POLITECNICO
MILANO 1863

DIPARTIMENTO DI
SCIENZE E TECNOLOGIE AEROSPAZIALI

Politecnico di Milano
Space Flight Mechanics 2

Nadalon Emanuele

Progetto # 1011

1 – PROJECT SPECIFICATIONS AND INITIAL CONSIDERATIONS.

1.1 Assigned specifications

Spinning satellite on sun synchronous orbit, quaternion parameterized attitude, sensors: 1 two axes sun sensor, actuators: 1 inertia wheel, 2 reaction wheels.

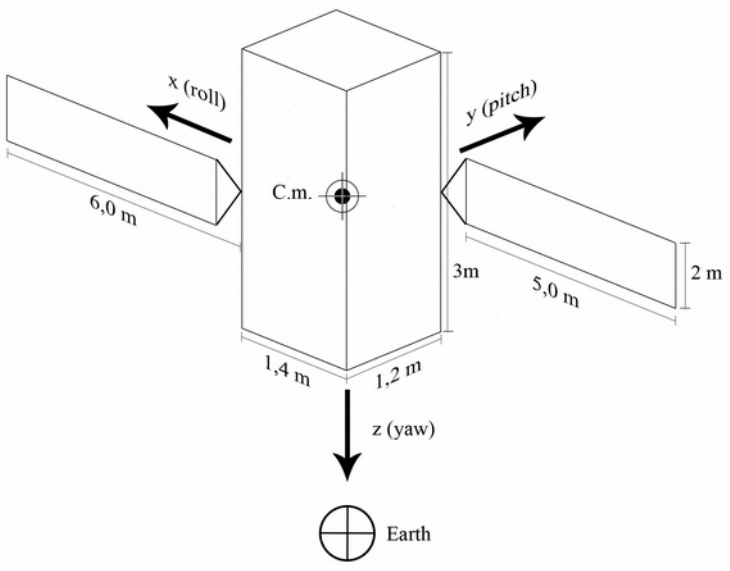
1.2 Adopted specifications

Because of the sun synchronous orbit, the assigned satellite project have been interpreted as an Earth observation satellite, particularly about take-overs to be taken on a daily basis under the same light conditions. Specifically, it has been chosen a circular dawn/dusk orbit, avoiding any eclipse condition; the orbit is then a 2D25R, with orbital period of 115,2 minutes (6912 seconds), an inclination of 101,77° and an orbit radius of 7842,6323 Km (1,22 Earth radii) equal to a flight altitude of 1464,4923 Km. At the project beginning, some inspiration has been taken by *Sentinel 1* satellite from ESA *Copernicus* program.

So, the adopted project specifications are: spinning satellite on sun synchronous orbit, quaternion parameterized attitude; sensors: 1 two axes sun sensor, 1 star sensor; actuators: 3 reaction wheels.

1.3 Masses and inertia moments

The satellite shape is that of a parallelepiped with two solar arrays antisymmetrically mounted with respect to the y-z satellite plane (picture 1.1).



Pic. 1.1

The satellite mass has been hypothesized to be homogeneous and evenly distributed, so that the barycenter is the parallelepiped geometric center considering the solar arrays too, and then the symmetry axes are also the principal axes. The satellite mass is 1750 Kg, while the solar arrays mass is 16 Kg, for a total mass of 1766 Kg. The total solar array surface is 20 m^2 and they produce a total nominal power of 5000 W (EOL).

Setting up a reference frame as that one shown in picture 1.1, satellite principal inertia moments will be those listed in table 1.1.

Satellite Inertia Moments
$I_x = 1486$ Kg m^2
$I_y = 1583$ Kg m^2
$I_z = 919$ Kg m^2

Table 1.1

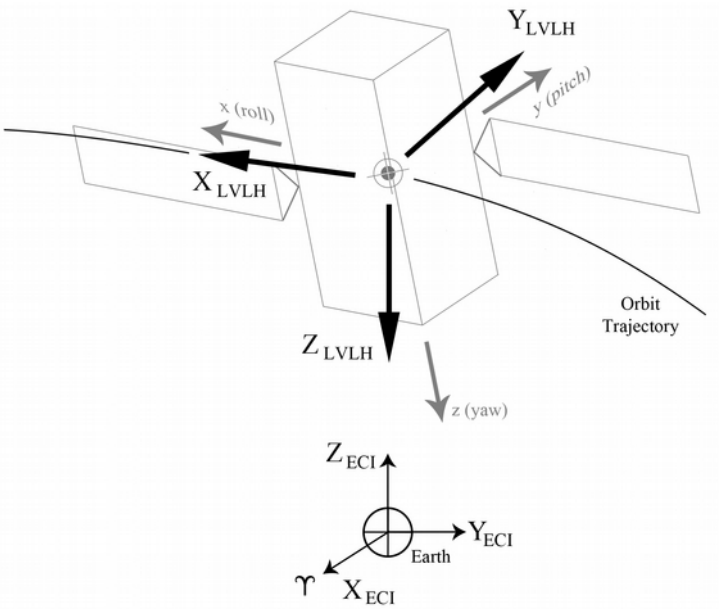
The inertia moments values comply with the Gravity Gradient stability constraints and they will be explained afterwards (section 4.1).

1.4 Reference frames

The project simulations were runned setting up the time at the 2018 Spring Equinox, and then an inertial reference frame, named ECI, were fixed "on the simulations short period", with the Origin in the Earth center, X axis pointing toward the firs point of Aries ♈ direction, Z axis parallel to the Earth rotation axis, and the Y axis to complete a right-hand reference frame.

An LVLH reference frame (orbital frame) were defined with the following axes set up: z axis is pointing toward the nadir to the Earth center, x axis is pointing in the tangential orbital (being a circular orbit) and y axis completes a right-hand reference frame.

The orbital frame is the target one, and the mission to accomplish to match x, y, e z satellite axes (named "*Body*" reference frame) with the target frame axis respectively. Picture 1.2 shows all the reference frames used in this project. The rotation matrices in order to switch from a reference to an other one (and vice versa by inverting the matrices themselves) are computed using the ***Orbit_generator.m*** script and the ***Orbit LVLH and ECI generation*** block, according to the reading in Stark [3] and Curtis [4].



Pic. 1.2

The orbital angular velocity is indicated by the letter *n,* and its value*,* at the chosen altitude, remembering that the orbit is a circular one, is:

$$n = \frac{2\pi}{6912\,s} = 9{,}09 \cdot 10^{-4} \text{ rad/s}$$

Satellite attitude was parameterized following the assigned specifications using quaternions, the latter being set up according to the Matlab and Simulink definition, which impose the first quaternion term to be a scalar, and the remaining one, vector components:

$$\mathbf{q}=[\, q_s,\, q_{v1}, q_{v2},\, q_{v3}]^T.$$

2 – MOTION EQUATIONS

2.1 Dynamics and Kinematics equations.

In order to realize this project, Simulink was used. The ***S/c Dynamics – Euler's equations*** and ***S/c Kinematics*** blocks were designed.

The *S/c Dynamics* block takes as inputs the torques acting on the satellite (in *Body* frame) and gives as outputs the angular velocity components. The moments of inertia, as well as all the quantities required to run the project simulations, are externally assigned as Matlab variables by the ***Start.m*** script. This script calls other scripts which initialize the variables of each "macro-block" in the Simulink model and that are described in appendix *A.2*.

