Escola Politècnica Superior de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

TREBALL DE FI DE CARRERA

TÍTOLDEL TFC: Use of hardware-on-the-loop to test missions for a low cost mini-launcher

TITULACIÓ: Enginyeria Tècnica Aeronàutica, especialitat Aeronavegació

AUTOR: Laia Navarro Morcillo

DIRECTOR: JoshuaTristancho Martínez

DATA: 4 de març de 2011

Título: Use of hardware-on-the-loop to test missions for a low cost mini-launcher
Autor: Laia Navarro Morcillo
Director: Joshua Tristancho Martínez
Fecha: 4 de marzode 2011

Resumen

Este estudio está dirigido al uso de lazos de control por Hardware para probar subsistemas tanto de femto-satélites como mini-lanzaderas.

Usaremos el simulador Moon2.0 para implementar comunicaciones entre una caja negra y el simulador. Gracias a lo cual será fácil comprobar y validar cualquier dispositivo dentro del ciclo de diseño de la misión.

En este trabajo de final de carrera (TFC), tratamos de comprobar si este método es capaz de validar el proyecto de una forma económica el Wiki-Launcher, una mini-lanzadera que pesa menos de 100 kg. Al mismo tiempo probamos si las comunicaciones entre la caja negra y el simulador no afectan o no es significativo para ser usado de una forma segura. Un profundo estudio se presenta sobre la probabilidad de fallo como impacto medioambiental.

Dos ejemplos de lazo de control entre una caja negra y el simulador son presentados: el subsistema de control vectorial de la mini-lanzadera basado en una plataforma inercial y el subsistema de control de actitud basado en el campo magnético terrestre y unas bobinas.

Palabras clave: Hardware-on-the-loop, Low cost, Mini-launcher, Femto-satellite, Rotation matrix

Title: Use of hardware-on-the-loop to test missions for a low cost mini-launcher
Author: Laia Navarro Morcillo
Director: Joshua Tristancho Martínez
Date: March, 4th 2011

Overview

This study is addressed to the use of hardware-on-the-loop to test subsystems of both femto-satellite and mini-launchers.

We will use the simulator Moon2.0 to implement the communication between a blackbox and simulator. Thanks to this, it is easy to check and validate any hardware part inside the cycle of a mission design.

In this final bachelor work, we try to check if this method is able to validate the project in a low cost way, the Wiki-Launcher, a mini-launcher less than 100 kg. At the same time we try to check if the communications between the blackbox and the simulator doesn't affect or the implication is lower enough to be usedsafely. A deep study of fault probability is presented as an environmental impact.

Two examples of control loop between a black-box and the simulator are presented: the vector control subsystem for the mini-launcher based on an inertial platform and the attitude control subsystem for the femto-satellite based on Earth's magnetic field and few coils.

Keywords: Hardware-on-the-loop, Low cost, Mini-launcher, Femto-satellite, Rotation matrix

INDEX

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

We define the space payload paradigm as the engineering process of designing a space mission around its payload and not around space industry[1].

This paradigm[2] statesthat it is feasible to design a launcher and the satellite as a result of the desired mission but nowadays it is not feasible in the aerospace industry. For this reason, some initiatives had promotes the industry in order to make a cheaper and accessible space use.

This is the case of the so called N-Prize, a contest promoted by Paul Dear[2], a British biologist, in 2008. Inside the N-Prize there is a contestant called WikiSat that involve students and teachers in order to develop and implement an engineering solution like this.

Satellites and space vehicles are very often classified by mass as presented in table 1:

Category	**Mass range**
Large satellites	> 1000 kg
Medium-sized satellites	From 500 to 1000 kg
Small satellites	Less than 500 kg
Mini-satellite	From 100 to 500 kg
Micro-satellite	From 10 to 100 kg
Nano-satellite	From 1 to 10 kg
Pico-satellite	From 0.1 to 1 kg
Femto-satellite	Less than 100 grams

Table 1 Categories for mass classification

One popular category is Pico-satellites because of the so called **Cube-Sat** program[3]. No one uses a launcher to put less than one kilogram single cube-sat in orbit. WikiSat approach is to define a mission based on small satellites without depending on piggyback opportunities in the current launcher market. Current market is not prepared for so small satellites. This is why WikiSat, under N-Prize contest, wants to design and build a mini-launcher able to do so, becausethese private companies like FALCON[4] or DNEPR[5] try to launch small payloads but no so small like these.

A mini-launcher is a small launcher able to inject into the orbit femto and picosatellites. The main idea is to reduce the costs using synergies between the launcher and the payload. To do this we chose the open community approach; all the information is available and accessible to the community in such a way that anyone can participate.

[1]Roger Jové, *Technical constrains for a low cost femto-satellite launcher*, 48th AIAA Aerospace Sciences Meeting (January 2011)
[2]http://en.wikipedia.org/wiki/N-Prize(Feb/2011)
[3]http://en.wikipedia.org/wiki/Cubesat (Feb/2011)
[4]http://en.wikipedia.org/wiki/Falcon_1 (Feb/2011)
[5]http://en.wikipedia.org/wiki/Dnepr_rocket (Feb/2011)

In this final bachelor work, we try to check if this method is able to validate the project in a low cost way, the Wiki-Launcher, a mini-launcher less than 100 kg. At the same time we try to check if the communications between the blackbox and the simulator doesn't affect or the implication is lower enough to be usedsafely. A deep study of fault probability is presented as an environmental impact.

Two examples of control loop between a black-box and the simulator are presented: the vector control subsystem for the mini-launcher based on an inertial platform and the attitude control subsystem for the femto-satellite based on Earth's magnetic field and few coils.

In chapter 1 we introduce some basic concepts related to satellite or launcher motion such as control loop theory and moment of inertia calculation that will be used in chapter 4 and five.

In chapter 2theWikisat space program is presented: the relation between Wikisat and the N-prize, the presentation of our femto-satellite and mini-launcher and the subsystems of the satellite.

In chapter 3the Hardware-On-The-Loop is defined, protocols and scope of the control loop. Some details about how we implemented this loop inside the Moon2.0 simulator are discussed.

In chapter 4we present our second example using the Black box in the WikiSat attitude control subsystem.

In chapter 5 we present our first example using the Black box in the Wiki-Launcher vector control subsystem.

In chapter 6 there is a study about the effect of this mission for the environment. A study of fault probability is presented.

Finally, in chapter 7, some conclusions will be stated to summarize the work.

ADVICE: Wikipedia references were checked for reliability by the author.

Acknowledgements

First of all, I would like to thank WikiSat members for their collaboration: Joshua Tristancho, Juan Martínez, EsteveBardolet, Roberto Rodríguez, Victor Kravchenko, etc.

I am grateful to Daniel Crespo, who introduced me to the WikiSat group. I am grateful to other EETAC teachers that motivated me throughout my studies: Daniel Crespo himself, Jordi Gutiérrez, Santiago Torres, Santiago Arias, Dagoberto José Salazar, Fernando Bussalino, Lucas García, Jordi Mateu, Pilar Gil, Esteban Meca, Marcos Quílez, Oscar Casas, Jordi Corbera, Adeline de Villardi, etc. Again, Joshua Tristancho, because without him, this could not have been possible. Thanks for such a huge help even when it seemed there was no exit.

Furthermore, I am thankful to my colleagues during my internship in AENA. I was always learning specially from my tutor, David Rebollo,who was always willing to teach me. Thanks for giving me the chance and such a good working experience.

Also, I am grateful to my classmates, those who have become friends and those who helped so much: Esther Rodríguez, Neus Blanco, Axel Verderi, Blas Vaquero, BelaidOuharrou, Marta Lorente, VíctorSáez, Marc Herver, SergiJarabo, JordiGarcía, Moisés González, Ismael Romero, OriolBetriu, Montserrat Álvarez, Antonio Álvarez, Domingo Margareto, Sergi Palau, Enric Fava, Francisco Campón, etc.

I have to say thanks to my friends and my doctor, who always had words of encouragement to me.

A very special mention is for my parents because they have always been caring and, last but not least, my boyfriend, the one who has suffered the most and the one who has had the patience to support me stoically through all this time.

Finally I want to thank our sponsors and collaborators:

Figure 1 WikiSat partners

THE GLOSSARY

APCP	Ammonium perchlorate composite propellant
ASCII	American Standard Code for Information Interchange
cp	Centre of pressure
CPU	Central Processing Unit
DOF	Degrees of freedom
GPS	Global Positioning System
IMU	Inertial Measurement Unit
INTA	InstitutoNacional de Técnica Aerospacial (National Institute of Aerospace Technology, Spain's space agency)
MCU	Microcontroller Unit
MIMO	Multiple In Multiple Out control system
PC	Personal Computer
PID	Proportional Integral Derivative
UTC	Coordinated Universal Time
WGS 84	World Geodetic System dating from 1984 and last revised in 2004

CHAPTER 1. TECHNOLOGIES FOR BLACK BOX TECHNIQUE

In this chapter few technologies that will be used in chapter 4 and 5 are presented. These concepts are needed for the Black box technique control loop and they are:

- Control loops theory
- Moment of inertia calculation
- Rotation matrix

1.1 Control loops theory

Following any basic book[6] or web[20] related to PID, control loop allows us to obtain a desired output of a system thanks to its controller.The desired output of a system is called the reference. When one or more output variables of a system need to follow a certain reference over time, a controller manipulates the inputs to a system to obtain the desired effect on the output of the system.

There are two types of controllers[6]: open-loop controllers and closed loop controllers. An open-loop controller, also called a non-feedback controller, is a type of controller which computes its input into a system using only the current state and its model of the system.A characteristic of the open-loop controller is that it does not use feedback to determine if its output has achieved the desired goal of the input. This means that the system does not observe the output of the processes that it is controlling. Consequently, a true open-loop system can not engage in machine learning and also cannot correct any errors that it could make. It also may not compensate for disturbances in the system.

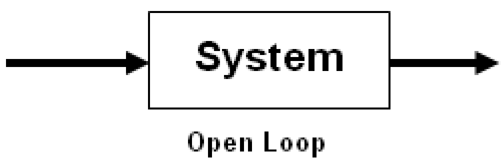

Figure 2 Open loop control system

On the other hand, in a closed-loop control system, a sensor monitors the system output and feeds the data to a controller which adjusts the control as necessary to maintain the desired system output. Feedbackallows the controller

[2]http://en.wikipedia.org/wiki/PID_controller(Feb/2011)

to dynamically compensate for changes. It is from this feedback that the paradigm of the control loop arises: the control affects the system output, which in turn is measured and looped back to alter the control.

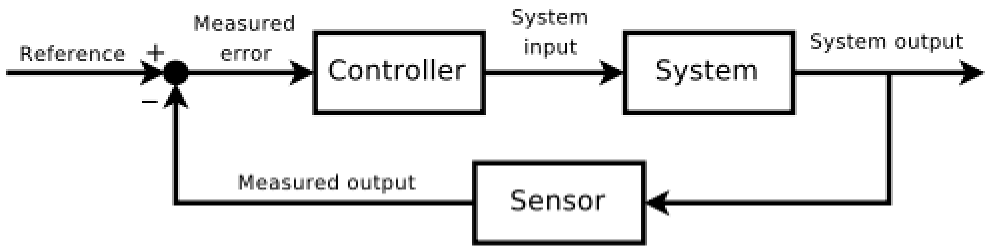

Figure 3 Concept of feedback loop to control dynamic behavior of systems

Closed-loop controllers have the following advantages over open-loop controllers:

- Disturbance rejection (such as unmeasured friction in a motor)
- Guaranteed performance even with model uncertainties, when the model structure does not match perfectly the real process and the model parameters are not exact
- U processes can be stabilized
- Reduced sensitivity to parameter variations
- Improved reference tracking performance

One of the most used closed-loop controller architecture is the PID controller. The PID controller calculation (algorithm) involves three separate parameters, and is accordingly sometimes called three-term control: the proportional, the integral and derivative values, denoted *P,I,* and *D.*Heuristically, these values can be interpreted in terms of time: *P* depends on the present error, *I* on the accumulation of past errors, and *D* is a prediction of future errors, based on current rate of change.The weighted sum of these three actions is used to adjust the process via a control element such as the position of a control valve or the power supply of a heating element.By tuning the three constants in the PID controller algorithm, the controller can provide control action designed for specific process requirements. The response of the controller can be described in terms of the responsiveness of the controller to an error, the degree to which the controller overshoots the setpoint and the degree of system oscillation.[20]

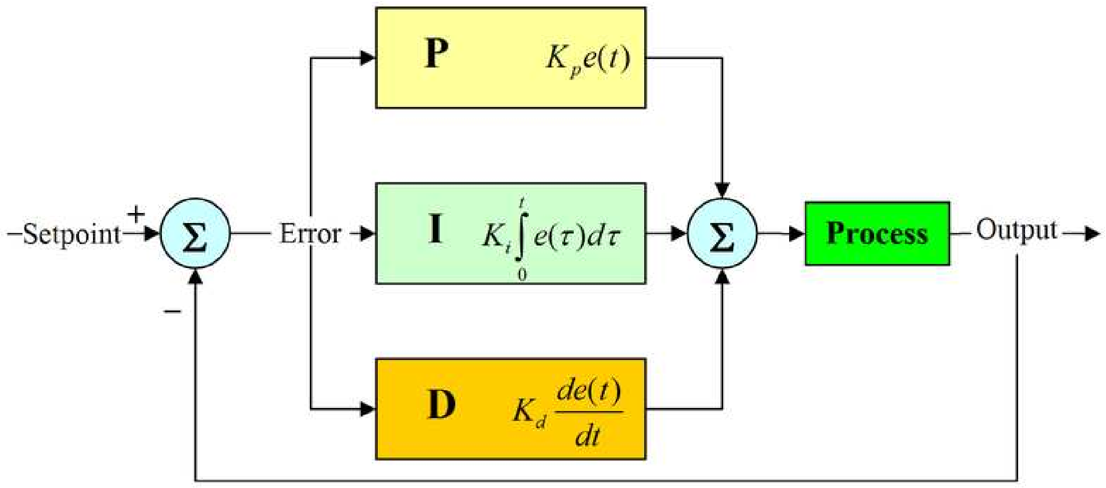

Figure 4 Block diagram of a PID controller

The function for a PID compensator in its Laplace transform form is:

$$G_C(s) = K_P + \frac{K_I}{s} + K_D s$$

Where:

- K_P: Proportional gain
- K_I: Integral gain
- K_D: Derivative gain

As a response of the system to a change from equilibrium, there is a transient response before reaching the steady-state, when the transients are no longer important.

Both proportional and integral controllers make the system become more unstable and improve the steady-state error although the first one is not able to eliminate it completely as the second one does. The price to pay for getting rid of the error is the system becomes slower. On the contrary, derivative controller adds stability and the system becomes faster. So, a proper combination of PID parameters allows an improvement of both transient and steady states.

Parameter	Peak time	Overshoot	Settling time	Steady-state error	Stability
P	Decreases	Increases	Small changes	Decreases	Degrade
I	Decreases	Increases	Increases	Eliminates (=0)	Degrade
D	Small changes	Decreases	Decreases	Small changes	Improve if K_D small

Table 2 Effects of increasing a parameter independently

These correlations may not be exactly accurate, because K_P, K_I and K_D are dependent of each other. Changing one of these variables can change the effect of the other two.Let us now change the values of these parameters one by one maintaining the other two constant in order to see their responses. A step function is used to do so.

