



TABLES OF CONTENTS

Contents

1. INTRODUCTION	5
2. OBJECTIVES	5
3. ATTITUDE DEFINITIONS AND REQUIREMENTS OF IGOSAT.....	8
3.1 Attitude Usage	8
3.2 Reference Frames	9
3.3 AOCS Requirements-IGOSat.....	11
4. BACKGROUND	13
4.1 Rotations	13
4.2 Angular Velocity	13
4.3 Earths Magnetic field	14
4.3.1 Theory and Mathematical Models	15
4.3.2 Comparison of Magnetic Fields -IGRF & Tilted Dipole model	17
4.4 Sun vector from ephemeris data	18
5. ATTITUDE DYNAMICS AND KINEMATICS.....	19
5.1 Dynamics	19
5.2 Kinematics	20
5.3 Environmental Disturbances Torques.....	21
5.3.1 Gravity-Gradient Torque	22
5.3.2 Aerodynamic torque.....	23
5.3.3 Magnetic residual torques.....	24
6. ATTITUDE CONTROL.....	25
6.1 Actuator Dynamics.....	25
6.2 De-tumbling Control	27
6.3 PID Control	28
6.4 Combined De-tumble & PID Control.....	29
7. ATTITUDE DETERMINATION.....	30
7.1 State vector	33
7.2 Filter Propagation & Prediction	33
7.3 Measurement model & Update	34
8. SIMULINK	37
Operating instructions	41
9. RESULTS	43
9.1 Detumbling Results	43
9.2 PID Control Results.....	47
9.3 Combined De-tumble & PID Control.....	48
9.4 Optimal Gain Calculation	49

Ionospheric and gamma-ray Observations Satellite		Ref. : SCA-DD-02
		Edition: 1 Date: 05-09-2016
		Revision : 2 Date :
		Page : 3

9.5 Attitude Estimation	55
10. AUTOCODE GENERATION	56
11. CONCLUSION.....	58
12. FUTURE WORK	58
13. REFERENCES	59
14. APPENDIX	60

LIST OF FIGURES

Figure 1: Typical Attitude control loop.....	6
Figure 2: IGOSat AOCS Communication-subsystems	7
Figure 3: Placing satellite axes relative to the PC port 104[5]	9
Figure 4: Visualization of reference frame in VTS	10
Figure 5: System modes on AOCS [2]	11
Figure 6: Local orbit & Earth Inertial reference frames	13
Figure 7: Earth’s Magnetic field map	14
Figure 8: Comparison of Magnetic field-IGRF & Tilted dipole.....	17
Figure 9: Sun vector calculated from ephemeris for 1 orbit	18
Figure 10: Satellite dynamics & kinematics in Simulink	20
Figure 11: Inside the Kinematics block.....	21
Figure 12: Inside the dynamics block	21
Figure 13: Disturbance models in Simulink	22
Figure 14: Gravity gradient over the orbit	23
Figure 15: Gravity gradient disturbance in simulink	23
Figure 16: Magnetic residual torques in Simulink.....	24
Figure 17: Comparison of disturbance torques.....	25
Figure 18: Magnetic torque rod & air-core torquer[13]	26
Figure 19: Magnetic moment, field and Torque relation.....	27
Figure 20: Satellite controllability with magneto-torquers[25]	27
Figure 21: Combined de-tumbling & PID control in Simulink	29
Figure 22: Attitude Determination System-process.....	31
Figure 23: Magnetic field when magnetometer is turned off.....	32
Figure 24: EKF Process.....	36
Figure 25: Angular rates & Magnetic moment : B-dot control	44
Figure 26: Angular rates & Magnetic moment : B-bang bang control.....	44
Figure 27: Angular rates & Magnetic moment : Desired torque control law	45
Figure 28: Angular rates & Magnetic moment(40°/sec) : B-bang bang control	45
Figure 29: Angular rates & Magnetic moment(40°/sec) : B-dot control.....	46
Figure 30: Angular rates & Magnetic moment(40°/sec): Desired torque control law.....	46
Figure 31: Nominal mode(PID) control for 10 orbits.....	48
Figure 32: Combined De-tumble & PID controller	49



Figure 33: Nominal mode with $k_p = 4e^{-6}$, $k_i = 2e^{-10}$ and $k_d = 2e^{-2}$ 50

Figure 34: Nominal mode with $k_p = 4e^{-5}$, $k_i = 2e^{-10}$ and $k_d = 2e^{-2}$ 51

Figure 35: Nominal mode with $k_p = 4e^{-4}$, $k_i = 2e^{-10}$ and $k_d = 2e^{-2}$ 52

Figure 36: Mapping of various PID gains54

Figure 37: Angular velocity & gyro bias estimated by EKF55

Figure 38: Attitude estimation results.....55



1. INTRODUCTION

The IGOSAT (Ionospheric and Gamma-ray Observation SATellite) is the first 3U CubeSat standard nanosatellite of the University Paris Diderot, APC (AstroParticule et Cosmologie) and IPGP (Institut de Physique du Globe de Paris) laboratories gathered within the LabEx (Laboratory of Excellence) UnivEarthS initiative. It is also supported by CNES via JANUS program. This satellite is scheduled to be launched at the beginning of 2018 for a one-year mission around the Earth on a quasi-polar orbit at an altitude of about 600 km. The nanosatellite carries two payloads to fill its two scientific missions:

- A scintillator for the study of gamma radiation in polar cusps and the South Atlantic Anomaly (SAA).
- A dual-frequency GPS receiver for the study of the total electron content (TEC) of the ionosphere

After deployed into space and the lack of control, the attitude of satellite changes in unpredictable directions. To carry out the tasks and meet the scientific and technical specifications, the satellite must point to specific targets, hence the need for an attitude and orbit control system (AOCS).

2. OBJECTIVES

Spacecraft control is usually synonymous with “Attitude Control,” the engineering discipline of keeping a satellite or spacecraft pointed in the right direction.

As an engineering discipline, Satellite Attitude Control system embodies four distinct areas:

1. Control system design
2. Dynamics and modeling of systems
3. Software design
4. User interface design

Usually the system aspects of a spacecraft control system are more important than the control laws themselves and often much more difficult to realize.

1. Control System Design

Control system design can be further decomposed into:

1. Attitude determination
2. Attitude control
3. Control distribution

This is the important part and has to be realized in the preliminary design process with the mission requirements and specifications.

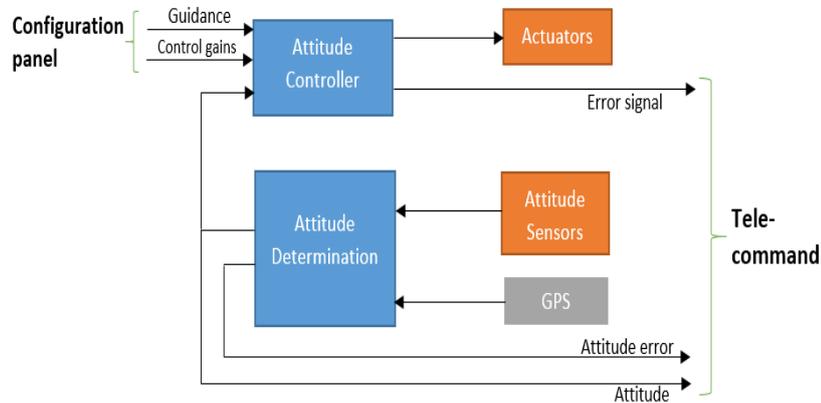


Figure 1: Typical Attitude control loop

2. Dynamics and Modeling of Systems

The second category can be decomposed into:

1. Modeling
2. Simulation

This section involves the implementation of the models into numerical simulations. Since it is rarely practical to test control systems on real satellites, the designer must rely on simulation to validate his or her designs. The modelling ranges from the satellite non-linear dynamics to the possible environmental disturbances- gravity gradient, aerodynamic & magnetic residual torques etc.

Simulations can range from all software running on the same platform, to software models with the control system running on a flight or equivalent board or one in which actuator and sensor hardware are integrated into the simulation(Matlab/Simulink).

3. Software Design

The third area is the implementation of both control systems and simulations in software. This is of critical importance.

Ultimately, spacecraft control algorithm is written in C/C++ or a block diagram language. Most of a spacecraft control system has little to do with control theory, but rather is related to how the satellite will operate. Aside from controlling the satellite, the software must:

- Implement the user interface (command and telemetry)
- Switch operational modes
- Provide fault detection
- Provide redundancy management
- Plan maneuvers, operations, etc.



4. User Interface Design

The fourth category is the most important. Most satellites that are lost are the result of operator error; operator error is often due to user interface problems. A user interface problem can be a wrong command, or one whose effect is not completely understood.

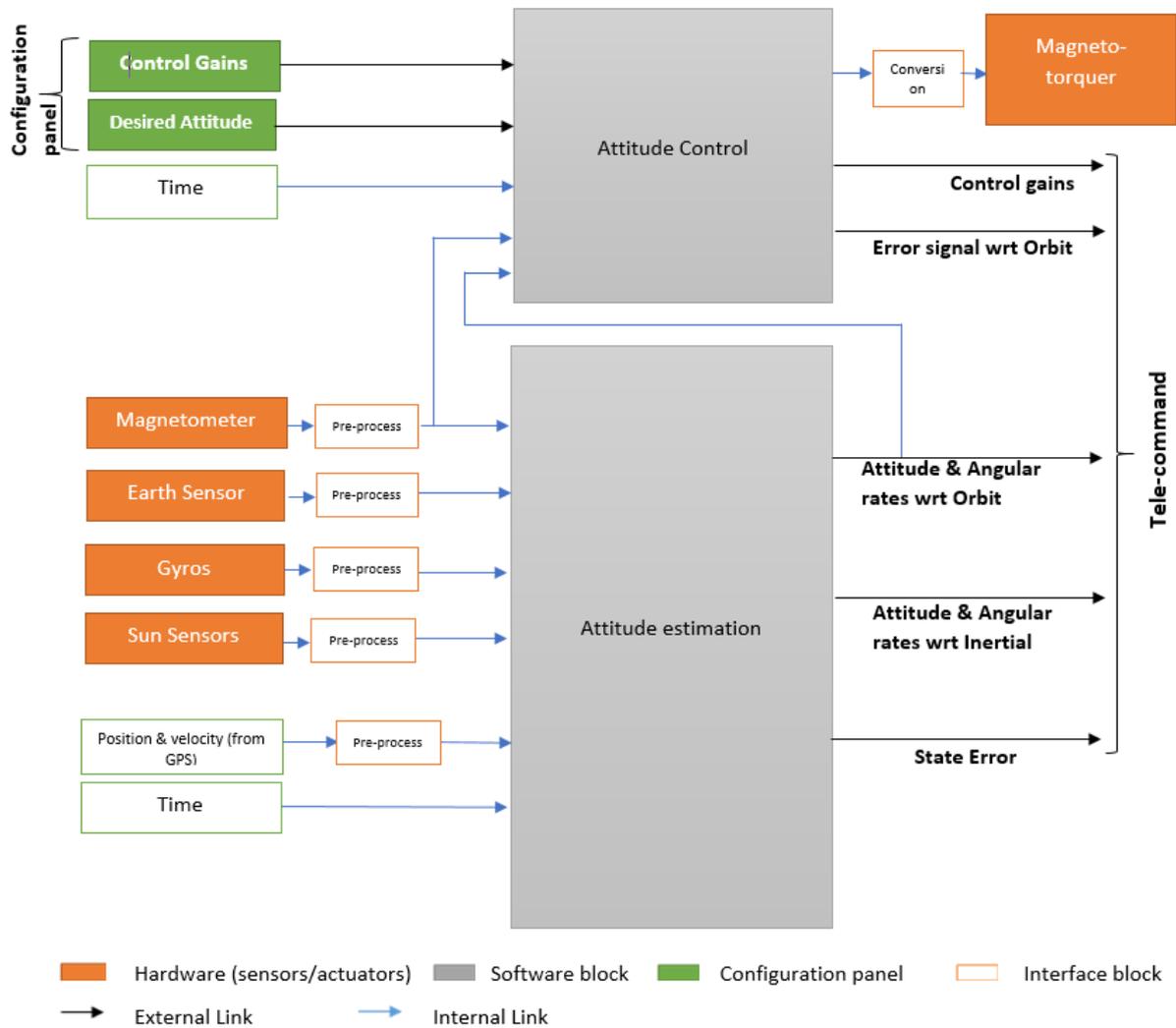


Figure 2: IGOSat AOCS Communication-subsystems



3. ATTITUDE DEFINITIONS AND REQUIREMENTS OF IGOSAT

3.1 ATTITUDE USAGE

The attitude used in the simulation & for the flight software are Quaternions of convention-2. While the Tele-command and attitude interpretation are carried out by Euler 321.

Why quaternions?

Quaternions have no inherent geometric singularity as do Euler angles. Moreover, quaternions are well suited for onboard real-time computer because only products and no trigonometric relations exist in the quaternion kinematic differential equations.

If (ψ, θ, φ) are euler321 angles from R_r to R_v then the rotation matrix is given by,

$$M_{R_r \rightarrow R_v} = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \sin \varphi & \cos \varphi \end{bmatrix} \quad \dots(1)$$

$$= \begin{bmatrix} \cos \psi \cos \theta & \cos \psi \sin \theta \sin \varphi - \sin \psi \cos \varphi & \cos \psi \sin \theta \cos \varphi + \sin \psi \sin \varphi \\ \sin \psi \cos \theta & \sin \psi \sin \theta \sin \varphi + \cos \psi \cos \varphi & \sin \psi \sin \theta \cos \varphi - \cos \psi \sin \varphi \\ -\sin \theta & \cos \theta \sin \varphi & \cos \theta \cos \varphi \end{bmatrix}$$

Computation of angles from matrix components:

Limitations of Euler 321

$$\psi = \arctan \frac{m_{2,1}}{m_{1,1}}$$

$$\theta = \arcsin(-m_{3,1})$$

$$\varphi = \arctan \frac{m_{3,2}}{m_{3,3}}$$

Axis	Angle	value
Z-axis	Yaw(ψ)	$\psi \in [-\pi : \pi]$
Y-axis	Pitch(θ)	$\theta \in \left[-\frac{\pi}{2} : \frac{\pi}{2}\right]$
X-axis	Roll(φ)	$\varphi \in [-\pi : \pi]$

The quaternion that transforms the vector in R_v to vector in R_r is denoted by Q_{R_v/R_r}

$$Q_{R_v/R_r} = (\bar{u}, p) = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \quad \text{and} \quad q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1 \quad \dots(2)$$

$$M_{R_r \rightarrow R_v} = \begin{bmatrix} 1 - 2(q_2^2 + q_3^2) & 2(q_1q_2 - q_4q_3) & 2(q_1q_3 + q_4q_2) \\ 2(q_1q_2 + q_4q_3) & 1 - 2(q_1^2 + q_3^2) & 2(q_2q_3 - q_4q_1) \\ 2(q_1q_3 - q_4q_2) & 2(q_2q_3 + q_4q_1) & 1 - 2(q_1^2 + q_2^2) \end{bmatrix}$$

It is also possible to express the rotation matrix in quaternion formulation



$$\begin{aligned}
 q_1 &= \text{sgn}(m_{3,2} - m_{2,3}) \cdot \frac{1}{2} \sqrt{1 + m_{1,1} - m_{2,2} - m_{3,3}} \\
 q_2 &= \text{sgn}(m_{1,3} - m_{3,1}) \cdot \frac{1}{2} \sqrt{1 - m_{1,1} + m_{2,2} - m_{3,3}} \\
 q_3 &= \text{sgn}(m_{2,1} - m_{1,2}) \cdot \frac{1}{2} \sqrt{1 - m_{1,1} - m_{2,2} + m_{3,3}} \\
 q_4 &= \frac{1}{2} \sqrt{1 + m_{1,1} + m_{2,2} + m_{3,3}}
 \end{aligned} \tag{3}$$

The conversion from Euler 321 to quaternion or vice versa is computed by comping & decomposing the rotation matrix $M_{R_v \rightarrow R_e}$. The more detail information on quaternion operations are discussed in [20]. A Simulink library is built ('MySimulinkLib') with the functions on quaternion operations and conversion to euler angles.(see Appendix 1).

3.2 REFERENCE FRAMES

The following reference frames are used in this report.

- Satellite reference frame(R_{sat})
- local reference frame(R_{loc})
- Inertial reference frame(R_g)
- Earth center earth fixed reference frame(R_{ecf})
- Target reference frame (R_c)

Satellite reference frame(R_{sat})

It is fixed wrt to the satellite body. The origin of R_{sat} is taken at the geometric center in accordance with the CubeSat specifications (DR3). For details, refer to the Interface Control Document [5].

- Z_{sat} axis will be along the length of nanosatellite.
- Y_{sat} axis will be opposite to the port PC104 electronic cards as shown in **Figure 3**.
- X_{sat} is given by the cross product between Z_{sat} and Y_{sat} to complete the Right hand System.

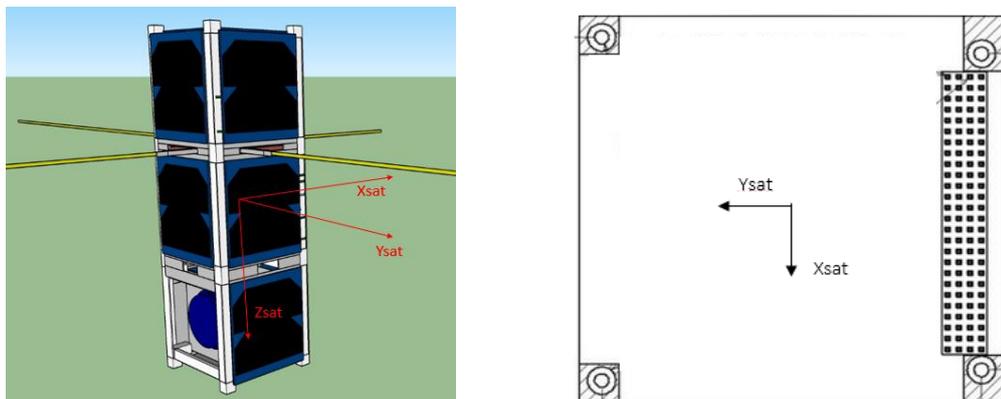


Figure 3:Placing satellite axes relative to the PC port 104[5]



Local reference frame(R_{loc})

It is local orbital frame taken such a way that,

- Z-axis is towards the earth zenith.
- X- axis towards the velocity of the satellite.
- Y-axis is to complete the Right hand system and also perpendicular to the orbital plane.

Inertial reference frame(R_g)

It is the Earth centered inertial frame, XYZ- axis are fixed and the epoch is taken at J2000/EME2000.

Earth center earth fixed reference frame(R_{ecf})

Origin is at the earth center, Z-axis towards the north pole, X-axis towards the vernal equinox at the green witch meridian and Y-axis to complete the right hand system rule. The X & Y axis rotates with the earth.

Target reference frame (R_c)

(In French: "Repère de consigne R_c ") It is the desired control frame that has to be achieved wrt the

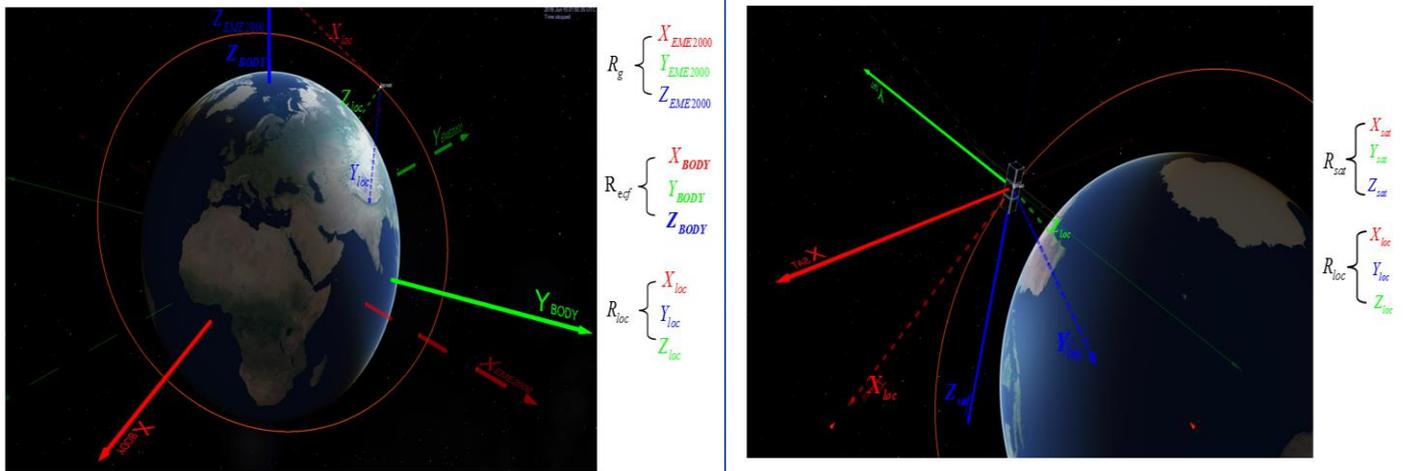


Figure 4: Visualization of reference frame in VTS

local orbital frame. It is also called desired attitude or guidance in our context. Satellite orientation R_{sat} has to be controlled over R_c . This reference frame is defined by

3.3 AOCS Requirements-IGOSat

Operating modes of AOCS

ID	Modes	Requirements
STB-PLA-XX	De-tumbling	The ACS must be able to damp the initial rotation rates of about ± 40 deg/sec on each axes from the launch in less than a week.
STB-PLA-XX	Nominal	It is the mission mode pointing GPS antenna to GPS satellite. In this mode ACS must be nadir pointed with a spin of 0.13 deg/sec in Zsat axis.
STB-PLA-XX	Caillou	ACS is turned off. This mode should consume less power than the nominal mode

System modes on AOCS

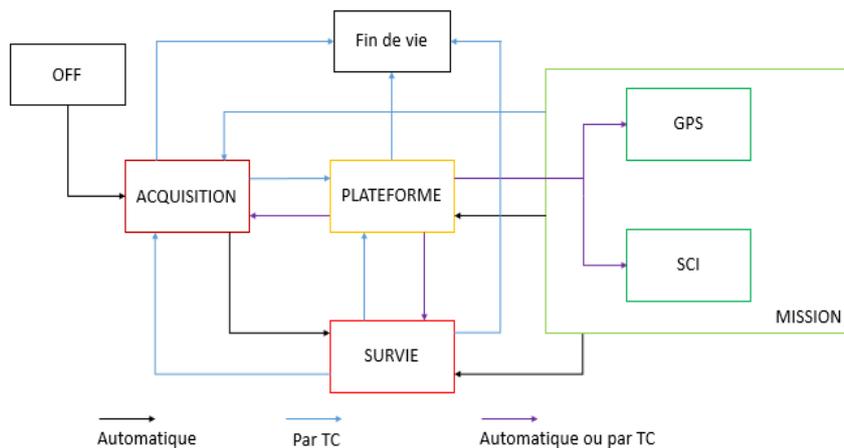


Figure 5: System modes on AOCS [2]

OFF mode is the satellite mode when every sub-system is off inside the POD or when the onboard computer reboots.

ACQUISITION mode is characterized at the beginning by the starting of the ODB and the sequence of the ignition of the different sub-system. It is also the deployment of the antennas and the “detumbling” phase, to reduce the rotation velocity of the satellite if the level of energy inside batteries is enough. A precise diagnostic will be done during this mode each time it is activated.

PLATEFORME mode is a waiting mode. The satellite transmits here his TM (Telemetry) and operates a soft diagnostic.

SURVIE mode is characterized by the priority given to the reception of power from the sun. Consequently, if the level of battery is too low, we stop the operations and just wait to recover a percentage of battery higher than 50%. If we are below 50%, the ACS (Attitude Control System) is turned off (mode caillou).

MISSION mode is when one or both instruments are activated.

FIN DE VIE mode is requested by the LOS (Loi sur les Opérations Spatiales) to ensure the passivation of the satellite before its re-entry in the atmosphere, less than 25 years after the launch. All the electrical equipment is cut-off and AOCS is deactivated.

ACS requirements [6]

ID	Requirements
STB-PLA-10	The ACS must determine the attitude of the satellite to +/- 5 ° in all axes
STB-PLA-11	The ACS must control the satellite attitude +/- 5 ° in all axes
STB-PLA-12	The ACS should be able to receive and process the instructions given by ODB
STB-PLA-13	The ACS shall transmit data to HK ODB
STB-PLA-14	The ACS must allow to compensate the disturbances induced by the deployment in less than a week.

Other requirements on ACS

- The *De-tumble mode* has to recover from the initial tumble condition and reach a stable condition of ≤ 0.5 deg/sec on each axes of the satellite with the magnetic-torquers.
- The *Mission mode* is the nominal mode of the IGOSat. It is activated, when the satellite has reached a stable condition of ≤ 0.5 deg/sec on each axes of the satellite.
- *Caillou mode* is activated, when the satellite is in low battery state (< 50 %) or due to the malfunction of the subsystem(sensor/actuator).
- The *De-tumble mode* should take less than a week to reach a slow steady tumble condition.
- The ACS must control & determine the satellite attitude up to $\pm 5^\circ$ in all the satellite axes.
- The ACS Controller should have a sampling rate of 1 Hz.
- The sensors and actuator are sampled at the same rate as of controller i.e 1 Hz.



4. BACKGROUND

4.1 Rotations

Let $R_{A/B}$ or R_{AB} or $R_{B \rightarrow A}$ be the rotation matrix from frame B to frame A.

If V_A and V_B are the vectors in frame A and B then,

$$V_A = R_{A/B} V_B \quad \dots(4)$$

The rotation matrix from local orbit frame to Inertial frame is given by,

$$R_{R_{loc}/R_g} = \begin{bmatrix} -\sin u \cos \Omega - \cos u \cos i \sin \Omega & -\sin u \sin \Omega + \cos u \cos i \cos \Omega & \cos u \sin i \\ -\sin i \sin \Omega & \sin i \cos \Omega & -\cos i \\ -\cos u \cos \Omega + \sin u \cos i \sin \Omega & -\cos u \sin \Omega - \sin u \cos i \cos \Omega & -\sin u \sin i \end{bmatrix} \quad \dots(5)$$

Which is a rotation based on orbital elements.

4.2 Angular Velocity

The angular velocity ω , is used to examine the angular displacements that occur over time. Angular velocities are dependent on the frame of reference, and are designated by $\omega_{c/a}^b$, which is a rotation from R_c with respect to R_a as seen by R_b .