The equations implemented in the *S/c Dynamics* block, comprising the Reaction Wheel Assembly terms provided by the adopted project specifications, are:

$$\begin{cases} M_x = I_x \dot{\omega}_x + (I_z - I_y)\omega_z \omega_y + I_{Rx}\dot{\omega}_{Rx} + \omega_y I_{Rz}\omega_{Rz} - \omega_z I_{Ry}\omega_{Ry} \\ M_y = I_y \dot{\omega}_y + (I_x - I_z)\omega_x \omega_z + I_{Ry}\dot{\omega}_{Ry} + \omega_z I_{Rx}\omega_{Rx} - \omega_x I_{Rz}\omega_{Rz} \\ M_z = I_z \dot{\omega}_z + (I_y - I_x)\omega_x \omega_y + I_{Rz}\dot{\omega}_{Rz} + \omega_x I_{Ry}\omega_{Ry} - \omega_y I_{Rx}\omega_{Rx} \end{cases}$$

The *S/c Kinematics* block receives as input the satellite angular velocity components and generates as output the satellite attitude quaternion *q* integrating in the time domain the formula:

$$\dot{q} = \frac{1}{2}\Omega q$$

Instead of compute the matrix Ω, as described in literature, in order to keep the computational load as lower as possible, a function which makes the calculation without matrices was implemented (on this topic take a look at Blanke e Larsen [1]). The quaternion obtained in such a way is then normalized in order to keep the mathematical consistency of the quaternion itself.

3 – DISTURBANCES

The disturbances affecting a satellite in orbit are the following:

1) Gravity Gradient;
2) Solar Radiation Pressure;
3) Earth Magnetic Field;
4) Aerodynamic Drag.

In this Simulink project model all the disturbance are modeled, but the Aerodynamic Drag torque, that is neglected because of the flight altitude and the consequent lower magnitude with respect to the other disturbances torques.

All the disturbances are modeled in the ***Environmental Torques*** macro-block inside which it is possible to turn them on or off through dedicated switches. Every disturbance is described in its *Body* reference components and added to the respective components of the other ones. The total amount of the obtained components are then sent to the *S/c Dynamics* block.

3.1 – Gravity Gradient.

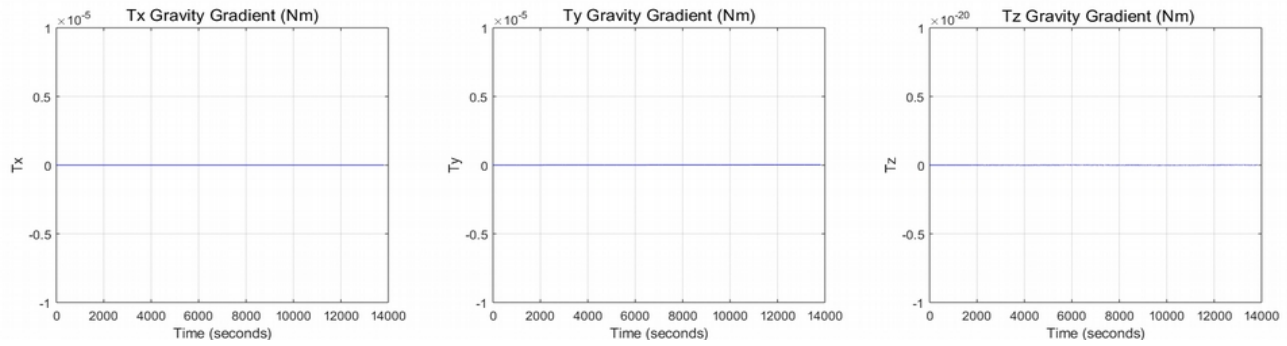
It is modeled in the ***Gravity Gradient Torque*** block.
The Gravity Gradient disturbance is intrinsic to the environment in which the satellite is flying, and it is generated by the Earth gravity field unevenness.

In order to compute it, the following definition was used:

$$T_{GG} = \frac{3\mu}{R_o^3}[\hat{\boldsymbol{R}}_o \times (\boldsymbol{I} \cdot \hat{\boldsymbol{R}}_o)]$$

where μ=3,986·105 Km^3/s^2 is the Earth gravity constant and R_o= 7842,6323 Km is the sun synchronous orbit radius, while **I** is the satellite moment of inertia matrix.
On this occasion, was taken care of use the attitude quaternion to rotate the last column of the LVLH reference frame matrix, to obtain the unit vector $\hat{\boldsymbol{R}}_o$ (Wertz [2]).

Considering the satellite perfectly aligned with the target LVLH frame, the gravity gradient disturbance will result to be zero. The following charts shows the gravity gradient components behavior during a couple of orbit periods (13824 seconds):



To be noted as the z axis gravity gradient component in the Body frame have an order of magnitude of 10^{-20}, that in practice is zero as stated in literature theory.

3.2 – Solar Radiation Torque.

It is modeled in the ***Solar Radiation Torque*** block.

In order to model the radiation torque disturbance just the Solar radiation was considered, neglecting the Earth albedo and the Moon light. Furthermore the shadows on the satellite surfaces was neglected. It has been considered a satellite barycenter shift of -5 cm along the pitch axis y. The satellite geometry has been modeled through the **ST** matrix, inside the *Solar Torque* Matlab function, and it is a 7x10 matrix, where each column contains the information about the 10 surfaces of the satellite. The first three rows contain the components of the vector which connects the satellite barycenter with each surface center of pressure (in the *Body* reference), the fourth row contains the surfaces areas expressed in m^2 and the last three rows contain the components of the unit vectors normal to the surfaces (in the *Body* reference).

The *Solar Radiation Torque* Matlab function, through a *for* cycle, compute the torques due to each surface, it adds them in order to get the three solar radiation torque components in the *Body* reference, according to the following formula:

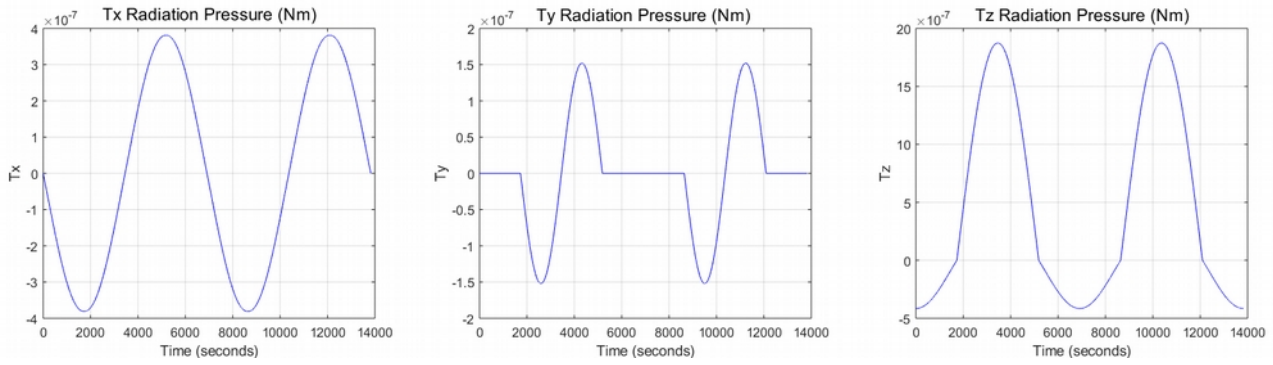
$$\boldsymbol{T}_{solar}=\sum_{i=1}^{10}\boldsymbol{r}_i\times\boldsymbol{F}_i=\sum_{i=1}^{10}\boldsymbol{r}_i\times\left(-A_i P\, d_i\left(\left(1-c_S\right)\hat{s}+2\left(c_S\cdot\hat{n}_i+\frac{1}{3}c_D\right)\hat{n}_i\right)\hat{n}_i\cdot\hat{s}\right)$$

where $\mathbf{r}_i$ are the components of the vector which connects the satellite barycenter with each surface center of pressure, A_i are the surfaces areas, c is the speed of light in the vacuum, P is the mean momentum flux given by F_e/c (where F_e=1358 W/m^2 is the solar radiation intensity at 1 A.U. from the Sun), $\hat{s}$ is the Sun direction vector (approximate to the ECI Sun unit vector as viewed from the satellite), the $\hat{n}_i$ are the unit vectors normal to the satellite surfaces, while c_S and c_D are the coefficients of specular reflection and diffusion reflecion, respectively.

