Closed Loop Simulation for Attitude Control of Nano-Satellite

Riya[1], Sanket Chirame[2], Anant A. Joshi[3], Ram Milan Kumar Verma[4], Sumit Agrawal[5], Ravit Anand[6], Shrita Singh[7], Sanskriti Bhansali[8], Piyush Jirwankar[9], Niket Parikh[10]

[1-10]Student Satellite Lab, IIT Bombay, Mumbai, Maharashtra, India, 400076

ABSTRACT

This document describes the implementation of a state estimator, and an attitude controller for Advitiy, IIT Bombay's 2nd Student Satellite. The algorithms used by the estimator and the controller are described. A closed loop simulation (CLS) framework for the testing of these algorithms is explained. A detailed description of each of the blocks in CLS framework has been provided. The problems faced due to actuation with magnetorquer are described. It also contains the description of practices followed to ensure the quality (correctness of results obtained and repeatability of the process).

Keywords: Closed Loop Simulation, Quality Assurance, Estimator, Controller

1. INTRODUCTION

The requirements of attitude control and the challenges involved in realizing these in the case of nano-satellites in Low Earth Orbit (LEO) are unique. The IIT Bombay Student Satellite, Advitiy, belongs to this class of satellites, having a mass of less than 880 gram and orbiting in a LEO of approximate altitude 600 km. The Attitude Determination and Controls Subsystem of Advitiy aims to maintain Earth pointing orientation of satellite within an accuracy of ±10° because signals are received with detectable strength at minimum expense of power for transmission in earth pointing orientation.

The closed loop simulation framework is developed for the analysis and on ground-verification of chosen estimators and controllers. The estimation and control accuracy of the

[+]Further author information:

Riya: E-mail:160100097@iitb.ac.in

Student Satellite Lab - https://www.aero.iitb.ac.in/satlab/

Aerospace Department, IIT Bombay, Powai, Mumbai, India – 400076

system is monitored in this framework and necessary design changes are made based on the results. A simulation framework for modeling attitude dynamics of satellite typically comprises of different blocks as shown in Fig. 1. The rest of article is structured as follows. In section 2, we introduce environment models used for simulation of environmental parameters. Section 3, 4 and 5 contain description of sensor models, propagator and scientific models respectively. In section 6, we describe the chosen estimator algorithm. Section 7 discusses two control modes and analysis of candidate control algorithms. Section 8 contains details of actuator modeling. Section 9 describes the attitude and angular velocity propagation step. Section 10 shows the simulation result we have obtained using the framework. Section 11 summarizes the guidelines followed to ensure the quality of framework and the correctness of results.

2. ENVIRONMENT MODEL

The environment model determines the external environmental parameters as a function of the state of satellite at some particular time. All the parameters that are modeled are explained below.

- Simplified Gravitational Perturbation model (SGP4)[1]: The SGP4 model is a standard gravity model for earth. It is a sophisticated model which iteratively calculates earth's gravity at any point by including effects like J2, J3, secular effects of gravitation, periodic effects of gravitation, short term perturbations. The inputs to this model are orbital elements of the satellite orbit (sun synchronous, 500 to 800 km altitude) in the form of a Two Line Element (TLE). The output is the position and velocity vector in the Earth Centered Inertial Frame (ECIF).