Angular velocities add, but only when they are in the same reference frame. For example, the following relation is valid

$$\omega_{R_{sat}/R_g}^{R_{sat}} = \omega_{R_{sat}/R_{loc}}^{R_{sat}} + \omega_{R_{loc}/R_g}^{R_{sat}} \quad \dots(6)$$

When the angular velocities are in different reference frames, however, it is necessary to perform rotations. This is evident in

$$\omega_{R_{sat}/R_g}^{R_{sat}} = R_{R_{sat}/R_{loc}} \omega_{R_{loc}/R_{loc}}^{R_{loc}} + R_{R_{sat}/R_g} \omega_{R_{loc}/R_g}^{R_g} \quad \dots(7)$$

where the angular velocity is seen by the satellite body frame.

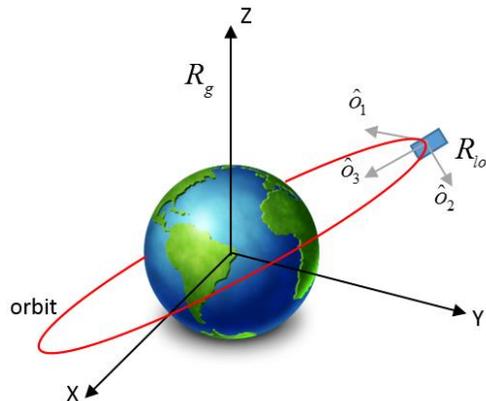


Figure 6: Local orbit & Earth Inertial reference frames

If ω_c is the orbital angular velocity, then the angular velocity of the local orbit frame with respect to the inertial frame is given by,



$$\Omega_{R_{loc}/R_g}^{R_{loc}} = \begin{bmatrix} 0 \\ -\omega_c \\ 0 \end{bmatrix} \quad \dots(8)$$

This is seen in local orbital frame.

4.3 EARTH'S MAGNETIC FIELD

Magneto-torquers are the only actuators for IGOsat. Magnetic control of spacecraft is dependent upon the local magnetic field. The magnetometer measures the local magnetic field of the earth in the spacecraft coordinates system. A discussion on the geomagnetic field is presented in this section. Mathematical models of the geomagnetic field using the International Geomagnetic Reference Field(IGRF) coefficients are implemented and compared with tilted dipole model. In addition, the methods used in the simulation and hardware required for measuring magnetic fields are discussed.

Geomagnetic Field

The magnetic field always flow from North to south pole. In case of earth, the geometric north is the South pole of earth's magnetic field and geometric south is the north pole of the earth's magnetic field. The magnetic field around the Earth resembles that of a uniformly magnetized sphere, or a dipole, which is tilted as shown in **Figure 7**. The fact that it approximates a tilted dipole was discovered in 1600 by William Gilbert, and was published in his treatise *De Magnete*[7]. In 1635, Gellibrand was the first to show that the geomagnetic field is both time and position dependent.

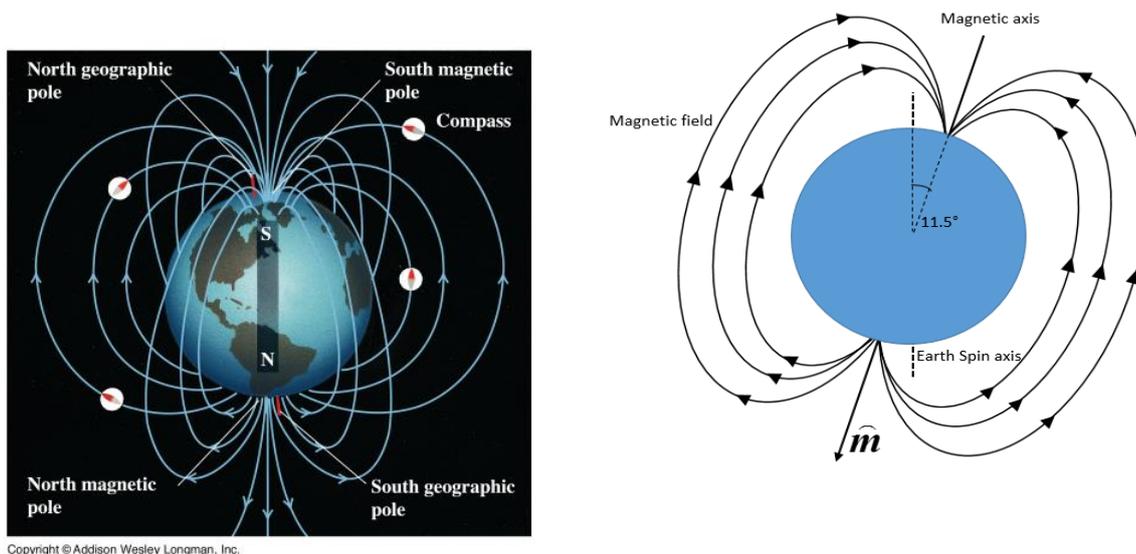


Figure 7: Earth's Magnetic field map

The strength of the magnetic field is approximately 30000 nT at the equator and 60000 nT at the poles on the surface of the Earth, as mentioned in the Geological Survey of Canada[24].

The magnetic dipole axis, designated as \hat{m} in **Figure 7**, is located at 79.8° N latitude and 107.0° W longitude, in the year 1999. This location is near the Ellef Rignes Island in Canada, and is approximately

Ionospheric and gamma-ray Observations Satellite		Ref. : SCA-DD-02 Edition: 1 Date: 05-09-2016 Revision : 2 Date :
		Page : 15

700 miles from the geographic North Pole. The magnetic dipole axis is currently at an inclination angle of 11.5° with the equatorial plane. The axis is drifting westward at about 0.2°/year, and the strength is decreasing by 0.05% per year.

The magnetic field is weakest at the magnetic equator, or the plane perpendicular to the magnetic dipole.

The magnetic field of the Earth is not a constant over time. The magnetic field can be an advantage to the spacecraft. The magnetic field has the ability to change the orientation of a body. While this may act to complicate the dynamics of the spacecraft, when used correctly the magnetic field may be used for satellite control.

4.3.1 THEORY AND MATHEMATICAL MODELS

The Earth’s magnetic field can be analytically calculated at any point around the Earth, and it is possible to describe this magnetic field in one particular coordinate system. Two mathematical models are used in my work. One is IGRF magnetic model and tilted dipole model.

In satellites application, this magnetic field is often expressed in geocentric inertial Coordinate system.

$$\begin{aligned}
 B_x &= (B_r \cos \delta + B_\theta \sin \delta) \cos \alpha - B_\phi \sin \alpha \\
 B_y &= (B_r \cos \delta + B_\theta \sin \delta) \sin \delta + B_\phi \cos \alpha \\
 B_z &= (B_r \sin \delta - B_\theta \cos \delta)
 \end{aligned}
 \tag{9}$$

Where r, θ and ϕ are tangential coordinate system parameters(co-elevation and longitude). The magnetic field is a function of both the right ascension α and declination δ .

International Geomagnetic Reference Field(IGRF)

The set of Gaussian coefficients for use in the analytical models describing the Earth’s magnetic field are called the International Geomagnetic Reference Field (IGRF). Every five years, a group from the International Association of Geomagnetism and Aeronomy examines the measured geomagnetic field representations and produces the coefficients for that particular year in nT. A specific field model is referred to by including the year of epoch in the name, i.e. IGRF2000. Two sets of coefficients are linearly interpolated to calculate the magnetic field for years in between IGRF epochs. The most recent publication is the IGRF2015.

The Earth’s magnetic field, B , can be described as the negative gradient of a scalar potential function, V ,

$$B = -\nabla V$$

V is described by a series of spherical harmonics,

$$V(r, \theta, \phi) = R_\oplus \sum_{n=1}^k \left(\frac{R_\oplus}{r} \right)^{n+1} \sum_{m=0}^n (g_n^m \cos m\phi + h_n^m \sin m\phi) P_n^m(\theta)
 \tag{10}$$

where R_\oplus is the equatorial radius of the earth, g_n^m and h_n^m are the Gaussian coefficients, $r, \theta, and \phi$ are the Geo-centric distance, co-elevation and East longitude from Greenwich P_n^m is the associated Legendre function of degree n and order m .



The Matlab code for Magnetic model using IGRF 2015 is taken from[18].

Tilted dipole model

For analytic purposes, a tilted dipole model of the geomagnetic field can be obtained by calculating the spherical harmonic model to the first degree ($n = 1$) and all orders ($m = 0, 1$). The scalar potential, V , becomes

$$\begin{aligned} V(r, \theta, \phi) &= \frac{R_{\oplus}^3}{r^2} \left[g_1^0 P_1^0(\theta) + (g_1^1 \cos \phi + h_1^1 \sin \phi) P_1^1(\theta) \right] \\ &= \frac{1}{r^2} (g_1^0 R_{\oplus}^3 \cos \theta + g_1^1 R_{\oplus}^3 \cos \phi \sin \theta + h_1^1 R_{\oplus}^3 \sin \phi \sin \theta) \end{aligned} \quad \dots(11)$$

The first degree Gaussian coefficients in the year 2015 are

$$\begin{aligned} g_1^0 &= -29442 \\ g_1^1 &= -1501 \\ h_1^1 &= 4797 \end{aligned} \quad \dots(12)$$

The total dipole strength is given by

$$R_{\oplus}^3 H_0 = R_{\oplus}^3 (g_1^{0^2} + g_1^{1^2} + h_1^{1^2})^{1/2} \quad \dots(13)$$

The co-elevation and East longitude are given by,

$$\theta'_m = \tan^{-1} \left(\frac{h_1^1}{g_1^1} \right) \text{ and } \phi'_m = \tan^{-1} \left(\frac{h_1^1}{g_1^1} \right) \quad \dots(14)$$

The magnetic field in the local tangential coordinates is given by

$$\begin{aligned} B_r &= 2 \left(\frac{R_{\oplus}}{r} \right)^3 \left[g_1^0 \cos \theta + (g_1^1 \cos \phi + h_1^1 \sin \phi) \sin \theta \right] \\ B_{\theta} &= \left(\frac{R_{\oplus}}{r} \right)^3 \left[g_1^0 \sin \theta - (g_1^1 \cos \phi + h_1^1 \sin \phi) \cos \theta \right] \\ B_{\phi} &= \left(\frac{R_{\oplus}}{r} \right)^3 \left[g_1^1 \sin \phi - h_1^1 \cos \phi \right] \end{aligned} \quad \dots(15)$$

This calculation leads to magnetic field components in geocentric inertial components of

$$\begin{aligned} B_x &= \frac{R_{\oplus}^3 H_0}{R^3} \left[3(\hat{\mathbf{m}} \cdot \hat{\mathbf{R}}) R_x - \sin \theta'_m \cos \alpha_m \right] \\ B_y &= \frac{R_{\oplus}^3 H_0}{R^3} \left[3(\hat{\mathbf{m}} \cdot \hat{\mathbf{R}}) R_y - \sin \theta'_m \sin \alpha_m \right] \\ B_z &= \frac{R_{\oplus}^3 H_0}{R^3} \left[3(\hat{\mathbf{m}} \cdot \hat{\mathbf{R}}) R_z - \cos \theta'_m \right] \end{aligned} \quad \dots(16)$$



Where $\hat{m} = \begin{bmatrix} \sin \theta'_m \cos \alpha'_m \\ \sin \theta'_m \sin \alpha'_m \\ \cos \theta'_m \end{bmatrix}$ and \hat{R} are the designate unit vectors.

$$\alpha'_m = \theta_{g0} + \omega_{\oplus} t + \phi'_m \quad \dots(17)$$

θ_{g0} is the Greenwich sidereal time at some reference time, ω_{\oplus} is the average rotation rate of the earth equal to 7.2921152×10^{-5} rad/sec, t is the time since reference.

4.3.2 COMPARISON OF MAGNETIC FIELDS -IGRF & TILTED DIPOLE MODEL

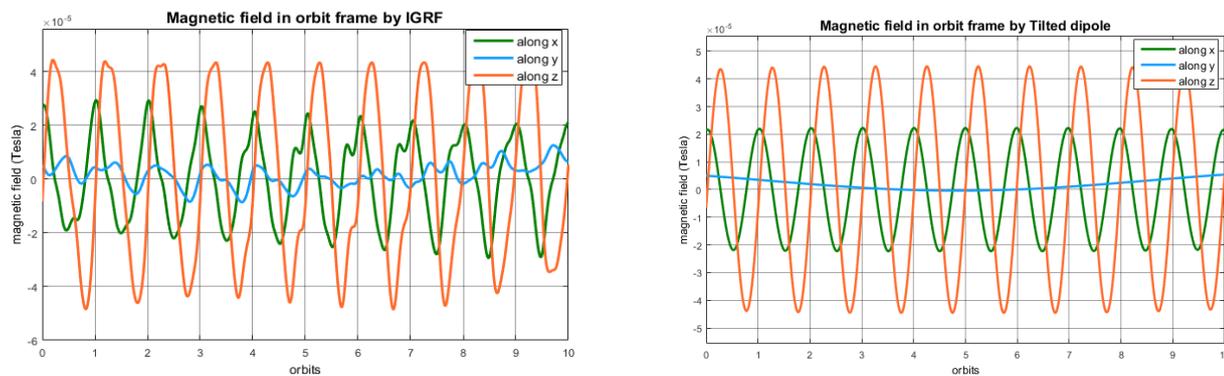


Figure 8: Comparison of Magnetic field-IGRF & Tilted dipole

- Requires complete IGRF coefficients
 - g –coefficients(195 x 1)
 - h –coefficients (195 x 1)
 - Coefficient order index (195 x 1)
- Requires only 1st order gaussian IGRF coefficients (3 x 1)

It is obvious that magnetic field computed using complete IGRF coefficients is more realistic to true magnetic field of the earth. In my simulation, both the magnetic models are considered. A complete IGRF model is used for Magnetometer & Magneto-torquer models in the simulation, while the tilted magnetic model is used in Attitude estimation model. This is because the algorithm for tilted dipole model is less complex and easy to embedded in the flight software.



4.4 SUN VECTOR FROM EPHEMERIS DATA

At any point of time, the direction vector between the satellite and Sun can be calculated by knowing the position of the satellite (in inertial coordinates) and the time (in UTC).

The algorithm is taken from 1993 Astronomical Almanac, p. C24. (see Appendix[5]). The Matlab code for the sun model is given in Appendix 3.g.

The below figure shows the sun direction calculated for 1 orbit of the satellite.

This theoretically computed sun vector is compared with the measured sun vector from the sun sensor and is used in the Attitude Determination System.

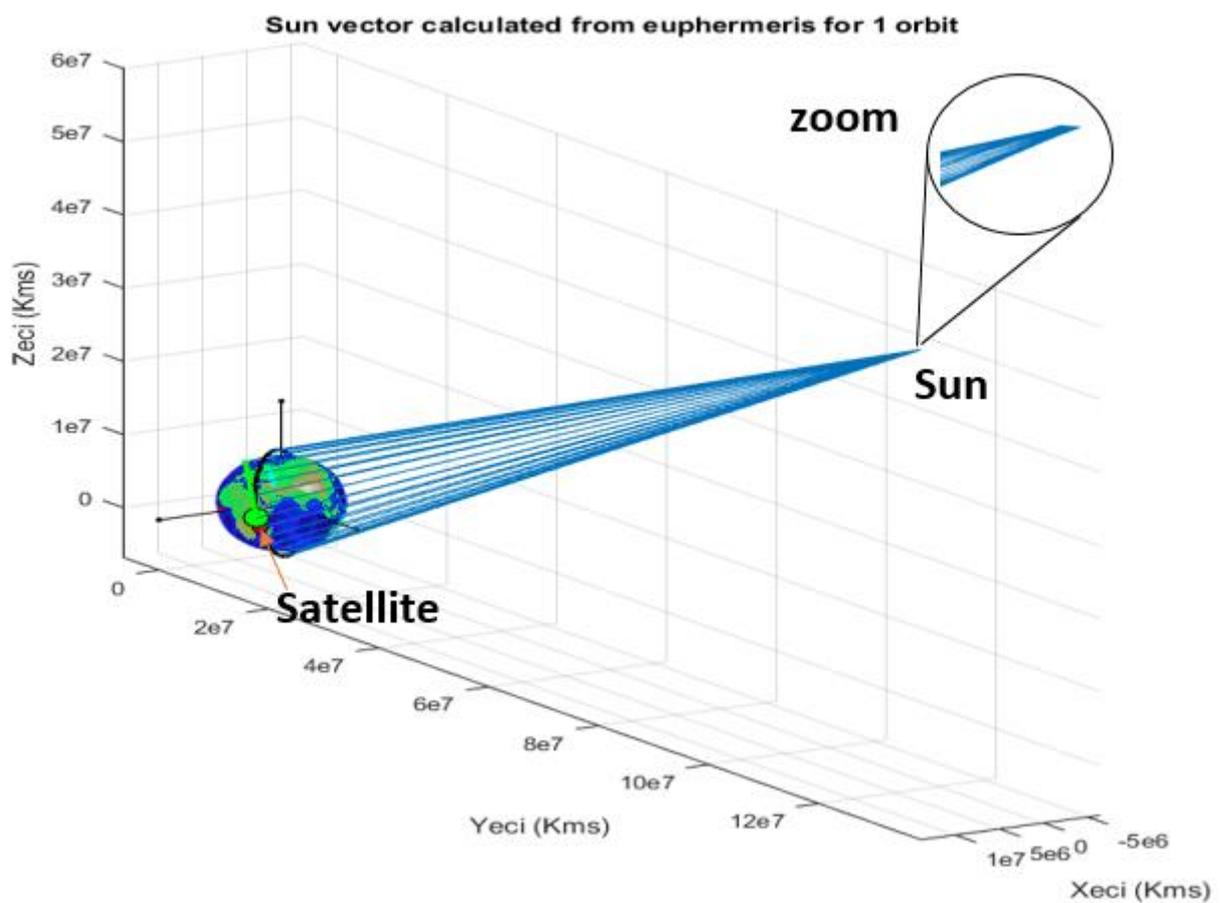


Figure 9: Sun vector calculated from ephermeris for 1 orbit



5. ATTITUDE DYNAMICS AND KINEMATICS

5.1 Dynamics

According to the law of conservation of angular momentum, for any rigid body, the rate of change of angular momentum is equal to the total external torques applied.

$$\frac{dL}{dt} = \tau_{ext}$$

Where $L = I\omega$ is the Angular momentum, $I = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix}$ is the Moment of Inertia of the satellite(symmetrical in our case) & ω is angular velocity.

$$\begin{aligned} I\dot{\omega} &= \omega^\times I\omega + \tau_{ext} \\ \dot{\omega} &= -I^{-1}\omega^\times I\omega + I^{-1}\tau_c + I^{-1}\tau_{dist} \end{aligned} \quad \dots(18)$$

So for a rigid satellite, if $\omega_{R_{sat}/R_g}^{R_{sat}}$ is the angular velocity of the satellite with respect to inertial frame as seen in satellite frame, then

$$\dot{\omega}_{R_{sat}/R_g}^{R_{sat}} = -I^{-1}\omega_{R_{sat}/R_g}^{R_{sat}\times} I\omega_{R_{sat}/R_g}^{R_{sat}} + I^{-1}\tau_c^{R_{sat}} + I^{-1}\tau_{dist}^{R_{sat}} \quad \dots(19)$$

Where $\tau_c^{R_{sat}}$ & $\tau_{dist}^{R_{sat}}$ are the control and disturbance torques observed in satellite frame R_{sat} .

Recalling equation(6) , $\omega_{R_{sat}/R_g}^{R_{sat}} = \omega_{R_{sat}/R_{loc}}^{R_{sat}} + \omega_{R_{loc}/R_g}^{R_{sat}}$, and substituting in equation(19),

$$\dot{\omega}_{R_{sat}/R_{loc}}^{R_{sat}} = -I^{-1}(\omega_{R_{sat}/R_{loc}}^{R_{sat}} + \omega_{R_{loc}/R_g}^{R_{sat}})^\times I(\omega_{R_{sat}/R_{loc}}^{R_{sat}} + \omega_{R_{loc}/R_g}^{R_{sat}}) + I^{-1}\tau_c^{R_{sat}} + I^{-1}\tau_{dist}^{R_{sat}} - \dot{\omega}_{R_{loc}/R_g}^{R_{sat}} \quad \dots(20)$$

This equation describes the Non-linear dynamics in local orbital frame. The control torques and disturbance torques act in the satellite body frame. Since the strength of some disturbances depends on the orientation relative to the local orbit frame, we implement the attitude controller wrt the orbital frame. Also the local orbit frame is considered as equilibrium for many satellite.

The term $\omega_{R_{loc}/R_g}^{R_{sat}}$ can be expressed from [22]as,

$$\begin{aligned} \omega_{R_{loc}/R_g}^{R_{sat}} &= R_{sat/loc} \omega_{R_{loc}/R_g}^{R_{loc}} = R_{sat/loc} \begin{bmatrix} 0 \\ -\omega_c \\ 0 \end{bmatrix} \\ \dot{\omega}_{R_{loc}/R_g}^{R_{sat}} &= -\omega_{R_{sat}/R_{loc}}^{R_{sat}} R_{sat/loc} \omega_{R_{loc}/R_g}^{R_{loc}} = -\omega_{R_{sat}/R_{loc}}^{R_{sat}} R_{sat/loc} \begin{bmatrix} 0 \\ -\omega_c \\ 0 \end{bmatrix} \end{aligned} \quad \dots(21)$$



5.2 Kinematics

The kinematics of the satellite, is described using the attitude quaternion $q = [q_1 \ q_2 \ q_3 \ q_4]^T$ to represent a rotation. The orientation of the satellite is obtained rotating from the local orbital frame R_{loc} to the satellite frame R_{sat} by the directional cosine (rotation) matrix. The rotation matrix is parameterized by the quaternion, $R_{sat/loc} \Leftrightarrow q_{sat/loc}$. The kinematics of the satellite is then,

$$\dot{q} = \frac{1}{2}\Omega(\omega)q = \frac{1}{2}\Xi(q)\omega$$

Where $\Omega(\omega) = \begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix}$ and $\Xi(q) = \begin{bmatrix} q_4 & -q_3 & q_2 & q_1 \\ q_3 & q_4 & -q_1 & q_2 \\ -q_2 & q_1 & q_4 & q_3 \\ -q_1 & -q_2 & -q_3 & q_4 \end{bmatrix}$

$$\frac{d}{dt} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} q_4 & -q_3 & q_2 & q_1 \\ q_3 & q_4 & -q_1 & q_2 \\ -q_2 & q_1 & q_4 & q_3 \\ -q_1 & -q_2 & -q_3 & q_4 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ 0 \end{bmatrix} \quad \dots(22)$$

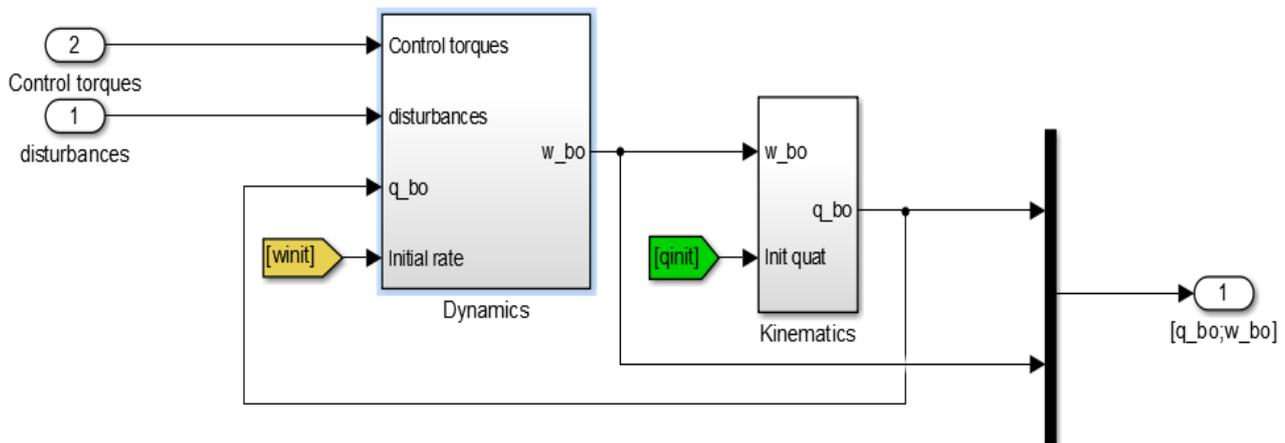


Figure 10: Satellite dynamics & kinematics in Simulink

Please note that for simplicity, notations have been changed in the Simulink model $R_{sat} \rightarrow b$, $R_{loc} \rightarrow o$, $R_g \rightarrow i$ and $\omega_{R_{sat}/R_{loc}} \rightarrow w_bo$.