Lastly d_i is a coefficient defined as follows:

$$d_i \begin{cases} 1 \; se \; \hat{n}_i\cdot\hat{s}>0 & \text{(enlightened surface)} \\ 0 \; se \; \hat{n}_i\cdot\hat{s}\le 0 & \text{(shadowed surface)} \end{cases}$$

The following graphics show the solar radiation torque components during a couple of orbital periods:



As will be seen later, the solar radiation torque is about an order of magnitude lower than other environmental torques, but it has to be underlined that, because of the chosen sun synchronous orbit, a 2R25R orbit, and the chosen orbital radius, the satellite never goes into eclipse and so the solar radiation torque is continuous, and it cannot be neglected over the entire mission lifetime, being a secular disturbance.

3.3 – Magnetic Torque.

It is modeled in the ***Earth Magnetic Field Torque*** block.

The Earth magnetic field torque is due to the interaction among the Earth magnetic field and the satellite residual magnetic field. Given the orbit inclination and the "low" orbit altitude, this torque has been considered of main importance, and so it has been modeled in the best possible way So it has been chosen the Spherical Harmonics model described in the appendix H of Wertz [2], defining the field until the sixth harmonic, using the IGRF_12 data, found on IAGA website, about the 2015 year and with the corrections for the 2018. In order to have a lighter computational load, the magnetic field coefficients have been described in a recursive manner, as explained in the aforementioned appendix H of Wertz [2]. Because of the method and model complexities, please refer to the aforesaid appendix to see the formulas used.

The magnetic torque $\mathbf{T_{mag}}$ is given by:

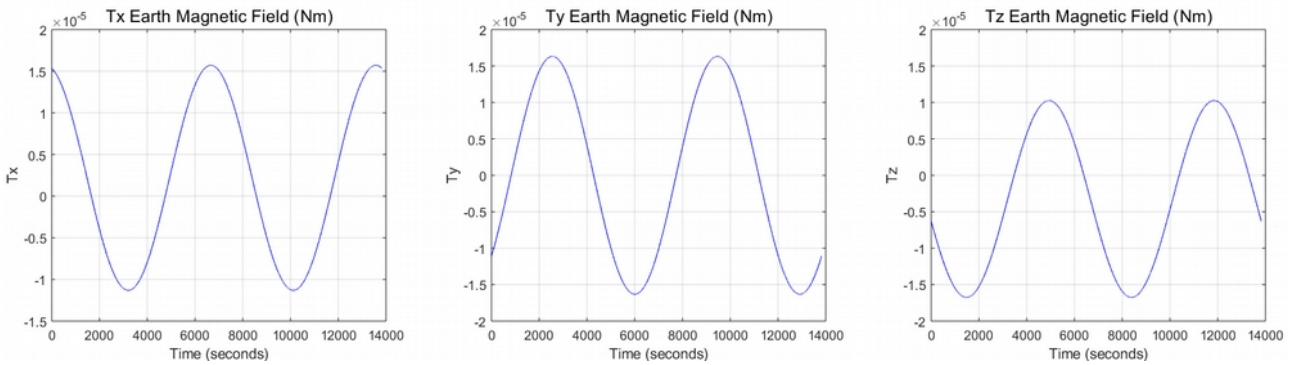
$$T_{mag} = \boldsymbol{D} \times \boldsymbol{B}$$

where **B** is the magnetic field in the satellite space location (in the *Body* reference frame), and **D** is the satellite residual magnetic dipole, defined as follows:

$$\mathbf{D} = [0.9 \quad 0.9 \quad 0.61]\ \mathrm{Am}^2.$$

The satellite total magnetic dipole is $D = 2\ \mathrm{Am}^2$.

The following graphics show the magnetic torque components during a couple of orbital periods:



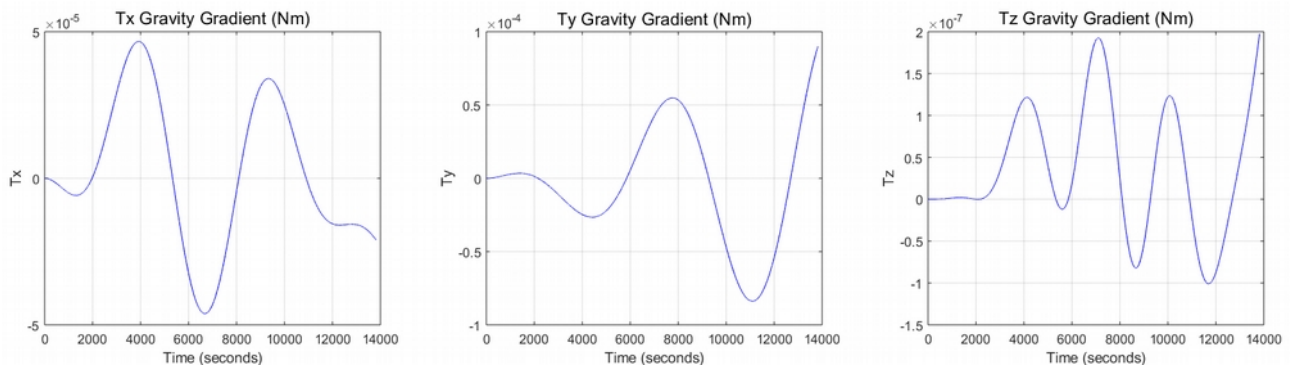
It could be noted that, because of the polar orbit, the Earth magnetic field torque is a considerable one, and in some cases stronger than the other disturbing torques.

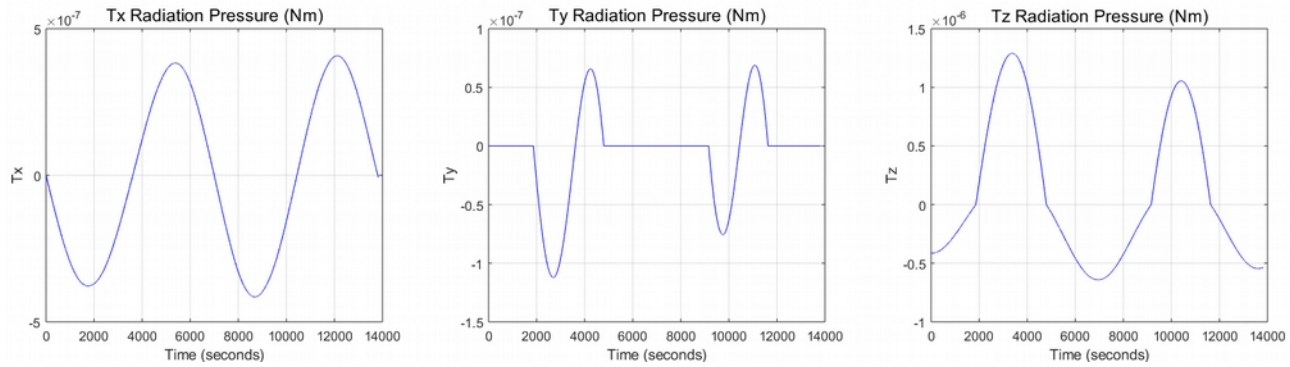
3.4 Motion under the total effect of disturbances.

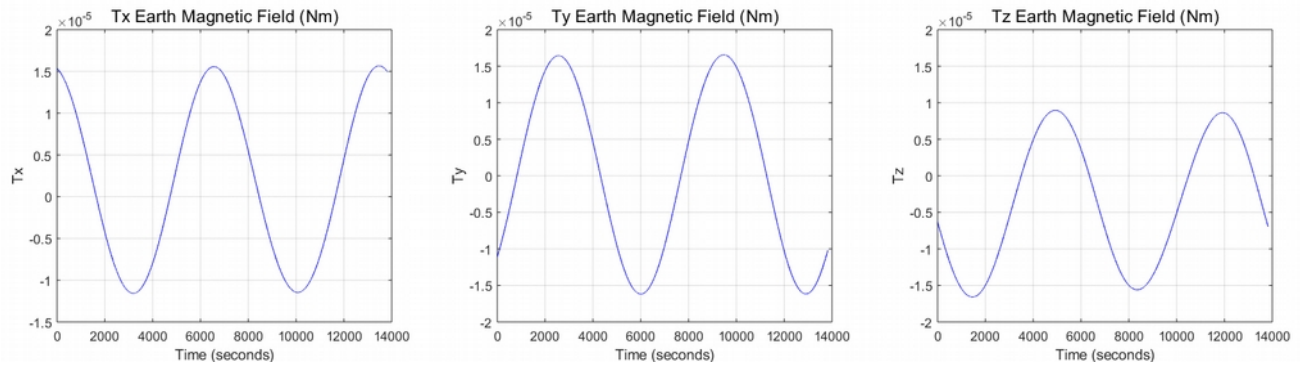
After setting up the following nominal initial conditions:

$$\begin{cases} \varphi_0=0 \\ \theta_0=0 \\ \psi_0=0 \end{cases} \quad \text{and} \quad \begin{cases} \omega_{x0}=0 \\ \omega_{y0}=n \\ \omega_{z0}=0 \end{cases} \quad \text{where} \quad \begin{cases} \varphi \text{ roll angle,} \\ \theta \text{ pitch angle,} \\ \psi \text{ yaw angle,} \end{cases}$$