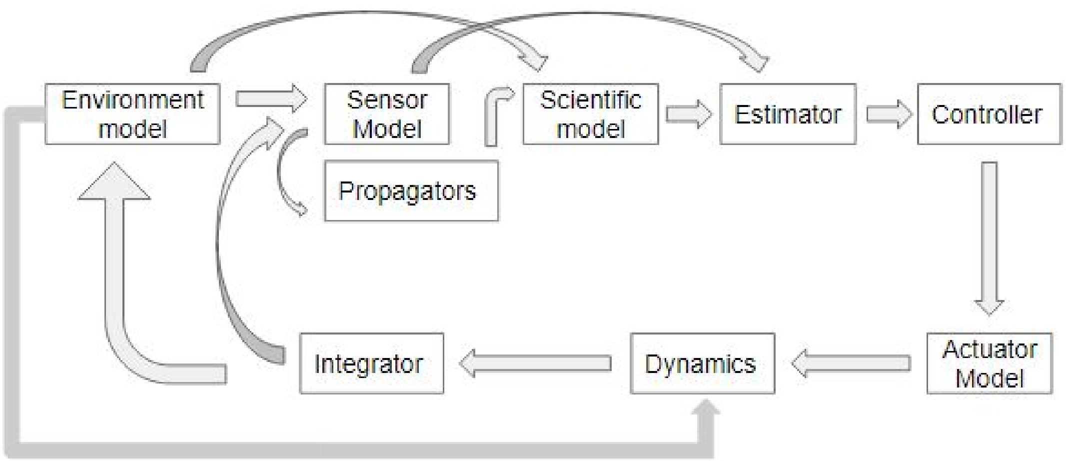

Figure 1: Block Diagram of CLS framework

- Sun model[2]: The sun model takes the time elapsed (from the latest vernal equinox) as its input and gives the sun vector in ECIF as the output.
- Light model: The light model takes the position of the satellite as its input and tells us whether the satellite is in the light region or in the eclipsed region (shadow of the earth on the satellite as shown in Fig. 2). The satellite is assumed to be a point mass while earth and the sun are assumed to be spheres of finite radius. When the satellite is in either umbra or penumbra, it is assumed to be in eclipse region. Otherwise it is in the light region.

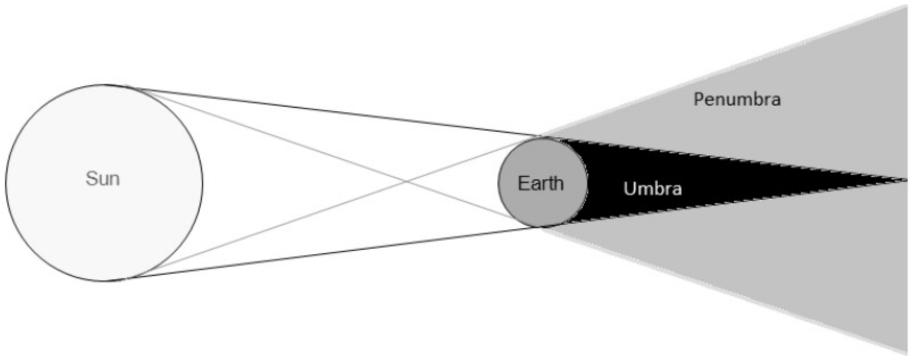

Figure 2: Shadow of earth due to sun

- Magnetic Field model: This block takes position of the satellite in the ECIF as input and calculates the geomagnetic field vector in the ECIF. The International Geomagnetic Reference Field 2011 (IGRF -11)[3] model is used. It models the earth's magnetism as a magnet with the desired order of polarity (dipole, quadrapole, octapole etc.). For ground simulation, a 12th order model[4] is used.

  IGRF library[5] requires latitude, longitude and altitude as input and generates magnetic field vector in North East Down (NED) frame. We perform intermediate frame transformation to obtain desired results.
- Disturbance model[6,7]: The satellite, in its orbit, is acted upon by various forces applied by external environmental entities. These forces result in torques which cause the satellite to go out of the desired attitude. Thus, it is essential that we consider these effects during simulation. Based on comparison of order of magnitudes of torques, it has been concluded that following three torques have critical effect on a nano-satellite like Advitiy.
  - Aerodynamic Torque: For a satellite in low earth orbit, the major disturbance torque is exerted by aerodynamic drag. The drag arises from friction between satellite and atmosphere and is in a direction opposite to that of satellite's velocity vector. Its order of magnitude is 10^{-9} Nm.

- Solar Radiation Torque: Radiation hitting the surface of the satellite has momentum associated with it and thus due to absorption and reflection of radiation, forces act on satellite which generates a torque around the centre of mass. The majority solar radiation experienced by the satellite is due to radiation emitted from the sun. Thus it does not act in eclipse region. Its order of magnitude is 10^{-10} Nm.
- Gravity Gradient Torque: An extended object in a non uniform gravitational field experiences a torque in general due to different forces acting at different parts of the object. Its order of magnitude is 10^{-11} Nm.

