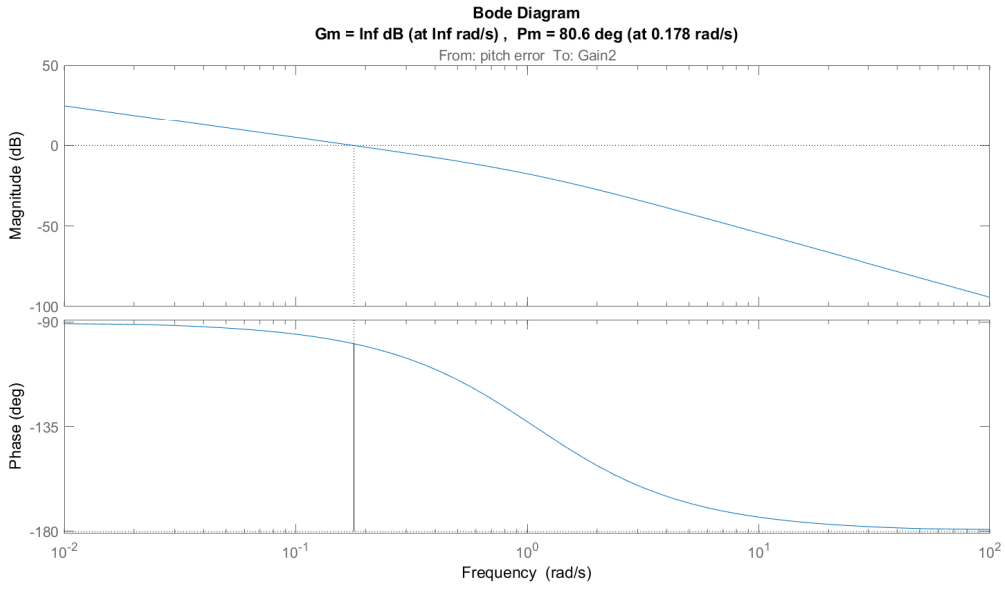

Figure 6.9: Bode plot of control loop for attitude

It is worth to mention that the infinite gain margin is only a theoretical value. In real life, there is always some certain value dependent on many variables such as for instance a bandwidth of actuator. In the considered model, there is however no such a limitation (actuator bandwidth is much larger with respect to motion dynamics and it is considered to be almost ideal), hence the infinite gain.

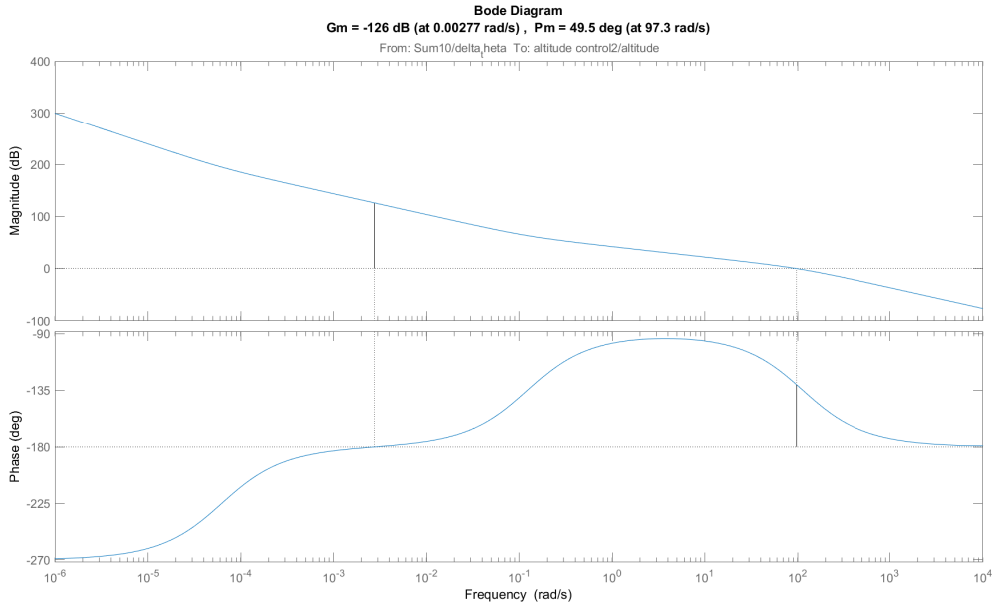

Figure 6.10: Bode plot of control loop for altitude

You can notice the negative gain margin. That means that the stability is lost by decreasing the gain. That does make sense because loosing the thrust will certainly results in instability of the closed loop controlling the altitude.

The following figure serves as display of poles and zeros for the mentioned closed loops and again confirms that none of them is present in the right half plane.

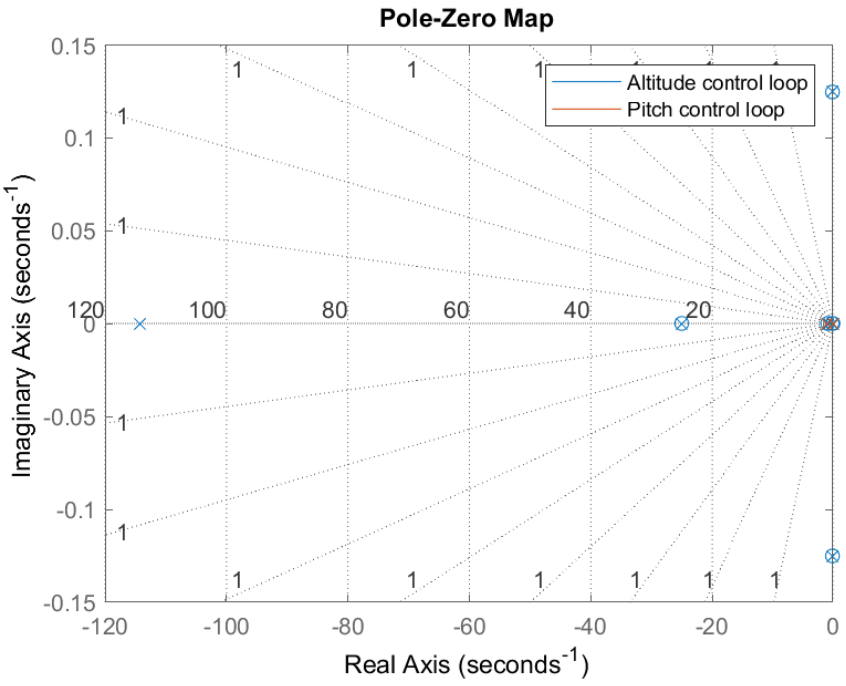

Figure 6.11: Poles-zero plot for altitude/pitch/position control closed loop

Nyquist stability and Root Locus

Finally, I present example of Nyquist plot and Root locus for the altitude closed-loop feedback. The Nyquist plot is once again common technique used in control engineering that incorporates system's frequency responses. There is important related equation N = Z - P, where N is the number of encirclement about -1, Z is number of closed loop poles in the right half plane and finally P represents the number of poles of the open loop in the right half plane. Since I know the dynamics of the open loop, and number of encirclement from the plots below, I can easily compute the number of the closed loop in right half plane. If that number is positive, it tells us that the closed-loop system is unstable. Together with the figure above which shows the location of Poles and Zeros of all open-loop systems, I can safely say that since the number of encirclement in all figures below is zero, there is no pole in the right half plane and the dynamics are indeed stable.

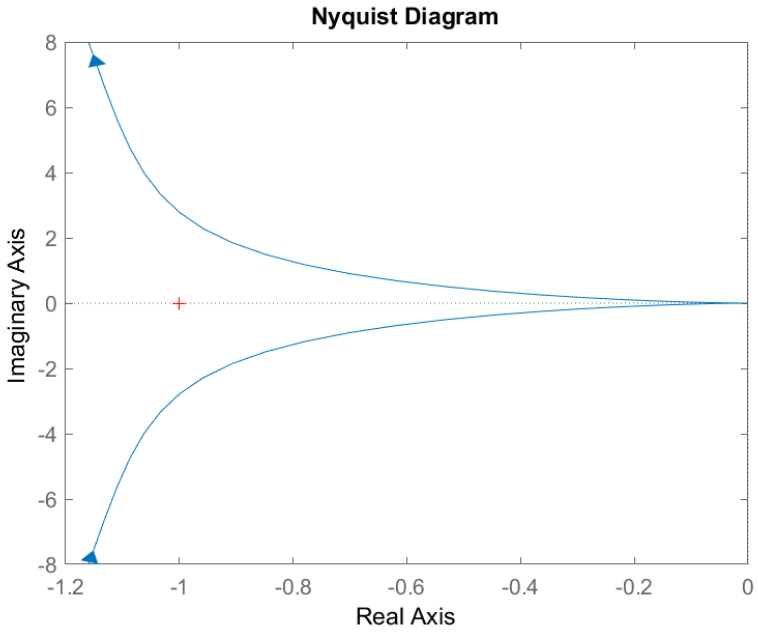

Figure 6.12: Nyquist plot of control loop for altitude

Subsequently, the Root locus how the poles move based on the changing the controller's gain. They again do not cross over to the right half plane.

