بِسْــــــمِ اللهِ الرَّحْمَنِ الرَّحِيمِ

IAP-SAT

3rd Project Report (2015)

Embedded Test and Simulation Environment for a meteorological data collecting Satellite Prototype

Authors:

Samir Mourad

Fatima Al Chaar

Ibrahim Ghanem

A member institute of AECENAR
www.aecenar.com/institutes/iap

Content

List of figures

Remark: Numbers of Figures/Graphs/Tables are not correct.

Project Status at Beginning of actual project phase

Bismillah

IAP Laboratory at Ras Nhache/Batroun, Lebanon	
Simulation Environment	In such a way the Graphical User Interface shall be

Master Thesis

Simulation of the system and of the operational flight of IAP-SAT to take meteorological data to monitor and estimate the energy supply potential of large scale photovoltaic energy plants to control a national alternative energy supply program

Detailed description and working plan

Modeling of actual IAP-SAT hardware and software components in octave	8 weeks
Modeling of IAP-SAT mechanical components with ProE	3 weeks
Integration to a operation flight model and visualization in IAP-ECS environment	3 weeks
Documentation	3 weeks

Keywords: Control of national energy mix programs, large scale photovoltaic plants, meteorological data, *Low Earth Orbit Satellite, Flight Mechanics&Dynamics, Matlab Simulink/Scilab*

Contact: Samir Mourad, Email: samir.mourad@aecenar.com, Mobile +961 76341526

1   Goal of actual phase

- Specification: Presentation Film

- Ground Track Visualization (done with Gpredict)

- Hardware-in-the-Loop (HIL) testing environment with scicos and server

- On-Board-Computer Integration with Control Algorithm

- Visualization of satellite movement in orbit based on scilab simulation data

- Specification of propulsion unit

2 Project Management

2.1 Organisation

1 Satellite Bus

> 1.1 FCS
>
> HIL Testrig

> 1.2 Ground Station
>
> with direction finder SDR tracking

2 Payload: Earth surveillance camera
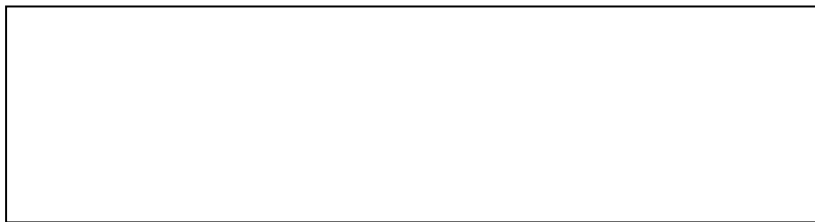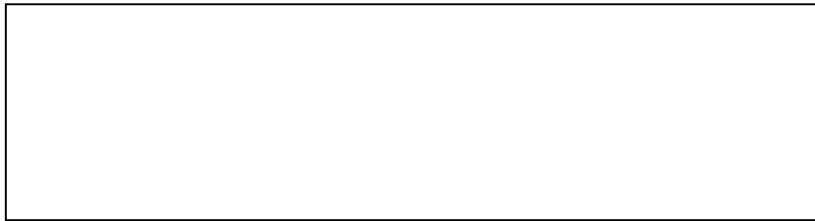
3 Mission Planning

> 3.1 Access to space (Planning: IAP, Operation: a contractor)

> 3.2 Operation of the satellite

2.2 Working Packages and Time Plan

WP No.	Working package content	Time span, costs	Development environment (HW, SW)	Responsible	Status
1	Specification, Cost Analysis. Result: Presentation Film	Mar-Apr 2015 (2 man months)	FreeCAD, Gpredict, OpenSat	Master Student Fatima Al Chaar	finished
2	Hardware-in-the-Loop test rig without adaption to board computer	May-Aug 2015 (4 man months)	Scicos, ubuntu, C compiler	Master Student Fatima Al Chaar	Finished
3	Concept for Propulsion Unit	July/August 20015 (1 man month)		Practical Student Ibrahim Ghanem	finished
4	Implementation of control algorithm at On-Board-Computer and Closed-Loop-Integration of HIL			A1	open
5	Visualization of satellite movement in orbit based on scilab simulation data			Muh. Amine	open
7	Specification of battery system			Practical Student Ibrahim Ghanem	finished

2.3 Estimated Costs of IAP-SAT (satellite development and the launching)

Satellite Development Cost and Launch Cost

Working Package	Material Cost	Man Month	Qualification	Salary/MM	Personnel Cost per item	Total item cost
Camera	$40,000	6	Eng	$5,000	$30,000	$70,000
Chemical Propulsion System	$50,000	10	Eng	$5,000	$50,000	$100,000
Gyroscopes	$20,000	10	Eng	$5,000	$50,000	$70,000
Accelerometers	$20,000	10	Eng	$5,000	$50,000	$70,000
Tank for fuel	$10,000	5	Eng	$5,000	$25,000	$35,000
Tank for oxygen	$10,000	5	Eng	$5,000	$25,000	$35,000
Solar panels incuding battery system	$15,000	10	Eng	$5,000	$50,000	$65,000
Communication's board	$5,000	10	Eng	$5,000	$50,000	$55,000
Board Control Computer	$10,000	10	Eng	$5,000	$50,000	$60,000
Antenna system	$10,000	10	Eng	$5,000	$50,000	$60,000
Integration	$10,000	5	Eng	$5,000	$25,000	$35,000
Test	$10,000	15	Eng	$5,000	$75,000	$85,000
Launch	$1,600,000			$5,000	$0	$1,600,000
Ground Station	$100,000			$5,000	$0	$100,000
				Total Cost		$2,440,000

Operational Cost per year

Working Package	Material Cost	Man Month	Qualification	Salary/MM	Personnel Cost per item	Total item cost
Maintenance	$40,000	12	Eng	$5,000	$60,000	$100,000
Ground Station	$20,000	36		$3,000	$108,000	$128,000
				Total Cost		$228,000

Table1: Costs of development and launching of IAP

With the consideration of the maintenance and operational cost, the total cost of IAP-SAT is around 2,668,000$. The satellite's expected life time is at least 5 years.

3 Basics

3.1 Meteorology

Meteorology is the study of weather, climate, and forces that cause changes in our environment. It uses math and physics to understand the atmosphere, which is consisted of layers of gases and moisture surrounding the earth. Most weather takes place in the lowest level of the atmosphere, known as the troposphere. Within meteorology there are a number of specialty fields which include climatology, severe storms and tornadoes, tropical cyclones, hydrology, and agriculture. [23]
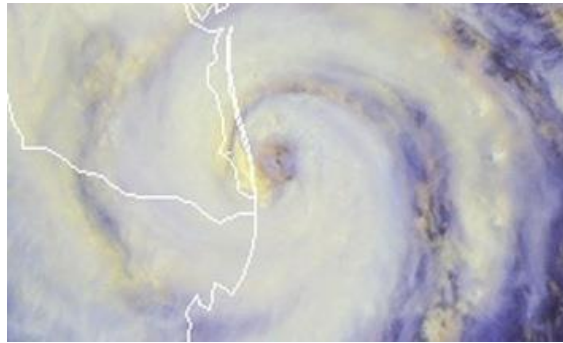
Figure 3.1: Meteorological image

Meteorology has application in many diverse fields such as military, transport, agriculture, construction and energy production. It offers a range of practical applications which make a major contribution to human safety and the security of air and sea transport.

3.2 Satellite

3.2.1 Definition

A satellite is an object that goes, or orbits, around a larger object, such as a planet. While there are natural satellites, like the moon, hundreds of man-made satellites also orbit around the earth (Fig.3.2). [24]

Figure 3.2: Artificial satellite

3.2.2 Orbit

3.2.2.1 Laws of motion

A satellite's orbit is the curved path it follows around earth (Fig.3.3).

The basis for classical mechanics Laws of motion (Newton's laws):

1. Everybody continues in its state of rest or of uniform motion in a straight line unless it is compelled to change that state by a force impressed upon it.

2. The rate of change of momentum is proportional to the impressed force and is in the same direction as that force.

Momentum = mass × velocity, so Law (2) becomes F= d (mv)/dt = m dv/dt =ma for constant mass

3. For every action, there is an equal and opposite reaction. [5]
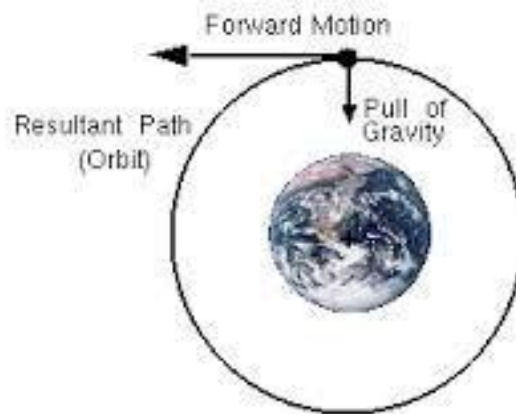
Figure 3.3: Third law of Newton

3.2.2.2 Laws of gravitation

The force of attraction between any two particles is

• Proportional to their masses

• Inversely proportional to the square of the distance between them

i.e. $F= G*m_1*m_2/r^2$ (treating the masses as points)

Where G = gravitational constant = 6.673 × 10-11 Nm2 /kg2

So, the followed laws explain how a satellite stays in orbit.

Law (1): A satellite would tend to go off in a straight line if no force were applied to it.

Law (2): An attractive force makes the satellite deviate from a straight line and orbit around earth.

The Laws of Gravitation:

This attractive force is the gravitational force between the earth and the satellite. Gravity provides the inward pull that keeps the satellite in orbit. Assuming a circular orbit, the gravitational force must equal the centripetal force.

$$m.\ v^2/r = G.m.m_E/r^2$$

Where: v = tangential velocity

r = orbit radius = R_E + h (i.e. not the altitude of the

orbit)

R_E = radius of Earth

h = altitude of orbit = height above Earth's surface

m = mass of satellite

m_E = mass of Earth

$v = \sqrt{(Gm_E/r)}$, so v depends only on the altitude of the orbit (not on the satellite's mass).

The period of the satellite's orbit is *$T = 2nr/v = 2\pi r/v = 2\pi \sqrt{(r^3/Gm_E)}$.*

Again, this is only dependent on the altitude, increasing as the orbit's altitude increases. [24]

Form of orbit:

Kepler's Laws were based on observations of the motions of planets.

     1)   All planets travel in elliptical orbits with the Sun is at one focus (Fig. 3.4) → defines the shape of orbits.
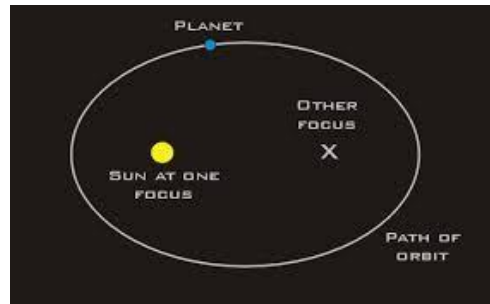
Figure 3.4: First law of Keppler (elliptical orbit)

2) The radius from the earth to the satellite sweeps out equal areas in equal time —determines how orbital position varies in time

3) The square of the period of a satellite's revolution is proportional to the cube of its semi major axis.

$$T^2 = 4\pi^2 \, a^3/GM; \text{ so } (T_1/T_2)^2 = (r_1/r_2)^3$$

The pull of gravity is stronger closer to the earth, so a satellite in a low orbit must travel faster than one in a geostationary orbit (Fig. 3.5). [25]
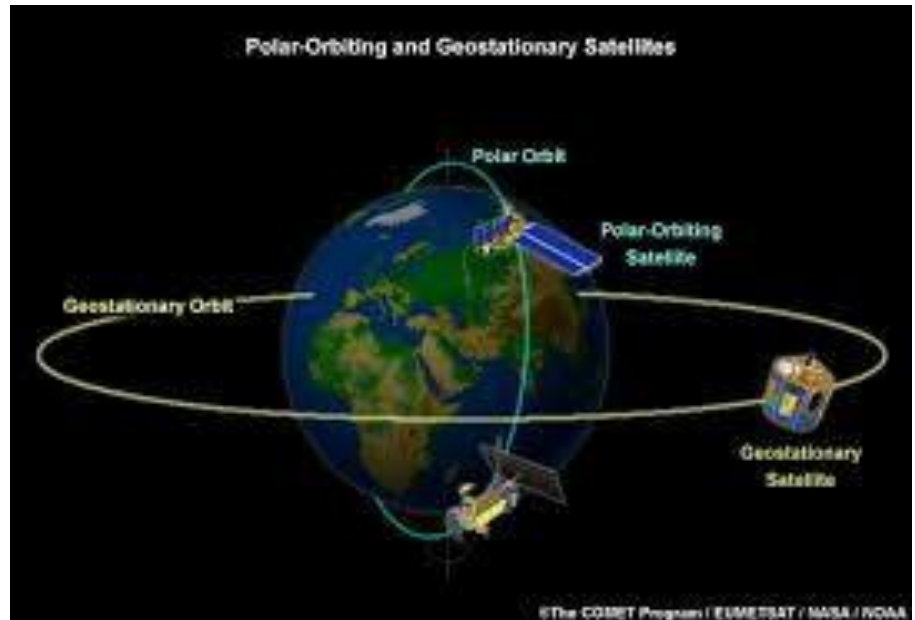
Figure 3.5: Third law of Keppler (satellite polar orbiting is faster than geostationary orbiting because of radius)

3.3 Low earth orbit

The Low Earth Orbit (LEO) is used for the vast majority of satellites. As the names imply, Low Earth Orbit is relatively low in altitude; the definition of LEO stating that the altitude range is between 200 and 1200 km above the Earth's surface (Fig. 3.6).

Low Earth Orbit
Minimum Atmospheric
Influence
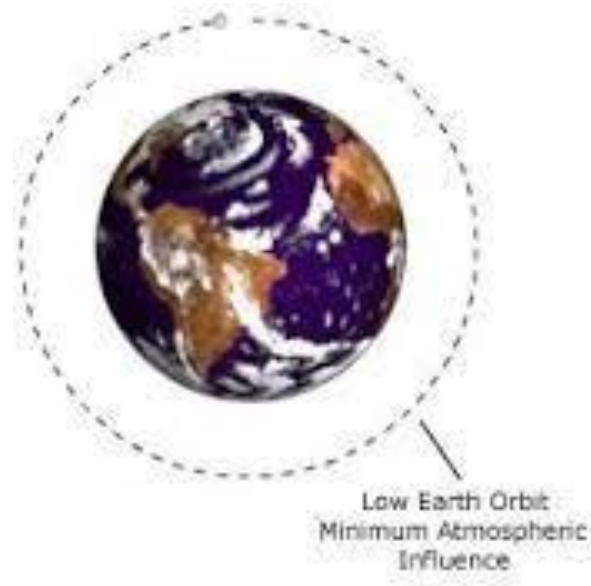
Figure 3.6: Low earth orbit

3.3.1 LEO basics

LEO is still very close to the Earth, especially when compared to other forms of satellite orbit including geostationary orbit. The low orbit altitude leads to a number of characteristics:

- Orbit times are much less than for many other forms of orbit. The lower altitude means higher velocities are required to balance the earth's gravitational field. Typical velocities are very approximately around 8 km/s, with orbit times sometimes of the order of 90 minutes.

- Radiation levels are lower than experienced at higher altitudes.

- Less energy is expended placing the satellites in LEO than higher orbits.

- Some speed reduction may be experienced as a result of friction from the low, as a result of the increasing drag from the presence of gasses at low altitudes. [22]

3.4 Sun synchronous orbit

Sun-synchronous orbit (SS-O) is a special case of the polar orbit. Like a polar orbit, the satellite travels from the north to the south poles as the earth turns below it (Fig. 3.7). [2]
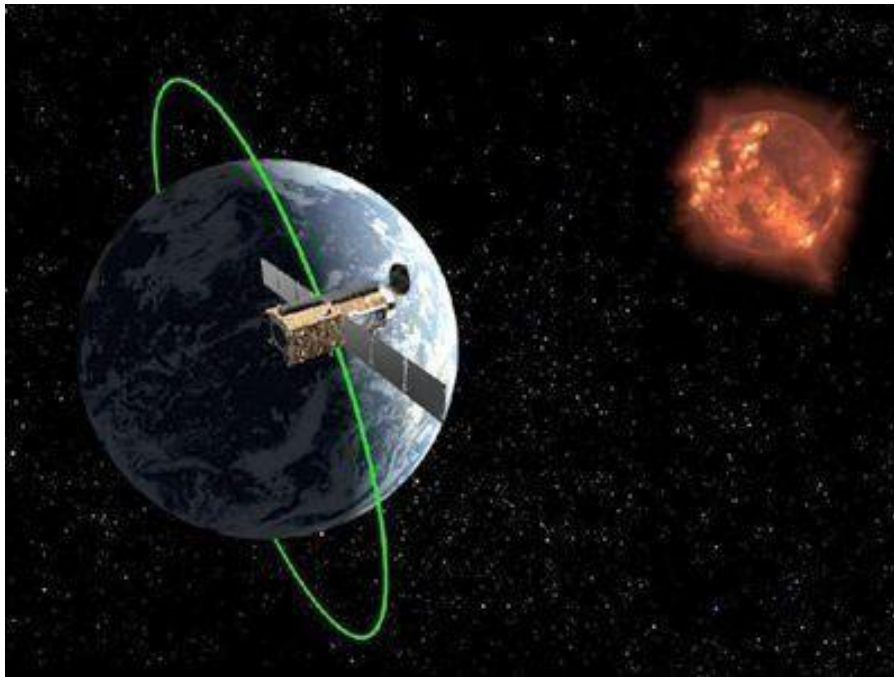
Figure 3.7: Sun synchronous orbit

A Sun-synchronous orbit is a geocentric orbit which combines altitude and inclination in such a way that an object on that orbit will appear to orbit in the same position, from the perspective of the Sun, during its orbit around the Earth (Fig. 3.8). [26]
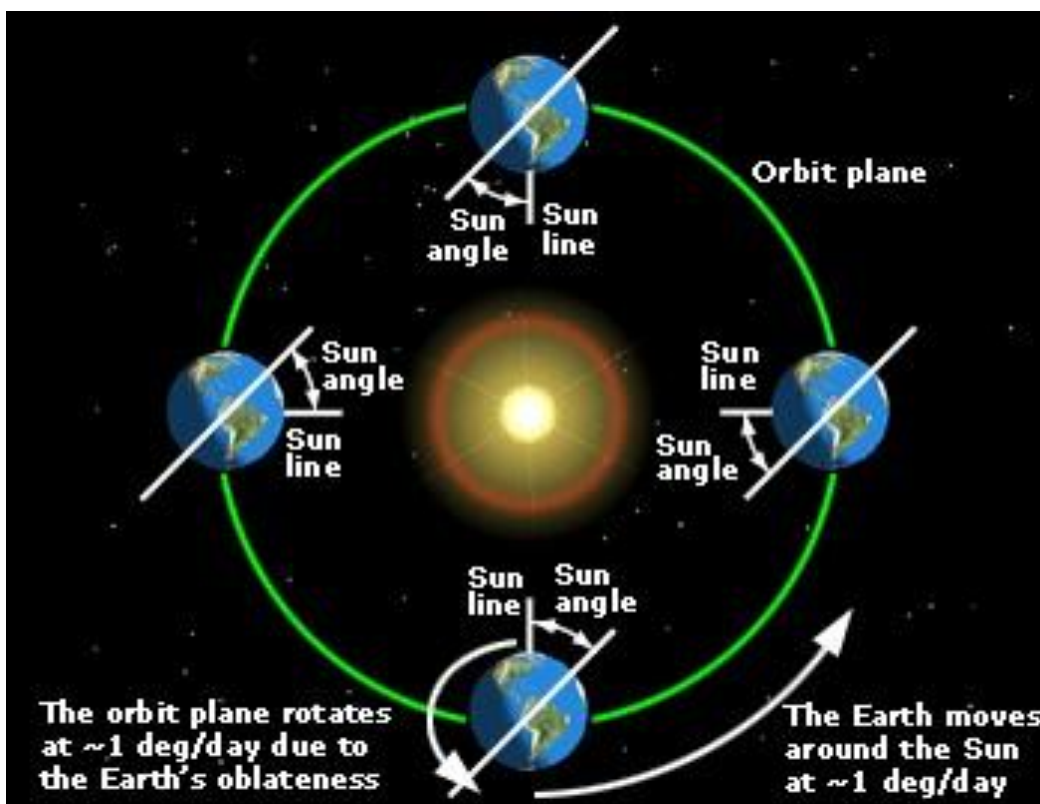
Figure 3.8: Sun synchronous orbit (conservation of the angle between the sun and the orbit plane)

The uniformity of Sun angle is achieved by tuning the inclination to the altitude of the orbit. The plane of the orbit is not fixed in space relative to the distant stars, but rotates slowly about the earth's axis. Typical sun-synchronous orbits are about 600–800 km in altitude, with periods in the 90–100 minutes range, and inclinations of around 98. [26]
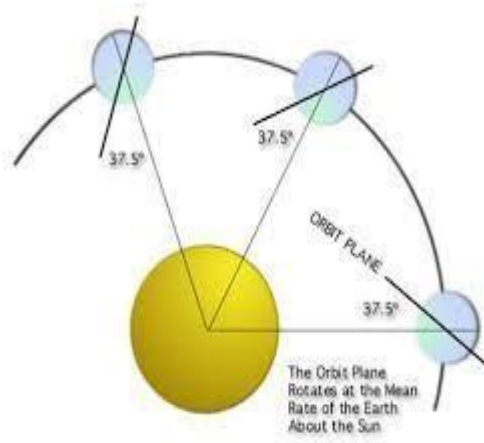
Figure 3.9: Sun synchronous orbit

Only the solar panels can be redirected to optimize the electric power depending on the position of the sun (Fig. 3.10).
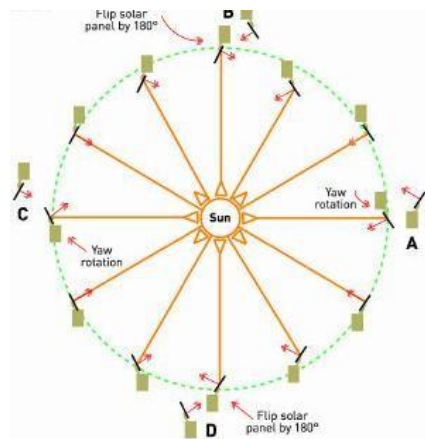
Figure 3.10: Sun synchronous orbit (only the solar panels can rotate)

3.5 Meteorological satellite

Meteorological satellite is an artificial satellite that gathers data concerning the earth's atmosphere and surface in order to aid meteorologists in understanding weather patterns and producing weather forecasts.

Satellite Meteorology refers to the study of earth's atmosphere and oceans using data obtained from remote sensing devices flown on board satellites orbiting the earth. Satellite

makes measurements indirectly by sensing electromagnetic radiations coming from the surfaces below.

These weather satellites are often placed in a sun-synchronous orbit. These satellites are in polar orbits. The orbits are designed so that the satellite's orientation is fixed relative to the sun throughout the year, allowing very accurate weather predictions to be made. Most meteorological satellites orbit the earth 15 to 16 times per day. [1]

3.5.1 Components of meteorological satellite

A meteorological satellite comprises two parts: satellite bus and payload.

3.5.1.1 Satellite bus (platform): power source, battery, control system, communication system, thermal protection...

i. **Power source**

Figure 3.11: Solar panels of the satellite

It is the solar panels in satellite; just like many other machines, satellites also need electrical power to function.

The sun is a very powerful, clean and convenient source of power, particularly for satellites. The only thing needed is a means to convert the energy contained in the sun's radiation – mainly light and ultraviolet rays – into electrical power. The most efficient way to achieve this today is by using panels composed of semiconductor photovoltaic cells. 'Solar panels', as they are usually called. [1]

ii. **Battery**

Figure 3.12: Battery of the satellite

Satellites orbiting the earth pass through a shadow region on the opposite side of the earth from the sun. Depending on the type of orbit; this can happen just a few times a year or every few hours. During these so-called 'eclipses', the solar panels cannot produce electrical energy and the satellite would not only be unable to operate, but would also freeze to incredibly low temperatures (eventually around –270°C) if a backup power source were not available. Electrical energy therefore has to be stored on board the spacecraft when in sunlight for consumption during these eclipses.

Batteries are used to store the energy, so that the satellite can continue to work when the Sun is eclipsed. [1]

iii. Control thermal system

Space is not a friendly environment either. Satellites have to survive temperature variations of more than 200°C – rather like someone standing in front of a fireplace with a blazing fire while an air conditioner pumps freezing air onto his back. [1]

Satellites orbiting the earth are subjected to similar extreme temperature variations. The delicate electronics on man-made satellites would not operate efficiently over this temperature range, so it is necessary to insulate the satellite from the space environment. Active thermal control systems such as resistive heaters, thermo-electric coolers, heat pipes... are critical tools for managing the temperature in localized areas of the satellite. Heating systems consume valuable electrical power. Cooling systems also consume power and actually only concentrate thermal energy in a small volume that must still be passively radiated into space. [21]

iv. Control system (sensors, actuators)

To position itself in space, a satellite has to maneuver using its own small propulsion (e.g electrical or chemical propulsion). It also has to maintain its orientation, using thrusters and gyroscopes, otherwise it will tumble along its orbit and its antenna will drift out of alignment with the earth. [1]

a. Sensors

1) Gyroscope

Gyro Rate Sensors determine the attitude by measuring the rate of rotation of the satellite. They are located internal to the satellite and work at all points in an orbit. Since they measure a change instead of absolute attitude, gyroscopes must be used along with other attitude hardware to obtain full measurements. At least three gyro sensors are used in a satellite. [30]

2) Accelerometer

An accelerometer is a device that measures proper acceleration. The proper acceleration measured by an accelerometer is not necessarily the coordinate acceleration (rate of change of velocity). [30]

b. Actuators

Satellite propulsion is characterized in general by its complete integration within the satellite. Its function is to provide forces in the space to:

 - position, adjust, and maintain orbits of satellite: used for orbit control

 - orient the satellite: used for attitude control

The propulsion is a reaction Jets which produce a control force by the expenditure of mass; These systems are based on jet propulsion devices that produce thrust by ejecting stored matter, called the propellant. [28]

Satellite propulsion can be classified according to the source of energy utilized for the ejection of propellant:

1)   Chemical propulsion (based on the principle of converting chemical energy into kinetic energy of the exhaust gases) uses heat energy produced by a chemical reaction to generate gases at high temperature and pressure in a combustion chamber. These hot gases are accelerated through a nozzle and ejected from the system at a high exit velocity to produce thrust force (Fig. 3.13).
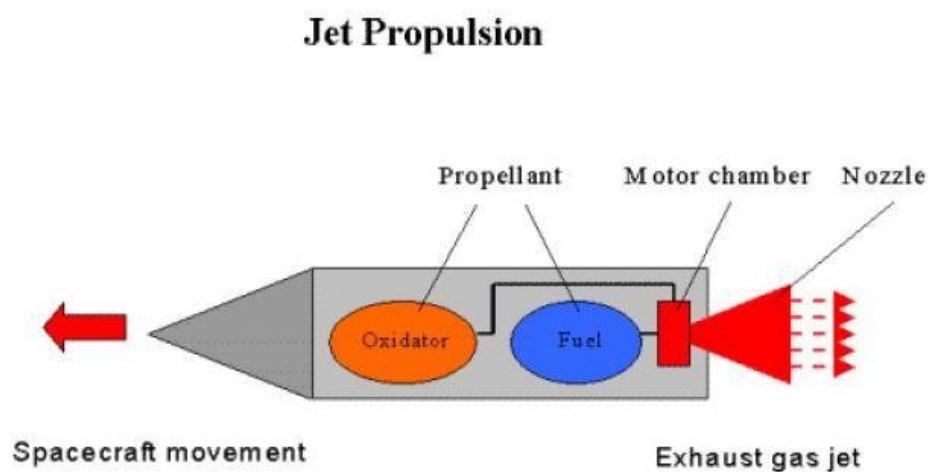
Figure 3.13: Chemical propulsion

It uses two separate propellants, a liquid fuel and liquid oxidizer. These are contained in separate tanks and are mixed only upon injection into the combustion chamber (Fig. 3.13). [4]

2) Electrical propulsion uses electric or electromagnetic energy to eject matter at high velocity to produce thrust force. [4]

3) Hall Effect thrusters

16

Figure 3.14: Hall Effect propulsion

Hall Effect thrusters have been classed as both electrostatic and electromagnetic propulsion systems. An electrostatic field accelerates the ions in the propellant stream but that field is, to a large extent, produced by the actions of plasma electrons interacting with a magnetic field, giving both classes a claim to the technique.

In Hall Effect thrusters, also referred to as closed drift thrusters, a heavy gas, xenon in most implementations to date, is ionized and accelerated by an electric field. An electron current passing through and being impeded by a magnetic field establishes the acceleration field. The processing electrons in this magnetic field follow a closed drift path, giving rise to one of the names for this technique. The flow of the drifting electrons, which is perpendicular to both the applied magnetic field and the current flow in the plasma, is due to the Hall Effect, the source of the other name for this type of thruster (Fig. 3.14). [15]

v. **Communication system**

Outside the protection of the Earth's atmosphere, the level of radiation (UV, X-rays, gamma-rays and all sorts of energetic particles) is much higher and more destructive than on the earth. Before they can even begin to operate in space, satellites have to survive the bone-shaking launch. Then the solar panels have to be opened and antennas, which are often stowed to take less space in the launcher, deployed before the satellite enters its operational orbit.

Once in orbit, a satellite usually carries out multiple functions, with different payloads or instruments. It then sends information to a ground station about the condition of its payload and its systems, and it receives instructions back from the ground operators. All this has to work well, with little possibility of recovery, not to mention repair. Apart from very special cases, there is no way backing. [1]

3.5.1.2 Payload

The main mission is carried out by the primary payload instruments, the imager and the sounder. The imager is a multichannel instrument that senses infrared and visible reflected solar energy from the Earth's surface and atmosphere. The sounder provides data for vertical atmospheric temperature and moisture profiles, surface and cloud top temperature, and ozone distribution. [3]

The meteorological mission needs an imager (camera) as payload.

3.6 Example of a meteorological satellite

NOAA (National Oceanic and Atmospheric Administration) is a three-axis stabilized American spacecraft (Fig. 3.15). It has launched into an 870-km circular, near-polar orbit with an inclination angle of 98.73° to the Equator. The total orbital period will be approximately 102.14 minutes. The sunlight period will average about 72 minutes, and the Earth shadow period will average about 30 minutes. Because the Earth rotates, the satellite observes a different portion of the Earth's surface during each orbit. The nominal orbit is planned to be sun-synchronous, and rotates eastward about the Earth's polar axis 0.986° per day. The precession keeps the satellite in a constant position with reference to the sun for consistent illumination throughout the year. [11]

Figure 3.15: NOAA satellite

3.7 Example: DubaiSat-1 orbit specifications

The frame consists of spacecraft (S/C) adaptors, six longerons, rails, inner ring, inner rods and three decks. Six longerons are connected to six rails at the bottom, middle, and top deck. The inner ring provides structure stiffness and stability for DMAC payload. Three S/C adaptors are connected to the shear brackets and mechanically couple the satellite and separation adaptor of the launch vehicle (LV); Dnepr Rocket. Each one of the S/C adaptors has two S/C-based separation sensors and one LV-based separation sensor to monitor separation condition between the satellite and the launch vehicle. The umbilical connector is attached on the lower surface of the bottom deck with its mounting support bracket. The key system features of DubaiSat-1 are introduced in Table 1.

