

Design and development of the ECOSat's Attitude Determination and Control (ADCS) onboard software

João Miguel Lousada
Instituto Superior Técnico, Universidade Técnica de Lisboa,

keywords: nanosatellite, ADCS, software, magnetometer, magnetorquer, Kalman filter.

Abstract

This paper reports the results and conclusion that came from the development and testing of a fully functional Attitude Determination and Control System (ADCS) software for the ECOSat project (University of Victoria project for the Canadian Satellite Design Challenge). The two subsystems of Attitude Determination System (ADS) and Attitude Control System (ACS) were developed separately and later integrated and tested.

The determination of the satellite's orientation involves an algorithm that takes into account measurements from highly noisy instruments, such as the magnetometers and the data from the solar arrays (used as an analog sun sensor). A Kalman filter algorithm was developed for this task. It performs a prediction of the satellite's future orientation through a dynamic model and a variety of quaternion operations, that have a certain inaccuracy. The Kalman filter estimates this inaccuracy at all times, in order to weight it against recurrent measures being taken.

The ACS acts throughout the satellite's mission time and provides the necessary attitude adjustments that allow it to perform its mission. For this a PD controller was used, with separate considerations for the de-tumbling stage of the mission, where instrument data is highly conditioned.

1 Introduction

When Geocentrix first proposed the Canadian Satellite Design Challenge (CSDC) back in September 2010, it called out to universities across Canada, challenging students to design and manufacture an operational nanosatellite. During the CSDC teams are challenged to create their own innovative satellite design, including a mission definition and subsystem design and manufacturing. Once constructed all satellites undergo a complete space environmental and launch qualification testing, leading up to the actual launch of the winners.

To answer this competition, the University of Victoria created the ECOSat team. The team is made up of 20 to 30 students from different backgrounds and it handles all the technical aspects of the satellite design. The work presented in this paper refers to the development and testing of the Software for the Attitude Determination and Control Software (ADCS) of the ECOSat nanosatellite.

It cannot be said that the idea of performing the attitude control of a satellite by means of a magnetic based set of torques is an innovative one. As a matter of fact, concepts for a magnetic control system have long been proposed by Wang [1] or Wisni [2], amongst others. However, the relatively small torques provided, the dependence on the

highly variable intensity of the Earth's magnetic field and the complexity of the control problem that arises from it, have kept these systems from a largely spread usage in current satellite designs.

The Kalman filter dates even further back and was originally developed as a tool for linear estimation [3], [4], [5] and soon developed into nonlinear applications in the Apollo program [6], [7]. Since then, a large number of applications of the simple nonlinear extended Kalman filter (EKF) have been presented (e.g. [8],[9]) and also a large number of variations of the EKF have been proposed. The work developed by Crassidis *et al.* [10] provides an extensive overview of current proposed methods of nonlinear attitude estimation, including EKF and its variations.

The Kalman filter provides a good means of integrated an ADCS based on magnetorques and although their usage is still far from other systems of attitude control, magnetorques provide some very interesting features. They are more energy efficient, lightweight and simple the majority of other systems of attitude control. They also have a higher lifetime (since they consume no propellant) and absence of moving parts, which greatly improves their reliability [11]. These characteristics make such a system a very promising candidate for performing the attitude control of a nanosatellite with no extremely high

pointing accuracy requirements.

Since these systems have a simple design and high reliability, they can be easily developed in a laboratory environment with a few specific materials. This is particularly useful for a student driven project such as the ECOSat as it greatly reduces the overall costs of the system.

2 Quaternions

Quaternions were first devised by William Rowan Hamilton in [12] and are, in essence, a representation of the attitude of a body in relation to a certain reference frame. Although more complex than other attitude representation concepts, quaternions don't suffer from the some of the problems the affect these other representations, such as the *gimbal lock* [13], making them more appropriate and reliable for an attitude determination system.

Quaternions are basically a four element object defined as

$$q = q_4 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k} \quad (1)$$

where \mathbf{i} , \mathbf{j} and \mathbf{k} represent imaginary components.

