Study of a nozzle vector control for a low cost mini-launcher

Roberto Rodríguez

SUPERVISED BY

Joshua Tristancho

Universitat Politècnica de Catalunya
Master in Aerospace Science & Technology
July 2011
Study of a nozzle vector control
for a low cost mini-launcher

BY
Roberto Rodríguez

DIPLOMA THESIS FOR DEGREE
Master in Aerospace Science and Technology

AT
Universitat Politècnica de Catalunya

SUPERVISED BY:
Joshua Tristancho
Applied Physics department
ABSTRACT

This study is addressed to the control system of a mini-launcher. A reaction control based on purge pressure from a solid propellant engine of a main stage, or using cold gas from a storage tank will be the control method. Pulse modulator system will be the actuation method studied, and simulations about its suitability will be done with Simulink.

In this thesis, we try to check if this method is able to be implemented in a low cost way the Wiki-Launcher, a mini-launcher less than 100 kg. At the time we try to check if the control allowed by the reaction control can manage all the phases of the flight. Both possible control configurations are going to be analyzed.

Keywords: Nozzle Vector Control, Low cost mini-launcher, Solid propellant
Acknowledgements

I’m grateful to my parents and my friends for supporting me during this work.

I want thank in a special way to Joshua Tristancho for his continuous help and support during this research work and for give me the opportunity to participate within his team in this project.

I’m very grateful to all the WikiSat members who helped and supporting me with the realization of the thesis, specially Joshua Tristancho, Victor Kravchenco, Esteve Bardolet, Sonia Pérez, Lara Navarro and Raquel González.

I want to thank the EETAC school for the facilities and tools they have provided to support this work.

Also I want to thank our partners from Team Frednet and D47 for their support to our group in general, and some parts of my work in particular.
CONTENTS

Introduction ................................................................. 1

1. Launcher control and stabilization ................................. 5
   1.1. Aerodynamic Control ........................................... 5
   1.2. Gimbal ............................................................. 5
   1.3. Reaction Control ................................................. 6
   1.4. Vernier Thrusters ............................................... 7
   1.5. Thrust Vane ....................................................... 7
   1.6. System Selection ............................................... 8

2. System Configuration .................................................. 9

3. Control System .......................................................... 13
   3.1. System Selection ................................................ 13
   3.2. Launcher Dynamic Equation ................................... 14
   3.3. Simulations ....................................................... 17
   3.4. Optimization ..................................................... 22
   3.5. Software implementation ..................................... 26

4. System Design ........................................................... 29
   4.1. Valve System ..................................................... 29
   4.2. Propellant ........................................................ 32
      4.2.1. Propellant storage ......................................... 34
   4.3. Nozzles ............................................................ 36
   4.4. Actuators ........................................................ 38
   4.5. System Integration on the Launcher ......................... 39
      4.5.1. Second Stage Control with Purge Gases ............... 40
# LIST OF FIGURES

1. Scale of satellites as a function of mass ........................................ 2

1.1. The Saturn IB was equipped with fins for stabilization purposes\(^1\) .......... 5

1.2. Ariane MPS solid booster gimbal system\(^2\) ........................................ 6

1.3. Apollo reaction control thruster assembly\(^2\) ...................................... 6

1.4. First view of Soyuz vernier thrusters and main engines nozzles\(^3\) .......... 7

1.5. HQ-9 missile’s thrust vane system, combined with aerodynamic control\(^4\) .... 7

2.1. Control System ................................ ........................................ 9

2.2. System Working Examples .......................................................... 10

2.3. Launcher Layout ................................ ..................................... 10

2.4. Second Stage System Location ...................................................... 11

3.1. Pulse Parameters\(^7\) .............................................................. 14

3.2. Vanguard Launcher\(^5\) .............................................................. 15

3.3. Launcher Force Diagram\(^3\) ......................................................... 16

3.4. Longitudinal Root Locus .............................................................. 18

3.5. Individual Three Axes Control System and Responses ......................... 19

3.6. Single Axis PID Control System and Responses .................................. 20

3.7. Simulink Pulse Modulator Diagram ................................................. 21

3.8. Complete Three Axes Control System Diagram .................................. 22

3.9. Subsystem Simulink Diagram ....................................................... 23

3.10. Rotation Continuous Control Responses ......................................... 23

3.11. Longitudinal Continuous Control Responses ..................................... 24

3.12. Rotation Pulse Control Responses ................................................. 25

3.13. Longitudinal Pulse Control Responses ........................................... 25


3.15. Second Stage Longitudinal Response and Vibration Detail ..................... 27

4.1. Some of the Available Equipment .................................................. 29

4.2. Valve Assembly ................................ ...................................... 30

4.3. Valve Body CAD Model and Stress Simulation .................................... 30

4.4. Metal Tensile Strength - Melting Point ........................................... 31

4.5. Conical nozzle profile\(^{10}\) .......................................................... 37

4.6. Control System Nozzle .............................................................. 38

4.7. From left to right: Solenoid, Servo motor, DC Motor and Stepper Motor .... 39

4.8. First Prototype ................................ ....................................... 42

4.9. Second Prototype ................................ ..................................... 42

4.10. Third Prototype ................................ ..................................... 43

4.11. Actuation system Prototype ......................................................... 43

4.12. Picture and Results of the Second Prototype Low Pressure Pendulum Test . 42

4.13. Configuration of the High Pressure Pendulum Test ............................... 44

4.14. Train of Pulses Measured During Second Prototype’s High Pressure Tests . 46

4.15. Experiment Configuration and Frame Where the Gas Flow Through the Nozzle Is Slightly Visible .............................................................. 47
# LIST OF TABLES

1. Small satellites classification ........................................... 1

4.1. Specific Impulse ......................................................... 33
4.2. Cold Gases Critical Points ........................................... 33
4.3. Cold Gases Required Mass and Volume (200 bar) ................. 35
4.4. Second Prototype Test Results ....................................... 45

A.1. Specific Impulse (Chemical propellants) .......................... 57
A.2. Specific Impulse (Chemical propellants) .......................... 58
INTRODUCTION

Along their history, launch rockets have seen their size and payload capacity increased, following the tendency of developing larger satellites. Current geostationary or geosynchronous telecommunication satellites, capable of provide television, INTERNET and telephone services over larger regions are bigger than a car and weight several tons. But those large launchers also can put into orbit multiple satellites at different altitudes in the same launch, offering the possibility of share the launching costs between different customers. In most cases, cost and complexity of those launchers limit their availability and/or suitability. Furthermore they and the space industry in general, have been, and still are, under governmental or military control, that impose important restrictions due to strategic and security reasons.

Components for the space industry require space qualification. The space environment is very exigent and not all materials are suitable for this environment. The conditions of the space environment are very different from the Earth environment ones. Radiation makes unusable materials like plastics. Electronic components can have an unpredictable behavior and they must be shielded. Safety protocols like code error correction have to be implemented. Thermal gradients are wide, the temperature changes fast and beyond the usual ranges inside the atmosphere. Due to the high vacuum condition, out-gassing occurs in every material, turning them weaker every day are exposed to vacuum. Insulators play an important role in this sense, but in addition thermal flow works in a different way: radiators and heaters are part of the main thermal subsystems because mainly all the cases there is no atmosphere to dissipate the excess of heat.

On the other hand, during the last years there has been a progressive growing in the number of small satellites for applications other than telecommunications, specially on scientific or Earth’s observation missions. These satellites have taken advantage of improvements in technology, including COTS\textsuperscript{6} technology. That have made possible the production of smaller, lighter and cheaper sensors, computers and on-board instruments. With performances no far away from their bigger counterparts they are robust enough to match the very demanding space conditions. Hence the satellites become smaller, lighter, more energy efficient and finally cheaper, allowing research groups and universities, with smaller budgets, perform investigations and studies.

<table>
<thead>
<tr>
<th>Small Satellite classification</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Mini satellite</td>
<td>100 to 500 kg</td>
</tr>
<tr>
<td>Micro satellite</td>
<td>10 to 100 kg</td>
</tr>
<tr>
<td>Nano satellite</td>
<td>1 to 10 kg</td>
</tr>
<tr>
<td>Pico satellite</td>
<td>0.1 to 1 kg</td>
</tr>
<tr>
<td>Femto satellite</td>
<td>less than 100 g</td>
</tr>
</tbody>
</table>

The variety of those small satellites is so large, that a classification attending to their weight is done as shown in Figure 1. Femto satellites are the last group in appear and they make use of the newest and more advanced COTS elements available. Reducing the size and weight of every component and with a very careful and clever design, some requirements are relaxed, and some components disappear as the support structure. Femto and Pico satellites are currently intended as technology demonstrators and for academic purposes, rather than as scientific or commercial instruments. But in a near future when the technology will be mature, they will open a new and interesting field of applications for space platforms. However large rockets have to be used to put in orbit even those very small satellites due to the lack of smaller and more economical transports. Usually as secondary or piggyback payloads, they are included in the payload of a launch to use the excessive payload capacity of the rocket. They don’t have any power of decision on the flight parameters, as scheduler, and they have to wait months or years for the launch or adapt their mission parameters according to the characteristics of the available flights. Even in those cases, the lower launch cost achievable is about $50,000/kg, including the associated services, depending on the launcher selection [4][5].

In order to promote research and development in this brand of satellites and in alternative ways to put them into orbit, the N-Prize was launched on April 2008. The N-Prize is a challenge to launch an impossibly small satellite into orbit on a ludicrously small budget, for a pitifully small cash prize [1]. The N-Prize organizers offer two prizes of £9,999.99 to the first persons or groups to put into orbit around the Earth a satellite with a mass between 9.99 and 19.9 grams, and to prove that it has completed at least 9 orbits. The SSO\(^7\) prize is for those entrant who will use non-reusable launch systems, and the RV\(^8\) Prize will be awarded to the first entrant to complete the challenge using a partially or wholly reusable launch system. The overall cost of the launch, excluding ground facilities, development and working time, must fall within a budget of £999.99. The WikiSat team is one of the contestant for the N-Prize composed by scientist, engineers, and collaborators that are developing a complex engineering system like this in an Open Source approach. Many of these collaborators are teachers and students that develops together each of the

\(^7\)SSO. Single Spend to Orbit
\(^8\)RV. Reusable Vehicle
subsystems that compose the parts of the Wiki-Launcher, the Wiki-Satellite and the ground system or Moon2.0 simulator. Our work is mainly focused in the launcher part but control modeling is also used in the satellite part.

The Wiki-Launcher is the part of the system in charge of put the satellite in orbit. It is a two stages solid propellant based launcher. It is deployed in near space conditions from a balloon at 37 kilometers height. Aerodynamic stabilizers are useless because the weak atmosphere in there. Common vector control is often based in moving the main nozzle. Due to the small size of the rocket, it is too complex to use this technique in each stage. As our first concept we decided to have a single vector control system in the nose of the launcher. That way we reduce the weight, the complexity and the possibility of failure as well. This method is based on purge pressure from the second stage. During the first stage burning, compressed helium will provide control. During the trajectory a control is needed while the engines are working and before the second stages ignition, but the rest of the time no control is required.

If the concept is achievable is going to be tested during the following chapters. In chapter 1 we will present and discus different control system in order to show advantages and disadvantages of each method and to do a reasoned selection. Although there are many control systems, only those which due to complexity are suitable for our design are discussed. In chapter 2 we will describe the configuration of the system inside the launcher, its elements and their function. Some sketches made with CAD software will be used for this purpose. In chapter 3 we will present the control system design process, from the discussion of different alternative methods, to the optimization process. Results from simulation performed during the process will be shown. In chapter 4 we will carry out the detail design of every hardware component. Several prototypes and results of tests done with those prototypes will be shown and discussed, including, if any, launches.
Study of a nozzle vector control for a low cost mini-launcher
CHAPTER 1. LAUNCHER CONTROL AND STABILIZATION

Although the launch last only a few minutes, the launcher vehicle is exposed to aerodynamic and gravitational forces and many other internal and external disturbances, which affect its trajectory and may even derail the launch. For these reasons any launcher must incorporate a system to stabilize and maintain the correct attitude to follow the trajectory determined to insert into the correct orbit its payload. Different types of control and stabilization systems are available, and a combination of them is usually found in most commercial launcher, depending on the several considerations taking into account during their design phases. We are going to discuss the most important systems, and select which is suitable for our mini-launcher.

1.1. Aerodynamic Control

Forces must be generated somehow to control and stabilize a launcher. The aerodynamic control system produce those forces by means of aerodynamic surfaces placed along the body of the rocket, and actuated in the correct way. This system is extensively employed in missiles, because it is very effective and efficient in the lower atmosphere. However in the upper atmosphere and in the space vacuum they become useless and a source of drag and weight, being minimal their use in launcher, limited to first stage control.

Figure 1.1: The Saturn IB was equipped with fins for stabilization purposes

1.2. Gimbal

The movement of the nozzle only a few degrees can provide enough traverse thrust to control most launchers. This system is commonly used in solid rockets booster, adding a gimbal system to the nozzle and powerful actuators. With only one nozzle we can achieve lateral and longitudinal control. But with two nozzles also roll control can be performed

by the differential motion of the nozzle. This systems, which in terms of control is very
effective and simple, present a hardware limitation. The flexible join needed between the
nozzle and the rocket body has to be protected from the hot exhaust gases with flaps of
ablative material. This issue excessively complicates the building of a rocket at the scale
we are dealing with[2]. The Space Shuttle and Ariane 5 boosters are only two examples
of implementation of this technology.

![Diagram of Ariane MPS solid booster gimbal system]

Figure 1.2: Ariane MPS solid booster gimbal system[2]

1.3. Reaction Control

The reaction control system is based in small rocket engines strategically placed in the
launcher. When they are fired they produce the forces and moments needed to control
the launcher. Usually they are employed for roll control of first stages on big launcher,
placed them tangentially to the launcher skin, and for the whole control of the second or
third stages. The rocket engines used for control, not only has to be capable of shut-off
and re-ignite, but do that at the rate specified by the control system. Hence common solid
rocket are not suitable for this application. Hypergolic propellant are usually employed,
eliminating the ignition system and doing the control more reliable.