Figure 2: DMAC Mechanical Structure.

The main characteristics of the DMAC are shown in the following table (Table 2).

Table 1: DubaiSat-1 key system features.

Parameters	Features
Orbit	Sun-Synchronous Low Earth Orbit
Nominal Altitude	680 km
Inclination	98.13 deg
Mass	~ 200 kg (Bus: 150 kg)
Dimension	Ø 1,200 x 1,200 (mm) - Hexagonal
Attitude Accuracy	0.2° (Full 3-axis stabilized)

Table 2: DMAC main system features.

Parameters	Features
Spectral Bands	Pan: 420-720 nm (the Pan and MS data are co-registered) MS1: 420-510 nm (Blue) MS2: 510-580 nm (Green) MS3: 600-720 nm (Red) MS4: 760-890 nm (Near Infrared)
Spatial Resolution	2.5m (Pan), 5m (MS)
Swath Width	20 km
FOR (Field Of Regard)	720 km (this provides an event monitoring capability)
Detector	Charged Coupled Device (CCD)
Quantization	8-bits
Instrument mass, power, and size	32 kg, < 60 W (max), Ø 420 mm x 610 mm

3.8 Ground Tracking

Ground tracks and orbital cycles

Figure 3.25. **Succession of descending tracks over a Landsat-1, -2 and -3 orbital cycle.** Above: descending tracks during the first day of the cycle. The distance between the track described in one revolution and the track described in the subsequent revolution is known as the tracking interval. On completion of 14 revolutions, track n = 14 is located to the west of track n since the combined length of 14 tracking intervals exceeds that of the Equator. Track n = 14 is executed on day two of the cycle. Below: descending

Figure 3.19. **Tracks described by International Space Station over a 24-hour period.** Orbit altitude is 385 km and inclination 51.6°. The orbital period is 92.18 minutes and the distance at the Equator between two successive tracks in the same direction is 2608.4 km.

Figure 3.20. **Tracks described by HCMM over a 24-hour period.** Orbit altitude is 630 km and inclination 97.5°. The orbital period is 97.2 minutes and the distance at the Equator between two successive tracks in the same direction is 2712 km.

3.9 Electrostatic Plasma Thrusters

The purpose of this research was finding the best Plasma Thruster for the satellite project in order to mechanically model a prototype of it.

Electrostatic thrusters are a branch of Plasma Thrusters that uses high voltage electric fields to accelerate ions out of the propulsion providing thrust. The core principle of this technology lies in first ionizing the propellant then accelerating it out using an electric field. Many different propulsion systems under this domain were analyzed and each one's adequacy to our project was checked.

• Ion Bombardment

neutralizing electron gun
magnets
electron gun
negative grid
positive grid

- electron
- neutral propellant atom
- positive ion
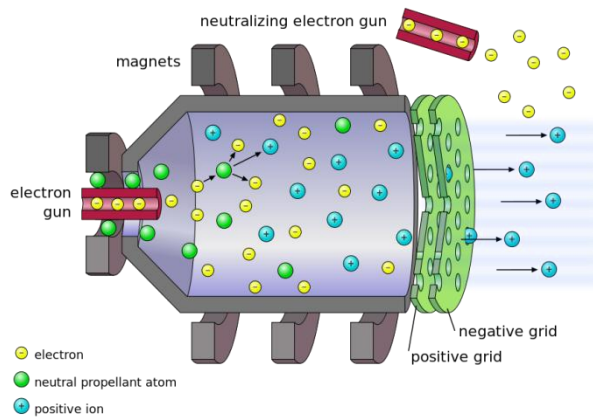
In these types of thrusters Xenon fuel is ionized in a discharge chamber by bombarding it with electrons using an electron gun. The formed plasma (positively charged gas) diffuses into a high-voltage electric field formed between 2 charged gridded plates where it is accelerated to high speeds out of the exhaust. In order to prevent the spacecraft from acquiring a negative net charge (which will block the system), electrons are injected into the exiting plasma in order to neutralize it. [1]

The following types of electrostatic thrusters were also analyzed in detail:

• Colloid ion
• Contact ion
• Field Emission (FEEP)
• Microwave or Radiofrequency ion
• Plasma separator ion
• Radioisotopic ion
• Hall Effect

[1] http://www.daviddarling.info/encyclopedia/E/electronbomthruster.html

3.10 Electromagnetic propulsion systems

Electromagnetic propulsion technology generally consists of 2 steps:
1. Production of plasma in an ionization chamber
2. Accelerating the plasma to the exhaust velocity by electromagnetic forces

Electromagnetic thrusters have higher efficiencies over their electrostatic counterparts. This is due to the fact that plasma in these thrusters can be brought up to very high temperatures and densities resulting in the ability to confine and direct this plasma. As a result, higher exhaust velocities and more economic propellant usage are achieved.

Electromagnetic technology embodies several propulsion systems each using different techniques and processes. All of these systems were carefully studied and analyzed in order to choose the best system that mechanically fulfills the IAP-satellite project's requirements.

• Magnetoplasmadynamic
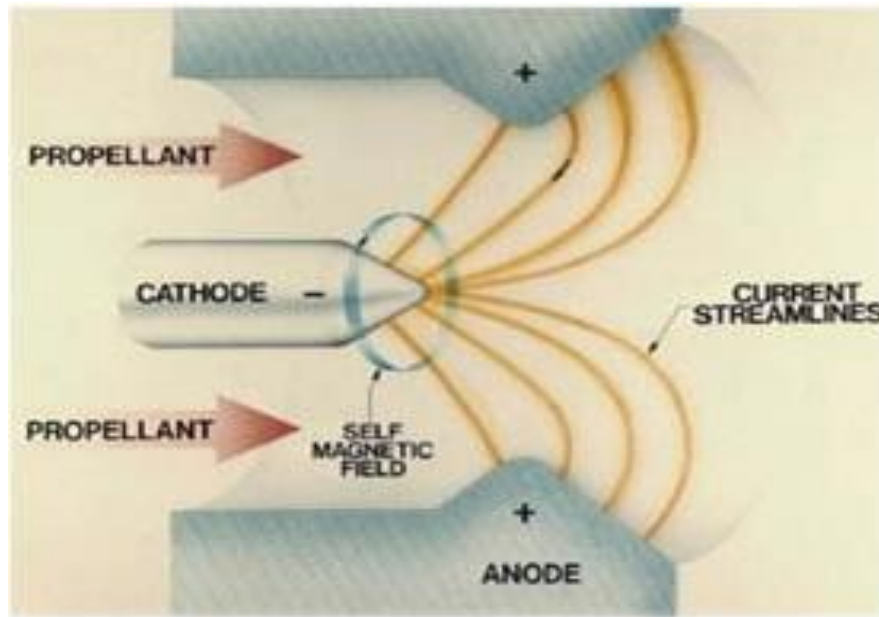
This type of thruster consists of a cathode rod centered in the middle and a cylindrical anode that surrounds the cathode. The high-voltage electric field between the anode and the cathode ionizes the fuel injected into the system. The positively charged ions complete the circuit and the resulting current flow induces a magnetic field. The interaction between both magnetic and electric fields produces a Lorentz force that acts on the plasma and accelerate it to high speeds.[1]

Other systems were also analyzed:

• Pulsed plasma
• Helicon plasma
• Inductive pulsed plasma
• Electron-cyclotron-resonance
• Variable specific-impulse plasma

[1] http://www.daviddarling.info/encyclopedia/M/MPDthruster.html

3.11 Block Diagrams in Signal and Control Theory

The block diagram modeling may provide control engineers with a better understanding of the composition and interconnection of the components of a system.

It can be used, together with transfer functions, to describe the cause-effect relationships throughout the system.

The common elements in block diagrams of most control systems include:

• Comparators

• Blocks representing individual component transfer functions, including:

・ Input sensor

・ Output sensor

・ Actuator

- Controller

- Plant (the component whose variables are to be controlled)

  - Input or reference signals

  - Output signals

  - Disturbance signal

  - Feedback loops   [17]

3.12 Tools

3.12.1 Scilab/Scicos

The software used in this project is scilab/Scicos (ubuntu version).

Scilab is a free open-source software package for scientific computation. It includes hundreds of general purpose and specialized functions for numerical computation, organized in libraries called toolboxes that cover such areas as simulation, optimization, systems and control, and signal processing. One important Scilab toolbox is Scicos.

Scilab is one of the major open-source alternatives to MATLAB. It is similar enough to MATLAB, and Scicos is the open source equivalent to Simulink from the MathWorks. [16]

Scicos is a scilab toolbox for modeling and simulating dynamical systems. It is particularly useful for modeling systems where continuous-time and discrete time components are disconnected.

The graphic editor of the toolbox Scicos allows the user to describe his dynamic system in a completely modular way. One of the basic module from which scicos diagram is constructed is the block. A block in Scicos defines an operation that can evolve continuously or discretely in time. By interconnecting blocks with links, the user can construct his algorithm. A large number of mostly used blocks are saved in palettes, these blocks provide elementary operations needed to construct models diagram. [29]

3.12.2 FreeCAD

Computer-aided design (CAD) (Fig. 3.16) is the use of computer systems to assist in the creation, modification, analysis, or optimization of a design. CAD software is used to increase the productivity of the designer, improve the quality of design, improve communications through documentation, and to create a database for manufacturing. CAD output is often in the form of electronic files for print, machining, or other manufacturing operations. [9]
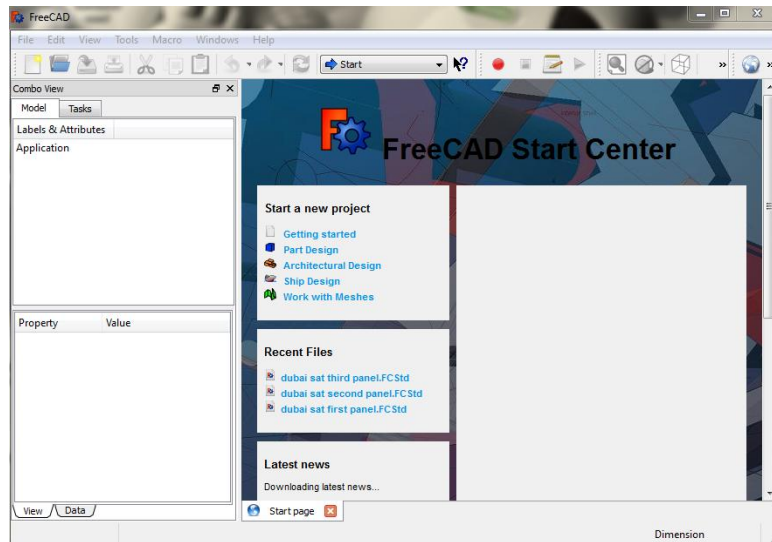
Figure 3.16: Window of FreeCAD

3.12.2.1 Example CAD model of satellite: Flying Laptop, Stuttgart University, Germany
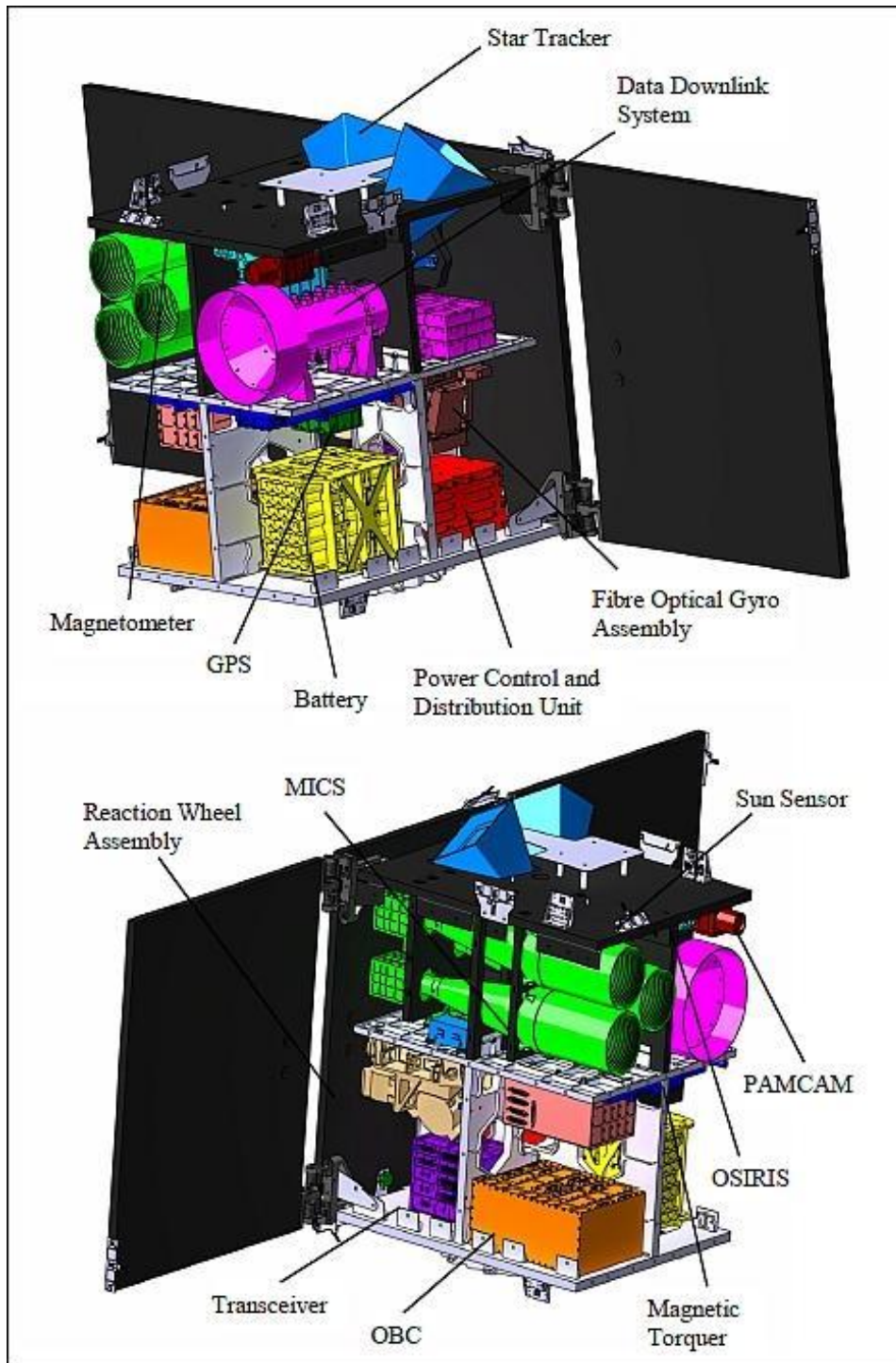
3.12.3 Tools for visualization of the ground track of satellites: Gpredict and OpenSat
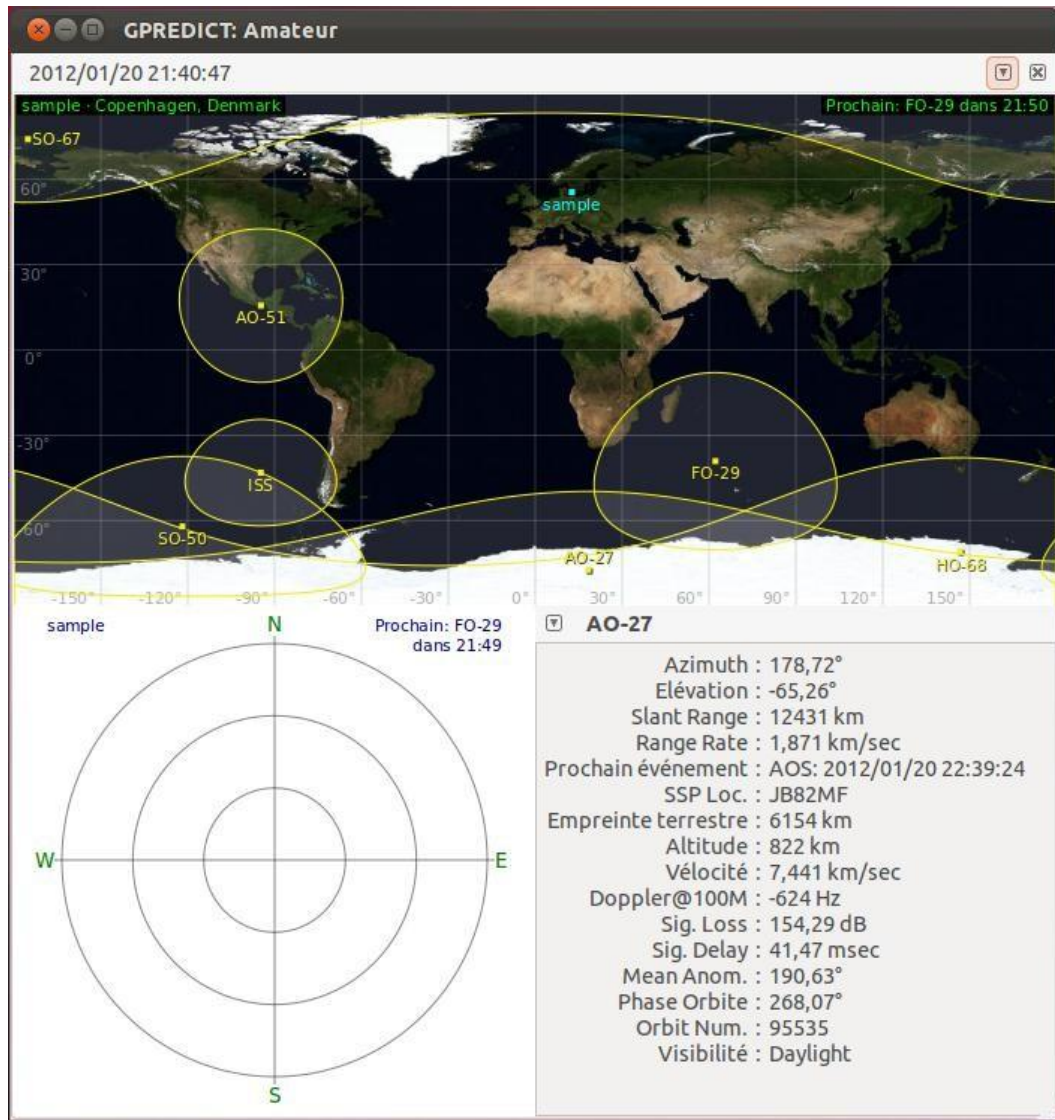
Figure 3.17: Gpredict software

To simulate the satellite around the earth, a specific program is necessary.

We want to take such open source program and adapt it to our specific requirements. We want to have a program which tracks our satellite (IAP-SAT). The source code of gpredict is used to make our program.

Gpredict is a real-time satellite tracking and orbit prediction program (Fig. 3.17). A satellite tracking program is a computer program that predicts the position and velocity of a satellite at a given time using a mathematical model of the orbit. It can also predict the time of future passes for a satellite, and provide you with detailed information about each pass. [13]

Gpredict can track an unlimited number of satellites and display their position and other data in lists, tables, maps, and polar plots (Fig. 3.18).
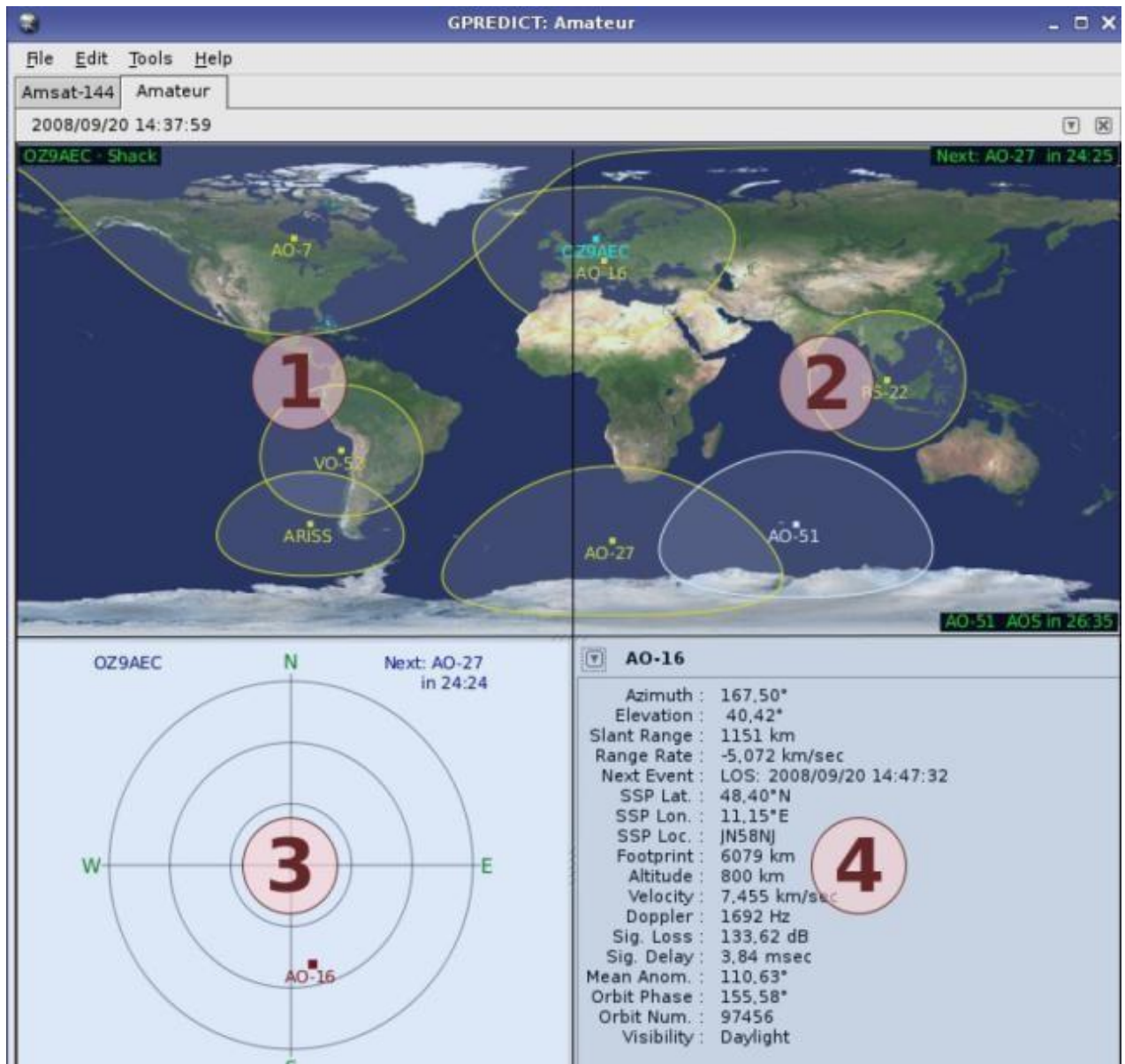
Figure 3.18: Window of Gpredict (list, map and polar plot)

Opensat

It is open source software, which also follows all satellites in the atmosphere, and gives their ground tracking (Fig. 3.19 and Fig. 3.20), as Gpredict.

Figure 3.19: Open-Satellite software
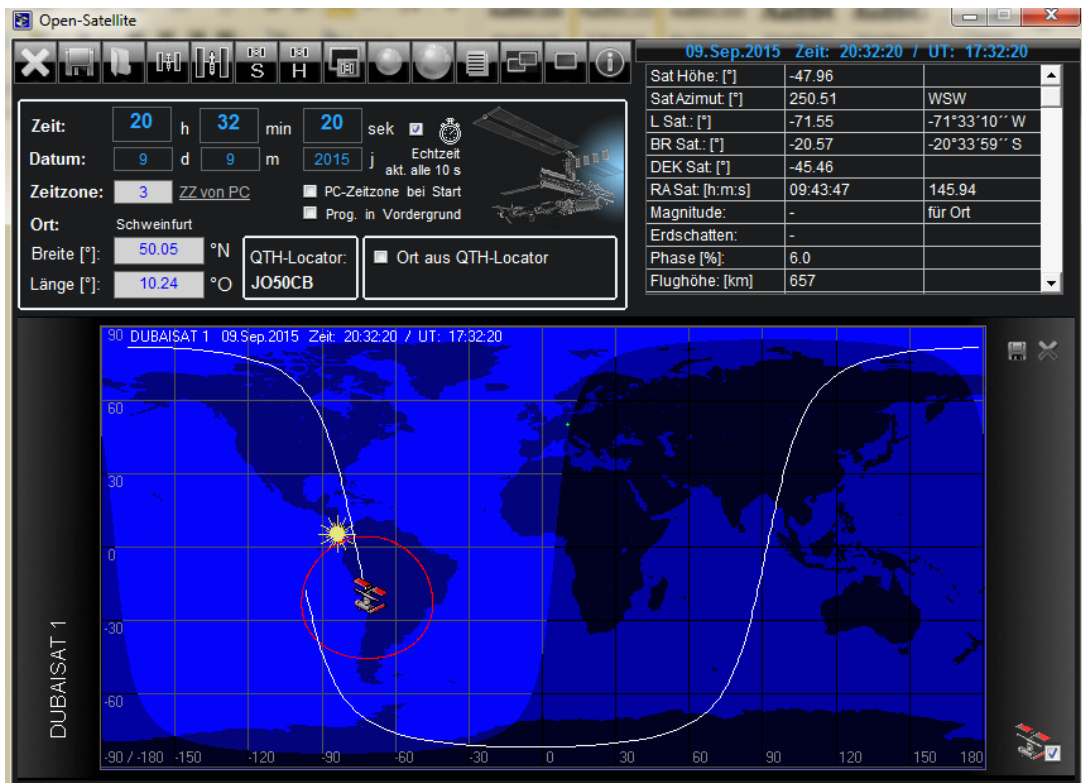
Figure 3.20: Window of OpenSatellite

4 Simulation of IAP-SAT mission

4.1 Specification of IAP-SAT's parameters

IAP-SAT is the first Lebanese meteorological satellite. It will be used to take meteorological data to estimate the state of weather in Lebanon.

This satellite has the same parameters of Dubai-SAT;

It will be placed on a sun-synchronous low earth orbit, on a nominal altitude of 680 Km. [10]

The parameters of this satellite are presented in the following table (Table 2.1):

Parameters	Features
Orbits	Sun-synchronous Low Earth Orbit
Altitude	680 Km
Inclination	98.13˚
Mass	200 Kg (Payload: 50Kg, Bus:150Kg)
Dimension	Hexagonal: 1.2×1.2(m)
Attitude Accuracy	0.2˚

Table 4.1: Parameters of IAP-SAT

4.2 Description and Modeling of IAP-SAT

This satellite contains, according to its mission:

- A camera
- Sensors: 3 gyroscopes, 3 accelerometers
- Actuators: 4 electrical propulsions (Hall Effect thrusters)
- Board of communication
- Board computer
- Battery
- Antenna
- 3 solar panels

After the specification of IAP-SAT and fixing its parameters, this satellite will be modeled. The program CAD is used to attain this object (Fig. 2.2).

IAP-SAT has a shape close to that of DubaiSat.

We construct geometrical shapes depending on the shape of the body represented by this geometric shape, for example: a hexagonal shape represents the body of the satellite, a rectangular shape represents the computer board; a cylindrical shape represents the antenna. Then, we approach and we collect all these forms together in a specific manner to obtain the model of our satellite.

<div align="center">(a)   (b)   (c)</div>
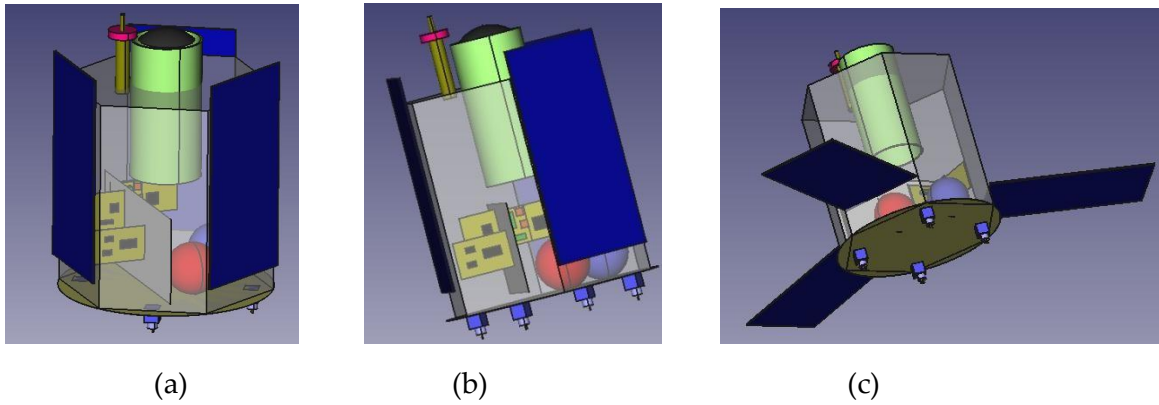
Figure 4.1 (a, b, c): Model of IAP-SAT in FreeCAD

4.3 Visualization of the ground track of IAP-SAT

After modeling this satellite, it will be constructed and launched to take the needed meteorological data. This satellite attains the sun-synchronous low earth orbit, and starts to turn around the earth. The ground station detects these moves by showing the ground track for this satellite when it changes its position with the time. We use the OpenSatellite software to follow the satellite and obtain its ground tracking represented in the next figures (Fig.2.3 and Fig. 2.4), and we can estimate the position of this satellite which changes with time, so we obtain the prediction of its ground tracking.

Figure 4.2: Window of OpenSatellite when DubaiSat is selected

IAP-SAT is not in the atmosphere now, so we try to follow the ground tracking of the closest satellite; it is Dubaisat.
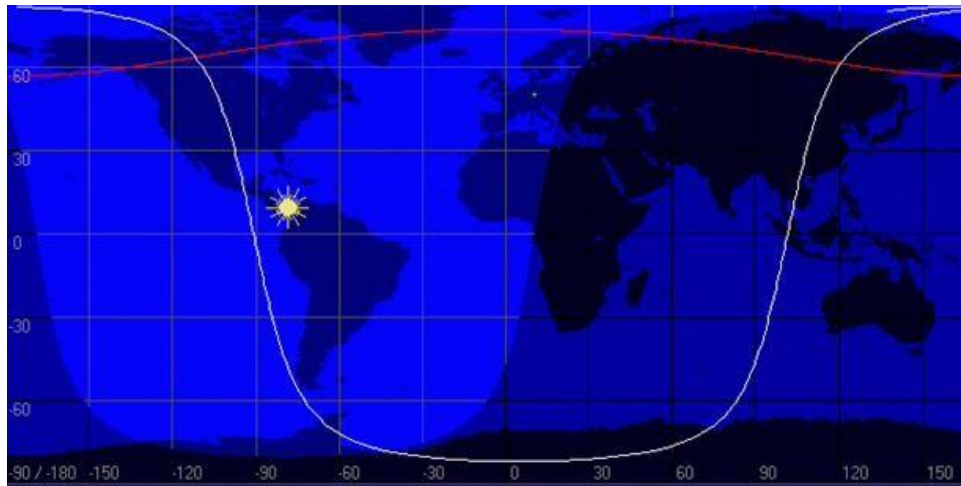
Figure 4.3: One complete rotation around the earth

The satellite accomplishes one rotation around the earth in 90 minutes.