The imaginary part of the quaternion is also called the vector part (designated as \mathbf{q}) while q_4 represents the real (or scalar) part of the quaternion. A quaternion can, therefore, also be represented as

$$q = \begin{bmatrix} \mathbf{q} \\ q_4 \end{bmatrix} = [q_1 \quad q_2 \quad q_3 \quad q_4]^T \quad (2)$$

If defined as an unit quaternion, i.e.,

$$|q| = \sqrt{|\mathbf{q}|^2 + q_4^2} = 1, \quad (3)$$

a quaternion can represent a rotation. These quaternions are called rotational quaternions and define a rotation according to

$$\mathbf{q} = \begin{bmatrix} k_x \sin(\theta/2) \\ k_y \sin(\theta/2) \\ k_z \sin(\theta/2) \end{bmatrix} = \hat{\mathbf{k}} \sin(\theta/2), \quad (4)$$

$$q_4 = \cos(\theta/2)$$

A quaternion defined this way will represent a rotation around the axis described by $\hat{\mathbf{k}}$ by the rotation angle θ [14]. It is also noteworthy that such a definition leads to q representing the same final orientation as $-q$, being the only difference the path of the rotation taken, with the shortest path always represented by positive q_4 . [15]

For the development of a model for the quaternion dynamics required for the Attitude Determination algorithm,

it is important to, firstly, introduce the concepts of the *skew-symmetric matrix operator*:

$$[\mathbf{q} \times] = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix} \quad (5)$$

and the Ω matrix:

$$\begin{aligned} \Omega(\omega) &= \begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix} \\ &= \begin{bmatrix} -[\omega \times] & \omega \\ -\omega^T & 0 \end{bmatrix} \end{aligned} \quad (6)$$

If we consider an object defined in the local coordinate frame $\{L\}$ and moving in with respect to the global reference frame $\{G\}$, we can compute the rate of change (or *time derivative*) of its attitude quaternion by

$${}^L_G \dot{q}(t) = \lim_{\delta t \rightarrow 0} \frac{1}{\delta t} \left({}^L_G q(t+\delta t) - {}^L_G q(t) \right) \quad (7)$$

which can be further developed into [16]:

$${}^L_G \dot{q}(t) = \frac{1}{2} \Omega(\omega) {}^L_G q \quad (8)$$

where $\Omega(\omega)$ is given by Equation (6).

Introducing now the *quaternion integrator matrix* $\Theta(t, t_k)$, one can obtain the attitude quaternion at current time t from a previous attitude quaternion at time t_k by:

$${}^L_G q(t) = \Theta(t, t_k) {}^L_G q(t_k) \quad (9)$$

Assuming a constant angular velocity ω over the time interval of the integration one can obtain the *zeroth order quaternion integrator* [14]:

$$\begin{aligned} \Theta(\Delta t) &= \cos\left(\frac{|\omega|}{2} \Delta t\right) \cdot I_{4 \times 4} \\ &\quad + \frac{1}{|\omega|} \sin\left(\frac{|\omega|}{2} \Delta t\right) \cdot \Omega(\omega) \end{aligned} \quad (10)$$

However, a more accurate approximation is to consider the angular velocity ω as a linear evolution within the interval of integration. This yields the *first order quaternion integrator*:

$$\begin{aligned} \Theta(t, t_k) &= \exp\left(\frac{1}{2} \Omega(\bar{\omega})\right) + \frac{1}{48} \left(\Omega(\omega(t)) \Omega(\omega(t_k)) \right. \\ &\quad \left. - \Omega(\omega(t_k)) \Omega(\omega(t)) \right) \Delta t^2 \end{aligned} \quad (11)$$

where $\bar{\omega}$ is simply the arithmetic mean of $\omega(t)$ and $\omega(t_k)$.

3 Kalman Filter

A *Kalman filter* is a type of recursive filter that takes into account a series of measurements, their inherent noise models and other inaccuracies to give a best estimation the current state (the satellite's attitude).

This filter makes use of a system's dynamics model (e.g. satellite attitude dynamics), known control inputs to that system (e.g. the torques applied), and multiple sequential measurements (e.g. from the satellite's attitude sensors) to form an estimate of the system's varying quantities (its *state*). Kalman filters are divided in two steps: the prediction step and the update step.

The 7-state Extended Kalman filter is a reliable control filter commonly used for the ADCS task. It uses relatively simpler mathematical operations, when compared to other EKF's. [17]

As its name indicates, the 7-state EKF uses a state vector of 7 elements. These are the 4 attitude quaternion elements, with respect to the inertial frame, and the 3 angular rates, also defined in the inertial frame:

$$\mathbf{x} = [q_1 \ q_2 \ q_3 \ q_4 \ \omega_x \ \omega_y \ \omega_z]^T \quad (12)$$

However, the 7-state EKF involves many 7 x 7 matrices, placing a fairly heavy computational demand upon the on-board computer of the typical low power and low cost nanosatellite. Hence, any means of significantly reducing the computational needs of the EKF while not sacrificing operability is highly desirable. [19]

The 6-state EKF, first theorized by Lefferts [20] aims at reducing the computational load of the 7-state EKF by reducing the attitude quaternion to a 3 element attitude vector within the filter computations. This needs to be performed with caution in order not to lose the advantages of using quaternions over 3D vectors for attitude representation.