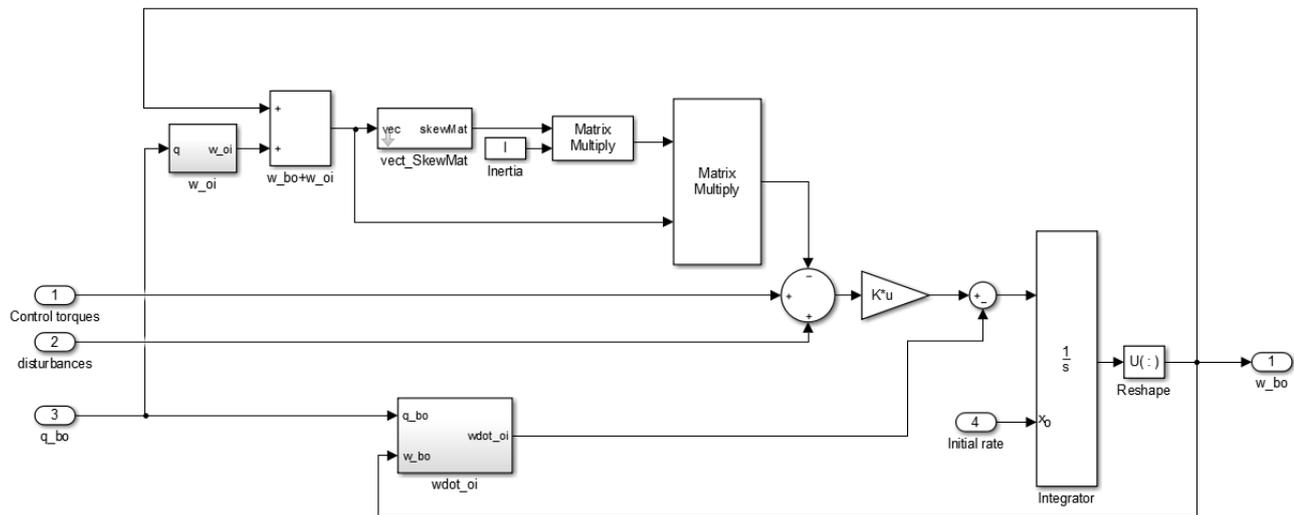


Figure 12: Inside the dynamics block

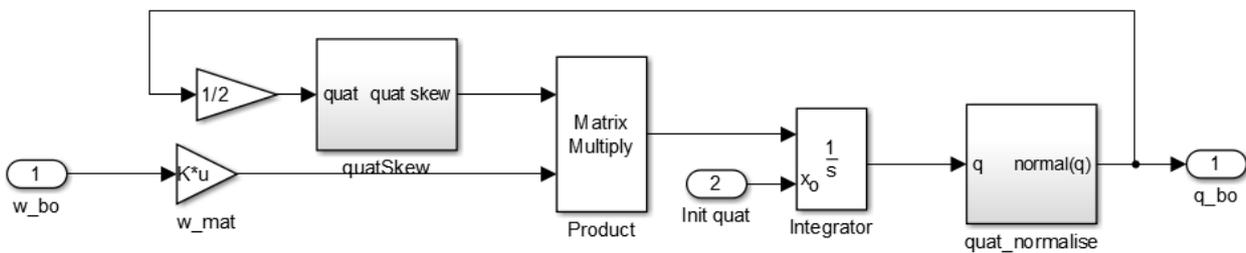


Figure 11: Inside the Kinematics block

5.3 Environmental Disturbances Torques

Spacecraft in orbit encounter small disturbance torques from various environmental sources. These torques are either secular, which accumulate over time, or cyclic, which vary sinusoidally over an orbit. Both types are discussed by Hughes[23].

Different environmental torques are more prevalent at different altitudes. In low Earth orbits (LEO) the largest environmental torques are gravity-gradient, magnetic, and aerodynamic. In this analysis, we considered these three torques for the study.

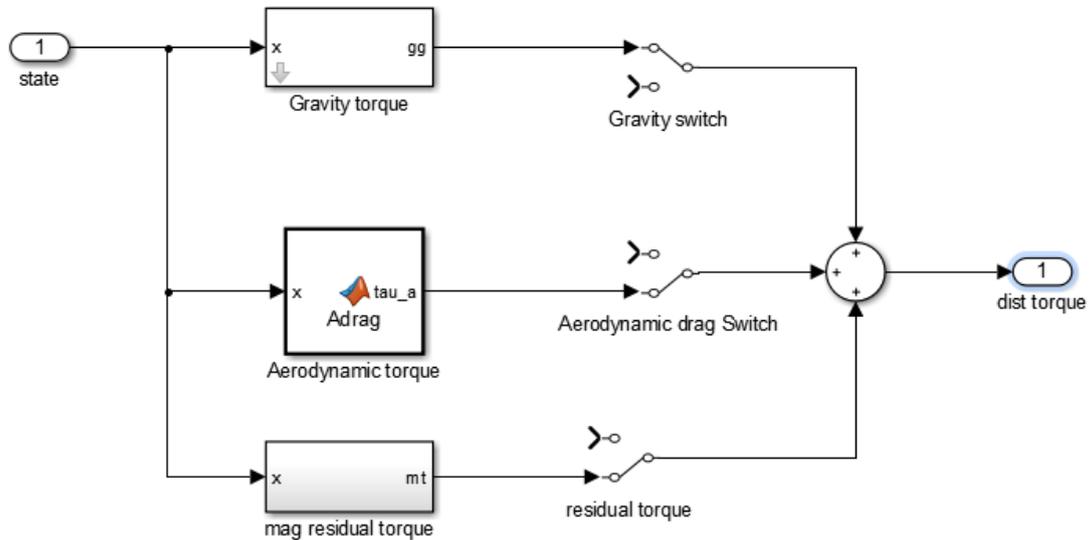


Figure 13:Disturbance models in Simulink

5.3.1 GRAVITY-GRADIENT TORQUE

A gravity-gradient torque exists because of the variation in the gravitational field. Since the gravitational force field varies with the inverse square of the distance from the Earth, there is a greater force on the portion of the spacecraft closer to the Earth.

The gravity-gradient torque is constant for Earth-oriented spacecraft and cyclic for initially oriented vehicles. It is mainly influenced by the moment of inertia of the spacecraft and the altitude of the orbit. From Newton's Universal Gravitational Law,

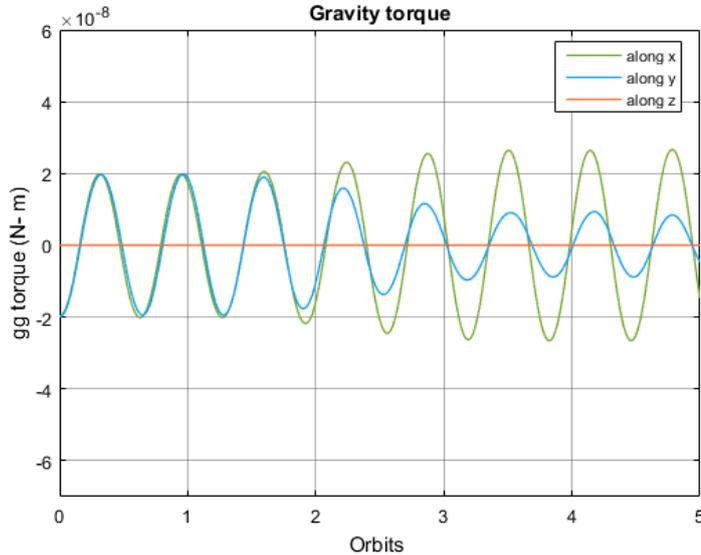
$$d\vec{f}_g = -\frac{GM}{r^2} dm \hat{e}_r$$

where G is the universal gravitational constant, M is the mass of the spherical primary, dm is the mass of a mass element of the body in orbit, r is the radial distance from the mass center of the primary to the mass element, and \hat{e}_r is a unit vector from the mass center of the primary to the mass element. A simplified derivation for gravity-gradient torque, τ_{gg} , from[9] is presented here,

$$\tau_{gg} = -3\omega_c^2 \hat{o}_3^\times I \cdot \hat{o}_3 \quad \dots(23)$$

where $\omega_c = \sqrt{\frac{GM}{r^2}}$ and \hat{o}_3 is the third column of the rotation matrix $R_{sat/loc}$

If only gravity torque is assumed in the orbit with initial conditions (10, 10, 40 deg), then it varies as follows in the orbit,



Maximum gravity torque observed,

$$\tau_{gg}^{\max} = \begin{bmatrix} 0.2668 \\ 0.1981 \\ 0 \end{bmatrix} 10^{-7} \text{ N-m}$$

Figure 14: Gravity gradient over the orbit

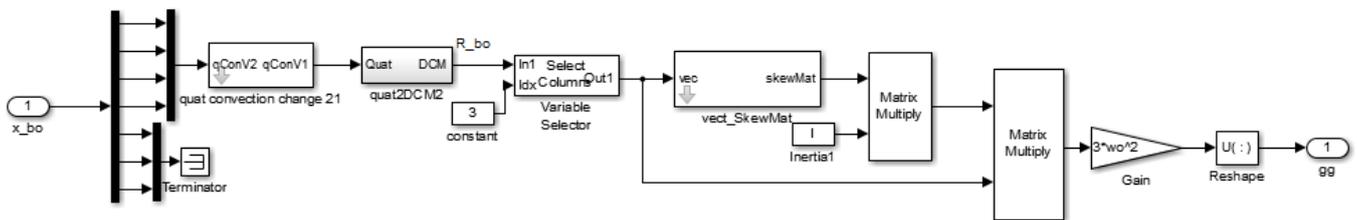


Figure 15: Gravity gradient disturbance in simulink

5.3.2 AERODYNAMIC TORQUE

This is due to the drag with the atmosphere and primarily affects satellites in LEO. The maximum possible torque is given by:

$$\tau_a = \frac{1}{2} \rho v^2 C_D A L$$

Where, C_D is the Coefficient of drag, A is the cross sectional area, L is distance b/w center of pressure and Gravity. One of the CubeSat design requirements is that the spacecraft CG be within 2 cm of the geometric center of the spacecraft.

It is important to remember that the aerodynamic drag sets the absolute limit on the lifecycle for any satellite without heave/altitude compensation. The orbit speed will eventually decrease so much, the satellite will fall towards Earth and burn up in the atmosphere. The IGOSAT is aimed for 1-year mission at 650 Km and return back to earth after a year.

Hughes [10] suggests an expression for the aerodynamic drag,



$$\tau_a = \rho_a v_{orb} \left[v_{orb} A_{drag} c_p \hat{V}_{orb} - (I + \hat{V}_{orb} J) \omega_{R_{sat}/R_{loc}}^{R_{sat}} \right] \hat{V}_{orb} \quad \dots(24)$$

Where, ρ_a - density of the atmosphere in kg/m^3

v_{orb} - magnitude of orbit velocity vector (constant)

A_{drag} - projected 2-dimensional surface area facing the velocity direction

c_p - center of pressure

\hat{V}_{orb} - unit velocity direction

I - moment of inertia

J - new moment of inertia for drag which is slightly off center from the normal I .

The Matlab code for aerodynamic drag is given in *Appendix- 3.c*.

5.3.3 MAGNETIC RESIDUAL TORQUES

This is a torque due to the magnetic field being created by the non-magnetorquer components of the satellite and includes residual magnetic field in the structure and that being generated by the operating electronic components. Since the governing equation is the same for the torque control itself, its magnitude can be related to torque ability as the ratio of the residual magnetic field to the induced dipole of the magnetorquers. The residual field was not measured, but it is reasonable to assume that it is at most one fifth of the torquer field according to [11].

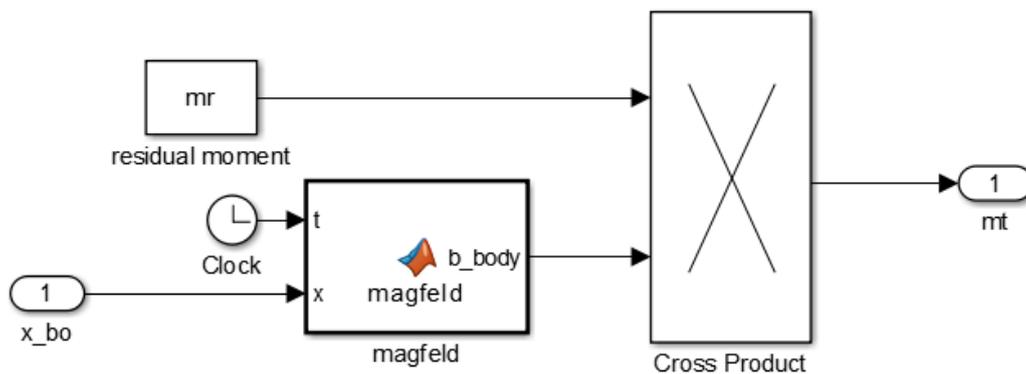


Figure 16: Magnetic residual torques in Simulink

In the simulations, we consider a residual moment of $m_r = [0.001, -0.001, 0.005]^T \text{ A}\cdot\text{m}^2$. Which is a very large, indeed, favors the simulation confidence. Residual torques are computed by the cross product between the residual magnetic moment and the Earth's magnetic field strength \mathbf{B} in satellite body frame coordinates.



Comparison of Disturbance torques

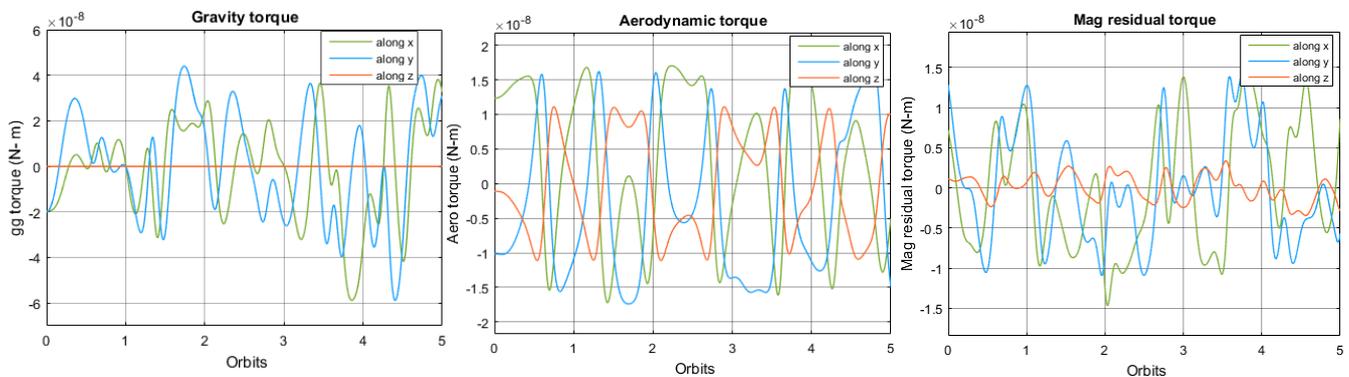


Figure 17: Comparison of disturbance torques

Max Disturbance torque	Along X (N-m)	Along Y(N-m)	Along Z (N-m)
Gravity gradient torque	0.38×10^{-7}	0.43×10^{-7}	0 N-m
Aerodynamic torque	0.172×10^{-7}	0.174×10^{-7}	0.11×10^{-7}
Magnetic residual torque	0.156×10^{-7}	0.13×10^{-7}	0.03×10^{-7}
Total Disturbance torque	0.708×10^{-7}	0.734×10^{-7}	0.14×10^{-7}

6. ATTITUDE CONTROL

Attitude Control of a satellite is classified as **1. Active control** & **2. Passive control**.

A passive controller for a CubeSat can be set up quite easy using strong rare-earth magnets. Most suitable way of stabilizing the CubeSat's is *Nadir pointing*, in which satellite longitudinal axis is pointed towards the earth center. It is also assumed that stable condition is when the satellite axes is coincided with the local orbital axes.

Before we go into the Control theory, let us first see the dynamics of Magnetorquers.

6.1 Actuator Dynamics

The magnetorquers are the only actuators available for attitude control of the IGOSat. A total of 2 ferromagnetic torque rods and 1 air-core torquer from *Cubespace* are used.

The two torque rods are aligned with X & Y axis of satellite, while the air-core torquer is along the Z axis.

When electrical current is supplied through the windings of the coils, a magnetic dipole field will be created. The magnetic dipole can be described as a vector, and the vector will be perpendicular to the coil, pointing in the direction of the extended thumb of one's hand according to the right-hand rule.

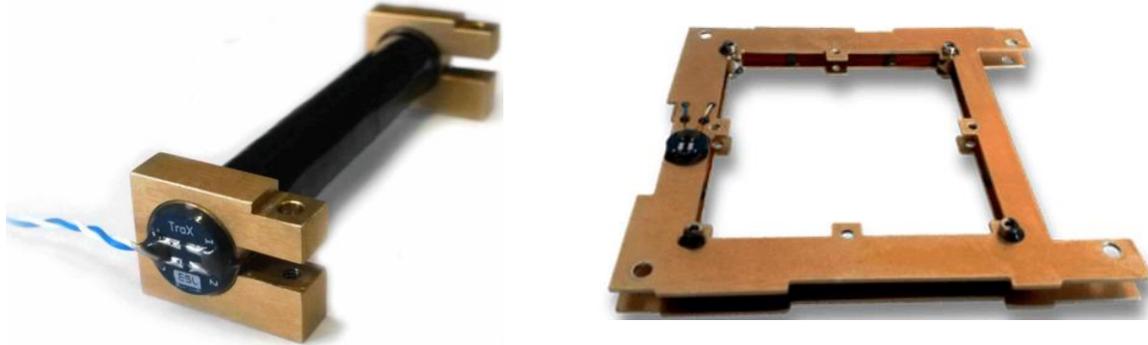


Figure 18: Magnetic torque rod & air-core torquer[13]

Characteristics of torque-rods & air-core torquer:

	Cube Torque rod	Cube Coil
Manufacturer	CubeSpace	CubeSpace
Dimensions	60 (L) x 10 (d) mm (rod only) 18 x 14 x 62 mm(with mounting)	90 x 96 x 6 mm
Mass	28 g	46 g
Operating Temperature	-35°C to +75°C	N/A
Maximum moment	±0.24 Am ² (@ 25°C , 2.5 V)	±0.13 Am ² (@ 25°C, 5 V)
Resistance	30 ohm	82 - 84 Ohm
Voltage Supply	N/A	-
Residual moment:	< 0.48 mAm ²	0 Am ²

When this field reacts with the local geomagnetic field, a magnetic moment is generated. In a very intuitive way, the magnetic field generated by the magnetorquers will try to align itself with the local geomagnetic field surrounding the satellite at that particular place.

$$m^{sat} = m_x^{sat} + m_y^{sat} + m_z^{sat}$$

$$\tau_m^{sat} = m^{sat} \times B^{sat}$$

Where $m^{sat} = \frac{NAV}{R}$ is the magnetic moment, B^{sat} = Geomagnetic field in satellite frame axes.

N=no. of windings, A= c/s of torque rod, R= Resistance in coil, V=voltage applied.

In our design, m_x^{sat} & m_y^{sat} are the magnetic moments generated by torque rods on X & Y axis of satellite. While m_z^{sat} is magnetic moment produced by cubecoil on Z-axis.

The earth's geomagnetic field allows only 2-axis control of the satellite at an instant. So by the definition of controllability [21], our system is uncontrollable. The 3-axis stability/control is achieved over time. Another drawback of magnetic control is, the torque τ_m , produced by the torque rod is perpendicular to the m & B . But the required torque (τ_d) is not often perpendicular to the m & B . So one has to do some modifications to the control algorithms as per[12].

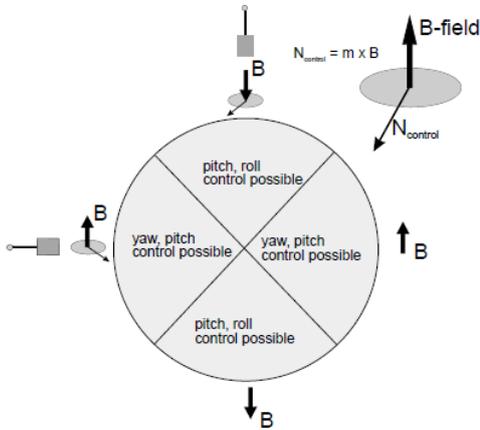


Figure 20: Satellite controllability with magneto-torquers[25]

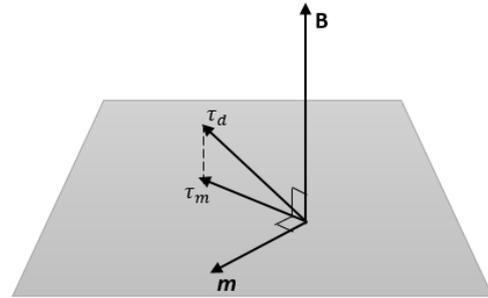


Figure 19: Magnetic moment, field and Torque relation

From Figure 19,

$$m = \tau \times B$$

The following expression is used for the desired moment as derived in [12],

$$m_d^{sat} = \frac{1}{|B^{sat}|^2} (B^{sat} \times \tau_d^{sat}) \quad \dots(25)$$

The stability analysis on this formulation is given in [21] and is presented in *Appendix 2*. With the torque rod specifications[19], If a voltage of 2.5 V is supplied to the torque rod, it produces a maximum current of $i_{r_max} = 0.24 \text{ Am}^2$. Similarly for Aircoil, a 5V supply produces a maximum current of $i_{c_max} = 0.13 \text{ Am}^2$.

According to a conservative estimate made by Fauske (2002), a magnetic field of 0.02 Am^2 is more than enough to control the satellite.

Coming to the Attitude Control, lets recall our operating modes

Mode	Controller(control law)
De-tumbling	Desired torque control
Nominal	PID control

6.2 De-tumbling Control

During launching the satellite into the orbit, there may be rotation rates induced by the launcher on the satellite. Hence these should be damped to stabilize the spacecraft. Detumbling consists of damping the angular velocity of a spacecraft to zero. Many detumbling controller algorithms are available from the literature. we analyze the performance of the *b-dot*, *B-bang bang* & *desired torque control* laws and compare the power & time consumed for tumbling the satellite.

The ***b-dot control*** law takes advantage of the fact that the derivative of the magnetic field vector \dot{b} is both perpendicular to b and proportional to ω , hence the commanded magnetic dipole can be expressed as:

$$m_c = -k \dot{b} \quad \dots(26)$$



where k is a positive gain and chosen according to [14]. In order to speed up the spacecraft spin rate decay, it is also possible to employ a **B bang-bang** solution of the form:

$$m_c = -m_{\max} \operatorname{sgn}(\dot{b}) \quad \dots(27)$$

where m_{\max} is the maximum torquer dipole. Since \dot{b} can be estimated from onboard measurements of the magnetic field, no information about the angular velocity of the satellite is required by eqs. (26) and (27). On the other hand, attitude information is typically available from onboard sensors, and the following alternative control law (**desired torque control**) can be used:

$$m_c = \frac{-k_\omega}{\|b\|^2} (b \times \omega) \quad \dots(28)$$

where k_ω is a positive gain and $(b \times \omega) \approx \dot{b} \propto \omega$.

6.3 PID Control

The following control law is used for the Nominal mode of IGOSat,

$$\tau_d = -K_p \varepsilon_e - K_i \int_0^t \varepsilon_e dt - K_d \omega_e \quad \dots(29)$$

Where K_p, K_i and K_d are the PID gains chosen proportional to the Inertia I . So that the control torque along each axis is proportional to satellite mass distribution.

$$K_p = k_p I \quad K_i = k_i I \quad \text{and} \quad K_d = k_d I \quad \dots(30)$$

Therefore k_p, k_i and k_d are PID tuning parameters. The initial stable gains are computed from the Matlab PID tuner with chosen settling time around 2000 seconds & rise time of 1500 seconds. There by gains are manually changed for the optimal gain calculation (section 9.4).

q_e and ω_e are the quaternion & angular velocity error between the desired to the current state in local orbital frame.

$$q_e = \begin{bmatrix} q_{d4} & q_{d3} & -q_{d2} & -q_{d1} \\ -q_{d3} & q_{d4} & q_{d1} & -q_{d2} \\ q_{d2} & -q_{d1} & q_{d4} & -q_{d3} \\ q_{d1} & q_{d2} & q_{d3} & q_{d4} \end{bmatrix} \begin{bmatrix} q_{s1} \\ q_{s2} \\ q_{s3} \\ q_{s4} \end{bmatrix} \quad \text{and} \quad \omega_e = \omega_s - \omega_d \quad \dots(31)$$

Where q_d and q_s are the desired and current attitude of the quaternion $q_{R_{\text{sat}}/R_{\text{loc}}}$.

ω_d and ω_s are the desired and current angular rates of $\omega_{R_{\text{sat}}/R_{\text{loc}}}^{\text{Rsat}}$.

Therefore, magnetic moment for the actuator can be computed as from eq(25),

$$m_d = \frac{1}{|B^{\text{sat}}|^2} \left(B^{\text{sat}} \times \left(-K_p \varepsilon_e - K_i \int_0^t \varepsilon_e dt - K_d \omega \right) \right) \quad \dots(32)$$

Where B^{sat} is the local magnetic field in satellite frame.



6.4 Combined De-tumble & PID Control

The De-tumbling and PID controllers are embedded into one control block. According to the IGOSat requirements, the transition between De-tumbling and PID is automatic and is switched to PID when the angular rates reach less than 0.5 °/ sec on each axis of the satellite.

We choose the *desired torque* control law for the De-tumbling mode.

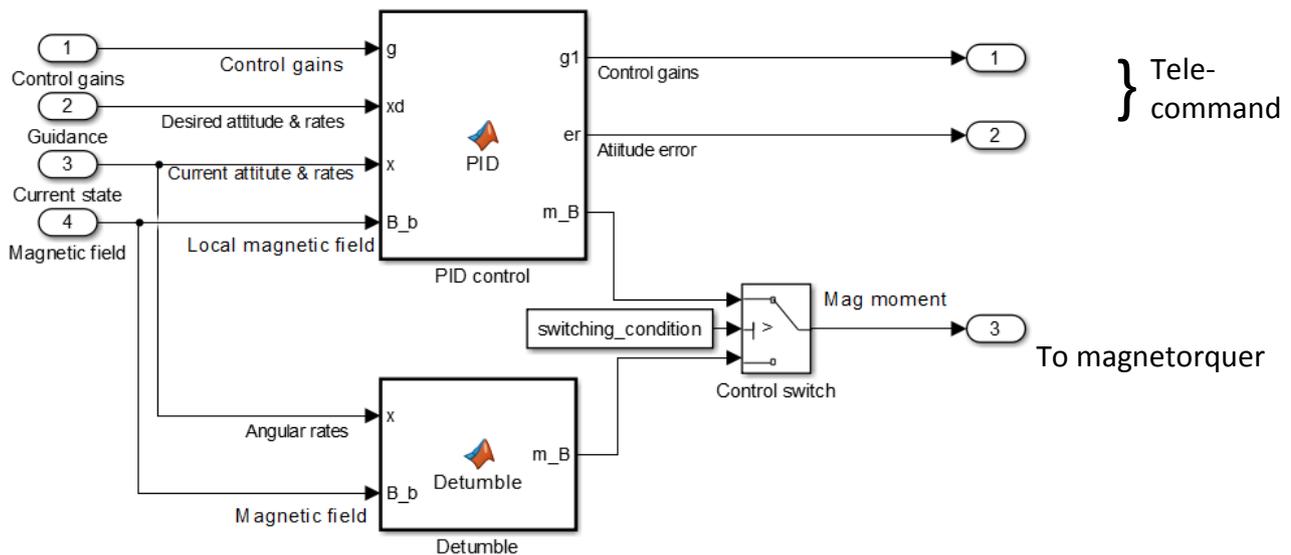


Figure 21: Combined de-tumbling & PID control in Simulink

Inputs

		Size	Type
Control gains	$[k_p \ k_i \ k_d]^T$	3 x 1	double
Guidance (R_c)	$[q_c \ \dot{\phi} \ \dot{\theta} \ \dot{\psi}]$	7 x 1	double
Current attitude & rates	$[q \ \omega]_{R_{sat}/R_{loc}}^T$	7 x 1	double
Magnetic field(in tesla)		3 x 1	double
Time (in UTC) [YYYY MM DD HH mm ss]		1 x 6	double

Outputs

		Size	Type
Voltage to torquer (V)		3 x 1	double
Attitude error quaternion	b/w desired & current	4 x 1	double

7. ATTITUDE DETERMINATION

The orientation of a spacecraft can be determined by describing the rotation between a spacecraft-fixed reference frame and a known reference frame. This description is accomplished by finding rotations between measured attitude vectors and known quantities. For example, a Sun sensor determines the vector, s^b from the spacecraft to the Sun in the satellite body frame R_{sat} .