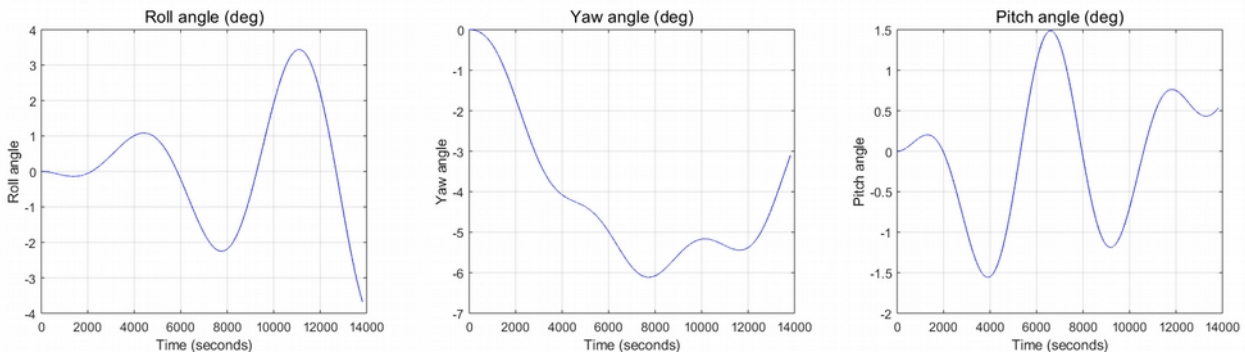
a simulation has been runned (over a two orbital periods time frame), in order to observe the disturbances effects over the satellite dynamics, getting the following graphics:







Finally, it can be seen how the satellite deviates from the desired position, exceeding the design specifications on angles offsets (imposed to be 0,01°) of more than two orders of magnitude, making clear the need for the attitude control (graphics on the following page).



4 – SATELLITE STABILITY.

4.1 Gravity Gradient Satellite Stability.

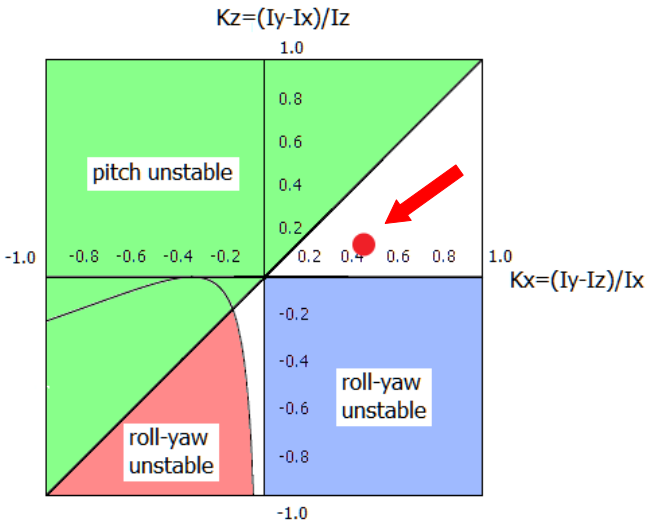
The satellite was designed so as to result stable under the Gravity Gradient effect, in fact it results to be: $I_y > I_x > I_z$. Furthermore, the moment s of inertia coefficients are:

$$K_z = \frac{I_y - I_x}{I_z} = 0,1055$$

$$K_x = \frac{I_y - I_z}{I_x} = 0,447$$

which satisfy the stability relation for roll and yaw: $(1+3K_x+K_xK_z)^2 > 16K_xK_z$.

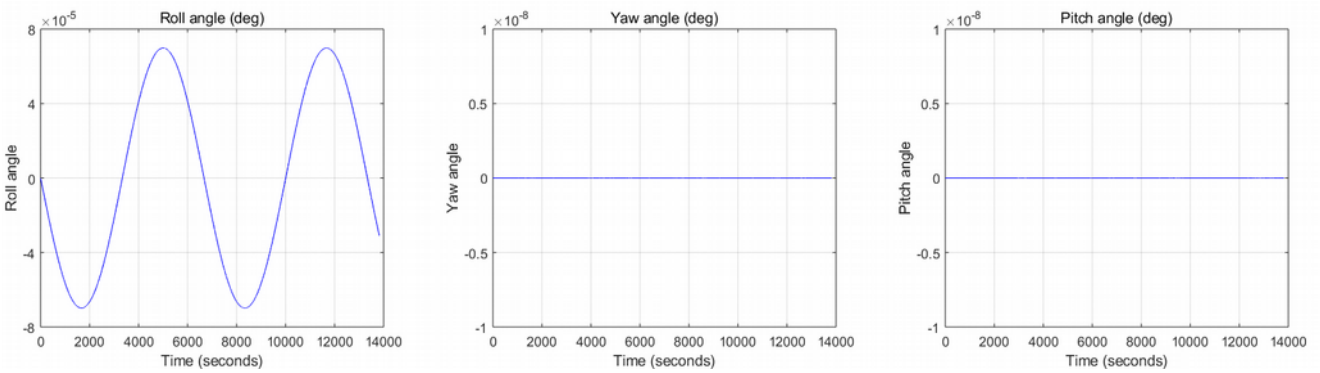
It is then possible to place the satellite configuration on the K_x/K_z diagram, that satys in one of the two allowable zones (the white colored zones in picture 4.1).



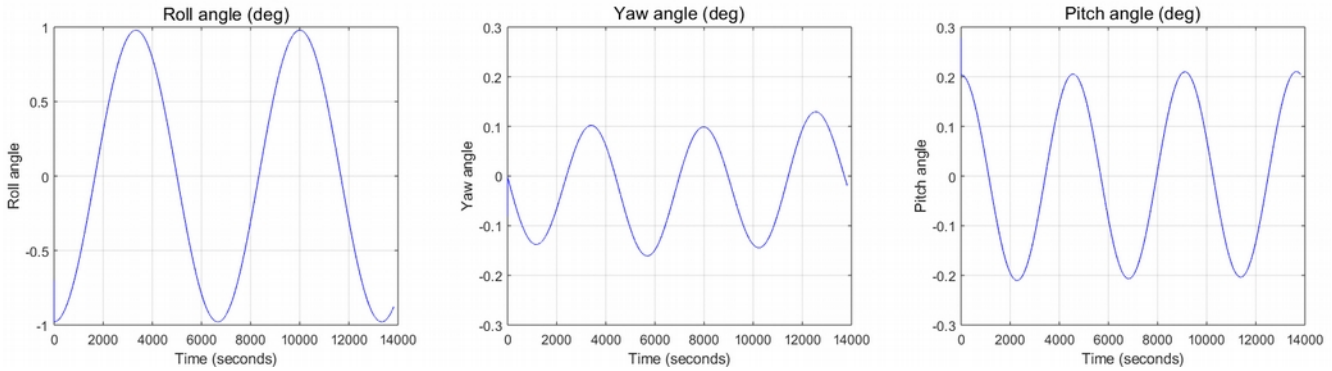
Pic. 4.1

Finally, it is possible to observe the graphics of gravity gradient effect on the satellite motion over a two orbital periods timeframe:

a) for the satellite body frame matching the target frame;



b) for the satellite body frame with a 1° offset about the target frame;



Note how in the case a) the satellite offset are well below the required specifications, suggesting that the study over the stability due to the satellite moments of inertia shown in picture 4.1 is correct and that such a satellite is then stable on all the three axes.