![Diagram of Apollo reaction control thruster assembly]

Figure 1.3: Apollo reaction control thruster assembly

http://www.flickr.com/photos/kazuhito/4494594353/
1.4. Vernier Thrusters

Vernier thrusters are small rocket engines, but bigger than those used in reaction control. They are placed with a fixed inclination around the main engines. When needed they are fired producing course changes. From a building point of view this system is simpler than the gimbaled nozzles, but add a lot of weight. Because of that vernier thruster are not very common in modern design, but it can be found in the Soyuz, currently in service.

![First view of Soyuz vernier thrusters and main engines nozzles](image)

Figure 1.4: First view of Soyuz vernier thrusters and main engines nozzles

1.5. Thrust Vane

This control method use vanes, which are finlike devices, to deflect the exhaust gases and produce the control forces. The vanes are placed inside the engine exhaust flow, hence they must be made of material which withstand the very high temperatures and velocities of the exhaust gases. Although is very simple to build this system reduce the engine efficiency. Practically only is found on designs done at the very beginning of the space exploration, like the V2 and Redstone launchers or at missiles.

![HQ-9 missile’s thrust vane system, combined with aerodynamic control](image)

Figure 1.5: HQ-9 missile’s thrust vane system, combined with aerodynamic control

---


1.6. System Selection

We already know the main characteristics, advantages and disadvantages of each system. We are able to do a justified selection of the more appropriate control system to our Wiki-Launcher. As was stated the launch will start from a balloon at 37 kilometers height, were the atmosphere is already very weak (its density is around 10% that we found at sea level). At that altitude the efficiency of aerodynamic surfaces is very low and it will be reduced as the rocket rise to higher altitudes, even at high Mach numbers. Vernier thrusters are a very heavy and complex solution for such a small launcher, even more for the second stage, which is estimated to burn for 3 seconds. Build a gimballed nozzle of that size is a real challenge. Any small failure in it will become in a source of problems, reducing the reliability of the launcher. Furthermore the availability and prize of the material needed for that system will difficult the construction of the launcher withing the allowed budget. The thrust vane could be a possibility, due to its simplicity and robustness. Cheap and widely available materials like steel could be used. Although it is too heavy for the second stage, it could be a good option for the first stage. But this would involve the necessity of a different control system for the second stage.

Because of that, we have determined as the best choice the implementation of a single reaction control system. Placed in the upper part of the launcher, this single system would be able to control the launcher during both stages. Common reaction system with hypergolic rocket result too complex, require additional propellants, storage tanks and plumbing, and also the risk associated with those chemicals. In order to simplify and reduce as much weight as possible, we have select a different approach based on a common pressure supply distributed to the nozzles through a valve system. The pressure will be purged from the second stage during its burning. And during the first stage burning, Helium or any other cold gas propellant may be the working flow. We expect reduce the overall weight of the system to a few grams, eliminating any kind of non-essential components, and by the way reduce it cost. And also reduce the complexity of the system to enhance its reliability, and the probability of success. As the burning of the second stage is so fast, we also consider the possibility of use spinning stabilization in case the control system would not be able to control this stage in a satisfactory way, or any other problem arise. In this method, before the stage ignition, a spinning motion is applied to the second stage to rigidify its longitudinal axis by gyroscopic effect and counteract any small thrust deviation from the engine.
CHAPTER 2. SYSTEM CONFIGURATION

The system layout and operation are going to be explained in the following lines. This is our first design and any change can be done attending the results obtained during the study.

As was stated in the introduction the Wiki-Launcher will be a two solid stages rocket. The second stage will have a smaller diameter, which will allow the payload, based on six Wiki-Satellites, be placed around the second stage. This results in a very compact configuration, which reduces the structural weight. Despite of put into orbit a Wiki-Satellite is enough for the N-Prize, a payload of six satellites is selected to exploit the maximum capacity of the launcher and increase the success possibilities.

![Figure 2.1: Control System](image)

The whole control system will be located over the second stage, in a plane perpendicular to the longitudinal axis, virtually at the rocket nose. This location offers several advantages. The purge of pressure is directly and easily performed through a collector installed on the top of the second stage engine. Furthermore it provides the largest moment's arm during all the flight, respect to the rocket's center of gravity. This reduces the maximum thrust needed to be produced by the system, and also its consumption.

In order to get a three axes control (longitudinal, lateral and roll) a common reaction control employ at least six actuators. But we have determined that four actuators with a proper distribution should be enough. Placing four nozzle with a relative inclination respect to the rocket's body axis, as shown in the picture, and firing them by couples we can achieve control over the three axes individually. Hence combining their effects, a simultaneous control of every axes is possible. Example of the control of each axis alone is found in figure2.2.

As a first approximation we have consider an angle of $45^\circ$ respect to the Z and Y axes, because it provides the same control over lateral and longitudinal dynamics, assuming the inertia moments are similar. The roll control is achieve displacing the nozzles axis from the longitudinal axis of the rocket, producing a torque. Firing two continuous nozzles the moments will be offset obtaining only a lateral force. But if two alternative nozzles are
fired the lateral forces produced by them will be canceled, and a momentum of twice the strength will be obtained.

To control the flow through the nozzles a group of four valves will be used. This is the more complex mechanical part of the system, as it is the only movable part. On its reliability depends the reliability of the whole system and by extension of the mission. Despite of the valves are very small and require a very precise construction, they are feasible to be produced even with amateur equipment. Also they have to withstand high temperatures, but for a very short time during the burning of the second stage. Electrical actuators will be used to open and close the valves. Those have to be as light as possible, have a low power consumption and be fast and powerful enough to move the valves. As a first choice common hobby servomotors are used due to their availability, reliability and low prize. Depending on the test’s results, faster, lighter or with lower power consumption actuators would be needed.

One femto-satellite will be the brain that will control the whole system. The Wiki-Satellite has everything needed to control the Wiki-Launcher during the flight. A microprocessor will run the control loop, process the sensor data and send the required signal to the actuator.
Furthermore it will control the ignition of the rocket stages. An IMU\(^1\) based on a three axes accelerometer and on a three axes gyroscope, will determine the rocket attitude within an inertial framework. Only a GPS\(^2\) module is added for tracking purposes. The use of a Wiki-Satellite eliminates the requirement of additional equipment, which would increase the weight and use a valuable space. If the workload is too high for the Wiki-Satellite capacity a different board can be used for the control system. The sensor data from the satellite should be send by an \(I^2C\) port.

\(^1\)IMU. Inertial Measurement Unit. http://en.wikipedia.org/wiki/Inertial_measurement_unit
Study of a nozzle vector control for a low cost mini-launcher
CHAPTER 3. CONTROL SYSTEM

Any launch vehicle, independently of its size, has a complex dynamical behavior, which depends on many variables. The hardware components described in the previous chapter require the implementation of a control system in order to obtain a proper control and stabilization of the launcher during its flight. Control engineers teams expend a lot of time to develop fine tuned control systems for large launchers. We have to focus in the extension and duration expected for the thesis, and we are also limited for the deadline of the N-Prize. But we must not forget that the objective of the N-Prize is put into orbit a femto-satellite. Hence any small deviation in the inclination, altitude, or in any other parameter of the initially chosen orbit is allowed, and the constraints of the control system can be relaxed. Further improvements in the control system, as well as in any other component of the launcher can be done.

3.1. System Selection

In launcher and space vehicles equipped with reaction control systems, we can find various types of on-off thrusters (hydrazine, cold-gas or pulse-plasma) which produce discontinuous control actions under switching constraints. The most commonly used approaches for thruster activation logic are direct bang-bang control and pulse modulation. Bang-bang control consists of the use of the nonlinear actuator as is. However, it presents difficulties to consider switching restrictions in the discontinuous optimal control setting when feedback control laws want to be applied[11]. Pulse modulators convert the continuous input command in a sequence of pulses which are the switching signals send to the on-off thrusters. The average value of the switching signal at modulator output is proportional to a constant modulator input. It follows a function chosen during the system design, which is highly dependent on the application. In control applications the quasi-linearisation of the switching actuator obtained with this method is useful because physical plants show low-pass behavior, attenuating the high frequencies introduced by the switching actuator. This allows the use of control method designed for systems with proportional actuators. However, quasi-linearisation fails at low switching frequencies[7].

In order to consider actuator switching restrictions some pulse modulator implementations are composed of a Schmitt trigger, a linear filter and a feedback loop, but it is complex and time-consuming hence no further discussion is going to be done. Another approach is the use of modulation curves, which are implementable in software or firmware. Their design process is easier than for the Schmitt trigger method[22][13]. We will use a pulse modulator per actuator based on curves explicitly consider switching restrictions. The switching restrictions are a minimum pulse duration and a minimum rest between successive pulses, which are dependent on the actuator characteristics and on the system behavior. The input of each pulse modulator will be a proper combination of the control commands of each axis.

The pulse modulator will allow us to use on-off thruster as proportional actuator, but as we will see the system's plants have an unstable behavior. Therefore a control technique within a feedback control loop must be used to obtain the desired control over the launcher.
It has to be designed taking into account the launcher dynamics, but also the actuator behavior and the thrust build up dynamics. Many different control techniques have been developed during the years such as PID\(^1\) or linear quadratic regulators. Each technique offers some advantages and disadvantages, and although dependent on the application, the final technique selection is a matter of the engineer’s choice.

Nowadays there is a trend toward using quaternion feedback in the attitude control of launchers as well as on satellites and spacecrafts. Quaternion control enables the attitude change along the shortest path by matching the control torque vector to the eigenaxis which is not possible with Euler angle, because they are based on the concept of sequential rotation. Although quaternion control offers that and other advantages, due to the low number of control actuation which are expected in our case, we are going to use a PID controller. The PID controller does not have the advantages of the quaternion control, but according with my background in this subject, it becomes an easier method to deal with and to implement in the software code. However quaternion control is not abandon. Due to its advantages it is considered for future improvements of the system.

### 3.2. Launcher Dynamic Equation

The dynamic equations of a body describe its movements and behavior as function of different parameters. These equations are the base of the plant used in the simulations. Find examples of modern launcher dynamic equations to start with is a hard task, and from a launcher which mets our constraints, a completely impossible mission. Hence as a starting point we are going to use the dynamic equations of the Vanguard rocket presented in [3]. The Vanguard rocket was intended as the first USA launcher, but due to some failures it was finally passed by the von Braun’s Juno rocket. Despite of their difference in size, the Vanguard is a suitable option, because it was completely cylindrical, with only one engine per stage, without any aerodynamic surface and the control was performed by means of gimbal nozzles. Even though gimbal is not our selected control method, the changes into the dynamic equations required to substitute this system for a reaction control are minimum. Adjustment of coefficients and other parameter to our model is also needed.

In order to simplify the equations, it has been assumed

1. The longitudinal axes of the rocket is the X axis, the Z axis is in the vertical plane pointing down and the origin at the center of gravity. The launcher is considered symmetric.

2. The mass and center of gravity of the missile is constant.\(^3\)

3. The rocket is a rigid body.\(^4\)

4. The launcher will follow a zero angle of attack trajectory.

5. The perturbations from equilibrium are small.

Equations 3.1 and 3.2 are the longitudinal dynamic equations of the Vanguard, describing its linear and rotational behavior respectively.

Vanguard rocket longitudinal equations

\begin{align}
\left(\frac{mU}{Sq} s - C_{z\alpha}\right) \alpha(s) + \left(-\frac{mU}{Sq} s - C_w \sin \Theta\right) \theta(s) &= C_{z\delta} \delta(s) \quad (3.1) \\
-C_{m\alpha} \alpha(s) + \left(\frac{I_y}{Sqd} s^2 - \frac{d}{2U} C_{m\gamma} s\right) \theta(s) &= C_{m\delta} \delta(s) \quad (3.2)
\end{align}

\(^2\)http://es.wikipedia.org/wiki/Archivo:Vanguard_rocket.jpg

\(^3\)Although the missile is consuming fuel at a terrific rate, if the instantaneous mass is used, the mass may be assumed constant during the period of analysis.

\(^4\)This is only a first approximation for the Vanguard rocket, but for the Wiki-Launcher due to its small size this assumption is closer to the reality
From these equations we can calculate the longitudinal transfer function of the Vanguard rocket presented in equation 3.3, where the input is the nozzle deflection (δ) in radians and the output the change on the rocket’s longitudinal attitude angle in radians (θ).

**Vanguard rocket transfer function**

\[
\theta(s) = \frac{C_{m_{\alpha}}C_{z\delta} + \left( \frac{mU}{Sq} (s) - C_{z\alpha} \right) C_{m_{\delta}}}{\left( \frac{mU}{Sq} (s) - C_{z\alpha} \right) \left( \frac{1}{Sq^2} S^2 - \frac{d}{2U} C_{mq} S \right) + \left( -\frac{mU}{Sq} s - C_{w} \sin \Theta \right) C_{m_{\alpha}}} \tag{3.3}
\]

This transfer function can be used for our launcher, using the appropriated parameters, but a change from the gimbal system to the reaction control is still needed. If we take a look to the dynamic equations, we can see that δ is multiplied by 3.4 and 3.5 providing the force and momentum control coefficients.

\[
C_{z\delta} = -\frac{T}{Sq} \tag{3.4}
\]
\[
C_{m\delta} = -\frac{TL}{Sq^2} \tag{3.5}
\]

Due to the small nozzle deflection angle, δ in radians is equivalent to sin δ, and multiplied by the engine thrust gives the lateral thrust component obtained deflecting the nozzle. The lateral thrust is the control parameter of the the reaction control, hence we substitute the deflection angle by the lateral thrust, eliminating the main engine thrust of 3.4 and 3.5. Also taking into account that now the lateral control force is applied in the upper part of the launcher and a positive force will produce a negative momentum, 3.4 and 3.5 become

\[
C_{zr} = \frac{1}{Sq} \tag{3.6}
\]
\[
C_{mr} = -\frac{1}{Sq^2} \tag{3.7}
\]
Therefore the longitudinal transfer function for the Wiki-Launcher is

**Wiki-Launcher longitudinal transfer function**

\[ \frac{\theta(s)}{T(s)} = \frac{C_{mq}C_{zr} + \left( \frac{mU}{Sq} - C_{za} \right) C_{mq}}{\left( \frac{mU}{Sq} - C_{za} \right) \left( \frac{I_z}{Sqd} - \frac{d}{2U} C_{mq} \right)} + \left( -\frac{mU}{Sq} - C_{w} \sin \Theta \right) C_{ma} \] (3.8)

In a similar way the lateral and rotational transfer functions for the Wiki-Launcher are determined.