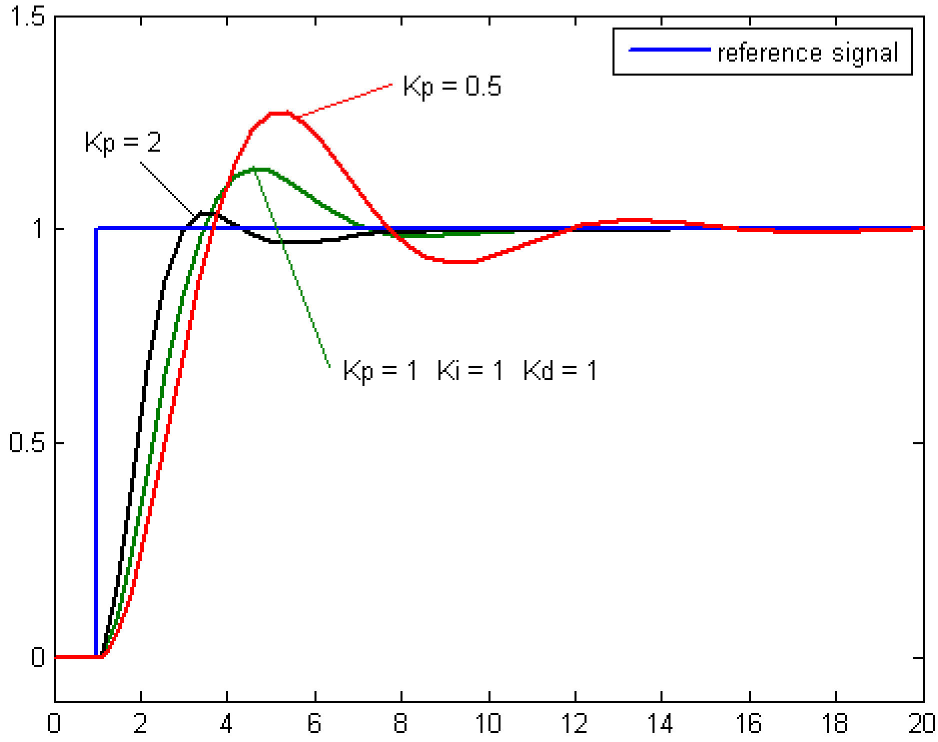

Figure 5 Plot for three values of K_p

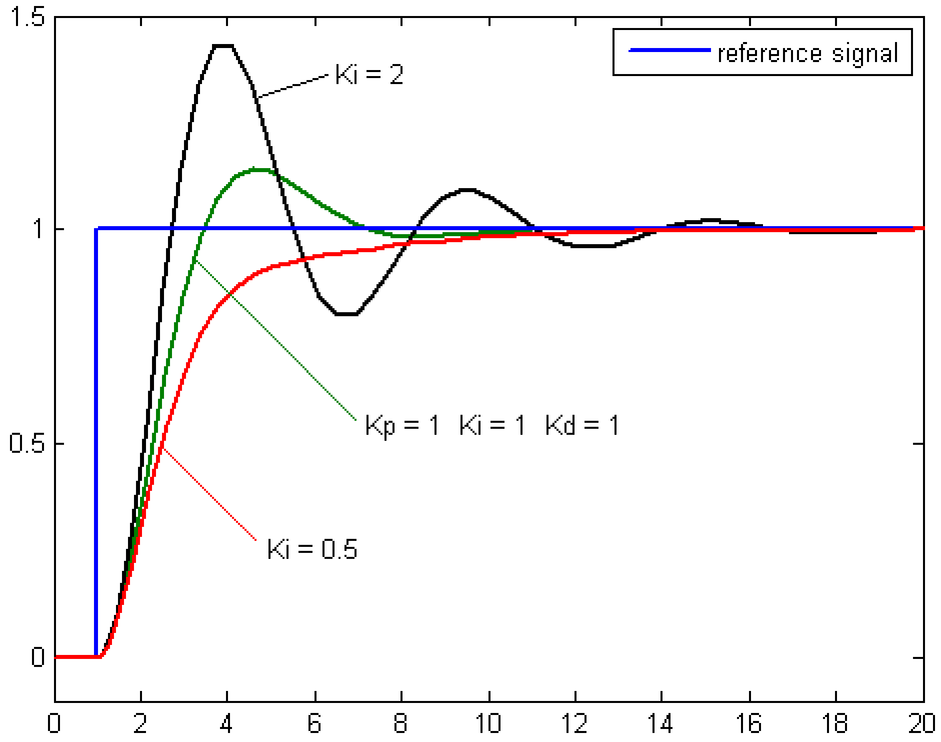

Figure 6 Plot for three values of K_I

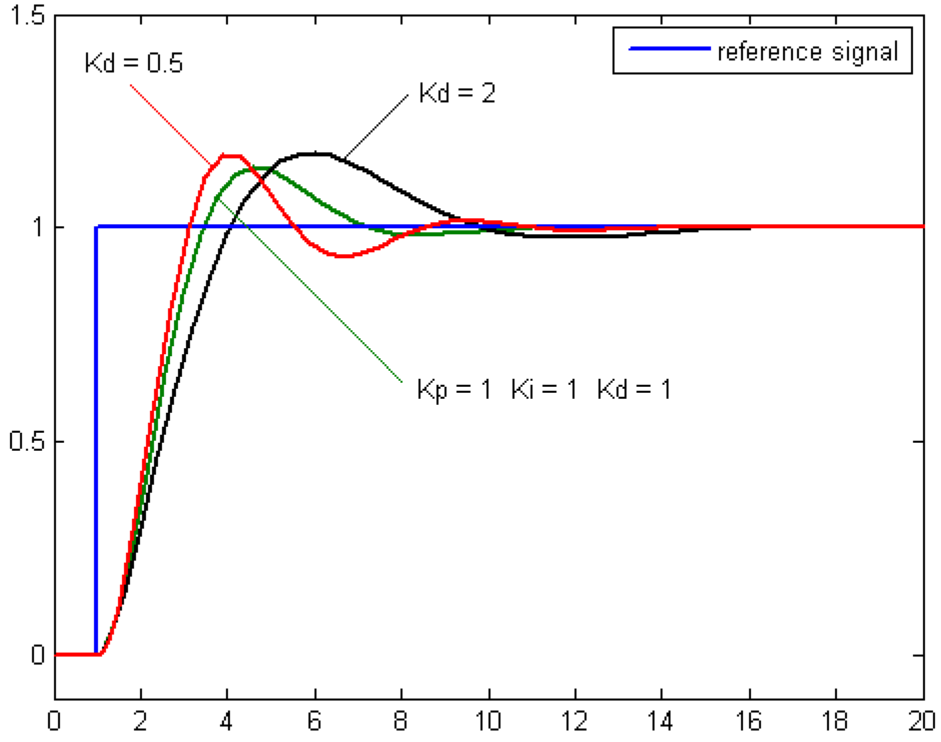

Figure 7 Plot for three values of K_D

1.2 Moment of inertia calculation

Moment of inertia (usually referred by the symbol *I*) is a measure of an object's resistance to changes to its rotation. Hence the moment of inertia of an object about a given axis describes how difficult it is to change its angular motion about that axis. It encompasses not just how much mass the object has overall, but how far each bit of mass is from the axis. This is easily deduced thanks to the formula:

$$I = \sum m_i r_i^2$$

Where *m* is the mass and *r* is the distance from the axis.

Similarly, the moment of inertia of a continuous solid body rotating about a known axis can be calculated by replacing the summation with the integral.

$$I = \lim_{\Delta m \to 0} \sum r^2 \Delta m$$

The value of the moment of inertia will be obtained when the portion of mass considered tends to zero, which will convert the addition in an integral.

$$\Delta m \to 0$$

$$I = \int r^2 \, dm$$

The differential can be described as:

- dm = ρdV for volumes
- dm = σdA for areas
- dm = γdx for wires

1.2.1 Geometrical calculation

Due to its shape, the satellite can be decomposed in two pieces. On one hand, the battery can be considered as a solid disk. On the other hand, the antenna can be considered as a rectangular plate. So, the moment of inertia will be calculated as a summation of their respective moments of inertia.

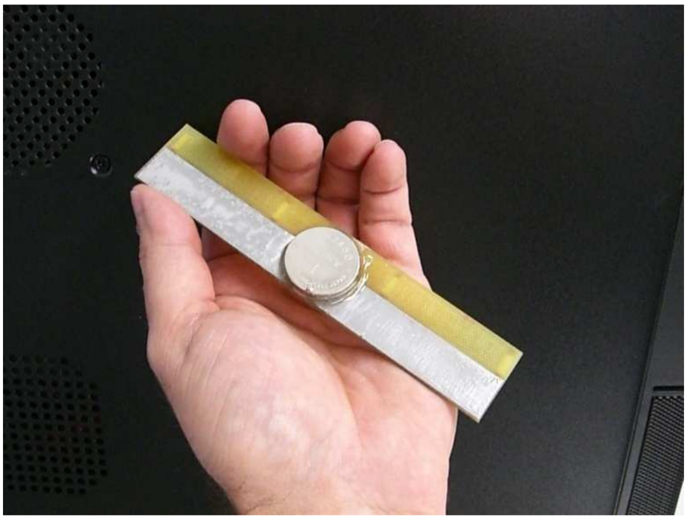

Figure 8 WikiSat v3 considered a solid disk plus a rectangular plate

Firstly, we focus on the solid disk. From the previous moment of inertia equation:

$$I = \int r^2\, dm$$

$$dI = r^2 dm$$

Now using the area density:

$$\sigma = \frac{M}{A} = \frac{M}{\pi R^2}$$

$$dm = \sigma dA = \sigma 2\pi r dr$$

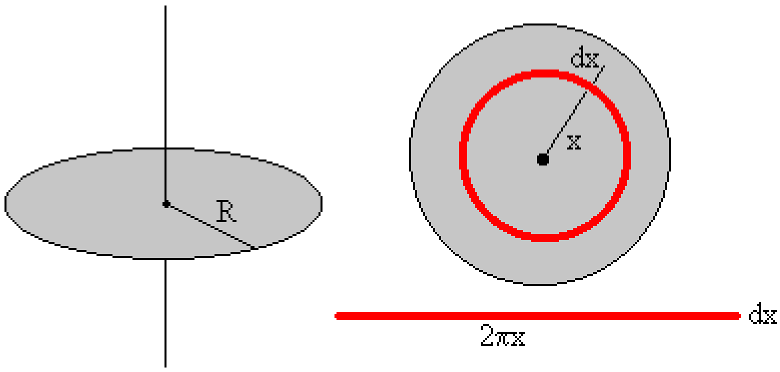

Figure 9 Solid disc

$$I = \int r^2 dm$$

$$I = \int r^2 \sigma 2\pi r dr$$

$$I = 2\pi\sigma \int_0^R r^3 dr$$

$$I = 2\pi\sigma \frac{r^4}{4}\Big|_0^R$$

$$I = 2\pi \frac{M}{\pi R^2}\frac{R^4}{4}$$

$$I = \frac{1}{2}MR^2$$

Secondly, we focus on the rectangular plate.

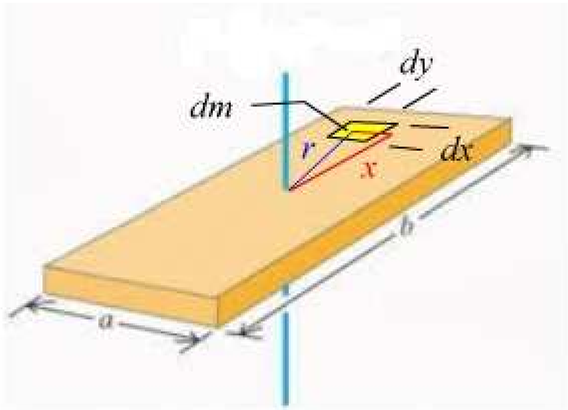

Figure 10 Rectangular plate

On the analogy of the preceding case, the next statements are deduced:

$$dm = \sigma dxdy$$

$$I = \int r^2 dm$$

$$I = \iint \sigma r^2 dxdy$$

$$r^2 = x^2 + y^2$$

$$I = \frac{1}{12}M(a^2 + b^2)$$

Finally, the total moment of inertia is calculated. To do so, an addition must be set out:

$$I_S = I_B + I_A$$

Where:

- I_S is the moment of inertia of the satellite
- I_B is the moment of inertia of the battery
- I_A is the moment of inertia of the antenna

Recall anterior equations in order to substitute, so:

$$I_S = \frac{1}{2} M_B R^2 + \frac{1}{12} M_A (a^2 + b^2)$$

The values for the satellite and the consequent result are shown in table 3:

	A	B	C
1	M_B	7,400E-03	kg
2	M_A	1,000E-02	kg
3	R	1,270E-02	m
4	a	2,600E-02	m
5	b	1,410E-01	m
6			
7	Is	1,773E-05	$kg \cdot m^2$

Table 3 Total moment of inertia of the satellite

1.2.2 Experimental calculation

In order to get an experimental value for the moment of inertia, software called DataStudio has been used. You plug in a sensor so that you can start data-collection. Then, graphs are shown and analysis can begin. Steps followed are described from now on.

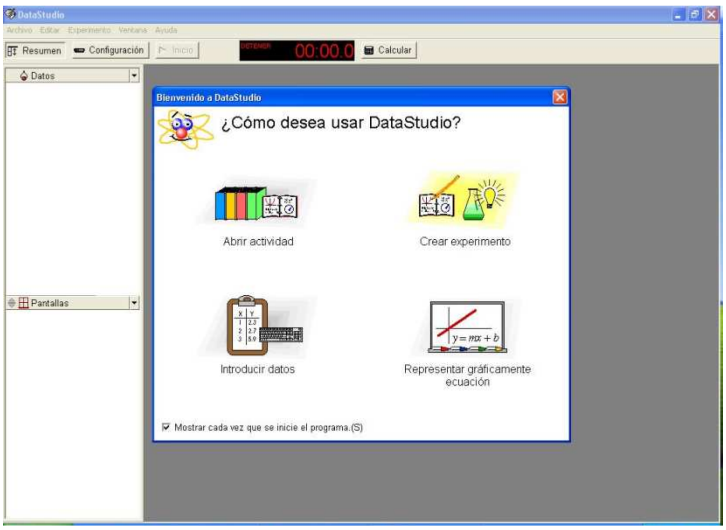

Figure 11 DataStudio welcome

This window appears when launching the software. “Crearexperimento” (new experiment) must be clicked and, as shown in the following figure, “Poleainteligente (lineal)” (Intelligent pulley - linear) must be selected.

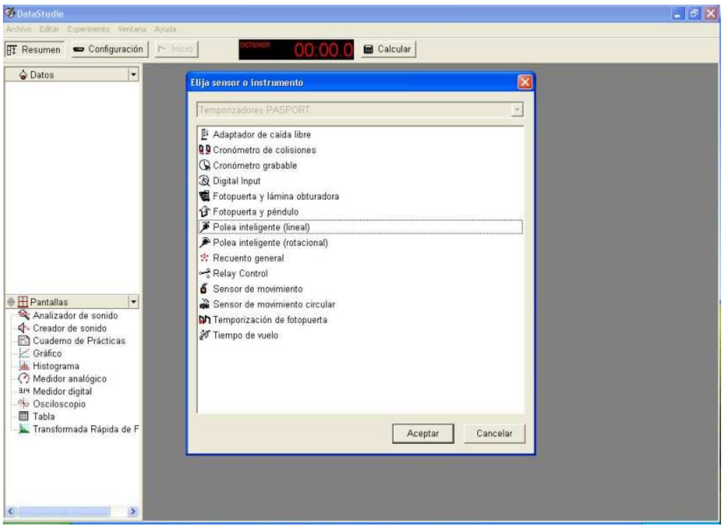

Figure 12 Intelligent pulley (linear)

Now, in the next window, everything is ready to start. So, the object to study, WikiSat in this case, has to have its center of mass placed properly on the 0 of the scale:

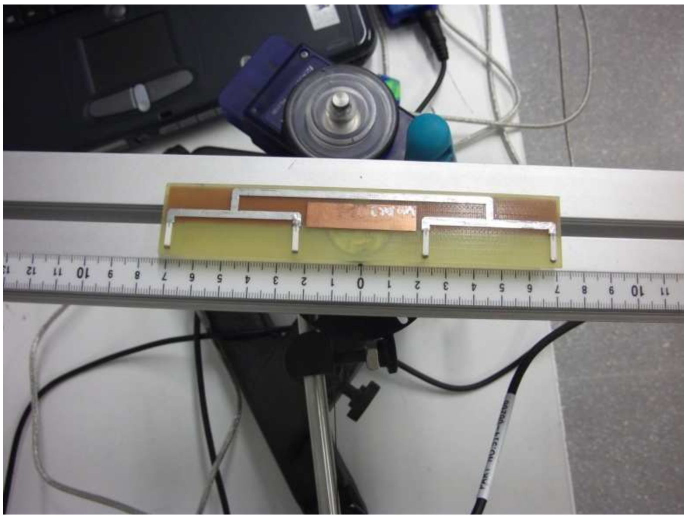

Figure 13 Properly placing WIkiSat on the scales

If the pulley is also ready, scale must be let go and, immediately, “Inicio” (Start) must be clicked so that the program is able to be drawing graphs while collecting data.

Figure 14 WikiSat and DataStudio ready to collect data.