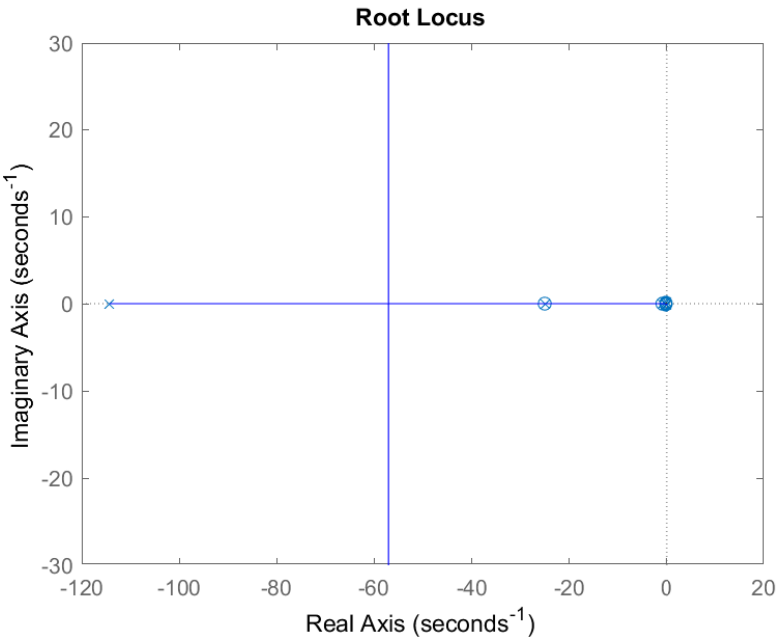

Figure 6.13: Root locus plot for altitude control loop

The plots for the attitude closed-loop feedback look very similar and once again confirm the stability.

6.4 Modern Control Methods

Although the PID control proven to be successful, I would like to use some of the modern control methods for comparison. There are multiple reasons for it. First, they work directly with the differential state equations in the time domain and second, they are inherently designed to work well with Multiple-input Multiple-output (MIMO) systems. The method of my choice is the Linear Quadratic Regulator (LQR).

6.4.1 Linear Quadratic Regulator

The LQR is based on the Full-state feedback, also referred to as a pole placement. That uses the same approach as classical methods, i.e. placing poles of the system to desired positions in the complex plane. The difference is that meanwhile the classical methods use the output values, the full-state feedback uses directly the states of the system.

Let me define system input u as

$$u = -Kx \tag{6.1}$$

where K is a closed-feedback matrix and x is the full-state vector. This definition can be subsequently substituted into the state-space equations

$$\dot{x} = (A - BK)x \tag{6.2}$$

$$y = (C - DK)x \tag{6.3}$$

The K needs to be set up in such a way, that the poles of matrix $\Big[A - BK\Big]$ will be as desired. Determining where to put these poles, however, can be a tedious process and that is where the LQR controller excels. The LQR is an *optimal* regulator, which sets the feedback matrix K directly based on desired performance while minimizing the control energy. That is done by formulating a quadratic cost function, which takes into consideration both control energy and transient energy. The cost function is defined as

$$J(t, t_f) = \int_t^{t_f} \left[x^T(\tau)Qx(\tau) + u^T(\tau)Ru(\tau)\right] d\tau \tag{6.4}$$

where Q is symmetric, positive semidefinite matrix called *state weighting matrix* and R is a symmetric, positive definite matrix called *control cost matrix*. Variables t, t_f are *initial* and *final* time. [20]

The feedback matrix K is computed as

$$K = R^{-1}B^T P \tag{6.5}$$

where P is a solution of *algebraic Riccati equation*

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \tag{6.6}$$

In MATLAB, it is possible to use command *lqr* with parameters being LTI system and R and Q matrices. The control designer can arbitrarily choose the values of Q and R to one's preference. [20]

In my case, I've empirically chosen the control matrix R to be diagonal matrix with values [14000000 1, 1 100000, 1000 10000000] and state matrix Q to be a diagonal matrix with following values on diagonal: [1000000 1 50000 10000000].

Calculated feedback matrix K is then determined as

$$K = \begin{bmatrix} 0.0000 & -0.026 & 0.0000 & -0.0000 & -2.4961 & -3.3550 \\ 3741.6659 & -0.0000 & 0.0000 & 316.2277 & -0.0000 & -0.0000 \\ 0.0000 & 0.0030 & -0.0000 & 0.0000 & 0.2946 & 0.3960 \\ -0.0000 & -0.0000 & 0.0000 & -0.0000 & -0.0000 & -0.0000 \end{bmatrix} \tag{6.7}$$

It is possible to check that the eigenvalues of the closed-loop dynamics matrix [A-BK] are all in

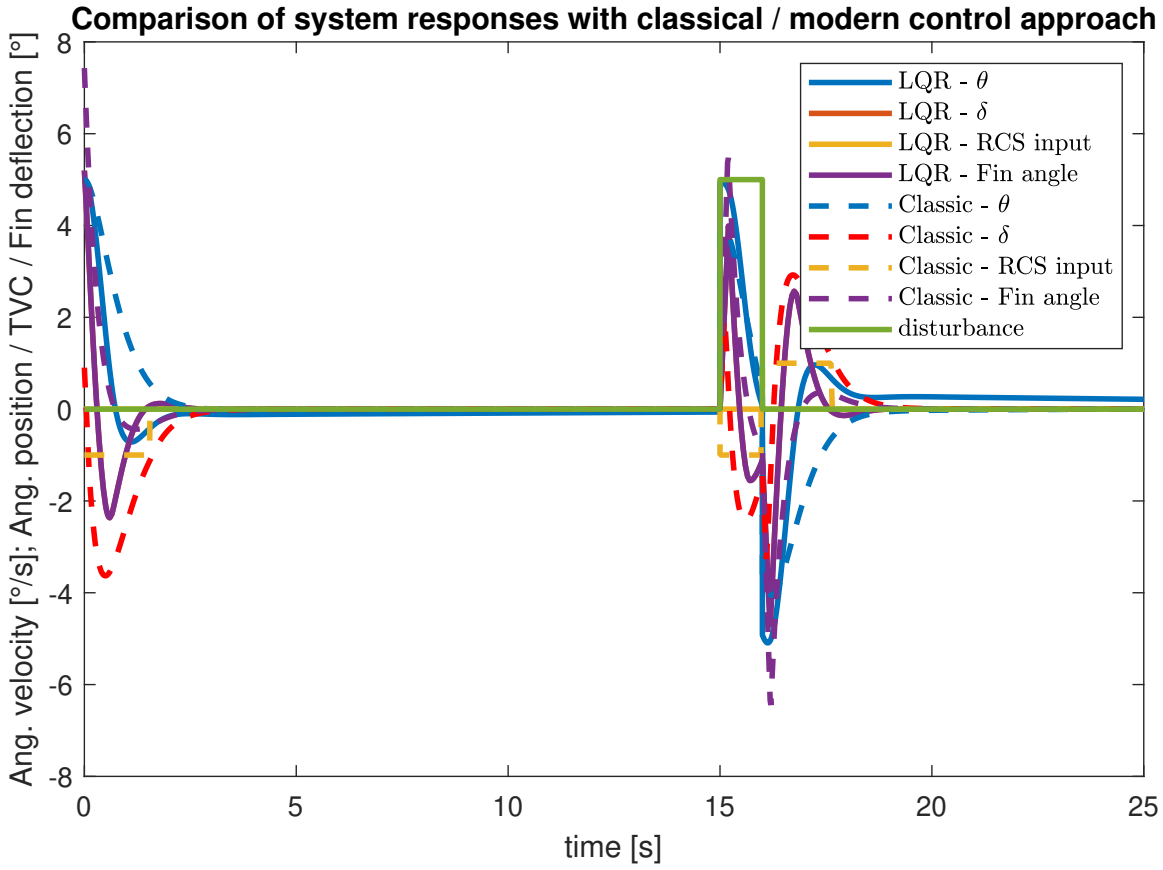

Figure 6.14: Response comparison of classical / modern control approach

the left-half plane

$$eig(A - BK) = \begin{bmatrix} -36719.1289 + 0.0000i \\ -0.0845 + 0.0000i \\ -1.3049 + 1.3047i \\ -1.3049 - 1.3047i \\ -0.0394 + 0.0393i \\ -0.0394 - 0.0393i \end{bmatrix} \tag{6.8}$$

which means the system is asymptotically stable. SIMULINK scheme for the full-state LQR feedback is shown below.

6.4.2 Gain Scheduling

Due to the great effect of the vehicle's velocity on aerodynamic forces created among others by control surfaces, I have decided to include gain scheduling. It is a common technique in aerospace used to control the system in different regimes. For instance, different regimes for aircraft involves steady climbing, cruising, landing, coordinated turn etc. As the name suggests, the controller gains are scheduled to be used based on the conditions such as the airspeed, altitude and others. I created two state feedback matrices for the LQR control. First takes into account the atmospheric forces at velocity of 200 m/s, the other does not. The latter is turned

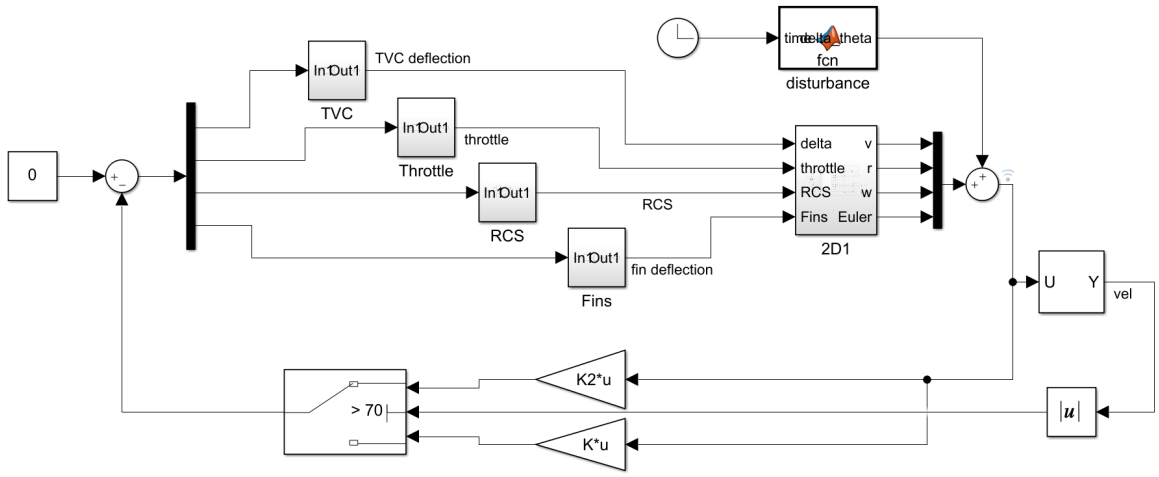

Figure 6.15: SIMULINK scheme of LQR control

on when the velocity decreases below 70 m/s. This way, the control is improved and it also helps to determine, which actuators is better to use at specific phase of the flight. For instance, the RCS is not used during the high-velocity flight, where it has only marginal effect and could be wasted before the final phase.