**Wiki-Launcher lateral transfer function**

\[ \frac{\psi(s)}{T(s)} = \frac{C_{mp}C_{yr} - \left( \frac{mU}{Sq} - C_{yp} \right) C_{mr}}{\left( \frac{mU}{Sq} - C_{yp} \right) \left( \frac{I_z}{Sqd} - \frac{d}{2U} C_{mq} \right)} + \left( -\frac{mU}{Sq} - C_{w} \sin \Psi \right) C_{mb} \] (3.9)

**Wiki-Launcher rotational transfer function**

\[ \frac{\phi(s)}{T(s)} = \frac{C_{mr}}{\frac{I_x}{Sqd} s^2} \] (3.10)

### 3.3. Simulations

In order to design the control system we are going to use the MATLAB software and some of its powerful tools as SISOTOOL and Simulink.

The first step, to validate that the transfer functions calculated from the Vanguard dynamic equations are correct, is to built up in Simulink a control loop like that presented in [3]. The transfer functions are written in state space form in the MATLAB environment using for that purpose M-files shown in Appendix B.1. Then the Simulink diagram is drawn including a gain block, a zero-pole block for the compensator, a second order linear actuator block to implement the servo transfer function and a LTI system block for the system plant. A step source is use as input and a scope is used to visualize the system response. The input of the plant is the nozzle’s deflection and its output is the attitude angle, and as we have seen for longitudinal control a positive deflection produce a negative momentum. Therefore if we want a positive step as response, the feedback must be positive and the input negative to obtain a negative error. The same situation applies for the rotational control.

As was expected the response of the system is stable in all the axes. The overshot is important, and the settling time for the longitudinal and specially for the lateral control is large, but we must take into account that we are simulating a step of 1 radian (57.3°).

After it has been proven that the transfer functions are correct, we modify the M-files according with the process explained in the previous section to adapt them to our launcher. Also a change between imperial and metric units is performed. Apart from the plants, the actuators transfer functions are also changed. In the vanguard the servo transfer functions
modeled the behavior of hydraulic actuators, but in the Wiki-Launcher they have to model the thrust build-up dynamic, more than the servo behavior. As starting point we assumed the thrust build-up dynamic shown in equation 3.11 \[7\], neglecting the servo dynamics.

**Thrust build-up transfer function**

\[
H(s) = \frac{4800}{(s + 60)(s + 80)} \tag{3.11}
\]

Because of the plant and the actuator transfer functions have been changed, the compensator, including the gain, will be different. The new compensator are determined using the MATLAB SISOTOOL, in order to obtain stable responses. The response’s overshoot or settling time are not important, because they will be modified by the PID.

As an illustrative example we present the process followed for the longitudinal axis. First of all we open SISOTOOL for the Wiki-Launcher longitudinal axis M-file. Then we add a real pole and a real zero in the compensator, as well as the complex poles representing the servo transfer function. Furthermore the feedback is established as positive and the input of the system as negative changing the F gain to \(-1\).

As starting point the pole and the zero are those for the Vanguard. As can be seen it is not possible achieve a stable response for any gain, the zero and pole are moved until a better root locus plot is obtained. Then the gain is varied to achieve a stable response. The final result of the system can be seen in the image 3.4.

![Longitudinal Root Locus](image)

Figure 3.4: Longitudinal Root Locus

The compensator and gain obtained with SISOTOOL must be exported to the Simulink diagram. SISOTOOL has an option to draw a simulink diagram of the studied system, but all the data is stored in the Workarea and the blocks of the simulink diagram cannot be modified. Hence the compensator is exported from SISOTOOL to simulink by hand in the form of a zero-pole block and a gain block. The gain block is not essential because within the zero-pole we can chose the gain, but we used it for schematic purposes. At this point we have to say that the gain shown in SISOTOOL, is not the correct one to use in simulink.
The correct gain is obtained from the compensator block of the simulink diagram created by SISOTOOL, dividing the gain shown in that block by the gain of the actuator transfer function. Unfortunately I was not able to determine the reason of the offset between those gains. Following this procedure we obtain stable responses on the three axes, shown in figure 3.5.

![Simulink Diagram](image)

Figure 3.5: Individual Three Axes Control System and Responses

The next step is implement in each control loop the PID controller. A PID controller uses the error between the measured and the setpoint values of the controlled variable to produce the proper control input for the system plant. It is based on three parameters: a proportional dependent on the current error, an integral dependent on the accumulation of past errors and a derivative dependent on the current rate of change. Each parameter consist on a tunable gain multiplied by the current error, its integration or its derivation, respectively. The summation of all the parameters is the output of the PID controller 3.12.

\[
    u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t) \tag{3.12}
\]

As we are working in the \( s \) domain, the derivation in time domain is equivalent to multiply by \( s \), and the integration to divide by \( s \). In Simulink there is a block to do the integration (divide by \( s \)) that simplifies the process, unlike what happen with the derivation. However, as the setpoint is considered to be constant during the simulation time, the error derivation becomes the rate of change of the plant output 3.13.

\[
    \frac{de}{dt} \equiv \frac{\Delta e}{\Delta t} = \frac{e - e_0}{t - t_0} = \frac{(\theta - \theta_c) - (\theta_0 - \theta_c)}{t - t_0} = \frac{\theta - \theta_0}{t - t_0} = \frac{\Delta \theta}{\Delta t} \tag{3.13}
\]
This can be easily determined multiplying by $s$ the elements of the C matrix (moving them one position to the right inside the matrix) of the State-Space definition of the plants inside the M-files. Doing this the new output of the plant is the attitude angle rate of change, being the angle obtained integrating the output with the Simulink's integration block. Now we have two feedback loops, one with the rate of change and other with the angle. The error between the angle and the command is calculated. By one side it is multiplied by the proportional gain and by the other is integrated with the integration block and multiplied by the integrative gain. These two signal are added to the result of multiply the rate of change by the derivative gain. The sign of each of these signal depend on the axis. It can be easily understood looking at the figure 3.6.

![Figure 3.6: Single Axis PID Control System and Responses](image)

Tuning the gains the response of the system can be adjusted. In general an increment in the proportional gain produces a faster response, reduces the overshoot and the number of oscillations. Increase the integral gain value increases the response speed, the oscillation frequency and reduces the settling time. And a larger derivative gain reduces the speed of the response, but also the overshoot and the needed control force.

Until now all the attention has been focused over the response stability. But an equally important parameter is the maximum control force required to obtain acceptable responses. This is because the smaller the control force, the smaller the system size, complexity, weight and consequently cost. Also because the scale, techniques and materials we are dealing with impose important restrictions over the maximum thrust it will be able to produce as it is shown in the next chapter.

During all these process the pulse modulator has been included in the Simulink diagram. It has not been mentioned until now because it was by-passed, just to see its behavior, but without affect the actual loop. However some test were done including it inside the loop.

The pulse modulator convert the continuous signal from the PID controller to pulses of the same amplitude, adjusting their width and time between successive pulses in order to match the continuous signal. This is performed by an algorithm based on the one presented on [7]. It is build up inside a Simulink subsystem block. It includes an Embedded MATLAB function that has the algorithm code shown in Appendix B.4 and a four data store systems. Each data store system consists of a write block, a memory block and a read block. They are use to do the internal loops of the pulse modulator because they cannot be included in the code due to the way Simulink execute it. The variables managed in each loop are time of last switch on, time of last switch off, pulse duration and thrust. The Simulink diagram that represent the pulse modulator is in figure 3.7.
The last loop is needed to avoid a computational problem observed during the simulations. A single input value inexplicably goes out of range, producing an error in the pulse modulator that since then does not produce pulses, with the subsequent brake of the control action. Another issue seen during simulations is the indetermination produced when the command thrust is equal to or bigger than the maximum thrust produced by the actuator, where the pulse duration and time between pulses tend to infinity. This can be easily avoided by a saturation block that limits the maximum and minimum value of a signal, a bit under the maximum produced by the actuator. This saturation block is also used when the pulse modulator is by-passed to simulate the actuator restriction imposed over the continuous signal.

Once the three individual axes have been defined, they must be combined to simulate the actual control system of the launcher. First using continuous signals and then including the pulse modulators. As the command from the PID is in fact the thrust, the resultant from the vectorial addition of the command of each axis can be decomposed to know the required thrust for each thruster. If the required thrust for a thruster is negative, no thrust will be produced as a negative thrust is not possible. The combined effect of every thruster for each axis is determined and used as the plant’s input.

The configuration of the nozzles was selected with an angle of $45^\circ$, hence the sine is equal to the cosine. This simplifies the calculation of each thruster thrust to the addition of the longitudinal and lateral axes command thrust divided by $2 \cdot \cos 45^\circ$, and the rotational command thrust divided by 2, with the proper sign. The plants inputs are calculated by the addition of the final thrust produced by every thruster with the correct sign. And for the longitudinal and lateral axes the result must be multiplied by $\cos 45^\circ$. This process is included inside a subsystem block to clarify the Simulink diagram. The inputs of the block are the commanded thrust of each axis, and the outputs are the plants inputs. The Simulink diagram of this subsystem is shown in figure 3.9. During the first simulation the pulse modulators were by-passed to simplify the validation of the system. Once the system was validated they were included in the control loop to start with the optimization process.
3.4. Optimization

During the optimization process the gains of the systems and the pulse modulator characteristics must be adjusted to obtain the best response. As all the control axes are joined, a change on any gain will affect the three responses. Due to the launcher’s flight pathway not all the axes will require the same control actuation. The launcher will abandon the launch ramp approximately vertical. During the first seconds of flight the rotational control should orient as quickly as possible the launcher axes properly before incline it. This will demand large control from the rotational control, but smaller from the lateral and longitudinal, just to maintain the vertical. Once the launcher is properly oriented, it must be inclined from the vertical to its final attitude to obtain the desired flightpath. This very fast attitude change of up to $60^\circ$, according with simulations done with Moon 2.0, greatly demands the longitudinal control. The role of the rotational and lateral control become secondary just to compensate very small disturbances. After the first stage’s burn, the launcher has to be oriented before the second stage’s is ignited, to ensure it is correctly pointing to obtain a circular orbit. The second stage is intended to be controlled purging gases from it. As was already mentioned in chapter 1 a second method is considered due to the short control time and the inconvenient that are imposed, like carry the control system within the second stage or higher constraints for the materials. This consist in rotate the second stage before start it, to rigidify the longitudinal axis and maintain the stage stable. Cold gas remaining in the tank is used for that. The controls system stay with the first stage, and the system constrains are relaxed as it does not have to deal with hot gases.

First we start with the complete control system with the pulse modulators by-passed, simulating a simultaneous rotational change of $180^\circ$ and a lateral and longitudinal change of $2^\circ$. This represents the first part of the flight at an altitude of around $37\, km$ and an average speed of $319.23\, m/s$. When something is moving at speeds close to or higher than the speed of sound shock waves are produces. Under the indicated conditions a shock wave is formed and the dynamic pressure after it is $350.5\, Pa$. The maximum weight of the launcher
of $3.77 \, kg$ is used for this and the next flight segment. The saturation for each actuator set at $1 \, N$ as maximum, which represent a thrust from each actuator of $1.41 \, N$. The optimal response is obtained for the following gains: Longitudinal(200,90,90) Lateral(200,90,55) Rotational(50,30,25). The system and its responses are shown in figure 3.10.

For the next part of the atmospheric powered flight the conditions were selected for the simulation as a speed of $1,000 \, m/s$ and a dynamic pressure of $1,100 \, Pa$, after the shock wave. Those are the conditions find at $43 \, km$ height after 12 seconds of powered flight. The maximum actuator thrust is the same as in the previous simulation, because it cannot be changed on flight. Different configurations were tested, but one of the best responses...
was for the same gains as for the other case. The system and it responses can be seen on figure 3.11.

Since for proportional control we have obtained a reasonable good response, the pulse modulators are included in the loop. The same conditions explained before are maintained for each of the two simulated cases. After many simulations the second order linear actuator parameters appear as the key point for the stability of the system. A high natural frequency is required from the system to be able to produce the correct control action over the launcher. For the results shown below, the natural frequency of the system was set to 1,000 Hz. However later simulations were performed with natural frequencies down to 500 Hz with stable responses. After adjust the system gains good responses were obtained for both cases, as with continuous control, maintaining the same gains: Longitudinal(200,80,90) Lateral(200,90,55) Rotational(50,30,30) Both systems and responses can be seen in the following figures 3.12 and 3.13.

Another simulation in vacuum conditions was done, with a dynamic pressure of 0.0001 Pa and a velocity of 2,666 m/s. The weight of the system was reduced to 854 grams, which is the launchers weight after the first stage burn out. The same configuration as in previous simulations was maintained for a longitudinal turn of 30° and a lateral and rotational turns of 2°. The same system configuration parameters are maintained with very good results. This try to simulate the pointing process before the second stage ignition. The results are presented in figure 3.14.
Figure 3.12: Rotation Pulse Control Responses

Figure 3.13: Longitudinal Pulse Control Responses
Also with this conditions a very large rotational turn (1,000°) with no longitudinal or lateral movement is performed. This is the simulation of the spinning stabilization of the launcher to stiffen its longitudinal axis, in case no control of the second stage will be incorporated.

The last simulation that must be done is for the control of the second stage during its acceleration. In the same vacuum conditions, the weight is set at 526.23 grams, the total length of the second stage is 140 mm and the control arm is 60 mm. Maintaining the same configurations as for the other flight paths, a stable response is obtained. A high frequency longitudinal vibration is obtained as during other phases of the flight, but here with a huge amplitude, up to 0.5°. Playing with the gains the vibration can be moved to the lateral axis, but never disappear. This is because the small control arm and the high control force for this condition. The result for the configuration that we have maintained through all the simulations can be seen in figure 3.15, including a detailed view of the huge vibration.