Once the experiment is finished, “Ajustar ->Ajustecuadrático” (Adjust -> Quadratic adjustment) is selected, as shown in the next figure:

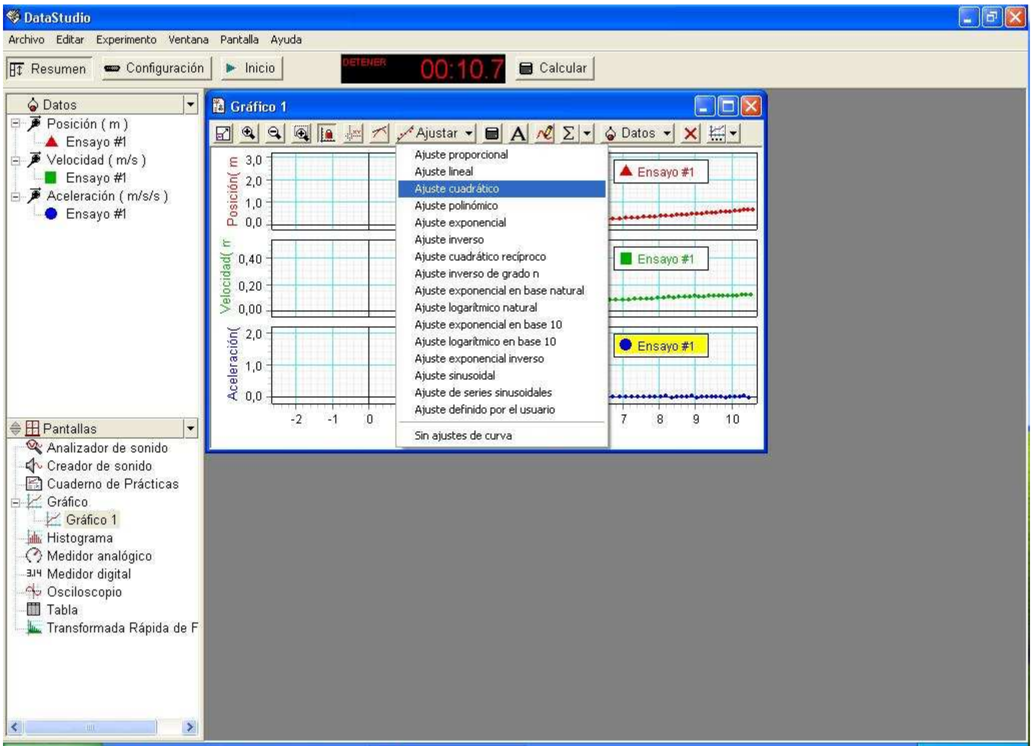

Figure 15 Quadratic adjustment selection

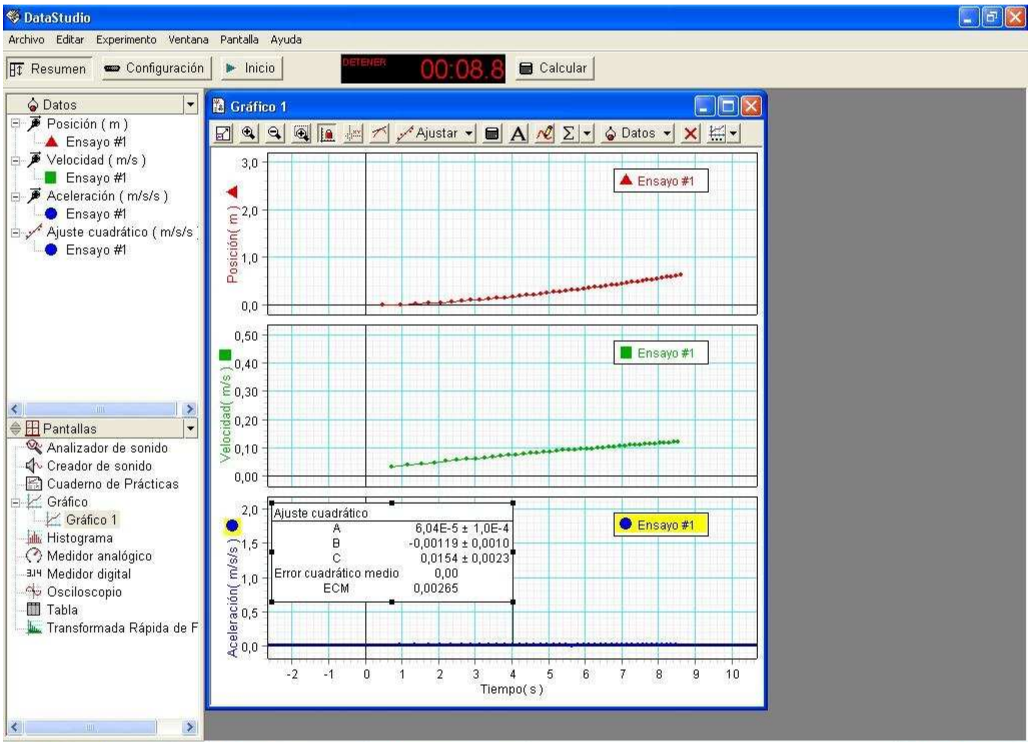

Figure 16 Graphs obtained after quadratic adjustment

For acceleration, A, B and C terms are obtained, as in the formula:

$At^2 + Bt + C$

Then:

$x = \frac{1}{2}at^2 + v_0 t + x_0$

So:

$A = \frac{1}{2}a$

As $A = 6{,}04 \cdot 10^{-5}$, then:

$a = 2A = 1{,}21 \cdot 10^{-4}\, m/s^2$

On the other hand, we know:

$$\tau = I\alpha \Longrightarrow Tr = I\frac{a}{r}$$

And:

$mg - T = ma$
$T = m(g - a) = 0{,}095kg(9{,}8\, m/s^2 - 1{,}21 \cdot 10^{-4}\, m/s^2) = 0{,}93N$

Then:

$$I_{total} = \frac{Tr^2}{a} = \frac{0{,}93N(12{,}96 \cdot 10^{-3}m)^2}{1{,}21 \cdot 10^{-4}\, m/s^2} = 1{,}29kg \cdot m^2$$

Where:

- $r = 12{,}96 \cdot 10^{-3}\, m$: Distance from the point of application of tension force T to the centre of the axis.
- $g = 9{,}8\, m/s^2$: Earth's gravitational acceleration.
- $m = 0{,}095\, g$: Mass used to be hanged from the pulley.

Now we need to subtract the momentum of inertia of the scales. So, the experiment must be done again without any piece or element on the scales.

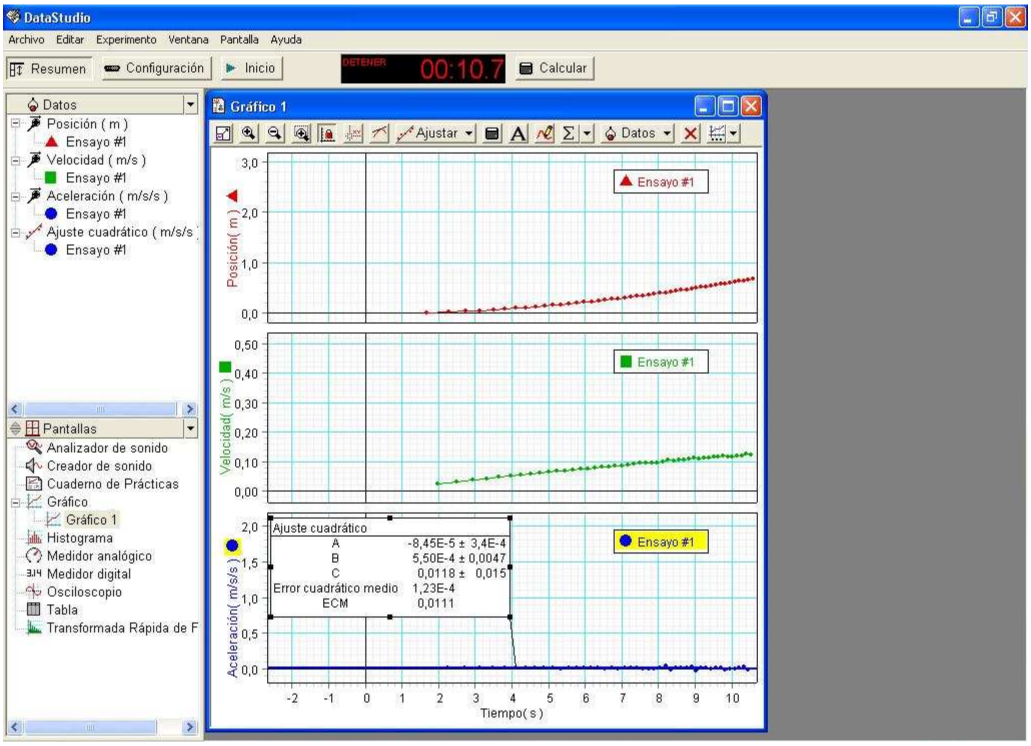

Figure 17 Graphs for only the balance

Same process as before, but with A=8,45·10^{-5}m/s^2in absolute value, will give the moment of inertia to subtract:

$$I_{scales} = 0{,}93\, kg \cdot m^2$$

$$I_{v3} = 1{,}29 - 0{,}93 = 0{,}36\, kg \cdot m^2$$

Finally, moments of inertia of prior versions of WikiSat have been measured. Results for WikiSat v1 are shown first, followed by results for WikiSat v2.

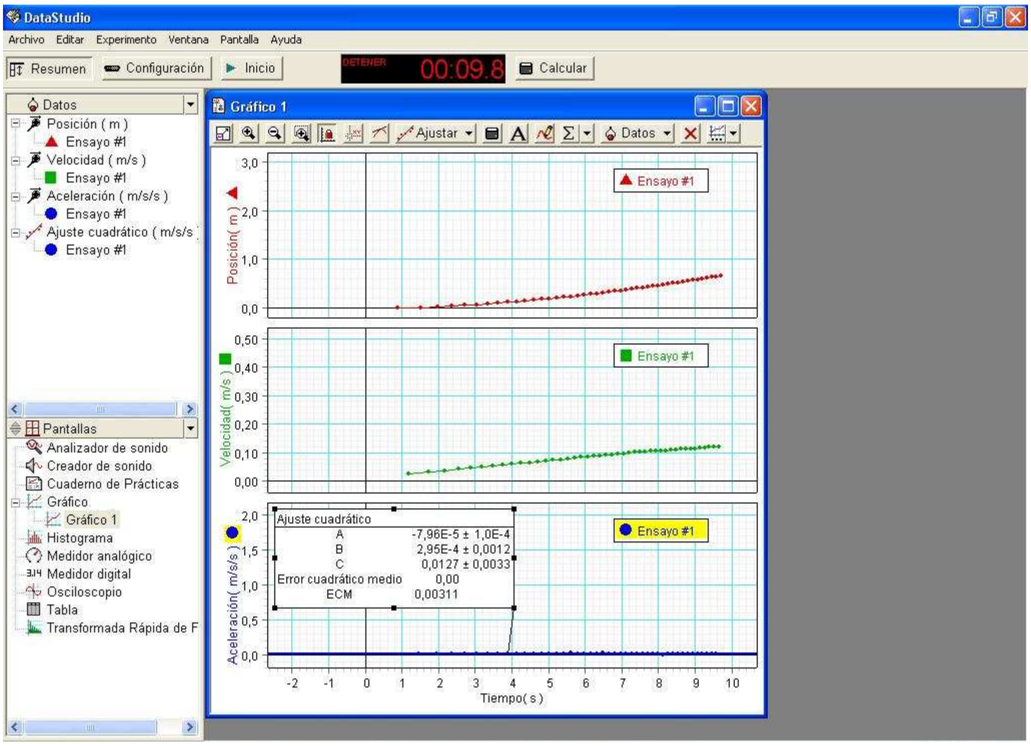

Figure 18 Graphs obtained for WikiSat v1

As it was done previously, the moment of inertia is calculated:

$A_{v1} = 7{,}96 \cdot 10^{-5}\, m/s^2$(Absolute value)

$I_{v1} = 0{,}98 - 0{,}93 = 0{,}05\; kg \cdot m^2$

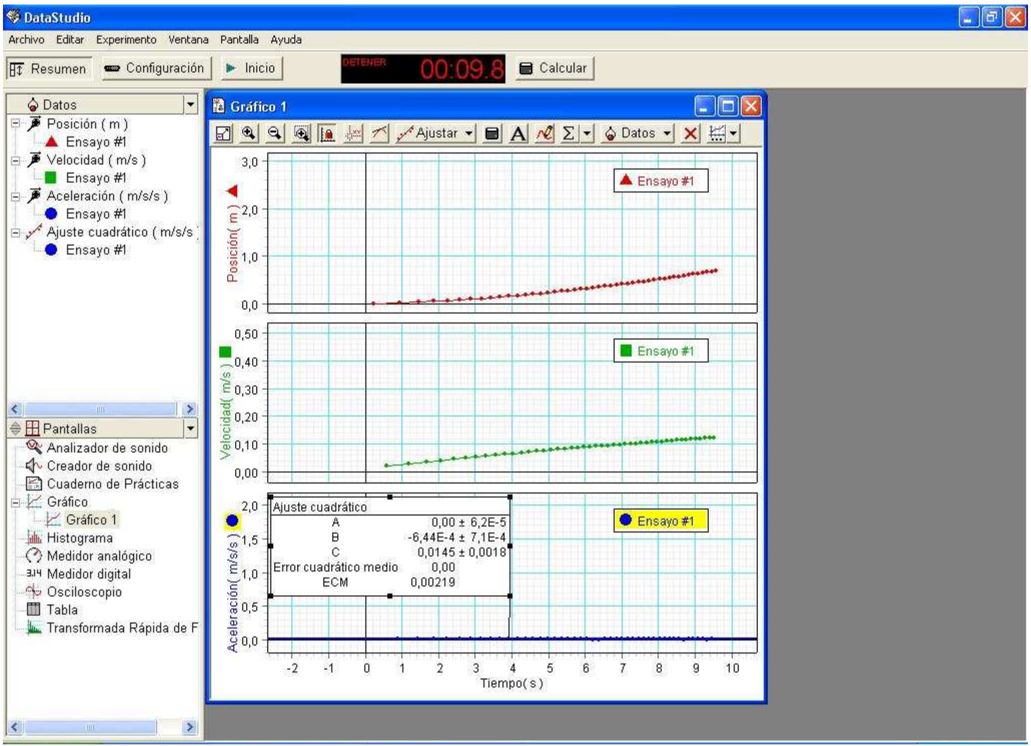

Figure 19 Graphs obtained for WikiSat v2

The term A for this last case equals 0. This fact plus the huge difference between experimental and theoretical results lead to the conclusion that the scales resolution is not precise enough to measure such tiny values. Acceleration is too small to use this kind of device in order to work with it.

1.3 Rotation Matrix

It is necessary to change from a system to another around coordinate axes because, at a moment, the position of the satellite is represented in the system but, later, this position will have changed due to its orbital movement and also the Earth would have rotated. So, there will be a new system that is not the initial one.