3. SENSOR MODEL

Following sensors are going to be used in Advitiy, so they are modeled to be included in simulations.

- **GPS model:** Position and velocity inputs from SGP4 are taken and noise and bias is added to it to generate output of GPS.
- **Gyroscope model:** Angular velocity of body frame with respect to orbit frame (earth-pointing frame) expressed in body frame is given as input to it and it converts it into angular velocity of body frame with respect to orbit frame expressed in body frame and adds noises and bias to it.
- **Magnetometer model:** It takes magnetic field data from IGRF in ECIF and converts it into magnetic field in body frame and adds noises and bias to it.
- **Sun-sensor model:** It takes sun vector in ECIF from sun-model converts it into sun-vector in body frame. Since, in the sun sensors, output is current value which is processed using Analog to Digital Convertor (ADC), quantization due to ADC is introduced into the model along with introduction of noise. Comparing the generated sunsensor reading with a cutoff value, it is tested whether satellite is in eclipse or not. The output of block is sunvector in body frame or null values depending on satellite's presence in eclipse.

4. PROPAGATOR

Since GPS can't be operated continuously on satellite due to power constraints, J2 propagator[8] will be used to determine position and velocity of the satellite during the duration for which GPS is off. So, J2 propagator is modeled in the simulation too.

It uses the SGP output as an initial condition and numerically solves the differential equation to obtain the position and velocity of the satellite. We use RK-4 method[9] for numerical integration.

5. SCIETIFIC MODEL

They are used to generate sun vector and magnetic field vector in orbit frame. They take sun vector and magnetic field vector in ECIF and convert them in orbit frame using quaternion corresponding to rotation matrix from ECIF to orbit frame[10]. The rotation matrix is generated by position and velocity obtained from propagator.

6. ESTIMATOR

It is not possible to obtain a direct measurement of the attitude of the satellite. It has to be calculated using the measurements from other sensors like coarse sun sensor, magnetometer, etc. These sensors have their inherent electronic noise. This makes deterministic attitude calculation impossible. Thus attitude has to be estimated using some estimation techniques. The MEKF is the primary attitude estimator in Advitiy which adresses the non-linear nature of system and constraints associated with the quaternion framework. QuEst will be used for providing an initial estimate to the filter.

- **QuEst**: Kalman filter requires an initial estimate of attitude. Quaternion Estimate (QuEst) algorithm will be used for obtaining this initial attitude estimate between orbit and body frame. QuEst algorithm uses two of non-parallel vectors in body frame and corresponding two vectors in orbit frame. Attitude is estimated by minimizing the Wahba's loss function[10]. In our case, QuEst will use the following two vectors:
    - Sun vector: The body frame components will be measured by the on-board coarse sun sensor. The orbit frame sun-vector will be computed using the on-board mathematical sun model. In CLS, these vectors will be obtained using sun sensor model and scientific model respectively.
    - Magnetic field vector: The body frame components are measured using on-board three axis magnetometer. The orbit frame components will be computed using the onboard IGRF magnetic field model. In CLS, these vectors will be obtained using magnetometer model and scientific model respectively.

  Due to requirement of sun-vector, QuEst will be taken only when the satellite is not in the eclipse region.

- **Multiplicative Extended Kalman Filter**[10]**:** Despite being a relatively complex and computationally demanding, we are implementing it because of following reasons:
  - Unlike QuEst, Kalman Filtering is a dynamic attitude estimator. It incorporates into it the past measurements made in time and hence is a better method.
  - We are using quaternions to represent the attitude. In EKF, error in quaternion is defined as algebraic difference between the true quaternion and its estimate. If we are to keep the estimate unbiased, the unit norm constraint of quaternion is violated. So instead MEKF is used where the true quaternion is the product of an error quaternion and the estimate. Here, product of two unit quaternions is a unit quaternion resolving the issue.
  - A typical EKF uses a dynamic model of the system in the propagation step. However, we are using gyro measurements instead for propagation. This has the advantage of avoiding inherent difficulties in modeling torques acting on a spacecraft. Also gyro measurements are usually more accurate than the models. All gyros have inherent variable bias which could cause inaccuracies in the propagated model. So the rate gyro bias is included in the state vector along with the quaternion and is used to correct the raw gyro measurements.