3.5. Software implementation

Once the feasibility of the system has been proven and a proper control has been obtained from the simulations, the control system has to be implemented in the launcher. That implementation is done in the form of a software, which is upload to the launcher control board, in our case the Wiki-Satellite. The software is done within the Arduino’s environment, in which is based the Wiki-Satellite. This software will integrate the control loops, the pulse modulators and the appropriated interface to send the pulses to the actuators.

The main differences between the simulations and the final implementation of the control loop are that actual behavior of the launcher is represented by the plants on the simulations, and the feedbacks are the gyroscopes and accelerometer measurements. The
software inputs are the IMU measurements in the form of angles, and the outputs the command signals for each actuator. The PID controller is implemented in the code by means of the PID_Beta6 library available from the Arduino Playground designed by Brett Beauregard. To simplify and optimize the code, we create another library for the pulse modulator.

The main code for the attitude control shown in Appendix C.1 only uses the two libraries described before. First it initializes two groups of variables, one for the PID and the other for pulse modulator. The PID variables are three controllers, one per axis, with the gains, Input, Output and Setpoint of each PID. The pulse modulator variables are already defined. They are the pin where the actuator is connected (pin), minimum pulse duration (ton), minimum rest between pulses (toff), maximum thrust (U), and the saturation limits (Tmin, Tmax), for each modulator.

Within the void setup() are defined the initial setpoints, the PID's input and output limits and their sample time. Finally the three PID's are started. In the void loop() the first step is check if a change in the gains of the system is necessary according with the system conditions. If the real model follows the simulations, no gains change is necessary and this step can be omitted. After the gyroscopes measurements are read and used as input for the PID's, which are then computed. The PID's outputs are combined to obtain the signals used as input for the pulse modulators, and the pulse modulators are then run, sending the open-close signal to the electrical actuators.

When the pulse modulator is called in the main code the code shown in Appendix C.2 is executed. This code is a modification of the Matlab code used in the simulations, which is based on the algorithm presented in [7]. At the beginning the variables defined in the main code for the pulse modulator are initialized. The thrust that determine the limit for the calculation of the pulse properties (fmin) is determined as a function of the maximum thrust (U), minimum pulse duration (ton) and minimum rest between pulses (toff). The pulse width is initialized equal to zero. After that the input of the system from the PID's combination is limited to ensure it is within limits. The upper limit is used to avoid indetermination that take place when the input is equal or bigger to the maximum thrust. The lower limit is employed.
to avoid the use of the control system for very small corrections and then save fuel.

The first time the code is executed it calculates an initial pulse width ($\gamma_k$) and a time between start of pulses ($T_k$). With this provisional characteristics two variables are defined. The time when the last pulse started ($t_{son}$) and the time when the last pulse finished ($t_{soff}$). Also the output is set to low (closed) as we are using a digital signal to control the actuator. During the next iterations, the code checks if at that time it is still within a determined pulse. If in the iteration it is inside the pulse, a new pulse width ($\gamma_k$) is calculated, and then it is checked again if it is still within the new pulse. If it is inside the output is high (open). If it is outside, that time is defined as $t_{soff}$ and the output changed to low. If in the iteration it is outside a pulse, a new $T_k$ is calculated, and as before checked again if with the new value it is still out of a pulse. If it remains out the output is keep low. And if now it is within a pulse, that time is defined as $t_{son}$ and the output is changed to high.
CHAPTER 4. SYSTEM DESIGN

In chapter two we have seen an overview of the hardware part of the system. Now with the data obtained from simulations we are going to discuss every main component in detail. Each component will be subject to different constraints as working temperatures for the mechanical parts, current consumption for the electrical actuators or actuating frequency of the valves and actuators. Those and other constraints are taking into account during the design and building processes. Furthermore reduced weight and cost are inherent constrains of the project.

![Figure 4.1: Some of the Available Equipment](image)

4.1. Valve System

A valve is a device which controls the flow of a fluid. It is a basic element of any fluid circuit. At the market there are so many different types of valves as application, types of fluids and operating conditions. Unfortunately there is not any valve that match our requirements. Hence we must build them. We need very tiny valves, as light as possible, capable of withstand gas flow temperatures on the order of $800\,^\circ C$ for at least 5 seconds without loosing their working capacity, as well as pressures up to 30 bar during the whole launch. As they are going to be manufactured by ourselves with a limited equipment, their design must be as simple as possible and take into account the techniques available for their construction. In that sense, the use of any components already available in the market which can be modified or adapted to our system is a very interesting option. Valves and actuators are closely interrelated and this relationship must be taken into account in their selection and design. The maximum force required from the actuator and the time it must be working at high demanding rates depend on the valve’s design.

After weighing different approaches we have chosen as the first option a design based on a ball valve. In this design a ball is pushed against an orifice to stop the flow. If the action over the ball is released, the pressure of the flow push back the ball and open the circuit. In order to reduce the force needed to be produced by the actuator to hold the ball against
the orifice a screw is used as intermediate part, in such a way that the actuator only has to apply a smaller force to screw or unscrew the bolt, and a negligible force to maintain its position. This allows the use of smaller, lighter and less powerful actuator and a reduction in the electrical power consumption.

Figure 4.2: Valve Assembly

To select appropriated materials all the constraints must be taken into account. First of all the materials for the body of the valve must be machinable, hence ceramics and glasses are discarded because they are almost impossible to machining. Due to the high temperatures they will be exposed also polymers are discarded. The price and density are parameters that must be maintained as low as possible. But also we must consider the availability of the materials, which at the end has an impact in the price. Furthermore as the size of the system is so small an increment in the density of the materials is not very important if an improvement in other parameters is achieve.

Because of they have to withstand pressures up to 30 bar even at high temperatures they must present a tensile strength bigger than $12 \text{ MPa}$, according with the SolidWorks simulation done for the weakest piece of the valve subject to a pressure of 30 bar seen on figure 4.3

Figure 4.3: Valve Body CAD Model and Stress Simulation
Following our decision of use available components, the valve is based in electrical strip connectors. Those component are found on any electrical store by a ridiculous price. They are made of brass and are available in multitude of sizes. Common brass alloys present tensile strengths greater than $200\,MPa$, and melting points over $800\,^\circ C$. Brass appears to be a good choice as material for our system.

The electrical strips present a geometry that enable the production of two valves in a very light single module, having a common input from the collector. Inside channels are made milling and drilling a solid bar of brass that fits inside the electrical strip connector. The use of brass in both components eliminates the problem of galvanic couple. Furthermore brass is a great material for our purposes because it presents very good machining properties. This combined with the modulated design, greatly simplifies the production of the valves using our limited equipment, even within relatively small tolerances. The screws which come with the electrical strip connectors are substituted by smaller steel screws. Those screws will push or release a 2 millimeter steel bearing ball against the orifice build into the brass’s bar. As a hobby servomotor was the first option as actuator, a small metallic platen is welded on top of the screw working as actuating arm to screw or unscrew the bolt. With this system the turn of the bolt is limited to $120^\circ$ more or less. With a screw pitch of 0.5 millimeters it produces a linear displacement of about 0.16 millimeters, but a total open area of around $0.7\,mm^2$. Those are the dimensions of the prototypes, but the actual system may be bigger or smaller, which is not a big issue because we can find those standard component in a multitude of sizes.

For the joining of the electrical strip connectors with the solid bar of brass and with the collector, also made of brass, brazing is used. This process consists of join different metallic component using another metal or alloy with lower melting point. When it is melted it fills up the space between the components by capillary action, joining them when it is cooled down. We must be careful to not use too much brazing alloy to avoid obstruct the internal...
channels. For the prototypes for tests with compress air and helium, we have use solder-ing, which is the same process but using alloys with melting points lower than 450$^\circ$C, like Sn – Pb alloys.

### 4.2. Propellant

The selection of the propellants are going to be used for an application depends on many parameters, one of the most important is the specific impulse because it has an strong influence on the system efficiency. The specific impulse for a rocket engine is defined as the thrust per unit of propellant weight flow. This means that a rocket with a higher specific impulse will produce the same thrust consuming lower quantities of fuel, or for a given amount of propellants the engine will produce more thrust or the same during a longer period of time. For a given mission the use of rockets engines with high specific impulse reduces the weight of propellant, which in addition reduces more the required amount of propellant. To obtain the greatest specific impulse the equivalent velocity ($C$) must be as large as possible according with equation 4.1

$$I_{SP} = \frac{T}{gm} = \frac{\dot{m}C}{gm} = \frac{C}{g} \tag{4.1}$$

Where:
- $I_{SP}$, s Specific impulse
- $T$, N Thrust
- $g$, m/s$^2$ Gravity at Earth’s surface
- $\dot{m}$, kg/s Propellants mass flow
- $C$, m/s Equivalent velocity

The equivalent velocity can be calculated for chemical or nuclear engines by the equation 4.2.

$$C = V_e + \frac{(P_e - P_0)A_e}{\dot{m}} = \sqrt{\frac{2\gamma R T_0}{\gamma - 1}} \left[ 1 - \left( \frac{P_e}{P} \right)^{\frac{\gamma - 1}{\gamma}} \right] + \frac{(P_e - P_0)A_e}{\dot{m}} \tag{4.2}$$

Where:
- $V_e$, m/s Exhaust velocity
- $P_e$, Pa Exhaust pressure
- $P_0$, Pa Outside pressure (atmospheric)
- $P$, Pa Combustion pressure
- $A_e$, m$^2$ Exhaust area
- $\gamma$ Exhaust gas adiabatic coefficient
- $T_0$, K Adiabatic flame temperature
- $R = r/M$ Specific gas constant
- $r = 8314.5 J/(kmol K)$ Universal gas constant
- $M$ Exhaust gas molecular weight
From the equation 4.2, the equivalent velocity depends on the chemical and physical properties of the exhaust gases such as molecular weight or temperature, and also the expansion rate which is dependent on the nozzles design. The maximum specific impulse is more dependent on the propellant, which usually are classified according with the specific impulse they can provide. The specific impulse of chemical propellant can be found in Appendix A, and for cold gases and other propulsion system on table 4.1.

### Table 4.1: Specific Impulse

<table>
<thead>
<tr>
<th>Propellant</th>
<th>Specific Impulse (s)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cold Gas[8]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Krypton</td>
<td>50</td>
<td>In vacuum with nozzle</td>
</tr>
<tr>
<td>Argon</td>
<td>57</td>
<td>area ratio of 50 : 1 and</td>
</tr>
<tr>
<td>Air</td>
<td>74</td>
<td>initial temperature of 20°C</td>
</tr>
<tr>
<td>Nitrogen</td>
<td>76</td>
<td></td>
</tr>
<tr>
<td>Methane</td>
<td>114</td>
<td></td>
</tr>
<tr>
<td>Helium</td>
<td>179</td>
<td></td>
</tr>
<tr>
<td>Hydrogen</td>
<td>284</td>
<td></td>
</tr>
<tr>
<td>Others</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nuclear</td>
<td>850</td>
<td>Using H$_2$ as working fluid</td>
</tr>
<tr>
<td>Ion thruster</td>
<td>3,000 to 21,400</td>
<td>Depending on the technology</td>
</tr>
<tr>
<td>VASIMR</td>
<td>30,000</td>
<td>Variable Specific Impulse Magnetoplasma Rocket</td>
</tr>
</tbody>
</table>

### Table 4.2: Cold Gases Critical Points

<table>
<thead>
<tr>
<th>Propellant</th>
<th>Critical Temperature (K)</th>
<th>Critical Pressure (kPa)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Krypton</td>
<td>209</td>
<td>5,500</td>
</tr>
<tr>
<td>Argon</td>
<td>151</td>
<td>4,870</td>
</tr>
<tr>
<td>Air</td>
<td>78.67</td>
<td>3,770</td>
</tr>
<tr>
<td>Nitrogen</td>
<td>126</td>
<td>3,390</td>
</tr>
<tr>
<td>Methane</td>
<td>190.6</td>
<td>4,604</td>
</tr>
<tr>
<td>Helium</td>
<td>5.24</td>
<td>230</td>
</tr>
<tr>
<td>Hydrogen</td>
<td>33.23</td>
<td>1,300</td>
</tr>
</tbody>
</table>

Although select propellants or technologies that provide the highest specific impulse looks like the obvious choice, they usually present some limitations or disadvantages that limit their use. Currently ion thrusters provide very small thrust and nuclear engines have shielding and cooling problems, beside the risk radiative materials present in case of accident. Also chemical propellants both liquid and solid, as fluorine, ammonia or perfluoro - type, are extremely hazardous, and because of that very difficult to handle. The common propellants that offer the higher specific impulse are LO$_2$ - Hydrogen.

Any way for our control system the propellant selection is limited to cold gas type for control during the first stage burning, since the propellants during the second stage burning are its propellants themselves, which are selected attending to the second stage's requirement. Looking at the different options of cold gas we have in table 4.1, first of all we must take into account that those values are in the vacuum, using a nozzle with an area ratio of 50 : 1 and
the initial temperature of 20°C. We are going to use the cold gas to control the launcher inside the atmosphere, then the area ratio will be smaller, and at the end their size would be limited by the available space. In order to take this into account the specific impulse presented in the table 4.1 will be reduced by a 10% in the calculations.

4.2.1. Propellant storage

Another important issue in the selection of the propellant is the required storage capacity and conditions. Our first idea was to storage the cold gas inside the second stage's channels, in such a way that when it is separated from the first stage, the remaining gas would be vented out through the nozzle. But this configuration is only feasible, if the required mass of propellant can be stored in the second stage's free space.