Since the vector in the inertial frame, s^i , can be calculated from ephemeris data, the following relation is useful

$$s^b = R_{R_{sat}/R_g} s^i \quad \dots(33)$$

Where R_{R_{sat}/R_g} is rotation matrix from Inertial frame to satellite frame.

A similar relation is valid for magnetic field vector m^b from magnetometer and earth vector e^b measured by earth sensor.

Table 1: Summary of Attitude Sensors in IGOSat

Sensor/Actuator	Model	Manufacturer	Number needed	Power Requirement	Bandwidth	Input	Output
Sun sensor	Photocells VT90N2	EXCELITAS	5	5mW	-	Solar flux	Sun vector
Magnetometer	Honeywell HMC 2003	Honeywell	1	42 mW	1 kHz	Magnetic value	Magnetic vector
Earth Sensor	MLX90615	Melexis	1	26.7 mW	10 - 100 kHz	Temperature	Pitch & Roll(earth direction)
3 axis - Gyros	A3G4250D	StMicroelectronics	1	21 mW	105/208/420/840 Hz	-	Angular rate

There are many ways to estimate attitude using these sensors. I present you some possible ways from the literature.

1) Using UKF[16]

- This is the considered to be most reliable attitude estimation approach as the complete non-linear system is considered & the algorithm is complicated to program. (requires to program many supporting functions to the flight software).
- Satellite non-linear dynamics are coded in the state transition (requires to measure the control torques, disturbance torques...)
- Biases of the gyros & magnetometer are considered in the state vector (however initial biases has to be computed from the tests).
- Need to have magnetic field model, sun model & earth model (requires position & time).
- Prediction & update is based on the previous estimate.
- Has higher performance in the case of magnetometer is turned off and eclipse scenarios.

2) Using EKF

- Requirements for EKF are similar to UKF but simpler in implementation.



- Can be more simplified by taking less states in state vector (only quaternion & bias).
- 3) Simple Linear KF approach from the CONSAT[15] attitude estimate
- Here state vector constitutes of quaternion and gyro bias.
 - The magnetometer bias is assumed constant.
 - Attitude quaternions computed from sensors (by TRIAD) are used in measurement update step. (requires sun, earth & magnetometer to calculate attitude separately).
 - Linear Kalman filter is used to estimate the attitude.

In our method, we combine (2) & (3) approaches. we use the gyros in the state vector prediction and sun, earth & magnetometer are in the measurement step and there by *Extended Kalman filter* is used for computing the Kalman gain. This kind of implementation has advantages in eclipses & magnetometer turned off scenario.

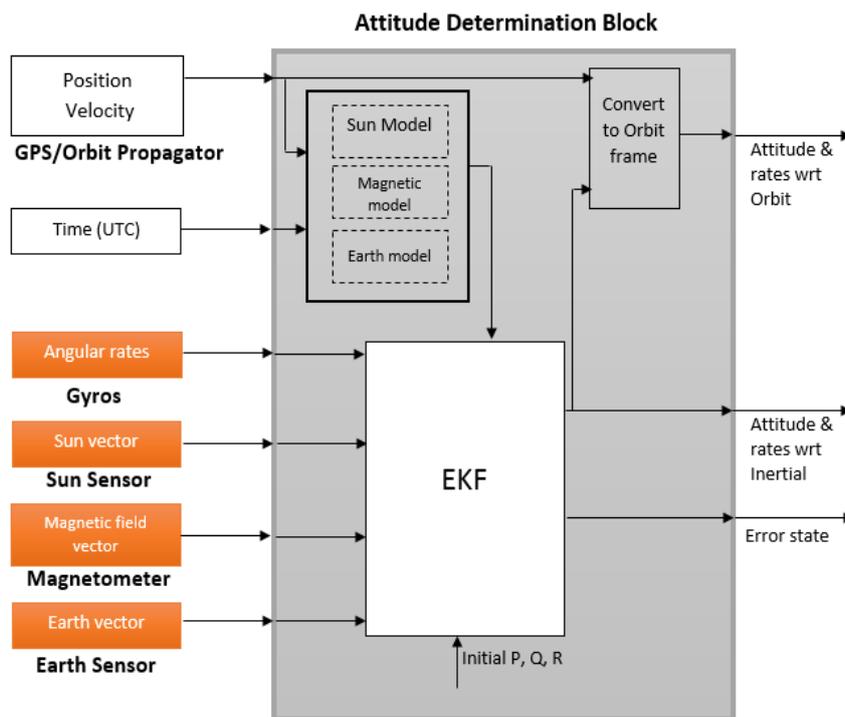


Figure 22: Attitude Determination System-process

Sensor	Measurement vector	Source of reference vector
Sun sensor	Unit sun vector	Solar Ephemeris
Magnetometer	Magnetic field vector	Tilted dipole model
Earth sensor	Unit earth/nadir vector	S/C Position

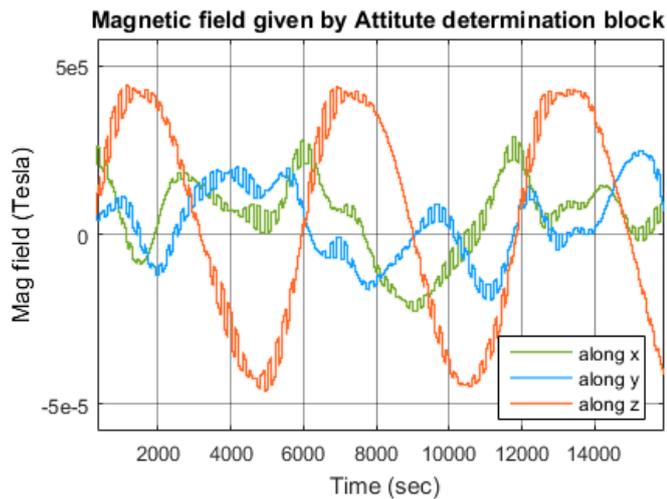


The gyros sense the angular rates wrt to Inertial frame. So the initial attitude estimate is wrt Inertial frame and thereby expressed in local orbit frame with known satellite position & velocity.

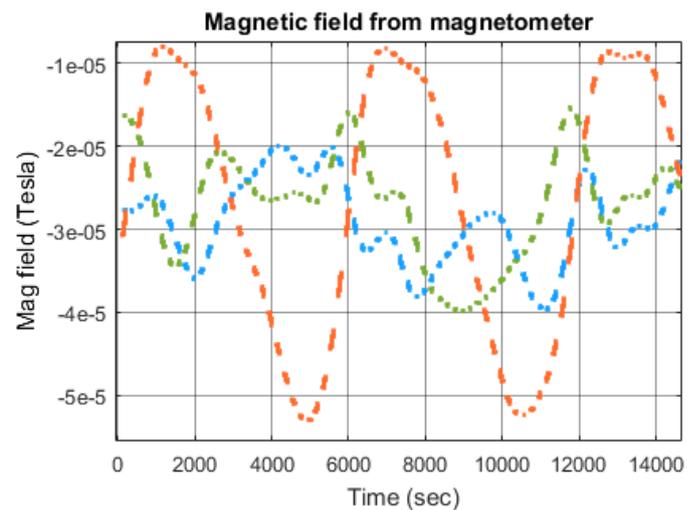
Magnetic field from Attitude determination system

It is known that the Attitude controller requires the local magnetic field. But due to the limitation of magnetometer design, the magnetometer couldn't be placed far away to the magneto-torquer. This effects the magnetometer in measuring local magnetic field while the magneto-torquer is turned on. So the team decided to use the magnetometer and magneto-torquer periodically. i.e, magnetometer is turned off whenever the magneto-torquer is turned on and vice versa.

Considering this, our attitude determination system also gives local magnetic field to controller. This benefits us of using magnetic field from mathematical modal whenever the magnetometer is turned off.



Ti...	Data:1	Data:2	Data:3
0	4.0674e-06	5.4086e-06	2.1937e-05
1	1.3388e-06	2.0082e-06	-2.2820e-05
2	2.0754e-05	7.6791e-06	6.0375e-06
3	2.0128e-05	9.4249e-06	-5.6348e-06
4	2.1404e-05	6.1547e-06	-5.4096e-06
5	2.1438e-05	5.5087e-06	-5.9060e-06
6	2.1427e-05	5.3239e-06	-6.0774e-06
7	2.1458e-05	5.1434e-06	-6.0894e-06
8	2.1452e-05	5.1168e-06	-6.0983e-06
9	2.1463e-05	5.0670e-06	-6.0672e-06
10	2.7395e-05	5.1575e-06	-7.6308e-06
11	2.7401e-05	5.1530e-06	-7.5656e-06
12	2.7406e-05	5.1485e-06	-7.5004e-06
13	2.7411e-05	5.1440e-06	-7.4352e-06
14	2.7416e-05	5.1395e-06	-7.3699e-06
15	2.7421e-05	5.1350e-06	-7.3047e-06
16	2.7426e-05	5.1306e-06	-7.2394e-06
17	2.7431e-05	5.1261e-06	-7.1741e-06
18	2.7436e-05	5.1216e-06	-7.1088e-06
19	2.7440e-05	5.1172e-06	-7.0435e-06
20	2.1498e-05	5.0327e-06	-5.5876e-06
21	2.1497e-05	5.0407e-06	-5.5476e-06



Time	Data:1	Data:2	Data:3
0	NaN	NaN	NaN
1	NaN	NaN	NaN
2	NaN	NaN	NaN
3	NaN	NaN	NaN
4	NaN	NaN	NaN
5	NaN	NaN	NaN
6	NaN	NaN	NaN
7	NaN	NaN	NaN
8	NaN	NaN	NaN
9	NaN	NaN	NaN
10	2.7395e-05	5.1575e-06	-7.6308e-06
11	2.7401e-05	5.1530e-06	-7.5656e-06
12	2.7406e-05	5.1485e-06	-7.5004e-06
13	2.7411e-05	5.1440e-06	-7.4352e-06
14	2.7416e-05	5.1395e-06	-7.3699e-06
15	2.7421e-05	5.1350e-06	-7.3047e-06
16	2.7426e-05	5.1306e-06	-7.2394e-06
17	2.7431e-05	5.1261e-06	-7.1741e-06
18	2.7436e-05	5.1216e-06	-7.1088e-06
19	2.7440e-05	5.1172e-06	-7.0435e-06
20	NaN	NaN	NaN
21	NaN	NaN	NaN

Figure 23: Magnetic field when magnetometer is turned off

Ionospheric and gamma-ray		Ref. : SCA-DD-02
Observations		Edition: 1 Date: 05-09-2016
Satellite		Revision : 2 Date :
		Page : 33

From the above figure, it is seen that when the magnetometer is turned off, the field is not available (*NaN*). In this case, the Attitude determination block computes the magnetic field from the stored magnetic model and given to the attitude controller.

7.1 State vector

The Extended Kalman Filter (EKF) is an optimal recursive method for linear state estimation in presence of noisy measurements. The original formulation applies to linear dynamic systems[15]. For spacecraft attitude estimation, where attitude sensors and gyros are available, it is customary to define the state vector composed by,

$$x = [q \ b]^T$$

Where $q = q_{R_{sat}/R_g}$ is the quaternion that transforms from inertial to satellite frame.
 b = gyros bias.

$$\dot{x} = \begin{bmatrix} \dot{q} \\ \dot{b} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \Omega(\omega)q \\ 0_{3 \times 3} \end{bmatrix} + w \quad \dots(34)$$

where w is the dynamical white noise representing uncertainties in the model, with covariance Q , i.e. $w = N(0, Q)$. The discrete measurement model is given by:

$$y_k = H_k x_k + v_k \quad \dots(35)$$

where H_k is the measurement matrix(see section 7.1.2), y_k is the sensor measured vector, and v_k is the gaussian white noise with covariance matrix R_k , i.e. $v_k = N(0, R_k)$.

The sun and earth sensors are unavailable when the sun and earth are not in the sensors field of view. While the magnetometer is turned off whenever the magneto-torquer is turned on. Considering these limitations of the sensors, we choose gyros as the base sensor for attitude estimation. The angular rates from gyros are considered in the state prediction step, while the sun, earth & magnetic field vectors are considered in the measurement step.

7.2 Filter Propagation & Prediction

Our implementation is similar to [17]. We use the gyros in the state prediction, so the priori estimate is given by, (higher order terms are neglected in quaternion prediction (see Appendix 6)).

$$x_{k+1}^- = \begin{bmatrix} q_{k+1}^- \\ b_{k+1}^- \end{bmatrix} = \begin{bmatrix} q_k^+ \\ b_k^+ \end{bmatrix} + \begin{bmatrix} \frac{\Delta t}{2} \Omega(\hat{\omega}_{k+1})q_k^+ \\ 0 \end{bmatrix} \quad \dots(36)$$

$\Omega(\omega) = \begin{bmatrix} -\omega^x & \omega \\ -\omega^T & 0 \end{bmatrix}$ Where, $\hat{\omega}_{k+1} = \tilde{\omega}_m|_{k+1} - b_k^+$ is the true angular rate from the measured gyro $\tilde{\omega}_m$.

From the state transition equation (34), the Jacobian of the state vector is derived in [16]. The angular rate $\hat{\omega}_{k+1}$ is a function of system dynamics(control & disturbances torques). So the state Jacobian in our case becomes,



$$F = \begin{bmatrix} \mathbf{1}_{4 \times 4} + \frac{\Delta t}{2} \Omega(\hat{\omega}_{k+1}) & \mathbf{0}_{4 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{1}_{3 \times 3} \end{bmatrix}_{7 \times 7} \quad \dots(37)$$

The priori estimate of the state covariance is given as,

$$P_{k+1}^- |_{7 \times 7} = P_k^+ F P_k^{+T} + Q_d |_{7 \times 7} \quad \dots(38)$$

The initial state noise covariance is given by,

$$P_0^+ = \begin{bmatrix} \sigma_{quat}^2 & \mathbf{0}_{4 \times 3} \\ \mathbf{0}_{3 \times 3} & \sigma_{bias}^2 \end{bmatrix}$$

According to [17], discrete noise covariance in our case becomes,

$$Q_d = \begin{bmatrix} Q_{11} & \mathbf{0}_{3 \times 4} & Q_{12} \\ \mathbf{0}_{1 \times 7} & & \\ Q_{12}^T & \mathbf{0}_{3 \times 4} & Q_{22} \end{bmatrix}$$

$$Q_{11} = \sigma_r^2 \Delta t \cdot \mathbf{1}_{3 \times 3} + \sigma_w^2 \cdot \left(\mathbf{1}_{3 \times 3} \frac{\Delta t^3}{3} + \frac{(|\hat{\omega}| \Delta t)^3 + 2 \sin(|\hat{\omega}| \Delta t) - 2|\hat{\omega}| \Delta t}{|\hat{\omega}|^5} \cdot [\hat{\omega} \times]^2 \right)$$

$$Q_{12} = -\sigma_w^2 \cdot \left(\mathbf{1}_{3 \times 3} \frac{\Delta t^2}{2} - \frac{|\hat{\omega}| \Delta t - \sin(|\hat{\omega}| \Delta t)}{|\hat{\omega}|^3} \cdot [\hat{\omega} \times] + \frac{(|\hat{\omega}| \Delta t)^2 + \cos(|\hat{\omega}| \Delta t) - 1}{|\hat{\omega}|^4} \cdot [\hat{\omega} \times]^2 \right)$$

$$Q_{22} = \sigma_w^2 \Delta t \cdot \mathbf{1}_{3 \times 3}$$

7.3 Measurement model & Update

There are 3 models implemented in the attitude EKF one for each type of measurement. In order from top to bottom, sun sensor model, magnetic field model and the earth sensor model:

The Measurement vectors from the sensors are formulated as,

$$y_{k+1} = \begin{bmatrix} s_{3 \times 1}^b \\ m_{3 \times 1}^b \\ e_{3 \times 1}^b \end{bmatrix}_{9 \times 1} \quad \dots(39)$$

The magnetometer biases m_{bias} is computed offline by sensor testing and is considered in magnetometer data $m_{3 \times 1}^b = \tilde{m}_{3 \times 1}^b - m_{bias}$

Measurement prediction is given by,



$$\hat{y}_{k+1} = \begin{bmatrix} M(q_{k+1}^-)_{3 \times 3} s_{3 \times 1}^i \\ M(q_{k+1}^-)_{3 \times 3} m_{3 \times 1}^i \\ M(q_{k+1}^-)_{3 \times 3} e_{3 \times 1}^i \end{bmatrix}_{9 \times 1} \quad \dots(40)$$

$M(q_{k+1}^-)_{3 \times 3}$ is the rotation matrix of the quaternion q_{k+1}^- from the prediction step

The measurement model is adopted from[16] and is given by

Sun sensor,
$$H_{ss} = \begin{bmatrix} \eta 1_{3 \times 3} + \varepsilon^\times & \\ & -\varepsilon^T \end{bmatrix}^T \begin{bmatrix} (s^i)^\times & s^i \\ -(s^i)^T & 0 \end{bmatrix},$$

Magnetometer,
$$H_{mg} = \begin{bmatrix} \eta 1_{3 \times 3} + \varepsilon^\times & \\ & -\varepsilon^T \end{bmatrix}^T \begin{bmatrix} (m^i)^\times & m^i \\ -(m^i)^T & 0 \end{bmatrix},$$

Earth sensor,
$$H_{es} = \begin{bmatrix} \eta 1_{3 \times 3} + \varepsilon^\times & \\ & -\varepsilon^T \end{bmatrix}^T \begin{bmatrix} (e^i)^\times & e^i \\ -(e^i)^T & 0 \end{bmatrix}$$

Combined
$$H_{k+1} = \begin{bmatrix} H_{ss} & \vdots \\ H_{mg} & 0_{9 \times 3} \\ H_{es} & \vdots \end{bmatrix}_{9 \times 7} \quad \dots(41)$$

Where ε and η are the vector & scalar part of the quaternion q_{k+1}^- .

H_{k+1} is a 9×7 matrix & s^i m^i e^i are the sun direction, magnetic field and earth direction unit vectors in the inertial frame computed from the ephemeris & magnetic models(see *Appendix 4 & 5*).

EKF Update equations are as follows,

$$S_{k+1} = H_{k+1} P_{k+1}^- H_{k+1}^T + R_{9 \times 9}$$

$$K_{k+1} = P_{k+1}^- H_{k+1}^T S_{k+1}^{-1}$$

$$x_{k+1}^+ = x_{k+1}^- + K_{k+1} (y_{k+1} - \hat{y}_{k+1})$$

$$P_{k+1}^+ = P_{k+1}^- - K_{k+1} H_{k+1} P_{k+1}^-$$

...(42)

State error

$$dx = x_{k+1}^+ - x_{k+1}^-$$

Where K_{k+1} is the 7×9 Kalman gain, and R is the measurement noise covariance,

$$R = \begin{bmatrix} \sigma_{ss}^2 1_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & \sigma_{mg}^2 1_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & \sigma_{es}^2 1_{3 \times 3} \end{bmatrix}_{9 \times 9}$$

Where σ_{ss} , σ_{mg} and σ_{es} are the measurement noise standard deviation in sun sensor, magnetometer and earth sensors respectively.



$x_{k+1}^+ = \begin{bmatrix} q_{k+1}^+ & b_{k+1}^+ \end{bmatrix}_{R_{Sat}/R_g}^T$ is the Kalman estimated attitude in Inertial frame. The Attitude controller requires attitude wrt to local orbit frame. Using the Orbit information, the attitude wrt orbit can be determined.

The tuning parameters $Q_{7 \times 7}$ and $R_{9 \times 9}$ are the state & measurement noise covariance's that are computed from the sensors tests.

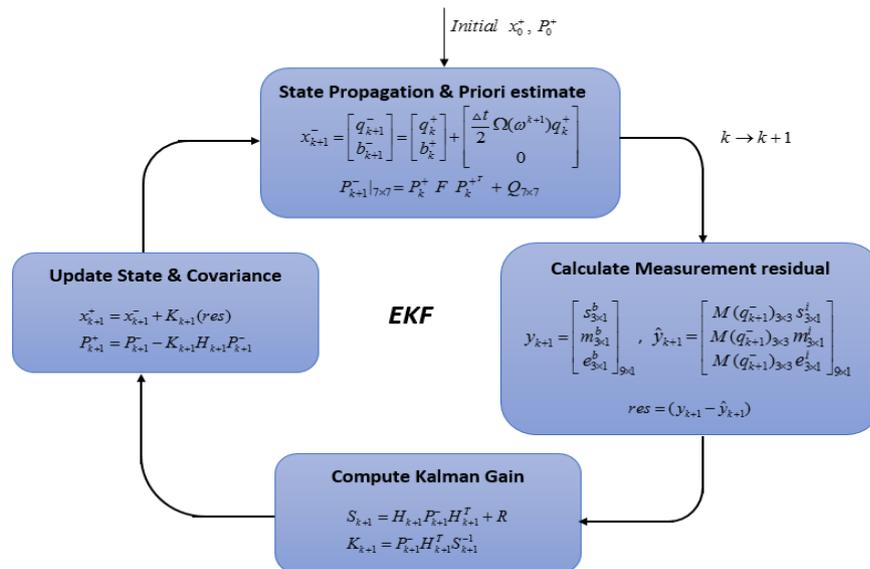


Figure 24: EKF Process

Inputs

	Size	Type
Position in Inertial frame (Km)	3 x 1	double
Velocity in Inertial frame (Km/sec)	3 x 1	double
Angular velocity from gyros(rad/ sec)		
Time in UTC [YYYY MM DD HH mm ss]	1 x 6	double
Unit sun vector	3 x 1	double
Magnetic field vector	3 x 1	double
Unit Earth/nadir vector	3 x 1	double

Outputs

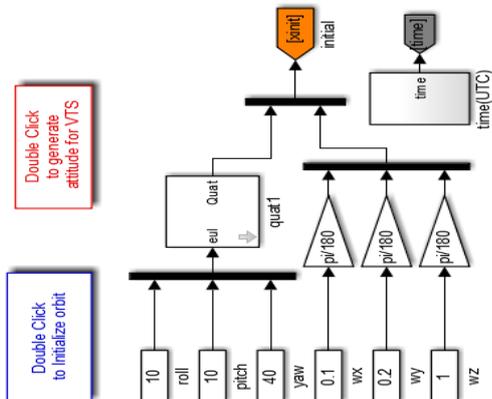
		Size	Type
State wrt Inertial	$\begin{bmatrix} q_{4 \times 1} & \omega_{3 \times 1} & b_{3 \times 1} \end{bmatrix}_{R_{Sat}/R_g}^T$	10 x 1	double
State wrt local orbit frame	$\begin{bmatrix} q_{4 \times 1} & \omega_{3 \times 1} \end{bmatrix}_{R_{Sat}/R_{loc}}^T$	7 x 1	double
State error wrt Inertial	$\begin{bmatrix} q_{4 \times 1} & \omega_{3 \times 1} \end{bmatrix}_{R_{Sat}/R_g}^T$	7 x 1	double
Magnetic field		3 x 1	double



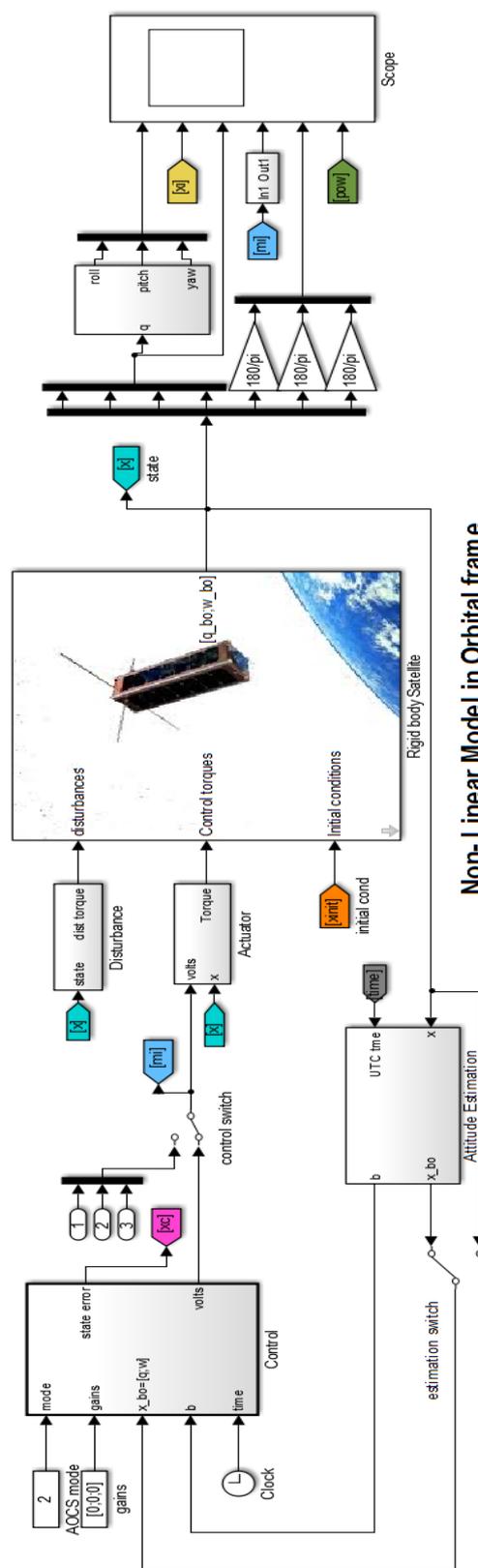
8. SIMULINK

Steps

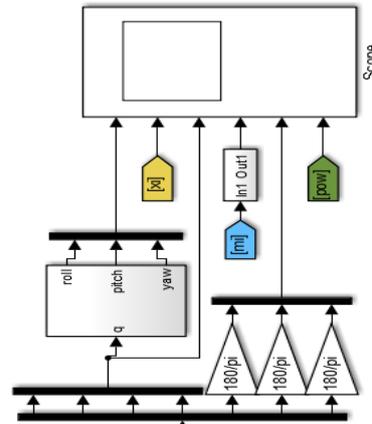
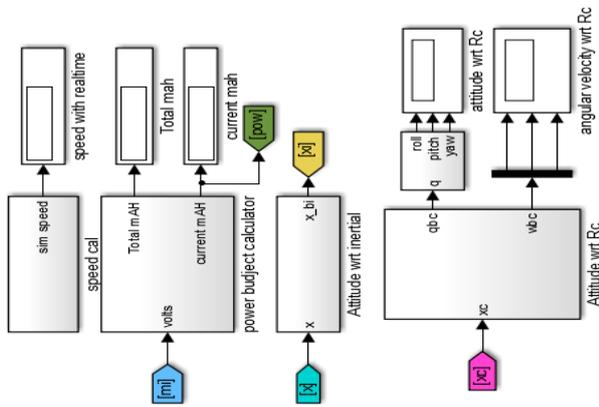
1. Double click the block (in blue colour) to load the orbit & satellite parameters.
2. Enter the initial conditions in degrees.
3. Enter the AOCSS mode
 Detumble 1
 Mission 2
 Combined 3
 Caullou 4
4. Run the simulation.
5. Double click block in red to generate attitude for VTS