After 90 minutes, it takes another position for a new rotation. The difference between the previous and the next track is shown in the figure 2.4:
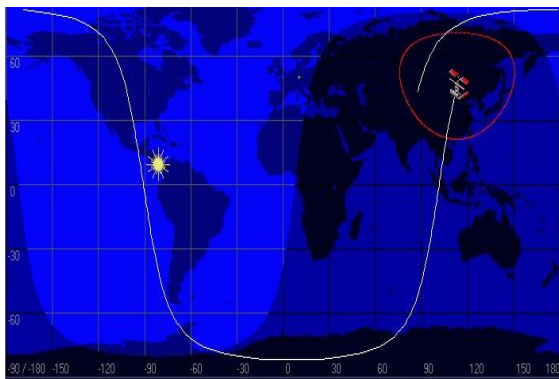
(a)
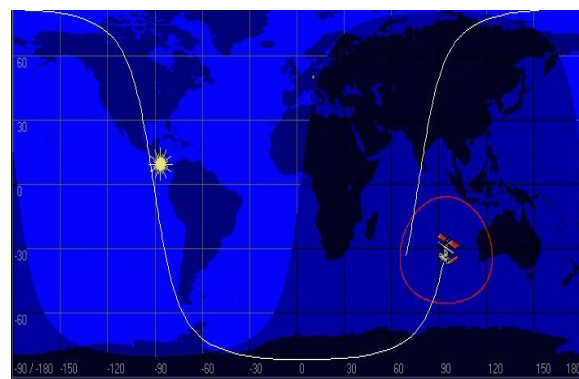
(b)
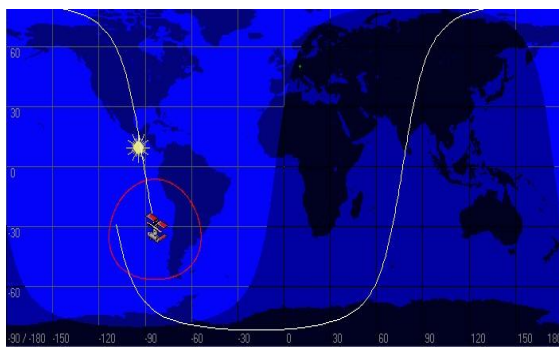
(c)
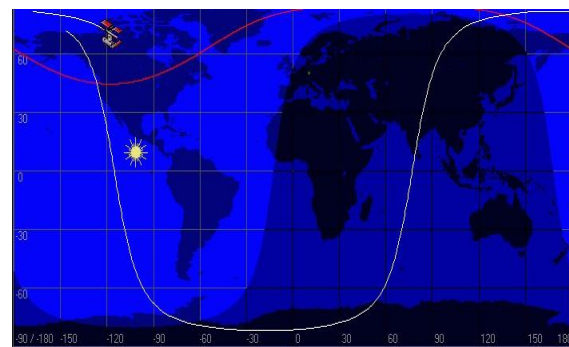
(d)

Figure 4.4 (a, b, c, d): Ground tracking of DubaiSat

The satellite follows the connected line, and the other line is the previous trajectory of this satellite before 90 minutes.

The satellite realizes about 16 rotations in the day. All these rotations are shown in the figure 2.5:

31
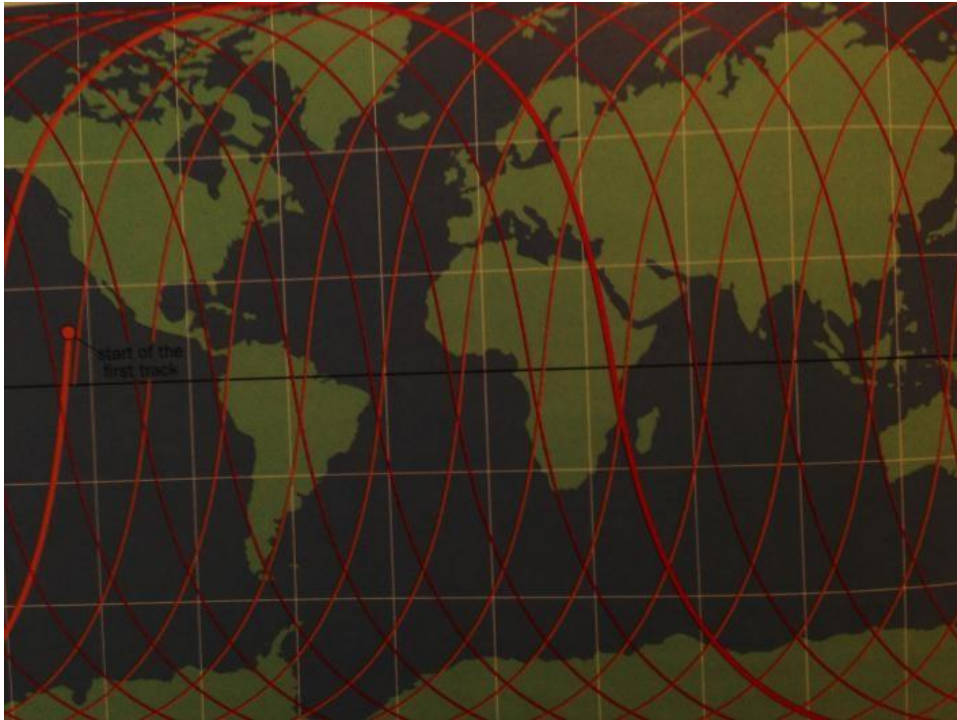
Figure 4.5: All rotations of the satellite around the earth during the day

5 Presentation Film

5.1 Payload Program

Earth Surveillance to take meteorological data to take meteorological data to monitor and estimate the energy supply potential of large scale photovoltaic energy plants to control a national alternative energy supply program.

The satellite shall be based on technology of Turkish Göktürk and DubaiSat-1

5.2 Presentation Film of MIS

After modeling, specification, prediction of position and tracking the satellite, we make a film of this mission simulation.

This film contains all results of the first part of the project (specification).

Screenshots of this film are represented in figure 5.1.
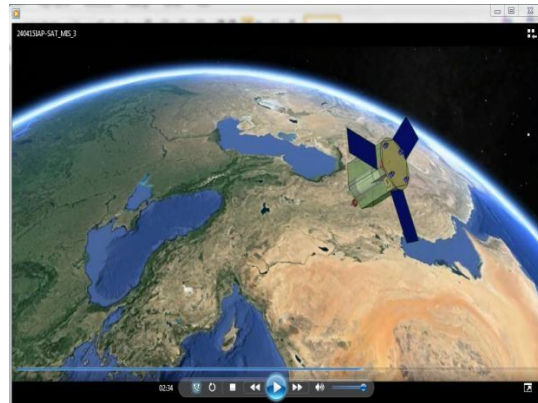
(a)   (b)

Figure 5.1 (a, b): Screenshots of the film

6 Hardware-in-the-loop (HIL) Testrig

6.1 HIL system architecture

Figure 6.1: System architecture of the satellite's function

IAP-SAT is launched; it attains the sun synchronous low earth orbit with a fixed orientation and starts to turn around the earth.

This satellite changes its orientation and its inclination due to disturbance and perturbation in the atmosphere. The electronic systems on board a satellite include attitude determination and control,

communications, command and telemetry, and computer systems. Sensors also refer to the electronic components on board a satellite.

The most important consideration that must be made in the design process is the environment in which the satellite systems and components must operate and endure.

One of the most vital roles that electronics and sensors play in the mission and performance of a satellite is to determine and control its attitude, or how it is orientated in space.

The orientation of a satellite varies depending on the mission. The satellite may need to be stationary and always pointed at earth, which is the case for a weather satellite. However, it may also be needed to fix the satellite on a given axis. The attitude determination and control system ensures the right and correct behavior of the satellite.

Another system which is vital to a satellite is the command and telemetry system, so much in fact, that it is the first system to be redundant. The communication from the ground to the satellite is the responsibility of the command system. The telemetry system handles communications from the satellite to the ground. Signals from ground stations are sent to command the satellite and precise the task that should be accomplished by the satellite, while telemetry reports back on the status of those commands including satellite vitals and mission specific data. [27]

6.1.1 Simulation on scilab/scicos

6.1.1.1 Parameters and equations used in the simulation

Equations of motion:

The only force applied on the satellite is the gravitation force: $F_{grav} = (G * M_{sat} * M_{Central}) / R^2$

Orbital Speed Equation: $v = \sqrt{\dfrac{G*Mcentral}{R}}$

The Acceleration Equation: $a = \dfrac{G*Mcentral}{R^2}$

Orbital Period Equation: $\dfrac{T^2}{R^3} = \dfrac{4*\pi^2}{G*Mcentral}$

With R= 6378.14 Km

   h= 680 Km

   G= 6.67*10^-11 N*m²*kg⁻²

   M$_{central}$ = 5,972 × 10^24 kg

So   *a= 7.9958 m/s², v= 7902.705 m/s and T=5071 s= 84.5 min.*

Attitude equals 680 Km

Inclination i is about 98.13°

Initial orientation: Teta(angle with X-axis) =0

Phi (angle with Y-axis) = 0

Psi (angle with Z-axis) = 0

This means that the initial position of the satellite is on the origin of the reference frame X, Y, Z.

All these parameters are used as inputs and variables in the equations used to simulate the satellite in scilab/scicos.

6.1.1.2 Modeling of the components of IAP-SAT in Scicos

To simulate our satellite, we have constructed the diagrams of IAP-SAT's components by referring to another satellites already simulated. [18]

The models in Scicos are based on the formulas in control engineering (Fig. 6.2).

Each diagram represents an algorithm and they are granted on these formulas. By transforming the variable (t) presenting in the equations, represented in the blocks, to another variable with the transfer function, we can obtain a block diagram.

Figure 6.2: Formulas in control engineering used in the simulation of IAP-SAT

I. Model of gyroscope

Gyroscopes are devices that sense rotation in three-dimensional space without reliance on the observation of external objects. [12]

Rate gyroscopes, also referred to simply as gyros, do not provide a measure of attitude, but are commonly used in satellite as they provide a measurement of angular velocity. Fusing the gyro with attitude measurements provides more accurate attitude estimates than the attitude measurements alone, with the degree of improvement dependent on a number of factors such as the sensor sampling frequency, satellite dynamics, and gyro noise characteristics. [14]
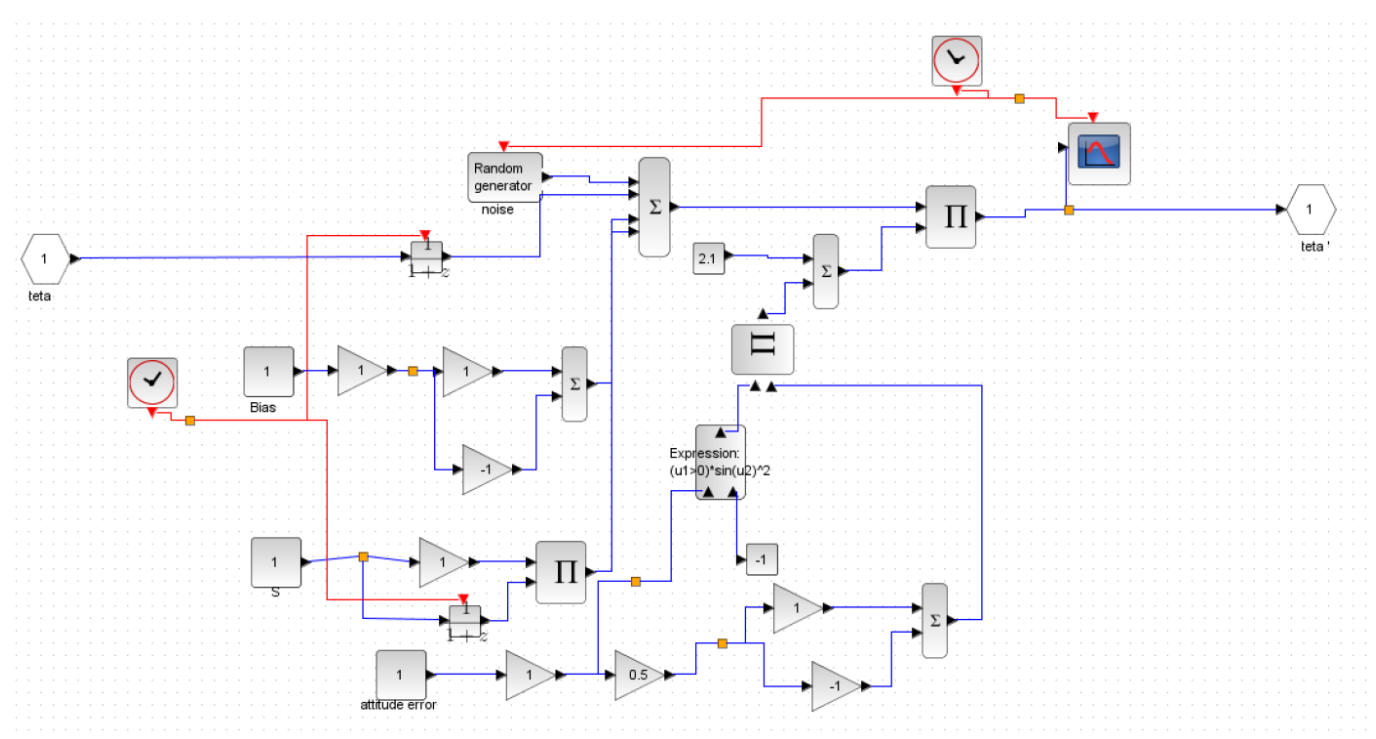
Figure 6.3: Diagram of gyroscope (representative of teta angle)
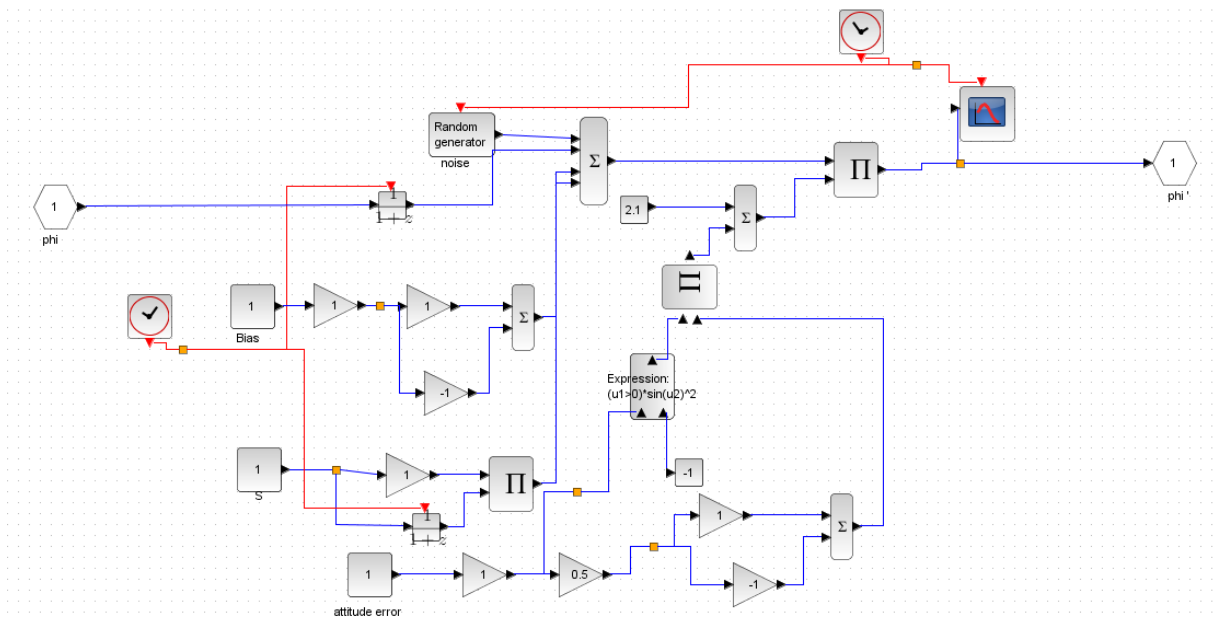
Figure 6.4: Diagram of gyroscope (representative of phi angle)

37

Figure 6.5: Diagram of gyroscope (representative of psi angle)

In the figures (Fig. 6.3, and 6.5), the input is the angle (teta with the x-axis, phi with y-axis, psi with z-axis) . These inputs are passed by attitude error, noise, bias, connected with mathematical expressions, to measure the final angular speed (how these angles change with time) as output of gyroscope teta', phi', psi'.

II.   Model of accelerometer

Accelerometer senses any difference in acceleration (non-gravitational accelerations) between the satellite and the earth, to determine the earth's gravity field from purely gravitational orbit perturbations. [19]

The main contributions of acceleration typically come from aerodynamic drag and lift, radiation pressure emanating from the sun and earth, residual linear acceleration caused by any offset between the sensor and the center of mass of the satellite, structural vibrations of the satellite and any attitude or obit actuation.

Instantaneous position of the proof-mass is measured by three capacitive sensors which permit a determination of the acceleration vector. [8]

Figure 6.6: Diagram of accelerometer (representative of X position)
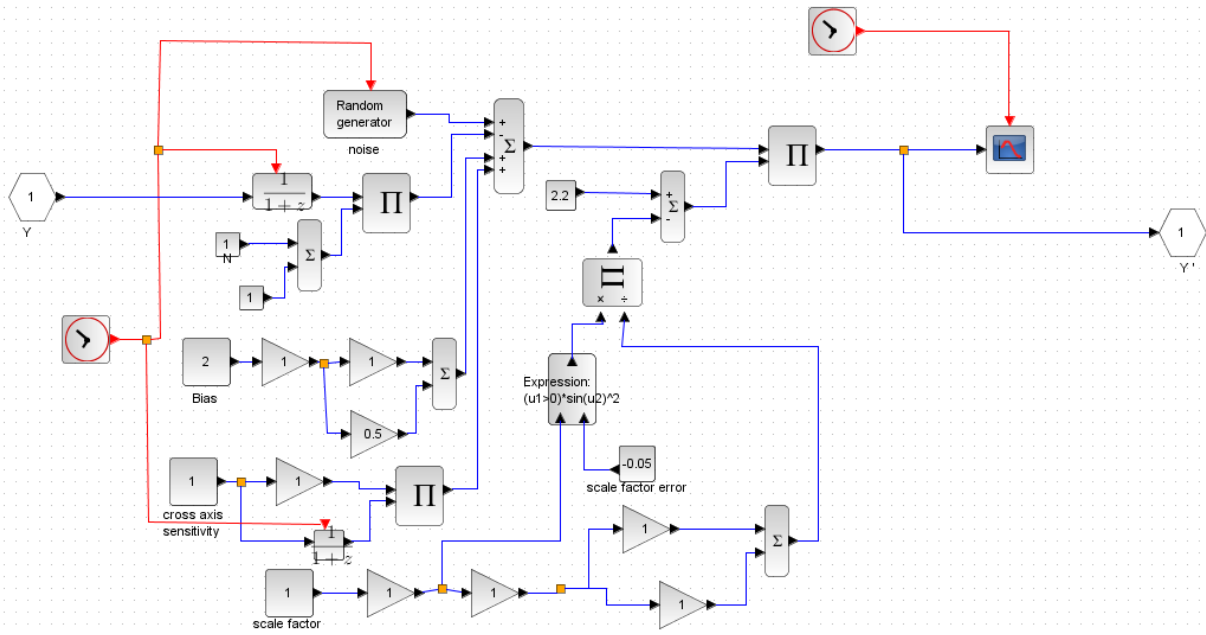
Figure 6.7: Diagram of accelerometer (representative of Y position)

39
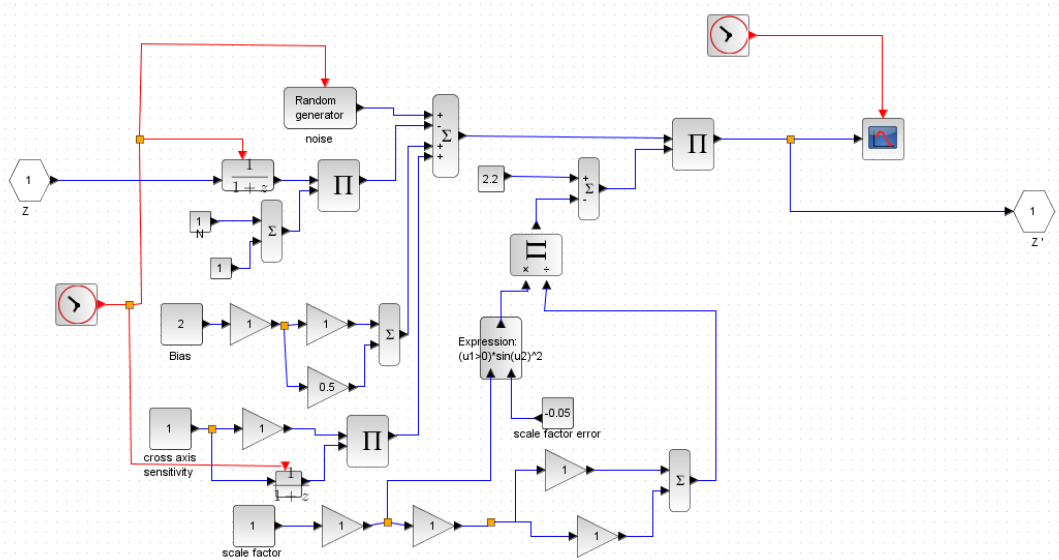
Figure 6.8: Diagram of accelerometer (representative of Z position)

The input of accelerometer (Fig. 6.6-6.8) is: the position of the satellite in the space (x, y, z); this position is changed depending of the noise, the scale factor, the cross axis sensitivity... to arrive to the measurement of the real position between the satellite and the earth.

The variation of this position with time can permit to obtain the acceleration vector.

III. Model of actuators

Satellites don't have to be adjusted often to maintain their orbits. The orbital of most artificial satellites degrade over time and are corrected from the ground (most satellites have small thrusters).

The purpose of a propulsion system is to change the velocity *v* of a satellite.

However this is more difficult for more massive satellite, therefore designers generally discuss momentum, *mv*. The amount of change in momentum is called impulse. So the goal of a propulsion method in space is to create an impulse. When launching a satellite from earth, a propulsion method must overcome a higher gravitational pull to provide a positive net acceleration. In orbit, any additional impulse, even very tiny, will result in a change in the orbit path.

The rate of change of velocity is called acceleration, and the rate of change of momentum is called force. To reach a given velocity, one can apply a small acceleration over a long period of time, or one can apply a large acceleration over a short time. Similarly, one can achieve a given impulse with a large force over a short time or a small force over a long time. [10]

40
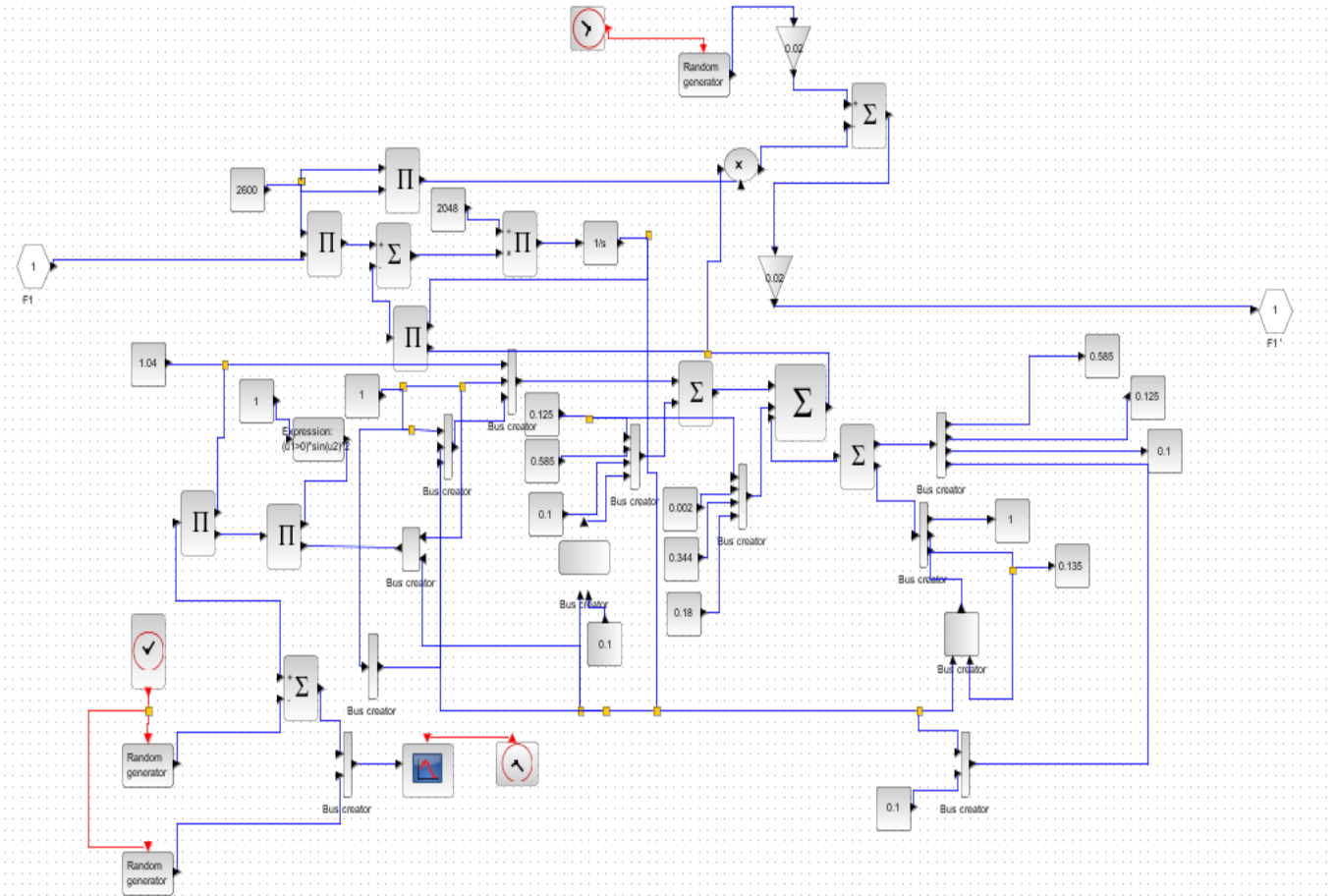
Figure 6.9: Diagram of Hall Effect propulsion (first propulsion F1)

As input of the Hall Effect thrusters (Fig.6.9), we have the force (rate of momentum); this input creates an impulse to reset the satellite in its correct position and the right orientation. The output of the actuators is the real values of this force after passing by the random generator, noise...

There are 4 actuators, each one has the same diagram, but with F1, F2, F3 and F4 as inputs.

IV.   Model of space

As we mentioned above, the most important consideration that must be made in the design process is the environment in which the satellite systems and components must operate and endure.
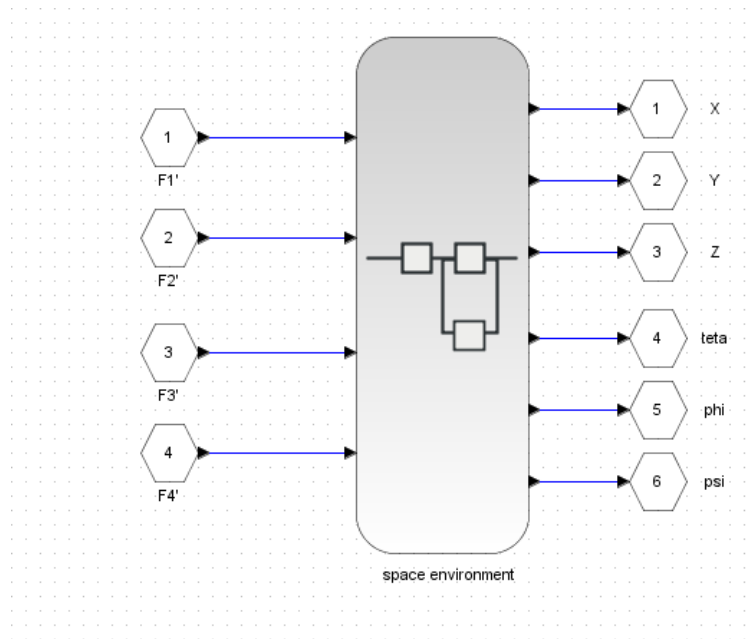
Figure 6.10: Diagram of the space environment

The diagram of the space environment is formed by four diagrams (Fig.6.10):

Diagram representing the positions and angles, another representing system dynamics (attitude and rate) , one represents the magnetic field and the last represents the sun vector. [6]
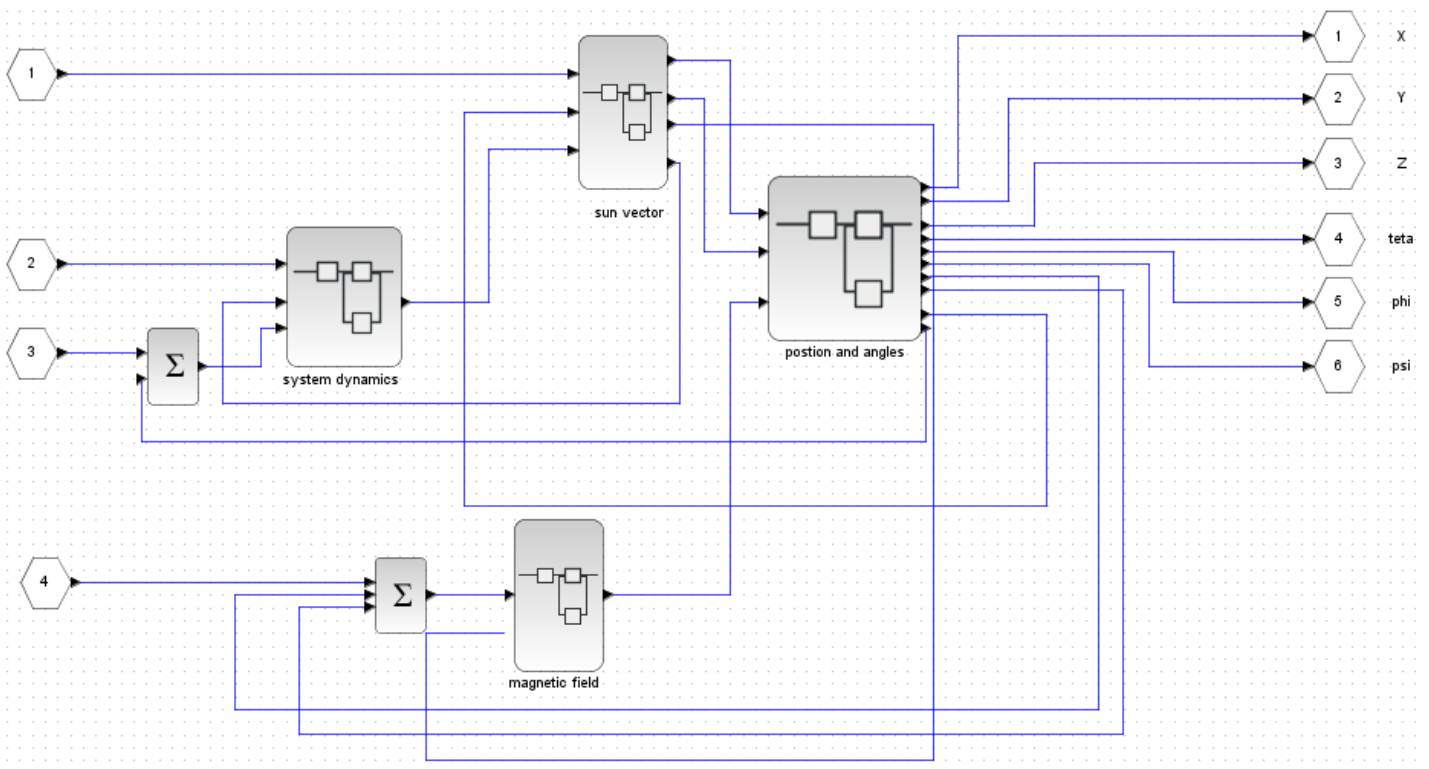
Figure 6.11: Diagram of the space environment with details of its components

The details of each related blocks in Fig.6.11, are shown in the next figures:

a- System dynamics
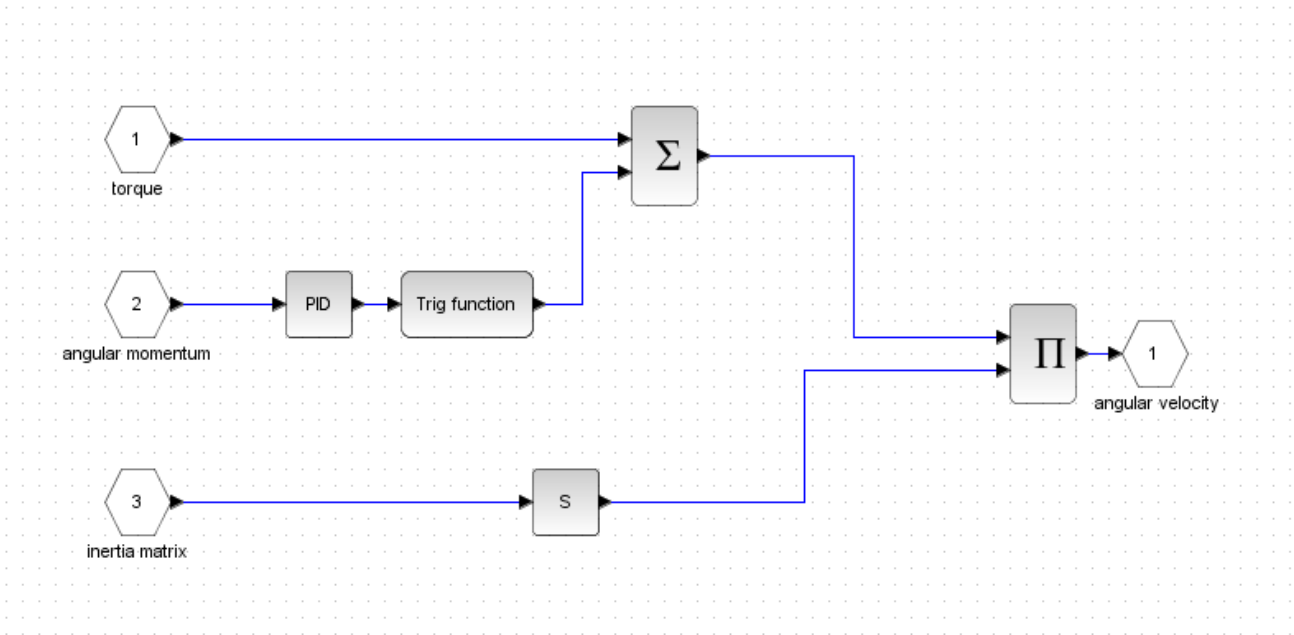
Figure 6.12: Diagram of the system dynamics

In the simulation of the system dynamic (Fig.6.12), the inputs are: the torque, the angular momentum and the inertia matrix.