Firstly, it is important to introduce the notions of quaternion and angular rate differential errors δq and $\delta \omega$. These are the main characteristic of the 6-state EKF and represent, in essence, small deviations in the attitude quaternion and body rates. These can be used to propagate the variables in time by:

$$q_k = \delta q \otimes q_{k-1} \quad (13)$$

$$\omega_k = \omega_{k-1} + \delta \omega \quad (14)$$

where the \otimes operator represents the quaternion multiplication.

While Equation (14) is somewhat intuitive and simply understood, Equation (13) does not come so naturally. Basis-

cally δq represents a deviation in the attitude quaternion. It is, itself a quaternion, but it does **not** correspond to the spacecraft's attitude. Considering all the quaternion components

$$\delta q = [\delta q_1 \ \delta q_2 \ \delta q_3 \ \delta q_4]^T \quad (15)$$

and quaternion properties (specifically, their unitary norm), it comes that

$$\delta q_4 = \sqrt{1 - \delta q_1^2 - \delta q_2^2 - \delta q_3^2} \quad (16)$$

Since δq_4 can be defined from the remaining properties, its use is redundant, and so, the 6-state EKF state vector is defined as

$$\mathbf{y} = [\delta q_1 \ \delta q_2 \ \delta q_3 \ \delta \omega_x \ \delta \omega_y \ \delta \omega_z]^T \quad (17)$$

This state vector is designated the *auxiliary* state vector and is the one used in the filter computations for the 6-state EKF. Also a *true* state vector, similar to the one defined in Equation (12) is used, in order to represent the attitude of the satellite. This *true* state vector will introduce the notion of angular rate bias, \mathbf{b} , that gives the difference between the measured and the estimated angular rates:

$$\hat{\omega} = \omega_m - \mathbf{b} \quad (18)$$

For very accurate measurements and estimations, $\mathbf{b} \approx \mathbf{O}_{3 \times 1}$, a 3×1 null matrix (or vector), thus facilitating calculations. The *true* state vector is, then, given by:

$$\mathbf{x} = [q_1 \ q_2 \ q_3 \ q_4 \ \mathbf{b}_x \ \mathbf{b}_y \ \mathbf{b}_z]^T \quad (19)$$

The auxiliary state vector, can also be further optimized [14] into:

$$\mathbf{y} = [\delta \theta_1 \ \delta \theta_2 \ \delta \theta_3 \ \Delta \mathbf{b}_x \ \Delta \mathbf{b}_y \ \Delta \mathbf{b}_z]^T \quad (20)$$

where

$$\delta q_i = k_i \sin(\delta \theta / 2) \approx \frac{1}{2} \delta \theta \quad (21)$$

with k_i corresponding to the normalized vector part of the quaternion

$$k_i = \frac{q_i}{|\mathbf{q}|} \quad (22)$$

and with

$$\mathbf{b}_k = \mathbf{b}_{k-1} + \Delta \mathbf{b} \quad (23)$$

3.0.1 Prediction Step

For each time step, the prediction step shall be the first part of the algorithm operations. In it, the state vector is estimated based on the system's dynamic model and the

control inputs, as in:

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_{k-1} \mathbf{u}_{k-1} \quad (24)$$

The state transition matrix, \mathbf{F}_k is obtained from [14]:

$$\mathbf{F}_k = \begin{bmatrix} \boldsymbol{\Theta} & \boldsymbol{\Psi} \\ \mathbf{O}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix} \quad (25)$$

The matrix $\boldsymbol{\Theta}$ is, in fact, a rotational matrix with ω as the axis and $|\hat{\omega}|\Delta t$ as the angle:

$$\begin{aligned} \boldsymbol{\Theta} = & \cos(|\hat{\omega}|\Delta t) \cdot \mathbf{I}_{3 \times 3} - \sin(|\hat{\omega}|\Delta t) \cdot \left[\frac{\hat{\omega}}{|\hat{\omega}|} \times \right] \\ & + (1 - \cos(|\hat{\omega}|\Delta t)) \cdot \frac{\hat{\omega}}{|\hat{\omega}|} \cdot \left[\frac{\hat{\omega}}{|\hat{\omega}|} \right]^T \end{aligned} \quad (26)$$