Figure 3 contains the flowchart which depicts the algorithm of MEKF. State vector of our estimator is taken to be change in angle corresponding to attitude change and change in angular rates bias. The initial estimate (i.e. of state vector and corresponding covariance matrix) for the filter is fed through QuEst which uses measurements from coarse sun sensor and magnetometer for the task. Angular rates from gyroscope readings are used to propagate

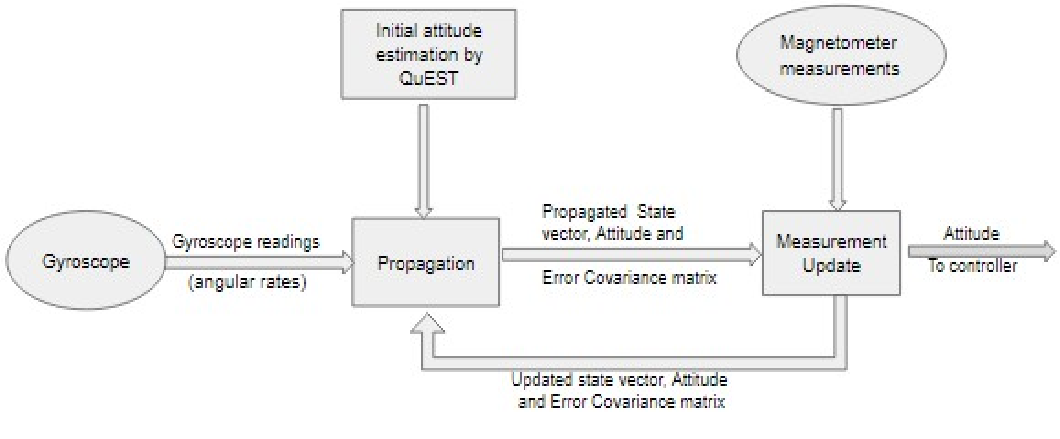

Figure 3: Schematic of MEKF operation

previous estimates of state vector and corresponding covariance matrix. The propagated state vector and corresponding covariance matrix are then updated for next instance using magnetometer readings. The updated attitude is then fed to controller and state vector is input for next iteration of filter.

7. CONTROLLER

The goal of controller of the satellite will be to align the body frame with orbit frame. This control law should be robust to the disturbances faced by the satellite in orbit. This means that even in the presence of disturbance torque, the satellite's attitude should be within desired bound of the reference attitude. The satellite will have two control modes.

- **Detumbling mode:** The first mode is detumbling. When it is separated from the launch vehicle, the satellite has high initial angular velocity of body frame with respect to orbit frame. The detumbling control law will be used to reduce the body angular velocity with respect to orbit frame. This will be achieved using B-dot controller[10]. The estimate of angular velocities will be obtained from magnetometer data.

  B-dot control law is based on the fact that the Earth's magnetic field in the body frame changes with time due to motion of the satellite. Note that the change due to translational motion of the satellite is neglected as contribution due to high angular rates during detumbling is considerably larger. The controller aims to make this rate of change of Earth's magnetic field equal to zero. The required magnetic moment is proportional to the magnitude of the derivative of body-fixed magnetic field but opposite in direction.

  A magnetic actuation based controller[11] was also reviewed. The magnetic moment is determined using the body-fixed Earth's magnetic field vector and angular velocity vector of the body with respect to inertial frame. But it was later rejected as it required gyroscope data, which was not possible to obtain reliably since the estimator cannot be executed in detumbling mode to correct the drifting bias of the gyroscope.

- **Nominal Mode**: Once the angular rates are within desired bounds, the satellite control law will switch to nominal mode. The nominal controller will control the attitude of satellite. Advitiy will have active control in the eclipse phase as well. This decision was based on the fact that half of the passes of Advitiy over India are in eclipse phase.