This diagram gives as output the vector of the angular velocity.

b- Sun vector
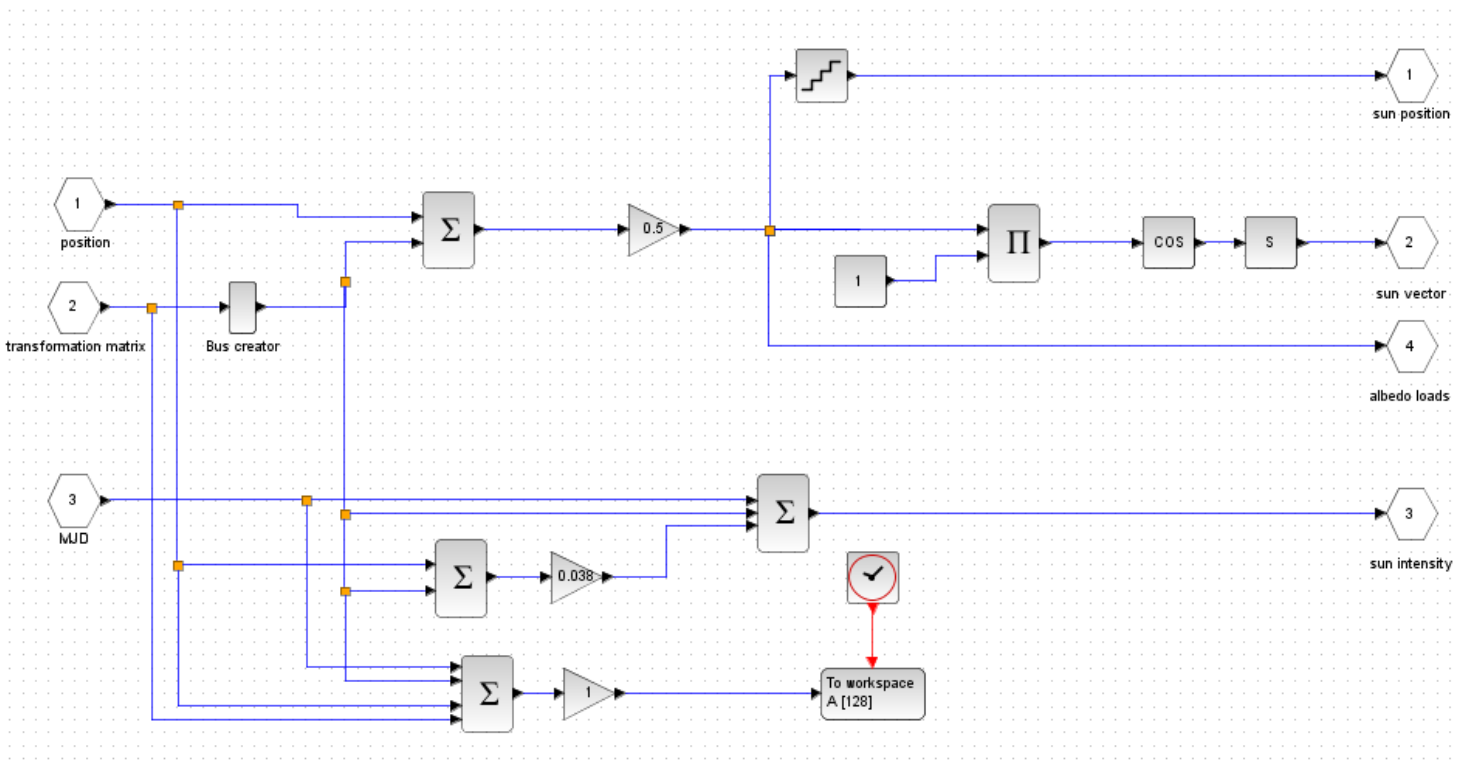
Figure 6.13: Diagram of the sun vector

In this diagram (Fig. 6.13), we have simulated the sun vector by entering as inputs the position, the transformation matrix and the MJD (Modified Julian Date). The outputs of this diagram are the sun position, the sun vector, the albedo loads and the sun intensity.

c- Positions and angles
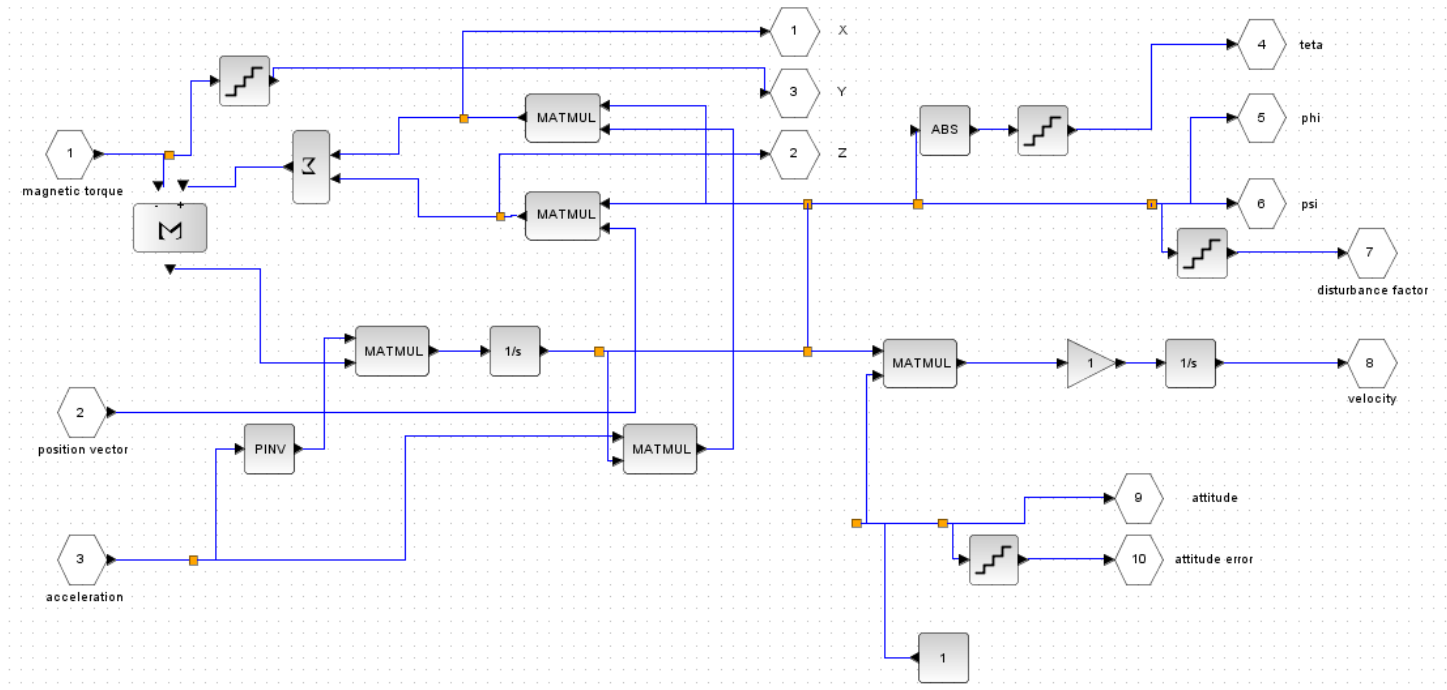
Figure 6.14: Diagram of the positions and angles

The diagram of positions and angles (Fig.6.14) has as inputs the magnetic torque, the position vector and the acceleration.

The outputs of this diagram are the position (X, Y and Z), the angles (teta, phi and psi), the velocity, the attitude, the attitude error and the disturbance factor.

d- Magnetic field
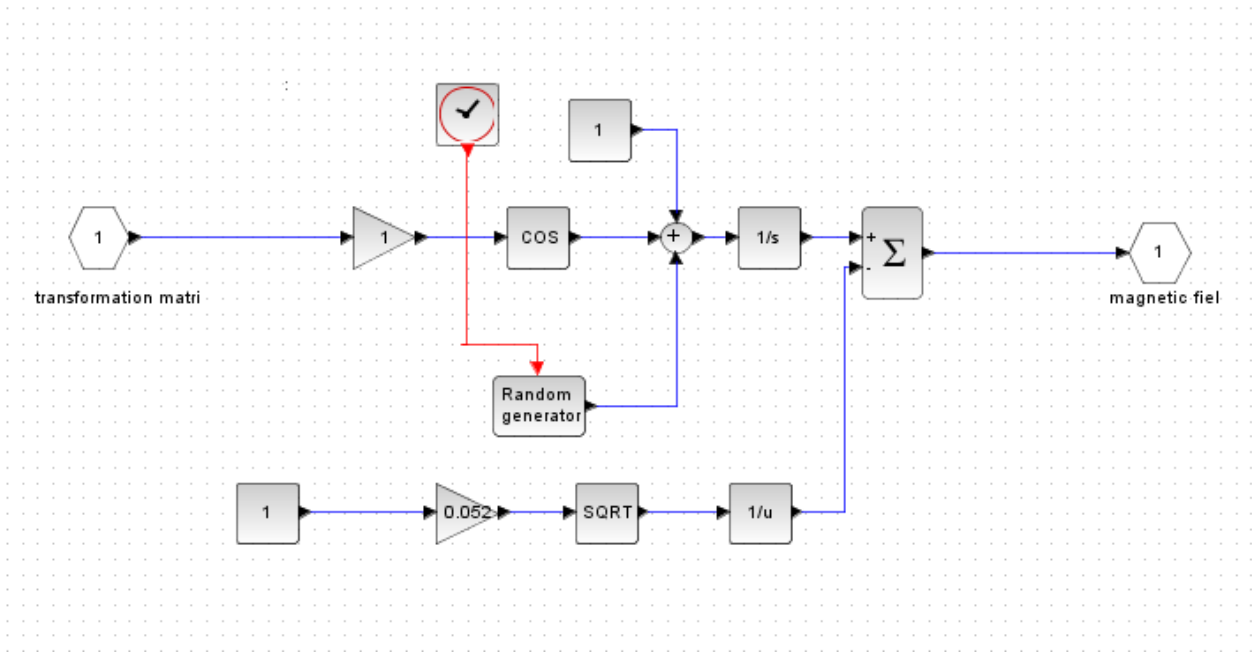
Figure 6.15: Diagram of the magnetic field

The magnetic field is simulated by using the transformation matrix as input, and we obtain as output the magnetic field (Fig.6..15).

V. Integration of diagrams

To simulate our satellite, all this models (model of space, actuators and sensors) will be integrated to obtain one diagram (Fig. 6.16).

In this diagram, we have integrated the four diagrams of actuators, the diagram of the space environment and the diagrams of sensors (three diagrams for the gyroscope and three for the accelerometer).

The sensors measure the values of position and orientation of the satellite from space, and give them real values.

By passing to the board computer, these values are checked and compared to the correspondent values.

If these values are not the fixed values to our mission, the corresponding actuators are activated to reset the satellite in the best situation.
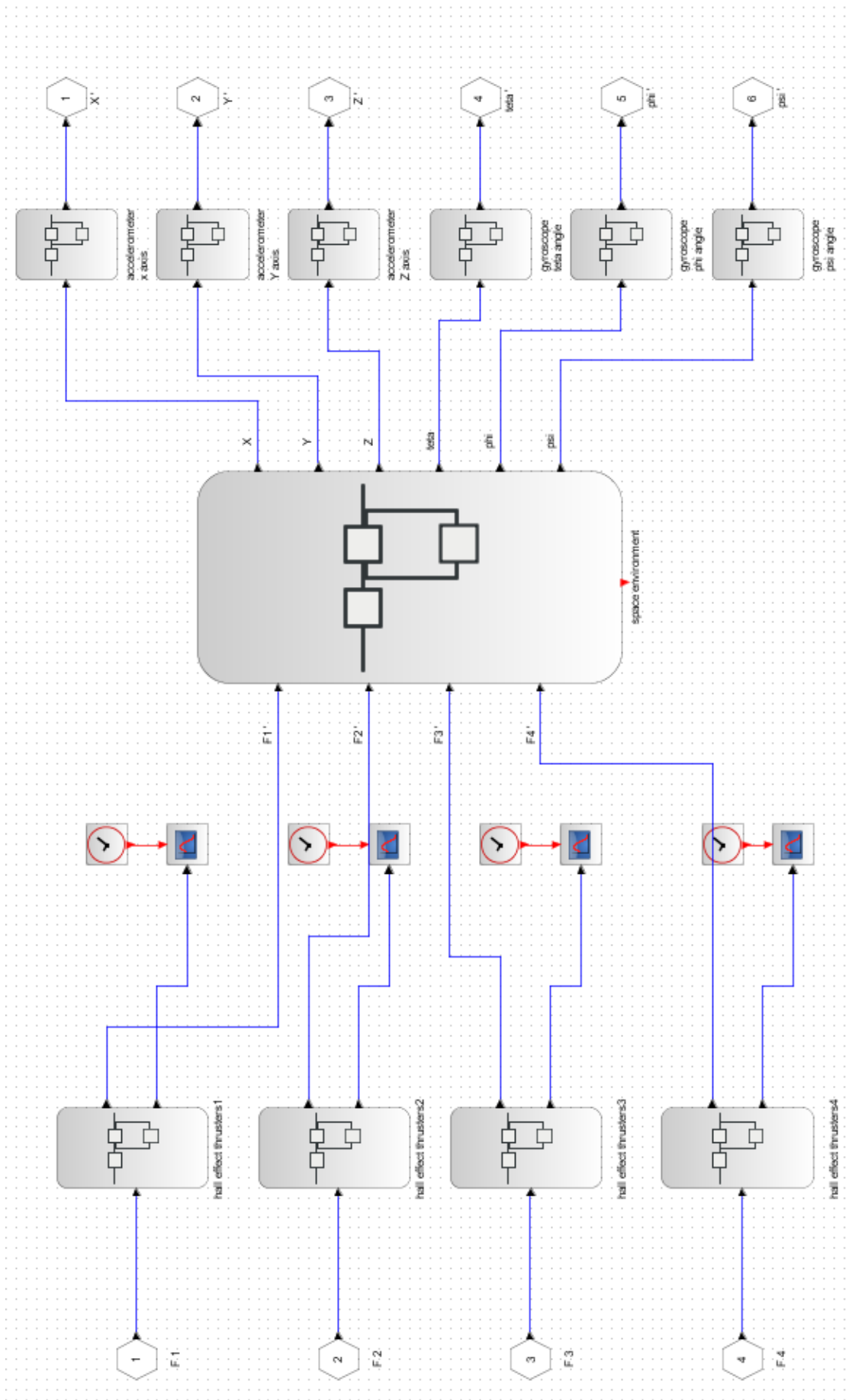
Figure 6.16: Integration of all diagrams (final model of simulation)

6.1.1.3 Test of simulation

Now we try to test our simulation, by giving different inputs (forces) to the actuators and see the difference in the result.

Many cases are experimented.

Case 1:

The first case is to consider that the last propulsion will not function, so the value of F4 will be zero, and the other forces will take each one the correspondent value to put the satellite on the requested orientation.
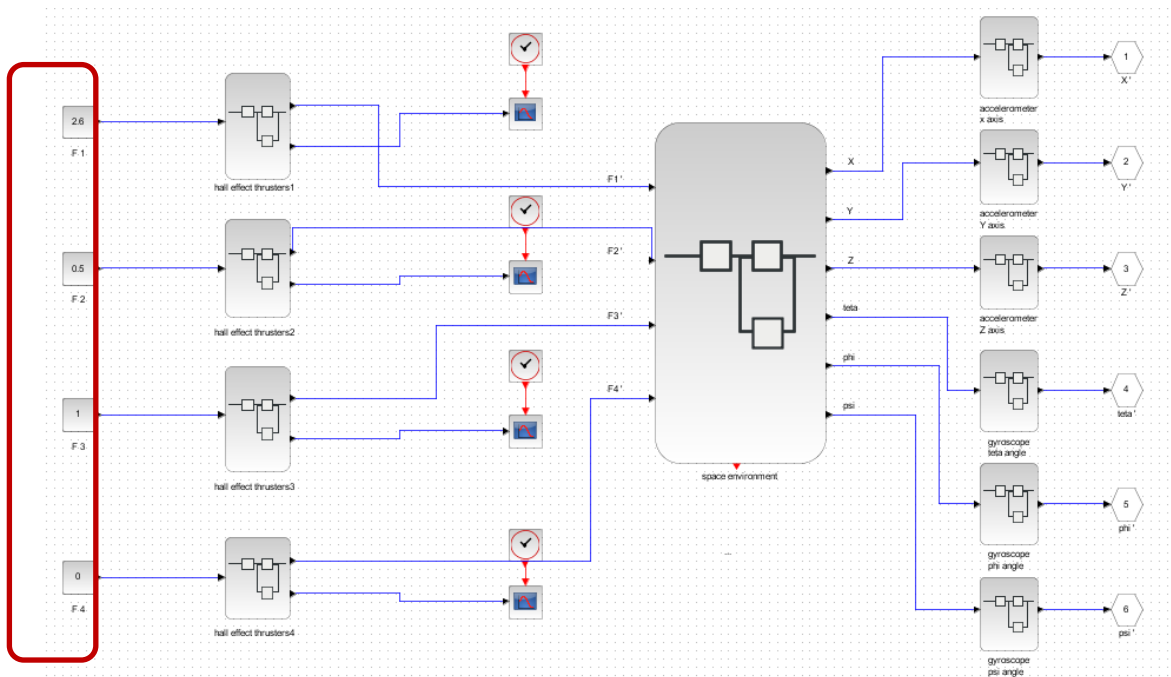
Figure 6.17: First case of the simulation's test

Case 2:

The second case is to consider that the input of the second propulsion will be a variable force (a step function), and the fourth force will also be zero.
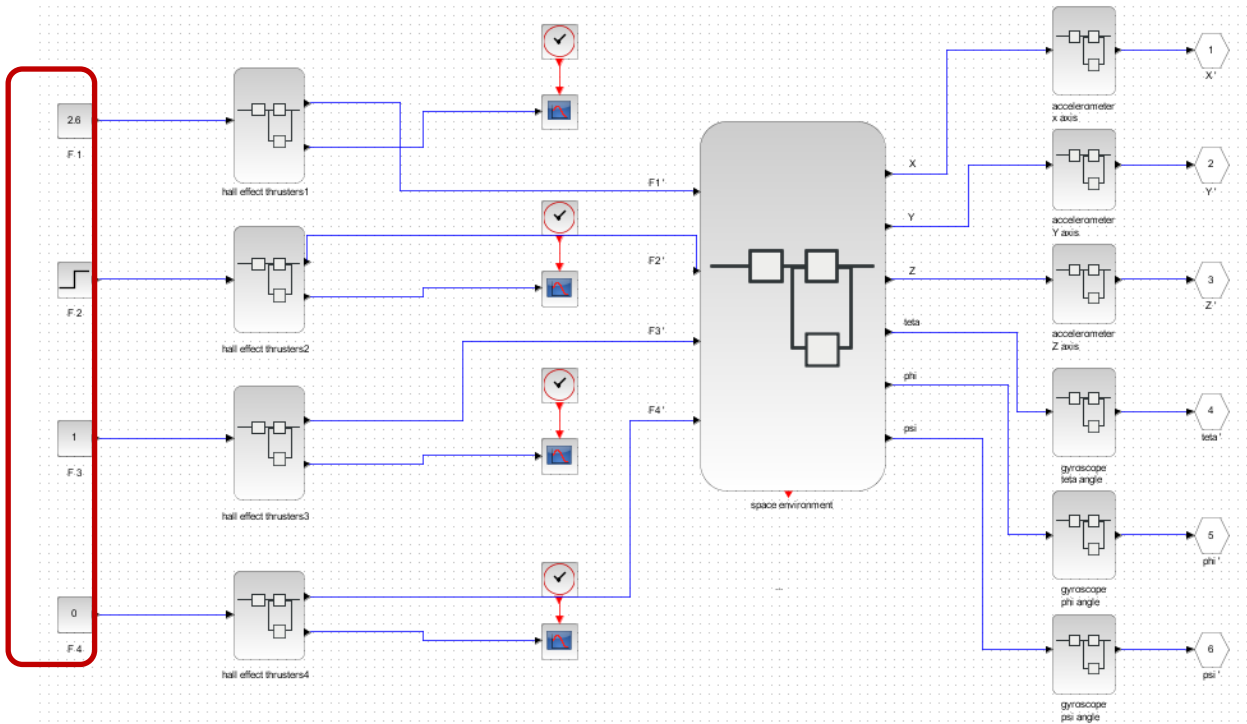
Figure 6.18: Second case of the simulation's test

Case 3:

This last case is to consider that it is necessary to activate the first propulsion (value of F1 different to zero), and the others have all the value zero for the force.
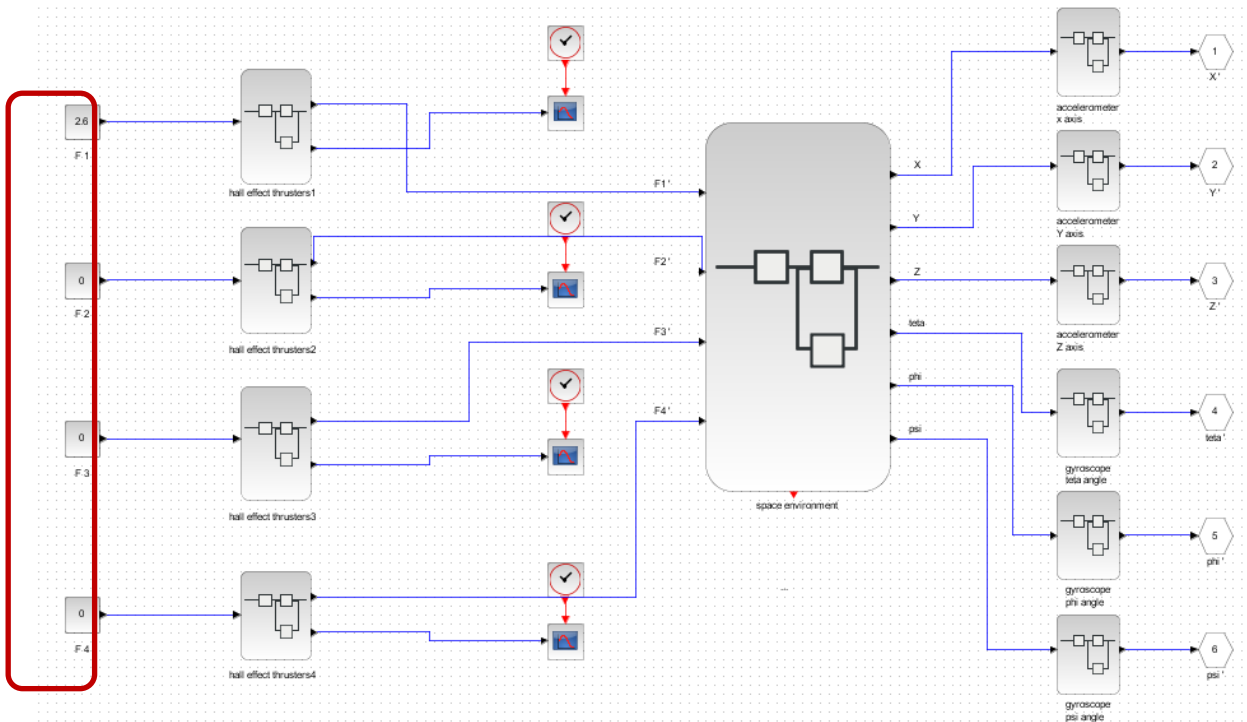
Figure 6.19: Third case of the simulation's test

6.2   Linking the software simulation on the server with the OnBoard-Computer

The sensors (gyroscope and accelerometer) take information about position, inclination and orientation of the satellite from the space environment and send this information to the computer board.

In the test of simulation (paragraph 6.1.1.3), we have manually changed the inputs of actuators to assay how the outputs of sensors change with the forces given by the propulsion. But in our system, this test will be realized automatically, by using the board computer.

This computer board is connected to the actuators. If the needed position and the orientation are not reached, the actuators are automatically activated to control these parameters.

By the action of the actuators, the space environment has a new position and orientation which are detected again by the sensors.

With this method, we arrive to control the attitude of the satellite on its orbit.

The modeling of the space environment, the sensors and the actuators is a software simulation realized with the scilab/scicos software on the HP server (paragraph 2.2.2). But the board computer is hardware, and it will be connected to this software simulation.

To join this software with the board computer and to attain the real control of the satellite, the board computer will read the outputs of sensors, to give the inputs for the actuators.

The computer board is activated, and we have connected this board with a screen, as shown in the figures 6.20, and the board should be connected to the internet.
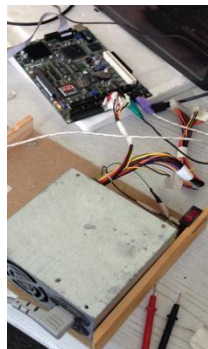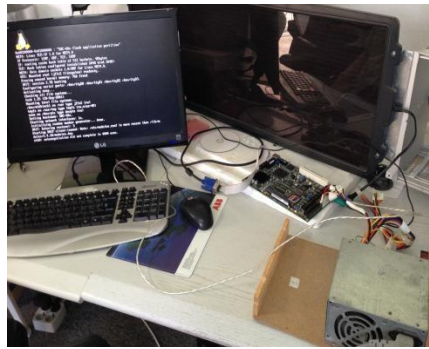
| (a) | (b) | (c) |

Figure 6.20 (a, b, c): Activation of the board computer

The simulation of the sensors, actuators and space environment is done on the HP

server. This HP server will be connected to the same network.

Figure 6.21: Screen appeared after the login to the server

We have to login firstly to the board computer in order to enter to the HP server.

Now all files on the server can be shown on the screen of board computer (Fig.6.22).
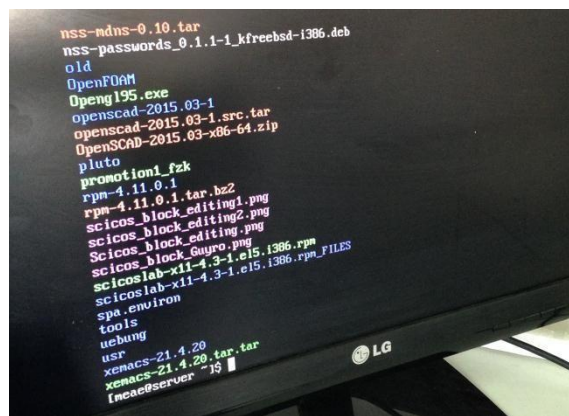
Figure 6.22: Appearance of all files present on the server on the screen of board computer

After this connection, we have to transfer the simulation in scilab/scicos to the board computer of the satellite by writing a C-code, because the board can't read the scicos file extension (.zcos).

The C-file will recognize the data in scicos file, and transform it to a format that can be read by the computer board.

Firstly, we arrive to read the inputs and the outputs of the scicos diagram from a C-binary file, by using the blocks in the palette of scicos.

Then, a method is necessary method to read this binary file as a txt file to link this software with the hardware.

Let's try to apply our method on a simple model simulated using Scicos (Fig.6.23).
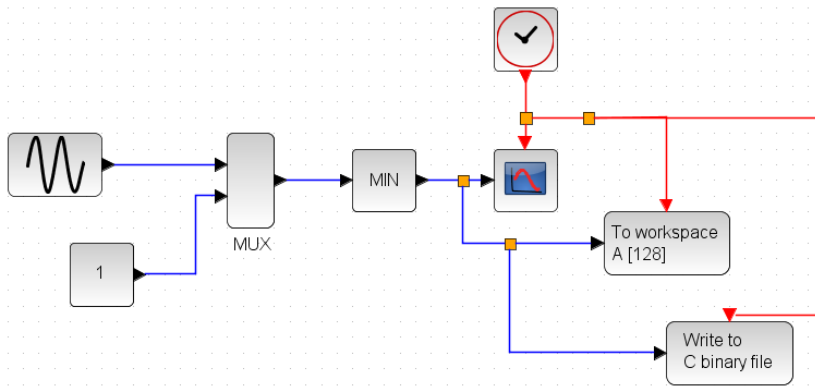
Figure 6.23: Example of a randomly simulation

The first step is to write the output of the file Scicos in a C binary file then to save this output in the workspace.