There are several methods for designing the proper gain scheduling. One, the most basic one, is ad hoc, based on designer's own thresholds for each flight regime. Then there are some more advanced such as for instance, technique designed by Pierre Apkarian. I have however opted for the simple empirical method.

The gain scheduling was put in effect only for the state-feedback as PID control was during testing effective enough to stabilize the attitude at all considered conditions.

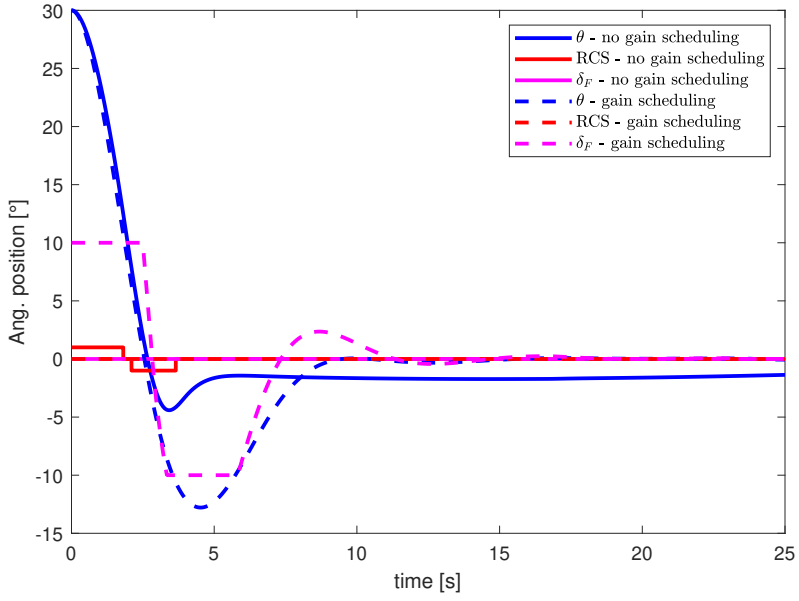

Figure 6.16: Comparison of responses for systems with and without gain scheduling implemented

Chapter 7

Control actuators limitations

The aim of this chapter is to experimentally test and recommend the best actuators' properties such as relative position on the LV's body, their power and the speed of response. During this process, I will inspect every each of them, combine them and will try to find the best option available.

7.1 Main Engine

The first to start with is the main engine and its thrust vectoring unit as they hold the greatest control authority and have the largest impact on overall attitude control. As I have mentioned earlier, I will expect that the main engine is liquid and it is possible to reignite the engine and throttle its thrust so the LV will begin the landing burn at the right time without any hassle. I will not consult the amount of maximum thrust, because engines differ for each rocket as clients of the companies providing launch services have usually very specific requirements about the final orbit of their spacecraft. It is thus expected that the maximum thrust is more than sufficient. I will, however, discuss the option of throttling as this could have a potential affect on the landing maneuver. If the thrust can be lowered down to be equal or lower than the gravitational force, then it is possible to control the descend more precisely or even hover if needed to be.

To best evaluate the options, I've decided to research some of the currently or in the near future available rocket engines to see whether they allow it. In order to keep this task feasible, I only consider landing of the LV with similar parameters like the one I work within this thesis. Therefore too powerful (more than 1 MN without possibility to throttle down) or too weak (provide thrust equal to less than 10 % of F_g) engines are not shown in this comparison. Also, only gimbaled engines are included. The comparison table was inspired by a table at [21].

Although some of the stated engines have thrust smaller than F_g, it is not necessarily an issue as it is possible to use multiple units (both SpaceX and RocketLab use the first stage with 9 engines). In such a scenario, it is actually possible to even use non-throttleable engines as the system can just turn off some of them to lower the throttle down. This solution will however not be as good as actually continuous throttling.

	Origin	Designer	Propellant	Weight [kg]	Thrust [kN] (at SL)	Throttle range [%]	Gimbal [°]
Merlin 1D	USA	SpaceX	LOX / RP1	470	854	40 - 100	?
Raptor	USA	SpaceX	LOX / LCH4	1,500	1,700	?	?
RD-191	Russia	NPO Energomash	LOX / RP1	2,290	1,920	27 - 105	8
RS-25	USA	Rocketdyne	LOX / LH2	3,527	1,860	67	10.5
Rutherford	USA	RocketLab	LOX / RP-1	35	24	?	?
YF-20	China	AALPT	N2O4 / UDMH	715	731.5	?	10
BE-4	USA	Blue Origin	LOX / LCH4	?	2,400	?	5
BE-3	USA	Blue Origin	LOX / LH2	?	490	?	?

Table 7.1: Comparison of rocket engines

7.1.1 TVC

There are three types of actuators which can be used for the TVC. Hydraulic, electronic and pneumatic. According to my research, pneumatic actuators are rarely used in the space industry, hence will be omitted in the following description and consideration.

Hydraulic servo actuators

The most popular type of servos in rocketry (and not only there) are hydraulic actuators. They consist of a piston inside a hollow cylinder. The incompressible fluid is pumped into the cylinder, moving the piston. Reversed movement is then created by spring force or pressure from the other side created similarly by liquid. [22]

Advantages of hydraulic actuators are that they are suited for high-force applications and can hold force and torque constant thanks to the incompressibility of used fluids. The pumps and motors can be located further away with minimal loss of power. On the other hand, they suffer from leakage of the fluids which decreases the efficiency. Besides that, they need a lot of supporting parts such as a fluid reservoir, motors, pumps, valves and so on. That adds to the weight and complexity of the LV.

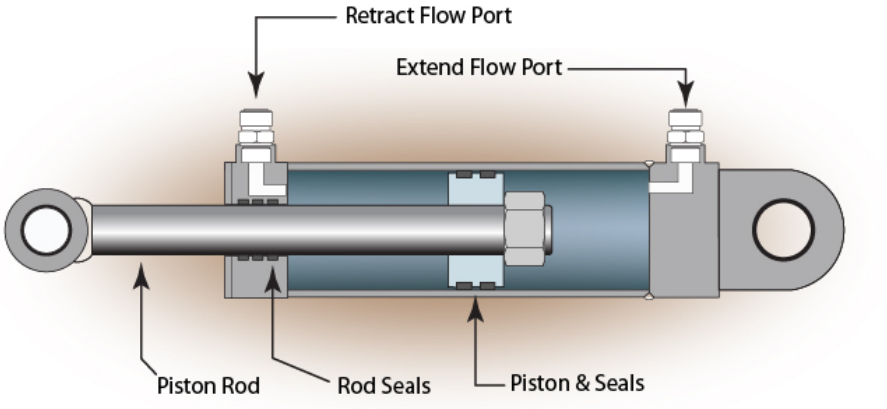

Figure 7.1: Hydraulic Linear Actuator [23]

For an information source, I've looked up to one of the most prominent manufactures in the

industry, Moog. Their TVC actuators have been and will be used on dozens of LVs and spacecraft (including Falcon 9) and unlike others (Airbus Space, Honeywell), they provide brochures with technical details to public. [24]

Electric linear actuators

Until recently, hydraulic actuators have been automatically considered as to go type of servos when designing a rocket. RocketLab has changed that with their electric actuators for the Rutherford rocket engine. This is known as the first use of such an actuator in modern rockets and it is possible due to enhancements in the battery industry.

Electric linear actuators are created by different components than the hydraulic ones. There is an outer and inner tube. The latter moves with the rotation of a screw and a nut inside the actuator. This causes the extension or retraction. [22]

A great advantages of this mechanism are speed and its superior precision. "Their setups are scalable for any purpose or force requirement, and are quiet, smooth, and repeatable." [22]

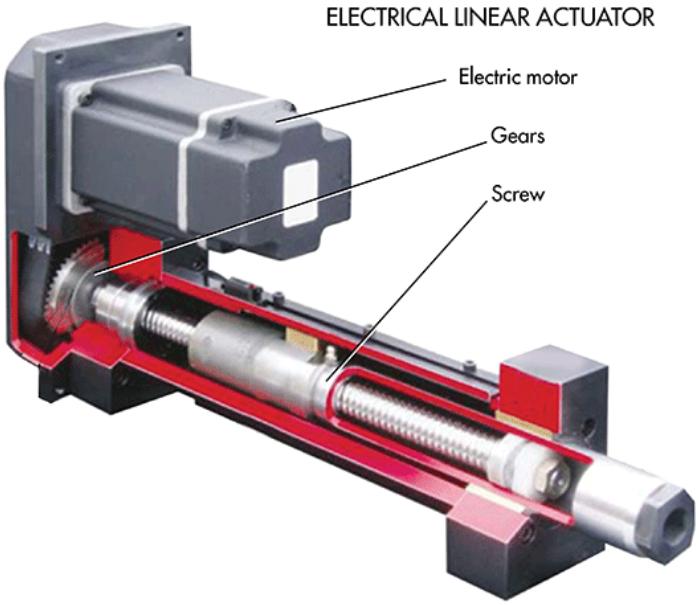

Figure 7.2: Electrical Linear Actuator [22]

TVC limits and specifications

Based on averaging data from real SpaceX's missions, the time constraint for the landing burn is approximately 35 seconds. Because it is desired for the LV to be vertical a certain period before final touchdown, I will set the time constrain for the θ stabilization to be 30 seconds.