A very rough calculation of the required Helium mass can be done from equation 4.1, calculating the mass flow required from each actuator to produce the maximum thrust determined before, and supposing two of them are continuously actuated during the control period (30 seconds).

\[
\dot{m} = \frac{T}{I_{SP} g} = \frac{9.81 N}{161.1 s \cdot 9.81 m/s^2} = 0.00621 kg/s
\]  

(4.3)

\[
m = 2 \dot{m} t = 2 \cdot 0.00621 kg/s \cdot 30 s = 0.37244 kg
\]  

(4.4)

This mass of helium would take at standard conditions (\(\rho_{He} = 0.1786 kg/s\)) a volume of 2.085 m³, which exceeds several times the overall volume of the whole launcher. When a common gas is compressed and/or its temperature is reduced, its density increases, reducing the required volume to storage a determined mass. If the temperature is low enough and the pressure high enough a change of state happens and the gas becomes liquid. Gases become more difficult to liquefy as the temperature increases because the kinetic energies of the particles that make up the gas also increase. For every substance there is a temperature above which vapor of the substance cannot be liquefied, no matter how much pressure is applied, this is known as the critical temperature. The pressure required at that temperature to liquefy the substance is the critical pressure.

A very usual way to storage propellants which appear as gases at standard conditions, is in cryogenic state, reducing their temperature to about 20K (−253°C). This reduces the pressure required to store them in liquid state (if the temperature is below the critical temperature), and hence the tank stresses. This method require very powerful refrigerator stations that cool down the propellant to that low temperature before load it into the launcher tanks, a few minutes before the launch. Due to the massive amount of propellant big launchers carry on inside their tanks, a lot of heat is needed to increase their temperature. Hence, although ambient heat is warming up the propellants since they left the refrigerator stations, they can remain at those very low temperatures for a time long enough for the launch. However an small propellant mass as in our case will be heated up very fast. This disable the use of this storage method for our application. Even more taking into account that the balloon ascension to 37 kilometers takes around two hours.
The pressure limit inside the second stage is 30 bar. For this pressure at room temperature (20°C) the helium density ($\rho_{He}$) is 4.92255 kg/m³. A volume of 0.07566 m³ (75.66 liters) should be required. As a result our first idea of storage cold gas inside the second stage must be discarded. If helium is stored as a liquid ($\rho_{LHe} = 69.93$ kg/s) this volume will be reduced to 0.005326 m³ (5.326 liters). However helium’s critical temperature is 5.24 K, a value completely out of our possibilities. Taking a look at table 4.2 we can see that the critical temperature for all the cold gases commonly used as propellants is too low, hence their storage as liquid is out of our possibilities.

Generally any of those gases is provided in metallic bottles at pressures around 200 bar. Common manometers reduce this pressure to less than 30 bar that is enough for most applications. Direct feeding from the bottle is possible, hence 200 bar will be our maximum storage pressure. But at this pressure the volume occupied by those gases is going to be too large. Up to now the required control force has been calculated to be able to control the launcher with a deviation of 1° of the nozzles axes. This has been done supposing the worst conditions during the whole burning period, plus an additional time needed to point the launcher before the second stage’s ignition. This is a very extreme case advised by the INTA¹ to be used as safety factor in the calculations. As this condition cannot be satisfied due to the impossibility of storage the required amount of propellant, we are going to relax it. The new control force will be the control force calculated in the simulations, including a safety factor of 20%, which results in 1.7 N. Following the calculations as before, the mass and volume required of each type of cold gas to accomplish with the new conditions are collected on table 4.3.

<table>
<thead>
<tr>
<th>Propellant</th>
<th>Mass Flow (g/s)</th>
<th>R (J/kg K)</th>
<th>Density (kg/m³)</th>
<th>Mass (kg)</th>
<th>Volume (l)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Krypton</td>
<td>3.446</td>
<td>99</td>
<td>637.401</td>
<td>0.208</td>
<td>0.309</td>
</tr>
<tr>
<td>Argon</td>
<td>3.040</td>
<td>208</td>
<td>320.5128</td>
<td>0.1824</td>
<td>0.569</td>
</tr>
<tr>
<td>Air</td>
<td>2.342</td>
<td>287</td>
<td>232.2880</td>
<td>0.1405</td>
<td>0.605</td>
</tr>
<tr>
<td>Nitrogen</td>
<td>2.280</td>
<td>297</td>
<td>224.4669</td>
<td>0.1368</td>
<td>0.609</td>
</tr>
<tr>
<td>Methane</td>
<td>1.520</td>
<td>518</td>
<td>128.7001</td>
<td>0.0912</td>
<td>0.709</td>
</tr>
<tr>
<td>Helium</td>
<td>0.968</td>
<td>2080</td>
<td>32.0513</td>
<td>0.0581</td>
<td>1.812</td>
</tr>
<tr>
<td>Hydrogen</td>
<td>0.610</td>
<td>4120</td>
<td>16.1812</td>
<td>0.0366</td>
<td>2.263</td>
</tr>
</tbody>
</table>

From those results we see that for the lightest gases the required mass is several times smaller than for the heaviest gases. But the required volume works in the opposite way and the volume required to store the smaller mass of the lighter gases is bigger than that required for a larger mass of the heaviest gases. A big tank also increase the weight, even more at those high pressures, reducing the benefit of the small mass of the gases, and increase the difficulties to locate the tanks inside the launcher. Even for the smallest required volume it is similar to the second stage volume, which gives an idea of the influence that it would have in the launcher’s design. The selected cold gas should provide a compromise between the mass and the required volume. Nitrogen requires a lower mass and twice the volume than Krypton, and has the advantage that we already have a bottle of nitrogen, and no further expenses are needed. The volume and weight are more realistic

¹INTA. Instituto Nacional de Técnica Aeroespacial. http://www.inta.es/
than those calculated before with a 1° deviation in the nozzle. However the volume is still very large and another method as a gas generator could be studied to reduce the volume problems.

4.3. Nozzles

Nozzles are ducts whose cross section area varies, and when a flow passes through them its thermodynamic properties are modified in an adiabatic way. This device is use in multitude of application, usually to modify the velocity or the pressure of the flow. We can find nozzles into the carburetor of reciprocating engines, on fuel injector or wind tunnels. But the application is of interest for us is their use on jet and rocket engines. The exhaust nozzles placed at the end of those engines expand the exhaust gas to increase its velocity and therefore its momentum. This means a force is acting on the gases and according with the Newton’s third law an equal and opposite force acts on the nozzle. They also collect and straighten the gas flow, and in vectorial or gimbal cases also direct it in the desired direction. The pressure ratio across the nozzle controls the expansion process. The maximum thrust for a given engine is obtained when the exit pressure \( P_e \) equals the ambient pressure \( P_0 \). If the exhaust gases are over expanded no further increase in the thrust is obtained, but shock waves and instabilities may appear inside the nozzle which can even destroy it or the whole engine.

The initial geometry of a circular-section rocket nozzle is fixed by the nozzle inlet, throat, and exit areas. The throat area \( A_t \) is determined by application of the mass flow parameter (MFP) shown in equation 4.5, because at the throat we find Mach 1 when it is choked. The total pressure \( P_t \) and temperature \( T_t \) are those at the combustion chamber, or at the collector in our case, were the velocity of the flow is so low that is considered it is in rest. When the throat area has been calculated the exit area \( A_e \) for a completely expanded nozzle is determined by means of the isentropic flow relations. With the relation between the total pressure and the exit pressure, which is the ambient pressure at the desired height, the exit Mach can be determined from equation 4.6. With it the relation between the exit and critical area (throat area) can be calculated from equation 4.7, and finally the exit area. An isentropic flow properties table can also be used to simplify the calculation. We must know the combustion chamber temperature \( T_c \) and pressure \( P_c \), mean values of \( \gamma \) and \( R \), the propellant mass flow rate \( \dot{m} \), and the designed nozzle pressure ratio \( P_c/P_a = P_c/P_e \).

\[
MFP = \frac{\dot{m} \sqrt{T_t}}{A} \frac{\sqrt{\gamma}}{R} \left( 1 + \frac{\gamma - 1}{2} M^2 \right)^{\frac{\gamma + 1}{2(\gamma - 1)}}
\]  

(4.5)

\[
P \frac{P}{P_t} = \left( 1 + \frac{\gamma - 1}{2} M^2 \right)^{\frac{\gamma - 1}{\gamma - 1}}
\]  

(4.6)

\[
\frac{A}{A^*} = \left( \frac{\gamma + 1}{2} \right)^{-\frac{\gamma + 1}{2(\gamma - 1)}} \left( 1 + \frac{\gamma - 1}{2} M^2 \right)^{\frac{\gamma + 1}{2(\gamma - 1)}}
\]  

(4.7)
The inlet area can be estimated supposing a low flow speed and obtaining the area relation for this condition. This will give us an approximation of the required inlet area. In our case a bigger one should be selected for the pipes to avoid problems of restrictions of the flow. After $A_c$, $A_t$ and $A_e$ have been established the problem of the nozzle contour design remains. In order to reduce the weight and due to space limitations, a nozzle with the minimum length is desired. Different profiles can be used, but to simplify the manufacture we are going to use a conical nozzles. This kind of nozzle represents a compromise between length, thrust, and ease of manufacturing design criteria weighted somewhat in favor of the last factor. It consists of two truncated cones, joined top to top along their axis by a suitable radius to form the nozzle throat. For the converging part of the nozzle a rather rapid change in cross section is permissible, with a half-angle of $40^\circ$ commonly used. But the divergence angle of the supersonic portion of the nozzle, is limited by flow separation considerations and must not exceed $15^\circ$.

![Figure 4.5: Conical nozzle profile](image10)

When a rocket engine have to work at very different altitudes, as is the case of a launcher, the atmospheric pressure it finds varies a lot. The nozzle design must be optimized for a determined pressure ratio, as the geometry of the nozzle is fixed. This ratio is taken for the lower altitude is expected the engine will work. This is done in order to avoid over-expansion problems, and also because it usually match with the point where maximum thrust is required, as the lift-off. Some aviation engines make use of variable geometry nozzles, but this system would result too heavy and complex to be able to work with the constraints find on rocket engines.

At the design of the control system nozzles, besides they have to work from 37 kilometers height to near vacuum conditions at the apogee at around 250 kilometers height, we also have to deal with two propellants with very different properties. However we are going to calculate and optimize the nozzle to work with the selected cold gas propellant, as it represent 90% of the control time. Also it is important take into account that the space inside the launcher is small, and the length of the nozzle is limited to 20 millimeters.

Nitrogen has been selected in the previous section as cold gas propellant, the mass flow is going to be $2.280 \text{ g/s}$. Although the gas would be stored at around 200 bar the system only has been tested up to 30 bar. Hence nitrogen must be expanded from 200 at the tank to 30 bar at the collector. That is going to be our total pressure. As the balloon’s ascension is very slow and the cold gas is expanded from 200 to 30 bar, it is going to be cold down. We have estimated its temperature at $250 \text{ K}$, that it is going to be our total temperature. The nozzles design point is at 37 km where the atmospheric pressure is $410.57 \text{ Pa}$ which gives a design nozzle pressure ratio of $7.317 \cdot 10^3$. Applying the MFP relation we obtain a throat area of $0.1013 \text{ mm}^2$, which means a circular throat of $0.3591 \text{ mm}$ in diameter. From
the nozzle design pressure ratio and with the help of an isentropic flow properties table the exit Mach is determined at 7.65 Mach and the area ratio 
\( A/A^* = 155 \) [14]. With this ratio the nozzle exit has an area of 15.70 mm\(^2\) and a diameter of 4.47 millimeters. For the nozzle inlet we have considered a flow speed of 0.02 Mach. For that Mach we have 
\( A/A^* = 28.94 \) which result in an inlet area of 2.93 mm\(^2\) and a diameter of 1.932 mm. A bigger area will not have an important effect in the nozzle as it is driven by the throat area. The pipes up to the nozzles must have a larger diameter to avoid fluid losses.

By trigonometry the minimum nozzle length is determined. According with it the convergent part should be at least 0.937 mm long for a half half-angle of 40\(^\circ\) and the divergent part have a length of 7.67 mm for a half-angle of 15\(^\circ\). Hence the minimum length of the nozzle is about 9 mm, much less than our maximum of 20 mm. Therefore we can use smaller half-angles if this simplifies the building process or improves the nozzle behavior.

4.4. Actuators

An actuator is a mechanical device for moving or controlling a mechanism or system. It is operated by a source of energy, usually in the form of an electric current, hydraulic fluid pressure or pneumatic pressure. It converts that energy into some kind of motion. Hydraulic and pneumatic actuators require storage tanks and plumbing for the working fluid, that increase the weight. Also a source of pressure, like a pump which requires another source of energy, or a pressurized storage is needed. Because of that, those systems are intended to be substituted for electrical, electro-hydraulic or electro-pneumatic systems, for aerospace applications.

In our case the required force from the actuator is very small and the displacement very short, hence the weight and complexity of hydraulic and pneumatic systems is not justifiable. Electric actuators usually make use of electric motors that produce rotational movement which can be used as that or converted to rectilinear movement by means of gears or levels. Another types of electrical actuators are based on solenoids\(^2\). This type of actuator produce the linear displacement of one part of the actuator, usually the core, when an electric current or a variation of it, is applied to the solenoid. They are very popular for

\(^2\)http://en.wikipedia.org/wiki/Solenoid
fluid circuits applications because they can be integrated within a valve, which is often referred to as solenoid valve. A type of electrical actuators between the previous mentioned is stepper motor\(^3\). Although there are different types of stepper motors, the working base is the use of different coils to orientate the core, feeding any of them. Rotation is achieve by a correct sequence of feeding, but the maximum torque produced by these devices is lower than that produced by a common electric motor. They are widely used in applications where control the position is the main aim, as numerical control (CNC) machines, robots or floppy drives.

![Figure 4.7: From left to right: Solenoid, Servo motor, DC Motor and Stepper Motor](image)

For the prototypes we have used standard hobby servomotor as actuators, because they are found in any hobby shop and are easily controlled by an Arduino which is the control board we employ on them. Even thought they are good enough for the prototypes to perform some tests, they result heavy, bulky, slow and their energy consumption is too high.

The actuating frequency required according with the simulations, of more than $500\text{Hz}$, is a very limiting factor, probably the most important. The only actuator type that usually shows higher working frequencies, and hence that can be used in our system is the solenoid. These actuators not only present high working frequency, they also present relatively low power consumption and less problems to work at very low temperatures, because they do not have mechanical elements as gear. Furthermore they are very compact and produce a linear movement, simplifying the lever actuation and the placement of the actuator.