In the SciNotes, we write a code to run the simulation automatically (Fig.6.24).

Figure 6.24: code written in SciNotes

A binary file is created after the simulation, see fig.6.25.
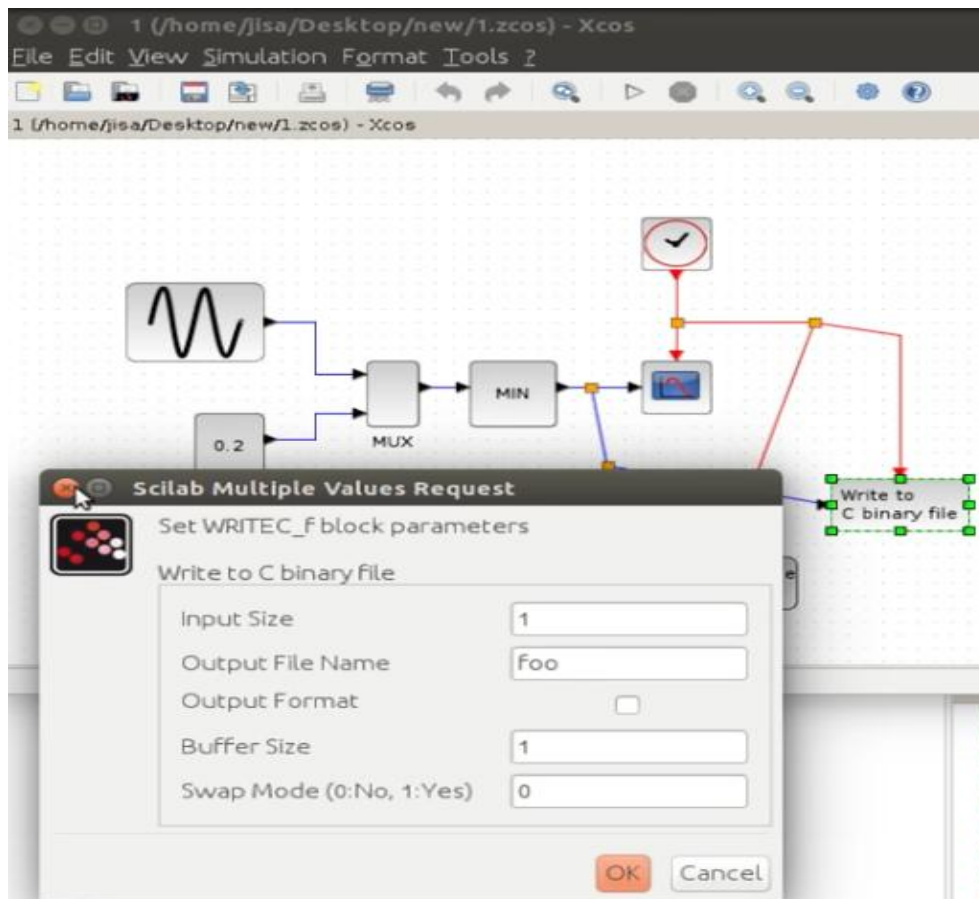
Figure 6.25: Output saved in the C-binary file (foo)

This file is named foo. Second step is to write the C-code (Fig.6.26).
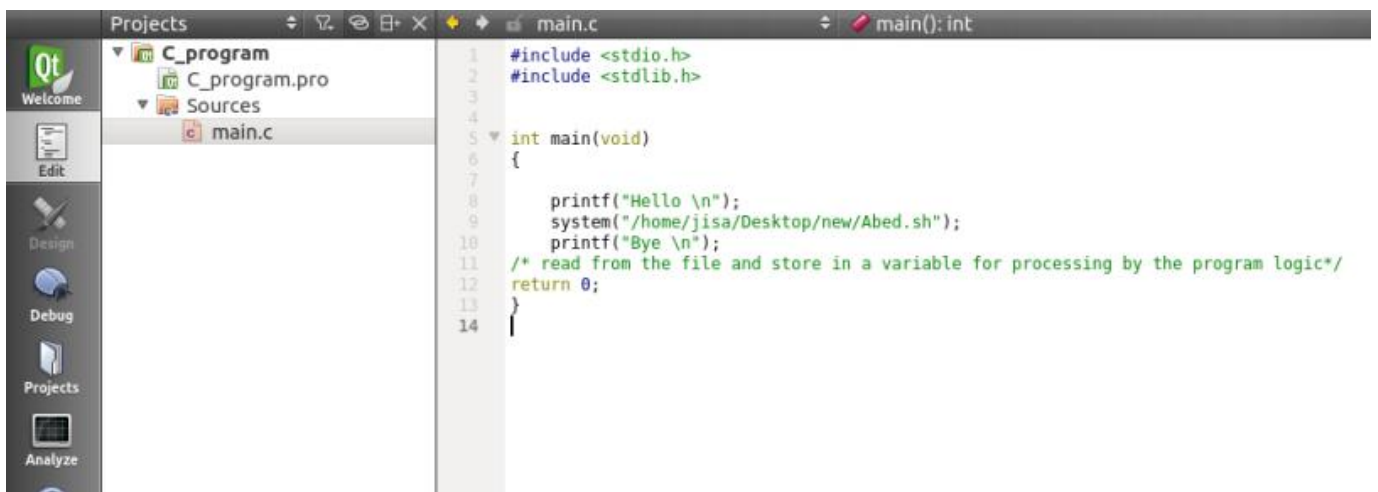
Figure 6.26: C-code written to integrate the hardware in the loop

In this code, we have a command line which let the program follows the commands written in a shell file: *(System ("/home/jisa/Desktop/new/Abed.sh").* This shell file is shown in the figure 6.27:

```
Abed.sh ×
scilab -f /home/jisa/Desktop/new/3.sce
echo Opening Scilab Simulation...
sleep 0.25
echo Simulation is Done...
sleep 0.25
echo foo \file is created...
sleep 0.25
cd /home/jisa/Desktop/new
sleep 0.25
echo Directory is Changed to Desktop\/new...
sleep 0.25
echo Converting foo from binary to text...
sleep 0.25
hexdump -v -e '1/4 "%f "' -e'"\n"' <foo> JAYCE
sleep 0.25
echo Converting is completed...
sleep 0.25
echo JAYCE.txt is created...
sleep 0.25
echo Sending File JAYCE.txt to arcom...
\\ you need to install sshpass
sleep 0.25
sshpass -p "arcom" scp -o StrictHostKeyChecking=no /home/jisa/Desktop/new/JAYCE arcom@192.168.1.100:/home/arcom
sleep 0.25
echo JAYCE.txt is sent...
```

Figure 6.27: Shell file containing all the steps to integrate the hardware

In this script, we have written all commands which conduct our program:

First command is to run the program scilab. Then, we write the directory of the file .zcos of simulation (the example file that i will transfer to the board computer) to open it.

The delay between two commands is 0.25s.

When the simulation of the file is done, a binary file foo (Fig.6.28) is created.

Figure 6.28: The binary file foo

So we write now in the shell file a command line to transfer this binary file to a txt file. This txt file is named Jayce (Fig.6.29).

53

Figure 6.29: The txt file Jayce

The final step is to send this txt file from my computer to the OnBoard-Computer. This OnBoard-Computer is an arcom board.

We write in the shell file a command line (sending File Jayce.txt to arcom).

This board is secured by a password, so we install sshpass (secure shell file), to permit the transfer without asking the passcode each time.

Then, we have to define the directory of the txt file (Jayce) and the directory on arcom where we want to send this file.

When we arrive to send the file of sensor's output to the board computer and take a file as inputs for the actuators from this board, so we attain the end of the satellite bus simulation.

6.2.1 Helping in building a program that simulates the satellite's movement in a space environment

It was to write a program that provides the linkage between the sensors, processor, actuators and the space environment such that the mediators are text files for every connection between 2 devices.

Unfortunately, the function "Write to output file" was obsolete in the newest Scilab version we were using (5.5.2) so we had to try another way to provide the linkage.

54

Scilab manual >> xcos > palettes > Sinks palette > WFILE_f

WFILE_f

Write to file

Block Screenshot

Then we tried the " Code generation" feature of Xcos in order to transform our control block diagrams into C code then implement the "C File Processing Functions " to write inputs/outputs to a text file.

However we failed to compile the generated standalone code due to linker errors. Even after intensive research in programming forums and adding linkers for the designated libraries to the compiler… all the attempts to debug the errors were in vain.

Line	File	Message
		[Linker error] undefined reference to `cstblk_'
		[Linker error] undefined reference to `cstblk_'
		[Linker error] undefined reference to `summation'
		[Linker error] undefined reference to `cscope'
		[Linker error] undefined reference to `cstblk_'
		[Linker error] undefined reference to `cstblk_'
		[Linker error] undefined reference to `summation'
		[Linker error] undefined reference to `cscope'
		[Linker error] undefined reference to `cscope'
		[Linker error] undefined reference to `cstblk_'
		[Linker error] undefined reference to `cstblk_'
		[Linker error] undefined reference to `summation'
		[Linker error] undefined reference to `cscope'
		ld returned 1 exit status

6.2.2 Working with shell programs to maintain the feedback system

The work was conducted with the help of a Linux specialist.
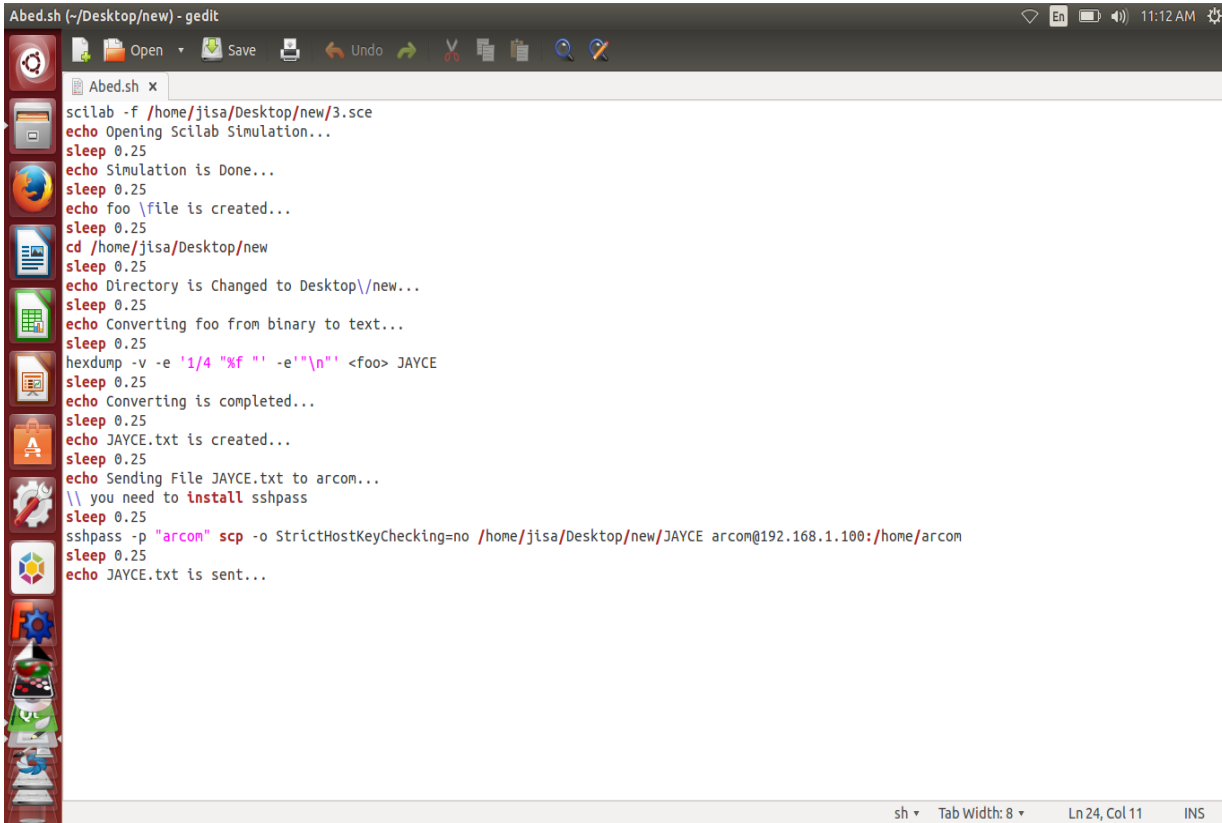
A shell script is a computer program designed to be run by the UNIX shell in order to execute designated commands.

The solution which is manifested in the shell script for the feedback system was as follows:

1) write the output of the sensor program in a text file but in binary form (writing the output of the simulation in binary format is provided by WRITEC_f block in XCos)

2) convert the binary file into float format using the shell command hexdump

3) send this file to the Arcom board which has an embedded linux operating system and which is mounted on the satellite

4) execute the C program on the arcom board which processes the sensor data then sends commands to the actuators.

The first successful prototype for the shell program was the following:

where 3.sce scilab program opens the Xcos simulation and saves the result in binary format to foo.txt file.

6.3 Receiving meteorological data

This satellite bus is connected to the ground station by the communication system and telemetry, and the ground station is also connected to the payload of satellite by telemetry.

From the payload system, we obtain image of the meteorological data.

The camera takes photo for the earth's surface, and sends this data to the ground station as signals. The ground station receives these signals and decodes it to obtain the represented image of weather.

For example, let's see the meteorological satellite NOAA followed by the ground station (Fig. 6.30).

NOAA turns around the earth on a low earth orbit. The prediction of its position in function of time is shown by the program Gpredict, and it turns under the frequency 137.915.5 MHz:

NOAA 18

Azimuth : 87.71°
Elevation : -47.44°
Slant Range : 10518 km
Range Rate : -3.247 km/sec
Next Event : AOS: 2015/08/13 11:59:50
SSP Loc. : PI75VP
Footprint : 6303 km
Altitude : 867 km
Velocity : 7.414 km/sec
Doppler@100M : 1083 Hz
Sig. Loss : 152.84 dB
Sig. Delay : 35.08 msec
Mean Anom. : 144.14°
Orbit Phase : 202.70°
Orbit Num. : 52920
Visibility : Daylight
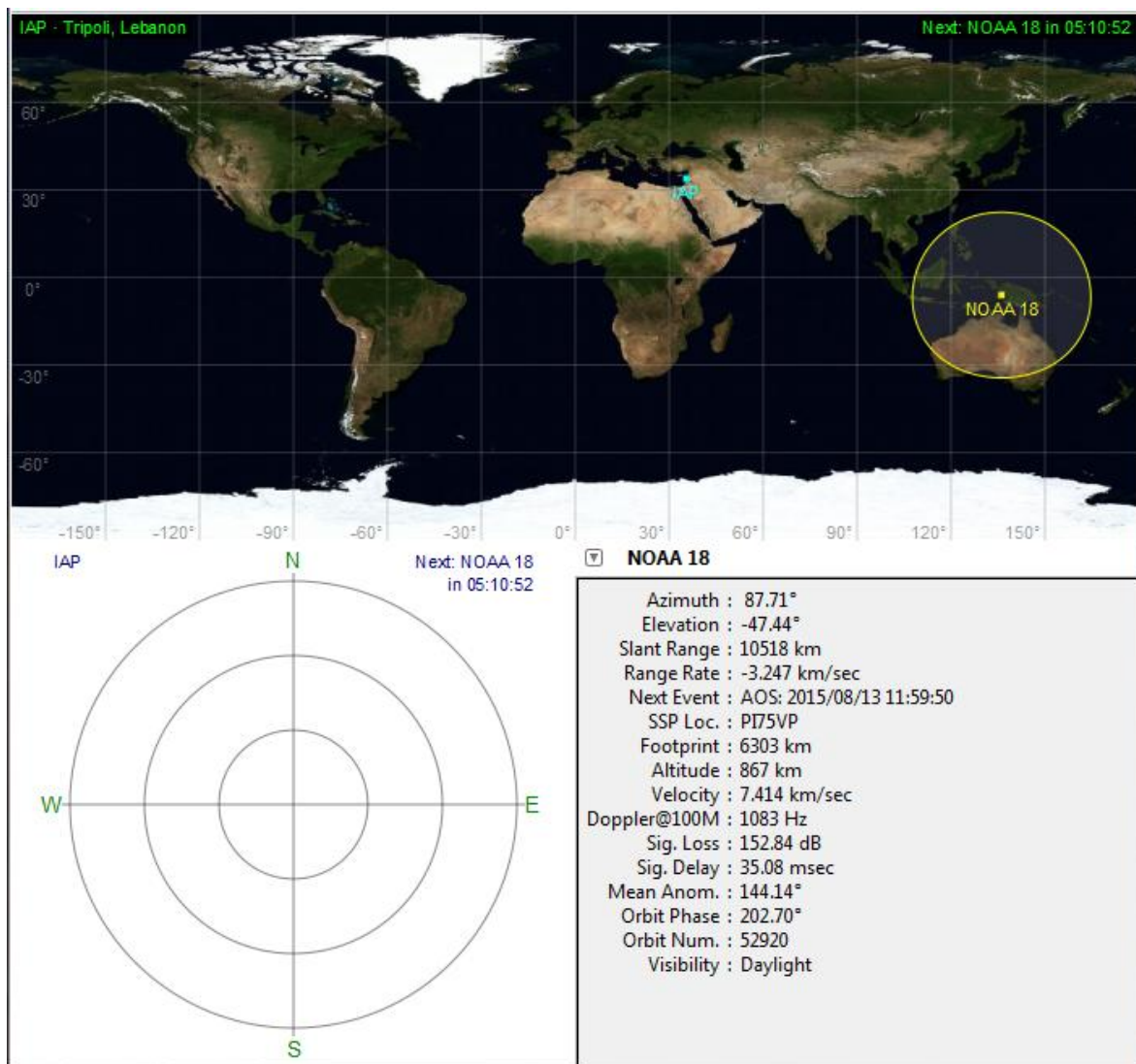
Figure 6.30: Gpredict following NOAA-SAT

How to receive and read this signal?

We need as hardware:

- An antenna to detect the frequency of satellite

- A RTL

- Our computer

As software:

- Zadig

- ExtIO_RTL2832

- HDSDR   program

- WXtoImg program

First, we have to plug the USB dongle receiver RTL2832 (Fig.6.31), and to install the driver.

RTL:

(a)   (b)

Figure 6.31 (a, b): The RTL

RTL-SDR is the cheapest method to get into the world of SDR (Software Defined Radio) - method of receiving (RTL-SDR has no capability to transmit) and processing radio transmission using your PC. Using it you can receive everything that is transmitted between about 20~2000 MHz depending on second chip on the device.

In brief it can hear i.e.:

- Ordinary FM radio stations

- Police radio transmissions

- Plane and ship tracking information

- GPS and meteorological satellites

- International Space Station and other space equipment that transmits below 2 GHz - every other device that you know transmits on these frequencies. [7]

Next we have to download Zadig.

Zadig is a Window application that installs generic USB drivers, such as WinUSB, to help you access USB devices (Fig. 6.32).
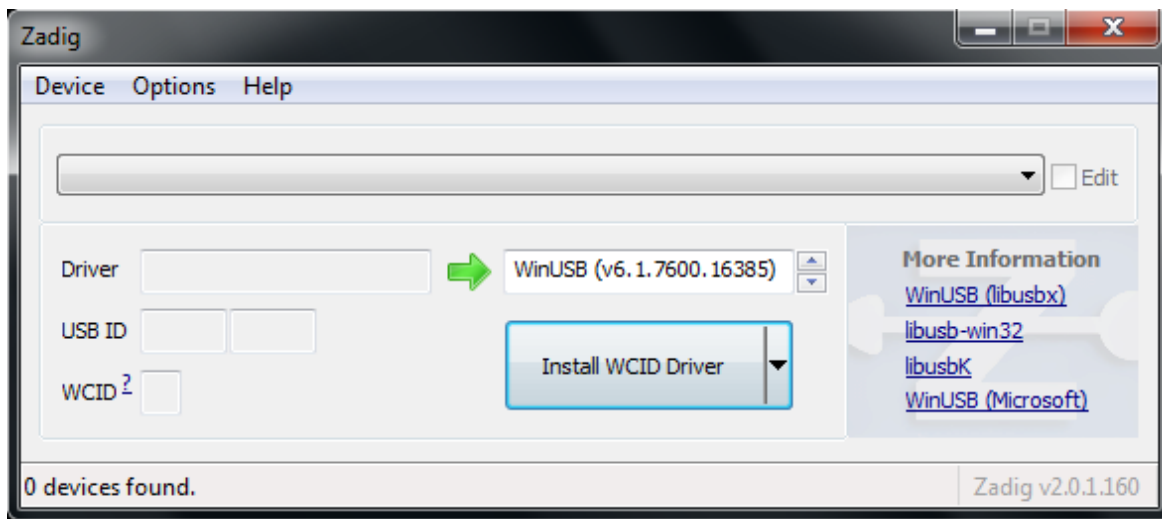
Figure 6.32: Installation of Zadig

We searched for devices, to install the driver RTL.

After finding devices, we choose the name of this device, Bulk-In, interface (Interface 0), then we click reinstall device.
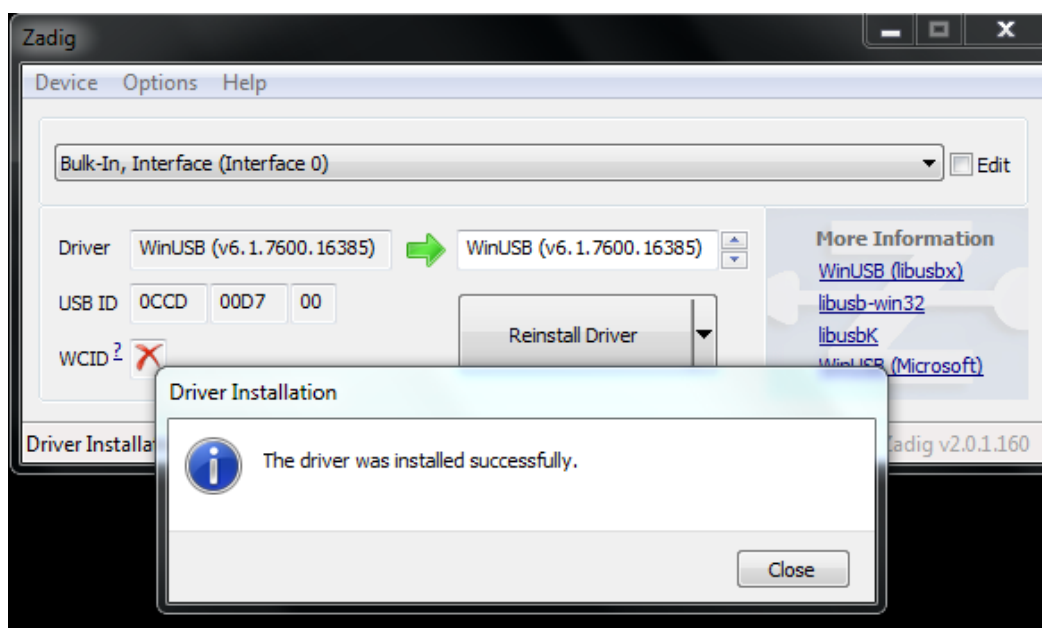
Figure 6.33: Installation for the driver RTL in Zadig

The next step is to download ExtIO_RTL2832. It is a software defined radio program that supports the ultra-cheap RTL-SDR.

The HDSDR software (Fig.6.34) doesn't communicate with the SDR hardware directly. It communicates with the SDR radio hardware through an External Input Output Dynamic Link Library (ExtIO-DLL) file, which is a type of plug-in.

Then, we download the software HDSDR (High Definition Software Defined Radio).

Figure 6.34: HDSDR

Before starting this program, we have to copy ExtIO_RTL2832.DLL into our HDSDR installation directory (default=C:\Program Files (x86)\HDSDR).

Now, we open HDSDR and select ExtIO_RTL2832.DLL:
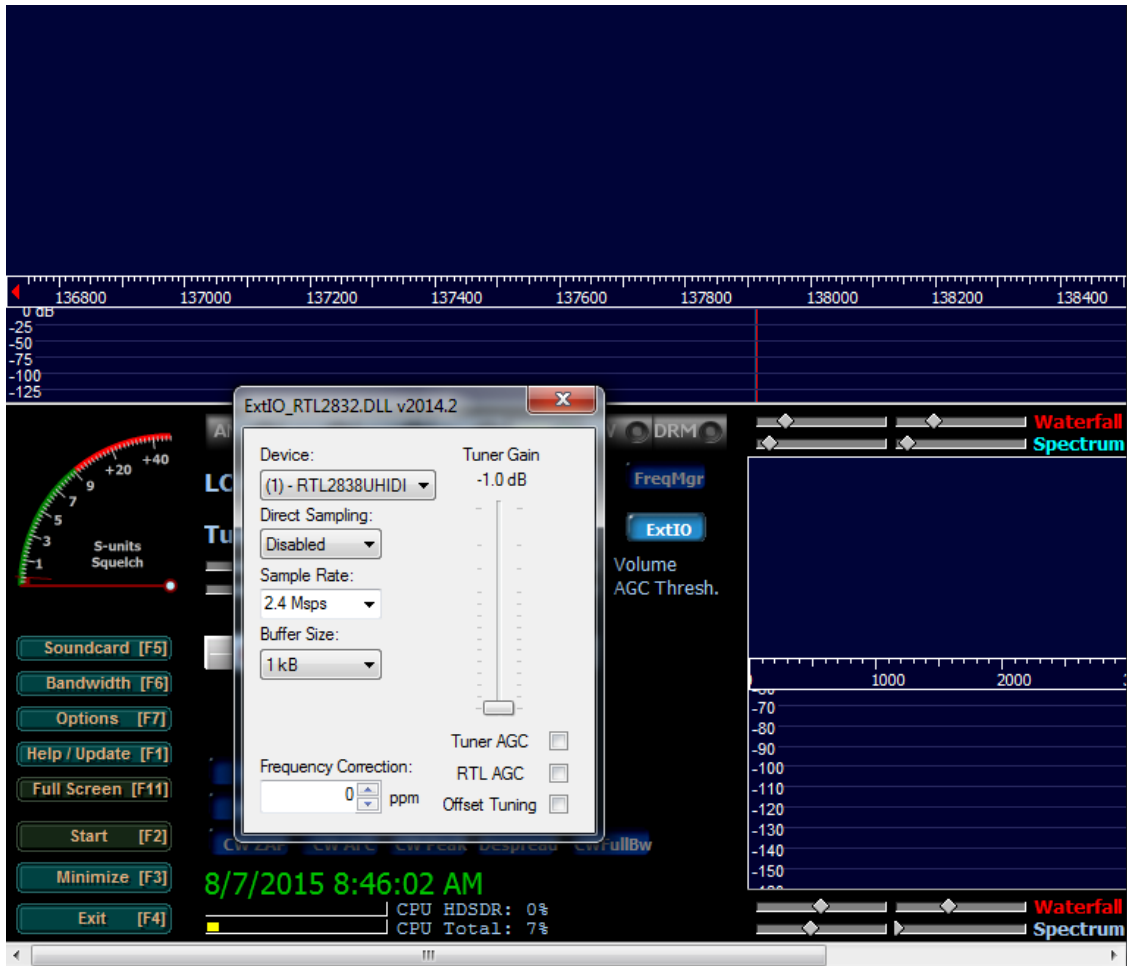
Figure 6.35: Choosing the driver RTL in HDSDR

We verify our driver RTL2832, and we fix in HDSDR the request frequency (to receive meteorological data from NOAA satellite, the request frequency is 137.915.5 MHz).

By using an antenna (Fig. 6.36), we can receive a real signal from the NOAA satellite, and visualize this signal in HDSDR.