In terms of specifications, they are not easy to find, however I was able to obtain a few of them for the Space Shuttle Main Engine (SSME) (RS-25). The minimal and maximal rates are 10 deg/sec and 20 deg/sec, respectively. [25] For the Saturn's F-1 engine, the rate was 5 deg/sec. [26]

7.2 RCS

In general, cold gas thrusters are used for the RCS. The motion force is simply created by the propulsion of the pressurized gas. The gas is typically inert and the most widely used one is nitrogen. As mentioned in the chapter 4, I will assume the thrust to be 1 kN based on the reasons mentioned before. Comparing to the response of TVC or fins, RCS's response is considered to be instantaneous.

7.3 Fins

Most of the information about the fins have been mentioned in the previous chapters. I will, therefore, use this section to mainly focus on the parameters of the fins system used in my 2D model. The force exerted in the respective axis is highly dependable on the velocity and angle of deflection.

The aerodynamic coefficients (for moment and drag) were estimated to be as follows:

$$C_{M_F} = 0.354$$

$$C_{D_F} = 0.005$$

The area was chosen to be 1 x 1 meters and the distance of the AC from the CG is considered to be 23 meters.

7.4 Actuator dynamics

To increase the similarity with the real systems, I will assume the actuators not to be ideal ones that would achieve the desired performance immediately. In real life, it takes certain time to achieve the requested level of performance or level of deflection. According to [13], the common time constants for the aileron and rudder servos for fighter aircrafts are around 0.05 s. Therefore, I will assume the same value to be my default assumption for used actuators. I will model the dynamics for the fins and for the TVC in such a way.

Beside the time constant, the other considered parameters in setting of the actuator are saturations (max values) and rate limits.

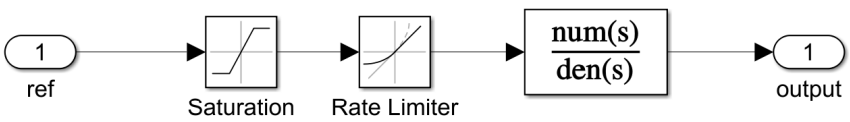

Figure 7.3: Simulink block for actuators

7.5 Sensors delay

As mentioned in the second chapter, I introduced a sensor parameter accounting for a delay between sensing the measured value and propagating this information to control systems. As part of actuators and sensors setup evaluation, I will consider several values for such a delay and put them into test to see how they effect the response of the system. For this evaluation, I considered following time constants: [1 s, 0.1 s, 0.01 s, 0.001 s, 0.0001 s]

7.6 Specific recommendations

The details and information about each of the used actuators were presented, it is time for rigorous simulations to find out, what combination of parameters is necessary for individual expected performance. The simulations were run on the nonlinear model with PID loops in place and the examined parameters are working range (saturation), rate limit, delay of the sensor and actuator's time constant.

There were several batches of simulations run. The set of batches included the following parameters:

- Use of ACS
  - only TVC in use
  - TVC + RCS in use
  - TVC + Fins in use
  - RCS + Fins in use
  - all parts of ACS in use
- Regime of flight
  - Hovering
  - Vertical landing

The table is built in the way, that it recommends the setting of actuators depending on the level of expected pitch deviation the rocket should be able to stabilize.

Pitch [deg]	**TVC Saturation** [deg]	**Rate limit** [deg/s]	**Sensor delay** [s]	**Actuator time constant** [s]
1	3	2	0.1	0.5
2	3	2	0.1	0.5
3	3	2	0.1	0.5
4	3	2	0.1	0.5
5	3	2	0.1	0.5
8	3	2	0.1	0.5
10	3	2	0.1	0.5
15	2	2	0.1	0.5
20	5	3	0.1	0.5
25	3	2	0.1	0.5
30	3	2	0.1	0.5
45	3	2	0.1	0.5

Table 7.2: Minimal limits for actuators for regime of hovering

Pitch [deg]	**TVC / Fin Saturation** [deg]	**Rate limit** [deg/s]	**Sensor delay** [Hz]	**Actuator time constant** [s]
1	1	2	0.1	0.5
2	1	2	0.1	0.5
3	2	2	0.1	0.5
4	2	2	0.1	0.5
5	3	2	0.1	0.5
8	4	2	0.1	0.5
10	5	1	0.1	0.5
15	8	1	0.1	0.5
20	10	1	0.1	0.5
25	15	1	0.1	0.5
30	15	1	0.1	0.5
45	20	1	0.1	0.5

Table 7.3: Minimal limits for actuators for regime of the vertical landing flight

To better understand the given results, let me present a specific example. If there is a landing vehicle deflected by 3 degrees, and the maximal TVC deflection is 2 degrees and maximal rate of the TVC's deflection is 2 deg/s, then the control system as currently set up will not be able to stabilize the rocket within the given time limit for the landing. If the minimal values would be 3 degrees and 2 deg/s for the TVC deflection, rate of the deflection respectively, the vehicle can be stabilized within the requested time frame.

This applies to the required bandwidth as well. If, in any case, the delay of the sensor would be smaller than 0.1 sec, then the system would struggle to be stabilized in time. The table also shows that if the actuator time constant (time to achieve at least 63 % of the desired value) is lower than 0.5 s, the vehicle is not stabilizable.

Chapter 8

Validation of control laws and actuators

In previous chapters, I suggested the position of actuators and sensors that would support a successful vertical landing of the LV. I subsequently introduced a 2D model of such LV, whose purpose was to verify the rocket's behaviour and to help to design control laws.

8.1 Validation model

To verify the correct function of those designed control laws, I will use a more complex, nonlinear model of a rocket, developed by Matyas Lustig. [16] This model is defined in 3D and several coordinate frames related to launching space vehicle is present and interconnected. It is not simplified as much as developed 2D and it uses quaternions for the attitude determination. However, it has been designed with only ascending phase of the flight in mind. Original governing equations of motion are listed below.

$$\dot{\vec{v}} = \frac{1}{m}\left[T_B^{ECI}(\vec{F}_{atm} + \vec{F}_{thr}) + \vec{F}_g\right] \tag{8.1}$$

$$\dot{\vec{r}} = \vec{v} \tag{8.2}$$

$$\dot{\vec{\omega}} = (\hat{J})^{-1}\left[\vec{M}_{atm} + \vec{M}_{thr} - \vec{\omega} \times \hat{J} \cdot \vec{\omega}\right] \tag{8.3}$$

$$\begin{bmatrix} \dot{q_0} \\ \dot{q_1} \\ \dot{q_2} \\ \dot{q_3} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix}\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{8.4}$$

Although it has controllable surface fins and the TVC, it does not contain all the systems for stabilization of landing rocket introduced until now. Therefore appropriate changes have been

made including:

- removed tail fins
- added grid fins on the top
- changed calculation of force and moments exerted by fins to vary with airspeed
- changed calculation of α
- added the RCS
- added wind model

From now on, I will use this model. Purpose of this chapter is to validate designed control laws and recommended parameters for actuators and sensors. Therefore, there is no need to linearize the model and all validation tests will be run on the nonlinear model.

8.2 Simple validation tests

Best way to begin with validation is to introduce some basic tests and compare them to responses of the same test run on a simplified 2D model. The visualization tool will be also used as a part of every simulation to help with validation, however, the screenshots of it will not be shown since the static images cannot reproduce the entire information.