  If angular velocities increase beyond the bounds then controller mode will switch back to detumbling to reduce angular velocities.

The control laws implemented in previous small satellite missions have been reviewed. The following controllers are currently under consideration:

- Quaternion Feedback Controller[10]: This controller is implemented in lots of previous missions. This is like a simple PID controller. The control torque is directly proportional to the error quaternion and error angular velocity and a few other non-linear terms. This controller is a continuous controller which guarantees asymptotic tracking if the torque applied is exactly equal to the torque demanded and if there are no disturbances present. In case of disturbances and discretized magnetic actuation, its performance degrades significantly.
- Adaptive Controller: The first controller in this category is a variable gain controller[6] which exactly rejects gravity gradient and designed for magnetically actuated systems. This overcomes the two major shortcomings of the quaternion feedback controller.

  Another controller in this category estimates the upper bound on the disturbances acting on the satellite[7]. The implementation of this controller is computationally simple but the tuning is difficult. Work on this controller has been halted since it was not designed specifically for magnetic actuation. Both of these controllers have not been implemented in previous satellite missions. However, they overcome some of the shortcomings of quaternion feedback controller.

8. ACTUATOR MODEL

As we have seen in Section 7, depending on control algorithm used, output of controller block can be desired torque or desired magnetic moment. So, two types of actuator models have been developed depending on two possible cases of output of controller block. Type A will take magnetic moment required directly as input while type B takes torque required as input and magnetic moment required is calculated by

$$\mathbf{m} = (\mathbf{B} \times \boldsymbol{\tau})/\|\mathbf{B}\|^2$$

where (bold character represents vector) **m** is required magnetic moment, **B** is magnetic field at that location, **τ** is the torque required. This formula comes from the logic that since we can't generate torque along magnetic field, so maximum efficiency (in terms of power consumed) will be in the case of magnetic moment perpendicular to both the magnetic field and torque required (as component of magnetic moment in direction of magnetic field will not contribute to torque).

For both the types, once we get the required magnetic moment, current required is calculated using the relation $\mathbf{m} = n\, I\, \mathbf{A}$ where n is number of turns of wire in magnetorquer (same for all 3), I is an array containing the required current in 3 magnetorquers aligned along x, y, z axis of body frame and $\mathbf{A}$ is the area of magnetorquer. Then, we assume magnetorquers to be purely resistive only. So, required voltage is calculated as $V = I * R$ where V represents voltage required, I represents current required and R represents resistance of magnetorquers. From the required voltage to actual current, we are following three approaches on the basis of ease of implementation.

- We will model the actuator as resistive circuit only in the most basic framework. So, in this case the required current obtained will be equal to the actual current.
- We will improve the basic model and model the actuator as an LR circuit. Voltage for a control cycle (the time taken by On-board computing to complete one loop of execution of all the commands) will be given and current for the environment cycle (integration time step used for satellite attitude dynamics simulation) will be calculated using

  $$I = ((1 - e^{-tR/L})(V - i_p R)) / R$$

  where i_p is current in torquers at the end of previous control cycle and t is the time elapsed after beginning of current control cycle.
- We will introduce the fact that voltage supplied to panels is not DC but PWM. So, during ON time

  $$I = ((1 - e^{-t'R/L})(V - i_p R)) / R$$

  and during OFF time

  $$I = (e^{-t'R/L})\, V_p / R$$

  where t' is the time after change of mode (ON mode or OFF mode and V_p is the voltage during previous mode).

9. DYNAMICS AND INTEGRATOR

State vector refers to the attributes of satellite which are propagated in simulation. The main aim of simulation is to check that these attributes are within desired tolerance limits. For our simulation state consists of seven values - initial 4 values for quaternion corresponding to a rotation matrix used for rotating a vector from orbit frame to body frame (qBO) and later 3 values for angular velocity of body frame with respect to orbit frame expressed in body frame. Also the quaternion chosen is according to vector-scalar convention (first 3 values represents vector part of quaternion and last value represents scalar part) and it follows JPL convention[4] of quaternion multiplication.