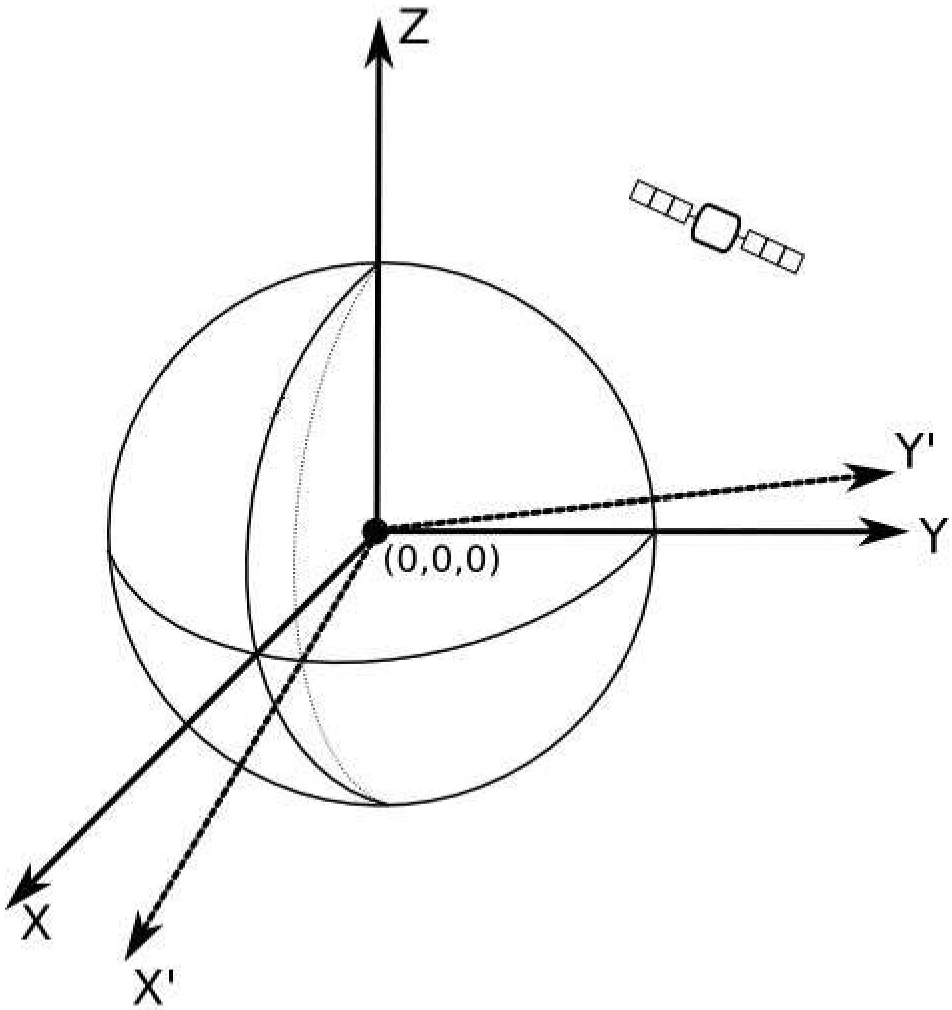

Figure 20 System rotation

A system common to both satellite and Earth is required. If time and Earth rotation speed are known, the angle between both systems is also known. Notice there is rotation just around Z axis.

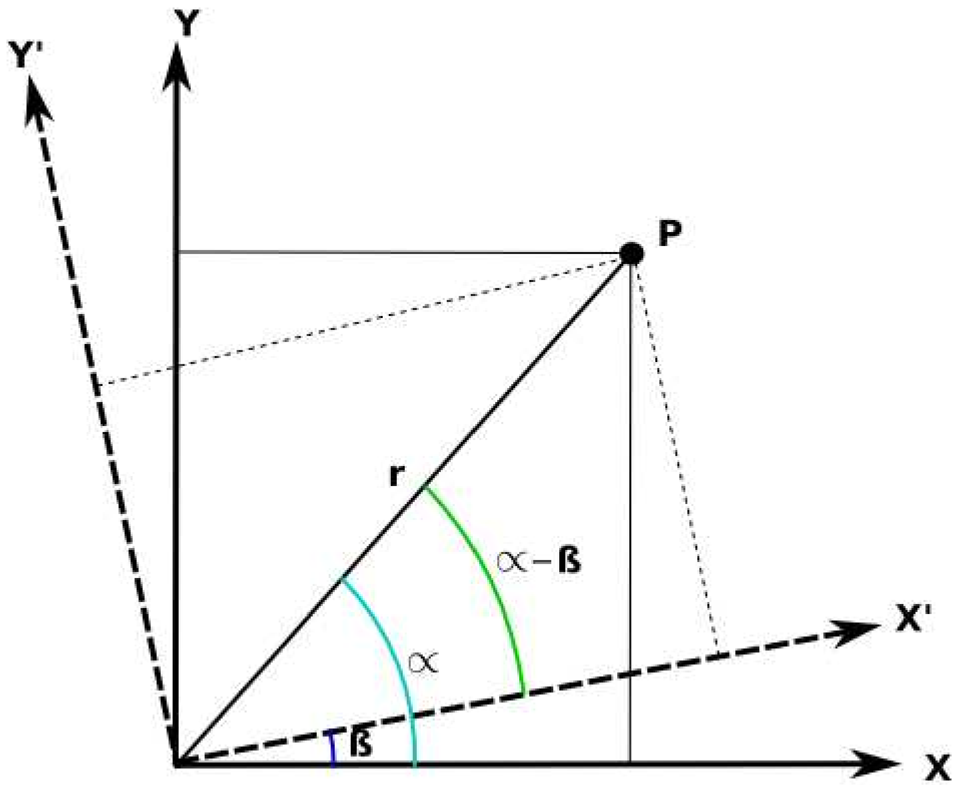

Figure 21 Rotation around the Z axis.

Distance *r* from origin to point *P* is the same for both systems. So, their respective coordinates can be written as:

$x = r\cos\alpha$

$y = r\sin\alpha$

$x' = r\cos(\alpha - \beta)$

$y' = r\sin(\alpha - \beta)$

Taking into account difference angle expressions:

$x' = r\cos(\alpha - \beta) = r(\cos\alpha\cos\beta + \sin\alpha\sin\beta)$

$x' = r\cos\alpha\, cos\beta + r\sin\alpha\sin\beta$

$x' = x\cos\beta + y\sin\beta$

$$y' = r\sin(\alpha - \beta) = r(\sin\alpha\cos\beta + \cos\alpha\sin\beta)$$

$$y' = r\sin\alpha\, cos\beta + r\cos\alpha\sin\beta$$

$$y' = -x\sin\beta + y\cos\beta$$

Now the following matrix can be written:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{pmatrix} \cos\beta & \sin\beta \\ -\sin\beta & \cos\beta \end{pmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

This expression states rotation in two dimensions. To obtain the tridimensional expression it is only necessary to know that, in this case, rotation is exclusively around Z axis, so z coordinates remain constant. Then, rotation around Z is:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{pmatrix} \cos\beta & \sin\beta & 0 \\ -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Following a similar process, rotation expressions around X and Y axes are possible to deduce. Since coordinate axes X, Y and Z are often notated as 1, 2 and 3respectively, corresponding rotation matrices are called R_1, R_2 and R_3.

$$R_1(\beta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\beta & \sin\beta \\ 0 & -\sin\beta & \cos\beta \end{pmatrix}$$

$$R_2(\beta) = \begin{pmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{pmatrix}$$

$$R_3(\beta) = \begin{pmatrix} \cos\beta & \sin\beta & 0 \\ -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Convention for β sign is very important. The angle will be positive when rotation around the axis is in the right hand rule sense. That is why in figure 10 rotation was considered positive.

Generally, rotation from a coordinate system *p* to another *q* can be expressed as:
$x_q = R_p^q x_p$

Where x_q is the vector defined in *q* system,$x_p$is the vector defined in *p* system and R_p^q is rotation matrix from *p* to *q*.

Rotation matrices defined here are orthogonal, so they have quite important properties:

- If R_p^q converts from *p* to *q*, then R_p^{qT} converts from *q*to *p* ($R_p^{qT} = R_p^q$).
- Determinant equals 1 ($\det\left(R_p^q\right) = 1$).
- $R_p^{qT} R_p^q = R_p^q R_p^{qT} = I$, where *I* is identity matrix.
- $R_p^{q-1} = R_p^{qT}$
- Any rotation from *p* to *q* is possible by means of successive rotations around 1, 2 and 3 axes, in that order:

  $$R_p^q = R_3(\beta_3) R_2(\beta_2) R_3(\beta_3)$$

1.3.1 Rotation matrix for small angles

Sometimes rotation angle can be small, this is, less than 10º. In this case, it is possible to take advantage using the following approximations:

$$\sin \epsilon \simeq \epsilon$$

$$\cos \epsilon \simeq 1$$

It is important taking into account that these approximations are only valid for angles expressed in radians.

Then, rotation matrices could be simplified as follows:

$$R_1(\beta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & \beta \\ 0 & -\beta & 1 \end{pmatrix}$$

$$R_2(\beta) = \begin{pmatrix} 1 & 0 & -\beta \\ 0 & 1 & 0 \\ \beta & 0 & 1 \end{pmatrix}$$

$$R_3(\beta) = \begin{pmatrix} 1 & \beta & 0 \\ -\beta & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

[5]

CHAPTER 2. WIKISAT SPACE PROGRAM

2.1 Wikisat Project

2.1.1 N-Prize contest

It is a competition to stimulate innovation towards obtaining cheap access to space. According to the rules, the objective is to put into orbit around the Earth a satellite with a mass between 9.99 and 19.99 grams, and to prove that it has completed at least 9 orbits. No part of any orbit may be lower than 99.99 km above the surface of the earth.

The cost of the launch must fall within a budget of £999.99.

A prize of £9,999.99 will be awarded to the first group to achieve it.

The WikiSat group is formed by teachers, students and collaborators. We are developing afemto-satellite called WikiSat and a mini-launcher for this satellite called WikiLauncher to participate in the N-prize.

2.1.2 WikiSat: The femto-satellite

The satellite WikiSat will be the brain in our mission. The idea is that the payload has the control of the launcher and takes predefined decisions when needed. This is one of the group statements related to the satellite. For precaution, the satellite will not have receiver. The WikiSat is being designed to fit N-Prize rules.

One of the most restrictive parameters in the satellite will be the weight. A non redundancy policy has been applied and the system must be Single Fault Tolerant.

2.1.3 WikiLauncher: The mini-launcher

The WikiLauncher will be controlled by the WikiSat. There is no available prototype yet but the design is mainly decided. It will use multinozzles approach and it won't have fins. A GPS will be used in the mini-launcher but not in the satellite.

Current simulations says that will be able to carry a 20 g payload to an altitude of 250 km for 8 days with a launcher of 1.5 m of length and 34.52 kg of weight, using APCP (ammonium perchlorate composite propellant) as a combustible.[3]

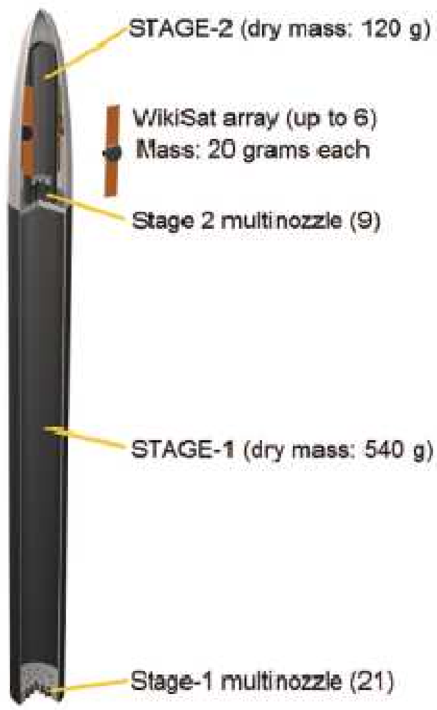

Figure 22 WikiLauncher and the WikiSat

2.2 WikiSat Satellite

2.2.1 Performances

Time in orbit: From one week to one month
Orbit: LEO 250 km
Communications range: 200 to 1,000 km
Mass: 20 grams

2.2.2 Subsystems

Because of the mass restrictions we cannot equip our satellite with a lot of subsystems. There is a minimum of required subsystems to accomplish its missionand those subsystemsare:

- Power Supply subsystem: Power cell and LiPoly battery
- Communication subsystem: S-Band full-duplex
- Structure subsystem: Glass fiber structure
- Attitude determination subsystem: 3 Gyros
- Position determination subsystem: 3 Accelerometers + Kalman filter
- Attitude control subsystem: 4 Magnetorquers + MCU computer

- Tracking subsystem: UHF band transceiver

- Payload subsystem: High Definition sensor

2.3 Prototypes

2.3.1 *Prototype 1*

On Monday, October 19th, 2009, the WikiSatprototype 1 was launched in a high altitude balloon at 04:00 am. This balloon was never recovered. This version was used to validate the coin battery, the initial IMU and the MCU through the flight telemetry.

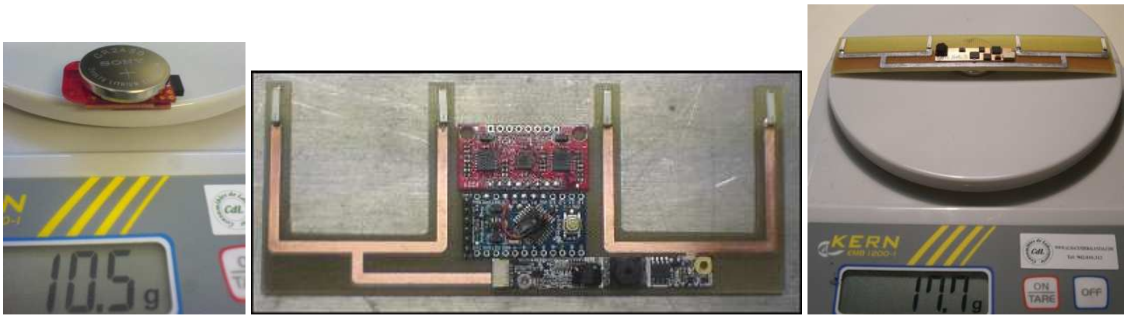

Figure 23 WikiSat prototypes

2.3.2 *Prototype 2*

The prototype is based on commercial boards. It uses:

- 6 DOF (Degrees Of Freedom) Razor – Ultra -Thin IMU from *SparkFun*.
- Arduino Pro Mini 168 from *SparkFun*.
- Transceiver nRF2401A with Chip Antenna from *Sparkfun.*
- Coin Lithium battery

2.3.3 *Prototype 3*

This is the current design where is installed our inertial platform to control the launcher and the satellite. This design includes the High Gain Antenna, the Coin battery, the IMU and MCU.

CHAPTER 3. HARDWARE-ON-THE-LOOP

Hardware-on-the-loop is used to test subsystems of both femto-satellite and mini-launchers.

The simulator Moon2.0 is used to implement the communication between a blackbox and a simulator. It is easy then to check and validate any hardware part inside the cycle of a mission design.

3.1 Control loop protocol

Moon2.0 is an open source Windows based application. It is a launcher simulator first created to simulate trajectories for the team FREDNET's mini-launcher for the Google Lunar X PRIZE. Now it is also used to simulate the trajectories of the WikiLauncher for the N-prize.

Tests have been made using a RS-232 connection and the following protocol. Either serial or socket connection uses ASCIItext. The protocol is the same in both options and there are coma separated numbers ("double precision" type) where decimal character is ".", negative values character is "-", optional positive values character is "+", exponential symbol is "e" or "E" and not a number characters are "NaN". The sequence ends with a zero code to be compatible with C++ strings.

Each block is 512 bytes in size and starts with a code of four characters:

- INIT: To initialize the Black box state
- DATA: To feed the Black box for a given state and
- CTRL: As the Black box command.

3.1.1 Black box initialization sequence (INIT)

When beginning a session, the Moon2.0 simulator will send an initialization sequence. The content of this sequence is variables that will not change during the loop but they are required by the Black boxlike constants or contour conditions.

```
INIT. This block is sent when T=0 and could be used by the black box as areset command.
Configuration variables from the simulator are "doubleprecision" type and the syntax is:

INIT,TimeStep,Date,Time,Lat,Lon,Alt,Speed,Heading,StartAngle,EndAngle,StartAlt,EndAlt,Ab
ort,TotalMass,Stages,StageNMass,StageNDry,StageNThrust,StgeNDelay,PayloadMass

Example:
INIT, .05,2007-06-26,12:09:00, 28.2,-16.6, 0, 409.895238938115, 87.2, 15, 88, 15000,
91000,
4, 34.7, 2, 31.212, 3.1212, 567.151258333333, 121, 3.468, .3468, 57.0093253333333, 3,
.02
```

TimeStep: Time step of simulation loop in seconds.

Date, Time: Launch date and time in UTC format: YYYY-MM-DD, HH:MM:SS.

Lat, Lon, Alt: Launch geo-coordinates using the WGS84 system indegrees and altitude in meters respect to average sea level.

Speed: Relative initial speed respect to the planet's center with planet's rotation in meters per second.

Heading: Clockwise launch direction where Northing is 0 degrees. (0=North, 90=East).

StartAngle, EndAngle: Start and final angle of the launch trajectory where vertical (azimuth) is 0 degrees. (0=vertical, 90=horizontal).

StartAlt, EndAlt: Start and final altitude check point of the launch trajectory in meters.

Abort: Number of stages before abort.

TotalMass: Wet launcher mass in kilograms.

Stages: Number of stages without the hover and the payload. The StageNparameters are repeated for every stage.

StageNMass: Wet mass of stage N in kilograms.

StageNDry: Dry mass of stage N in kilograms.

StageNThrust: Vacuum thrust of stage N in Newtons.

StageNDelay: Delay after stage N in seconds.

PayloadMass: Payload mass in kilograms.