Initial conditions

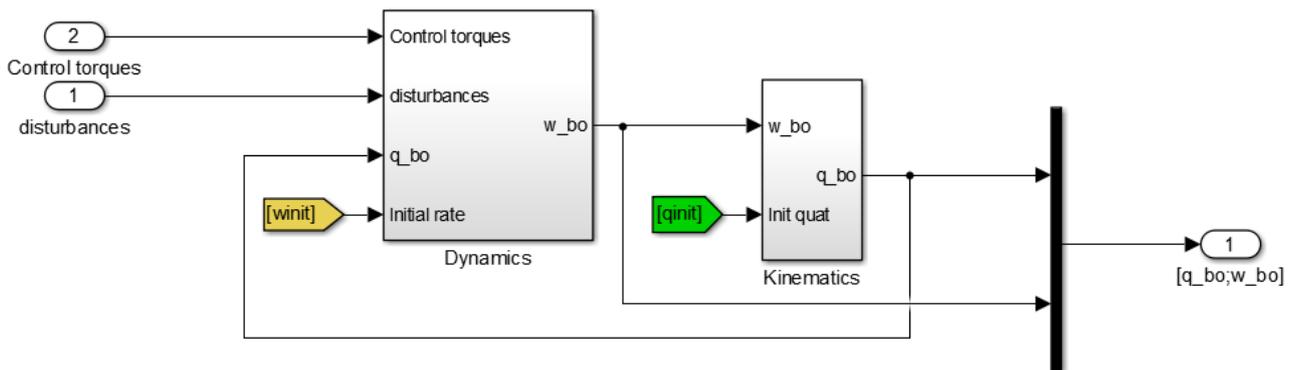


Non-Linear Model in Orbital frame

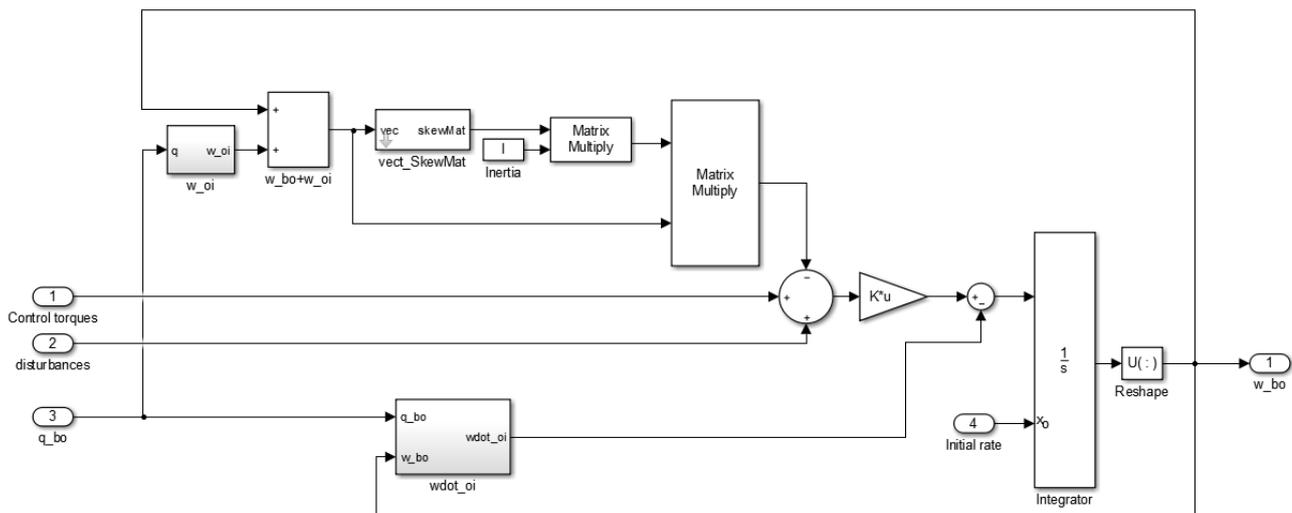




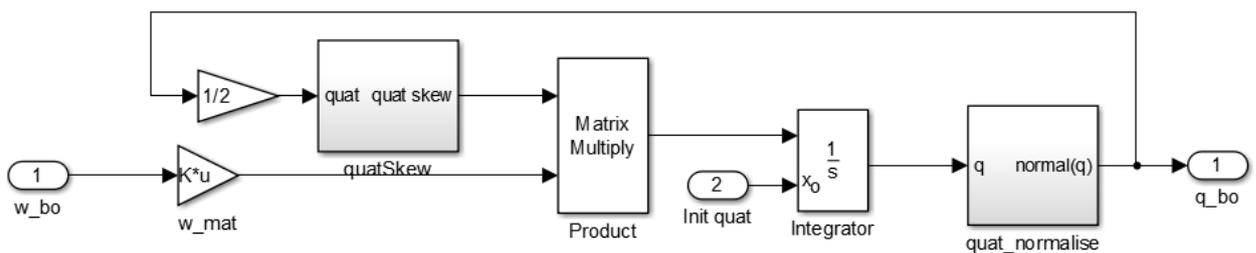
Rigid body satellite sub-system block



Dynamics subsystem block

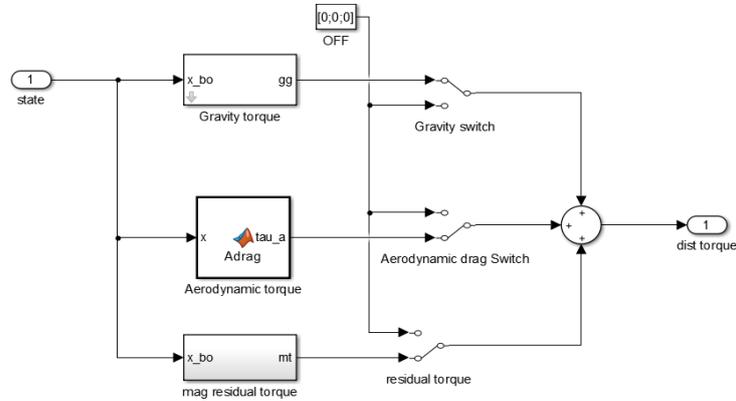


Kinematics subsystem block

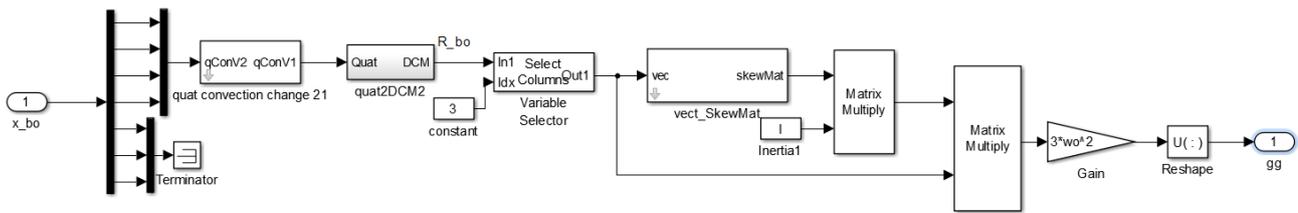




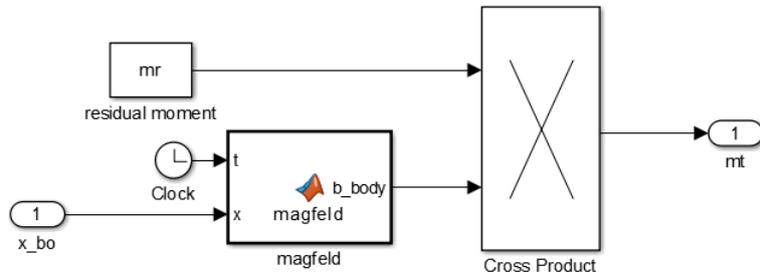
Disturbance block



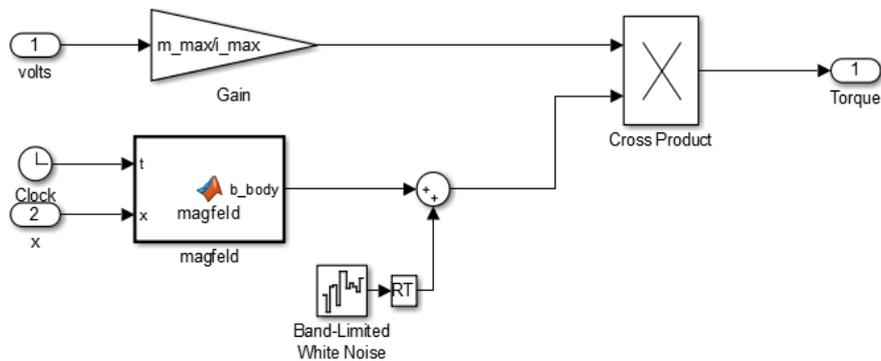
Gravity gradient



Residual torque

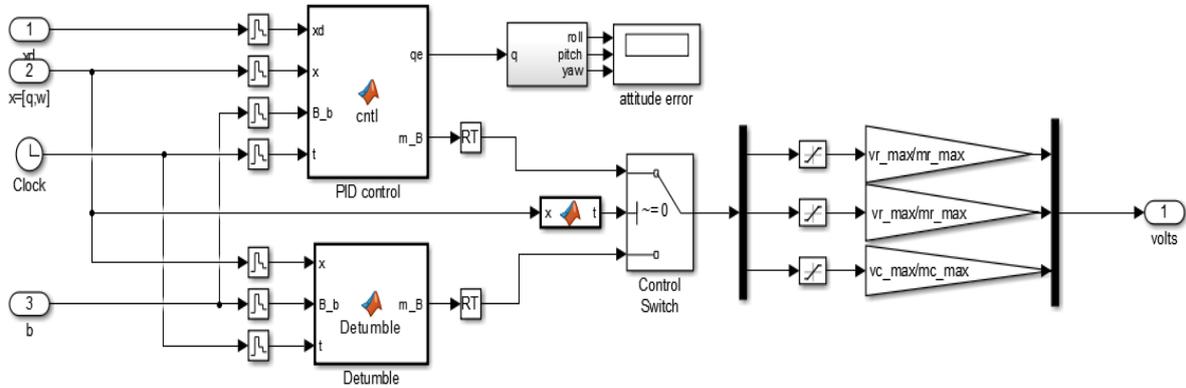


Actuator (magnetorquer)

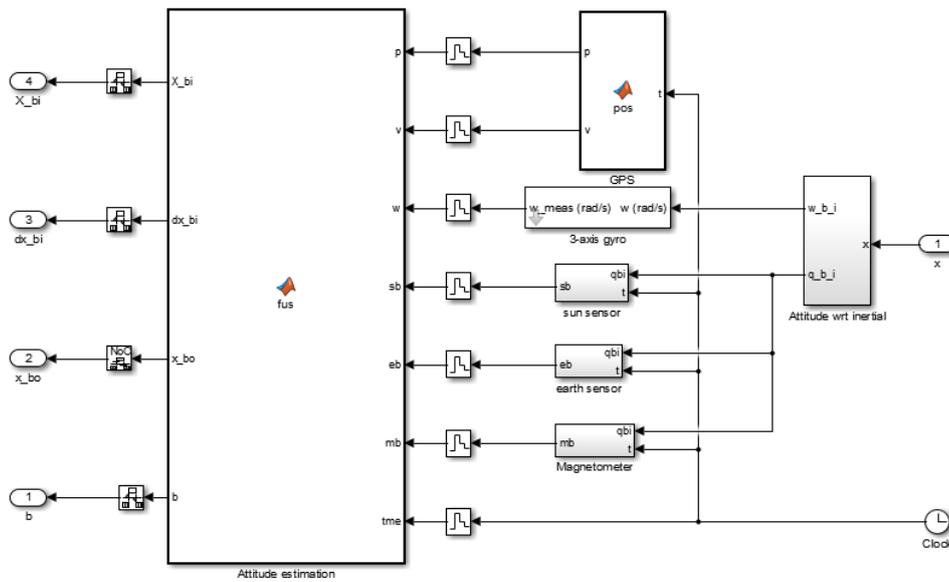




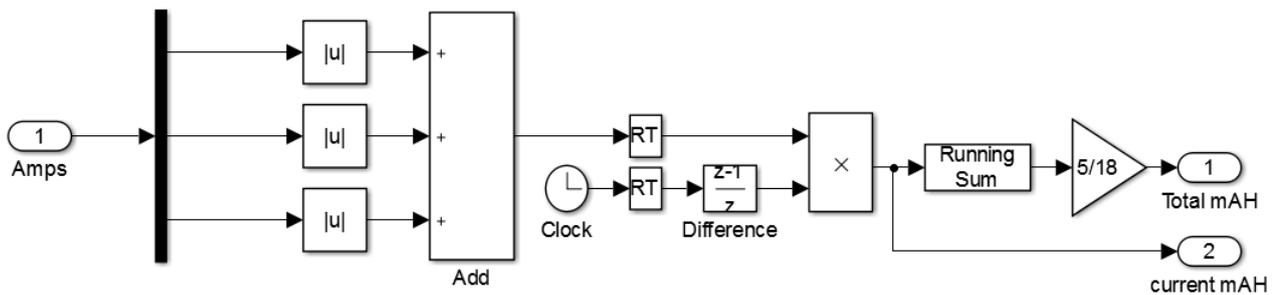
Attitude Control



Attitude Determination block



Power Budget block





OPERATING INSTRUCTIONS

A brief instructions to operate the Simulink model is presented in the Simulink itself.

Before running it, please keep the following points in mind,

1. The Attitude control with Magneto-torquers are verified for only nadir and nadir with spin along- Z_{sat} axis. The Desired attitude frame R_c is comprised of six parameters.

a) For nadir with spin, the desired attitude frame R_c is defined by,

$$\begin{aligned}\varphi &= 0 \\ \theta &= 0 \\ \psi &= \text{spinrate} \times t \\ \dot{\varphi} &= 0 \\ \dot{\theta} &= 0 \\ \dot{\psi} &= \text{spinrate}(= 0.1^\circ / \text{sec})\end{aligned}$$

b) For nadir (no spin), the desired attitude frame R_c is defined by,

$$\varphi = 0, \theta = 0, \psi = 0, \dot{\varphi} = 0, \dot{\theta} = 0, \dot{\psi} = 0$$

2. A switch to turn-off the disturbances (gravity, aerodynamic & residual), Attitude estimation and control are to be checked before running the simulation.
3. Choosing the sampling time of sensors, attitude estimation, and simulation step are to be entered in the Initialization file "*orbitParameters.m*". Satellite characteristics, actuator and orbital information are also entered in the same file.
4. Set the VTS directory in "*toVTS.m*" file.
5. The Simulink file and its supported Matlab files should be in the same directory.

Steps to operate

1. Double click the block (in blue colour) to load the orbital elements.
2. Enter the initial conditions in degrees.
3. Enter the AOCS mode(default is set to combined detumble & PID)
 - Detumble 1
 - Mission 2
 - Combined 3
 - Caullou 4
4. Run the simulation.
5. Double click the red block to generate attitude file for VTS.

After the simulation is performed, the results can be exported to VTS for 3D visualization of attitude.

In VTS,

Ionospheric and
gamma-ray

Observations

Satellite



Ref. : SCA-DD-02

Edition: 1 Date: 05-09-2016

Revision : 2 Date :

Page : 42

- Set the VTS Initializing parameters(directory,....)
- Load the 3ds file of IGOSAT
- Load positions & quaternion texts file generated by Simulink.
- Click on run.



9. RESULTS

The following parameters are considered in the simulation

Orbital parameters

Altitude	650 Km
Eccentricity	0
longitude of the ascending node	37.9 deg
Orbit Inclination	98.6
Argument of perigee	0 deg
Mean Anomaly	0 deg

Launch date is 15th June 2018 at 00:00 IST.

Satellite characteristics

Length	34.5 cm
breadth	10 cm
height	10 cm
Mass	3.77 kg

Actuator characteristics

	Along x	Along y	Along z
Maximum moment(A-m ²)	± 0.2	± 0.2	± 0.2
Maximum Voltage(V)	2.5	2.5	5

Sampling time

Magnetometer	Sun sensor	Earth sensor	Gyros	Attitude control	Attitude determination
1 Hz	1 Hz	1 Hz	1 Hz	1 Hz	1 Hz

Please note that gravity gradient, aerodynamic and magnetic residual torques are considered in all the simulations except in the optimal gain calculation (section 9.4), where only gravity torque is considered.

9.1 Detumbling Results

As mentioned in section 6.2, a total of 3 de-tumbling control laws are tested for 5 orbits.

Initial attitude and angular rates with respect to orbit are considered to be,

Along x φ (deg)	Along y θ (deg)	Along z Ψ (deg)	ω_x (deg/sec)	ω_y (deg/sec)	ω_z (deg/sec)
10	10	40	10	10	10



B-dot control law

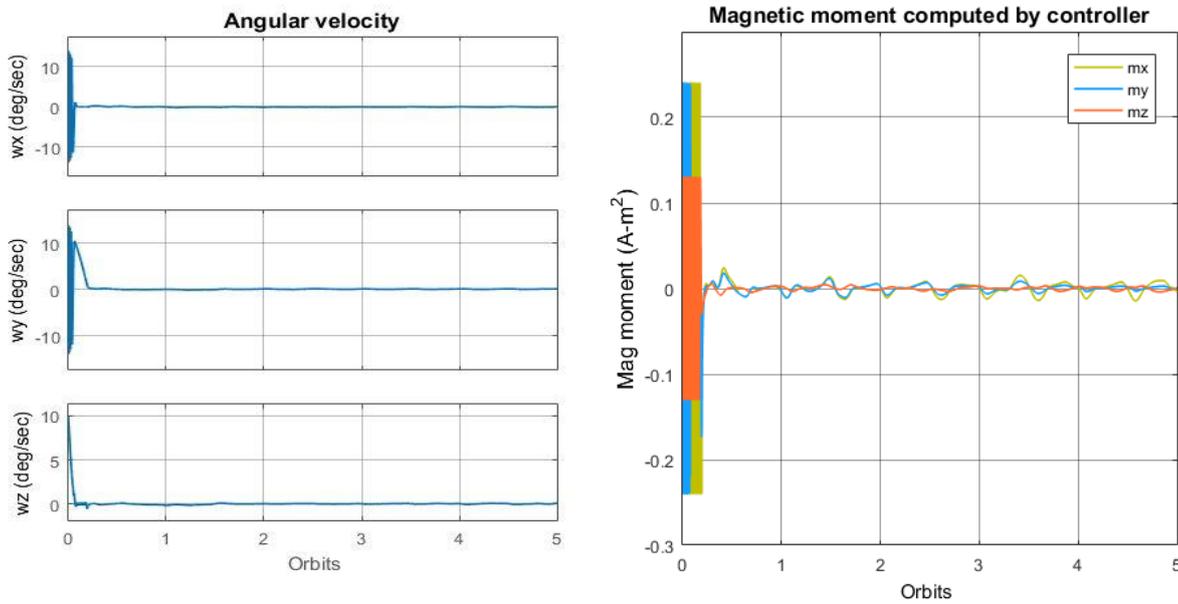


Figure 25:Angular rates & Magnetic moment : B-dot control

B-bang bang control law

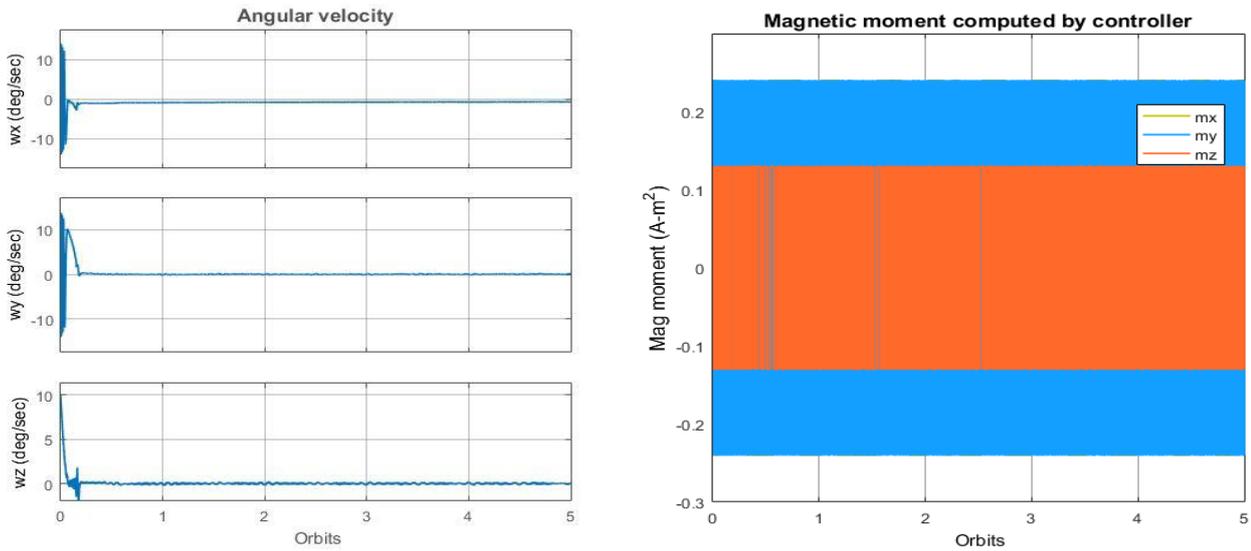


Figure 26:Angular rates & Magnetic moment :B-bang bang control



Desired torque control law

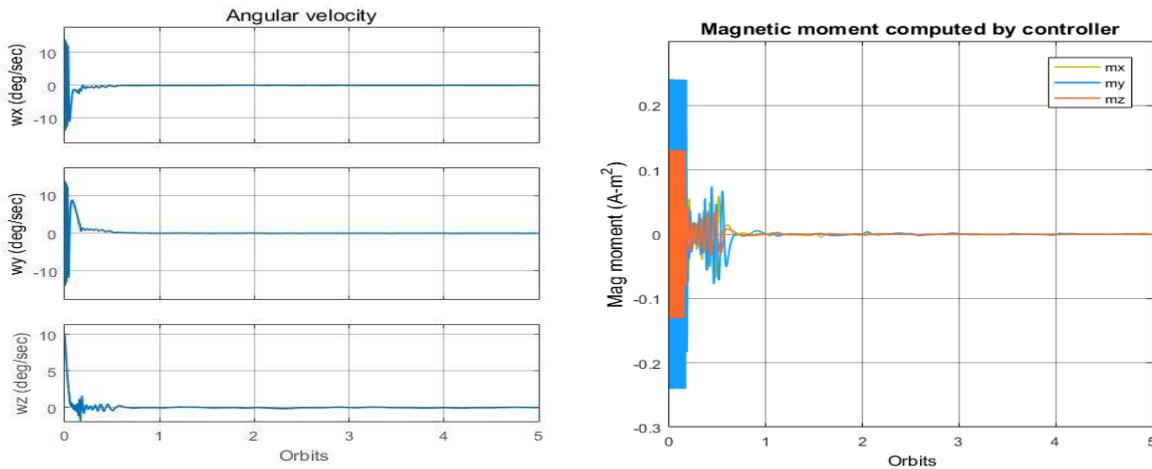


Figure 27:Angular rates & Magnetic moment : Desired torque control law

Comparison

	B-dot	B-bang bang	Desire torque control
Tumbling time to reach 0.5 deg/sec	~0.6 orbits	~0.5 orbits	~0.6 orbits
Power consumed(maH)	86.5 maH	1845 maH	82.1 maH

It clearly seen that *B-bang bang* tumbles faster but the performance is not smooth in reaching the tumble state. Also the power consumed is far higher compared to other control laws.