Figure 6.36: Antenna multidirectional used in our experience

We also install the program WXtoImg (program which transferred the signal to a meteorological image). WXtoImg (Fig. 6.37) is a fully automated APT (automatic picture transmission) and WEFAX (weather fax) weather satellite decoder. The software supports recording, decoding, editing, and viewing. It supports real-time decoding, map overlays, advanced color enhancements, 3-D image [31]

When we start WXtoImg, it will detect the satellite NOAA, because of fixing its frequency in HDSDR.
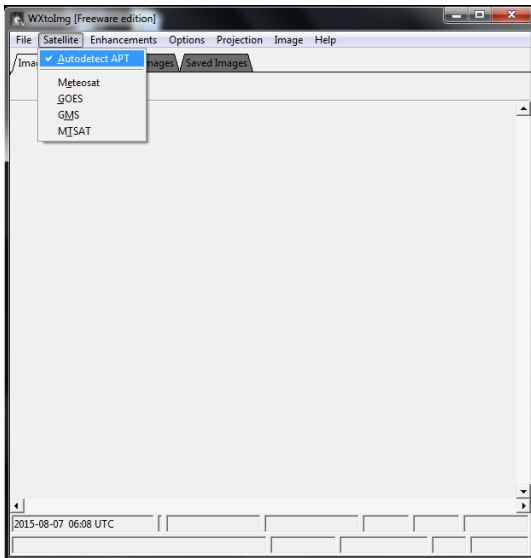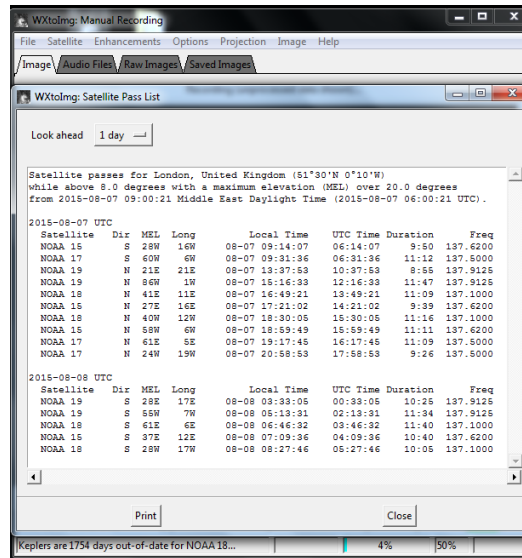
(a)   (b)

Figure 6.37 (a, b): WXtoImg program

7 Operation of HIL test rig

7.1 Results of the Simulation in Scicos

7.1.1 Simulation of gyroscope

When we execute the diagrams which represent the simulation of gyroscope, the results are shown in the next figures (Graph 7.1, 7.2 and 7.3):
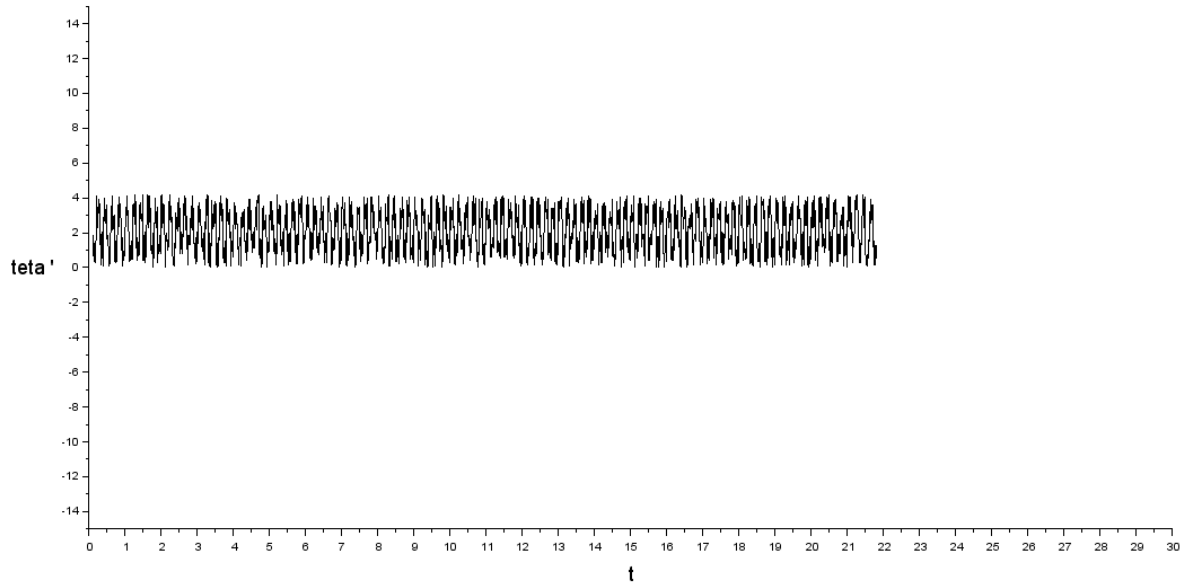
Graph 7.1: Variation of teta' (angle with x-axis) function of time
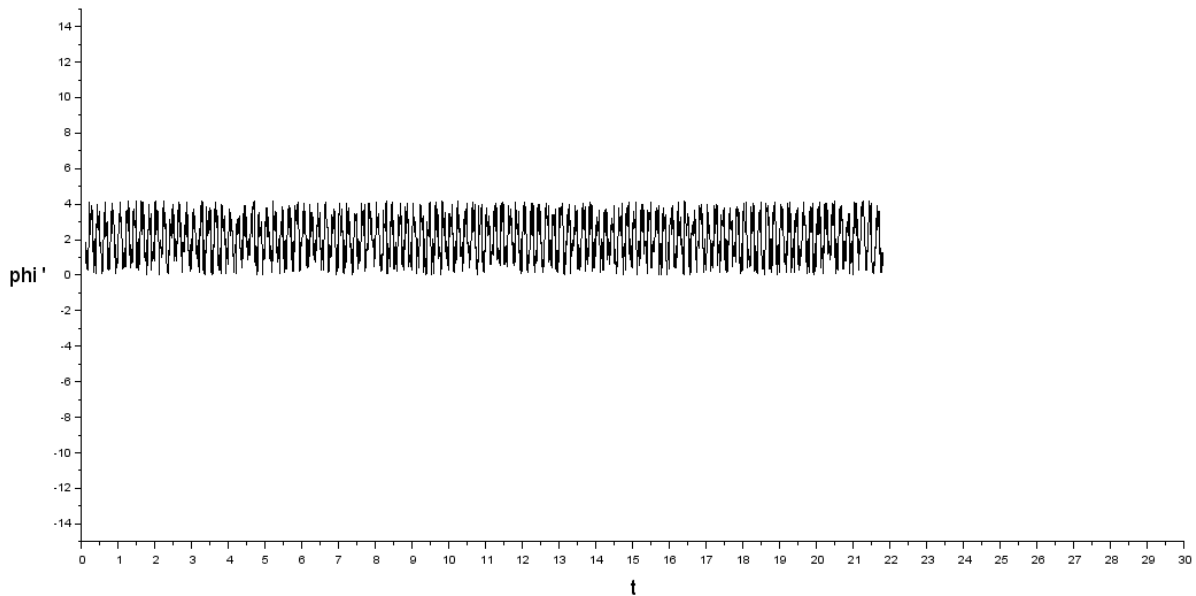
Graph 7.2: Variation of phi' (angle with Y-axis) function of time
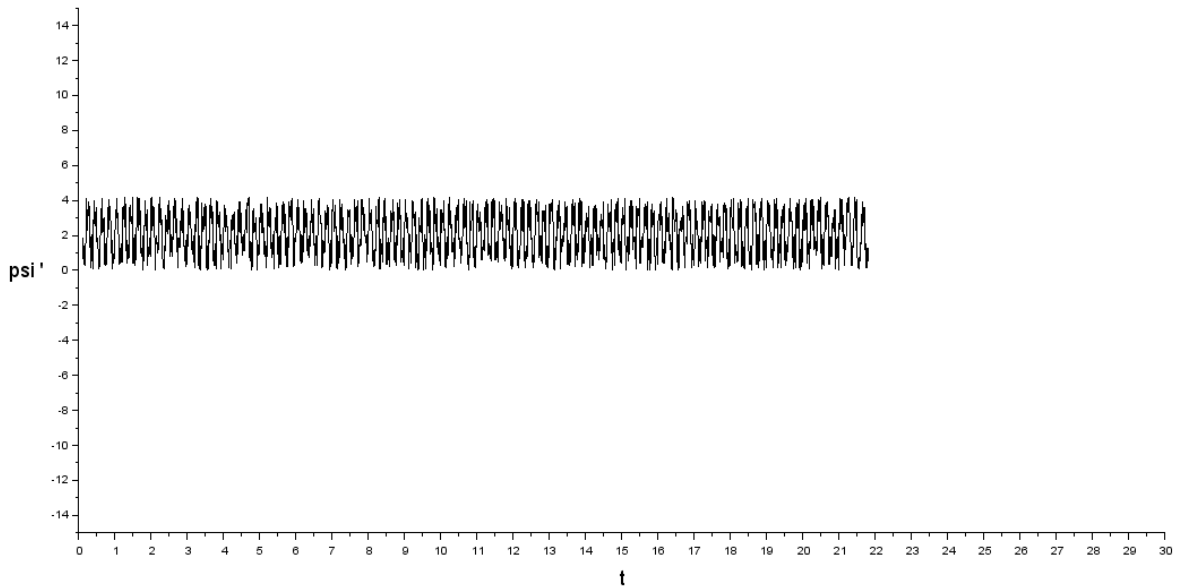
Graph 7.3: Variation of psi' (angle with Z-axis) function of time

These graphs represent the simulation of the diagrams represented in the figures (2.8, 2.9, 2.10).

It is obvious that the values of the angles with the X-axis (Y-axis and Z-axis) change, they are not the same values entered as inputs to these diagrams.

As we see in those graphs, the angle teta (phi, psi) changes to be teta' (phi', psi'), after passing by the bias, the attitude error, the random generator...

7.1.2 Simulation of accelerometer

In this section, we will see the results obtained after executing the diagrams representative of the accelerometers simulation.
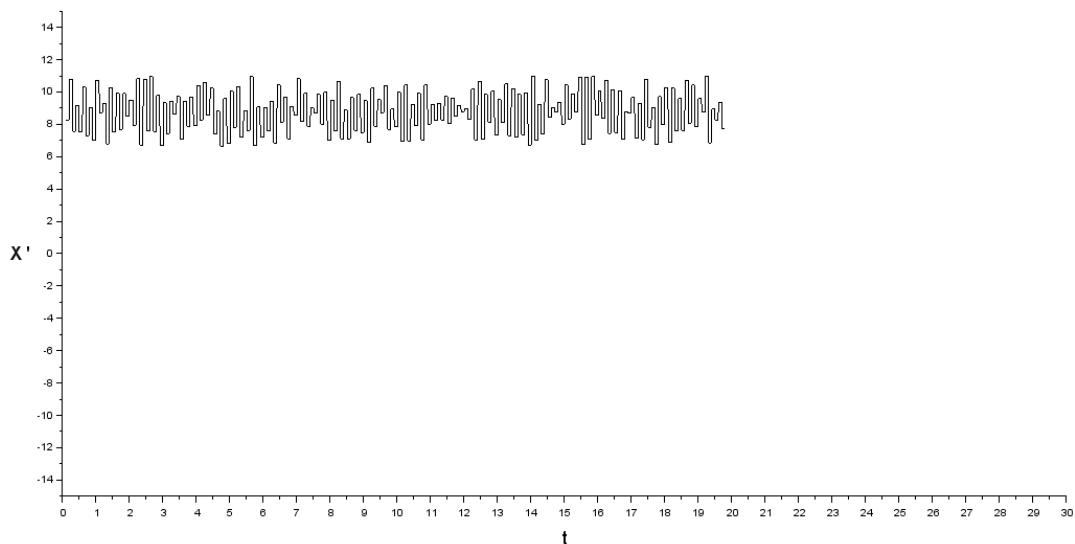
Graph 7.4: Variation of the position X' with time

64
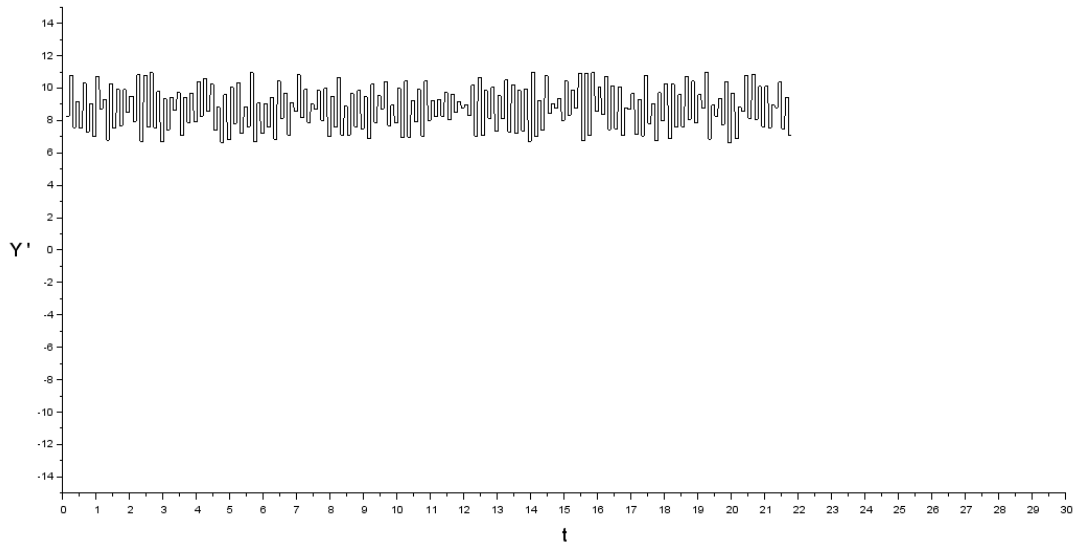
Graph 7.5: Variation of the position Y′ with time
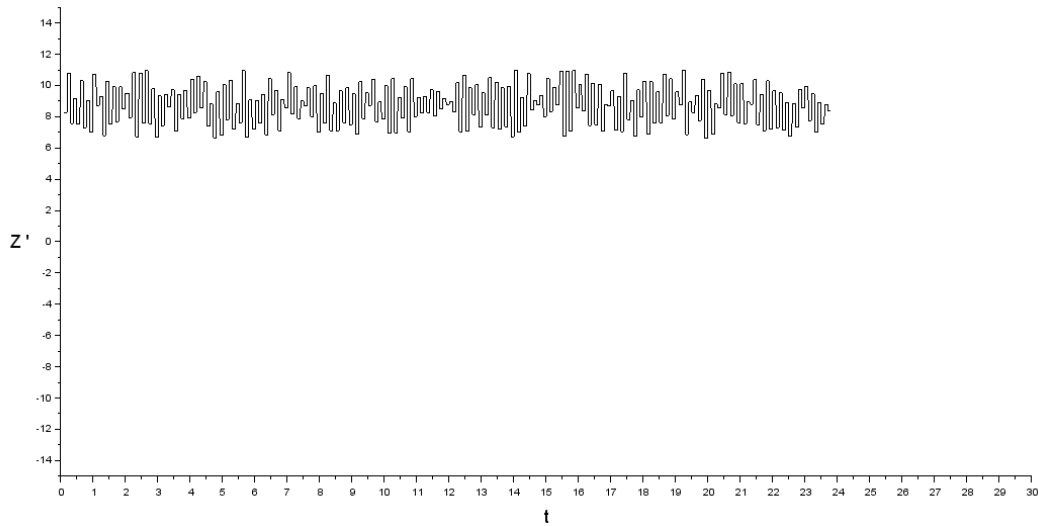
Graph 7.6: Variation of the position Z′ with time

In these graphs (Graph 7.4, 7.5 and 7.6), we see the variation of the position X′, Y′, Z′ with the time (X′, Y′ and Z′ are the real values of X, Y and Z positions after passing by the diagram of the accelerometer simualtion that we have already seen them in figures (2.11, 2.12 and 2.13).

7.1.3 Simulation of actuators

The results of the actuators simulation are represented in the next figures.

65
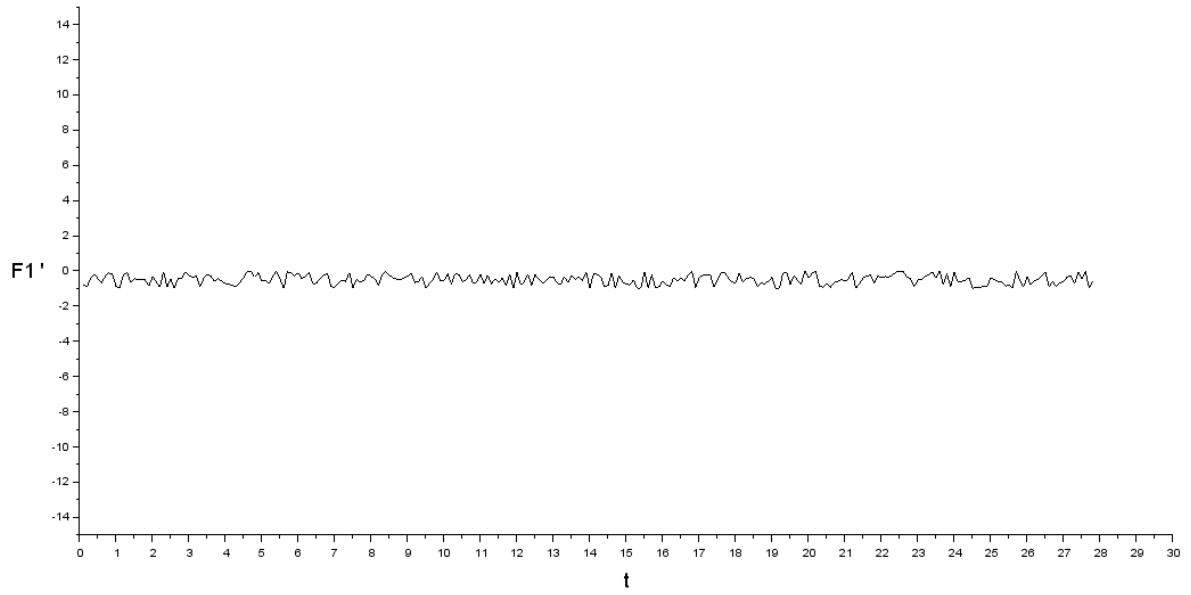
Graph 7.7: Variation of F1′ with time
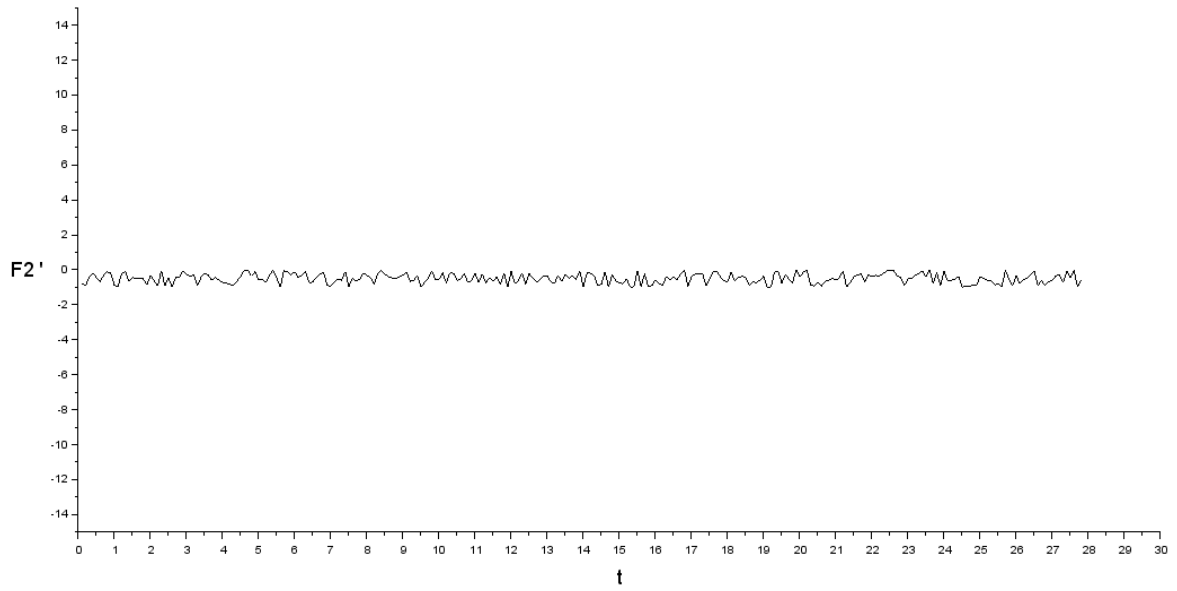
Graph 7.8: Variation of F2′ with time
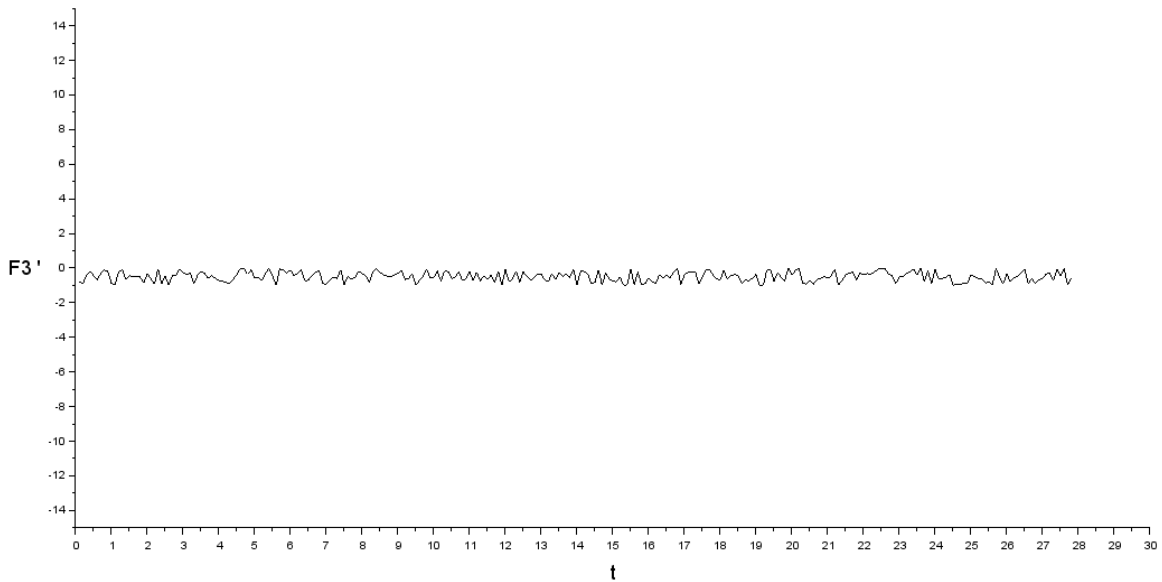
Graph 7.9: Variation of F3′ with time
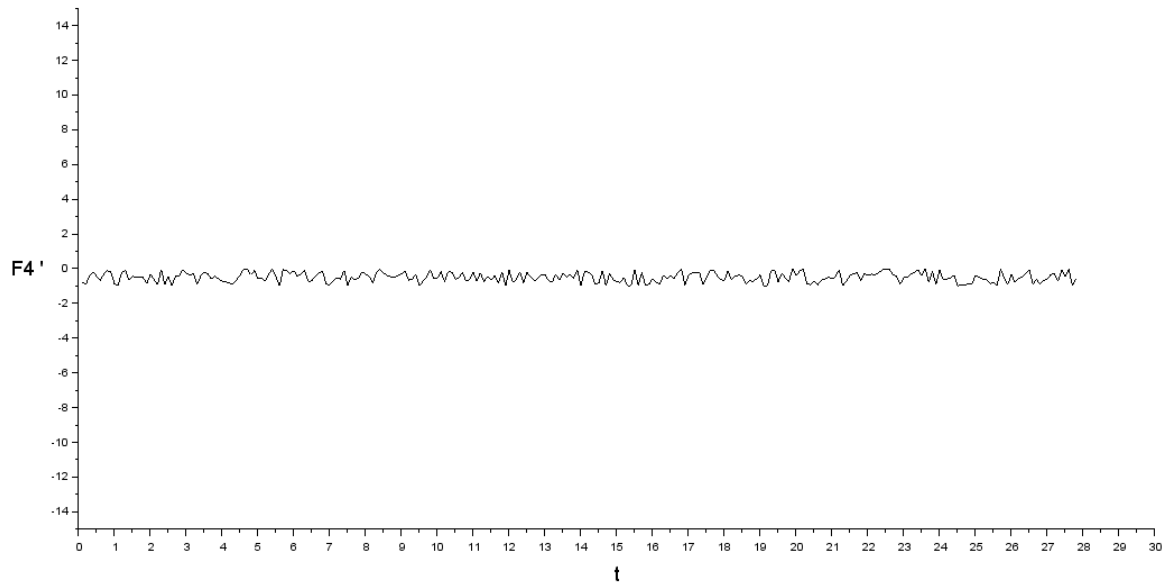
Graph 7.10: Variation of F4′ with time

It is clear in the previous graphs (Graph 7.7, 7.8, 7.9 and 7.10), that the values of the forces given to the propulsion to control the attitude of the satellite; these forces (F′) are also different than the real values given as inputs because of random generator and others blocks (represented in the figure2.14). In the four previous graphs, we see the variation of F′ (F1′, F2′, F3′, F4′) with time.

7.1.4 Results of the integration of diagrams

To represent our system, it is necessary to integrate all the previous diagrams to simulate the state when the satellite is really in the atmosphere.

The results of this simulation (represented in figure 6.16) are the next (graph 7.11, 7.12 and 7.13):
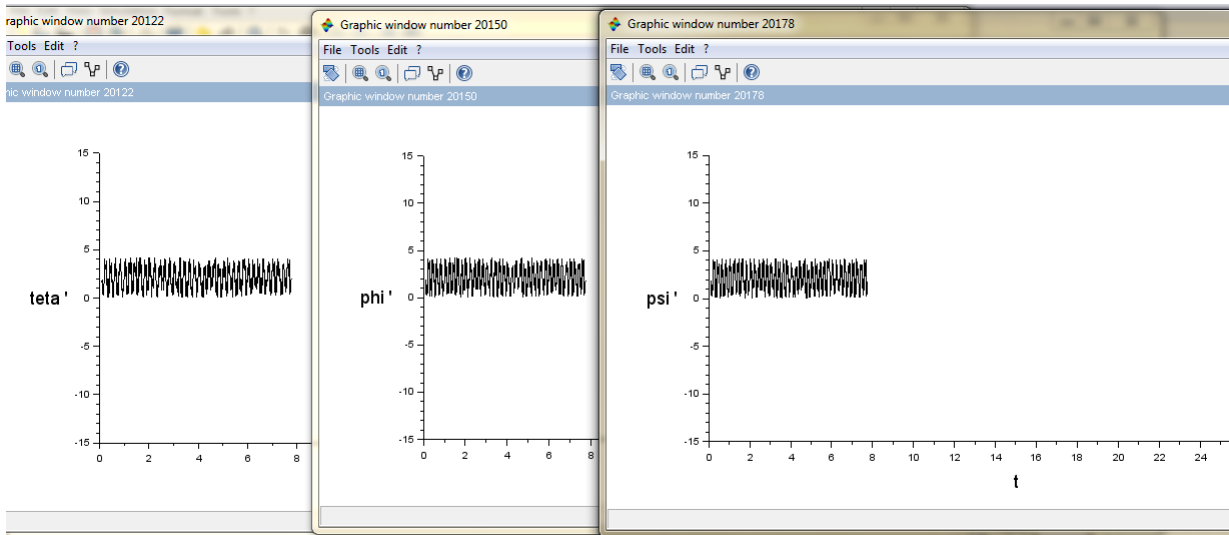
Graph 7.11: Variation of the angles with time
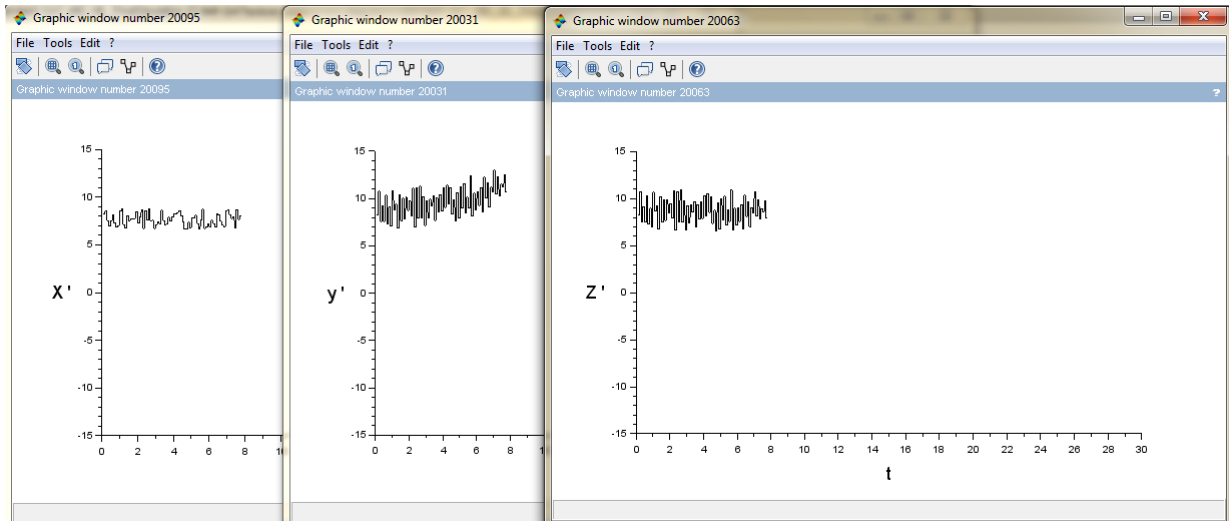
Graph 7.12: Variation of the position with time
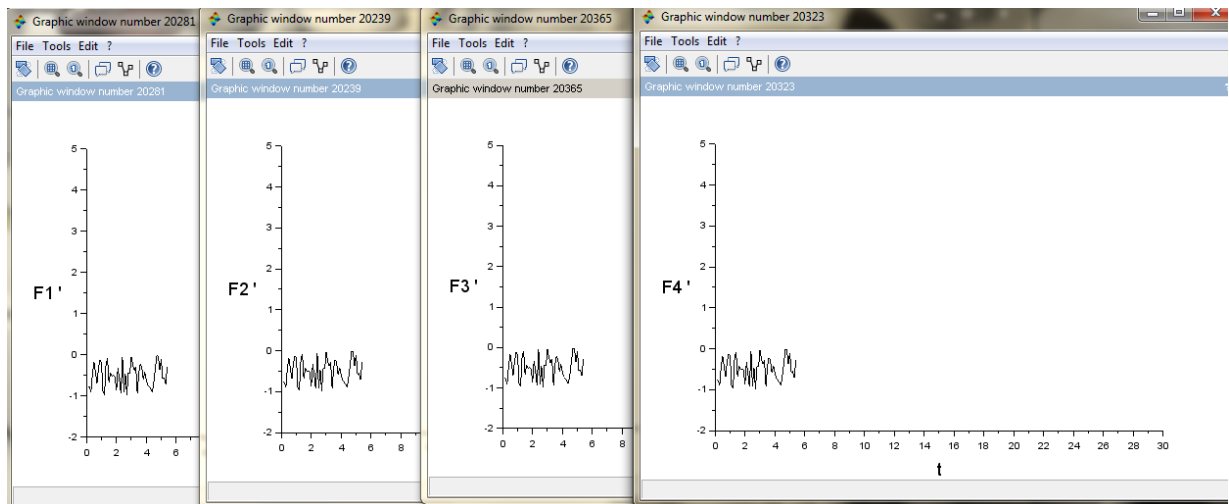
Graph 7.13: Variation of forces with time

68

When the model is executed, all these graphs appear in the same time to represent how the orientation and the position change with the forces during the time of the simulation.

So we arrive to simulate our satellite by representing all the components of the satellite in scicos, and by integrating all these models to obtain the representative diagram.

7.1.5 Test of simulation

In this section, we will see the result of the simulation when we change the values of forces.