3.1.2 Black boxfeeding for a given state (DATA)

Each loop, the Moon2.0 simulator will send the variables that defines the simulation state. The content of this sequence are variables required by the Black box like the time, environmental parameters, etc.

```
DATA. This block is sent each simulation loop. Input variables for the blackbox
are "double precision" type and the syntax is:

DATA,T,M,Mt,Mf,F,Te,eG,eH,eX,eY,eZ,eA,eR,eVR,eVRX,eVRY,eVRZ,eVA,eVAX,eVAY,eVAZ

Example:
DATA,    0,    34.7,    34.7,    6.6092,    0,    .222435897435897,    9.80060472243182,
6378137,968534.704408334,6278974.38685793,563074.387898972,9.31322574615479E10,9.3132257
4615479E10,409.895238938115,403.764758270368,64.7999721286988,28.0907546200786,1.4375495
1773129E12,5.04984942750752E-13,-1.22130777713281E-12, 5.65637814364806E-13
```

T: Time since the launch in seconds.

M: Current launcher mass in kilograms.

Mt: Current launcher wet mass without burned stages in kilograms.

Mf: Current launcher dry mass without burned stages in kilograms.

F: Current thrust in Newtons as a function of the thrust curve and the atmospheric pressure.

Te: Propellant consumption at full thrust in kilograms per second.

eG: Current planet's gravity in meters per second squared.

eH, eX, eY, eZ: Current altitude respect to the planet’s center and its X, Y, Z components in meters.

eA: Current altitude respect to the sea level in meters.

eR: Current height respect to the ground in meters. This parameter is valid only when the altitude to the surface is less than one planet radius and the option “matrix terrain” is active in the simulator.

eVR, eVRX, eVRY, eVRZ: Current relative speed respect to the planet’scenter and its X, Y, Z components in meters per second.

eVA, eVAX, eVAY, eVAZ: Current relative speed respect to the planet’satmosphere and its X, Y, Z components in meters per second.

3.1.3 Black box commandment (CTRL)

Each loop, as a response of each Moon2.0 simulator iteration, the Black box will send the commandment parameters included the time stamp that synchronizes the loop.

```
CTRL. This block is expected by the simulator to be replied before the “timeout”
interval, if not then it goes to PAUSE. Output variables from the black box for attitude
control are “double precision” type and the syntax is:

CTRL,Tt,St,Nt,Rt,Ut,Vt,Wt

Example:
CTRL,0, 2, 1, 1, .151852289219914, .984452730767299, 8.82819525355084E-02
```

Tt: Time since the launch in seconds to check the correct packet.

St: Current mission phase: Crashed=-2, Landed=-1, Aborted=0, Launch=1, TakeOff=2, Transfer=3, Landing=4, Revert=5, Satellite=6, StandBy=7. NOT YET IMPLEMENTED

Nt: Active stage (1, 2, … N). NOT YET IMPLEMENTED.

Rt: Thrust control from 0=stopped to 1=full. NOT YET IMPLEMENTED.

Ut, Vt, Wt: Attitude control. It must be a normalized vector, that is to say

Ut*Ut+Vt*Vt+Wt*Wt=1, in the simulator reference system that is aligned with the ecliptic plane (the plane XY where the Earth orbits around the Sun). Tosee the simulator X, Y, Z axes in the 3D viewer you may select the tool“Tools->Axes”.

3.2 Source examples

Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. The hardware is a single-board microcontroller with an Atmel AVR microcontroller. The connectors are exposed allowing the CPU board to be connected to interchangeable modules.

We will present some examples in this platform to show how to program a Black box.

Some versions of original Arduino hardware have been commercially produced but the designs are distributed under Creative Commons license. We have got the components and the technology to build our own*Arduino*.

We have purchased a commercial Arduino Pro Mini but, learning a little about the processor, it was decided to redesign it as an MCU that exploits all the capacities of Atmel processor.

The Arduino software can be downloaded for free in the Arduino’s website[7]. It is an application written in Java. It includes a code editor and it is easy to compile and upload.

In order to establish communication, it has been used Windows HyperTerminal. It is very important to select the communication port where the device is connected as well as the baud rate. Once everything is ready, tests can begin.

The Arduino code finally written to achieve communication between simulator and black box is the following:

```
/*
 * Ejemplo de satélite que usa Hardware-on-the-loop
 * http://wiki.teamfrednet.org/index.php/Lunar_Bus_Trajectory
*/

intledPin = 13;      // select the pin for the LED
int i;
char a[512];
intrecibiendo;
int t;

void setup()
{
pinMode(ledPin, OUTPUT);
  // Serial
Serial.begin(9600);
  //Serial.println("SATELLITE READY");
  i=0;
  t=0;
recibiendo=false;
```

[7]http://www.arduino.cc/ (Feb/2011)

```
}

void loop()
{
intm,n,o;
if(recibiendo)
{
digitalWrite(ledPin,HIGH);
if(Serial.available()>0)
    {
a[i]=Serial.read();
i++;
    }
if(i>=512)
    {
for(n=0;(a[n]!=',')&&(m<512);n++);
for(m=n+1;(a[m]!=',')&&(m<512);m++);
      i=0;
t++;
if(t>1)recibiendo=false;
    }
  }
else
  {
digitalWrite(ledPin,LOW);
Serial.print("CTRL, ");
i=6;
for(o=n+2;(o<m)&&(o<512);o++)
{
Serial.print((char)a[o]);
i++;
    }
Serial.print(", 0, 2, 1, 1, .151852289219914, .984452730767299, 8.82819525355084E-02");
    i+=70;
for(o=i;o<512;o++)
    {
Serial.print((char)65);
i++;
    }
recibiendo=true;
while(1);
  }
}
```

By means of this code, it has been possible to check if a led turns on and off. That proves communication exists and it is working properly.

Moreover, it has been also possible to get a block. Here, it is possible to visualize the first part of the block included in annex A[8]:

```
INIT, .05,2007-06-26,12:09:00, 28.2,-16.6, 0, 409.895238938115, 87.2, 15, 88, 15000,
91000, 4, 34.7, 2, 31.212, 3.1212, 567.151258333333, 121, 3.468, .3468, 57.0093253333333
0093253333333, 3, .02

DATA, 0, 34.7, 34.7, 6.6092, 0, .222435897435897, 9.80060472243182, 6378137,-
968534.704408334,-6278974.38685793,-563074.387898972, 9.31322574615479E-10,
9.31322574615479E-10, 409.895238938115,-403.764758270368, 64.7999721286988,-
28.0907546200786, 1.43754951773129E-12,-5.04984942750752E-13,-1.22130777713281E-12,
5.65637814364806E-13

CTRL, 0, 2, 1, 1, .151852289219914, .984452730767299, 8.82819525355084E-02
```

It is clear that blocks obtained correspond with blocks expected, those described previously as INIT, DATA and CTRL in the Moon2.0 protocol.

Another code was implemented in order to receive data. A problem related to the type of variable was detected so, the main object of the following code is to find out whether if we are working with float or double.

Float test:

[8] Annex A, page 78.

```
char a[512];
int i;

void setup()
{
Serial.begin(115200);
  i=0;
}

void loop()
{
if (i==0)
  {
    i=1;
float x=9.117621E+11;
Serial.write((uint8_t *)&x, 4);
Serial.println("FLOAT");
double y=9.84001179603715E+25;
Serial.write((uint8_t *)&y, 8);
Serial.println("DOUBLE");
  }
}
```

Next, two figures are shown implementing the code. First of all, we comment the part of the code referred to floats so that it does not interfere with the other part, the one of interest.

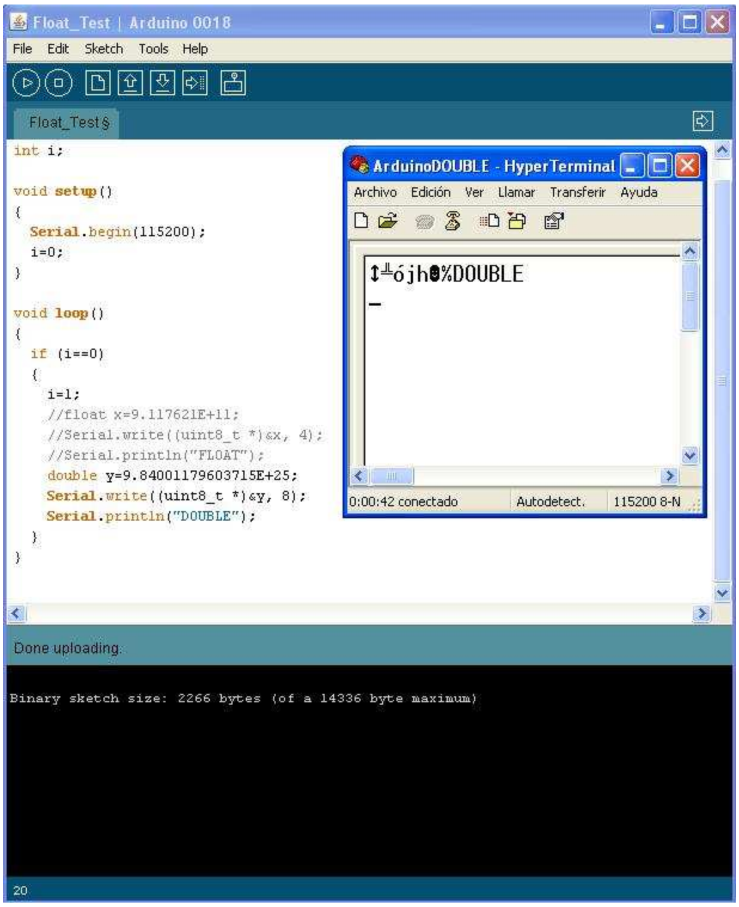

Figure 24 Variable type test to prove double

It was expected to obtain the word “BYTEDOUBLE”, but that did not come out. The reason is double variables in Arduino occupies 4 bytes, while in the PC occupies 8 bytes.

Let us see what happens on the contrary. We now comment the double part in order to check the proper functioning.

Figure 25 Variable type test to prove float

In this case, the exact word expected "BITSFLOAT" was obtained. Float variables in Arduino occupies 4 bytes, as PC does.

Then, we can conclude the double implementation on the Arduino is currently exactly the same as the float, with no gain in precision.

3.3 Documentation

The following step was to write the proper documentation that was included since the version 2.0.100712.1007. You can see any version in the official Moon2.0 web page[9].

A section called *REMOTE CONTROL* was written. Three ways or methods to control the launcher were presented: *Software-on-the-loop*, *Hardware-on-the-loop* and *Network-on-the-loop*.

[9]http://code.google.com/p/moon-20/downloads/list (Feb/2011)

- Software-on-the-loop: This loop is used in the initial guess because it is really easy to implement an algorithm internally in the code. The time effect is not considered and only functionality is taken into account.
- Hardware-on-the-loop: This loop is used to test some aspects of the hardware when we try to validate a subsystem without a real flight. Also it can be used to adjust the subsystem if it is considered by the simulator as a Black Box. The time effects are not really significant but some delays issues can be tested.
- Network-on-the-loop: This loop is very often used when delays in communications take an important place. This case happens when the satellite communicates with the ground station.

In any case, same protocol and variables are used to allow the migration between the different simulating loop methods as presented in the following figure.

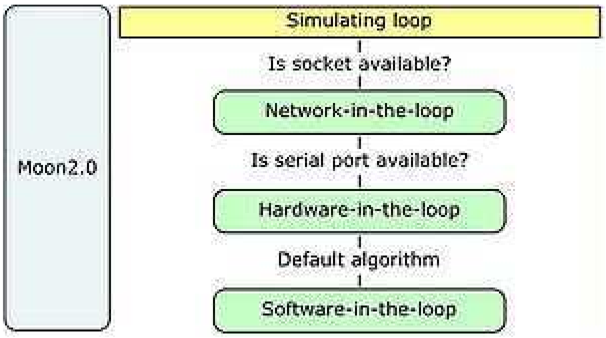

Figure 26 Simulating loop methods

CHAPTER 4. ATTITUDE CONTROL SUBSYSTEM

The attitude control subsystem in the WikiSat is based on few magnetorques[10] assuming that the Earth's magnetic field is strong enough due to the low orbit. Opposite to this, the capability of control using small coils is really limited and stability is achieved in many cycles.

4.1 Magnetorquers

For the Attitude control subsystem of the femto-satellite, two orthogonal copper coils are used as the actuators. When the current is supplied to the coils, magnetic dipole moments are generated. This magnetic dipole moment interacts with the magnetic field of the Earth in order to generate the torque control.

The coils are made of AWG 32 gage magnet wire coil. The maximum power dissipated by each coil is 33 mW. The resulting magnetic dipole moment for each coil is 0.02 Am^2. This means a control torque of $1{\cdot}10^{-6}$ Nm, having enough control during orbit [22].

To compare with similar satellites, if we look at the so called NCUBE satellite [23], the maximum magnetic moment is 0.1 Am^2 thus the cross section is 0.0075 m^2 . The coil they used is made of 100 windings and the coil resistance has 20 ohms fed by a voltage of 3.3 V and a maximum current of 0.165 A.

4.2 Typical attitude control requirements

In this section we are assuming a typical payload for Earth observation like a low resolution camera in order to have an idea about the requirements for such a subsystem.

4.2.1 Minimum resolution per pixel. Camera specifications

Now we calculate the minimum resolution per pixel that a tiny SMD camera of 6x6 mm is able to generate in a 250 km Low Earth Orbit. We are going to use a TOSHIBA TCM8230MD[11] high definition camera, having a resolution of 660x492 pixels in 32 bits in a format called BGGR Bayer filter. The Frames per second is 30 but we will see that is not required such a frame rate. The field of view (FOV) in the horizontal axis is 57.4 degrees and 44.5 degrees in the vertical axis. The memory required for store each picture or tile is:

Tile_memory = Tile_horizontal * Tile_vertical * Pixel_deep = 660 * 492 * 8 = 2,597,760 bits or 324,720 Bytes or 317 kBytes

[10] http://microsat.sm.bmstu.ru/e-library/Algorithms/mss/mag.pdf (Feb/2011)
[11] http://pdf1.alldatasheet.com/datasheet-pdf/view/227785/TOSHIBA/TCM8230MD.html (Feb/2011)

The minimum accuracy required by the attitude subsystem should be lower than the half of a pixel in degrees

Attitude_accuracy<FOV_horizontal / Tile_horizontal / 2 < 0.043º (2.61 minutes of arc)

4.2.2 Scanning width. Satellite specifications

The satellite has a typical orbital speed of 7,516 m/s (27,000 km/h) and a period of 5,369 seconds; what means 1 hour, 29 minutes and 29 seconds per revolution or orbit. The revisiting period is about 85,876 seconds; which is 1 day, 57 minutes and 53 seconds thathappens each 16 orbits.

The scanning width will be:

Scanning_width = Height * tan (FOV_horizontal / 2) = 250,000 * tan (28.7) = 136,871 meters

The width of the scanning pattern is 137 km. The same procedure is applied for the vertical size.

Scanning_vertical = Height * tan (FOV_vertical / 2) = 250,000 * tan (22.25) = 102,278 meters

The vertical length of the scanning is 102 km divided by 492 pixels means that the resolution of each pixel is:

Pixel_vertical = Scanning_vertical / Pixels_vertical= 102,278 / 492 = 207.882 meters

Assuming a sphere model, the projection over the Earth surface is 40,000,000 meters. The number of pictures or tiles needed covered by the vertical scanning are:

Tiles = Earth_perimeter / Scanning_vertical = 40,000,000 / 102,278 = 391 tiles

4.2.3 Tile rate time. Memory required

The tile rate time defines if the camera can be used properly by the MCU and the communications subsystem. This is the minimum time to process and to send this information to ground.

Tile_rateTime = 5,369 / 391 = 13.732 s

This number gave us an idea about the required band width if we know the required memory per picture

BW = Tile_memory / Tile_rateTime = 2,597,760 / 13.732 = 189,182 bps

The communications subsystem has a link budget of 250 kbps in order to allow tracking information in addition to the payload information.

Finally, the amount of memory per revisiting time is:

Revisiting_memory= Tile_memory * Tiles * Revisiting_orbits = 324,720 * 391 * 16 = 1.9 GBytes

4.3 Experiment setup

In order to implement a black-box for the attitude control, we generate the space behavior of the WikiSat using a thread attached to the center of gravity and far away from any magnetic field.

4.3.1 Control schema

The control loop cannot use magnetic sensors that are deformed by the communications antenna and the coils; we use the inertial platform instead. Also, optical sensors can be used like a star-tracker does.