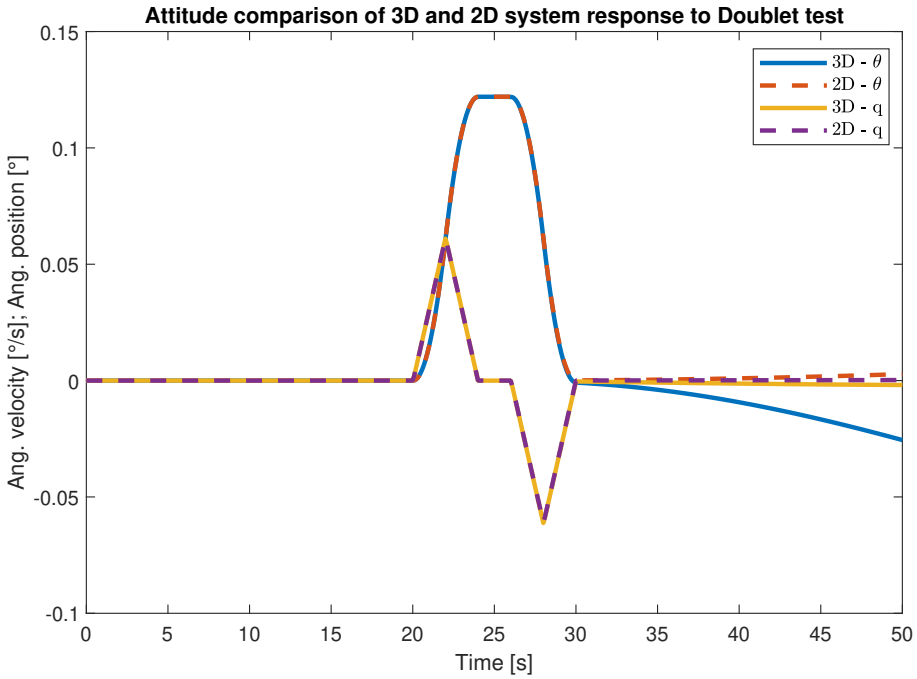

Figure 8.1: Response comparison for TVC Doublet test

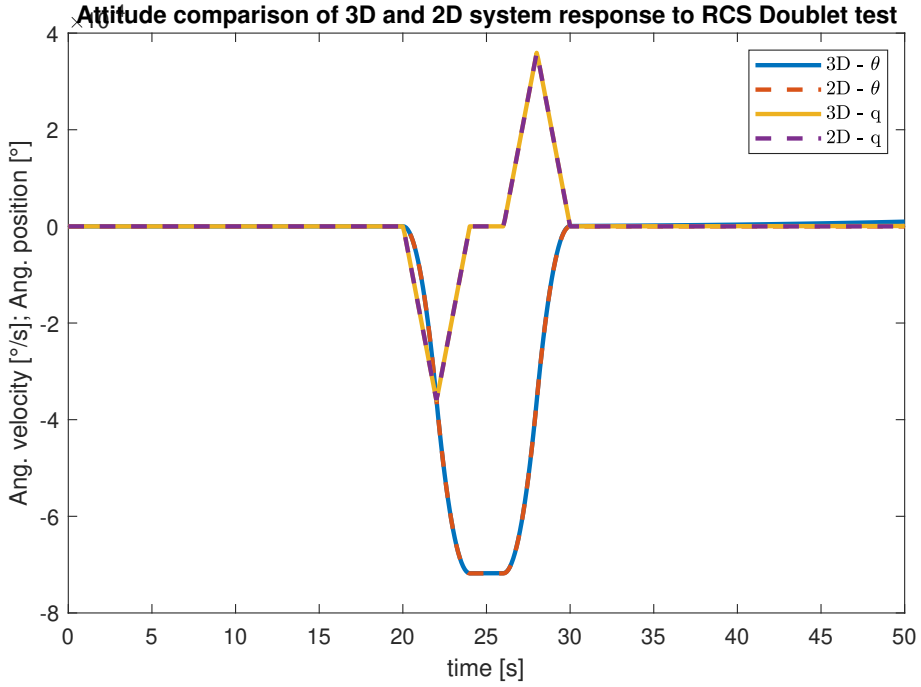

Figure 8.2: Response comparison for RCS Doublet test

8.3 Inclusion of a wind model into the environment

To improve the real simulated conditions for the final valuation test, an advanced stochastic model of wind dynamics is included. The model was inspired by the NASA HL-20 Model in the SIMULINK [27].

It is composed of three sub-components, each represented by one specific wind block from Aerospace toolbox for the SIMULINK. Individual blocks are following:

1. The Wind Shear Model block adds wind shear to the model.
2. The Discrete Wind Gust Model block implements a wind gust of the standard "1 - cosine" shape
3. The Dryden Wind Turbulence Model (Continuous) block uses the Dryden spectral representation to add turbulence to the aerospace model by passing band-limited white noise through appropriate forming filters.

Each of the mentioned blocks requires default setting. Hence, default setting for the Wind Shear model is wind speed of 3 m/s at altitude 6 metres. The same applies for the wind turbulence model. The gust model is set to be initiated 20 seconds with additional 3 m/s after the simulation began. For the reference, the Falcon 9 will not launch if the sustain winds at 50 m excess the 15.5 m/s [28]

All three blocks output the stochastic wind speed in all three axis. This wind velocity is then added to the α. The turbulence block also produces the angular rates that affects the nominal rates.

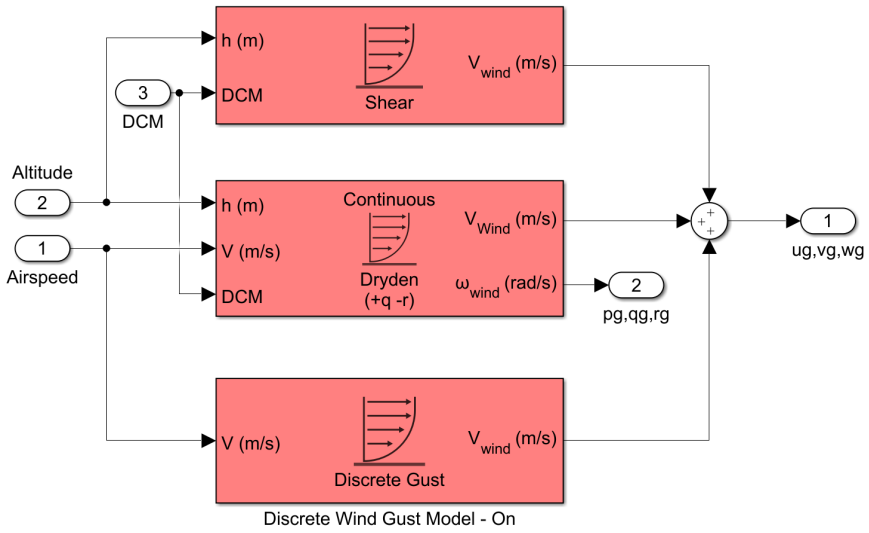

Figure 8.3: Implementation of stochastic wind dynamics

8.4 Validation on real telemetry data

The second phase of validation testing is to simulate the model with initial conditions taken from the real flights. These data are acquired from [10] and as it is stated by the author, they are taken from the online webcast of SpaceX's YouTube channel. I have taken data which also include timestamps of events during the flight. I've been particularly interested in landing burn timestamp since that is the final period of the flight, which was the topic of this work.

Subsequently, all data available for landing burn from all available flights have been averaged. Following parameters of the LV have been obtained

vertical velocity	- 291	m/s
horizontal velocity	41	m/s
vertical position	4,880	m
horizontal position	-500	m
angular velocity	0	deg/s
angular position	30	deg

Table 8.1: Parameters of launch vehicle at the beginning of landing burn

The horizontal position was estimated as I could not find this information in the dataset. Validation simulations also include the stochastic wind dynamics which affect both velocity and rotations.

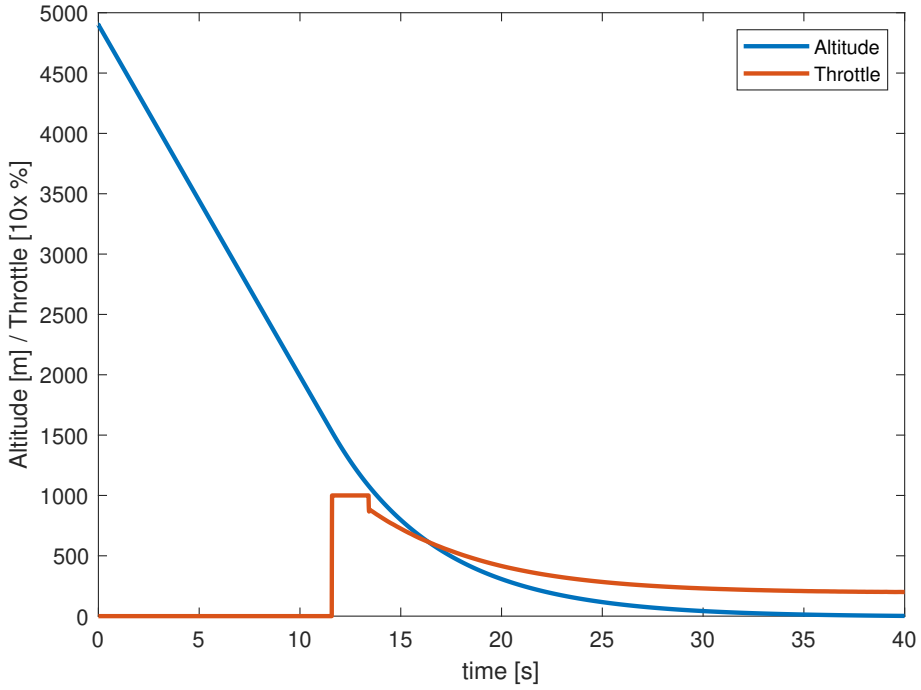

Figure 8.4: Altitude response for 3D nonlinear model with real initial conditions