According to the QB50 CubeSat requirements, the de-tumbling controller should be able to tumble the initial 40 deg/sec on each axis. The initial conditions with respect to orbit are,

Along x φ (deg)	Along y θ (deg)	Along z Ψ (deg)	ω_x (deg/sec)	ω_y (deg/sec)	ω_z (deg/sec)
10	10	10	40	35	25

B-bang bang

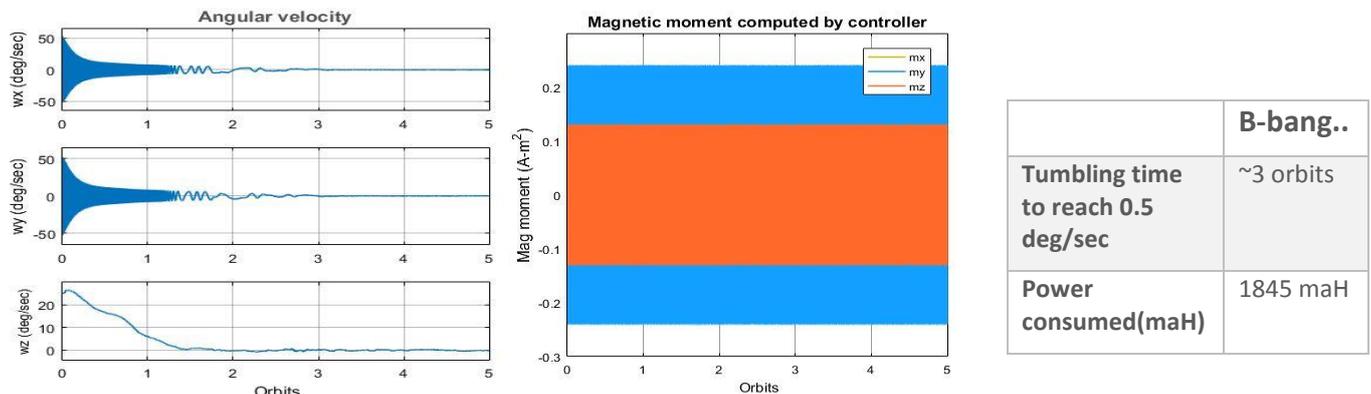
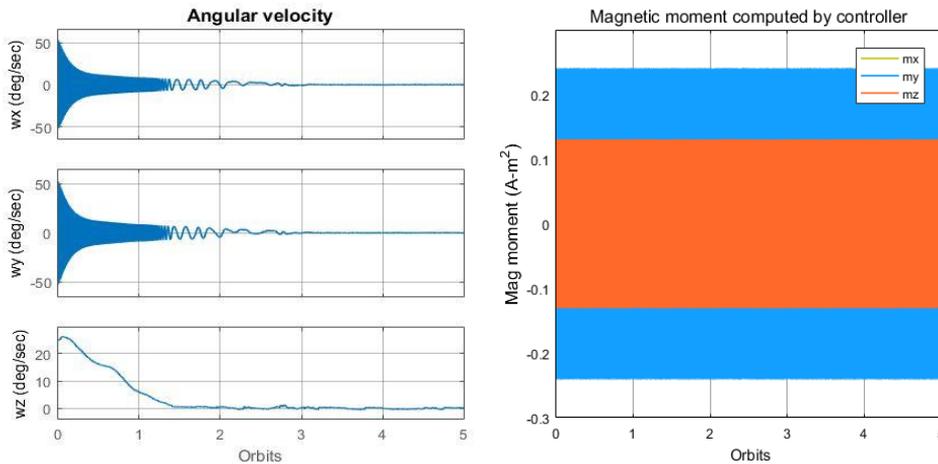


Figure 28:Angular rates & Magnetic moment(40°/sec) : B-bang bang control



B-dot



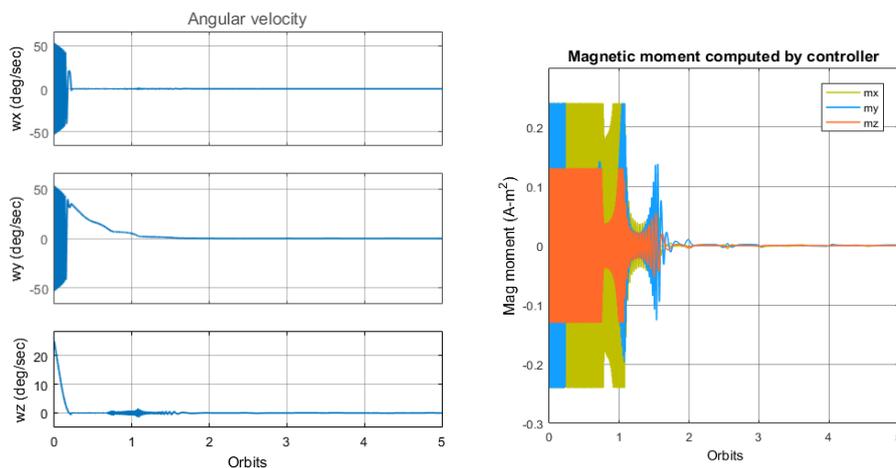
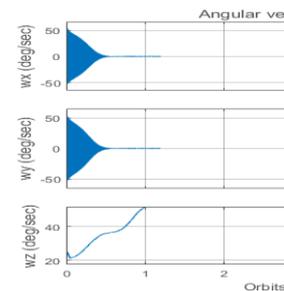
	B-dot
Tumbling time to reach 0.5 deg/sec	~2.8 orbits
Power consumed(maH)	1836 maH

Figure 29: Angular rates & Magnetic moment(40°/sec) : B-dot control

Desired torque control law

Here we observe that with the sampling time 1 Hz, the desired torque control couldn't damp the rotations and indeed increases the rotation(see beside figure).

So we reduced samling time to 2 Hz.



	Desired torque control
Tumbling time to reach 0.5 deg/sec	~1.7 orbits
Power consumed(maH)	949.5 maH

Figure 30: Angular rates & Magnetic moment(40°/sec): Desired torque control law

Though desired torque control damps faster and consumed less power, it is observed an unexpected behavior at 0.6 orbit even after changing to 2 Hz. From this simulation shows that, desired torque control is not valid for higher rotation rates with a sampling time of 1 Hz. While the both B-dot & B-bang bang shows good performance with higher initial angular rates & of sampling time at 1 Hz. Although power consumption is little higher in B-bang bang, it reached the tumbling state little slower than B-dot.



	B-dot	B-bang bang	Desire torque control
Tumbling time to reach 0.5 deg/sec	~2.8 orbits	~3 orbits	~1.7 orbits
Power consumed(maH)	1836 maH	1845 maH	949.5 maH

In our further work, we decided to implement B-dot control for de-tumbling.

9.2 PID Control Results

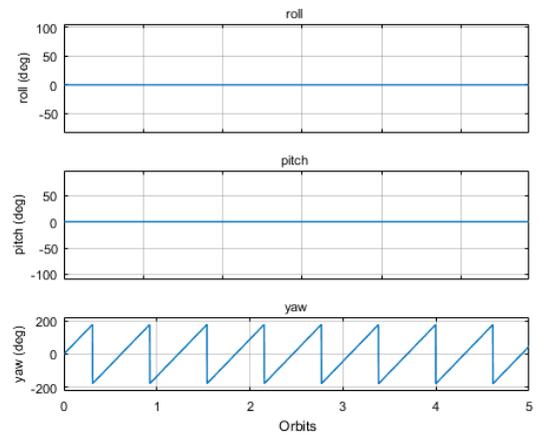
PID control is used for the nominal mission mode. The satellite should be nadir pointing with a spin of $0.1^\circ/\text{sec}$ in Z-axis.

So our Desired roll, pitch & yaw with respect to local orbital axes will varies as shown in beside figure,

The PID gains are, $k_p = 4e^{-5}$, $k_i = 2e^{-10}$ and $k_d = 2e^{-2}$

These are the optimal gains which are calculated in the section 9.4.

The initial conditions of the satellite with respect to local orbital frame are,



Along x $\varphi(\text{deg})$	Along y $\theta(\text{deg})$	Along z $\Psi(\text{deg})$	$\omega_x(\text{deg}/\text{sec})$	$\omega_y(\text{deg}/\text{sec})$	$\omega_z(\text{deg}/\text{sec})$
10	10	40	0.1	0.2	1

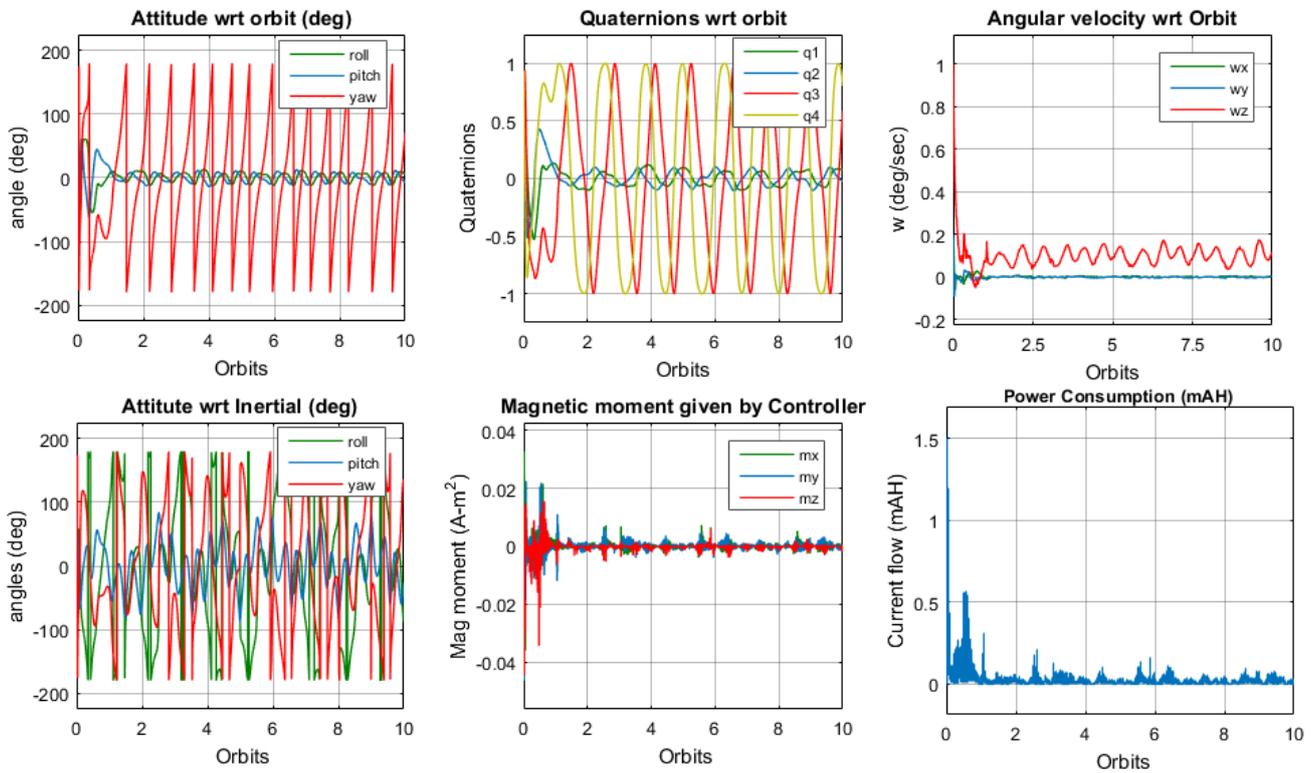


Figure 31: Nominal mode(PID) control for 10 orbits

9.3 Combined De-tumble & PID Control

According to attitude requirements the switching between detumbling mode & Nominal mode should be automatic. So the switching to Nominal (PID) control is defined based on angular velocity reaching ≤ 0.5 deg/sec on each axis.

Initial condition with respect to orbit are,

Along x φ (deg)	Along y θ (deg)	Along z Ψ (deg)	ω_x (deg/sec)	ω_y (deg/sec)	ω_z (deg/sec)
10	10	10	10	10	10

We observe that tumbling is achieved in ~ 0.9 orbit and detumbling is switched to PID after 0.9 orbit i.e, angular velocity at ≤ 0.5 deg/sec.

The power consumed for this combination and for 5 orbits is 268.5 maH.

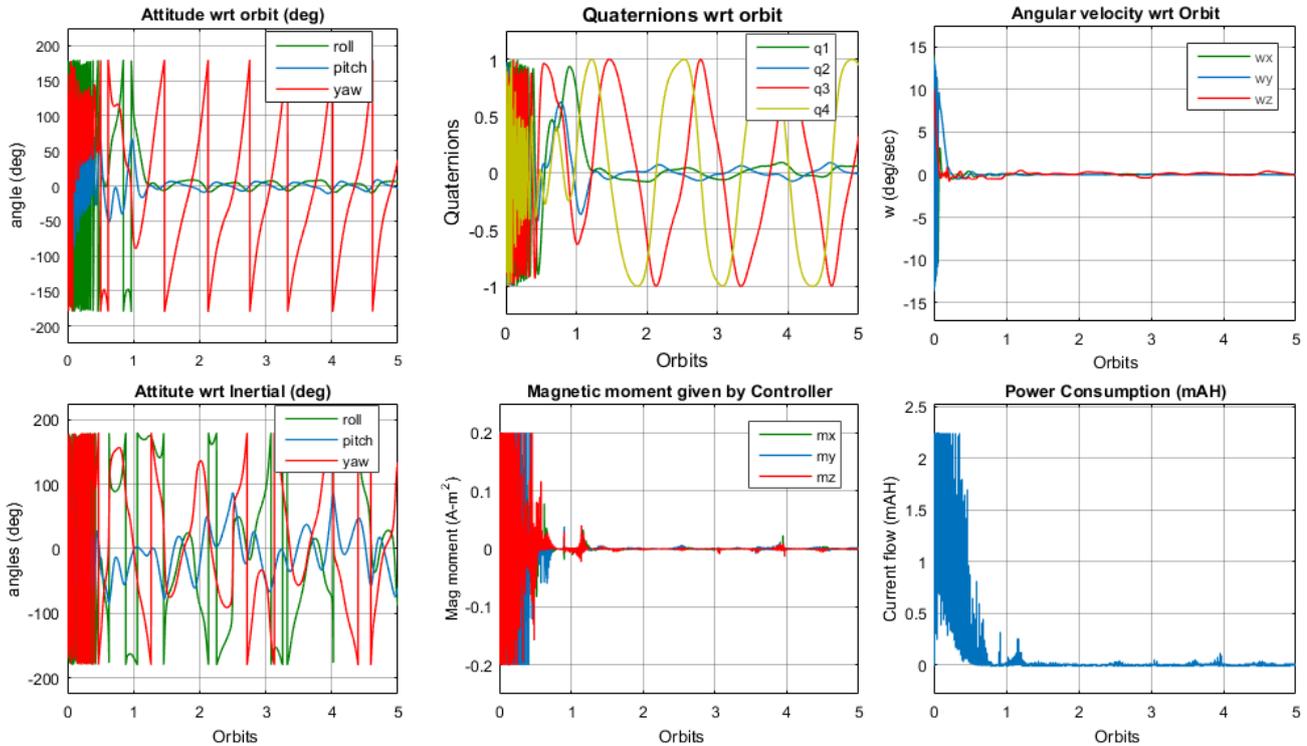


Figure 32: Combined De-tumble & PID controller

9.4 Optimal Gain Calculation

In this section, we consider only gravity gradient torque for the optimal gain calculation.

Since $k_i = 2e^{-10}$ is very small and seems to not influence much the control performance, we decided to tune only k_p and k_d . And note that the gains are proportional to the mass matrix I .

We use the parameters *mean attitude error* and *power consumption* criteria for the optimal gain calculation. A map of various gains with their power consumption & mean attitude error are plotted and optimal gain region is defined by balancing the power consumption & mean attitude error.

Initial conditions considered are

Along x φ (deg)	Along y θ (deg)	Along z Ψ (deg)	ω_x (deg/sec)	ω_y (deg/sec)	ω_z (deg/sec)
10	10	40	0.1	0.2	0.3



Case I: $k_p = 4e^{-6}$, $k_i = 2e^{-10}$ and $k_d = 2e^{-2}$

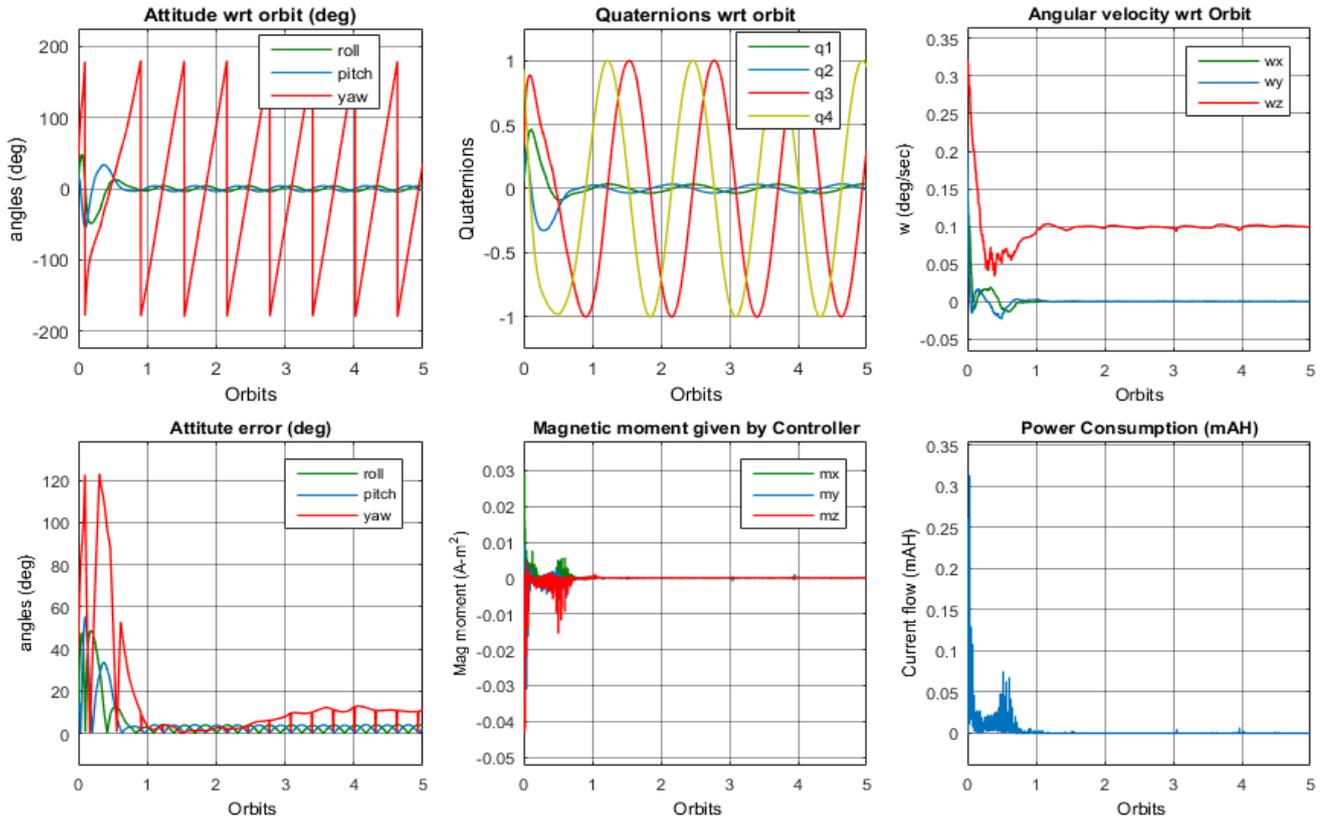


Figure 33: Nominal mode with $k_p = 4e^{-6}$, $k_i = 2e^{-10}$ and $k_d = 2e^{-2}$



Case II: $k_p = 4e^{-5}$, $k_i = 2e^{-10}$ and $k_d = 2e^{-2}$

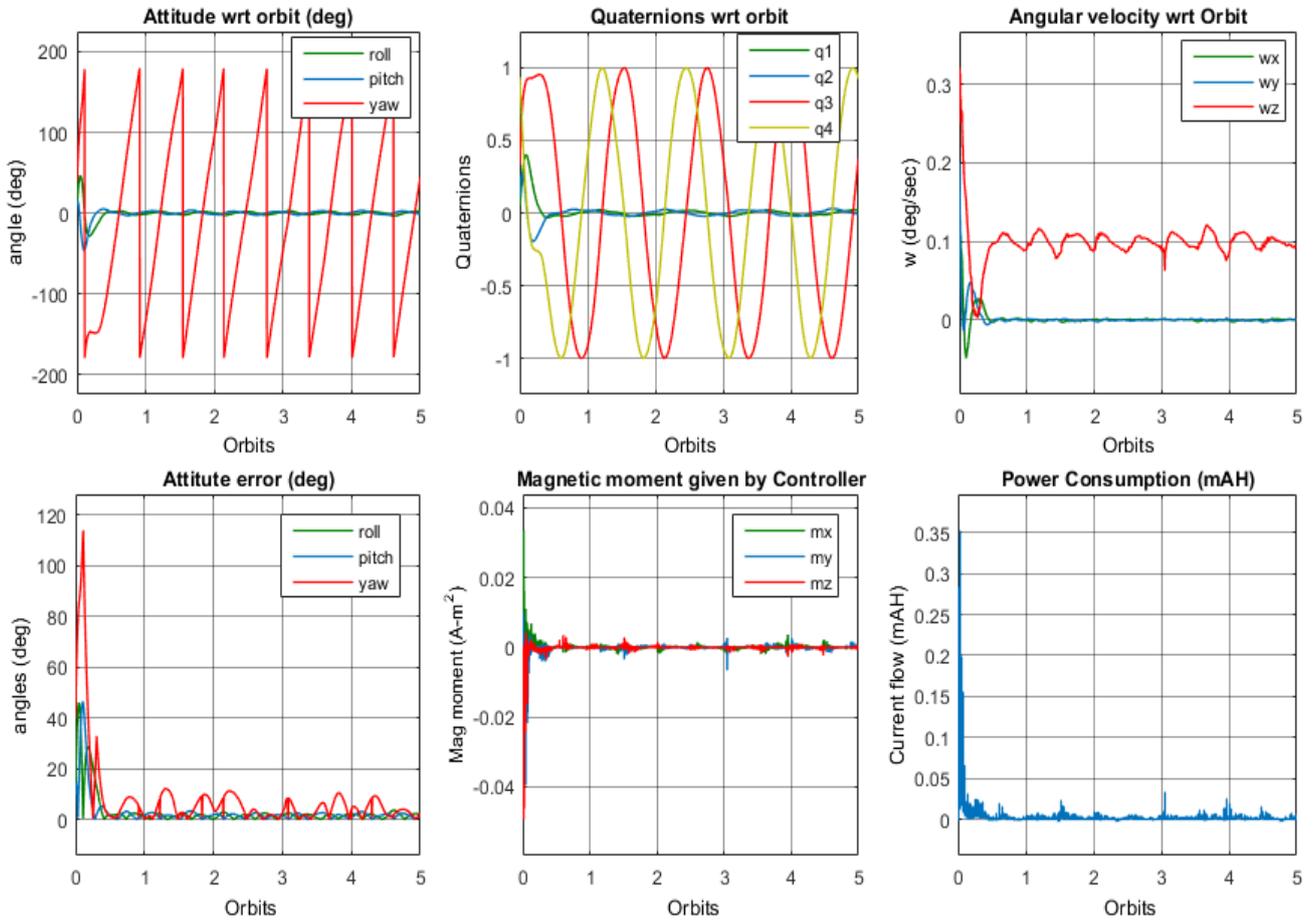


Figure 34: Nominal mode with $k_p = 4e^{-5}$, $k_i = 2e^{-10}$ and $k_d = 2e^{-2}$



Case III: $k_p = 4e^{-4}$, $k_i = 2e^{-10}$ and $k_d = 2e^{-2}$

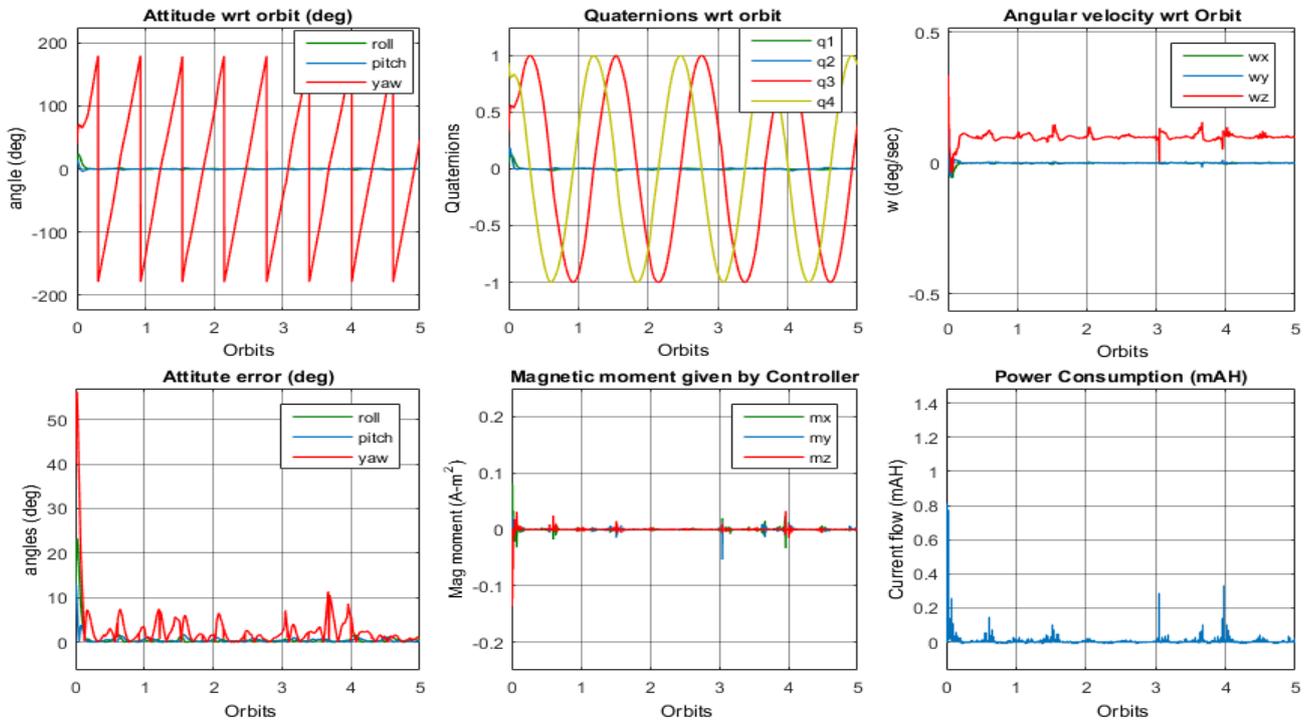


Figure 35: Nominal mode with $k_p = 4e^{-4}$, $k_i = 2e^{-10}$ and $k_d = 2e^{-2}$

	k_p	k_d	Roll error (deg)	Pitch error (deg)	Yaw error (deg)	Power consumed(maH)
Case I	4,00E-06	2,00E-02	5,225	4,932	15,67	7,61
Case II	4,00E-05	2,00E-02	2,969	2,629	7,73	10,39
Case III	4,00E-04	6,00E-02	0,6466	0,4925	2,878	22,08



A further different gains are simulated as tabulated below,

S no	k_p	k_d	Roll error (deg)	Pitch error (deg)	Yaw error (deg)	Power consumed(maH)
1	1,00E-08	1,00E-02	10,2	8,1	66,45	6,95
2	4,00E-08	1,00E-08	10,15	8,06	66,28	6,94
3	1,00E-07	1,00E-02	10,14	8,082	66,3	6,9
4	1,00E-07	2,00E-02	5,59	5,338	26,67	7,262
5	1,00E-06	2,00E-02	5,5	5,24	23,33	7,3
6	3,00E-06	2,00E-02	5,344	5,033	17,55	7,506
7	4,00E-06	2,00E-02	5,2	4,93	15,67	7,61
8	7,00E-06	2,00E-02	4,985	4,635	13,19	7,92
9	9,00E-06	2,00E-02	4,8	4,4	12,07	8,1
10	1,00E-05	2,00E-02	4,724	4,345	11,62	8,2
11	4,00E-05	2,00E-02	2,969	2,629	7,73	10,39
12	5,00E-05	2,00E-02	2,603	2,462	8,669	11,78
13	6,00E-05	4,00E-02	2,103	2,191	6,397	12,14
14	7,00E-05	5,00E-02	1,978	1,978	5,776	12,84
15	9,00E-05	6,00E-02	1,75	1,681	5,08	13,99
16	1,10E-04	6,00E-02	1,557	1,462	4,712	14,97
17	4,00E-04	6,00E-02	0,6466	0,4925	2,878	22,08
18	1,90E-03	6,00E-02	0,3912	0,3507	1,491	67,54
19	2,20E-03	6,00E-02	0,4329	0,3481	1,614	81,56
20	4,00E-03	6,00E-02	0,8715	0,7199	4,548	287,2

Since the satellite is spinning around Z axis. We are interested in lower mean error in roll & pitch, than in yaw error.