I. Results of case 1

The first case is the case with the value zero of F4 (figure 6.17).
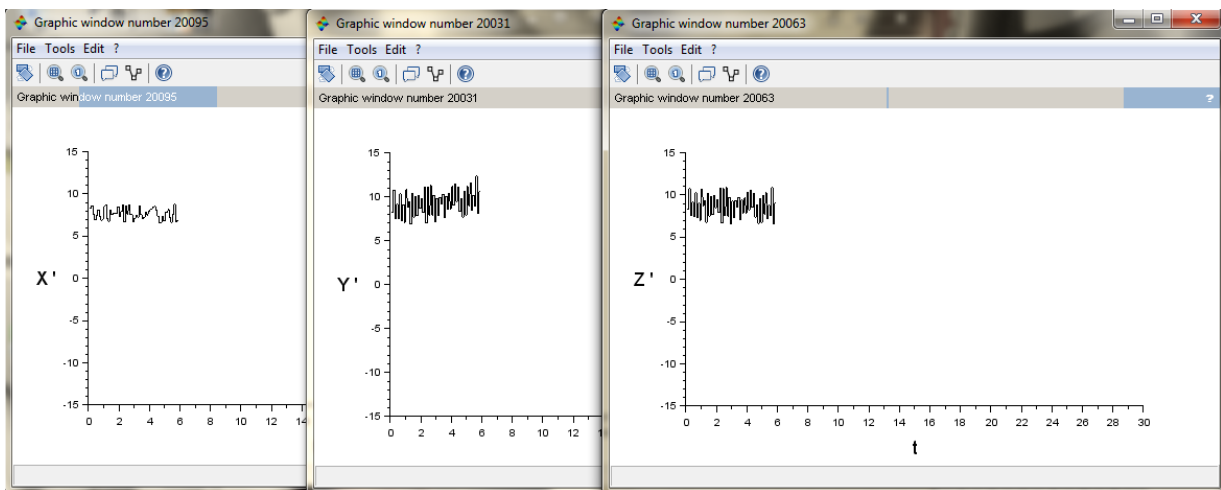
Graph 7.14: Variation of position with time
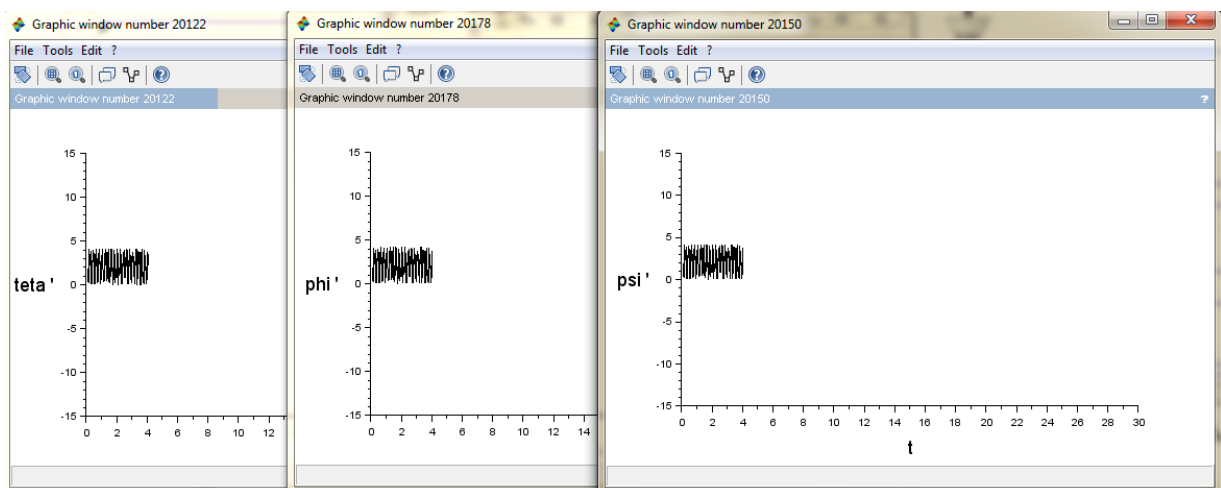
Graph 7.15: Variation of position with time

69
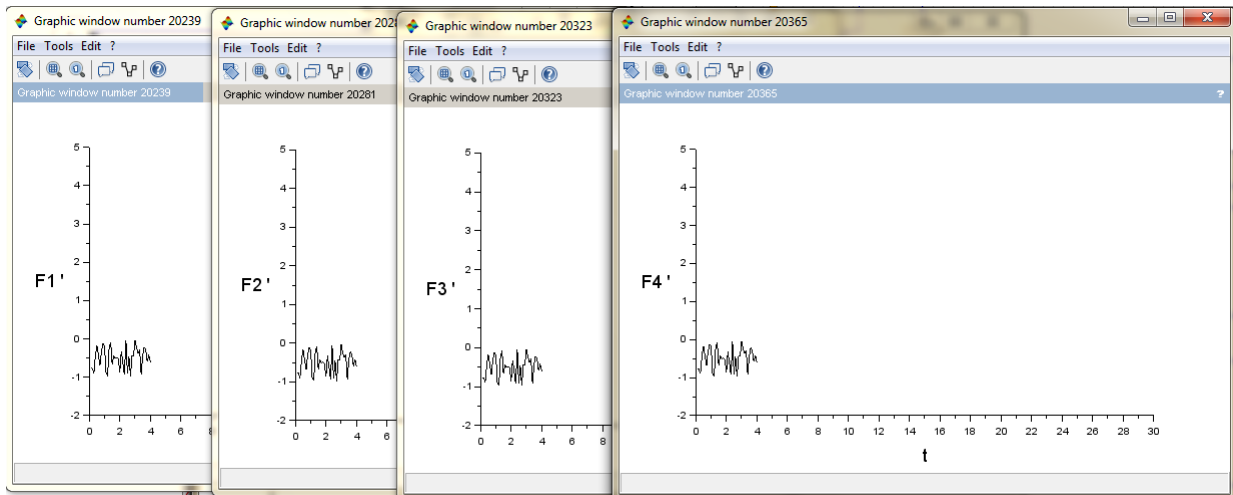
Graph 7.16: Variation of forces with time

We see a small difference in the graphs of position. We can detect that the variation of Y′ and Z′ with time is different when all the forces have a nonzero value.

II.   Results of Case 2

This case represents the test when F4 is zero, and F2 is a step function (figure 6.18).
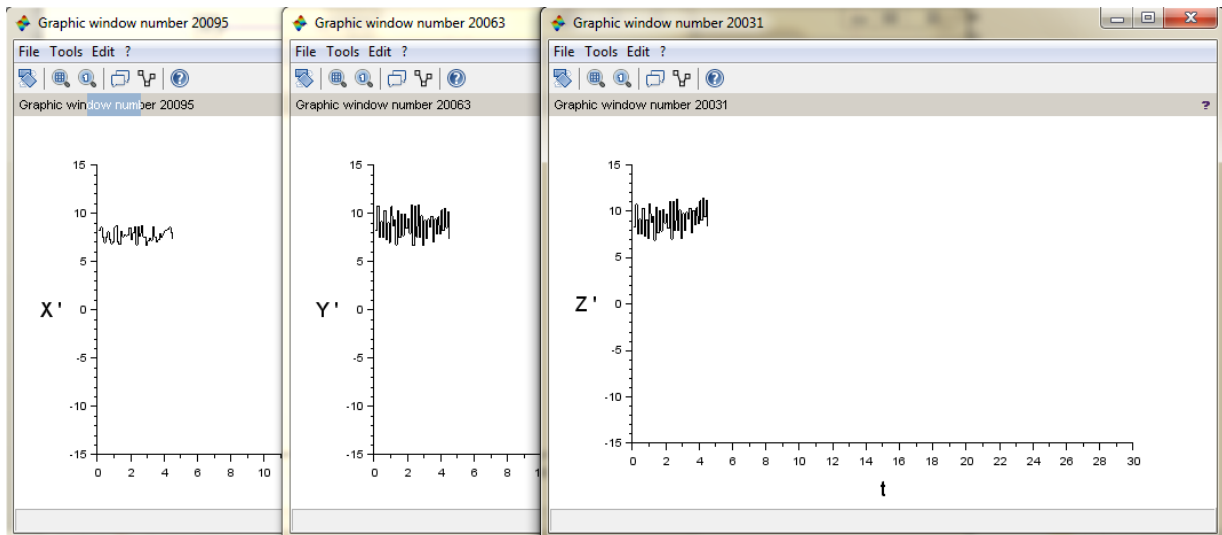
Graph 7.17: Variation of position with time
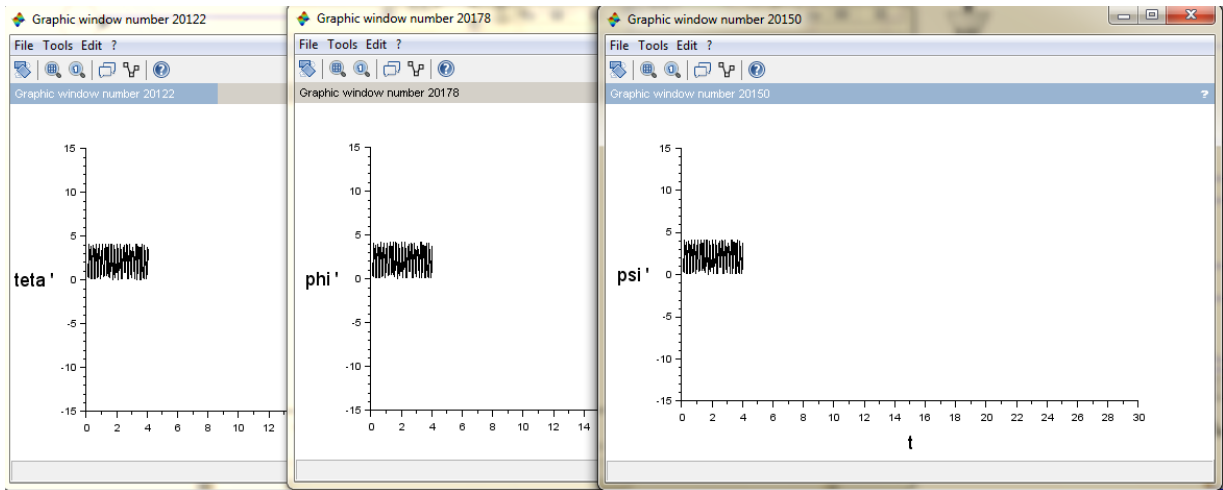
Graph 7.18: Variation of angles with time
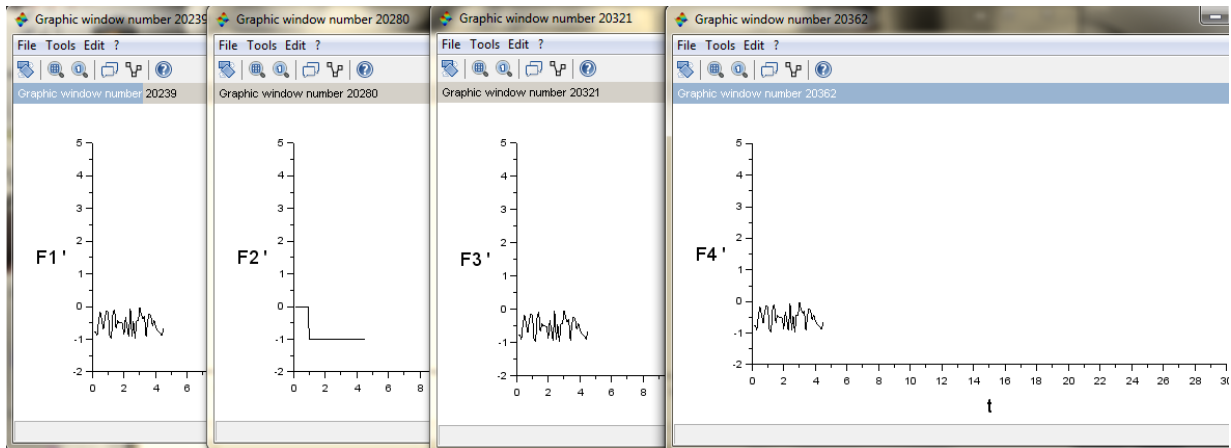
Graph 7.19: Variation of forces with time

We can see clearly the change of the graph of position (Z' only), and the change in the graphs of actuators. The graph of the second propulsion is different; it is as a step function with a little change due to noise and other blocks.

III.   Results of case 3

In this case, all the forces have zero as value, exception F1 (Figure 6.19).

71
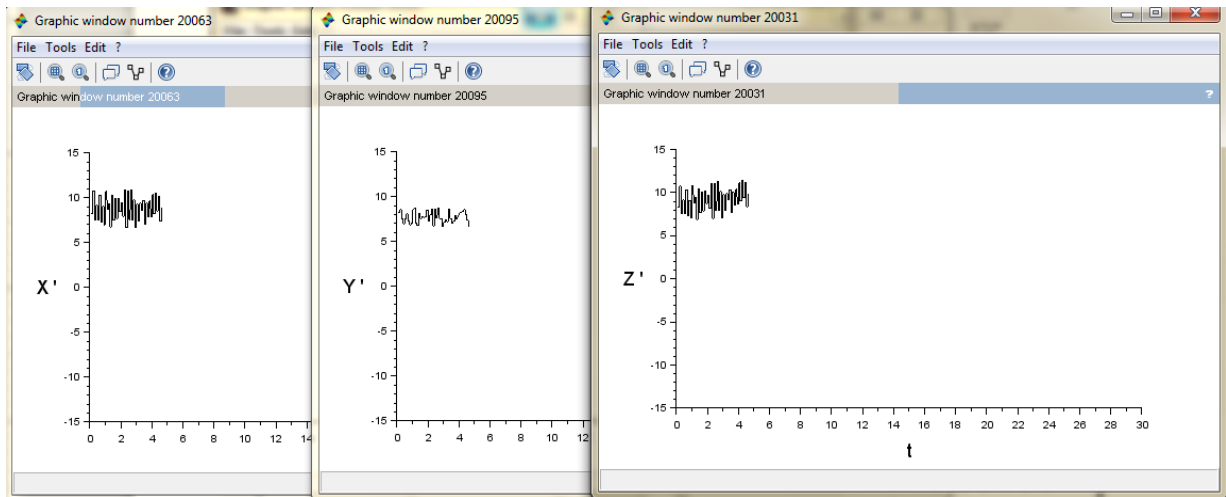
Graph 7.20: Variation of position with time
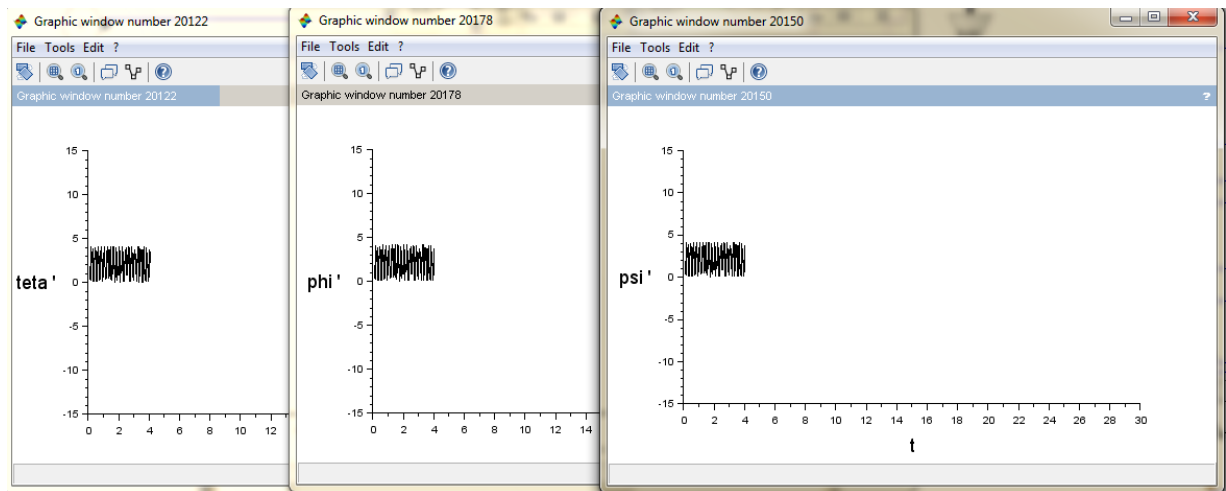
Graph 7.21: Variation of angles with time
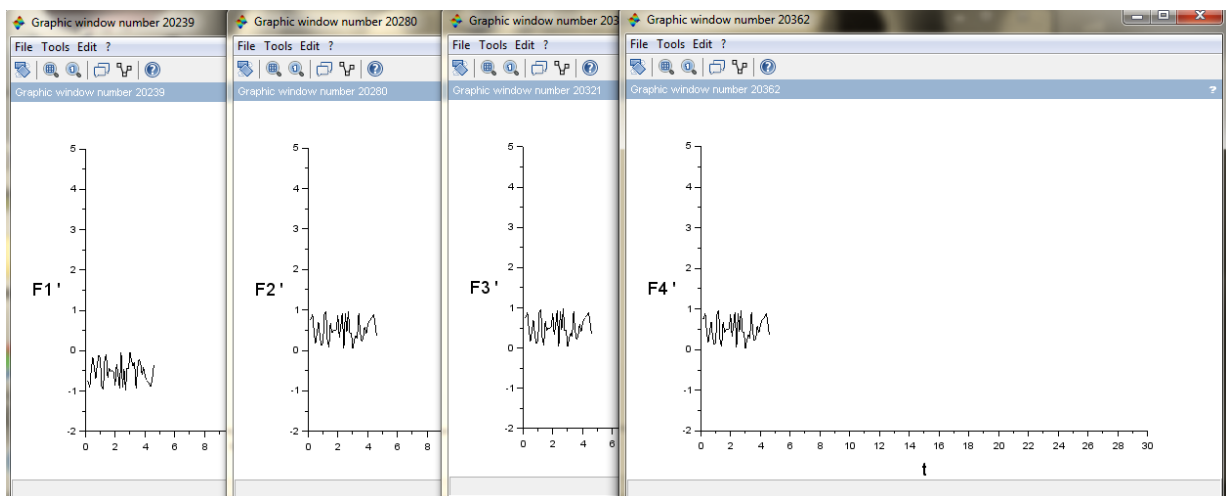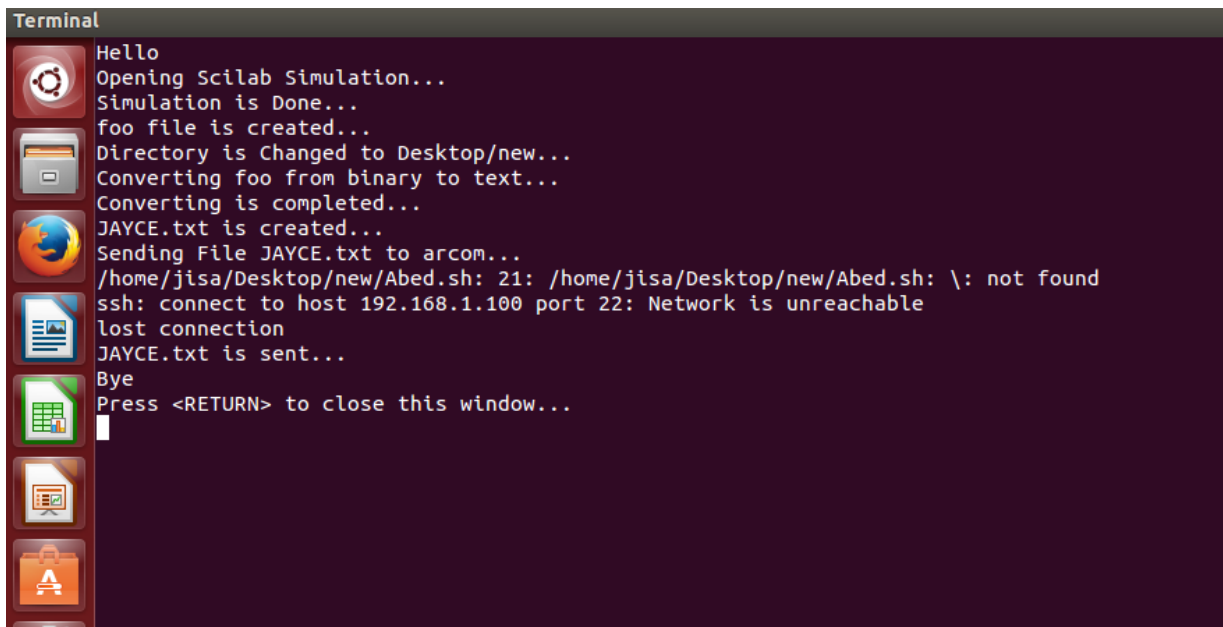
Graph 7.22: Variation of forces with time

72

Here the results are different from the other cases. We can see the difference in the graphs of position (X', Y' and Z'), and we can see also the difference in the graph of the force F1. This difference is due to the nonzero value of F1, when all the others value are zero.

So, we can deduce that this simulation is really representative to the state of the satellite. Any changes in the forces given to the actuators appear in the measurement of the position and the orientation of the satellite.

7.1.6 Integration of the Hardware-in-the-Loop (HIL) test rig

Now, after finishing the simulation, we should integrate it with the computer board in a closed loop. The next figure (Fig. 7.1) represents the first result when we run the C-code (this result is the presence of the txt file, values of the simulation, on the board computer).

Figure 7.1: Results after executing the C-code

When we run the C-code, the file scicos is executed automatically, its output is saved in a C binary file (foo); this file is converted to a txt file (Jayce).

Finally, the txt file Jayce is sent to the arcom board, after connecting this board and my computer to the same network.

The file Jayce.txt is now on the arcom board.

This loop is not done now. Another code is needed to let the board to check the values of position and orientation, and to regulate the values of forces which will be given to the actuators.

So, we will write a new C-code which read the txt file, compare the values with the requested one and create an impulse (as values written in txt file).

This txt file (values of forces) will be sending to my computer, where it will be transferred to a binary file. At the end, the file of simulation .zcos takes its input from this C binary file (Fig. 7.2).

73
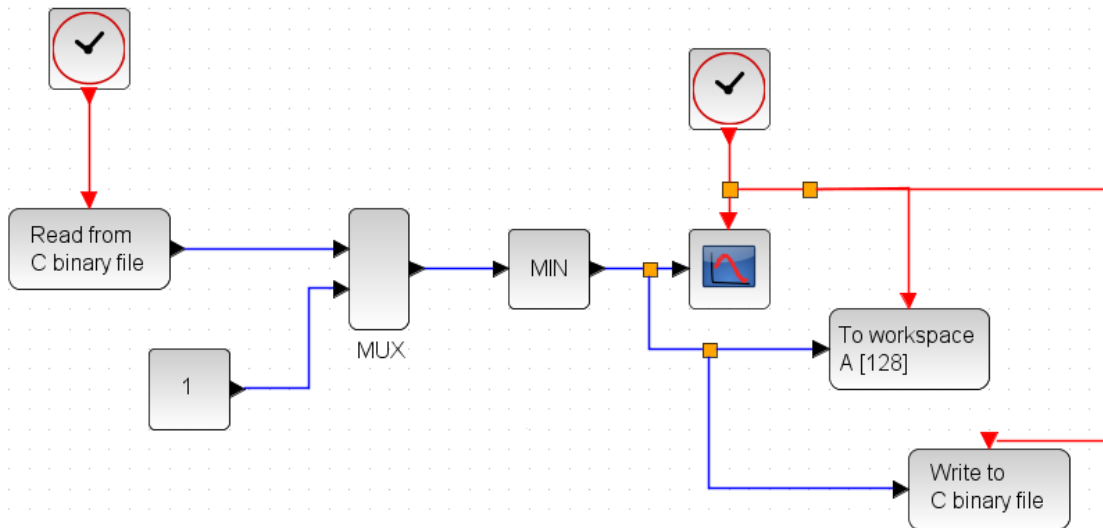
Figure 7.2: The randomly example with input from a binary file and output to a binary file

So the next task (to be continued) is to write the correct C-code which will create values of forces depending on the values of position and angles present in this txt file (Jayce) to read. Finally, a new txt file is created and it contains the real inputs of the file .zcos.

When we find this C-code, the loop will be closed with integration of hardware. Using the same technique, we can connect the HP server (where the simulation of the satellite's components is) with the board computer; by writing the C-codes on the server.

After debugging, we attain the complete simulation of the satellite bus, and we can test this simulation for a long time.

7.2 Receiving meteorological data

After fixing the frequency to detect the meteorological satellite NOAA in the HDSDR, we can see a real signal, see figures 7.3 and 7.4:

Figure 7.3: the first appearance of signal in HDSDR
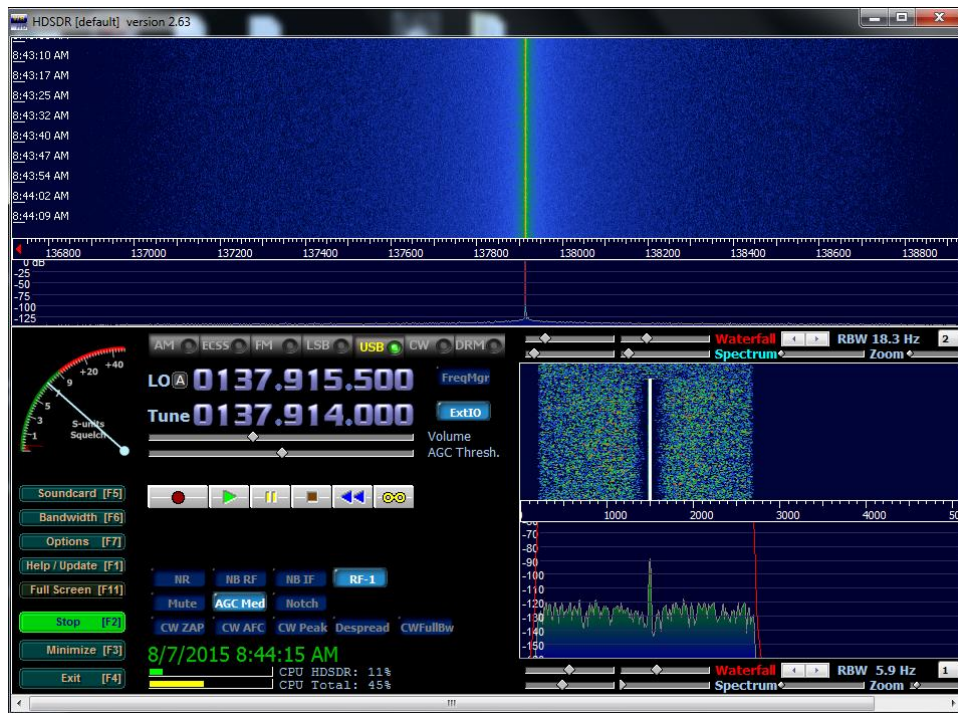
Figure 7.4: The signal received in HDSDR then

And after starting WXtoImg, an image will start to appear (Fig. 7.5).
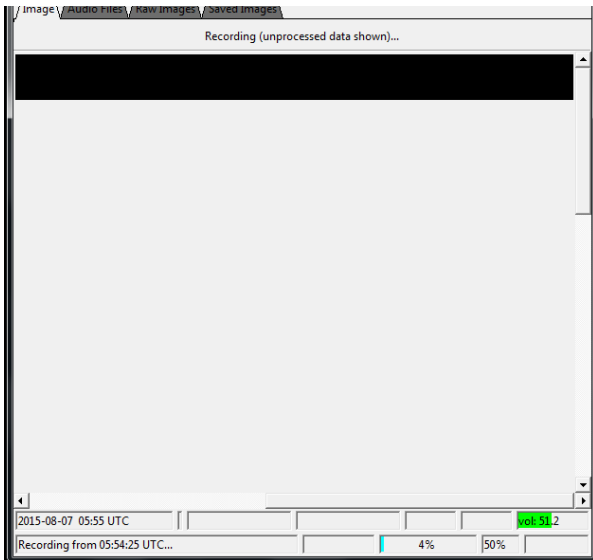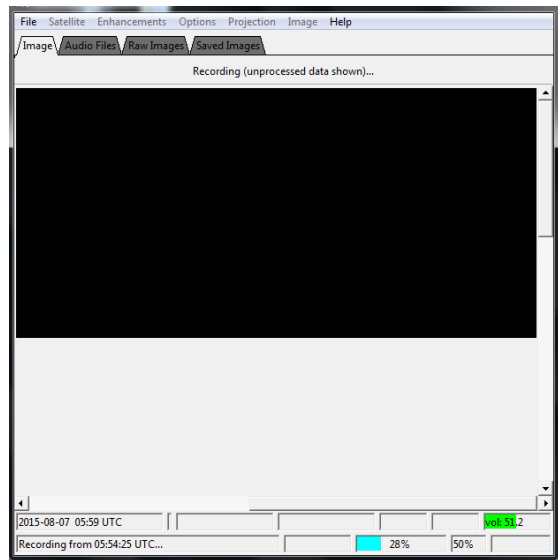
(a)   (b)

Figure 7.5 (a, b): Appearance for a meteorological image

This image is not representative (it is all black), because we have used an antenna with multidirectional detection, so it radiates the same pattern all around the antenna in a complete 360 degrees pattern, therefore a lot of noise are generated and the signal isn't clear and apparent.

It is necessary to use a directive antenna (Fig 7.6) very high frequency VHF to obtain the request image.

Figure 7.6: Directive antenna

By using this directive antenna, the image will be as the following image (Fig. 7.7):
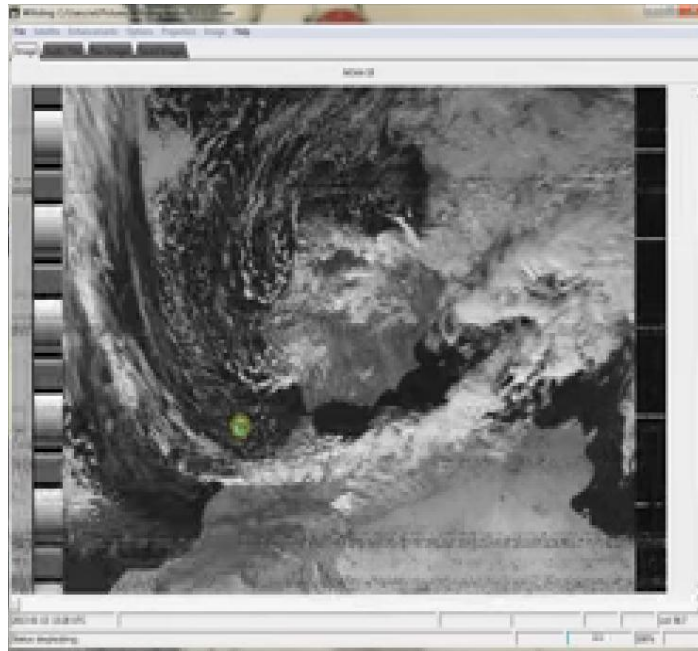
Figure 7.7: The real meteorological image [35]

Also, in this program, by the processing of the image, we arrive to obtain a colored image, so we can easily estimate the state of weather.

Figure 7.8: The real colored meteorological image [35]

Using the same methodology with the same technics, programs and steps, we arrive to take meteorological data after launching the IAP-SAT.

8 Concept for the Propulsion Unit[1]

8.1 Primitively modeling the chosen Pulsed Plasma Thruster (PPT)

Amongst all the inspected Propulsion systems, PPT was chosen and it had to be designed on FreeCad.

PPT's working mechanism was carefully studied and its different components were identified. The primitive 2D sketch that our design depends on is below.
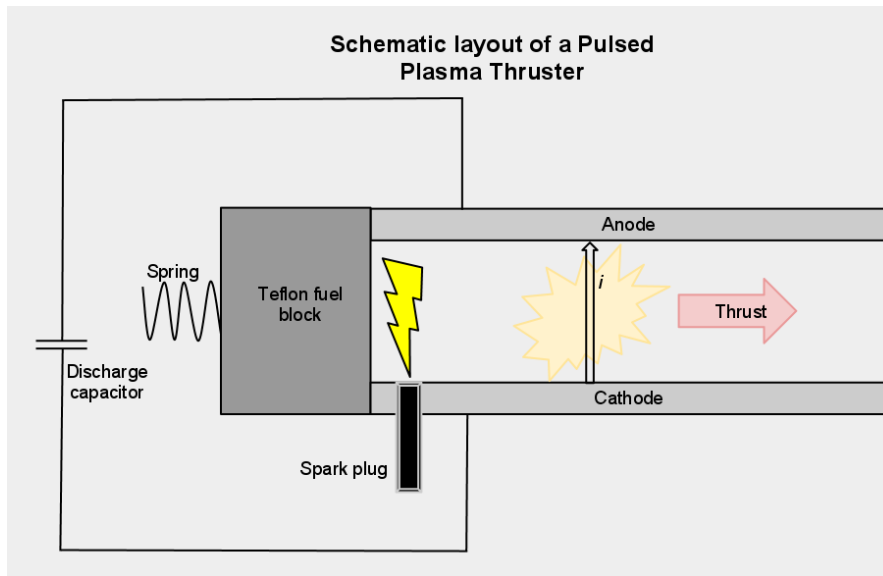
The components necessary for the PPT are:

- Parallel anode/cathode plates
- Spark igniter
- Teflon circular block
- Spring system
- High voltage capacitor

Mechanism: The first step of PPT mechanism is plasma formation. An igniter directed to the solid fuel bar produces a spark that ablates and sublimates the surface of the propellant forming plasma. Since the plasma is positively charged, it completes the circuit between the 2 plates. The interaction between the formed electric and magnetic field produces a Lorentz force that acts on the plasma and accelerates it out of the exhaust at high speeds.

8.2 Modeling our PPT design in FreeCad

After identifying the different components of our PPT system and taking into consideration its working mechanism, the propulsion system was modelled on FreeCad.