5 – SENSORS

5.1 Sensors choice.

The two sensors onboard of the satellite are modeled in the ***Sensors*** macro-block.

Given the assigned specifications, which imposed just a Sun sensor, finding an other kind of sensor was necessary, because of one sensor alone is not enough for the satellite attitude determination. An interesting choice, given the orbit kind, should be that of a magnetic field sensor, but such a sensor would not satisfy the Earth observation mission precision requirements, like not even a horizon sensor. A star sensor has then be chosen, mounted on the satellite opposite side (the surface with the unit vector $+\hat{y}$) to the one o which the sun sensor has mounted (the surface with unit vector $-\hat{y}$). It was not possible mount the star sensor on a surface with unit vector normal to that one of the Sun sensor (this would optimize the asset determination). The star and sun unit vectors are then not too much misaligned (read 5.2 e 5.3 sections). The star sensor has a precision an order of magnitude greater than that one of the Sun sensor, and the two sensors grant an asset determination within the constraints by working together.

5.2 Sun sensor.

It is modeled in the ***Sun Sensor*** block.

The Sun sensor is based on the Wertz [2] mathematical model. The block takes as input the Sun direction (which in ECI reference is the unit vector [1 0 0] initialized by the script ***ECI_Sun_direction.m***) as seen from the satellite *Body* reference frame, and through the rotation matrix **SunSRM** (initialized by the script ***Sensors_rotations.m***), the block rotates it in the sensor reference frame. Once the Sun direction has measured, introducing the sensor noise, the latter has again rotated in the Body frame and sent to the block which computes the attitude quaternion error.

For this mission the Sun sensor ISS-DX D5 by SolarMems Technologies (solar-mems.com) has chosen, which has a FOV of 10°x10° and a precision 0,005° (see note on page 22).

5.3 Star sensor.

It is modeled in the ***Star Sensor*** block.

The star sensor is based on the Wertz [2] mathematical model.The block takes as input the star direction (which in ECI is the unit vector generated by the script ***ECI_Star_direction.m***) as seen from the satellite *Body* reference frame, and through the rotation matrix **StarSRM** (initialized by the script ***Sensors_rotations.m***), the block rotates it in the sensor reference frame. Once the star direction has measured, introducing the sensor noise, the latter has again rotated in the Body frame and sent to the block which computes the attitude quaternion error.

For this mission the star sensor Autonomous Star Tracker AA-STR by Leonardo Airborne and Space Systems (leonardocompany.com) has chosen, which has a FOV of 20°x20° and a precision of 8 arcsec, that is 0,0022° (see note on page 22).

5.4 Sensors implementation and attitude error quaternion computation.

The output of the two sensors give the Sun and star directions in the *Body* reference frame, appropriately tainted by a *random* error induced by the sensors' precision. The measured directions are than obtained, which together with the Sun and star directions in the LVLH frame are sent to the ***Attitude determination from sensors*** block to generate the attitude quaternion error.

Instead of employ the **TRIAD** method, which compute two attitude frames starting from the available vectors to generate a direction cosines matrix from which obtain the angles errors (to be converted in an error quaternion), the mathematical **Direct Quaternion Method** has adopted (described in Markley [5]), directly obtaining the error quaternion from the measured and reference vectors.

In order to make the Simulink block at issue more readable (which has appropriate comment inside itself), the following names have been given t the Sun and star unit vectors in the two LVLH and Body reference frames:

- $\mathbf{b_1}$ star unit vector measured in *Body* frame;
- $\mathbf{b_2}$ Sun unit vector measured in *Body* frame;
- $\mathbf{r_1}$ star unit vector measured in LVLH frame;
- $\mathbf{r_2}$ Sun unit vector measured in LVLH frame;

The error quaternion has computed according to the following formula:

$$\boldsymbol{q} = c^{-(\frac{1}{2})} \cdot [(\boldsymbol{b_1} + \boldsymbol{r_1}) \cdot (\boldsymbol{b_2} - \boldsymbol{r_2}), (\boldsymbol{b_1} - \boldsymbol{r_1}) \times (\boldsymbol{b_2} - \boldsymbol{r_2})]$$

where c is the normalization factor equal to:

$$c = [(\boldsymbol{b_1} + \boldsymbol{r_1}) \cdot (\boldsymbol{b_2} - \boldsymbol{r_2})]^2 + |(\boldsymbol{b_1} - \boldsymbol{r_1}) \times (\boldsymbol{b_2} - \boldsymbol{r_2})|^2$$

The error quaternion, because of quaternions algebraic properties that are not mentioned here, must take on the value $\mathbf{q} = [\ 1\ 0\ 0\ 0]^T$ when the *Body* and LVLH (the target frame) reference frames coincide.

6 – CONTROLLER

6.1 The controller equations.

In order to design the controller, it is necessary to linearize the satellite dynamics Euler's Equations. In this case, because of the chosen reference frames, satellite is simply spinnig about the *pitch y axis*, and than the linearized equations are than:

$$\begin{cases} I_x \ddot{\alpha}_x + (I_z - I_y + I_x) n \dot{\alpha}_z - (I_z - I_y) n^2 \alpha_x = M'_{cx} \\ I_y \ddot{\alpha}_y = M'_{cy} \\ I_z \ddot{\alpha}_z + (I_y - I_x - I_z) n \dot{\alpha}_x + (I_y - I_x) n^2 \alpha_z = M'_{cz} \end{cases}$$

Note that the second equation is uncoupled with the other two.

The controller design is simplified gathering together some control torques terms, as follows:

$$\begin{cases} I_x \ddot{\alpha}_x = M'_{cx} - (I_z - I_y + I_x) n \dot{\alpha}_z + (I_z - I_y) n^2 \alpha_x = M_{cx} \\ I_y \ddot{\alpha}_y = M'_{cy} = M_{cy} \\ I_z \ddot{\alpha}_z = M'_{cz} - (I_y - I_x - I_z) n \dot{\alpha}_x - (I_y - I_x) n^2 \alpha_z = M_{cz} \end{cases}$$

Finally it has been noted that, because of the quaternion attitude parameterization, the angles errors are obtained by the following mathematical relations:

$$\begin{cases} \alpha_x = q_s \cdot q_{v1} \\ \alpha_y = q_s \cdot q_{v2} \\ \alpha_z = q_s \cdot q_{v3} \end{cases}$$

while the angular velocities variations $\dot{\alpha}_i$ are obtained simply computing the difference between the angular velocities that come out from the ***S/c Dynamics – Euler's equations*** block and the wanted ones.

6.2 Control laws.

They are modeled in the ***PID*** block.

Despite in the ***PID*** block the Integral control blocks are modeled, the K_I coefficients are all set to zero, making in fact the ***PID*** block a PD controller (proportional and e derivative).

The control action about all the body axes could be expressed as follows:

$$\frac{M_{ci}}{I_i} = -K_{pi}\alpha_i - K_{di}\dot{\alpha}_i \quad \text{where i= x, y, z.}$$

The equations become so:

$$I_i\ddot{\alpha}_i + K_{di}\dot{\alpha}_i + K_{pi}\alpha_i = 0 \quad \text{where i= x, y, z.}$$

By comparing the latter equations with the characteristic equation (that is just the denominator of the Laplace transform) $s^2 + 2\zeta\omega_n s + \omega_n^2 = 0$ the following quantities are defined:

system natural frequency $\omega_n = \sqrt{\frac{K_{pi}}{I_i}}$ where i= x, y, z.

and dumping ratio $\zeta_i = \frac{K_{di}}{\sqrt{4K_{pi}I_i}}$ where i= x, y, z.

and so, finally, the expressions that allow to compute the desired coefficients are obtained as follows:

$$K_{pi} = I_i\omega_n^2 \quad \text{and} \quad K_{di} = 2\zeta\omega_n I_i \quad \text{where i= x, y, z.}$$

Now, imposing that the system natural frequency ω_n be a multiple of the orbital frequency n, in the ***PID_control.m*** script the parameters ***multiplo*** (=900) and ***csi*** ($\zeta = 0{,}707$) have been defined in order to obtain the coefficients as functions of $\omega_n = multiplo \cdot n$.