Four coils placed in an orthogonal way are able to generate a torque, rotating the satellite around its center of gravity. See following picture.

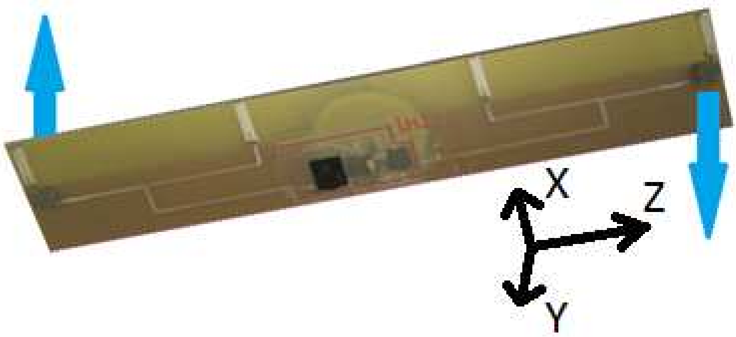

Figure 27 Attitude control schema

Because the satellite is longer than deeper, different gains are modeled for each axis. We only can actuate in two axes but control is performed in three axes. This control system is called 3 to 2 MIMO system because it has multiple inlets and multiple outlets.The inlets are: three gyros. The outlets are: X axis couple of magnetorquers and Z axis couple of magnetorquers.

The reference system is orthogonal, composed by three axes: the X axis is in the direction of the antennas, the Y axis is in the direction of the radiation pattern or direction of view from the sensor, and Z axis is the longer satellite size. The satellite has its longer size towards the launcher flight direction, the so called Z axis thus, in orbit, the satellite has its smaller size towards the Earth.

4.3.2 Attitude control discussion

Due to the lack of time to have a full operative femto-satellite before conclude this bachelor work, we have not gotreal results for this experiment. This experiment requires the communications subsystem implemented in the satellite. We can't use a wire for the black-box as we did with the vector control subsystem. The magnetorquers have a tiny force soany wire can deform the control model.

From the previous calculations, the satellite should be able to point the sensor with an angle error lower than 0.043º and hold the attitude for at least 14 seconds.

4.4 Attitude performances overview

Example of camera modelTOSHIBA TCM8230MD

Tile resolution	660x492x8	bits
Scanned area at LEO 250 km	137x102	km
Pixel size	207	m
Field of view	57.4 by 44.5	º
Attitude accuracy	0.043	º

CHAPTER 5. VECTOR CONTROL SUBSYSTEM

Any launcher able to put a satellite in orbit needs a control system that guarantees the stability during the trajectory and, at the same time, should follow specific trajectory. Different trajectories may end in a excess propellant consumption.

5.1 Vector control description

The WikiLauncher is a two-stage launcher[3], as shown in the later figure. First stage reaches the apogee and second stage reaches the orbital speed. For optimization and for simplicity, we have decided to use only one control system installed in the second stage. During the first burn, helium compressed gas allows the control. Pressure is less but length is higher. When the second stage starts it happens that the first stage is jettisoned and at the same time, extra pressure is used for the control. In this case, length is less but pressure is higher. The trajectory should be more accurate during the second stage burn than the first stage burn in order to ensure the satellite injection.

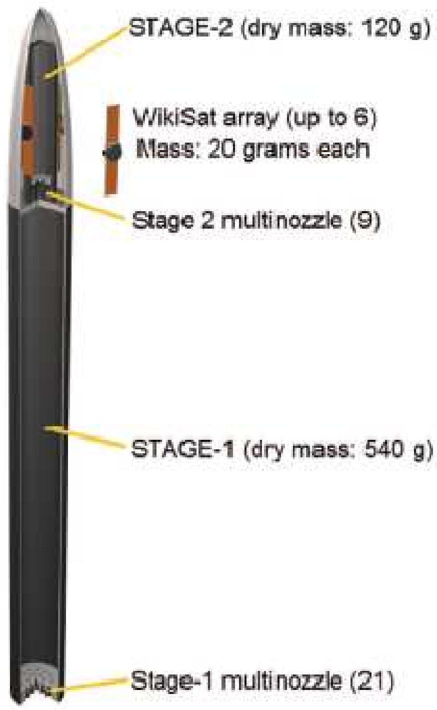

Figure 28 WikiLauncheris a two-stage launcher

5.1.1 Control scheme

The control loop will be fully implemented using MIMO of 7 to 4. The seven inlets are: 3 gyros, 3 accelerometers and 1 temperature sensor. The four outlets are: 4 pseudo-orthogonal nozzles. In addition, many configuration parameters will be needed. These parameters depend on which stage is burning and what is the manifold pressure that fits the four valves for each nozzle.

There are four small lateral nozzles. An alternate use of nozzles produce a Roll movement thus a continuous use of nozzles produces a movement in Pitch or Yaw movements. Control is performed only using four purge valves, reducing to the minimum the number of parts.

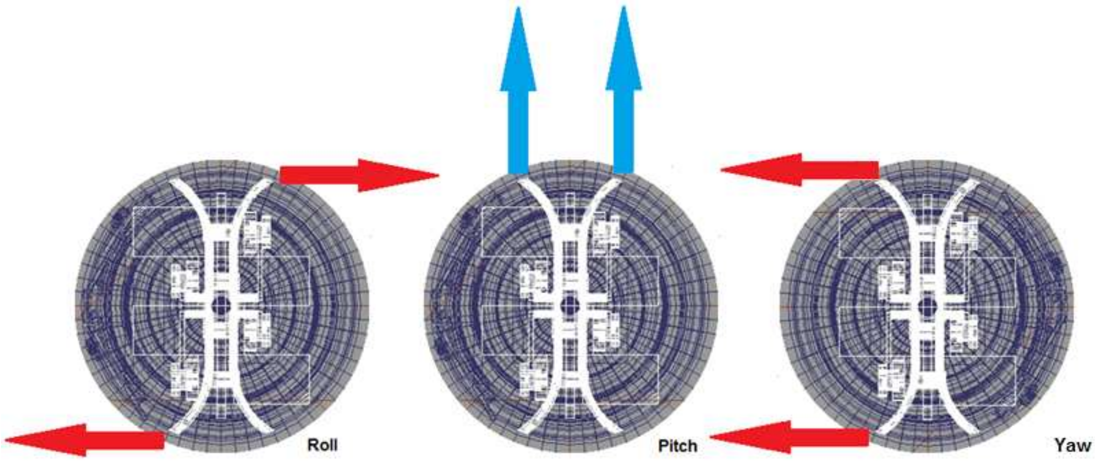

Figure 29 WikiLauncher vector control principle

5.1.2 Launcher flight model

The mathematical model of the launcher could be represented by a transfer function. There are few modes and only lateral mode is presented in equation 5.1 extracted from a technical source[12]. This is the transfer function of a single-engine launcher where δ is the commanded nozzle angle and θ is the launcher Pitch angle. Similar transfer function is used for Yaw angle:

$$\frac{\theta(s)}{\delta(s)} = \frac{C_{m_\alpha} C_{z_\delta} + (\frac{mU}{Sq}(s) - C_{z_\alpha}) C_{m_\delta}}{\left(\frac{mU}{Sq}(s) - C_{z_\alpha}\right)\left(\frac{I_y}{Sqd} s^2 - \frac{d}{2U} C_{m_q} s\right) + (-\frac{mU}{Sq} s - C_w \sin\Theta) C_{m_\alpha}} \tag{5.1}$$

The Vanguardmissile is used here to illustrate the control problems as well as the preceding formula. The stability derivatives for the rigid missile, the structuraltransfer function, and the block diagram of the autopilot were furnishedthrough the courtesy of the Martin Company, Baltimore, Maryland (United States).

To derive the transfer function of the missile, it is necessary to orient the missile axis system. The trajectory of ballistic-type missiles is planned to maintain the missile at a zero angle of attack. This is normally attempted by programming the pitch attitude or pitch rate to yield a zero-g trajectory. This, of course, assumes a certain velocity profile, which may or may not be realized due to variations in the fuel flow rate. This condition and the presence of gusts result in the angle of attack not being zero at all times; however, the angle of attack must be kept

[12] John H. Blakelock, *Automatic Control of Aircraft and Missiles.* 2nd edition, 1991.

small to avoid excessive loading of the structure. Therefore the assumption of zero angle of attack for the equilibrium condition is quite valid, and any changes in angle of attack can be considered perturbations from the equilibriumcondition. With this assumption, if the Xaxis of the missile is along the longitudinal axis of the missile, this body axis system becomes a stability axis system. By placing the Z axis in the trajectory plane, all pitch motion is about the Y axis (see Figure 30).

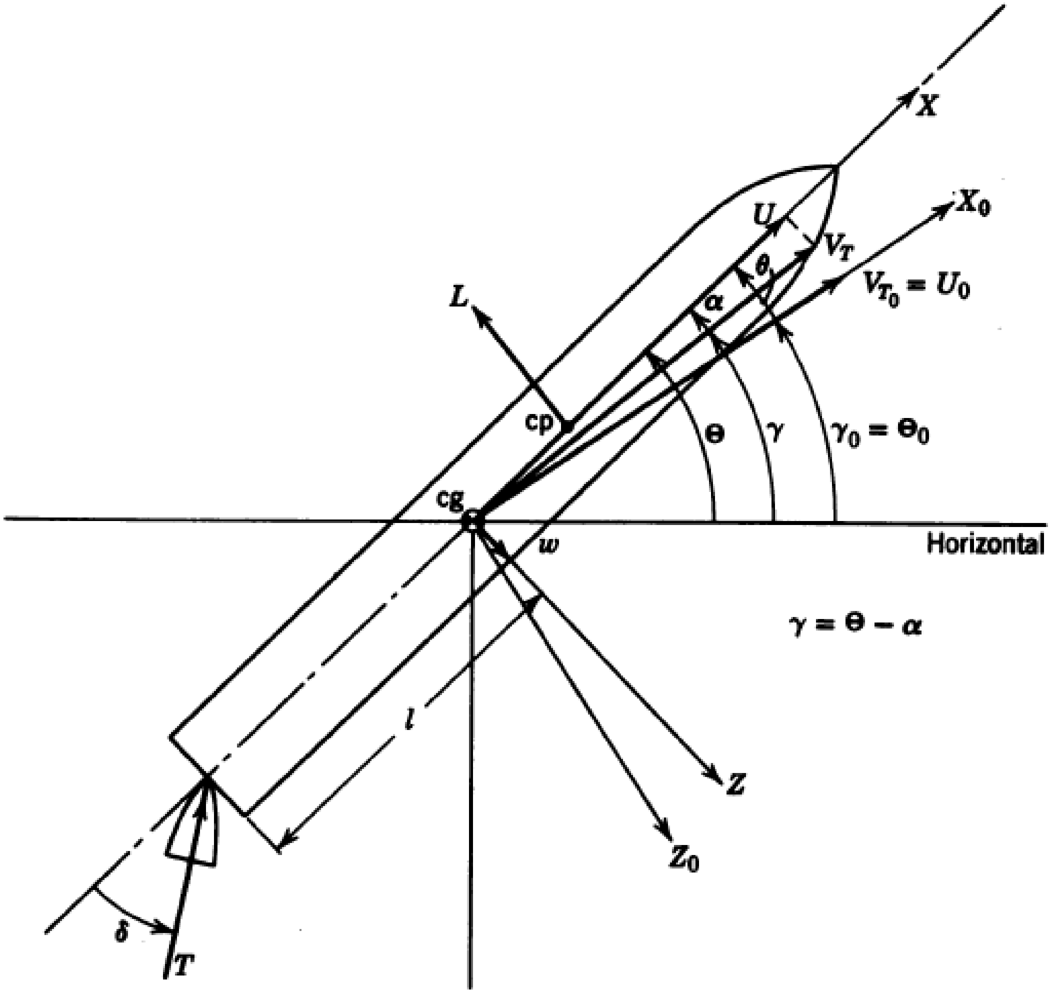

Figure 30 Ballistic missile axis system

For the missile to fly a zero-angle-of-attack trajectory, the centripetal acceleration resulting from the curved trajectory ($-V_T\dot{\gamma}$)must equal the component of weight normal to the Xaxis, which establishes the equilibrium condition. The analysis to follow is based on perturbations from this equilibrium condition. It should be noted from Figure 30 that if an angle of attack is generated, the lift vector acting at the center of pressure (cp) has a destabilizing effect. This factor is one of the major control problems of the missile.
Certain assumptions are made. The assumptions are:

1. The Xand Z axes lie in the plane of symmetry, and the origin of theaxis system is located at the center of gravity of the missile. (Actuallythe Xand Y axes also lie in a plane of symmetry.)

2. The mass of the missile is constant.

3. The missile is a rigid body.

4. The Earth is an inertial reference.

5. The perturbations from equilibrium are small.

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. The ballistic missile in general cannot be considered a rigid body; however, the missile is here analyzed as a rigid body.

The assumption that the Earth is an inertial reference is satisfactory for the analysis of the control system but not of the inertial guidance system that would be sending the command signals to the control system. The assumption of small perturbations is even more valid for the ballistic missile than for an aircraft. Because the perturbations have been assumed small, and as the duration of the disturbance is short, the velocity can be assumed constant during the period of the dynamic analysis.[1]

5.2 Experiment setup

Thanks to Roberto Rodríguez[13] collaboration and his tests, system response has been proved. By means of this test, hardware-on-the-loop based in black-box concept protocol has been validated. System response is shown in the following figure:

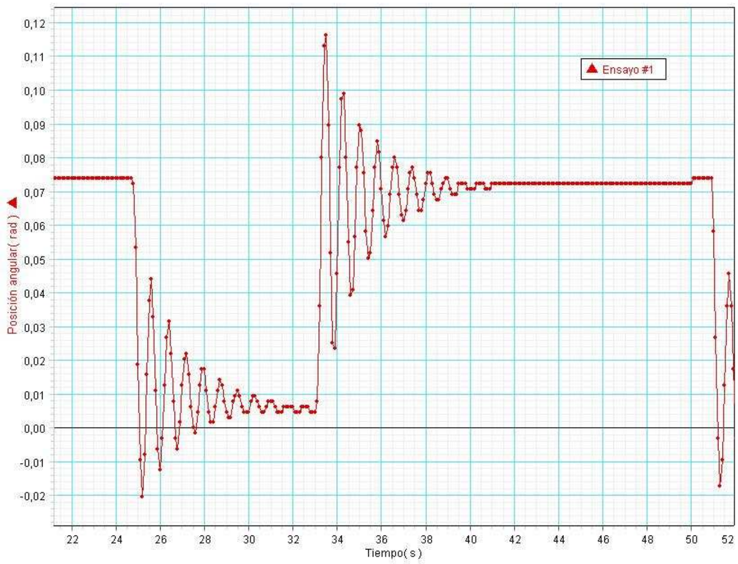

Figure 31 System response

[13] Roberto Rodríguez, "Study of a nozzle vector control for a low cost mini-launcher". UPCommons. Publish pending (2011). https://mitra.upc.es/SIA/PFC_PUBLICA.DADES_PFC?w_codipfc=6450

Launcher attitude angle is represented in the ordinate axis in radians while time is represented in the abscise axis in seconds. System is stable although it has some oscillations preceding movement damping. This is because control force is small. A bigger force is not effective in this case because the system would oversize and it would weight too much.

Next figure shows the setup of the experiment:

Figure 32 Setup of the experiment

Annitrogen bottle gives a pressure of 30 atmospheres actuating in two valves simultaneously thanks to the vector control system. Experiment only measures the angle in one axis though. Hose resistance has been modeled and subtracted from the final result. Figure 33 shows how femto-satellite prototype (The white board) is able to control, by four servos, each valve in function of measures obtained by an inertial platform and boundary information given by Moon2.0 (Conected in the laptop computer) in order to simulate a real flight.

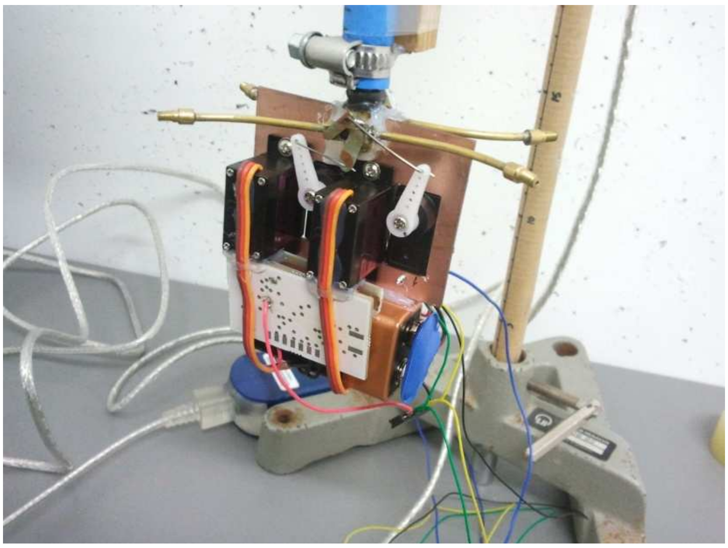

Figure 33 Femto-satellite controlling valves

This case became a basic example of black-box implementation in the Moon2.0 documentation[14].