The following map is plotted from the above simulations with various gains. The optimal region is defined by balancing minimum power consumption and lower mean attitude error. The selection of gains is the trade-off between the performance & power requirements.

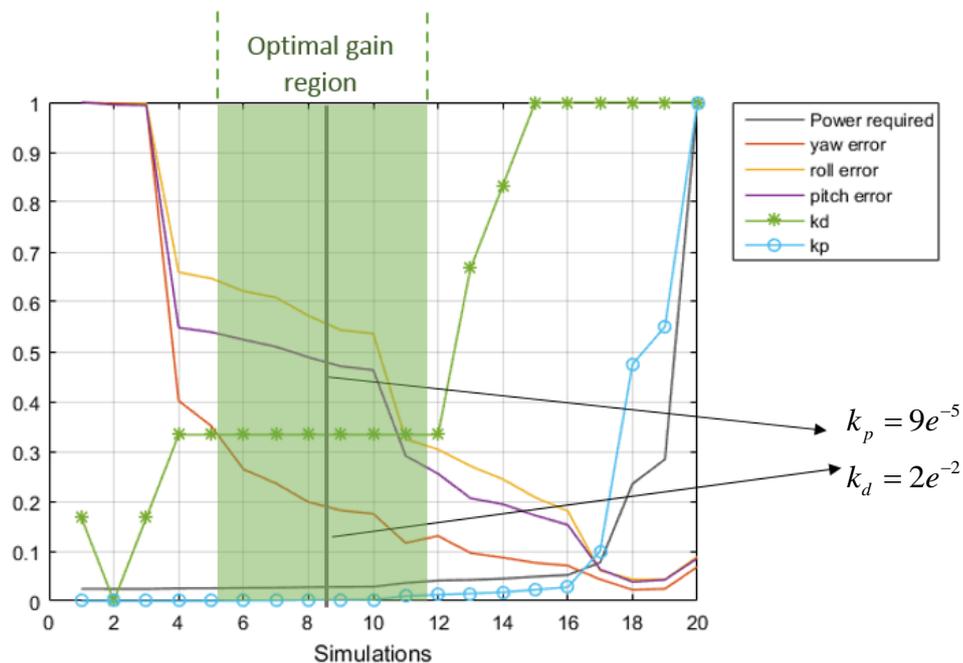


Figure 36: Mapping of various PID gains

These optimal gains are simulated with the environmental disturbances and we found that,

$$\text{Control Gains} \propto \frac{1}{\text{Disturbances}}$$

We propose two set of gains,

Optimal (Lower sensor noise)	Optimal (default) (Higher sensor noise)
$k_p = 4e^{-5}$	$k_p = 9e^{-6}$
$k_d = 2e^{-2}$	$k_d = 2e^{-2}$
$k_i = 2e^{-10}$	$k_i = 2e^{-10}$

In the beginning of the mission, the satellite is unaware of its attitude and initially ADS has higher state error and noisy estimates. So the default gains are set as column-2 of above table. Therefore, once the satellite has reached the steady state, gains can be changed as in column-1 from Tele-command or configuration panel.



9.5 Attitude Estimation

For the simulation confidence, we added the white noise in sensors & actuators models.

The following plots shows the kalman estimates of Angular velocity in the Inertial frame. Note that the left plot is zoomed.

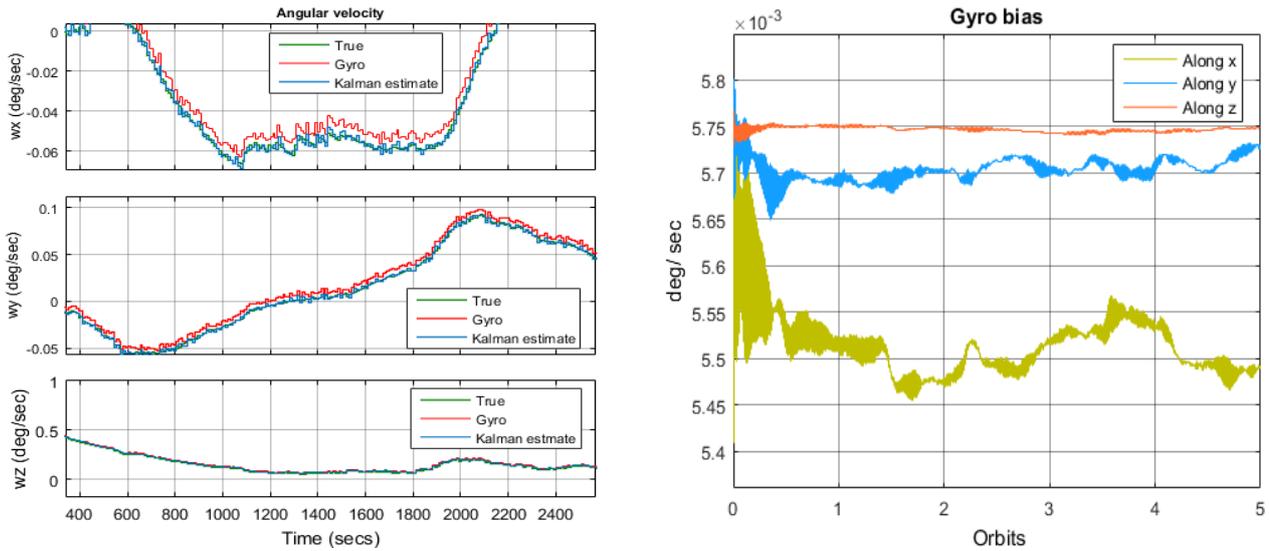


Figure 37: Angular velocity & gyro bias estimated by EKF

These sensors measurements are fed to the attitude estimation block. The below figure shows the attitude estimated by the estimation block with the fusion algorithm as discussed in section 7.

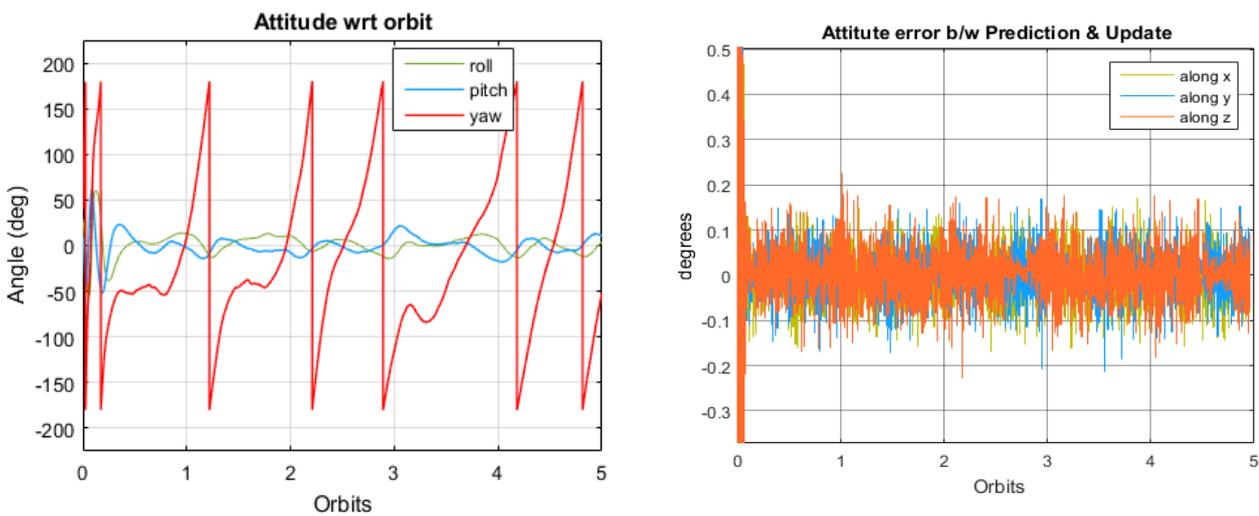


Figure 38: Attitude estimation results



10. AUTOCODE GENERATION

The Matlab interface provides the automatic generation of C and C++ codes. That can be integrated to the flight Software. There are two ways of generating autocode.

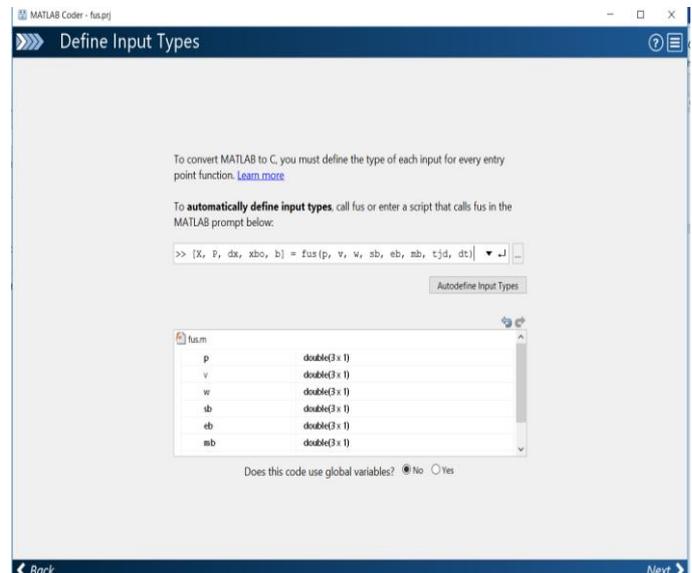
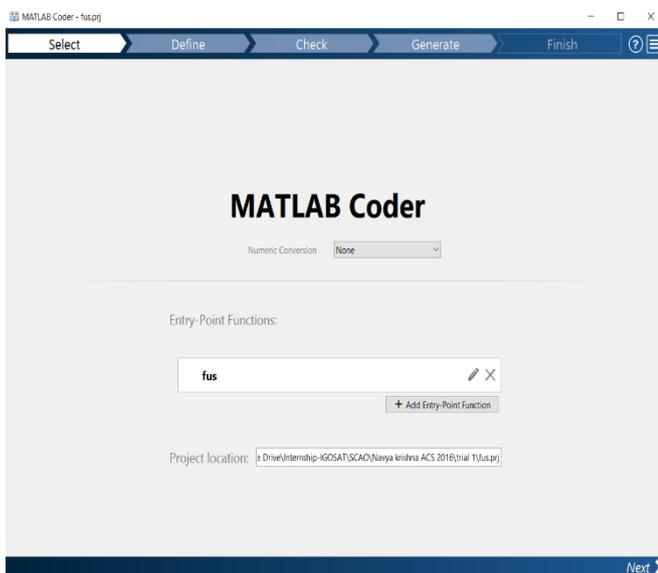
1. Matlab Coder (for the Matlab created functions).
2. Simulink Coder (for Simulink systems).

Our Attitude estimation is a Matlab function file and Simulink coder cannot support standalone code generation. So we use the Matlab Coder for Attitude estimation part and Simulink coder for the Attitude Controller.

Matlab Coder can also be used for the Attitude Controller &, but for combined De-tumble and PID, Simulink Coder is simpler in C++ code generation.

Code generation for Attitude estimation

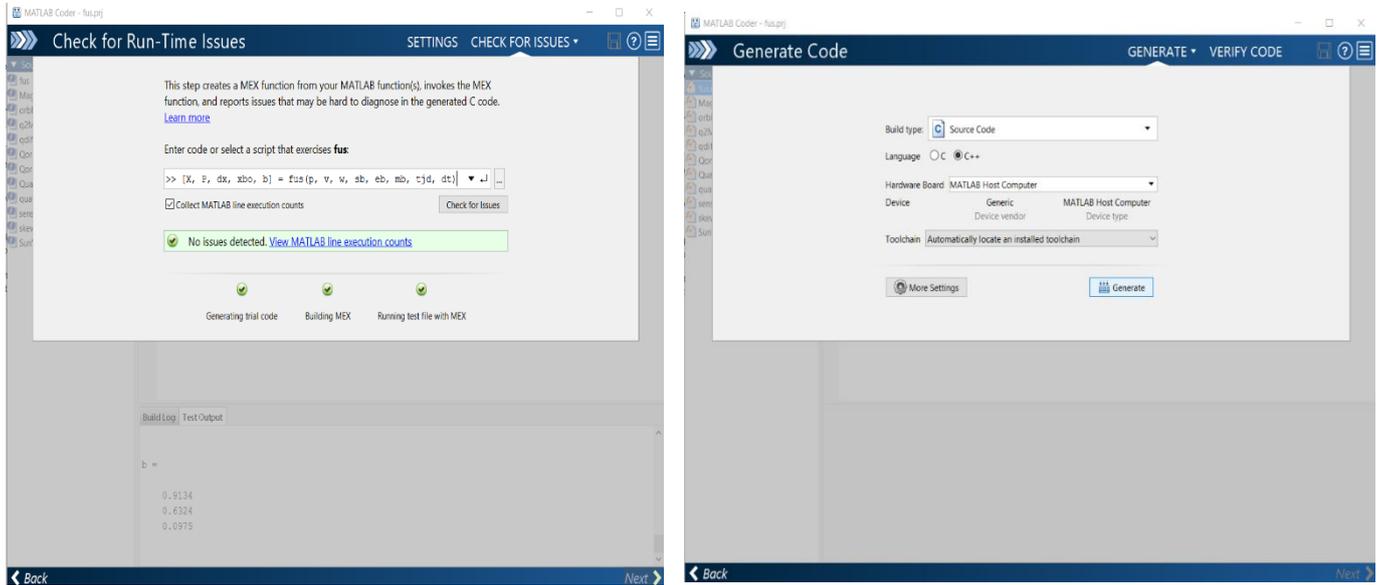
1. Open Matlab Coder from Matlab Apps and select the corresponding Function file.
2. define the input / output parameters and their size & type.



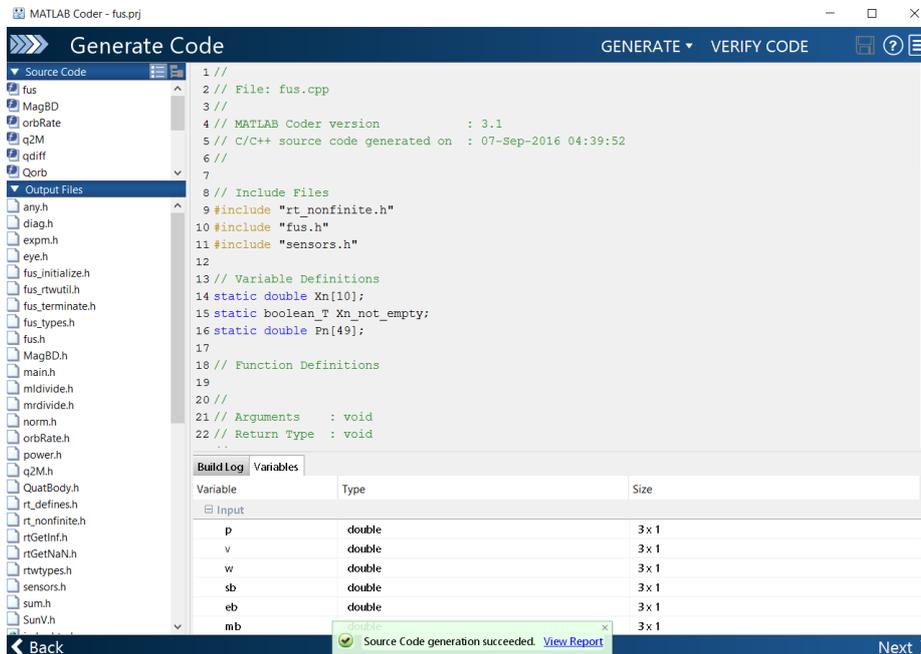
3. Check for the run time coding issues.



4. Select the desired language and type and click on Generate.

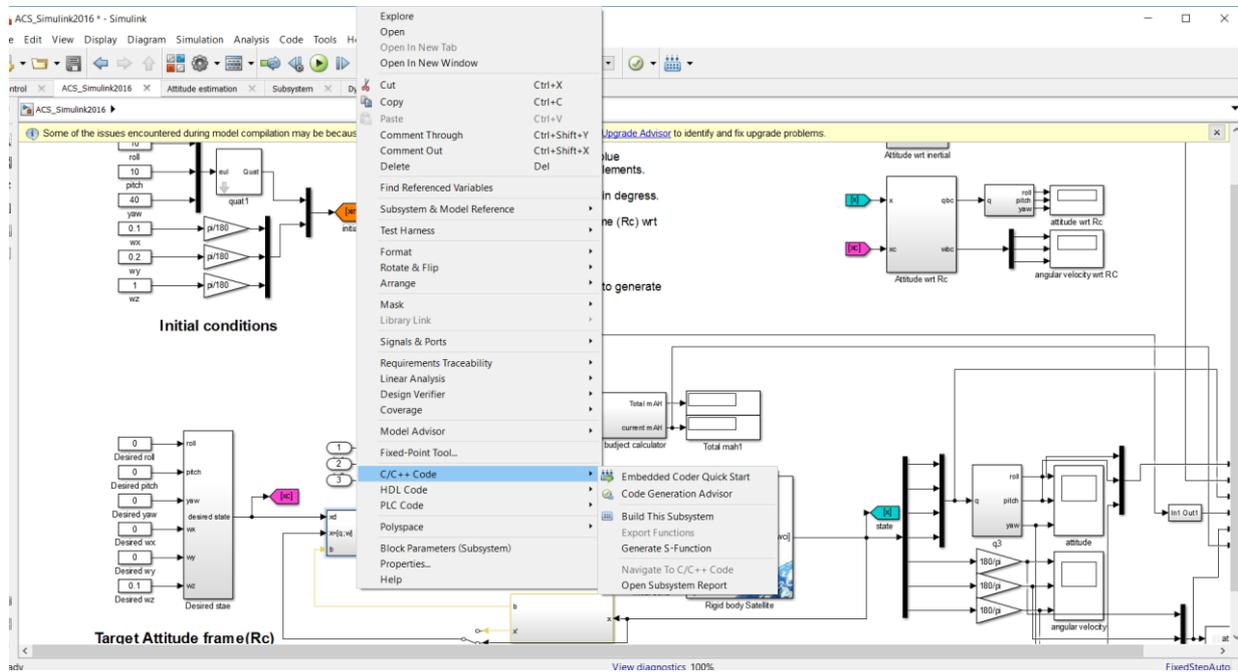


5. Check and verify the generated code, and save the report.



Code generation for Attitude controller

Make sure that, there are no standalone matlab functions in selected Simulink subsystems. Initialiase the Simulink model. Right click on the controller subsystem & select build this system.



11. CONCLUSION

Summing up, an Attitude control loop of IGOSat is designed. A Simulink model of the Attitude Control System is developed. The environmental disturbances, sensors, actuators and power budget are modelled in the simulation. An attitude controller (de-tumbling & PID), for the various operating modes of the IGOSat are designed. An Extended Kalman filter algorithm is developed for fusing the sensors to determine the attitude and integrated to the Simulink. Simulations has been carried out for the Controller design, verifying the performance, power consumption and attitude requirements imposed on the IGOSat. The attitude modes of the IGOSat has been verified.

12. FUTURE WORK

This report completely talks about the Attitude Controller and determination results from the Simulink simulations. For the future interns, there is a part of the work to be done in both Control & determination.

1. The sensors team should provide the measurements at 1 Hz. And the sensors unavailability condition are defined in the software if the sun/earth are not in sensors field of view.
2. The bias and noise of the sensors measurements are to be estimated and stored in Attitude determination algorithm.
3. Interface between the sensors/actuators to the Attitude Control and Determination system are to be defined.
4. The Control software is integrated to AOCS board and tests to be done on the Controller and Determination algorithms on the AOCS test bench (Helmholtz cage...)



13. REFERENCES

- 1] *Revue des Exigences Préliminaires(REP)*. (2015, July 09).
- 2] LabEx UnivEarthS website, <http://www.univearths.fr/fr/le-labex-univearths>. (n.d.).
- 3] APC website, http://www.apc.univ-paris7.fr/APC_CS/en/home. (n.d.).
- 4] IPGP website, <http://www.ipgp.fr/en/institute>. (n.d.).
- 5] *Interface Control Document (ref: SYS-ICD)*, Hillton Tang, May 2016.
- 6] *Specification Technique de Besoin (ref: PLA-STB)*, Hillton Tang , May 2016. (n.d.).
- 7] Gilbert, W. (1958). *On the Magnet (De Magnete)*. The Collector's Series in Science, New York.
- 8] Makovec, K. L. (2001, July 19). *A Nonlinear Magnetic Controller for Three-Axis Stability of Nanosatellites*.
- 9] Chris Hall, R. (2002). *Chapter 6: Spacecraft Attitude Dynamics*.
- 10] Hughes, P. C. (1986). *Spacecraft Attitude Dynamics*, (ISBN 0-471-81842-9). John Wiley and Sons.
- 11] Guerrant, D. V. (2005, June). *Design and Analysis of Fully Magnetic Control for Picosatellite Stabilization*.
- 12] Alvenes, F. (2012). *Satellite Attitude Control System*
- 13] *Cube torquer & coil*
http://www.isispace.nl/brochures/CubeSpace/CubeTorquerCoil/CubeSpace_CubeTorquerCubeCoil_Brochure_2016_01.pdf. (n.d.).
- 14] Leomanni, M. (n.d.). *Comparison of control laws for magnetic detumbling*.
- 15] Valdemir Carrara, H. K. (n.d.). *Attitude Determination, Control and Operating Modes for Conasat Cubesats*.
- 16] Ibrahim, N. S. (2013). *Chapter-6: Attitude and Orbit Control of Small Satellites for Autonomous Terrestrial Target Tracking*
- 17] Nikolas Trawny, S. I. (2005, March). *Indirect Kalman Filter for 3D Attitude Estimation*.
- 18] *Mathematical Modeling of Earth's Magnetic Field*, Jeremy Davis Virginia Tech, Blacksburg, May 12, 2004.
- 19] *Cube torquer & coil*.
http://www.isispace.nl/brochures/CubeSpace/CubeTorquerCoil/CubeSpace_CubeTorquerCubeCoil_Brochure_2016_01.pdf.
- 20] Hubert Halloin, *Note on quaternions and rotations in space Application to dynamics and kinematics*, Feb 2015.
- 21] Gaute Bråthen(January 2013,) *Design of Attitude Control System of a Double CubeSat*.
- 22] Chapter 2, pg 18, "Flexible Spacecraft Dynamics, Control and Guidance", Giovanni Campolo, 2016.
- 23] Peter C. Hughes. *Spacecraft Attitude Dynamics*. John Wiley and Sons, New York, 1986.



24] Geological Survey of Canada. National Geomagnetism Program.

<http://www.geolab.nrcan.gc.ca/geomaq>.

25] Rafał Wisniewski, Satellite Attitude Control Using Only Electromagnetic Actuation, Decemeber,1996.

14. APPENDIX

1. My_library.slx (basic functions on quaternion operations.)

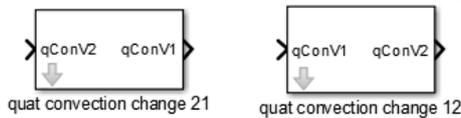
Skew Matrices



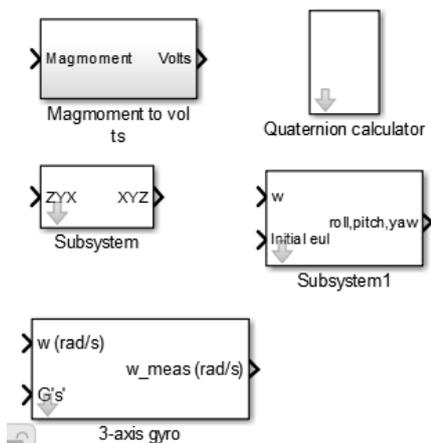
Quaternion normalise



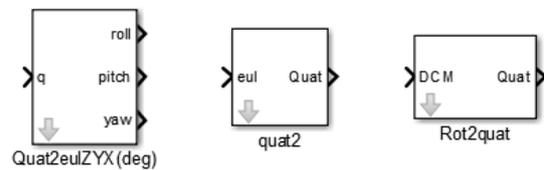
Quaternion Convection change



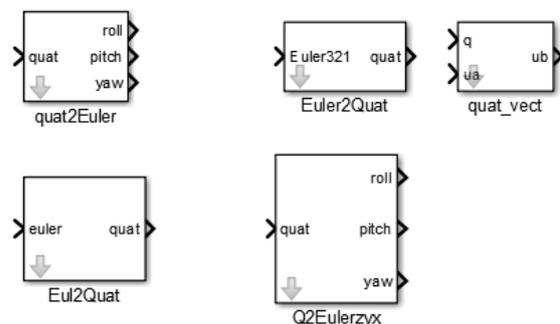
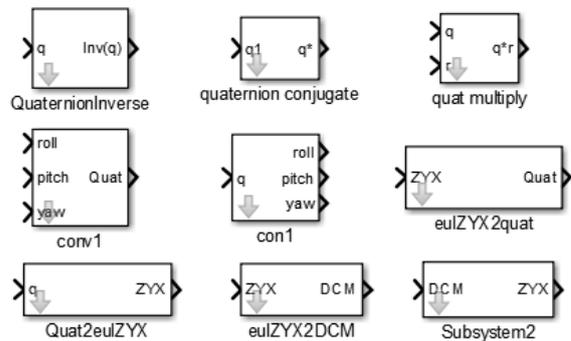
Miscileneous



Axes Transformation



Quaternion Convection 1



2. Stability Analysis

Lyapunov theory can be used to both to test stability of a system, and to arrive at a stabilizing control law. We present the stability approach from [21] & [12]. The Lyapunov stability candidate with the controller can be extended to,



$$V = \frac{1}{2} \omega_{R_{sat}/R_{loc}}^{R_{sat}} I \omega_{R_{sat}/R_{loc}}^{R_{sat}} + \frac{3}{2} \omega_c^2 \hat{\sigma}_3^T I \hat{\sigma}_3 \frac{1}{2} \omega_c^2 (I_y - 3I_z) + k (\varepsilon^T \varepsilon + (1-\eta)^2)$$

Here $k > 0$. Because of the quadratic expression, the extension will be zero for the correct attitude, i.e, for equilibrium state ($\varepsilon = 0$ and $\eta = 1$).