The coefficient are than:

$$\begin{cases} K_{px}=994 \\ K_{py}=1059 \\ K_{pz}=615 \end{cases} \qquad \begin{cases} K_{dx}=1718 \\ K_{dy}=1831 \\ K_{dz}=1063 \end{cases}$$

The ***PID*** block sends the computed control torques to the actuators.

7 – ACTUATORS

7.1 Reaction wheels.

They are modeled in the ***Reaction Wheel Assembly*** block.

To control the satellite in this mission it has been chosen to use three Reaction Wheels, each one having its rotation axis aligned with each satellite principal axis respectively, instead of choosing a tetrahedral configuration with four reaction wheels, as it should be to achieve redundancy. No measure uncertainties on the rotors angular velocities have been introduced, nor delays on the command actuation. By the way, it has been set up a friction on the wheels, represented by the gain *fric* (friction) inside the block.

The ***Reaction Wheel VRW-1*** by Vectronic Aerospace (vectronic-aerospace.com) have been chosen, which specifications of interest are mentioned in the table 7.1 (see note on page 22).

Moment of Inertia	$2.0*10^{-3}$ kgm^2
Nominal speed	± 5000 rpm
Max. speed	± 6500 rpm
Angular momentum	0.5 Nms
Max torque	± 25 mNm

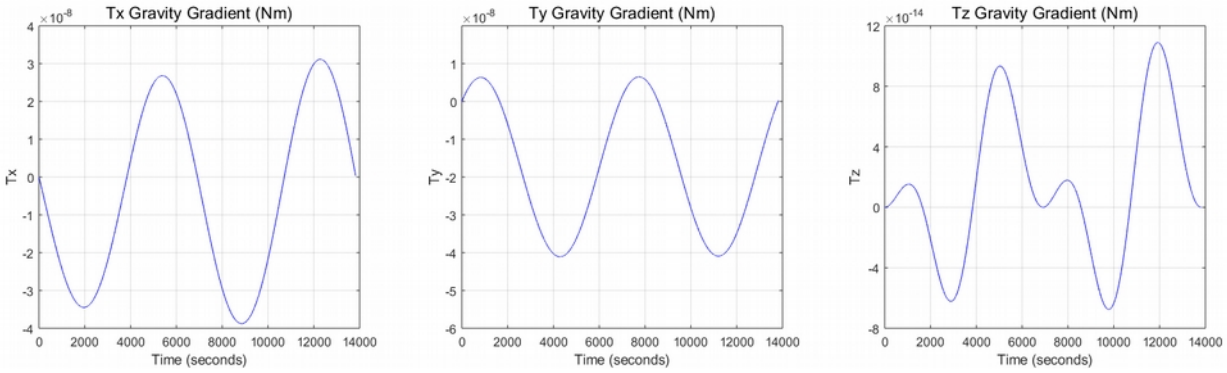
Table 7.1

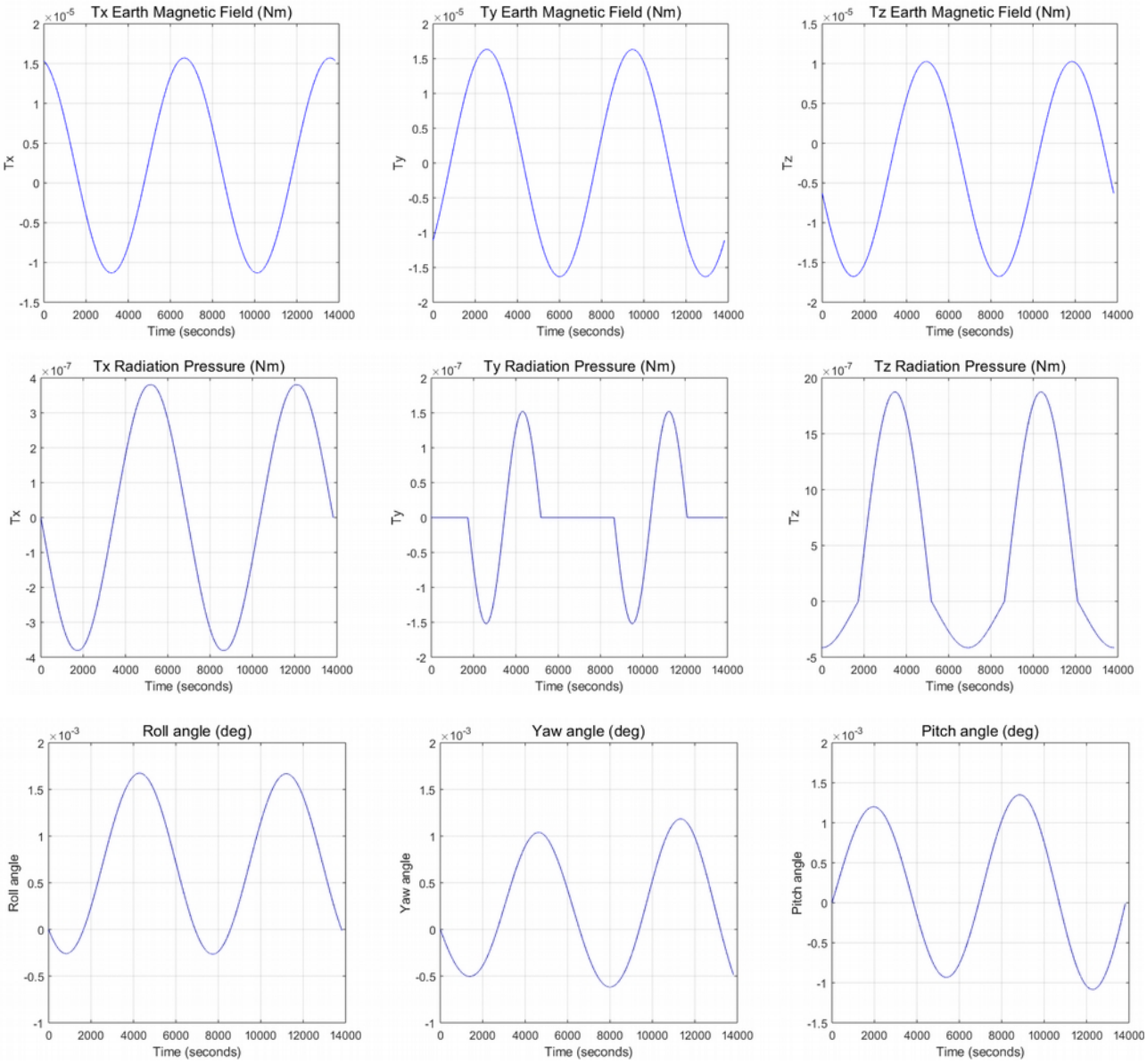
8 – WHOLE SATELLITE SIMULATIONS

Two simulations have been runned in order to check the controller goodness, which are described below.