Dynamics and integrator are used to find next state of satellite using current state of satellite and total torque (sum of disturbance torque and control torque) acting on the satellite. Dynamics block calculates the derivative of state vector. Since, dynamics block will be different depending on whether the state is in accordance to exact dynamics (Body and Inertial frame) or error dynamics (Body and Orbit frame), different results are obtained for the next state in the two cases. From simulations, it was realized that due to numerical stability results in exact dynamics diverges while results in error dynamics converges. This is the reason behind our current choice of state.

The integrator uses the derivative of the state calculated by the dynamics block and integrates it using RK-4 method to get next state. The new state is given as input to the environment and sensor block for continuation of loop thus making it closed.

10. SIMULATION RESULTS

Simulation of uncontrolled satellite in eclipse region to determine the need of control in eclipse

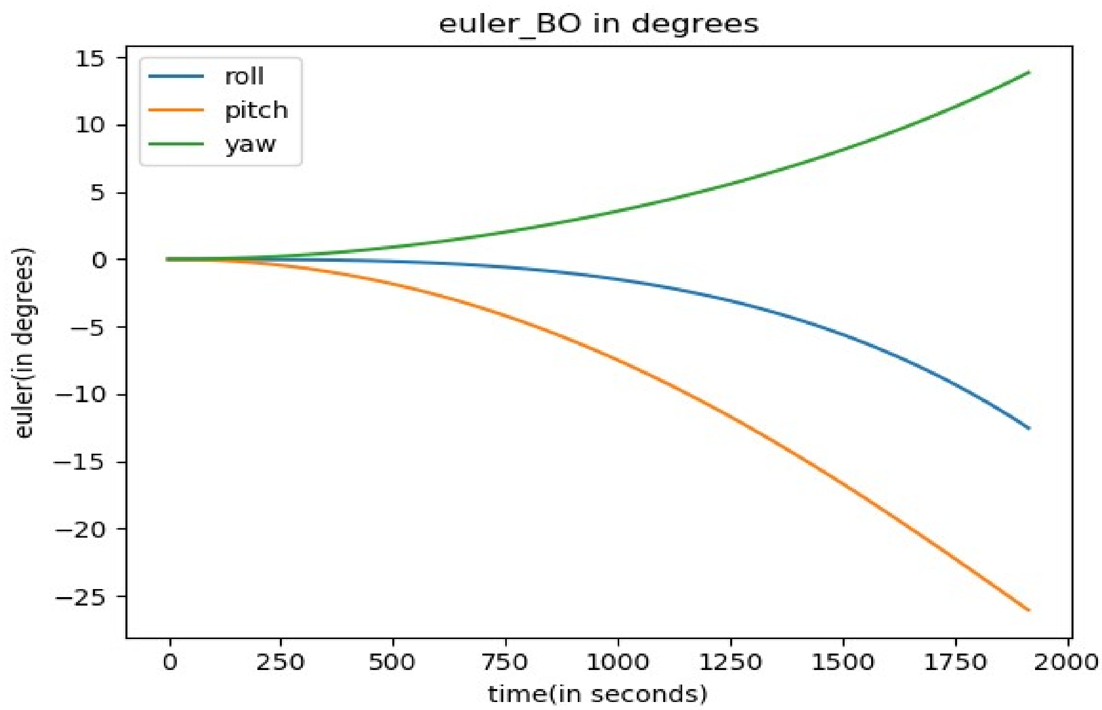

Figure 4: Euler angles versus time

Initial Angular Rates - [0 0 0] deg/s along the X, Y, Z body axes respectively.
Initial Euler Angles - 0 deg roll, 0 deg pitch, 0 deg yaw
Duration - From start of eclipse to end of eclipse
Disturbance torques - Present
Control torques – Absent

One should observe that value of maximum euler angle is 25 degrees at the end of eclipse even if the satellite was completely earth pointing and its angular velocity with respect to earth pointing frame of reference was zero at the start of eclipse. Since in real scenarios it is likely to be already deviated at the start of eclipse, the attitude deviations at the end of the eclipse will be worse. Since half of the passes of Advitiy over India are in eclipse, to transmit the signals with maximum efficiency, satellite has to be controlled during eclipse.