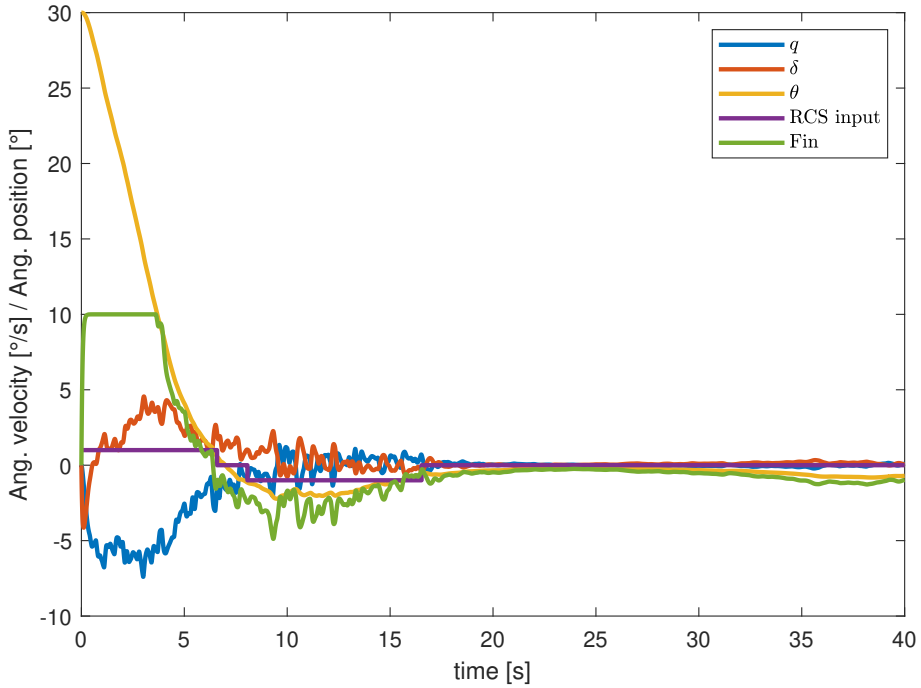

Figure 8.5: Attitude response for 3D nonlinear model with real initial conditions

The LV at the beginning continues at the free fall and it reignites the engines at time T = 12 seconds. Control systems then stabilize the rocket and land smoothly. It is worth to mention that the final attitude deflection does not have to be literally zero because the landing legs together with the LV's low CG are able to withstand touchdown with estimated deflection up to about 5 degrees. Good example is landing on a drone ship that floats in the open ocean and hence is inherently deflected as well.

In case of the LQR control, the response to altitude control is similar to the response to the classic control method. Because the gain scheduling was used in this case, there is a switch between two controllers around 18 seconds into the simulation. This helps to use actuators accordingly. At the beginning, the fins control the attitude alone. When the velocity decrease significantly, the RCS is turned on and used to stabilize the vehicle alongside the TVC before the touchdown.

The LQR control among others brings the advantage of bringing all states to zero in requested time-frame. Therefore, it also successfully controls the horizontal position which was not included in the proposed Proportional-Integral-Derivative (PID) control that was primarily focused only on the ACS.

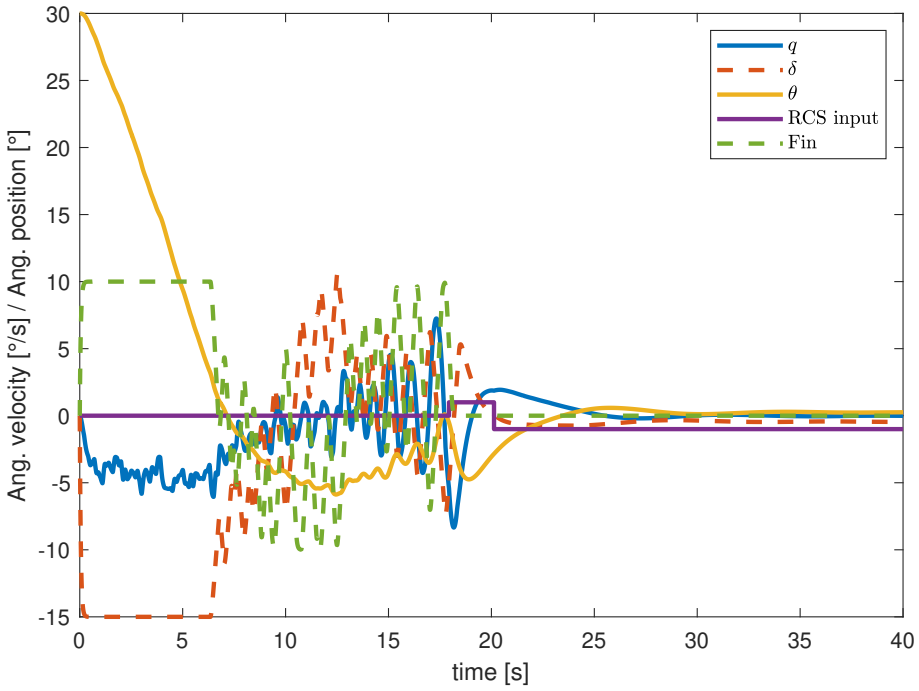

Figure 8.6: Response of the 3D nonlinear system's attitude with LQR in place

Control via the LQR also significantly helps with decreasing the horizontal velocity at touchdown to zero in order to prevent topple. Thus even though the attitude control can seem performing slightly worse in comparison to PID, it still accomplishes the task and proves to be overall better technique because it brings all the states to zero and tuning for MIMO system such as this is more straightforward.

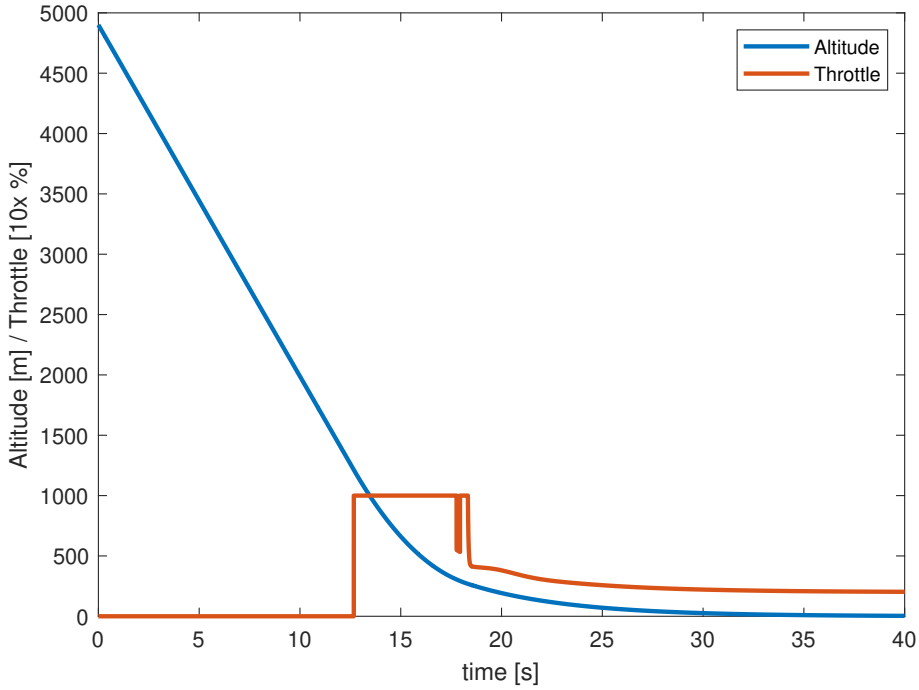

Figure 8.7: Response of the 3D nonlinear system's altitude with LQR in place

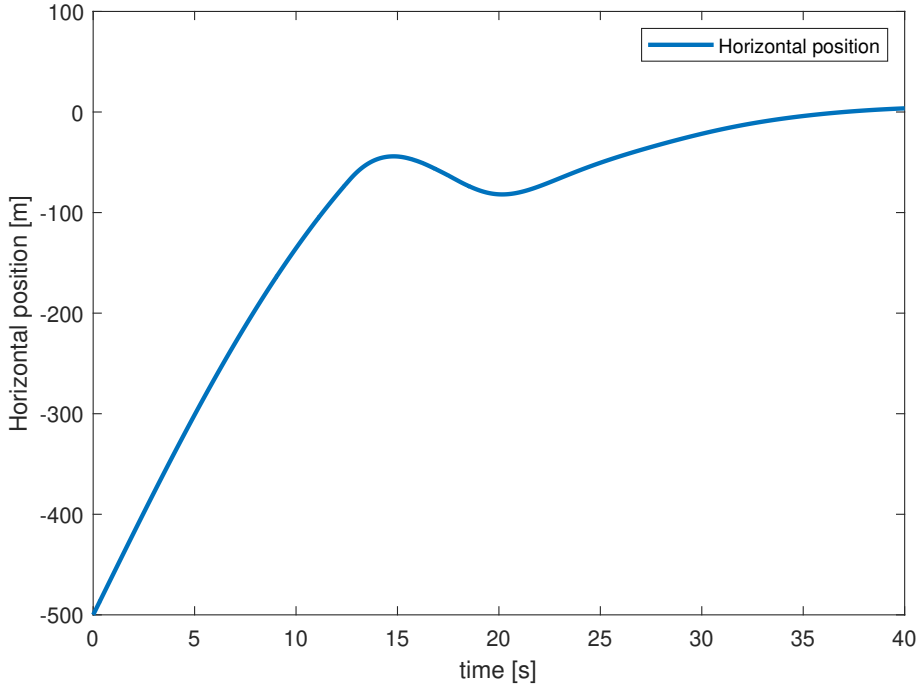

Figure 8.8: Position in y axis in time with the LQR in place

Chapter 9

Sub-scale experimental platform

During the consultations, an idea to build a real model in smaller scale came to discussion. Since I have already been interested in building a small rocket model before, I decided to push this idea forward. In this chapter, I will therefore shortly describe the process and progress of this related side project.

9.1 Model's components

The goal for the smaller model will be identical in terms of stabilization the vehicle during its descend. The first thing to consider is the propulsion system, for which I have considered two technologies.

First idea was to use a solid rocket motor used in consumer rocketry. However, for the purpose of simulations and testing the stabilization and control of the vehicle, motors available in the European market are not suitable for this project. Even though they may have thrust high enough, the burn time is only few seconds at maximum which is insufficient for testing and tuning the controllers.