5.2.1 *Black-box example source of vector control*

The black-box was implemented in an Arduino computer. The communication protocol was implemented as well. Basic algorithm can be programmed in there.

```
#include <Wire.h>

// GYRO ITG-3200
#define GyroID B1101001
//Register 22 - FS_SEL, DLPF_CFG
#define R22 0x16
#define R22INIT B00011011
//Register 62 - Power Management
#define R62 0x3e
#define R62INIT B00000001
//Registers 27 to 34 - Sensor Registers
#define TempH 0x1b
#define TempL 0x1c
#define GyroXH 0x1d
#define GyroXL 0x1e
#define GyroYH 0x1f
#define GyroYL 0x20
#define GyroZH 0x21
#define GyroZL 0x22

//Accelerometer LIS331HH
#define AccID 0x18
//Register CTRL_REG0 - Power management and conf
#define RCTR0 0x20
#define RCTR0INIT B00100111
//Sensor Registers
#define AccXH 0x29
#define AccXL 0x28
#define AccYH 0x2B
#define AccYL 0x2A
#define AccZH 0x2D
#define AccZL 0x2C
#define AccStat 0x27

void i2cSetRegister(intdevice, int address, int value)
{
Wire.beginTransmission(device);
Wire.send(address);
Wire.send(value);
Wire.endTransmission();
}

byte i2cGetRegister(int device, int address)
{
Wire.beginTransmission(device);
Wire.send(address);
Wire.endTransmission();
Wire.requestFrom(device, 1);
if(Wire.available())return Wire.receive();
return B00000000;
}

int i2cGetValue(int device, intaddressH, intaddressL)
{
return i2cGetRegister(device, addressH) * 256 +
i2cGetRegister(device, addressL);
}

// Globals
intledPin=13;
int i;
char a[512];

void setup()
{
Wire.begin();
```

[14] http://code.google.com/p/moon-20/source/browse/win32/Moon.pdf

```
// Hyperterminal at 57600 and Arduino at 115200 bauds
Serial.begin(115200);

// Accelerometer LIS331HH setup
i2cSetRegister(AccID, RCTR0, RCTR0INIT);

// Gyro ITG-3200 setup
i2cSetRegister(GyroID, R62, R62INIT);
i2cSetRegister(GyroID, R22, R22INIT);

// Hardware-on-the-loop config
i=0;
}

void loop()
{
double t, u, v, w;
int m, n, o, p;
digitalWrite(ledPin, HIGH);
if (Serial.available()>0)
a[i++]=Serial.read();
if (i>=512 || a[i-1]==';')
{
if ((a[0]=='I'))
{
// Process INIT block
    }
else if ((a[0]=='D'))
{
// Look for the time stamp
m=4;
for(n=m+1; n<512 && a[n]!=',' && a[n]!=';'; n++);

// Process DATA block
digitalWrite(ledPin, LOW);
Serial.print("CTRL,"); o=4;
for(p=m+1; p<=(n-1); p++, o++)
Serial.print(a[p]);
Serial.print(",2,1,1,"); o+=13;
w=i2cGetValue(AccID, AccXH, AccXL);
u=i2cGetValue(AccID, AccYH, AccYL);
v=i2cGetValue(AccID, AccZH, AccZL);

// Normalize vector
      t=sqrt(u*u+v*v+w*w);
if(t<1E-45)
{
        u=0;
        v=0;
w=0;
}
else
{
u/=t;
        v/=t;
w/=t;
      }
Serial.print(u);
Serial.print(",");
Serial.print(v);
Serial.print(",");
Serial.print(w);
Serial.print(";");
}
i=0;
}
}
```

In the previous source code we used the inertial platform attitude to correct the launcher attitude in the Moon2.0 simulator. Good results were obtained validating the subsystem for a real flight.

Inertial platform attitude vector is obtained in the parameters u, v and w. Because the inertial platform is rigid with respect to the launcher body, the value is the correction itself plus the desired trajectory. The correction vector is normalized and inserted in the control message type CTRL. The previous source example does not include the desired trajectory algorithm implementation for simplicity.

5.2.2 *WikiLauncher vector control discussion*

The method used, based on four nozzles, is simple to implement but it has different gains for each combination of nozzles. This problem can be fixed in the code, modeling a different behavior for each case.
This vector control technique is efficient only in vacuum and outside the atmosphere. Wikisat group decided not to use fins to prevent the horizontal flight as a political requirement. Only vertical flight is allowed. Fins are efficient inside the atmosphere. The launch itself takes place in the stratosphere from a free balloon, outside the atmosphere. Using this approach, the drag is reduced to the minimum that only depends on frontal area and nose shape.

An important reduction in mass is reached because of this synergy between stages. For the same payload, the total size of the launcher, starting from the sea level is ten times bigger and complexity is reduced.

Also, the fact that the launcher is controlled by the satellite, which has the inertial platform, is important in order to reduce the wiring mass. Satellite bus is closer to the second stage vector control bus in the launcher.

5.3 Documentation generation

Technical documentation for Moon2.0 was written thanks to this attitude control study.[15]

A section called *Black-box connection example* was written based on the inertial platform measurement (IMU). Other similar functionalities were written in the same line like: *GPS tracking* and *Rover tracking*.

[15] http://code.google.com/p/moon-20/downloads/list (Feb/2011)

CHAPTER 6. Environmental impact

This final chapter is a study about the effect of this mission in the environmentand a deepstudy of fault probability that is presented in the second section.

6.1 Study upon the environmental impact

6.1.1 Basic considerations

Our design will deliver a satellite in space. In the last phase of the design we have taken into account disposal of the stages in the Earth in case they are not burned completely in the atmosphere.

The launch phase is the most contaminant source in the mission, but debris will reenter soon Earth's atmosphere.

We have tried to optimize resources in order to reduce the impact on the Environment,to waste the lowest electricity as possible and to reuse materials in the development process.

6.1.2 Environmental effects

Because of the sizes weworking with and the low orbit that we have used, this project doesnot represent an important impact on the environment.

Reentry's estimated time is between one week and one month so space debris should be negligible. The small size of the femto-satellite and its materials, assures that it will be disintegrated during the reentry. For the mini-launcher we have studied all the cases and they do not generate debris and the pollution generated is really lower than the generated by a large launcher.In the next section some aspects about population impact in case of system fault are discussed.

The use of small technologiestransforms into power savings. The use of small mini-launchers not only signifies a reduction of costs but also a reduction in pollution and a lower impact upon the natural environment.

In this sense, on the economic and social field, the use of low cost technologies for space uses can open a sector that has been inaccessible for many years. Governments, companies and organizations, that until now were not able to afford the costs of such space missions,can be benefited thanks to theselow cost technologies.

6.2 Risk study and civil responsibility

Spanish regulations state in the *orden 4 mayo 1968 (MinisterioAire. BOE 9 mayo de 1968) - Lanzamiento de ingenios de carácterprivado*[16] that any amateur initiative to explore the space is allowed if they are supervised by INTA.

Also, some rules about unmanned free balloons and unmanned rockets are stated in the PART 101 - MOORED BALLOONS, KITES, UNMANNED ROCKETS AND UNMANNED FREE BALLOONS[17].

The case study is a combination between a free balloon, a small launcher and a femto-satellite.

6.2.1 Risk analysis in the case study

The probability of seeing the launch by a citizen is estimated in $3.5 \cdot 10^{-11}$ which is the population density divided by the people involved in the launch. The damage to the population probability is estimated in $8{,}0 \cdot 10^{-15}$ which is the fragment area (7 cm^2) divided by the exposed area 87,268 km^2 (Andalucía). This area is huge because the range of the balloon trajectory. Population in sea is neglected.

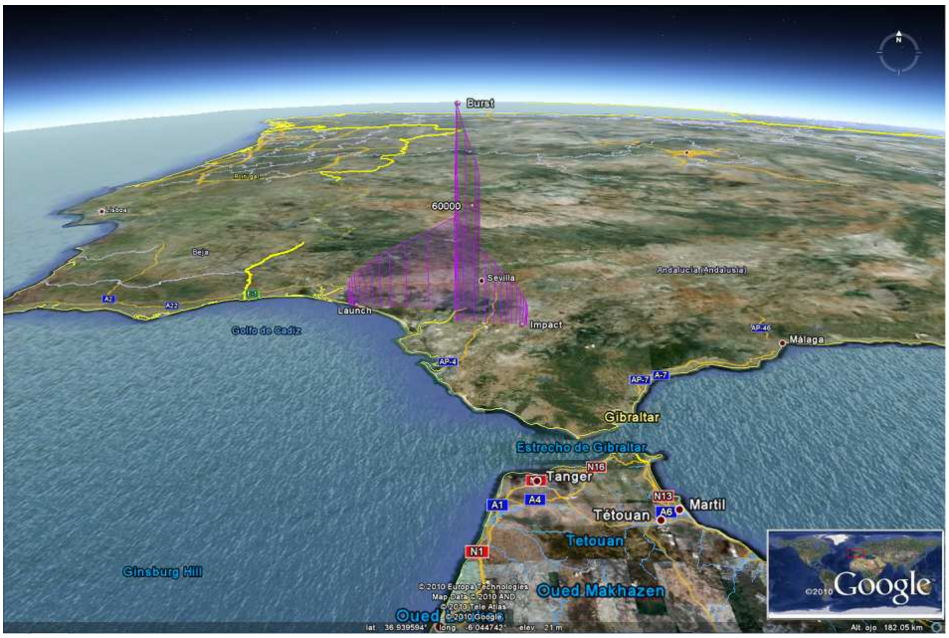

Figure 34 Free balloon trajectory prediction from El Arenosillo launch base

6.2.2 System Failure Assessment

The purpose of this System Failure Assessment[18] is to establish the number of system fault probability in order to detect the weak points in the system and to ensure the minimum safety rules in case of fault. This study is done by a Fault Tree presented in the next figure.

[16] http://www.boe.es/boe/dias/1968/05/09/pdfs/A06766-06767.pdf (Feb/2011)
[17] http://www.eoss.org/pubs/far_annotated.htm (Feb/2011)
[18] http://www.jhberkandassociates.com/systems_failure_analysis.htm (Feb/2011)

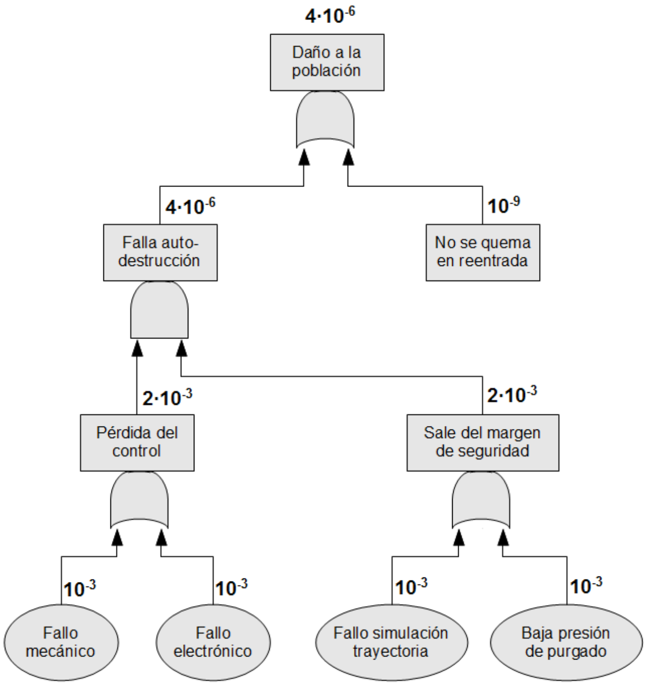

Figure 35 Fault Tree

Next, we present a deep study for each case, the probability of failure for each case that we assumed and the reasons that could make this happen.

Failure probability is small although it would not be admissible for a manned launcher, due to the absence of redundant systems.

6.2.3 Case - Probability of population damage

Case **Daño a la población**. The probability of *Population damage*is$4{\cdot}10^{-6}$ and it happens when any part touches a person which is part of the population.

This case occurs when both cases happens:

- Debries*don't burn during the reentry* and
- *Sefl-destruction mechanism failed*.

6.2.4 Case - No burn during reentry

Case **No se quema en reentrada**: The probability of*No burn during the re-entry* is10^{-9}. This information is based on other real flights and corrected by thermal and ablative calculations in simulations done in Moon2.0 studied by EsteveBardolet[3].

This case occurs when:

- Above a safe altitude, temperature reached due to the friction is not higher enough and
- Ablation energy was lower as expected.

6.2.5 Case - Self-destruction failed

Case **Falla auto-destrucción**: This is the weakest point for surface launch procedures. The probability of *Self-destruction failed* is $4 \cdot 10^{-6}$. This information is based on the study of reliability about the explosives used.This information was studied by EsteveBardolet[3].

This case occurs when:

- The system fails to open the combustion chamber in order to make it explode.

Self-destruction mechanism is not really necessary after burnout of the first stage if the launch was done in a high altitude balloon in the stratosphere.

This situation happens if there is a *Total control lost* and at the same time *Exited from the satety area* case.

6.2.6 Case - Exited from the safe area

Case **Sale del margen de seguridad**: The probability of *Exited from the safe area*is$2 \cdot 10^{-3}$. This information was studied by EsteveBardolet[3].

This case occurs when

- During the first stage burn, the trajectory is different from the simulated safe area.

This situation happens if there is a *Trajectory simulation fault* or there is a *Low purge pressure*.

6.2.7 Case - Control lost

Case **Pérdida del control**: The probability of*Control lost* is$2 \cdot 10^{-3}$. This information was studied by EsteveBardolet[3].

This situation happens if the control and guidancesubsystem does not reach the commanded path. This is due to an *Electrical fault* or a *Mechanical fault*.

6.2.8 Case - Low purge pressure

Case **Baja presión de purgado**: The probability of*Low purge pressure* is10^{-3}. This information is based on some real experiments done by Ernest Arias[19] in his work called "*Design and implementation of a second stage nozzle for a low cost mini-launcher*".

This case occurs when

[19] https://mitra.upc.es/SIA/PFC_PUBLICA.DADES_PFC?w_codipfc=6519(Feb/2011)

- There is not enough pressure in the valve collectorbecause it cannot guarantee the system stability.

This situation happens if the second stage re-fill pressure is not adapted to the needs during the first stage, there is a pressure leak in the second stage, there is a block in the piping system, there is a damage in the pipes or valves or there is a damage in the lateral thrust nozzles.

6.2.9 *Case - Trajectory simulation fault*

Case **Fallosimulacióntrayectoria**: The probability of*Trajectory simulation fault* is10^{-3}.This information was studied by EsteveBardolet[3].

This case occurs when:

- There is no information for the simulation subsystem.

This situation happens if the simulations system does not work or the launcher capacity is not appropriate for the desired mission.

6.2.10 *Case - Electrical fault*

Case **Falloelectrónico**: The probability of*Electrical fault* is10^{-3}.This information is based on some real experiments done by Ernest Arias[20].

This case occurs when:

- The payload does not assume the responsibility of the trajectory control and mission management.

This situation happens if there is not enough battery, there is any fault in the communication bus, there is an interference in the communication bus, there is a jam in the control algorithm or time expectations are not reached.

6.2.11 *Case - Mechanical fault*

Case **Fallomecánico**: The probability of*Mechanical fault* is10^{-3}.This information is based on some real experiments done by WikiSat group.

This case occurs when:

- Control valves are jammed or
- Valves are stuck or
- Valve travel does not move properly.

This situation happens if any part is loosed or bended or a servo is stuck or any valve component is damaged.

[20] https://mitra.upc.es/SIA/PFC_PUBLICA.DADES_PFC?w_codipfc=6519(Feb/2011)

CHAPTER 7. CONCLUSIONS

In this final bachelor work we calculated the moment of inertia of WikiSat. We concluded our experimental tools have not the resolution required for so small values.