11. QUALITY ASSURANCE

Following quality assurance (QA) practices is essential to minimize chances of mission failure. So, a rigorous procedure is followed to ensure the results generated from simulation are correct and the framework is easy to understand and debug. A document containing guidelines of writing codes and reviewing it has been made. It also consists of instructions for naming the files so that one can infer the type of document/code and its version from just looking at the name. Naming conventions of various elements of code such as functions and variables have also been described to maintain consistency across different codes and avoid confusion. Comments are added along with the header of functions where its inputs, outputs and objective are defined. Comments are also added in the code to explain any mathematical or logical operation. The author of the code also writes a readme file, which gives information on the technical aspects of the code, i.e., the mathematical formulae and scientific laws used with references at the end.

An exhaustive set of test cases is also prepared, where expected outputs for particular inputs are written for each function in code. Finally, a test code is written using unittest module[14] to automate the process of checking whether the actual output from the functions of code match the expected output for the test case inputs. Process has been automated to remove manual error while testing. Now, a second person reviews all the steps identified above and also the correctness and logical consistency of code in general. The author of the code prepares a QA report based on existing format in place for it, and includes there the links for code, readme file and test code. The reviewer writes his suggestions in this QA report, which are then implemented in the code by the author.

GitHub is used for version control. A master repository is maintained for all codes, their readme files, test cases and test codes. Individual team members update their personal GitHub repository after each change made in their code, and then push the completed and reviewed code to the master repository. A team member has been entrusted with responsibility of maintaining the master repository and ensuring that the laid down guidelines are followed by everyone involved in the development of framework.

REFERENCES

1. Felix R Hoots, Ronald L Roehrich, T S Kelso. “Spacetrack report no. 3”. *Project Spacetrack Reports,* Office of Astrodynamics, Aerospace Defense Center, ADC/DO6, Peterson AFB, CO 80914. 1980.
2. Howard D Curtis. *Orbital mechanics for engineering students*. Butterworth Heinemann. Oxford. 2013
3. Christopher Finlay et al. “International Geomagnetic Reference Field: The eleventh generation”. *Geophysical Journal International*. **Vol. 183**, **No. 3**. pp. 1216 – 1230. 2010
4. Erwan Thébault et al. “International Geomagnetic Reference Field: the 12th generation”. *Earth, Planets and Space.* **Vol. 67.** 2015
5. *https://github.com/scivision/igrf12*
6. Oliver Montenbruck, Eberhard Gill. *Satellite Orbits Models, Methods and Applications.* Springer. Berlin. 2000
7. James R Wertz. *Spacecraft Attitude Determination and Control.* Springer. Netherlands. 1978
8. Marcel J Sidi. *Spacecraft Dynamics and Control: A Practical Engineering Approach.* Cambridge University Press. Cambridge. 1997
9. S Baskar, S Sivaji Ganesh. “Introduction to Numerical Analysis”. *Lecture Notes for MA 214*. Department of Mathematics, Indian Institute of Technology Bombay. Mumbai. 2013
10. F Landis Markley, John L Crassidis. *Fundamentals of Spacecraft Attitude Determination and Control.* Springer. New York. 2014
11. Giulio Avanzini, Fabrizio Giulietti. “Magnetic Detumbling of a Rigid Spacecraft”. *Journal of Guidance, Control, and Dynamics*. **Vol. 35**, **No. 4**. pp. 1326 - 1334. 2012
12. Marco Lovera, Alessandro Astolfi. “Global magnetic attitude control of spacecraft in the presence of gravity gradient*”. IEEE transactions on aerospace and electronic systems*. **Vol. 42, No. 3**. pp. 796 - 805. 2006
13. Wenchuan Cai, Xiaohong Liao, David Y Song. “Indirect robust adaptive fault-tolerant control for attitude tracking of spacecraft”. *Journal of Guidance, Control, and Dynamics.* Vol. 31, No. 5. pp. 1456 - 1463. 2008
14. *https://docs.python.org/2/library/unittest.html*