Second option, and the one I eventually opted for, was to use an Electric Ducted Fan (EDF). The EDFs are not very commonly used in rocketry, however it is an ideal way of propulsion for the testing purposes of this project. Similarly to the situation with long burning rocket motors, it was unfortunately clear soon after the project started that to find a suitable EDF providing enough thrust is no easy task. This eventually led to ordering one from a low volume Czech manufacturer Vasamodel. The chosen EDF is the model Vasafan 90 with parameters listed in the table below.

Together with 6 cell Li-Po battery and motor, it can produce around 25 N of thrust. Every EDF motor usually needs an Electronic Speed Control (ESC) which controls that Rotations per Minutes (RPMs) are consistent and as required. I have chosen the Maytech FALCON 150A. The next phase of the project is to make the thrust steerable. There were two options considered at the beginning:

1. control flaps within the exhaust to steer the airflow

Figure 9.1: Vasafan 90 [29]

vnější průměr / outer diameter	91mm
vnitřní průměr / inner diameter	90mm
délka / length	65mm/150mm
hmotnost / weight	73g
maximální průměr motoru / maximal motor diameter	39.7mm
pro průměr hřídele motoru / for diameter motor shaft	5mm (8mm)
listy rotoru / rotor blade	7

Figure 9.2: Parameters of used EDF [29]

2. maneuverable nozzle - the same way like real TVC in rockets and air fighters

Advantage of the second option is that nozzle vectors the entire airflow while the flaps would effectively control only half of it. The nozzle will be further controlled by 3 or 4 JX PDI-HV0903MG digital servos with parameters show in table below.

The last considered components are another 4 small fans or EDFs at the top of the model to simulate the RCS system. At the current stage, these are not purchased and the focus is on finishing the rocket with only the TVC unit. Fins are also not used for this model since the velocity will be significantly lower. The body of the model will be made from 1 meter tall paper tube with diameter of 100 mm.

Every rocket needs to have its brain that is responsible for controlling all actuators based on

Rate	60 deg / 0.07 sec
Weight	9.4 g
Torque	2 - 2.6 kg/cm
Bandwidth	330 Hz

Table 9.1: Parameters of JX PDI-HV0903MG digital servo

current state of the vehicle. Because an extra Navio2 was available at my faculty department, it was an easy decision to use this board for controlling my rocket. Navio2 is not a standalone flight computer. Instead, it is a Hardware attached on top (HAT) for Raspberry Pi which brings it specifics. Mainly, Navio2 uses special distribution of Linux customized to be able to run in Real Time Operating System (RTOS) mode.

Significant advantage of the Navio2 board is its large array of ports and sensors. It contains two IMU units, each consisting of accelerometers, gyroscopes and magnetometers for complete orientation and motion sensing. GNSS receiver and high resolution barometer are on board as well. The board has also 14 PWM output channels for controlling motors and servos.

9.2 Software

9.2.1 ArduPilot

The Navio2 comes with preinstalled ArduPilot. That is very popular open source autopilot software system capable of controlling unmanned and autonomous vehicles such as drones, fixed-wing and VTOL aircrafts, helicopters, submarines, rovers and others. One of the supported modes is single-copter which is currently further investigated whether it can offer sufficient control and work well with the introduced design. If that will not be the case, there is an option to edit the source code and implement own code for the rocket attitude control.

9.2.2 ROS and Python

If default ArduPilot modes will show to be insufficient, there are other options for both assessing the sensor measurements and control the attitude. One is to use Robotic Operating System (ROS) to read measurement data and interact with the actuators.

The last option is to use pure Python scripts that will access sensor data via provided Python library. The setup is more straightforward and easy to set up. There can be, however, more effort involved afterwards in the data fusion and filtering process, which ArduPilot takes care of in at least some way.

9.3 Construction

At the time of writing the thesis, the assembly of the model is under way. To make sure the construction is optimal, it takes a lot of time spent in 3D modeling and later on, it will take a significant amount of time to test the vehicle on test stand before the flight. The most recent 3D model contains 3 batteries alongside the body and Navio2 inside the structure and the airframe with holes for enough airflow. Expected wiring is shown on figure below as well.

Unfortunately, the current situation regarding the CoVID-19 significantly pushed the realization back as the delivery of several parts from abroad is still ongoing. The work on this prototype will however continue even after this thesis is finished.

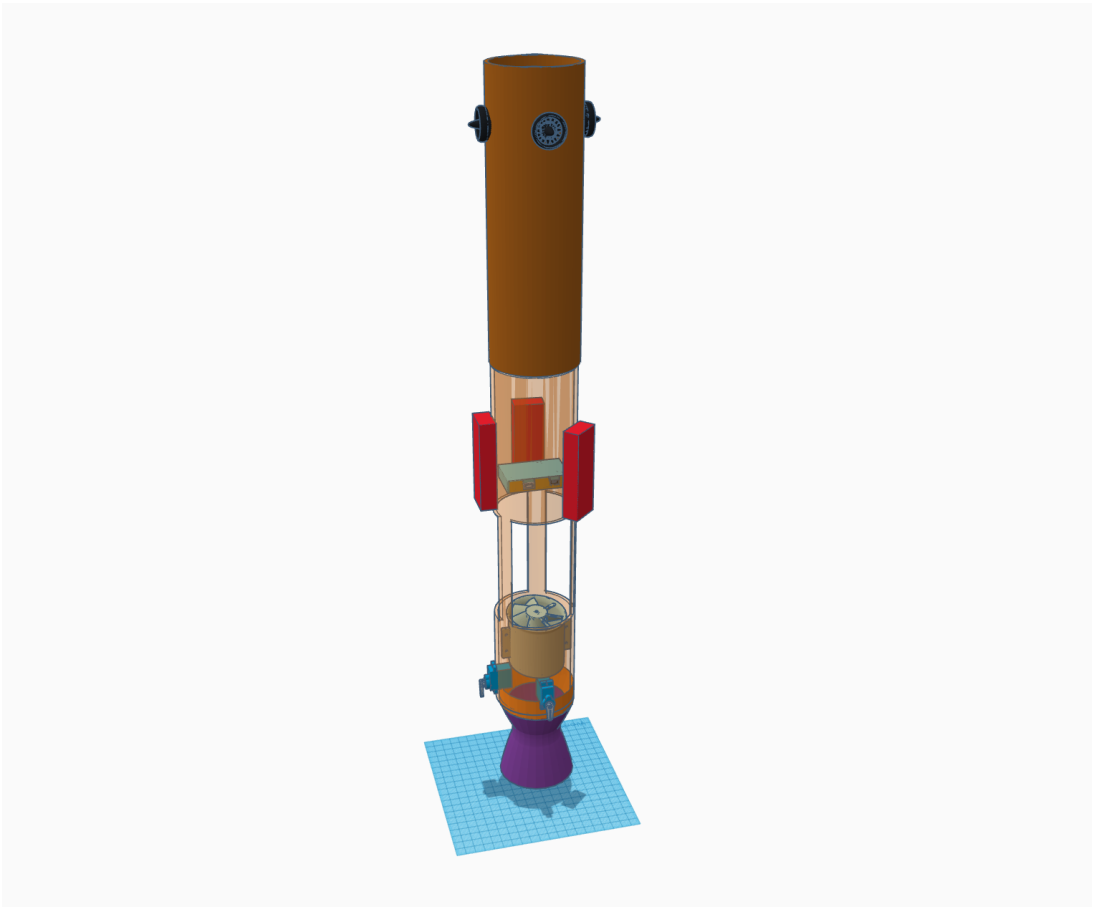

Figure 9.3: 3D prototype of upcoming small scale model

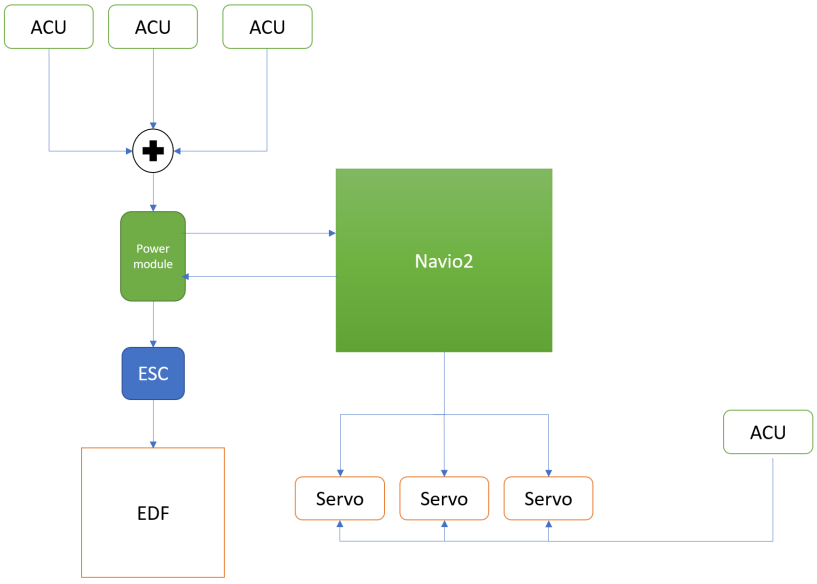

Figure 9.4: Expected wiring of the model's components

Chapter 10

Results

Before final conclusion of the thesis, I will use this chapter to present all results in one place. The order of the sections is related to tasks given in the chapter 2.