8.1 Controlled motion starting from the nominal condition.

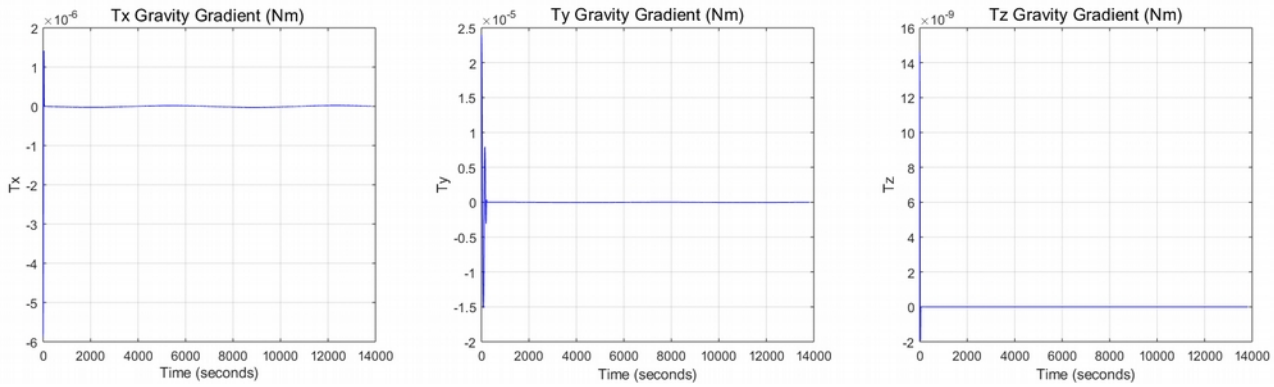
The graphics below are obtained considering the satellite starting perfectly aligned with the LVLH reference. Note as the angles swings are abundantly within the imposed constraints of 0,01°.

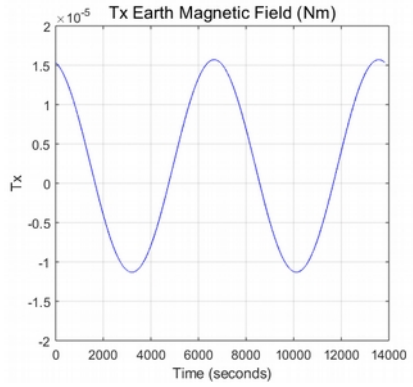


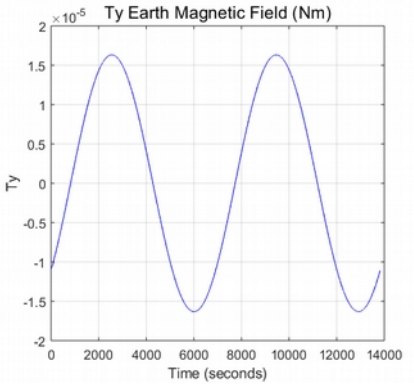


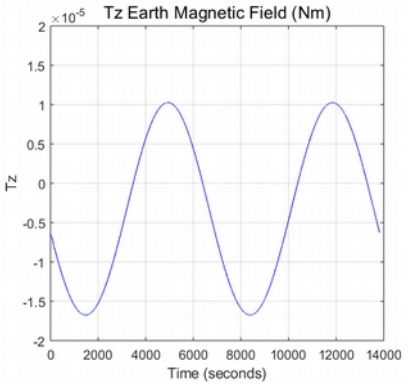
8.2 Controlled motion with 1 degree offset on the satellite axes.

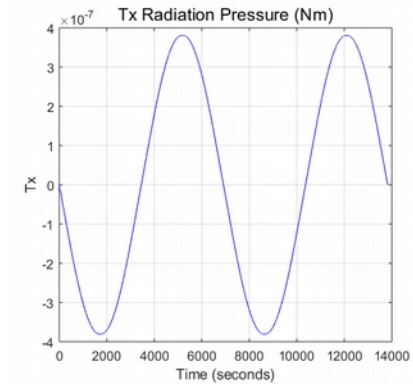
The graphics below are obtained considering the satellite starting with 1 degree offset about the LVLH reference frame. Note as, after a transient of few tens of seconds, the angles swings return inside the imposed constraints.

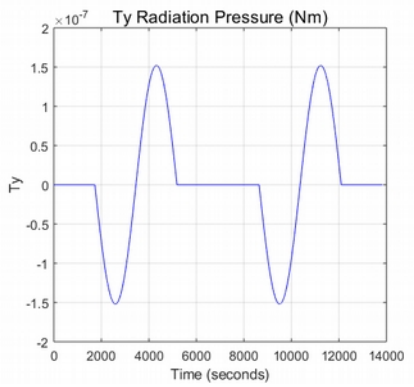


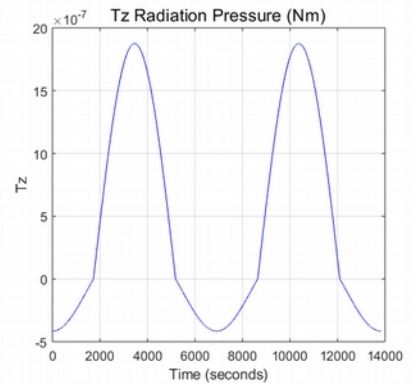

Tx Earth Magnetic Field (Nm)
×10^-5
Tx
Time (seconds)

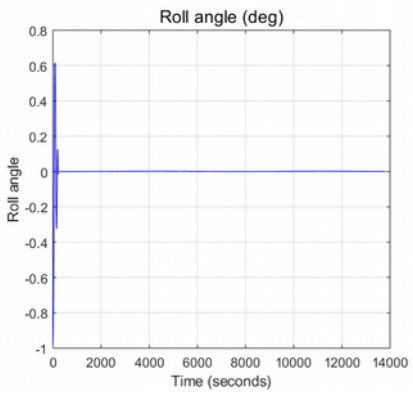

Ty Earth Magnetic Field (Nm)
×10^-5
Ty
Time (seconds)

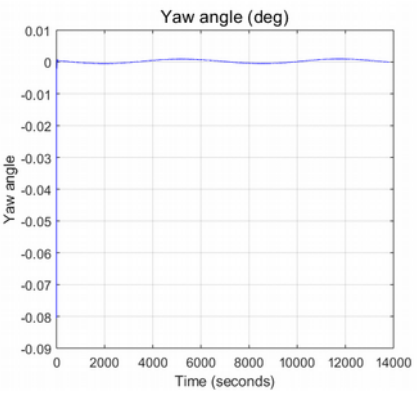

Tz Earth Magnetic Field (Nm)
×10^-5
Tz
Time (seconds)

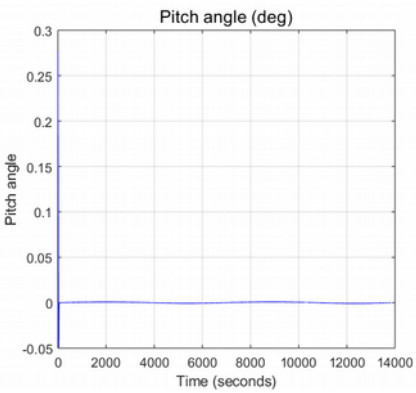

Tx Radiation Pressure (Nm)
×10^-7
Tx
Time (seconds)

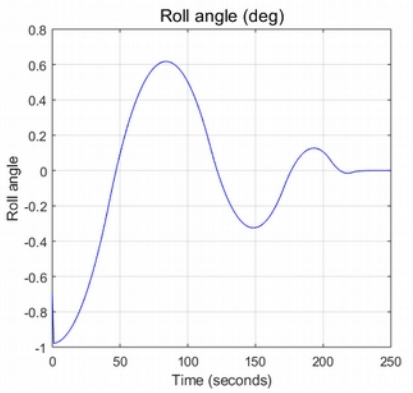

Ty Radiation Pressure (Nm)
×10^-7
Ty
Time (seconds)

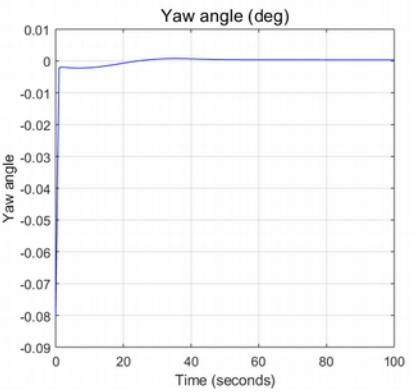

Tz Radiation Pressure (Nm)
×10^-7
Tz
Time (seconds)

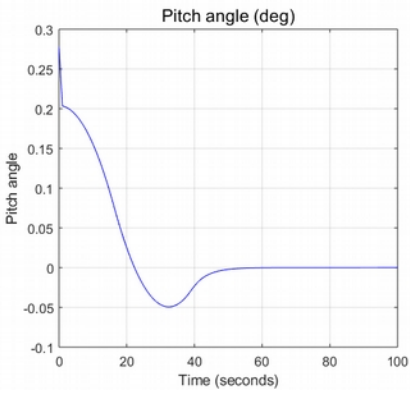

Roll angle (deg)
Roll angle
Time (seconds)


Yaw angle (deg)
Yaw angle
Time (seconds)


Pitch angle (deg)
Pitch angle
Time (seconds)


Roll angle (deg)
Roll angle
Time (seconds)


Yaw angle (deg)
Yaw angle
Time (seconds)


Pitch angle (deg)
Pitch angle
Time (seconds)

8.3 Controlled motion with measures noises turned on.

It has been tried to conduct simulations by turning on the noises on the sensor measurements, but unfortunately it has not been possible to filter them in an effective way, and so all the simulations runned have given unexpected results. By the way, the measures noises blocks could be found, turned off in the Simulink model, for the sole purpose of conceptual completeness.