### 4.5. System Integration on the Launcher

Once all the main components have been sized, a better arrangement of them inside the launcher can be done. The heaviest and biggest component is the cold gas tank. Cold gas is only necessary before the second stage ignition for any of the considered cases. Hence it will stay with the first stage after the separation and it must be located in the interstage part. The location of the other components is very dependent on the final control procedure. If control of the second stage is performed all the other components of the control system must be located within the second stage. On the contrary, if spinning stabilization is the selected procedure, all the components but the nozzles can be placed down and remain with the first stage and the cold gas tank after the separation. With the first concept a marginal control of the second stage can be produced, with an important

\(^3\text{http://en.wikipedia.org/wiki/Stepper\_motor}\)
increase in the weight and quantity of fuel of the second stage. The second concept reduce the weight and amount of fuel, and also reduce the component constraints as they only work with cold gas, especially on the materials. Both configurations are going to be detailed.

4.5.1. Second Stage Control with Purge Gases

This configuration is the first considered, and the design of every component has been oriented towards it. All the components but the cold gas tank are located in the second stage as was explained in the second chapter. At that point our intention was to use the second stage as storage tank and use the same collector to purge out form it both cold gas and combustion gases. As we have seen the second stage is not suitable as cold gas tank, and a connexion between the actual tank and the collector must be provided. The collector has to be connected on top of the second stage and a provisional seal or a valve should be placed in the collector to avoid cold gas flow inside the stage. Another possibility is connect the tank and the collector through the grain channel, and use the stage as expansion chamber to reduce the pressure from 200 bar to 30 bar. In this case the influence that low temperature and high pressure can have on the grain must be taken into account.

The valves are brazed to the collector and the nozzles connected to them by pipes, as was shown in chapter two. Special attention must be put on the sections of the different parts. According with the calculations the minimum cross area required for each actuator at the nozzle inlet is almost $3 \text{ mm}^2$, bigger valves than those build for the prototypes should be done, to maintain a larger area and counteract the pressure drop. As the valve are in couples the cross section of the link between the collector and the valve body must be at least twice (more than $6 \text{ mm}^2$). Some times the four actuator may be opened at the same time, hence the collector should allow a flow high enough to feed all of them. This means a cross sectional area larger than $12 \text{ mm}^2$, or a diameter of around $4 \text{ mm}$.

The electric actuators will be placed on top of the valves and will be joined to the valves level by means of a rigid steel wire. They will be mounted in a plate or structure that is not already defined as the size of the actuator is not already know. The properties of the batteries required for the systems are very dependent on the actuator characteristics. They should be located some where in the same structure as the actuators. If and independent micro-controller board is used for the control system it must be located in a place where it will not be affected by the magnetic fields produced by the actuators, as at the launcher’s nose.

4.5.2. Second Stage Uncontrolled

With the second configuration any component but the nozzles must be in the second stage. The utilization of only cold gas allows the use of lighter materials with lower melting points. Due to the location of the nozzles in the second stage, an interface between the cold gas and them is still required. The collector on the second stage engine is not required any
more, because due to the problems the collector and the flow of the cold gas through the grain could produce, another method has been chosen. In this method the gas is expanded inside a collector to which the valves are attached. From there it will be conducted to the nozzles within pipes. Those pipes have two parts, one that remains with the first stage, and other of larger section, that stay in the second stages and at the same time works as structural part to maintain the satellites in place. The first part of the pipe go a few millimeters inside the second part, and a rubber o-ring will seal the join, while allows the displacement of the pipe required for the stages separation. As the pipe is very long its section must be much bigger than in the previous configuration, at least $4\, mm^2$, to counteract the higher pressure drop on long pipes.

The actuators should be placed if possible around the second stage nozzle to reduce the length of the launcher as much as possible. In this case the use of individual valves instead of the couples of valves used in the previous configurations may be required to achieve this arrangement. The batteries can be used also to feed the satellite release system, hence they should be placed on the second stage and an electrical interface should be done between both stages. If an independent board is used, it should be placed in the first stage if it is possible avoid the electro-magnetic interferences from the actuators. Even if the satellite is in charge of the control another interface is necessary. It can be between the satellite sensors and the control board, or between the control device and the actuators.

4.6. Prototypes

We have build several prototypes in order to validate the design and run some test with them. They have allowed us see failures, improve the design and make measurements of different parameters of the system, which will be detailed in the next section.

Different prototypes of the valve and nozzle system have been build. The first prototype was build at the very beginning of the design, to validate the concept we had in mind. It already employed electrical strip connectors for the valve body as the final version explained before. But inside instead of have different channels it was a straight tube, which was blocked by screws whose tips were mechanized in the lathe to a spherical shape. This mechanical process was very complicated to perform in such a tiny screw ($M3 \times 0.5 \times 5$) and also it was very inaccurate.

The pipes from the valves to the nozzles, which in fact were made in the tube itself, were done of copper tube used in air conditioning equipments. This tube has a very thick wall, that in addition with the high density of the copper result in a very heavy part. Furthermore copper present very bad machining behavior, because it is very soft. Nozzles were made on the copper tube simply by varying the exit area pressing the tips of the pipe. Everything was soldered together using soldering paste as solder alloy. It had a final mass of 17.9 grams without actuators.
The second prototype already included the final valve system. The copper pipes were substituted by thinner brass tubes, what reduced significantly the weight. It was equipped with convergent nozzles that were made from a solid bar of brass drilling it with a bit of 0.7 millimeter of diameter to obtain the exit area and a guide hole for the conic mill used for the convergent section. The same soldering process and material as in the first prototype were used here. The valves in this prototype were already done following the design presented in section 4.1. As they were the first valves to be build some inaccuracies were committed, but their behavior was still good, with only a very tiny loss of gas in the closed position.

Another two prototypes were build. Those did not represent the whole system, but only one valve and nozzle was included. They were build to perform pressure and temperature test, and the first on-flight test. The nozzles in those prototypes were substituted by convergent divergent nozzles made of aluminum. Their length of 35 mm was adjusted to fit inside the final launcher dimensions.
Also a prototype for the actuation system was build up. It consist in four actuator, which are common hobby servomotors, an Arduino Mini board and a 9 V AA battery. Everything was mounted on a fiberglass plate. The Arduino Mini is a small microcontroller board based on the ATmega168 microprocessor. It has 14 digital input/output pins, and 6 of them can be used as PWM$^4$, required to control the actuators. It operates at 5 V and can be feed at 7 – 9 V which is perfect to use a 9 V AA battery. The Arduino Mini has a 16 kB flash memory were the code is uploaded from a computer using and FTDI$^5$. The servos of 1 kg/m torque at 5 V were feed through the Arduino board but due to their high consumption the board resets and not enough power was provided to move four servos at the same time. Because of that they were feed from an outer power supply, in order to avoid expend many batteries for the tests. The battery was used to avoid the flash memory was erased when the power supply was removed.

---

$^4$PWM Pulse-With Modulation  
$^5$USB to serial interface
4.7. Tests

Some tests have been driven to obtain experimental results and to validate the design and the prototypes. Further tests should be done to check other components of the system, their integration in the launcher and finally flight tests to validate the control system.

The first series of tests was performed with the first valve prototype, that was connected to an air compressor. The first test consisted in apply pressure with the valves closed to check their sealing. The valves worked, but their sealing was not very good. The next test was to measure the thrust produced by the system. For that the prototype was placed over a digital scale with two nozzles up, whose valves were opened and a pressures from 2 to 7 bars were applied and the mass measured by the scale recorded. With this method the results were not very accurate because of the interference the plastic tube, that conducts the pressurized air from the compressor to the valve system, did in the measurements.

Another series of tests was driven with the second prototype. To minimize the interferences seen during the test with the first prototype, this prototype was installed in a pendulum, where the thrust was calculate as function of the sustained angle. An angle meter was used to measure the angle and as structure for the pendulum. The results obtained are shown in table 4.4 where the angle obtained for each couple of nozzles is tabulated against the pressure.

For the next test we used the high pressure nitrogen bottle as gas source. It has a manometer that reduce the pressure, which can be adjusted from 1 to 30 bar. The first test was to check the valve sealing and their behavior at high pressure (30 bar). The valve were open and closed by hand at different pressures, and the response was very good with only a very small leak of nitrogen even at high pressure. The pendulum test was repeated, but in this case an electronic gyroscope was the measurement element. It has a high accuracy and takes measurements with a frequency of $10 \text{Hz}$, allowing us to record and study the angle and its variation with a computer. The gyro axis has bearings that make softer the turn and improve the accuracy of the results.
Table 4.4: Second Prototype Test Results

<table>
<thead>
<tr>
<th>Pressure (bar)</th>
<th>1st Couple</th>
<th>2nd Couple</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0°</td>
<td>0°</td>
</tr>
<tr>
<td>1</td>
<td>1°</td>
<td>0.5°</td>
</tr>
<tr>
<td>2</td>
<td>4°</td>
<td>3°</td>
</tr>
<tr>
<td>3</td>
<td>6°</td>
<td>7°</td>
</tr>
<tr>
<td>4</td>
<td>9°</td>
<td>9°</td>
</tr>
<tr>
<td>5</td>
<td>11°</td>
<td>13°</td>
</tr>
<tr>
<td>6</td>
<td>14°</td>
<td>17°</td>
</tr>
<tr>
<td>7</td>
<td>17°</td>
<td>21°</td>
</tr>
</tbody>
</table>

The prototype of the actuation system was incorporated to actuate the valves to simulate a closer system to the actual one. Furthermore open or close the valves by hand would introduce big disturbances in the measurements. The actuation system was placed at the end of the pendulum, and the actuators arms were joined to the valve levers by means of steel wires. As the servos required an external power supply very tiny cables, big enough for the operating voltage and current, were used to join the Arduino Mini board with the FTDI that was fixes to the table. With this configuration the computer USB was used at the same time to send the open/close commands to the board and to feed the actuators. Several trains of open close commands were done. Angles around 3° were obtained due to the high torque produced by the important mass of the actuation system and its distance to the turn center. From a dynamic study of the pendulum we calculated the system thrust at 0.05 N for each nozzle. This thrust is very low, but it is achieved using a very rough convergent nozzle. With a convergent divergent nozzle the thrust would be higher. The important result of this experiment is the consistency of the results obtained during the open/close trains show in figure 4.14, and the successful integration of the actuation and valve systems.

Figure 4.13: Configuration of the High Pressure Pendulum Test
With the last valve prototypes we have run only one test. In it we install the prototype on a can, which was charged with half a black powder grain. That grain is from a rocket engine used in rocket models. The engine has its own nozzle, but we removed it and place only the grain inside the can. In this test we wanted to test the valve and a new convergent divergent nozzle with high temperature flow. At the same time an igniter feed with two tiny cables that went inside the can through the control system collector was also tested as well as the installation sealing. The grain was located in the top part of the can close to the igniter and the collector. The igniter test was a success, and the valve and nozzle seems to work well during the first hundredths of second until they were obstructed due to the high volume of solid particles black powder produces during its combustion.

Figure 4.14: Train of Pulses Measured During Second Prototype’s High Pressure Tests
Figure 4.15: Experiment Configuration and Frame Where the Gas Flow Through the Nozzle Is Slightly Visible
Study of a nozzle vector control for a low cost mini-launcher
CHAPTER 5. CONCLUSIONS

5.1. General conclusions

We have studied the implementation of a reaction control system on-board a low cost mini-launcher. Reaction controls have already been used in launchers, but this new concept of launcher present many challenges. Reduction of weight and cost, and the small size of the launcher and therefore of every component are the main constraints for this system. Furthermore, the amateur construction of the system, required to fit the small budget imposed by the N-Prize rules, is an extra challenge.

According with the simulations the control of the launcher with our new concept reaction control is possible. High frequency control commands, in the order of $500 \text{ Hz}$, are needed. Electrical actuator presenting that and higher working frequencies are easily found. However those high frequencies can be a problem due to the inertia fluid systems present, which requires some time to achieve a steady flow, and therefore thrust. Further investigation about this point must be done to finally validate or discard this method.

The employment of hot gases purged from the stages during the whole flight would require the use of high temperature resistant materials in many components. Also the control action would be limited to the burning time of each stage. Furthermore a duplication of the systems is required as a detachable interface for high pressure and temperature gas between stages is not reliable. These issues increase the weight and cost of the system, showing the system as inefficient.

The use of cold gas presents a big inconvenient that is its storage tank. A high pressure tank of a size equivalent to the second stage has a big impact over the launcher layout. Build a very light tank capable of withstand more than 200 bar of pressure is the main challenge of this configuration. On the other hand the high temperature requirements over the other components of the system are greatly relaxed, even if purged gases from the second stage are finally used. Although, due to the tank, this system is almost as heavy and expensive as the other, it provide more flexibility as the control is not limited to the burning time. Hence this system is the most efficient of those studied in this work.

The final validation of the system must be done with ground test of the complete system, including the IMU platform, and finally flight tests.

5.2. Environmental impact

When this new technology will become a reality it will change the current standards for the space platforms and drastically reduce the energy required to put this platforms in orbit. This means the reduction of the launchers size, and the fuel quantity, which will represent a huge reduction in the emission during the launch. But also the waste production, energy consumption and process contamination related with their construction.
With respect to the environmental impact of this study it has been maintained to the mini-
mum. The small size and the use of already available metal components, have limited the
issues related with the construction of the prototypes to anecdotal values. The use of inert
gases for most of the tests has reduce their impact. The main sources of contamination
are the used batteries, and the test done with black powder. Future test will require the
use of larger amounts of propellant, and complete systems that in some cases can be lost.
However even in those cases the environmental impact is very small, compared with the
impact traditional development works have in this field.

5.3. Future work

This new concept of launching systems is very promising and will open a new field in
the access to space, with an important potential market in universities and technological
enterprises. However maturation of this very new technology is still required before this
will take place.

Attending to the launcher control, we advice some possible improvements on the system,
according to problems found during this study. The utilization of a solid based gas gen-
erator instead of cold gas, in order to reduce the volume of the system, and/or move to
a combination of thrust vanes and reaction control system, specially if problems with high
frequencies are found.
CHAPTER 6. GLOSSARY

NOTE: Wikipedia references were revised for accuracy in the scope of this master thesis.