10.1 Constructing simulation mathematical model

After theoretical introduction, I constructed the model in SIMULINK environment. I considered important aerodynamic effects such as drag and lift (expressed as *axial* and *normal force in the body reference frame*. I took into account varying α and its effect on overall motion. Results can be seen in the **chapter 5, sections 5.1, 5.2 and 5.3**.

After completing the nonlinear model in 2D, I performed linearization in operating point, for which all states variables are equal to zero as the vehicle is considered to hover. Subsequently, I validated this linear model with respect to original nonlinear model and verified, that both models behaved accordingly. At the end, I elaborated sensitivity of the model to uncertainties in the physical parameters of the launcher [**subsection 5.4**].

10.2 Actuators and sensors recommendations

In the chapters 4 and 7, I introduced the most commonly used and popular types of actuators and sensors for such a task as a landing a launch vehicle. I gave specific recommendations about the actuators, specifically the TVC unit, control fins and the RCS. These judgments were based both on the simulations of developed model and on realistic assumptions that were inspired by excessive online research and intuition.

10.3 Control architectures

Following completion of the previous task, I moved to designing methodologies and architectures suitable for stabilization and control of the launcher during the landing phase. I focused on the attitude and altitude control as the GNC was not main focus of this thesis. I considered two popular methods, one as a representative of a classical method, second as a representative of

modern method. Steps and results of this part can be seen in the **chapter 6.** An overall design procedure for all actuators is explained in the **section 6.1.** Final SIMULINK scheme and analysis of implemented control is in the **section 6.3.**

The second used method is the LQR which represents the full state feedback control. Its implementation and results are presented in the **section 6.4.**

10.4 Design/parametrize the controller

Once the basic architectures of control systems were in place, I could tune the gains for each of used controller. For the PID control, tuning nested loops individually was necessary, starting with the innermost to cover the fastest dynamics. All steps and results are presented in **section 6.1.** In terms of the LQR control, tuning the matrices penalizing the states and control inputs proved to be a lengthy process but eventually, I arrived to successful parameters that accomplished the expectations from such a controller. [**Section 6.4**]

10.5 Validation of the performance

The previously designed controls laws together with actuators were put into test on more complex, 3D, model that was provided early as a result of different bachelor thesis. To simulate realistic conditions, I set the initial conditions to be equal to estimated conditions of the launcher at the moment of ignition during the landing burn. That means, the vehicle moves with velocity approximately 290 m/s under 20 - 30 degree deflection at altitude of 4900 metres above the landing pad.

Both classical and modern control methods successfully vertically stabilized the rocket and subsequently landed. Moreover, the LQR additionally successfully controlled also parameters with respect to GNC such as horizontal position and velocity. [**Chapter 8**]

10.6 Additional experimental sub-scale model

Besides all this, my supervisor suggested an idea of building smaller experimental model and attempt to reproduce the designed techniques on smaller scale. I was more than happy to comply and hence in my free time, I was working on sub-scale rocket model powered by the EDF. At the time of finishing the thesis, the model is under construction and initial software is being developed, however, the vehicle was not finished partially due to global pandemic lockdown. [**Chapter 9**]

Chapter 11

Conclusion

The goal of this thesis was the stabilization and control of vertically landing launcher. First, a simple 2D model of the descending LV was created. This model served as the main unit for control laws design as well as for designing the actuators and sensors placement and parameters which was another part of the task.

Initial assumptions regarding the actuators and sensors have been made early in the thesis based on personal knowledge and experience. Subsequently, arrays of different placement and parameters for each actuator based on online research have been created. This allowed to test many combinations of parameters, hence creating the optimal setup for this particular LV.

Changes were then made into the provided, more complex 3D nonlinear model, so it could use all actuators and sensors as it was intended. After the modification, this model served as a tool for validation of the designed control laws and actuators and sensors placement and setting of their parameters.

The final validation test of this thesis consisted of using average telemetry data from several SpaceX Falcon 9's flights at the moment of landing burn engine ignition.

It is shown that validation was successful as the LV in the 3D nonlinear model landed and behaved as expected during these tests. All given tasks were accomplished.

Bibliography

[1] T. S. Taylor, *Introduction to Rocket Science and Engineering*. CRC Press, Taylor & Francis Group, 2017, ISBN: 9781498772327.

[2] Hero's engine, [Online]. Available: https://en.wikipedia.org/wiki/Aeolipile#/media/File:Aeolipile_illustration.png (visited on 05/13/2020).

[3] T. Benson. (2014). Center of pressure, [Online]. Available: https://www.grc.nasa.gov/WWW/K-12/rocket/cp.html.

[4] T. Benson. (2014). Determining Center of Pressure, [Online]. Available: https://www.grc.nasa.gov/WWW/K-12/rocket/rktcp.html.

[5] "Dr. Wernher von Braun explains: Why Rockets Have Fins", *Popular Science*, vol. 185, no. 3, pp. 68–69 / 184–185, 1964, ISSN: 0161-7370.

[6] W. Haeussermann. (1970). Description and Performance of the Saturn Launch Vehicle's Navigation, Guidance, and Control System, [Online]. Available: https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19700023342.pdf.

[7] Erich Schülein, Daniel Guyot, "Novel High-Performance Grid Fins for Missile Control at High Speeds: Preliminary Numerical and Experimental Investigations", *Innovative Missile Systems*, 35–1 —35–28, 2006. [Online]. Available: {https://www.sto.nato.int/publications/STO%20Meeting%20Proceedings/RTO-MP-AVT-135/MP-AVT-135-35.pdf}.

[8] (2015), [Online]. Available: https://www.spacex.com/news/2015/08/31/grid-fins (visited on 2019).

[9] E. Musk. (2015). I am Elon Musk, CEO/CTO of a rocket company, AMA!, [Online]. Available: https://www.reddit.com/r/IAmA/comments/2rgsan/i_am_elon_musk_ceocto_of_a_rocket_company_ama/.

[10] SpaceX Telemetry Data, [Online]. Available: https://github.com/shahar603/Telemetry-Data (visited on 11/2019).

[11] Dragon – Cargo Version, [Online]. Available: http://spaceflight101.com/spacecraft/dragon/ (visited on 11/2019).

[12] (2019). Rocket Lab Announces Reusability Plans For Electron Rocket, [Online]. Available: https://www.rocketlabusa.com/news/updates/rocket-lab-announces-reusability-plans-for-electron-rocket/.

[13] A. Tewari, *Automatic Control of Atmospheric and Space Flight Vehicles*. 2011, ISBN: 978-0-8176-4863-3.

[14] W. Huenssermunn. (1970). DESCRIPTION AND PERFORMANCE OF THE SATURN LAUNCH VEHICLE'S NAVIGATION, GUIDANCE, AND CONTROL SYSTEM, [Online]. Available: http://www.ibiblio.org/apollo/Documents/tn_d-5869_1970023342.pdf.

[15] C. RNDr. Jan Zajíc. (2010). Momenty setrvačnosti geometricky pravidelných homogenních těles, [Online]. Available: http://mech.fd.cvut.cz/education/bachelor/18sat/download/zajic_momenty_setrvacnosti.pdf.

[16] M. Lustig, "Modeling of Launch Vehicle during the Lift-off Phase in Atmosphere", 2017.

[17] Lift & Drag vs. Normal & Axial Force, [Online]. Available: http://www.aerospaceweb.org/question/aerodynamics/q0194.shtml (visited on 03/12/2020).

[18] S. F. Hoerner, *Fluid-dynamic drag: Theoretical, experimental and statistical information.* 1965.

[19] Lift-coefficient curves for three airfoils with different aerodynamic behaviour?, [Online]. Available: http://www.boundvortex.com/images/flat_plate.png (visited on 04/15/2020).

[20] A. Tewari, *Modern Control Design with MATLAB and SIMULINK*. John Wiley & Sons, Inc., 2002, ISBN: 0-471-496790.

[21] Comparison of orbital rocket engines, [Online]. Available: https://en.wikipedia.org/wiki/Comparison_of_orbital_rocket_engines (visited on 05/2020).

[22] C. Gonzalez. (2015). What's the Difference Between Pneumatic, Hydraulic, and Electrical Actuators?, [Online]. Available: https://www.machinedesign.com/mechanical-motion-systems/linear-motion/article/21832047/whats-the-difference-between-pneumatic-hydraulic-and-electrical-actuators.

[23] I. Control Products. (2020). What is a Linear Actuator?, [Online]. Available: https://www.cpi-nj.com/what-is-a-linear-actuator/.

[24] (2020). Launch vehicle actuators, [Online]. Available: https://www.moog.com/products/actuators-servoactuators/space/launch-vehicles.html.

[25] NASA. Space Shuttle News Reference Manual, [Online]. Available: https://science.ksc.nasa.gov/shuttle/technology/sts-newsref/sts-mps.html#sts-mps-tvc (visited on 2020).

[26] NASA. (1966). Technical Information Summary Apollo-9: Apollo Saturn V Space vehicle, [Online]. Available: https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19700009516.pdf.

[27] MathWorks. Explore the NASA HL-20 Model, [Online]. Available: `https://www.mathworks.com/help/aeroblks/nasa-hl-20-lifting-body-airframe.html` (visited on 05/2020).

[28] NASA. Falcon 9 Launch Weather Criteria, [Online]. Available: `https://www.nasa.gov/pdf/649911main_051612_falcon9_weather_criteria.pdf` (visited on 05/21/2020).

[29] (2020). Vasafan 90 - vasamodel eshop, [Online]. Available: `http://www.vasamodel-eshop.cz/en/d/vasafan-90-1000027/`.