It can be further simplified, for small values of $|\hat{\omega}|$, to:

$$\boldsymbol{\Theta} = \mathbf{I}_{3 \times 3} - \Delta t [\hat{\omega} \times] + \frac{\Delta t^2}{2} [\hat{\omega} \times]^2 \quad (27)$$

The matrix $\boldsymbol{\Psi}$ is given by:

$$\begin{aligned} \boldsymbol{\Psi} = & -\mathbf{I}_{3 \times 3} \Delta t + \frac{1}{|\hat{\omega}|^2} (1 - \cos(|\hat{\omega}|\Delta t)) [\hat{\omega} \times] \\ & - \frac{1}{|\hat{\omega}|^3} (1 - \sin(|\hat{\omega}|\Delta t)) [\hat{\omega} \times]^2 \end{aligned} \quad (28)$$

Which can also be approximated, for small values of $|\hat{\omega}|$, by:

$$\boldsymbol{\Psi} = -\mathbf{I}_{3 \times 3} \Delta t + \frac{\Delta t^2}{2} [\hat{\omega} \times] - \frac{\Delta t^3}{6} [\hat{\omega} \times]^2 \quad (29)$$

The $\mathbf{O}_{3 \times 3}$ and $\mathbf{I}_{3 \times 3}$ matrices represent a null and an identity 3×3 matrices, respectively.

The state transition matrix \mathbf{F}_k is fully defined, and is used to determine the state covariance matrix:

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{Q}_k \quad (30)$$

3.0.2 Update Step

In the update step, a measurement (or observation) vector, \mathbf{z}_k , is now available. This measurement is compared to a predicted estimation of the measurement vector, $\hat{\mathbf{z}}_{k|k-1}$. It is defined by the measurement model matrix \mathbf{H}_k and the estimated state vector, $\hat{\mathbf{x}}_{k|k-1}$, determined in the prediction step.

$$\hat{\mathbf{z}}_{k|k-1} = \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1} \quad (31)$$

Comparing the measured and predicted measurement vector, we obtain the innovation (or measurement residual),

$$\mathbf{r}_k = \mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1} \quad (32)$$

and with it, the innovation covariance matrix

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k \quad (33)$$

where \mathbf{R}_k is the covariance matrix of the observation/measurement noise.

Given the innovation covariance matrix, it is now possible to determine the optimal Kalman gain (from [17]):

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T [\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k + \mathbf{R}_k]^{-1} \quad (34)$$

Finally, we are now in condition to determine the state estimate vector,

$$\mathbf{y}_{k|k} = \begin{bmatrix} \delta\theta \\ \Delta \mathbf{b} \end{bmatrix} = \mathbf{K}_k \cdot \mathbf{r} \quad (35)$$

where \mathbf{r} represents the *innovation* given by the difference between the measured and the predicted measurement vectors.

and its covariance matrix,

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1} \quad (36)$$

Thus fully defining the current state. The algorithm would then move forward in time with:

$$\hat{\mathbf{x}}_{k-1|k-1} = \hat{\mathbf{x}}_{k|k} \quad (37)$$

$$\mathbf{P}_{k-1|k-1} = \mathbf{P}_{k|k} \quad (38)$$

and then repeat the whole process, starting with the new prediction step. Finally, the attitude quaternion and the angular rate (*i.e.* the true state vector) can then be updated according to Equations (13),(14),(21) and (23).

4 Attitude Determination Algorithm

The algorithm implemented was firstly developed in simulation mode, *i.e.* not attached to the hardware. The algorithm has non noisy models of the instruments and can determine the actual attitude of the satellite. Simulated noise is then added and the determination attitude algorithm tests against the reference *true* attitude.

The determination of the orbital parameters and attitude of the satellite is based on the 6 state Kalman filter, previously defined. The flowchart presented in Figure 1 represents the structure of this algorithm.