The following is the FreeCad model of the PPT:

---

[1] From Ibrahim Ghanim, Practical Work at AECENAR Ras Nhache, July/August 2015
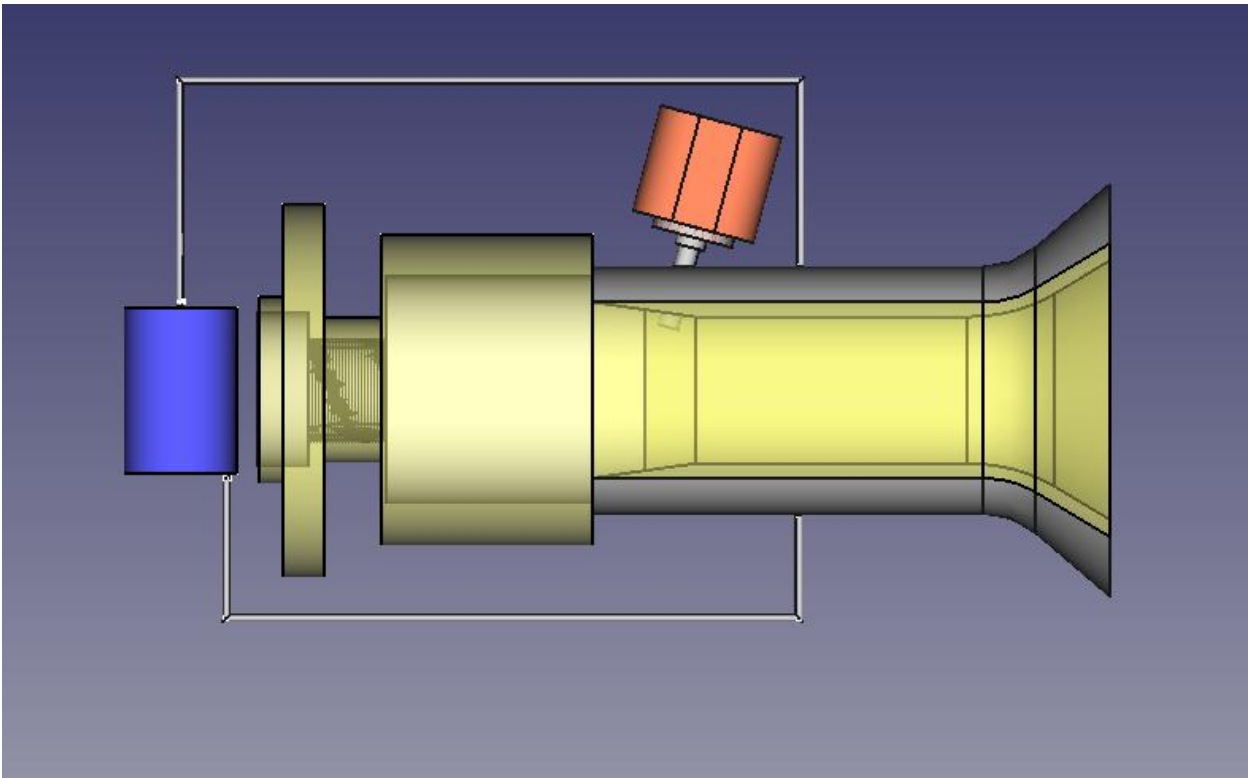
8.2.1 Modelling "fuel continues supplying system"

Most of propulsion systems use fluid fuel which its supply is easily provided by maintaining a pressure difference between the fuel reservoir and the ionization chamber. However pulsed plasma thrusters use solid fuel, that's why its continuous fuel supply was insured by a spring system mounted on the fuel rod.
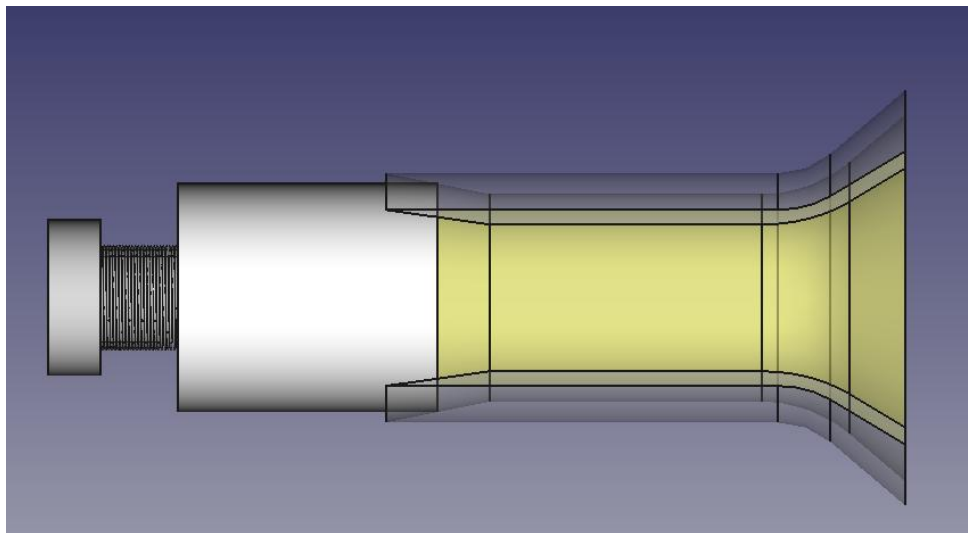
The propulsion unit is designed with a cone shaped ending such that its outer nozzle is slightly larger than the fuel rod and its inner nozzle is smaller than the fuel rode. A compressed spring is mounted to the end of the fuel rod such that the rod is always under pushing force. When the igniter produces a spark and the outer surface of the fuel rod gets ablated and sublimated, the

spring system pushes the rod inward such that it returns to its previous position in the propulsion unit.

8.2.2 2D projections of important parts of the propulsion system

1) Propulsion unit
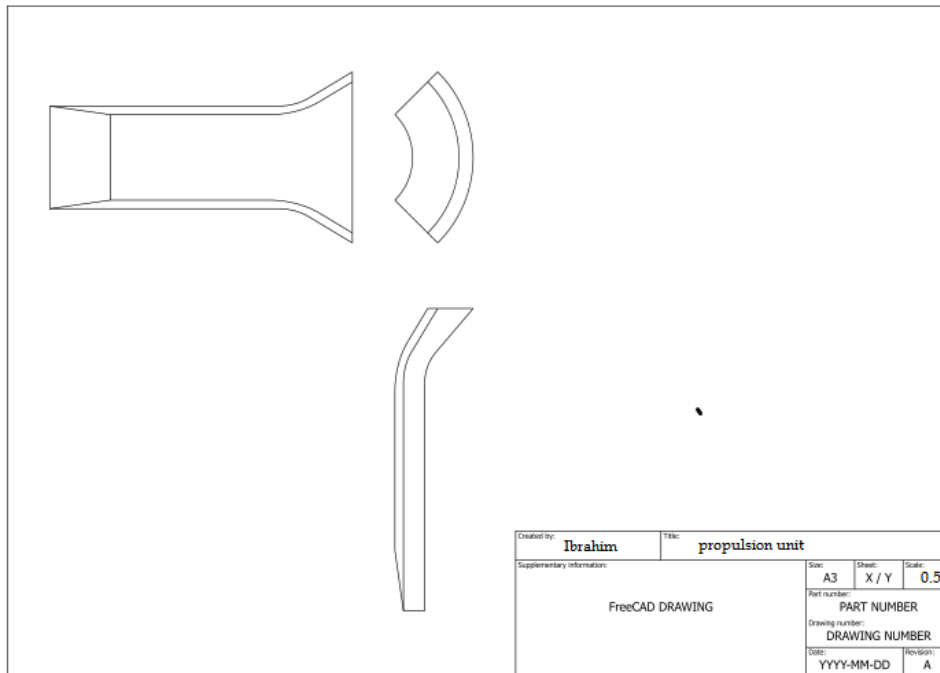
2) Igniter
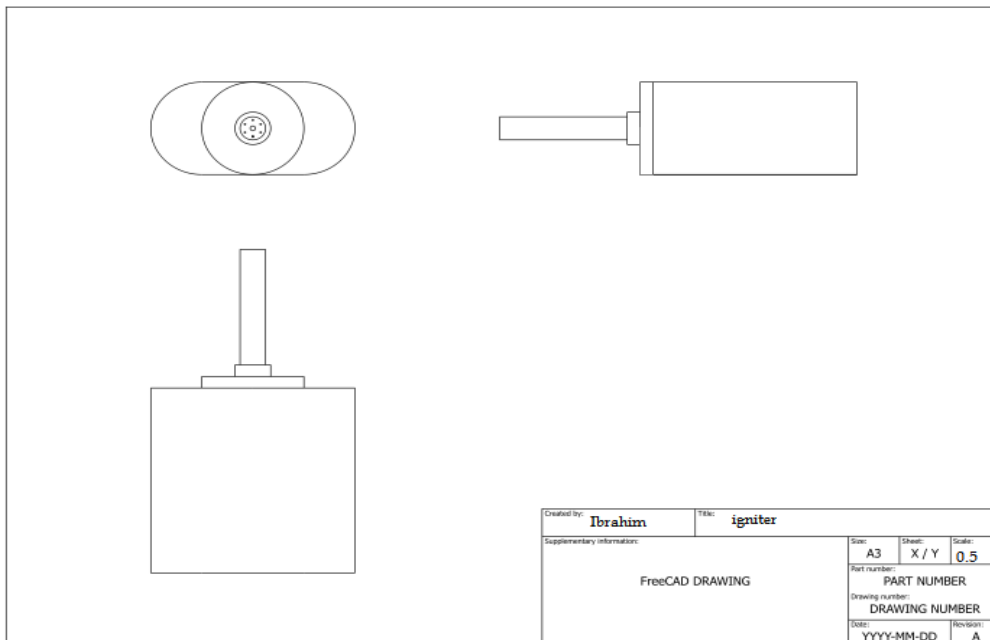
9   Concept for Power Supply and Battery

Satellites need electric energy to pursue operation. This energy is acquired from solar power using solar cells which transform radiation energy into electricity. New developments in solar panel's technology have increased the efficiency of such systems to 30%. However the satellite will be deprived of such source of energy during eclipses, that is when the satellite is passing through the shadow of the earth. This phenomenon becomes more profound for LEO satellites which pass through an eclipse every 100 minutes and which the eclipse represents up to 40% of the total cycle time. In order to provide the continuity of electric energy supply, satellites are equipped with rechargeable batteries. These batteries simply charge during times when solar energy is available and discharge during shadow periods.

The first utilized battery in a spacecraft was the silver-zinc battery which provided high energy density. However because Ag-Zn batteries have low cycle life, they were replace by Ni-Cd batteries that dominated the space industry for 20 years. After that, various batteries were used however restrictions and robust standards imposed on batteries used in space industry rendered many of them obsolete.

Aerospace industry now adopted the lithium-ion-based batteries in its satellite projects since lithium has:

- High energy density
- High specific energy
- Long cycle life
- Ability to be charged and discharged fast
- Compact shape

Battery:

Body:
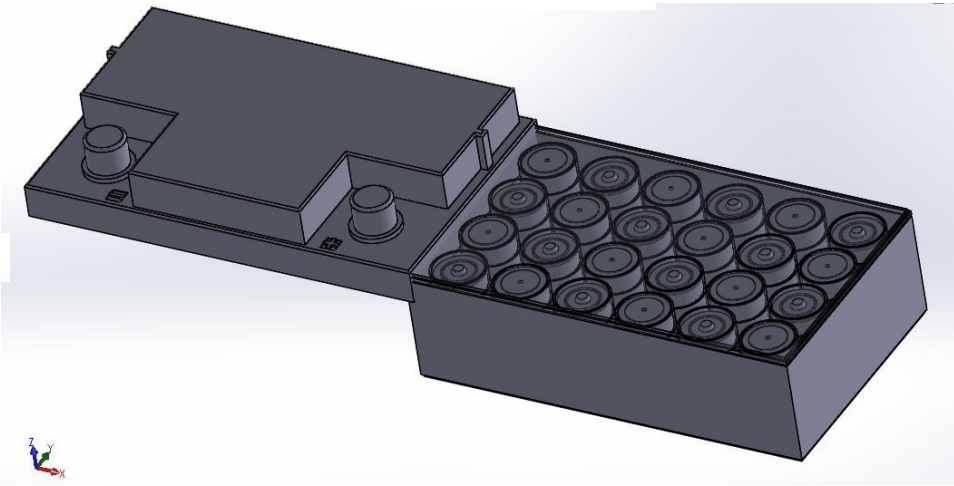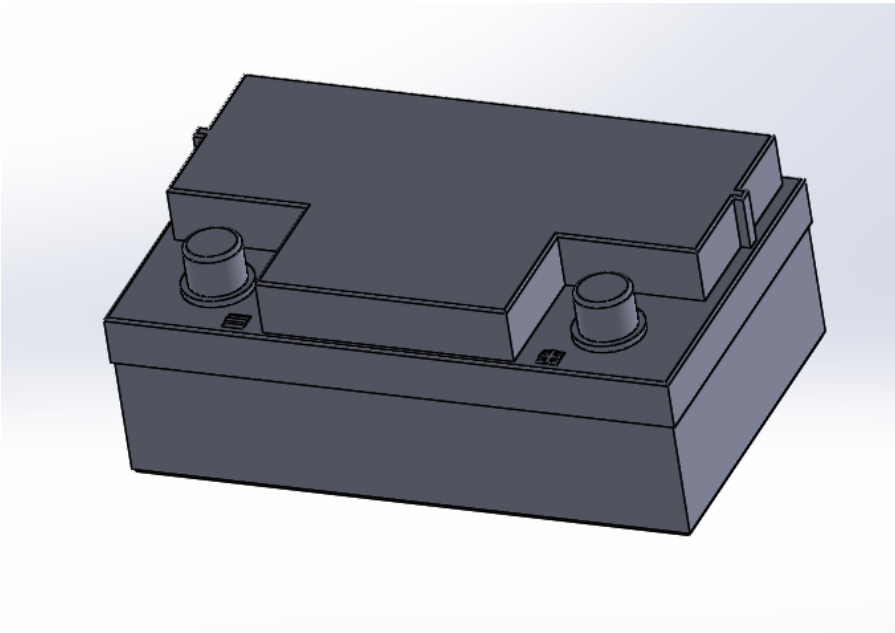
10   Next working packages

10.1 PhD Tasks concerning the propulsion unit

IAP Laboratory at Ras Nhache/Batroun, Lebanon

IAP-SAT is the first Lebanese meteorological satellite. It will be used to take meteorological data to estimate the state of weather in Lebanon.

In 2015 there was established a hardware-in-loop test rig for IAP-SAT where the space environment was simulated. One of the next steps shall be the development of an electrical propulsion unit based on pulsed plasma thruster (PPT) technology. In two PhD theses there shall be investigated the PPT thruster and its interaction with Van Allen Belt magnetic field.

PhD Thesis:

CFD Simulation of interaction of IAP-SAT PPT with Van Allen magnetic field

Detailed description an working plan:

- Development of appropriate optimized CFD algorithm package able to undergo a fast simulation of PPT similar magnetic field environments
- Taking Simulation Data for interaction of IAP-SAT PPT with Van Allen Belt magnetic field.

Keywords: Electrical Space Propulsion Units, pulsed plasma thruster (PPT) technology, computational fluid dynamics.

Contact: Samir Mourad, Email: samir.mourad@aecenar.com, Mobile +961 76341526

AECENAR

Association for Economical and Technological Cooperation
in the Euro-Asian and North-African Region
www.aecenar.com

Bismillah

IAP
INSTITUTE FOR
ASTROPHYSICS

Ras Nhache/Batroun
www.aecenar/institutes/iap

IAP Laboratory at Ras Nhache/Batroun, Lebanon

Figure 3.4: Working Principle of SIMP-LEX[60, 48]
From [Nawaz et. al. 2005]:

Figure 3.5: Test of SIMP-LEX propellant feed system[60, 48]

Figure 3. One of the I-MPD Test Facilities

similar project environment at IRS, Stuttgart (

IAP-SAT is the first Lebanese meteorological satellite. It shall be used to take meteorological data to estimate the state of weather in Lebanon. In 2015 there was established a hardware-in-loop test rig for IAP-SAT where the space environment was simulated. One of the next steps shall be the development of an electrical propulsion unit based on pulsed plasma thruster (PPT) technology. In two PhD theses there shall be investigated the PPT thruster and its interaction with Van Allen Belt magnetic field.

PhD Thesis: Measurement of interaction of IAP-SAT PPT with a laboratory Van Allen belt environment

Detailed description an working plan:

- Development of IAP-PPT unit
- Development of measurement environment
- Taking measurement data

Keywords: Electrical Space Propulsion Units, pulsed plasma thruster (PPT) technology, Van Allen Belt magnetic field

Contact: Samir Mourad, Email: samir.mourad@aecenar.com, Mobile: +961 76341526

References

[Yassir 2010] Muhammad Yassir, *Development and Implementation of the attitude control algorithms for the e micro-satellite Flying Laptop,* PhD thesis, IRS, University of Stuttgart, 2010

[1] Satellite Power Systems Solar energy used in space; BR-202 May 2003; European Space Agency; Published by: ESA Publications Division ESTEC, PO Box 299 2200 AG Noordwijk; The Netherlands; Editors: Niels E. Jensen & Bruce Battrick)

[2] The Space Grid; Sun-synchronous orbiting SBSP Satellites with Equatorial orbiting Reflector Satellites for Earth and Space Energy By Royce Jones.

[3] Goddard Space Flight Center. SpaceRef Interactive Inc. June 16, 2009, Retrieved 2009-

[4] Introduction to Spacecraft Propulsion; Peter Erichsen, September 2006

[5] Satellite orbits; Kidder and Vonder Haar: chapter 2; Stephens: chapter 1, pp. 25-30; Rees: chapter 9, pp. 174-192.

[6] A thesis accepted by the Faculty of Aerospace Engineering and Geodesy of the Universität Stuttgart in partial fulfillment of the requirements for the degree of Doctor of Engineering Sciences (Dr.-Ing.), By Muhammad Yassir.

[7] Prototype for a base station for supernova remnant HI line radio wave detector and analyzer (SRWDA), Master Thesis, Suheeb Kassar. Supervisor: Samir Mourad, IAP/AECENAR, KIT, Karlsruhe, April 2014

[8] STAR is an accelerometer system provided by CNES and developed by ONERA (Office National d'Etudes ET de Recherches Aerospatials), France.

[9] Narayan, K. Lalit (2008). Computer Aided Design and Manufacturing. New Delhi: Prentice Hall of India. p.3. ISBN 812033342X'.

[10] Hess.M, Martin.K.K, Rachul.L.J.; (February 7, 2002). "Thrusters Precisely Guide EO-1 Satellite in Space First", NASA, Archived from the original on 2007-12-06.

[11] U.S. Department of Commerce National Oceanic and Atmospheric Administration National Environmental Satellite, Data, and Information Service Suitland, Maryland; National Aeronautics and Space Administration Goddard Space Flight Center Greenbelt, Maryland

[12] S¯adhan¯a Vol. 29, Part 2, April 2004, pp. 189–203. © Printed in India

[13] Gpredict User Manual; by Alexandru Csete OZ9AEC; 12 Oct 2010.

[14] Satellite Attitude Determination with Low-Cost Sensors By John C. Springmann in the University of Michigan 2013.

[15] European space agency; science and technology; electrical space craft propulsion; September 2003.

[16] 2006 Springer Science + Business Media, Inc, Printed in the United States of America.

[17] Saturday, May 21, 2011; Control Systems- Block Diagrams and Signal Flow Graphs; by ramesh akula).

[18] INTECH; 2014, Modelling and Simulation Based Matlab/Simulink of a Strap-Down Inertial Navigation System' Errors due to the Inertial Sensors; Teodor Lucian Grigorie and Ruxandra Mihaela Botez; http://dx.doi.org/10.5772/57583

[19] Precise Orbit Determination for Low Earth Orbit Satellites, David Hobbs, Lund Observatory, Sweden; http://www.astro.lu.se.

[20] DUBAISAT-1: MISSION OVERVIEW, DEVELOPMENT STATUS AND FUTURE APPLICATIONS; by Adnan Al Rais, Ali Al Suwaidi, and Hosni Ghedira.

Associate Research Engineer, DubaiSat-1 Program; Emirates Institution for Advanced Science & Technology (EIAST), Dubai, UAE.

[21] SHELDAHL; A Multek brand; WWW.SHELDAHL.COM | 1150 SHELDAHL ROAD, NORTHFIELD, MINNESOTA 55057 USA.

[22] Choueiri, Edgar Y. (2009) http://.alfven.princeton.edu/papers/sciam2009.pdf

[23]http://www.srh.noaa.gov/images/bro/outreach/pdf/CareerOpportunitiesMeteorology.pdf

[24] http://www.imd.gov.in/section/satmet/dynamic/satfaq.pdf

[25] http://www.astro.washington.edu/users/smith/Astro150/Tutorials/Kepler/

[26] Sun-Synchronous Orbit dictionary entry, from U.S. Centennial of Flight Commission

[27] Radio-Electronics.com: resources and analysis for electronics engineers (http://www.radio-electronics.com/info/satellite/satellite-orbits/low-earth-orbit-leo.php)

[28] http://history.nasa.gov/conghand/propulsn.htm.

[29] http://www.scicos.org/book/ScilabScicosBook-Ch7.pdf (Scicos Block; By Fady NASSIF).

[30] Texas Instruments; satellite payload: gyro sensor.

[31] http://www.wxtoimg.com/

[32] VGB Congress Power Plants 2001, Brussels, October 10 to 12, 2001

[33] 6 Factors that Effect Solar PV System Efficiency; Posted on March 6, 2013 by Goodpal

[34] Technical information, SMA, performance ratio, Quality factor for the PV plant, Perfratio-UEN100810

[35] http://www.youtube.com/watch?v=fopnIkYnFPI; 2 jan. 2013 - Ajouté par themrworf1701.

Annex1: Master Thesis: Task, Time Plan and Result in Short (Results are Chapters 4-7 and parts of Ch.2)[2]

AECENAR
Association for Economical and Technological Cooperation in the Euro-Asian and North-African Region
www.aecenar.com

Bismillah

IAP
INSTITUTE FOR ASTROPHYSICS

Ras Nhache/Batroun
www.aecenar/institutes/iap
12.2.2015

IAP Electronics and Simulation

IAP Laboratory at Ras Nhache/Batroun, Lebanon

Simulation Environment	In such a way the Graphical User Interface shall be

Master Thesis

Simulation of the system and of the operational flight of IAP-SAT to take meteorological data to monitor and estimate the energy supply potential of large scale photovoltaic energy plants to control a national alternative energy supply program

Detailed description and working plan

Modeling of actual IAP-SAT hardware and software components in octave	8 weeks
Modeling of IAP-SAT mechanical components with ProE	3 weeks
Integration to a operation flight model and visualization in IAP-ECS environment	3 weeks
Documentation	3 weeks

Keywords: Control of national energy mix programs, large scale photovoltaic plants, meteorological data, *Low Earth Orbit Satellite, Flight Mechanics&Dynamics, Matlab Simulink/Scilab*

Contact: Samir Mourad, Email: samir.mourad@aecenar.com, Mobile +961 76341526

Figure1: Project proposal

ID	Name	Start	Finish
Master 2		3/10/201	7/31/201
Ch.1: Intro	Introduction	3/10/20	3/21/20
	The project IAP-SAT, description of task	3/10/201	3/21/201
Ch.2: Basi	Basics	3/18/20	5/8/201
	Meteorology, orbits, LEO	3/18/201	3/30/201
	Signals&blocks, signal proccessing, control blocks	4/26/201	5/8/2015
	Satellite bus, meteorological satellites, payload for meteo	3/18/201	4/8/2015
Ch.3.1	IAP-SAT Mission simulation (IAP-SAT_MIS)	3/23/20	4/15/20
31.1	Specifications of orbit parameters of IAP-SAT [meteorolog	3/23/201	4/1/2015
31.2	Visualization of ground track of only IAP-SAT with gpredi	4/2/2015	4/8/2015
31.3	Modeling of IAP-SAT in FreeCAD	4/9/2015	4/15/201
Ch.4.1	Presentation film	4/16/20	4/25/20
	Meteorological sats, FreeCAD model of IAP-SAT, ground	4/16/201	4/25/201
Ch.3.2	IAP-SAT Hardware in the loop test in rig (IAP-SA	5/9/201	7/10/20
	System design of simulated blocks and real HW (Accel,	5/9/2015	6/3/2015
	Implementation: simulated Blocks (e.g Space enviornme	6/4/2015	7/10/201
Ch.4.2	Documentation film of long enduring SIL test	7/12/20	7/23/20
	IAP-SAT_HIL operation and testing (Full operation in orbi	7/12/201	7/23/201

Figure2: Time Plan of Master Thesis, Last updated April 2015

90

Result of Master Thesis in short

The task of this project is the simulation of the first Lebanese meteorological satellite IAP-SAT. This satellite will take photos to the earth and sen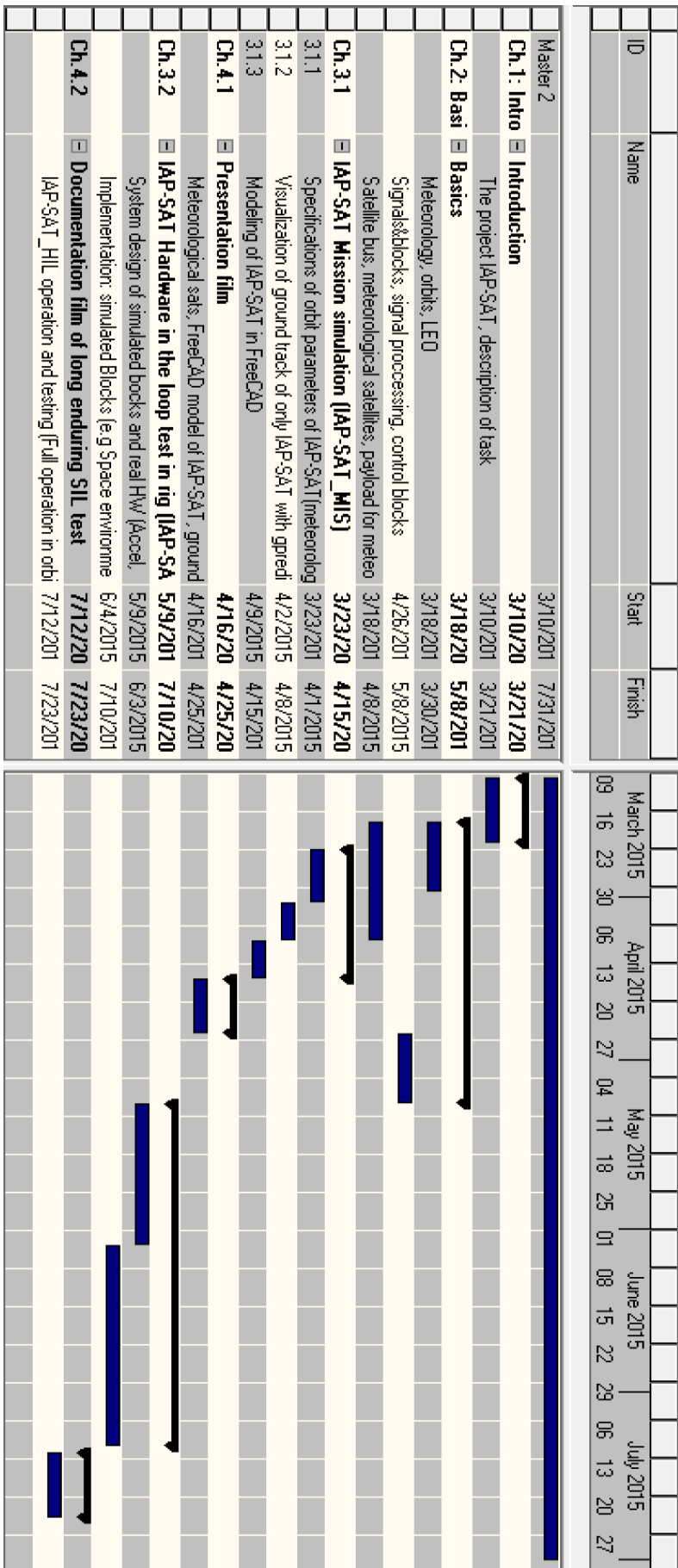d it to the ground station as signals, where these images will be decoded in weather data. These data permit to estimate the energy supply potential for a photovoltaic plant, and then we can monitor this photovoltaic plant.

Our duty during this internship was to build a framework to control the IAP-SAT satellite in a metrological data collection mission. This framework consists of many steps that are detailed below: The mission simulation, hardware in the loop, receiving metrological data.

The first step in this project was the mission simulation of IAP-SAT.

We have described in general the meteorological satellite, its task and its components. Then, we have specified and modeled our satellite by approaching IAP-SAT to DubaiSat in the attitude, the inclination, the mass and the shape. It has 680 km as attitude, 98.9° as inclination, 200 Kg as mass and a hexagonal shape.

After, to estimate the status of the ground tracking of IAP-SAT after launching, we have followed DubaiSat to view its ground tracking. We found that such satellite accomplishes about 16 rotations around the earth during the day, and we have made a representative film resuming all the results of this pursuit.

Next step in our project was the hardware in the loop.

We have simulated all the non-present components of the satellite (sensors, actuators and space environment) as block diagrams in Scilab/Scicos. We have integrated all these diagrams to obtain one diagram representative of the real state and function of the satellite.

Then, a closed loop was simulated between the sensors that provide the actuators positions and angles, and the actuators that generate forces to change the attitude and put the satellite on the orbit again after a possible disturbance based on the data of the sensors.

As perspectives, we got success in writing a C-code which reads the outputs of the simulation from a C-binary file and transforms them to a txt file. This txt file is sent and read on the computer board; however, this task remains incomplete and need more investigation.

Therefore, the next task consists of writing a new C-code with the duty of examining these outputs and sending the request inputs to the diagram of simulation. These inputs are sending as txt file and should be again transformed to a C-binary file to be read by the simulation. This task must be more examined and investigated.

Also we got a problem when decoding the signals and transforming them into image, in fact the obtained image was black.

Therefore, our perspectives include also the part that should be more investigated using a directive antenna instead of multidirectional one, and certainly, the obtained image will be clear and very representative to the state of weather with this directive antenna.

Finally, we have attained our goal and we have gathered all this steps in a framework that arrives to simulate the mission of IAP-SAT and controls its position using the simulated loop, and at the end we have arrived to receive meteorological data from NOAA satellite.

We are waiting know to other tasks to be finalized by our colleagues in our team to finish the simulation part. Then in our perspectives we will try to pass to the application of what we obtain in this simulation study on the ground truth in order "to make our dream true" and launch the first Lebanese satellite, IAP-SAT with the meteorological mission.

Annex 2: C code of Control Algorithm at the Board Computer

The next figure represents the first attempt of writing the C-code, which will examine the outputs of the sensors on the board computer.

This code is written by my colleague, during his master training; however, I have finished my project before that he accomplished his project.

```c
3.sce ×    read.c ×
#include <stdio.h>
#include <stdlib.h>

 int main()
 {

      float num;
      FILE *cfptr;
      FILE *fptr;

 if ( ( cfptr = fopen( "/home/arcom/JAYCE.txt", "r")) == NULL) {
      printf("file could not be opened. \n");
      }

       else    {

          fseek ( cfptr, 10 * sizeof(float), SEEK_SET);
          fscanf( cfptr, "%f", &num)  ;
          fclose (cfptr);


           }

 if ( ( fptr = fopen( "/home/arcom/ibra.txt", "w")) == NULL) {
      printf("file could not be opened. \n");
      }

       else    {
              fprintf(fptr, "%f\n", num);
              fclose (fptr);

       }

       system("PAUSE");
       return 0;
       }
```

Figure3: The C-code first trial

A similar code, but more detailed and delicate, will close our loop, and complete this task.