So $\varepsilon^T \varepsilon + \eta^2 = 1$, this means it is possible to rewrite the last term as $2k(1-\eta)$ and with its time derivative $-2k\dot{\eta}$. Thus the new expression for \dot{V} is,

$$\dot{V} = \omega_{R_{sat}/R_{loc}}^{R_{sat}} \left[k\varepsilon + \tau_d^{R_{sat}} \right]$$

Keeping in mind for pitch stability (from [9]) $I_x > I_z$

And the expression for control law is derived as,

$$\tau_d^{R_{sat}} = -p\varepsilon - d\omega_{R_{sat}/R_{loc}}^{R_{sat}}$$

With the gains, $d > 0$, $k > 8\omega_c^2 (I_y - I_z) > 0$

This is the same controller found in chapter[166.3 PID Control].

3. Matlab Script files

a. *Orbitparameters.m* (Simulink Initialization file)

```
clear;
close all; clc;

addpath MySimulinkLib
sblocks;

% Constants
%-----
mu = 3.9860e+05; % gravitation constant (km/sec^3)
eqRad = 6371*1e3; % Mean earth radius
deg2rad = pi/180;
rad2deg = 180/pi;

% Initialise Kepler elements for the orbit
%-----
Altitude = 650.0; % km
eccentricity = 0;
inc_SunSynchronous = 97.78739; % deg
inclinationOrbit = 98.602802; % deg
Omega_init = 0.6631315 ; % rad

kep_init = [ Altitude+(eqRad/1000); % (km)
            inclinationOrbit*deg2rad; % rad
```



```

    Omega_init; % rad
    0.0; % rad
    eccentricity;
    0.0 ]; % rad
    omega_o = sqrt(mu*(1/kep_init(1)))./kep_init(1); % Orbit rate

% Time parameters
%-----
tstart = Date2JD([2018 06 15 0 0 0]); % start time in Julian days
tstep = 10; % sampling time
torbit = 2*pi/(sqrt(mu/abs(kep_init(1))^3)); % in sec
torbit_hr = torbit/3600; % in hrs
tfinal = 5*torbit; % time for 5 orbits(sec)
timeD = tstart + [0:tstep/86400:tfinal/86400]; % Time grid in Julian days
timeS = 0:tstep:tfinal; % Time grid in seconds

% satellite characteristics
%-----
mass=3.77; % in kg
lenth=34.5*10^-2; % in m
width=10*10^-2; % in m
Iz =mass/6*(width^2);
Ix=mass*(lenth^2+width^2)/12;
Iy=Ix;
I=diag([Ix,Iy,Iz]);
Ixy=Ix/Iy;

% Actuator characteristics
%-----
% Actuator
m_max = 0.2; % maximum magnetic moment (A-m^2)
i_max = 250*1e-3; % maximum current (Amps)

% Area of each coil
Ax = 60*10*10^-6; % m^2
Ay = 60*10*10^-6; % m^2
Az = 90^2*10^-6; % m^2

% Resistance in each coil
Rx = 30;
Ry = 30;
Rz = 83;

% residual magnetic moment
%-----
mr = [0.0001; -0.0001; 0.0005];
bavg = 1.0e-05 *[-0.0067;0.3398;0.0344];%1.0e-05*[-0.2507; 0.2889; 0.0701];

% Sampling time
%-----
st=1; % sensor noise
Ts=1; % sampling time of controller & Attitude estimation
ss=1; % Simulation step

% Sensor & Actuator noise

```



```

%-----
mn = 1e-7; % Magnetometer noise (Tesla)
sn = 1e-8; % sun sensor noise
en = 1e-8 ; % earth sensor noise
gn = 1e-3*[0.001 0.01 0.001]; % gyro noise (rad/sec)
mtn=1e-9; % magnetorquer noise

```

b. *sensors.m* (Attitude determination code..)

```

function [X, P, dx, xbo, b] = sensors(Xo, Pin, p, v, w, sb, eb, mb, tjd, dt)
%-----Attitude estimation by kalman.....%
% state vector Xo =[q; quaternion
%                   w; Angular velocity
%                   bg; gyro bias
%                   bm;] magnetometer bias
%-----%
% Inputs: Xo, P - Previous state & covariance
%         I - Inertia
%         p - position in ECI
%         v - Velocity (km/sec)
%         w - gyro reading
%         ts - gyro sample time
%         sb - sun vector in body
%         eb - earth vector in body
%         mb - magnetic field vectr in body
%         tme - time in julian
%-----&
% Outputs: X =[q; quaternion
%             w; Angular velocity
%             bg; gyro bias
%             bm;] magnetometer bias
%           P = covariance

X = [zeros(3,1);1;zeros(6,1)];
P = eye(10);
xbo = zeros(7,1);

% time to julian
%-----%
tjd = Dt2jD(tme) ;
tjd = tstart+tme/86400;

% Eccentricity and orbit rate
%-----%
[ecc, worb] = orbRate(p,v);
wor = [0;-worb;0];

%-----%
%-----Attitude Estimation Starts-----%
%-----%
Q = zeros(10);
R = 1e-10*eye(9);

%-----Prediction-----%

```



```

qo = Xo(1:4); wo = Xo(5:7); bg = Xo(8:10); %bm = Xo(11:13);

qhat = qo+0.5*dt*[-skew(w-bg) w-bg;-(w-bg)' 0]*qo;
what = (w-bg);
bghat = bg;

Xo = [qhat; what; bghat];

A11 = eye(4)+0.5*[-skew(what) what;-(what)' 0];
A12 = 0.5*[qhat(4)*eye(3)+skew(qhat(1:3));-qhat(1:3)'];
A21 = zeros(3,4);
A22 = eye(3);
F = [A11*dt A12*dt zeros(4,3);
     A21 A22 zeros(3,3);
     zeros(3,7) eye(3)];
P = F*Pin*F' + Q;
%-----%
%-----Update-----%
%-----%
Mi2b = q2M(qhat);
qm = [qhat(4)*eye(3)+skew(qhat(1:3));-qhat(1:3)'];

% vector models
siu = SunV(p, tjd);
si = siu/sqrt(sum(siu.^2));
mi = MagBD(p, tjd);
ei = -(p/sqrt(sum(p.^2)));

yhat = [Mi2b * si
        Mi2b * mi;
        Mi2b * ei];

y = [sb;
     mb;
     eb];

hss = qm*[skew(si) si; -si' 0];
hmg = qm*[skew(mi) mi; -mi' 0];
hes = qm*[skew(ei) ei; -ei' 0];
H1 = [hss;hmg;hes];
H = [H1 zeros(9,6)];
del=y-yhat;

S = H*P*H'+R;
K = P*H'*inv(S);
P = P - K*H*P;
X = Xo+K*(del);
dx =X-Xo;

%-----Complete-----%
qbo = QuatBody(X(1:4), p, v);
mbo = q2M(qbo);
woi = mbo*wor;
wbo = X(5:7) - woi;

```



```
xbo = [qbo;wbo];
X = X(:);
xbo = xbo(:);

if mb == 0
    b= Mi2b * mi;
end

end
```

c. *Aerodynamic_torque.m*

```
function tau_a = Adrag(x, omega_o, kep_init, I)
%#codegen

% Aerodynamic drag model

tau_a = zeros(3,1);

m=3.77;
jx = 0.01; % X-axis length
jy = 0.01; % Y-axis length
jz = 0.03; % Z-axis length
Jx = (m/12)*(jy^2+jz^2); % X-axis inertia
Jy = (m/12)*(jx^2+jz^2); % Y-axis inertia
Jz = (m/12)*(jx^2+jy^2); % Z-axis inertia
J = [Jx 0 0;0 Jy 0;0 0 Jz]; % Inertia matrix

q_B_O = [x(4);x(1:3)]';

dcm = zeros(3,3,size(q_B_O,1));
dcm(1,1) = q_B_O(1)^2 + q_B_O(2)^2 - q_B_O(3)^2 - q_B_O(4)^2;
dcm(1,2) = 2*(q_B_O(2)*q_B_O(3) + q_B_O(1)*q_B_O(4));
dcm(1,3) = 2*(q_B_O(2)*q_B_O(4) - q_B_O(1)*q_B_O(3));
dcm(2,1) = 2*(q_B_O(2)*q_B_O(3) - q_B_O(1)*q_B_O(4));
dcm(2,2) = q_B_O(1)^2 - q_B_O(2)^2 + q_B_O(3)^2 - q_B_O(4)^2;
dcm(2,3) = 2*(q_B_O(3)*q_B_O(4) + q_B_O(1)*q_B_O(2));
dcm(3,1) = 2*(q_B_O(2)*q_B_O(4) + q_B_O(1)*q_B_O(3));
dcm(3,2) = 2*(q_B_O(3)*q_B_O(4) - q_B_O(1)*q_B_O(2));
dcm(3,3) = q_B_O(1)^2 - q_B_O(2)^2 - q_B_O(3)^2 + q_B_O(4)^2;

R_B_O = dcm;
w_B_OB =x(5:7);

x = 0.1;
y = 0.1;
z = 0.2;

A_drag = z*sqrt(x^2 + y^2); % Maximum area for calc disturbance drags

vel = kep_init(1)*1e3*omega_o; % Orbit velocity
rho_a = 4.89e-13;
```



```

c_p = [0.01 0.01 0.02]';

c_p_x = [0 -c_p(3) c_p(2);
         c_p(3) 0 -c_p(1);
         -c_p(2) c_p(1) 0];

%Vr = R_B_O*[-1 0 0]';

Vr = R_B_O*[0 -1 0]';
Vr_x = [0 -Vr(3) Vr(2);
        Vr(3) 0 -Vr(1);
        -Vr(2) Vr(1) 0];

tau_a = rho_a*vel*( vel*A_drag*c_p_x*Vr-(I + Vr_x*J)*w_B_OB );
tau_a = tau_a(:);

```

d. Control

1. De-tumble

```

function m_B = Detumble(x,B_b,t,m_max, I)
%#codegen

persistent bp tp
coder.extrinsic('norm')
m_B=zeros(3,1);
w_bo = x(5:7);

% Controller gain
kd = 2e-2*I;
k =kd/(3.4160e-06^2);

if isempty(bp)
    tp = 0; bp = zeros(3,1);
    bdot = B_b;
else
    bdot = (B_b - bp)/(t - tp);
end
tp = t;
bp = B_b;

% % Bdot
% m_B = -k*bdot;
%
% % B bang bang
% sbt = bdot./abs(bdot);
% m_B = -m_max*sbt;

% Desired torque controller
m_B = (-1/norm(B_b)^2)*(kd*cross(B_b,w_bo));

```

2. PID Control

```

function [qe,m_B] = cnt1(xd,x,B_b,t,I,)
%#codegen

```



```

coder.extrinsic('norm')
persistent epso tm

% Initialisation
%-----
m_B = zeros(3,1);
qe = zeros(4,1);
B_b = B_b(:);
dd = zeros(3,1);
bm = zeros(3,1);

if isempty(epso)
    epso = zeros(3,1);tm = 0;

end
dt = t -tm;
epso= epso + eps;
epsi = epso*dt;
tm = t;

qd = xd(1:4);
qs = x(1:4);
qe =[qd(4) qd(3) -qd(2) -qd(1);
    -qd(3) qd(4) qd(1) -qd(2);
    qd(2) -qd(1) qd(4) -qd(3);
    qd(1) qd(2) qd(3) qd(4)]*qs;

wd = xd(5:7);
ws = x(5:7);
we = ws-wd;
eps = qe(1:3);

% Controller gains
%-----
d = 2*1e-2;
p = 9*1e-6;
i = 2e-10;

Kd = d*I;
Kp = p*I;
Ki = i*I;

tau_B = kp*eps + ki*epsi + kd*we;

% Moment set up by controller
m_B = (-1/((norm(B_b)) ^2)) *cross(B_b,tau_B);
m_B = m_B(:);

```

e. **toVTS.m** (generates VTS data from Matlab simulations)

```

%%%%.....VTS file generation.....%%

```



% Note:

% 1) Get the position & quaternion data from simulink
% 2) specify Vts directory
% 3) Run the code
% 4) select pos.txt & quat.txt in VTS interface

```
ps = pos; % position from simulink
at=aty; %quaternion from simulink
tms = tme; % time from simulink
```

```
VtsDir = 'C:\Users\Navya\Downloads\Vts-WindowsNT-32bits-2.7';
MatlabDir = 'C:\Users\Navya\Google Drive\Internship-IGOSAT\SCAO\Navya krishna
ACS 2016\trial 1';
```

% Position generation

```
header1='CIC_OEM_VERS = 2.0';
header2='COMMENT Generated by Spacebel CCSDS library';
header3='CREATION_DATE = 2016-06-14T14:23:59.015921';
header4='ORIGINATOR = VTS Propagated Orbit';
header5='META_START';
header6='OBJECT_NAME = IGOSAT';
header7='OBJECT_ID = IGOSAT';
header8='CENTER_NAME = EARTH';
header9='REF_FRAME = EME2000';
header10='TIME_SYSTEM = UTC';
header11='META_STOP';
```

```
p = [58284*ones(size(ps,1),1) tms ps];
```

% Quaternion generation

```
header21='CIC_AEM_VERS = 1.0';
header22='COMMENT Generated by Spacebel CCSDS library';
header23='CREATION_DATE = 2016-06-14T14:23:59.015921';
header24='ORIGINATOR = VTS Propagated Orbit';
header25='META_START';
header26='COMMENT Exemple de fichier de attitude';
header27='OBJECT_NAME = IGOSAT';
header28='OBJECT_ID = IGOSAT';
header29='REF_FRAME_A = EME2000';
header30='REF_FRAME_B = SC_BODY_1';
header31='ATTITUDE_DIR = A2B';
header32='TIME_SYSTEM = UTC';
header33='ATTITUDE_TYPE = QUATERNION';
header34='META_STOP';
```

```
q = [58284*ones(size(at,1),1) tms at];
```

```
cd(VtsDir);
```

```
fid=fopen('pos.txt','w');
fprintf(fid, [header1 '\n' header2 '\n' header3 '\n' header4 '\n\n' header5
'\n' header6 '\n' header7 '\n' header8 '\n' header9 '\n' header10 '\n\n'
header11 '\n\n']);
fprintf(fid, '%5.0f %6.2f %f %f %f \n', [p]);
fclose(fid);
```



```
fid1=fopen('quat.txt','w');  
fprintf(fid1, [ header21 '\n' header22 '\n' header23 '\n' header24 '\n\n'  
header25 '\n\n' header26 '\n\n' header27 '\n' header28 '\n\n' header29 '\n'  
header30 '\n' header31 '\n\n' header32 '\n\n' header33 '\n\n' header34 '\n\n']);  
fprintf(fid1, '%5.0f %6.2f %f %f %f %f \n', [q]');  
fclose(fid1);
```

```
cd(MatlabDir);
```

f. **magnetigrf.m(Computes magnetic field using IGRF 2015 field)**

```
function b = magnetigrf(pos, jD, Coef)  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
    %  
% Computes Magnetic field in Inertial frame  
% Inputs:  
%     pos -Position in ECI (Kms)  
%     jD  -Julian date  
%     Coef- IGRF & Schmidt coefficients  
% Outputs:  
%     b   -Magnetic field in ECI (tesla)  
%  
% References: Mathematical Modeling of Earth's Magnetic  
%             Field,  
%             Jeremy Davis Virginia Tech, Blacksburg, May 12,  
%             2004.  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
    %  
J2000_days = jD - 2451545;  
omega_earth = 7.292115855377074e-005; % (rad/sec)  
Go = 1.727564365843028; % (rad)  
    http://www.amsat.org/amsat/articles/g3ruh/106.html  
GMST = Go + omega_earth*86400*(J2000_days + 0.5);  
GMST = GMST - 2*pi*floor(GMST/(2*pi));  
sGAST = sin( GMST );  
cGAST = cos( GMST );  
cECIToEF = [cGAST, sGAST, 0; -sGAST, cGAST, 0; 0, 0, 1];  
  
rEF = cECIToEF*pos;  
  
r = sqrt( rEF(1).^2 + rEF(2).^2 + rEF(3).^2 );  
phi = atan2( rEF(2), rEF(1) )*180/pi;  
theta = acos( rEF(3)./r )*180/pi;  
days=J2000_days;
```



```

if nargin<3
    load('igrf2015.mat', 'igrfSg');
    load('igrf2015.mat', 'igrfSh');
else
    load(coef);
end

[Br,Bt,Bp] = magnet(r,90-theta,phi,days,igrfSg,igrfSh);

% Spherical to NED
e = 0.0*pi/180; % the term used to correct for the oblateness
of the
X = -Bt*cos(e) - Br*sin(e);
Y = Bp;
Z = Bt*sin(e) - Br*cos(e);

b = [X;Y;Z]*10^-9;%, [Br,Bt,Bp]';

iR      = rEF/sqrt(sum(rEF.^2));
iPh     = cross(iR,[0;0;-1]);
iPhi    = iPh/sqrt(sum(iPh.^2));
iThet   = cross(iPhi,iR);
iTheta  = iThet/sqrt(sum(iThet.^2));
cRPT2Cart = [-iTheta iPhi -iR ];

    b = cRPT2Cart * b;
    b = cECIToEF' * b;

function [Br,Bt,Bp] = magnet(r,theta,phi,days,igrfSg,igrfSh)
% Inputs
% r      Geocentric radius
% theta  Latitude measured in degrees positive from
equator
% phi    Longitude measured in degrees positive east from
Greenwich
% days   Decimal days since January 1, 2000
%
% Outputs - magnetic field strength in local tangential
coordinates
% Br     B in radial direction
% Bt     B in theta direction
% Bp     B in phi direction

```



```

% Checks to see if located at either pole to avoid
% singularities
if (theta>-0.00000001 && theta<0.00000001)
theta=0.00000001;
elseif(theta<180.00000001 && theta>179.99999999)
theta=179.99999999;
end
% The angles must be converted from degrees into radians
theta=(90-theta)*pi/180;
phi = phi*pi/180;
a=6371.2; % Reference radius used in IGRF
% This section of the code simply reads in the g and h
% Schmidt
% quasi-normalized coefficients
gn = igrfSg(:,1);gm =igrfSg(:,2); gvali= igrfSg(:,3); gsvi=
    igrfSg(:,4);
hn = igrfSh(:,1); hm= igrfSh(:,2); hvali= igrfSh(:,3); hsvi=
    igrfSh(:,4);
N=max(gn);
g=zeros(N,N+1);
h=zeros(N,N+1);
for x=1:length(gn)
g(gn(x),gm(x)+1) = gvali(x) + gsvi(x)*days/365;
h(hn(x),hm(x)+1) = hvali(x) + hsvi(x)*days/365;
end
% Initialize each of the variables
% Br      B in the radial direction
% Bt      B in the theta direction
% Bp      B in the phi direction
% P       The associated Legendre polynomial evaluated at
%         cos(theta)
%         The nomenclature for the recursive values
%         generally follows
%         the form P10 = P(n-1,m-0)
% dP      The partial derivative of P with respect to theta
Br=0; Bt=0; Bp=0;
P11=1; P10=P11;
dP11=0; dP10=dP11;
for m=0:N
for n=1:N
if m<=n
% Calculate Legendre polynomials and derivatives recursively
if n==m

```



```

P2 = sin(theta)*P11;
dP2 = sin(theta)*dP11 + cos(theta)*P11;
P11=P2; P10=P11; P20=0;
dP11=dP2; dP10=dP11; dP20=0;
elseif n==1
P2 = cos(theta)*P10;
dP2 = cos(theta)*dP10 - sin(theta)*P10;
P20=P10; P10=P2;
dP20=dP10; dP10=dP2;
else
K = ((n-1)^2-m^2)/((2*n-1)*(2*n-3));
P2 = cos(theta)*P10 - K*P20;
dP2 = cos(theta)*dP10 - sin(theta)*P10 - K*dP20;
P20=P10; P10=P2;
dP20=dP10; dP10=dP2;
end
% Calculate Br, Bt, and Bp
Br = Br + (a/r)^(n+2)*(n+1)*...
((g(n,m+1)*cos(m*phi) + h(n,m+1)*sin(m*phi))*P2);
Bt = Bt + (a/r)^(n+2)*...
((g(n,m+1)*cos(m*phi) + h(n,m+1)*sin(m*phi))*dP2);
Bp = Bp + (a/r)^(n+2)*...
(m*(-g(n,m+1)*sin(m*phi) + h(n,m+1)*cos(m*phi))* P2);
end
end
end
% Br=Br;
Bt=-Bt;
Bp=-Bp/sin(theta);
end
end

```

g. **SunV.m** (Computes satellite to sun vector in inertial frame)

```

function [u] = SunV( rSc, jD )
%-----
% Generate the sun vector in the earth-centered inertial
% frame.
%-----
% Inputs
% -----
% jD      (:)      Julian date of UTC time
% rSc     (3,:)    Spacecraft vector in the ECI frame (km)

```



```

% Outputs
% -----
% u      (3,:)   Unit sun vector in ECI frame
%
% References: The 1993 Astronomical Almanac, p. C24.

% Degrees to radians
%-----
d2r = pi / 180;

% Days from J2000.0
%-----
n = jD - 2451545.0;

% Mean anomaly
%-----
g = 357.528 + 0.9856003*n;

% Ecliptic longitude
%-----
lam = rem(280.460 + 0.9856474*n,360) + 1.915*sin(g*d2r) +
    0.02*sin(2*g*d2r);
lam = d2r*lam;

% Obliquity of ecliptic
%-----
obOfE = 23.439 - 4.00e-7*n;
obOfE = d2r*obOfE;

% Equatorial rectangular coordinates of the Sun
%-----
sLam = sin(lam);
u = [cos(lam); cos(obOfE).*sLam; sin(obOfE).*sLam];
r = (1.0014 - 0.01671*cos(g*d2r) -
    0.00014*cos(2*g*d2r))*149600e3;

% Account for parallax
%-----
u = [r.*u(1,:) - rSc(1,:);...
    r.*u(2,:) - rSc(2,:);...
    r.*u(3,:) - rSc(3,:)];
u = u/sqrt(sum(u.^2));
end

```



4. Julian Day

Given year, month, day, hour, minute, second, compute the Juliadate, JD.

$$JD = 367 \text{ year} - INT \left\{ 7 \left[\frac{\text{year} + INT \left(\frac{\text{month} + 9}{12} \right)}{4} \right] \right\} + INT \left(\frac{275 \text{ month}}{9} \right) + \text{day} \\ + 1,720,996.5 + \frac{\text{hour}}{24} + \frac{\text{minute}}{1440} + \frac{\text{seconds}}{86400}$$

5. Sun Vector ephemeris

Reference: The 1993 Astronomical Almanac, p. C24.

The following formulas give the apparent coordinates of the Sun to a precision of $0^\circ.01$ and the of time to a precision of $0^m.1$ between 1950 and 2050; on this page the time argument n is the number of days from J2000.0.

$$JD = \text{Julian date of time in UTC} \\ n = JD - 2451545.0$$

Mean longitude of Sun, corrected for aberration: $L = 280^\circ.460 + 0^\circ.9856474 n$

Mean anomaly: $g = 357^\circ.528 + 0^\circ.9856003 n$

Put L and g in the range 0° to 360° by adding multiples of 360° .

Ecliptic longitude: $\lambda = L + 1^\circ.915 \sin g + 0^\circ.020 \sin 2g$

Ecliptic latitude: $\beta = 0^\circ$

Obliquity of ecliptic: $\varepsilon = 23^\circ.439 - 0^\circ.0000004n$

Right ascension (in same quadrant as λ): $\alpha = \tan^{-1}(\cos \varepsilon \tan \lambda)$

Alternatively, α may be calculated directly from

$$\alpha = \lambda - ft \sin 2\lambda + (f/2)t^2 \sin 4\lambda \\ \text{where } f = 180/\pi \text{ and } t = \tan^2 \varepsilon / 2$$

Declination: $\delta = \sin^{-1}(\sin \varepsilon \sin \lambda)$

Distance of Sun from Earth, in au: $R = 1.00014 - 0.01671 \cos g - 0.00014 \cos 2g$

Equatorial rectangular coordinates of the Sun, in au:

$$x = R \cos \lambda, \quad y = R \cos \varepsilon \sin \lambda, \quad z = R \sin \varepsilon \sin \lambda$$

Equation of time (apparent time minus mean time):



E , in minutes of time = $(L - \alpha)$, in degrees, multiplied by 4.

Horizontal parallax: $0^{\circ}.0024$

Semidiameter: $0^{\circ}.2666 / R$

Light time: $0^d.0058$

6. Quaternion first order Integration[17]

The first order quaternion integrator makes the assumption of a linear evolution of ω during the integration interval Δt . For that purpose, the average turn rate $\bar{\omega}$, defines as

$$\bar{\omega} = \frac{\omega(t_{k+1}) + \omega(t_k)}{2}$$

By Taylor series,

$$q(t_{k+1}) = q(t_k) + \dot{q}(t_k)\Delta t + \frac{1}{2}\ddot{q}(t_k)\Delta t^2 + \dots$$

Recognizing the first term as the Taylor series expansion of the matrix exponential, we obtain,

$$q(t_{k+1}) = \left(\exp\left(\frac{1}{2}\Omega(\bar{\omega})\Delta t\right) + \frac{1}{48}\left(\Omega(\omega(t_{k+1}))\Omega(\omega(t_k)) - \Omega(\omega(t_k))\Omega(\omega(t_{k+1}))\right)\Delta t^2 \right) q(t_k)$$