In the first step, the variables for the simulation are set. These include the initial time and date of the simulation, the position and the attitude of the satellite, *i.e.*, the initial attitude quaternion and angular velocity vector. These variables correspond to the true attitude and position of the satellite that will be estimated by the attitude deter-

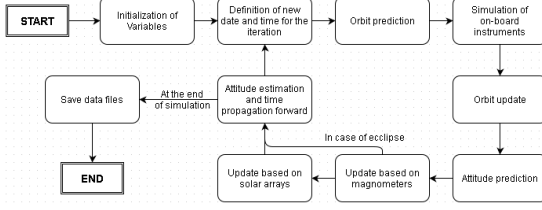


Figure 1: Flowchart of the Attitude Determination Algorithm

mination algorithm. They change throughout the iteration according to satellite dynamics and serve as reference for the later incorporation of the instrument measurements. The initial conditions of the Kalman filter are then defined. These provide starting point of the filter. From this starting point the filter will try to converge to the real attitude of the satellite. Since the the initial attitude of a real satellite is impossible to guess, the definition of the variables can be done rather arbitrary. Also the expected noise of the different instruments is defined at this point. This is dependent on the type and precision of the instrument and was defined according to each instrument's manufacturer's specifications.

In the second step, already included inside the iteration phase of the algorithm, the time frame of the iteration is defined based on the date and time and time step set previously (if it is the first iteration, the initial values are considered). This allows for the computation of the sun's position for that specific time frame, among other time dependent variables. Following that, the orbital elements are propagated forward according to the initial conditions.

On the third step, the instrument measurements are simulated. The process for the simulation of these measurements takes into account the true position, attitude and angular rates of the satellite and is completed with the addition of random white noise with a norm as defined in step one. The simulated measurements include gyroscope's data, electromagnetic field, GPS data and the Sun's position at the iteration time.

The forth step consists on the estimation of the orbital parameters. This takes into account the Keplerian prediction of the orbit and the current (noisy) GPS measurements. Since the satellite is not provided with a thrusting system, it is expected that the Keplerian prediction is a fairly accurate one. The algorithm then takes a weighted average (where more weight is given to the prediction) in order to best estimate the current orbital parameters.

The fifth step is already part of the Kalman filter algorithm that estimates the attitude and angular rates of the satellite. In this step, the propagation step of the Kalman filter, the angular rate and the attitude quaternion are propagated according to Equations 18 and 24, respectively. In Equation 24, \mathbf{F}_k is given by Equation 25. Also the

state covariance matrix, $\mathbf{P}_{k|k-1}$, is calculated according to Equation 30, where the noise covariance matrix, \mathbf{Q}_k , is given by

$$\mathbf{Q}_k = \begin{bmatrix} \mathbf{Q}_{11} & \mathbf{Q}_{12} \\ \mathbf{Q}_{12}^T & \mathbf{Q}_{22} \end{bmatrix} \quad (39)$$

with

$$\mathbf{Q}_{11} = \sigma_r^2 \cdot I_{3 \times 3} + \sigma_w^2 \cdot (I_{3 \times 3} \frac{\Delta t^3}{3} \quad (40)$$

$$+ \frac{(|\hat{\omega}|\Delta t)^3}{3} + 2 \sin(|\hat{\omega}|\Delta t) - 2|\hat{\omega}|\Delta t}{|\hat{\omega}|^5} \cdot [\hat{\omega} \times]^2)$$

$$\mathbf{Q}_{12} = -\sigma_w \cdot (I_{3 \times 3} \frac{\Delta t^2}{2} \quad (41)$$

$$- \frac{|\hat{\omega}|\Delta t - \sin(|\hat{\omega}|\Delta t)}{|\hat{\omega}|^3} \cdot [\hat{\omega} \times] + \frac{(|\hat{\omega}|\Delta t)^2}{2} + \cos(|\hat{\omega}|\Delta t) - 1}{|\hat{\omega}|^4} \cdot [\hat{\omega} \times]^2)$$

$$\mathbf{Q}_{22} = \sigma_w^2 \Delta t \cdot I_{3 \times 3} \quad (42)$$

The sixth step corresponds to update step of the Kalman filter. Within it, the measurement matrix, \mathbf{H}_k is extracted from the simulated measurements followed by the computation of the Kalman gain matrix, \mathbf{K}_k , according to Equation 34. The state vector of the 6 state kalman filter can then be obtained from Equation 35. Finally, the attitude quaternion and the angular rate (i.e. the true state vector) can than be updated according to Equations (13),(14),(21) and (23).