8.4 Instructions to run the model.

In order to run a simulation, the **Satellite_Start** script must be started firstly.
To change the initial satellite orientation, it is necessary to modify the angles values (in degrees) inside the *Orientamento iniziale* block, highlighted in light green, and that is in the ***S/c Kinematics*** macro-block. The satellite orientation is set up with angles values which align the satellite itself to the LVLH target frame, and that are equal to [90 78.23 0] by a XZY rotation.

It is also possible to modify the ***multiplo*** and ***csi*** parameters into the ***PID_control*** script in order to change the controller gains.

Finally, it is possible to observe the satellite behavior in the “scope” block named ***Satellite Behavior*** and which is highlighted in green color, or in the “scope” blocks named ***Roll***, ***Pitch***, ***Yaw***, highlighted in green color as well.

Emanuele Nadalon

APPENDICES

A.1 – Symbols list in the Simulink model

a orbital radius;
Adapt adjustment matrix among LVLH and *Body* reference frames;
argp argument of perigee in degrees;
argpr argument of perigee in radians;
b1 Star unit vector in *Body* reference frame;
b2 Sun unit vector in *Body* reference frame;
C rotation matrix from LVLH to ECI;
C1i rotation matrix about the x axis of the angle i (orbital inlination);
C3argp rotation matrix about the z axis of the angle argp;
C3RAAN rotation matrix about the z axis of the angle RAAN (Right Ascension of Ascending Node);
cargp cosine of the argument of perigee;
ci cosine of the orbital inclination;
cRAAN cosin of the right ascension of the ascending node;
csi dumping coefficient of the PID controller;
Ct transpose of matrix C (rotation from ECI to LVLH);
Dx satellite magnetic residual dipole about the x axis;
Dy satellite magnetic residual dipole about the y axis;
Dz satellite magnetic residual dipole about the z axis;
ECIstd star direction in ECI;
ECIsund Sun direction in ECI;
esplir explemenatry angle of the orbit inclination in radians;
fric reaction wheels friction coefficient;
GM Earth gravitation constant;
I satellite principal moments of inertia matrix;
inc orbit inclination in degrees;
ir orbit inclination in radians;
IRw matrix of the reaction wheels inertia moments;
IRwinv IRw matrix inverse;
IRx moment of inertia of the reaction wheel about the x axis;
IRy moment of inertia of the reaction wheel about the y axis;
IRz moment of inertia of the reaction wheel about the z axis;
Ix satellite principal moments of inertia about x axis;
Iy satellite principal moments of inertia about y axis;
Iz satellite principal moments of inertia about z axis;
multiplo orbital frequency multiplier for the PID coefficients compuutation;
n orbital angular velocity or orbital frequency;
omegan multiple of the orbital frequency;
q satellite attitude quaternion;
qerr attitude error quaternion;
RAAN right ascension of the ascending node in degrees;
RAANr right ascension of the ascending node in radians;
sargp sine of the perigee argument;
si sine of the orbital inclination;
sRAAN sine of the right ascension of the ascending node;
st1 inclination angle of the star direction about the XZ ECI plane in radians;
st1d inclination angle of the star direction about the XZ ECI plane in degrees;

st2 inclination angle of the star direction about the XY ECI plane in radians;
st2d inclination angle of the star direction about the XY ECI plane in degrees;
StarSRM rotation matrix from Body reference to the star sensor reference frame;
SunSRM rotation matrix from Body reference to the Sun sensor reference frame;
r1 star unit vector in LVLH frame;
r2 Sun unit vector in LVLH frame;
TRwmax maximum torque provided by the reaction wheels (saturation limit);
V satellite tangential velocity;
wRwmax reaction wheels maximum angular velocity;
wx0 satellite initial angular velocity about x axis;
wy0 satellite initial angular velocity about y axis;
wz0 satellite initial angular velocity about z axis;

A.2 – Description of the scripts

(1) **Satellite_Start**: it starts all the following scripts contained in the model;

(2) **Orbit_generator**: it initializes the orbital parameters used in the *Orbit LVLH and ECI generation* block;

(3) **ECI_Sun_direction**: it generates the Sun unit vector in the ECI reference frame;

(4) **ECI_Star_direction**: it generates the star unit vector in the ECI reference frame;

(5) **Sensors_rotations**: it generates the rotation matrices in order to define the Sun and star unit vectors in the Sun and star sensors reference frame, respectively, with respect to the *Body* reference frame (these matrices are used in the *Sun Sensor* and *Star Sensor* blocks);

(6) **Satellite_inertia**: it generates the satellite inertia tensor;

(7) **Sat_mag**: it initializes the satellite magnetic dipoles which will be used in the *Earth Magnetic Field Torque* block;

(8) **Sat_omega_ini**: it initializes the satellite initial angular velocities about the three principal axes;

(9) **RWA**: it initializes the reaction wheels parameters, such as inertia moments, maximum angular velocities and maximum reaction torque;

(10) **PID_control**: it generates the K_{pi} and K_{di} constants used in the *PID* block.

A.3 – Bibliography

A.3.1 Books used during the project developing.

[1] Blanke M., Larsen M. B., *Satellite Dynamics and Control in a Quaternion Formulation*, DTU Technical University of Denmark, 2010
[2] Wertz J.R., *Spacecraft Attitude Determination and Control*, Dordrecht, Kluwer Academic Publishers Group, 1980
[3] Stark J., *Spacecraft Systems Engineering,* Chichester, UK, John Wiley & Sons, 2003
[4] Curtis H., *Orbital Mechanics for Engineering Students*, Burlington, MA, Elsevier Butterworth-Heinmann, 2005
[5] Markley F. L., *Attitude determination using two vector measurments*, Greenbelt, MD, NASA's Goddard Space Flight Center, 1981.

A.3.2 Other consulted books but not recalled in the report text.

[6] Sidi M. j., *Spacecraft Dynamics and Control - A practical Engineering Approach*, New York, Cambridge University Press, 1997
[7] Gran R. J., *Numerical Computing with Simulink, - vol. I- Cretaing Simulations I*, Norfolk, Massachussetts, Mathematical Analysis Company, 2007
[8] Griffin M. D., *Space Vehicle Design*, Reston, Virginia, AIAA education series, 2004
[9] Tewari A., *Modern Control Design With MATLAB and SIMULINK*, Chichester, UK, John Wiley & Sons, 2002
[10] Mangiascale L., Fondamenti di Automatica e Controllo di Veicoli Aerospaziali, Roma, ARACNE Editrice, 2008.

A.4 – Acknowledgements

I sincerely thank following people and Institutions because they timely helped me when I asked for their support in the development of this project:

- Prof. Mogens Blanke, DTU Electrical Engineering Department, Technical University of Denmark;
- Prof. Paolo Massioni, INSA de Lyon, France;
- SCI Team Copernicus Services Coordinated Interface ESA service operated by Airbus Defence and Space Intelligence.

Index

Note: below are the links to the sensors and reaction wheels datasheets.

Sun sensor: https://www.solar-mems.com/solar-tracking/

Star sensor: https://www.leonardocompany.com/it/products/aastr?f=/search

Reaction wheels: https://www.vectronic-aerospace.com/reaction-wheels/