ASTRONAUTIX. http://www.astronautix.com/props/index.htm


COTS. Commercial-off-the-shelf


ESA. European Space Agency. http://www.esa.int/

Femto-satellite. A less than 100 grams satellite


Li-Ion batt. http://en.wikipedia.org/wiki/Lithium-ion_battery

LNA. Low Noise Amplifier


MCU. Main Control Unit
**MEO.** Medium Earth Orbit. http://en.wikipedia.org/wiki/Medium_Earth_orbit

**Mini-launcher.** A less than 100 kg launcher

**Modulus of Elasticity.** http://en.wikipedia.org/wiki/Elastic_modulus

**Moon2.0 project.** http://code.google.com/p/moon-20/

**N-Prize.** http://en.wikipedia.org/wiki/N-Prize


**Sonic boom.** http://en.wikipedia.org/wiki/SonicBoom

**Space Payload Paradigm.** Is the engineering process of designing a space mission around its payload and not around the space industry.

**Specific heats.** http://www.grc.nasa.gov/www/k-12/airplane/specheat.html


**Thermal conductivity.** http://en.wikipedia.org/wiki/Thermal_conductivity

**Thermal expansion, Table of.** http://www.wisetool.com/designation/te.htm

**Trial and error methodology.** http://en.wikipedia.org/wiki/Trial_and_error

**Universal gas constant.** http://www.grc.nasa.gov/www/k-12/airplane/eqstat.html

**WikiSat organization.** http://code.google.com/p/moon-20/wiki/WikiSat_Engineering_Management_Plan
[1] N-Prize official web page (September 2010)
   http://www.n-prize.com/


[36] Google code project Moon2.0 (February, 2010)

http://code.google.com/p/moon-20/
### APPENDIX A. SPECIFIC IMPULSE OF LIQUID AND SOLID PROPELLANTS

#### Liquid Propellants

**Table A.1: Specific Impulse (Chemical propellants)**

<table>
<thead>
<tr>
<th>Propellant</th>
<th>Specific Impulse (s)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Monopropellants</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low-energy</td>
<td>160 to 190</td>
<td>Hydrazine</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ethylene oxide</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hydrogen peroxide</td>
</tr>
<tr>
<td>High-energy</td>
<td>190 to 230</td>
<td>Nitromethylene</td>
</tr>
<tr>
<td><strong>Bipropellants</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low-energy</td>
<td>200 to 230</td>
<td>Perchloryl fluoride - Available fuel</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Analine - Acid</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JP-4 - Acid</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hydrogen peroxide - JP-4</td>
</tr>
<tr>
<td>Medium-energy</td>
<td>230 to 260</td>
<td>Hydrazine-Acid</td>
</tr>
<tr>
<td>High-energy</td>
<td>250 to 270</td>
<td>Ammonia-Nitrogen tetroxide</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$LO_2$ - JP-4 or Alcohol</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$LO_2$ - Hydrazine - Chlorine trifluoride</td>
</tr>
<tr>
<td>Very high-energy</td>
<td>270 to 330</td>
<td>$LO_2$ - fluorine - JP-4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$LO_2$ - ozone - JP-4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$LO_2$ - Hydrazine</td>
</tr>
<tr>
<td>Super high-energy</td>
<td>300 to 385</td>
<td>Fluorine - Hydrogen</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fluorine - Ammonia</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ozone - Hydrogen</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fluorine-Diborane</td>
</tr>
<tr>
<td>Propellant</td>
<td>Specific Impulse (s)</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>----------------------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td>Potassium perchlorate</td>
<td>170 to 210</td>
<td>With thiokol or asphalt</td>
</tr>
<tr>
<td>Ammonium perchlorate</td>
<td>170 to 210</td>
<td>With thiokol</td>
</tr>
<tr>
<td></td>
<td>170 to 210</td>
<td>With rubber</td>
</tr>
<tr>
<td></td>
<td>210 to 250</td>
<td>With polyurethane</td>
</tr>
<tr>
<td></td>
<td>210 to 250</td>
<td>With nitropolymer</td>
</tr>
<tr>
<td>Ammonium nitrate</td>
<td>170 to 210</td>
<td>With polyester</td>
</tr>
<tr>
<td></td>
<td>170 to 210</td>
<td>With rubber</td>
</tr>
<tr>
<td></td>
<td>210 to 250</td>
<td>With nitropolymer</td>
</tr>
<tr>
<td>Double base</td>
<td>170 to 250</td>
<td></td>
</tr>
<tr>
<td>Boron</td>
<td>200 to 250</td>
<td>Metal comp. and oxidant</td>
</tr>
<tr>
<td>Lithium</td>
<td>200 to 250</td>
<td>Metal comp. and oxidant</td>
</tr>
<tr>
<td>Aluminum</td>
<td>200 to 250</td>
<td>Metal comp. and oxidant</td>
</tr>
<tr>
<td>Magnesium</td>
<td>200 to 250</td>
<td>Metal comp. and oxidant</td>
</tr>
<tr>
<td>Perfluoro - type</td>
<td>250 and above</td>
<td></td>
</tr>
</tbody>
</table>
In this appendix are listed all the different codes used for the simulations.

B.1. Vanguard Transfer Function Definition

This section contains the M-files used to define the Vanguard transfer functions.

Longitudinal Vanguard Transfer Function

```matlab
function [sys] = Long_control_delta(sys)
clear all

%Launcher Longitudinal Dynamical Motion

% Variables
m = 445; % Mass
U = 1285; % Velocity
d = 3.75; % Launcher diameter
S = pi/4*d^2; % Launcher section
q = 585; % Dynamic pressure
g = 32; % Gravity
% T = ; % Thrust
l = 27; % Control arm
THETA_0 = ; % Angle between horizon and reference axes (degrees)
theta = ; % Attitude angle (degrees)
THETA = 68.11; % THETA_0 + theta; % Inclination respect to horizon (degrees)
Iy = 115000; % Inertia moment
Czalpha = -3.13; % Aerodynamic force coefficient
Cmalpha = 11.27; % Aerodynamic moment coefficient
Cmdelta = -34.25; % -T*l/(S*q*d); % Control moment
Czdelta = -4.63; % -T/(S*q); % Control force
Cw = -m*g/(S*q); % Weight coefficient
Damp = -0.321; % Damping in pitch

% Matrix definitions
s3 = m*U*ly/(S^2*q^2)*d);
a31 = -(Cw*sind(THETA) + Cmalpha)/s3;
a32 = -((Czalpha*Damp) - m*U*Cmalpha/(S*q))/s3;
a33 = -((Iy*Czalpha/(S*q*d)) - m*U*Damp/(S*q))/s3;
c11 = ((Cmalpha*Czdelta) - (Czalpha*Cmdelta))/s3;
c12 = (m*U*Cmdelta/(S*q))/s3;
c13 = 0;
A = [0 1 0
     0 0 1
     a31 a32 a33];
B = [0;0;1];
C = [c11 c12 c13];
D = [0];

% System definition
sys = ss(A,B,C,D);
```

Lateral Vanguard Transfer Function

```matlab
function [sys] = lat_control_delta(sys)
clear all
% Variables
m = 445; % Mass
U = 1285; % Velocity
d = 3.75; % Launcher diameter
S = pi/4*d^2; % Launcher section
q = 585; % Dynamic pressure
g = 32; % Gravity
T = % Thrust
l = 27; % Control arm
CHI = 68.11; % Lateral inclination respect to horizont (degrees)
Iz = 115000; % Inertia moment
Cybeta = 3.13; % Aerodynamic force coefficient
Cmbeta = 11.27; % Aerodynamic moment coefficient
Cmdelta = -34.25; % T*l/(S*q*d); % Control moment
Cydelta = 4.63; % -T/(S*q); % Control force
Cw = -m*g/(S*q); % Weight coefficient
Damp = -0.321; % Damping in pitch

% Matrix definitions
s3 = m*U*Iz/(S^2*q^2*d);
a31 = 0; % (-Cw*sind(CHI)*Cmbeta)/s3
a32 = ((Cybeta*Damp)+(m*U*Cmbeta/(S*q)))/s3;
a33 = ((Iz*Cybeta/(S*q*d))+(m*U*Damp/(S*q)))/s3;
c11 = ((Cybeta*Cmdelta)+(Cybeta*Cmdelta))/(S*q)/s3;
c12 = (m*U.Cmdelta/(S*q))/s3;
c13 = 0;
A = [0 1 0
     0 0 1
     a31 a32 a33];
B = [0; 0.1];
C = [c11 c12 c13];
D = [0];
% System definition
sys = ss(A,B,C,D);
```
function [sys] = Rot_control_delta(sys)
clear all

% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables

% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Variables
% Vari
B.2. Wiki-Launcher Transfer Function Definition (angle as output)

This section contains the M-files used to define the Wiki-Launcher transfer functions before the PID was implemented in the control loop.

Longitudinal Wiki-Launcher Transfer Function

```matlab
function [sys] = longitudinal_control(sys)
clear all
% % % % %
% Launcher Longitudinal Dynamical Motion % %
% attitude angle in radians
% Variables
m = 3.77; % Mass (kg) 3.77kg total
U = 2738; % Velocity (m/s)
d = 0.066; % Launcher diameter (m) 66mm
S = pi/4*d^2; % Launcher section (m^2)
q = 6884.95; % Dynamic pressure (Pa = N/m^2)
g = 9.81; % Gravity (m/s^2)
l = 0.389; % Control arm (m)
L = 0.769; % Rocket total length (m)

% Variables
THETA_0 = ; % Angle between horizontal and reference axes (degrees)
THETA = 40; % THETA_0+theta; % Inclination respect to horizon (degrees)
Iy = m*(L^2)/12; % Inertia moment (kg/m^2)
Czalpha = -2.5; % Aerodynamic force coefficient
Cmalpha = 11.27; % Aerodynamic moment coefficient
Cmdelta = -l/(S*q*d); % Control moment
Czdelta = 1/(S*q); % Control force
Cw = -m*g/(S*q); % Weight coefficient
Damp = -0.321; % Damping in pitch

% Matrix definitions
s3 = m*U*Iy/(S^2*q^2*d);
a31 = -(Cw*sind(THETA)*Cmalpha)/s3;
a32 = -((Czalpha*Cmdelta)-(m*U*Cmalpha/(S*q)))/s3;
a33 = -((Czalpha/Czdelta)-(Czalpha+Cmalpha))/s3;
c11 = ((Cmalpha*Czdelta)-(Czalpha+Cmdelta))/s3;
c12 = (m*U*Cmdelta/(S*q))/s3;
c13 = 0;
A = [0 1 0
     0 0 1
     a31 a32 a33];
B = [0:0:1];
C = [c11 c12 c13];
D = [0];
% System definition
sys = ss(A,B,C,D);
```
Lateral Wiki-Launcher Transfer Function

```matlab
function [sys] = Lateral_control(sys)
% % Lateral Wiki-Launcher Transfer Function %
% attitude angle in radians

% Variables
m = 3.77; % Mass (kg)  3.77kg total
U = 2738; % Velocity (m/s)
d = 0.066; % Launcher diameter (m) 66mm
S = pi/4*d^2; % Launcher section (m^2)
g = 6884.95; % Dynamic pressure (N/m^2)
l = 9.81; % Gravity (m/s^2)
l = 0.389; % Control arm (m)
S = pi/4*d^2; % Launcher section (m^2)
q = 68.11; % Lateral inclination respect to horizont (degrees)
Iz = m*(l^2)/12; % Inertia moment (kg/m^2)
Cybeta = 2.5; % Aerodynamic force coefficient
Cmbeta = 11.27; % Aerodynamic moment coefficient
Cmdelta = l/(S*q*d); % Control moment
Cydelta = f/(S*q); % Control force
Cw = -m*g/(S*q); % Weight coefficient
Damp = -0.321; % Damping in pitch

% Matrix definitions
s3 = m*U*Iz/(S^2*q^2*d);
a31 = 0; % -(Cw+sin(0+Cmbeta))/s3
a32 = (Cybeta+Damp)/(m+U+Cmbeta/(S*q))/s3;
a33 = (Iz+Cybeta/(S*q*d))/(m+U+Cmbeta/(S*q))/s3;
c11 = (Cmbeta*Cydelta)+(Cybeta*Cmdelta)/s3;
c12 = (m+U+Cmdelta)/(S*q))/s3;
c13 = 0;
A = [0 1 0
0 0 1
a31 a32 a33];
B = [0:0:1];
C = [c11 c12 c13];
D = [0];

% System definition
sys = ss(A,B,C,D);
```

Simulation Source Codes 63
Rotational Wiki-Launcher Transfer Function

```matlab
function [sys] = Rotational_control(sys)
    clear all
    \% \% \% Launcher Rotational Dynamical Motion \% \% \%

    \% Variables
    m = 3.77; \%Mass (kg) 3.77kg total
    d = 0.066; \%Launcher diameter (m) 66mm
    S = pi/4*d^2; \%Launcher section \( \text{m}^2 \)
    q = 6884.95; \%Dynamic pressure \( \text{N/m}^2 \)
    ry = 0.5*d*sind(60); \%Distance between Nozzle position and Y axe (m)
    rz = 0.5*d*cosd(60); \%Distance between nozzle position and Z axe (m)
    Ix = 0.5*m*(d/2)^2; \%Inertia moment \( \text{kg/m}^2 \)
    Cldelta = -cosd(45)*(ry-rz)/(S*q*d); \%Rolling control moment

    \% Matrix definitions
    s2 = Ix/(S*q*d);
    c11 = Cldelta/s2;
    c12 = 0;
    A = [0 1
          0 0];
    B = [0; 1];
    C = [c11 c12];
    D = [0];

    \% System definition
    sys = ss(A,B,C,D);
```
B.3. Wiki-Launcher Transfer Function Definition (angle rate as output)

This section contains the M-files used to define the Wiki-Launcher transfer functions used when the PID was implemented in the control loop.