We have validated the Hardware-On-The-Loop method between a Black box and the Moon2.0 simulator. Thanks to this validation, we can conclude that, while programming with Arduino, the use of the double type variable implementation is currently exactly the same as the float, with no gain in precision.

We have presented a real case of implementation of this method in a femto-satellite such as the WikiSat v3 in the Attitude control subsystem.

We also have presented a real case of implementation of this method in a mini-launcher such as the WikiLauncher in the vector control system.

We have studied the environmental impact of such a mission.

We have calculated the probability of seeing the launch by a citizen and the system fault number using a fault tree.

We have studied each fault case and we have detected potential faults to consider during the manufacturing and operation phases. That fault tree was build based on real work of WikiSat members.

We conclude that the use of this approach is suitable for larger launchers and satellites.

BIBLIOGRAPHY

[1] John H. Blakelock, "*Automatic Control of Aircraft and Missiles*", 2nd edition.New York: John Wiley & Sons Inc., 1991.ISBN 0-471-50651-6.

[2] Joshua Tristancho, "*Implementation of a femto-satellite and a mini-launcher*", Master Thesis, MAST – UPC (May 2010).

[3] Esteve Bardolet, "*Study of a low cost inertial platform for a femto-satellite deployed by a mini-launcher*", TFC, Enginyeria Tècnica Aeronàutica, esp. Aeronavegació - UPC (July 2010).

[4] Roger Jové, "*Technical constrains for a low cost femto-satellite launcher*", 48th AIAA Aerospace Sciences Meeting (January 2011).

[5] Dagoberto José Salazar Hernández, "*Navegación Aérea, Cartografía y Cosmografía*", UPC. http://nacc.upc.es/nacc-libro/nacc-libro.html

[6] Notes and slides from the subject "*Control and Guidance*", Adeline de Montlaur. Escola d'Enginyeria de Telecomunicació i Aeroespacial de Castelldefels.

[7] Notes from the subject "*Engineering Physical Fundamentals*", Pilar Gil. Escola d'Enginyeria de Telecomunicació i Aeroespacial de Castelldefels.

[8] Ping Wang, et all, "*Satellite Attitude Control Using only Magnetorquers*", IEEE Xplorer, 2002

[9] Toshiba TCM8230MD datasheet, available at: http://pdf1.alldatasheet.com/datasheetpdf/view/227785/TOSHIBA/TCM8230MD.html

[10] http://dvf.mfc.uclv.edu.cu/SEFISAC/sefisac/Contenidos%20teoricos/L-1/node13.html#SECTION04850000000000000000

[11] http://www.profisica.cl/comofuncionan/como.php%3Fid%3D21+campo+magnetico+tama%C3%B1o +im%C3%A1n&cd=1&hl=es&ct=clnk&gl=es

[12] http://www.profisica.cl/comofuncionan/como.php?id=21

[13] http://microsat.sm.bmstu.ru/e-library/Algorithms/mss/mag.pdf

[14] http://www.slideshare.net/kurtmilach/clculo-de-momento-de-inercia

[15] http://www.arduino.cc/

[16] http://es.wikipedia.org/wiki/Momento_de_inercia

[17] http://es.wikipedia.org/wiki/Tensor_de_inercia

[18] http://en.wikipedia.org/wiki/Control_theory

[19] http://en.wikipedia.org/wiki/Open-loop_controller

[20] http://en.wikipedia.org/wiki/PID_controller

[21] Roberto Rodríguez, "Study of a nozzle vector control for a low cost mini-launcher". UPCommons. To be published (2011) https://mitra.upc.es/SIA/PFC_PUBLICA.DADES_PFC?w_codipfc=6450

[22] G. James Wells, et all, "Canda's smallest satellite: The canadian advanced nanospace experiment (canx-1)". 2003

[23] Kjell Magne Fauske, "NCUBE Attitude Control". Departament of Engineering Cybernetics (2002)

ADVICE: Wikipedia references were checked for reliability by the author.

Annex A

Example of data streaming in a typical communication between the black-box and the Moon2.0 simulator.


```
INIT, .05,2007-06-26,12:09:00, 28.2,-16.6, 0, 409.895238938115, 87.2, 15, 88, 15000,
91000, 4, 34.7, 2, 31.212, 3.1212, 567.151258333333, 121, 3.468, .3468,
57.0093253333333, 3, .02

DATA, 0, 34.7, 34.7, 6.6092, 0, .222435897435897, 9.80060472243182, 6378137,-
968534.704408334,-6278974.38685793,-563074.387898972, 9.31322574615479E-10,
9.31322574615479E-10, 409.895238938115,-403.764758270368, 64.7999721286988,-
28.0907546200786, 1.43754951773129E-12,-5.04984942750752E-13,-1.22130777713281E-12,
5.65637814364806E-13

CTRL, 0, 2, 1, 1, .151852289219914, .984452730767299, 8.82819525355084E-02

DATA, .05, 34.6998696057617, 34.7, 6.6092, 5.98228824294984, 2.60788476623399E-03,
9.80060472237322, 6378137.00001907,-968514.51617158,-6278977.62684228,-563072.983361248,
1.90706923604012E-05, 1.90706923604012E-05, 409.895171017881,-403.764735083845,
64.7996870253319,-28.0907544909241, 1.24778599970316E-03, 2.8069450785409E-04,
1.06559389969135E-03,-5.85397416509812E-04

CTRL, .05, 2, 1, 1, .151849123665978, .984453238798535, 8.82817323173373E-02

DATA, .1, 34.6994784230468, 34.7, 6.6092, 17.9468647327312, 7.82365429870196E-03,
9.80060472221342, 6378137.00007107,-968494.327928388,-6278980.86679286,-
563071.578838162, 7.10692256689072E-05, 7.10692256689072E-05, 409.89510309783 2,-
403.764711897318, 64.7994019219655,-28.0907543617697, 2.49557185375651E-03,
5.61386336166231E-04, 2.13118823177862E-03,-1.17079502029855E-03

CTRL, .1, 2, 1, 1, .151845957542517, .984453746907987, 8.82815120895646E-02

DATA, .15, 34.6988264518552, 34.7, 6.6092, 29.9114412388595, 1.30394238311699E-02,
9.80060472205362, 6378137.00012307,-968474.139672321,-6278984.10667591,-
563070.174344352, 1.23067758977413E-04, 1.23067758977413E-04, 409.895035177967,-
403.764688710788, 64.799116818599,-28.0907542326152, 3.74335741198922E-03,
8.42072800521093E-04, 3.19678342526668E-03,-1.75619299749699E-03

CTRL, .15, 2, 1, 1, .151842791417038, .984454255006846, 8.82812918663815E-02

DATA, .2, 34.697913692187, 34.7, 6.6092, 41.8760177591004, 1.82551933636379E-02,
9.80060472189381, 6378137.00017507,-968453.951403379,-6278987.34649142,-
563068.769879818, 1.75066292285919E-04, 1.75066292285919E-04, 409.894967258291,-
403.764665524257, 64.7988317152335,-28.0907541034608, 4.99114267441066E-03,
1.12275389739513E-03, 4.26237948107282E-03,-2.3415913481085E-03

CTRL, .2, 2, 1, 1, .151839625289539, .98445476309511, 8.82810716477879E-02

DATA, .25, 34.6967401440422, 34.7, 6.6092, 53.8405942934539, 2.34709628961059E-02,
9.80060472173401, 6378137.00022707,-968433.763121562,-6278990.5862394,-563067.365444561,
.000227065756917, .000227065756917, 409.8948993388,-403.764642337723, 64.798546611868,-
28.0907539743063, 6.23892764102414E-03, 1.40342963036061E-03, 5.32797639824861E-03,-
2.9269900721162E-03

CTRL, .25, 2, 1, 1, .151836459160004, .984455271172782, 8.82808514337837E-02

DATA, .3, 34.6953058074208, 34.7, 6.6092, 65.8051708419207, 2.86867324285739E-02,
9.80060472157421, 6378137.00027906,-968413.574826871,-6278993.82591985,-563065.96103858,
2.79062427580357E-04, 2.79062427580357E-04, 409.894831419493,-403.764619151185,
64.7982615085029,-28.0907538451518, 7.48671231219068E-03, 1.68409999921712E-03,
6.39357417724878E-03,-3.51238916952009E-03

CTRL, .3, 2, 1, 1, .151833293028499, .984455779239856, 8.82806312243701E-02

DATA, .35, 34.6936106823227, 34.7, 6.6092, 77.7697474044979, 3.39025019610418E-02,
9.80060472141441, 6378137.00033106,-968393.386519306,-6278997.06553275,-
563064.556661875, 3.31060960888863E-04, 3.31060960888863E-04, 409.894763500371,-
403.764595964643, 64.7979764051383,-28.0907537159973, 8.7344966879735E-03,
1.96476500431333E-03, 7.45917281804218E-03,-4.09778864030667E-03

CTRL, .35, 2, 1, 1, .151830126894942, .984456287296336, 8.82804110195453E-02

DATA, .4, 34.6916547687481, 34.7, 6.6092, 89.7343239811891, 3.91182714935098E-02,
9.80060472125461, 6378137.00038306,-968373.198198867,-6279000.30507813,-
563063.152314447, 3.83058562874794E-04, 3.83058562874794E-04, 409.894695581438,-
403.764572778102, 64.7976913017742,-28.0907535868428, 9.98228076751569E-03,
2.24542464179739E-03, 8.52477232062102E-03,-4.68318848447255E-03

CTRL, .4, 2, 1, 1, .151826960759382, .984456795342221, 8.82801908193105E-02
```

```
DATA,  .45,  34.6894380666968,  34.7,  6.6092,  101.698900571992,  4.43340410259778E-02,
9.80060472109481,                 6378137.00043506,-968353.009865555,-6279003.54455597,-
563061.747996296,    4.35058027505875E-04,    4.35058027505875E-04,    409.89462766269,-
403.764549591557, 64.79740619841,-28.0907534576883, .011230064551261, 2.52607891542844E-
03, 9.59037268449941E-03,-5.26858870200779E-03

CTRL, .45, 2, 1, 1, .15182379462177, .984457303377517, 8.82799706236648E-02

DATA,  .5,  34.6869605761688,  34.7,  6.6092,  113.663477176909,  4.95498105584458E-02,
9.80060472093501,                 6378137.00048706,-968332.821519369,-6279006.78396627,-
563060.343707421,    4.87056560814381E-04,    4.87056560814381E-04,    409.894559744126,-
403.764526405008,        64.7971210950464,-28.0907533285338,        1.24778480395782E-02,
2.80672782509203E-03, 1.06559739101243E-02,-5.8539892929019E-03

CTRL, .5, 2, 1, 1, .151820628482155, .984457811402215, 8.82797504326089E-02

DATA,  .55,  34.6842222971643,  34.7,  6.6092,  125.628053795938,  5.47655800909137E-02,
9.80060472077521, 6378137.00053905,-968312.63316031,-6279010.02330904,-563058.939447822,
5.39053231477737E-04,      5.39053231477737E-04,      409.894491825747,-403.764503218456,
64.7968359916831,-28.0907531993793,      1.37256312324712E-02,      3.08737137071907E-03,
1.17215759974957E-02,-6.43939025715845E-03

CTRL, .55, 2, 1, 1, .151817462340554, .984458319416316, 8.82795302461434E-02

DATA,  .6,  34.6812232296831,  34.7,  6.6092,  137.592630429077,  5.99813496233817E-02,
9.80060472061541,  6378137.00059105,-968292.444788379,-6279013.26258427,-563057.5352175,
5.91052696108818E-04,      5.91052696108818E-04,      409.894423907557,-403.764480031903,
64.7965508883203,-28.0907530702248,      1.49734141291245E-02,      3.36800954893435E-03,
1.27871789465435E-02,-7.02479159474702E-03

CTRL, .6, 2, 1, 1, .151814296196885, .984458827419828, 8.82793100642665E-02

DATA,  .65,  34.6779633737253,  34.7,  6.6092,  149.557207076333,  6.51971191558497E-02,
9.8006047204556, 6378137.00064305,-968272.256403574,-6279016.50179197,-563056.131016455,
6.43052160739899E-04,      6.43052160739899E-04,      409.894355989552,-403.764456845347,
64.7962657849575,-28.0907529410702,      1.62211967299888E-02,      3.64864236316192E-03,
1.38527827568596E-02,-7.61019330569145E-03

CTRL, .65, 2, 1, 1, .151811130051196, .984459335412746, 8.82790898869792E-02

DATA,  .7,  34.6744427292909,  34.7,  6.6092,  161.521783737701,  7.04128886883177E-02,
9.80060472029581,                 6378137.00069505,-968252.068005898,-6279019.74093213,-
563054.726844686,    6.95048831403255E-04,    6.95048831403255E-04,    409.894288071731,-
403.764433658787,        64.7959806815957,-28.0907528119157,        1.74689790357738E-02,
3.92926981336983E-03, 1.49183874292836E-02,-8.19559538995424E-03

CTRL, .7, 2, 1, 1, .151807963903538, .984459843395062, 8.82788697142825E-02

DATA,  .75,  34.6706612963799,  34.7,  6.6092,  173.486360413177,  7.56286582207857E-02,
9.80060472013601, 6378137.00074705,-968231.87959535,-6279022.98000476,-563053.322702193,
7.47048296034336E-04,      7.47048296034336E-04,      409.894220154095,-403.764410472224,
64.7956955782338,-28.0907526827611,      1.87167610457741E-02,      4.2098918997146E-03,
1.59839929629214E-02,-8.78099784754569E-03

CTRL, .75, 2, 1, 1, .151804797753811, .984460351366792, 8.82786495461744E-02

DATA,  .8,  34.6666190749922,  34.7,  6.6092,  185.450937102771,  8.08444277532536E-02,
9.8006047199762,  6378137.00079905,-968211.69117193,-6279026.21900985,-563051.918588977,
7.99046829342842E-04,      7.99046829342842E-04,      409.894152236648,-403.764387285661,
64.7954104748724,-28.0907525536066,      1.99645427595062E-02,      4.49050861845396E-03,
1.70495993581811E-02,-9.36640067844909E-03

CTRL, .8, 2, 1, 1, .151801631602083, .984460859327925, 8.82784293826563E-02

DATA,  .85,  34.6623160651279,  34.7,  6.6092,  197.415513806475,  8.60601972857216E-02,
9.80060471981641, 6378137.00085104,-968191.502735639,-6279029.4579474,-563050.514505038,
8.51044431328774E-04,      8.51044431328774E-04,      409.894084319385,-403.764364099094,
64.7951253715109,-28.090752424452,      2.12123241774064E-02,      4.77111997315344E-03,
1.81152066146157E-02,-9.95180388266072E-03

CTRL, .85, 2, 1, 1, .151798465448351, .984461367278461, 8.82782092237283E-02

DATA,  .9,  34.657752266787,  34.7,  6.6092,  209.38009052429,  9.12759668181896E-02,
9.8006047196566, 6378137.00090304,-968171.314286476,-6279032.69681743,-563049.110450376,
9.0304296463728E-04,      9.0304296463728E-04,      409.894016402307,-403.764340912523,
64.7948402681504,-28.0907522952975,      2.24601053002833E-02,      5.05172596400666E-03,
1.91808147331036E-02,-1.05372074601742E-02

CTRL, .9, 2, 1, 1, .151795299292584, .984461875218407, 8.82779890693896E-02

DATA,  .95,  34.6529276799695,  34.7,  6.6092,  221.34466725622,  9.64917363506575E-02,
9.80060471949681, 6378137.00095504,-968151.125824443,-6279035.93561991,-563047.70642499,
9.55040566623211E-04,      9.55040566623211E-04,      409.893948485414,-403.764317725949,
64.7945551647899,-28.0907521661429,      2.37078861272989E-02,      5.33232659075086E-03,
2.02464237127197E-02,-1.11226114109756E-02

CTRL, .95, 2, 1, 1, .151792133134815, .984462383147753, 8.82777689196407E-02
```

```
DATA,   1,   34.6478423046753,   34.7,   6.6092,   233.30924400226,   .101707505883126,
9.80060471933701,                 6378137.00100704,-968130.937349539,-6279039.17435486,-
563046.302428881,   1.00703909993172E-03,   1.00703909993172E-03,   409.893880568706,-
403.764294539371,       64.7942700614299,-28.0907520369883,        2.49556666589053E-02,
5.61292185364889E-03, 2.13120335539031E-02,-1.17080157350654E-02

CTRL, 1, 2, 1, 1, .151788966975011, .984462891066509, 8.82775487744813E-02
```