**Longitudinal Wiki-Launcher Transfer Function**

```matlab
function [sys] = Longitudinal_control(sys)
    clear all
    % Launcher Longitudinal Dynamical Motion
    % attitude angle in radians
    m = 0.52623; % Mass (kg)
    U = 1000; % Velocity (m/s)
    d = 0.066; % Launcher diameter (m) 66mm
    S = pi/4*d^2; % Launcher section (m^2)
    q = 0.0001; % Dynamic pressure (Pa = N/m^2)
    g = 9.81; % Gravity (m/s^2)
    l = 0.060; % Control arm (m)
    L = 0.140; % Rocket total length (m)
    THETA_0 = THETA; % Angle between horizon and reference axes (degrees)
    theta = THETA_0 + THETA; % Attitude angle (degrees)
    Iy = m*(L^2)/12; % Inertia moment (kg/m^2)
    Czalpha = -2.5; % Aerodynamic force coefficient
    Cmalpha = 11.27; % Aerodynamic moment coefficient
    Cmdelta = -l/(S*q*d); % Control moment
    Czdelta = 1/(S*q); % Control force
    Cw = -m*g/(S*q); % Weight coefficient
    Damp = -0.321; % Damping in pitch 

    % Matrix definitions
    s3 = m*U*ly/(S^2*q^2*d);
    a31 = (-Cw*sin(THETA)*Cmalpha)/s3;
    a32 = -(Czalpha*Damp-m*U*Cmalpha/(S*q))/s3;
    a33 = -(-(ly+Czalpha*(S*q*d)-m*U*Damp/(S*q)))/s3;
    c11 = 0;
    c12 = ((Cmalpha+Czdelta)-(Czalpha+Cmdelta))/s3;
    c13 = (m*U+Cmdelta/(S*q))/s3;

    A = [0 1 0
         0 0 1
         a31 a32 a33];
    B = [0 0 1];
    C = [c11 c12 c13];
    D = [0];

    % System definition
    sys = ss(A,B,C,D);
```

Lateral Wiki-Launcher Transfer Function

function \[ \text{sys} = \text{lateral\_control(sys)} \]
clear all

% % % % % Launcher Lateral Dynamical Motion % % % % %

% Variables
\( m = 0.52623 \); %Mass (kg) 3.77 kg total
\( U = 1000 \); %Velocity (m/s)
\( d = 0.066 \); %Launcher diameter (m) 66mm
\( S = \pi/4 \cdot d^2 \); %Launcher section (m^2)
\( q = 0.0001 \); %Dynamic pressure (N/m^2)
\( g = 9.81 \); %Gravity (m/s^2)
\( \theta = \); %Thrust
\( l = 0.060 \); %Control arm (m)
\( L = 0.140 \); %Rocket total length (m)
\( \chi = 68.11 \); %Lateral inclination respect to horizont (degrees)
\( Cybeta = 2.5 \); %Aerodynamic force coefficient
\( Cmbeta = 11.27 \); %Aerodynamic moment coefficient
\( Cmdelta = 1/(S \cdot q \cdot d) \); %Control moment
\( Cydelta = 1/(S \cdot q) \); %Control force
\( Cw = -m \cdot g/(S \cdot q) \); %Weight coefficient
\( \text{Damp} = -0.321 \); %Damping in pitch"

% Matrix definitions
\( s3 = m \cdot U \cdot l / (S^2 \cdot q^2 \cdot d) \);
\( a_{31} = 0 \); %\(-Cybeta \cdot \sin(\chi) \cdot Cmbeta / s3\)
\( a_{32} = (Cybeta \cdot Damp + (m \cdot U \cdot Cmbeta / (S \cdot q))) / s3; \)
\( a_{33} = ((L \cdot Cybeta / (S \cdot q \cdot d)) + (m \cdot U \cdot Damp / (S \cdot q))) / s3; \)
\( c_{11} = 0; \)
\( c_{12} = (Cmbeta \cdot Cydelta + (Cybeta \cdot Cmdelta)) / s3; \)
\( c_{13} = (m \cdot U \cdot Cmdelta / (S \cdot q)) / s3; \)
\( A = [0 1 0; 0 0 1; a_{31} a_{32} a_{33}]; \)
\( B = [0;0;1]; \)
\( C = [c_{11} c_{12} c_{13}]; \)
\( D = [0]; \)

% System definition
\( \text{sys} = \text{ss}(A,B,C,D); \)
function [sys] = Rotational_control(sys)
clear all

% % % % %
Launcher Rotational Dynamical Motion % % % % %

% Variables
m = 0.52623; % Mass (kg) 3.77 kg total
d = 0.066; % launcher diameter (m) 66mm
S = pi/4*d^2; % Launcher section (m^2)
q = 0.0001; % Dynamic pressure (N/m^2)
ry = 0.5*d*sin(60); % Distance between Nozzle position and Y axe (m)
rz = 0.5*d*cosd(60); % Distance between nozzle position and Z axe (m)
Ix = 0.5*m*(d/2)^2; % Inertia moment (kg/m^2)
Cldelta = -cosd(45)*(ry-rz)/(S*q*d); % Rolling control moment

% Matrix definitions
s2 = Ix/(S*q*d);
c11 = 0;
c12 = Cldelta/s2;

A = [0 1
     0 0];
B = [0;1];
C = [c11 c12];
D = [0];

% System definition
sys = ss(A,B,C,D);
B.4. Simulink Pulse Modulator Code

Code of the embedded Matlab function used to modeling the pulse modulator, where its parameters can be tuned.

```matlab
function [y,tson1,tsoff1,gammak1,T1] = Modulator(T,t,tson,tsoff,gammak,T0)

% Pulse Modulator for Pulsed Control System to Implement in Simulink
% (by Roberto Rodriguez)

% Variables
U = 1;  % Nozzle Thrust Component
ton = 0.001;  % Minimum Pulse Duration (sec)
toff = 0.0015;  % Minimum Rest Between Successive Pulses (sec)
if (abs(T)>=U)
    T = T0;  % To avoid T goes out of limits
end

% Minimum Pulse 's Thrust
fmin = U*ton/(ton+toff);

% Calculation of Pulse Properties

if (t==0)
    % Pulse Duration (gammak) & Interval Between Starts of Pulses (Tk)
    if (abs(T)>=fmin)
        gammak = toff*abs(T)/(U-ABS(T));
        Tk = U*toff/(U-ABS(T));
    elseif (T==0)
        gammak = ton;
        Tk = ton + toff;
    else
        gammak = ton;
        Tk = U*ton/ABS(T);
    end

    % Time of Last Switch On
    tson = -Tk;

    % Time of Last Switch Off
    tsoff = -Tk + gammak;
    amp = 0;
else
    if ((t>tson) && (tson>tsoff))
        % Pulse Duration (gammak)
        if (abs(T)>=fmin)
            gammak = toff*abs(T)/(U-ABS(T));
        else
            gammak = ton;
        end

        if ((t-tson)<=gammak)
            amp = sign(T);
        else
            amp = 0;
            tsoff = t;
        end
    else
        % Interval Between Starts of Pulses (Tk)
        if (abs(T)>=fmin)
            Tk = U*toff/(U-ABS(T));
        elseif (T==0)
            Tk = gammak + toff;
        else
            Tk = U*ton/ABS(T);
        end
    end
end
```

if ((t - tsoff) <= (Tk - gammak))
    amp = 0;
else
    amp = sign(T);
    tson = 1;
end
end

y = U*amp;
tson1 = tson;
tsoff1 = tsoff;
gammak1 = gammak;
T1 = T;
APPENDIX C. CONTROL SYSTEM CODES
BASED ON ARDUINO LANGUAGE

In this appendix is collected the control system code developed in Arduino language which is going to be executed by a satellite or the control board to control the launcher. The pulse modulator library developed by us is also included.

C.1. Main Code

This is the main code of the of the control system software.

```c
#include "PID_Beta6.h"
#include "PulseModulator.h"

// PID Variables
double Input1, Output1, Setpoint1;
double Input2, Output2, Setpoint2;
double Input3, Output3, Setpoint3;
PID Lon(&Input1, &Output1, &Setpoint1, 3, 4, 1);
PID Lat(&Input2, &Output2, &Setpoint2, 3, 4, 1);
PID Rot(&Input3, &Output3, &Setpoint3, 3, 4, 1);

// Pulse Modulators
PulseModulator Out1(0, 1, 1, 1, 0.01, 0.98);
PulseModulator Out2(1, 1, 1, 1, 0.01, 0.98);
PulseModulator Out3(2, 1, 1, 1, 0.01, 0.98);
PulseModulator Out4(3, 1, 1, 1, 0.01, 0.98);

void setup()
{
    Setpoint1 = 0;
    Setpoint2 = 0;
    Setpoint3 = 0;
    Lon.SetInputLimits(-180.0, 180.0);
    Lat.SetInputLimits(-180.0, 180.0);
    Rot.SetInputLimits(-180.0, 180.0);
    Lon.SetOutputLimits(-1.0, 1.0);
    Lat.SetOutputLimits(-1.0, 1.0);
    Rot.SetOutputLimits(-1.0, 1.0);
    Lon.SetSampleTime(200);
    Lat.SetSampleTime(200);
    Rot.SetSampleTime(200);
    Lon.SetMode(AUTO);
    lat.SetMode(AUTO);
    Rot.SetMode(AUTO);
}

void loop()
{
    // Set Tuning Parameters based on
    // how close we are to setpoint
    if(abs(Setpoint1 - Input1) > 1.0)
        Lon.SetTunings(6.0, 4.0, 1.0);
    else
```
Lon.SetTunings(3.0, 4.0, 1.0);
if(abs(Setpoint2 - Input2) > 1.0)
    Lat.SetTunings(6.0, 4.0, 1.0);
else
    Lat.SetTunings(3.0, 4.0, 1.0);
if(abs(Setpoint3 - Input3) > 1.0)
    Rot.SetTunings(6.0, 4.0, 1.0);
else
    Rot.SetTunings(3.0, 4.0, 1.0);

// Read the attitude angles from the gyros
Input1 = analogRead(0); // X gyro
Input2 = analogRead(1); // Y gyro
Input3 = analogRead(2); // Z gyro

// Give the PID the opportunity to compute if needed
Lon.Compute();
Lat.Compute();
Rot.Compute();

// Pulse modulators to control the nozzle actuators
Out1.Control(Output1 + Output2 + Output3);
Out2.Control(Output1 - Output2 - Output3);
Out3.Control(-Output1 - Output2 + Output3);
Out4.Control(-Output1 + Output2 - Output3);
C.2. Pulse Modulator Library

This is the library that contains the code used to produce the modulated actuation signal which is send to the actuators.

```cpp
#ifndef PulseModulator_h
#define PulseModulator_h

class PulseModulator
{
    public:
        PulseModulator(
            int pin1,
            int ton1,
            int toff1,
            float U1,
            float Tmin1,
            float Tmax1
        );
        // Sets the initial values for the pulse modulator.
        void Control(
            float T    // Required nozzle thrust
        );
        // Performs the conversion from analog input
        // to modulated digital output.
        // The first time is executed, initializes 'gammak',
        // 'Tk', 'ton' and 'tsoff'.
        // Checks if the ON or OFF period has passed and
        // sets the digital pin accordingly. This function
        // must be called every time loop() cycles.

    private:
        int pin;
        int ton;
        int toff;
        float U;
        float Tmin;
        float Tmax;
        float fmin;
        int Tk;
        int gammak;
        unsigned long tson;
        unsigned long tsoff;
        unsigned long t;
    }

#undef PulseModulator_h
```

#include "wiring.h"
#include "PulseModulator.h"

PulseModulator::PulseModulator(
    int pin1, // Digital pin number to control the nozzle actuator
    int ton1, // Minimum pulse duration (millis)
    int toff1, // Minimum rest between successive pulses (millis)
    float U1, // Nozzle thrust component (Newtons)
    float Tmin1, // Minimum required thrust (Newtons)
    float Tmax1 // Maximum required thrust (Newtons)
)
    // Constructor. Sets the initial values for the pulse modulator.
{
    // Copy the parameters to the local variables
    pin = pin1;
    ton = ton1;
    toff = toff1;
    U = U1;
    Tmin = Tmin1;
    Tmax = Tmax1;
    fmin = U * ton / (ton + toff); // Calculate the minimum pulse thrust
    gammak = 0; // Initialized to 0 to detect the first Control() execution
}

void PulseModulator::Control(
    float T // Required nozzle thrust
)
    // Performs the conversion from analog input to modulated digital output.
    // The first time is executed, initializes 'gammak', 'Tk', 'tson' and 'tsoff'.
    // Then checks if the ON or OFF period has passed and sets the digital pin
    // accordingly. This function must be called every time loop() cycles.
{
    t = millis(); // Retrieve the current time pointer

    // Limit T to the specified range
    if (T <= Tmin)
    {
        T = 0;
    }
    else if (T > Tmax)
    {
        T = Tmax;
    }

    // Initialize 'gammak', 'Tk', 'tson' and 'tsoff', but only the first time
    if (gammak <= 0)
    {
        // Pulse duration (gammak) and interval between starts of pulses (Tk)
        if (T >= fmin)
        {
            gammak = T * toff / (U - T);
            Tk = U * toff / (U - T);
        }
        else if (T == 0)
        {
            gammak = ton;
            Tk = ton + toff;
        }
        else
        {
            gammak = ton;
            Tk = U * ton / T;
        }

        // Time of the last switch on (tson)
        tson = t - Tk;
        // Time of the last switch off (tsoff)
        tsoff = t - Tk + gammak;

        digitalWrite(pin, LOW);
    }
}
else if (((t > tson) && (tson > tsoff))

    // Pulse duration (gammak)
    if (T <= fmin)
    
        gammak = T * toff / (U - T);
    
    else
    
        gammak = ton;

    if ((t - tson) <= gammak)
    
        digitalWrite(pin, HIGH);
    
    else
    
        digitalWrite(pin, LOW);

        tsoff = t;

    
    // Interval between starts of pulses (Tk)
    if (T >= fmin)
    
        Tk = U * toff / (U - T);
    
    else if (T <= 0)
    
        Tk = gammak + toff;
    
    else
    
        Tk = U * ton / T;

    if ((t - tsoff) <= (Tk - gammak))
    
        digitalWrite(pin, LOW);
    
    else
    
        digitalWrite(pin, HIGH);

        tson = t;

    }