Note that, in this filter, two different update steps exist (corresponding to steps six and seven of the algorithm). The first accounts for the sun position measurements while the second takes into account magnetometer measurements. This method allows us to consider two different sets of measurements in the same iteration. Note also that the satellite's designed orbit will contain moments when the satellite is eclipsed by the Earth and, thus, there will be occasions with no measurements of the Sun's position available. This situation was also accounted for in the simulation, leading to use of only magnetometer measurements for 27% of the satellite's orbit. Since it is a Sun synchronous orbit, we can consider this value to be constant for simulation purposes.

The eighth and final step contained in the iteration process corresponds simply to the setting of the Kalman filter initial variables according to the final estimation of the iteration. This will be necessary to start the iteration process once more.

Finally, after all the iterations defined in the first step are completed, the algorithm stores all the data obtained for later analysis. These files are later analysed using the *Mat-*

Table 1: Convergence times for both static and non static reference attitudes of different variations of the PID controller.

Type of controller	Time of convergence (min)	
	Static reference	Non static reference
<i>PID</i>	174	231
<i>PI</i>	342	387
<i>PD</i>	167	198
<i>P</i>	153	507
<i>D</i>	—	—
<i>I</i>	—	—

Table 2: Best values obtained for the gains of the normal mission controller.

Gains	Best Value Found
K_p	0.00015
K_d	0.2
K_i	—

lab software.

5 Attitude Control Algorithm

A PID controller was initially selected for the task of attitude control of this project. The PID controller, with the three terms (proportional, integral and derivative) is the most commonly used controller in nowadays applications. However, for some applications only one or two terms are required to provide an appropriate control of the system. These are designated PI, PD, P, or I controllers, according to the terms used.

From a general point of view, a PID controller, with all its terms, is given by:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t) \quad (43)$$

where $u(t)$ represents the controller's output and $e(t)$ the controller's input, i.e., the variable's error in relation to the reference value. K_p , K_i and K_d represent the proportional, integral and derivative gains, respectively. These gains need to be adjusted as incorrect values can lead to poor controller performance or even instability of the controller. A series tests yielded the values presented in Table 1. This results allow us to conclude the better performance of the PD for this application in particular. The exact values obtain for its parameters are presented in Table 2.

For the case of a satellite's attitude control based on

quaternions, Equation 43 takes the form of

$$T(t) = K_p q_e(t) + K_i \int_0^t q_e(\tau) d\tau + K_d \omega_e(t) \quad (44)$$

where q_e represents the error in the attitude quaternion in relation to the reference attitude quaternion and ω_e , the error in the angular rates (also in relation to the reference values).

During these tests of Table 1, it was found that controller performed best with only the proportional and derivative terms. The use of the integral term not only led to a higher settling time, but also implied higher computational loads. The controller was, thus, reduced to a PD controller defined by:

$$\begin{aligned} T(t) &= K_p q_e(t) + K_d \omega_e(t) \\ &= 0.00015 q_e(t) + 0.2 \omega_e(t) \end{aligned} \quad (45)$$

Equation 45 yields the necessary torque for correcting the attitude given the reference and the real values of the attitude variables. However, it does not account for limitations of the actuation system (the magnetorquers). For the sake of correctness, the simulation software takes into account the local magnetic field and the maximum magnetic moment of each coil (established, from the hardware design, as $1 A.m^2$) in order to calculate the maximum achievable torque at each time. This leads to a more accurate simulation of the real mission conditions.

During the initial tumbling phase of the mission, correct attitude estimations cannot be made since GPS data (and, consequently, correct magnetometer and solar array data) will be unavailable - due to the initial incorrect pointing of the GPS antenna. The only available data corresponds to the angular rate measurements from the gyroscopes.

Due to the lack of attitude information, the de-tumbling algorithm needs to be very measurement independent. One commonly used option is the B-dot controller [?]. This controller only makes use of the derivative of the magnetic field (and not the locally dependent magnetic field measurements themselves), so it can be used even without GPS data. However, an even simpler and faster control strategy is to take the momentary angular rates measured by the gyroscopes and apply an inversely proportional controller to them. A series of simulations was performed to obtain the best value for the gain.

Note that, like for the normal mission controller, the simulation algorithm verifies at all times that the torques being simulated never surpass the imposed physical limit of the hardware.

The de-tumbling controller is used in the initial phase, when the unpredictable and elevated angular rates make it impossible to obtain GPS data. However, once the angular rates are low enough (the threshold was defined as a total angular rate of $0.001 rad/s$), the control algorithm will

Table 3: Best values obtained for the gains of the detumbling controller.

Gains	Best Value Found
K_p	-0.005
K_d	—
K_i	—

change to the normal mission "mode", where all attitude information is available and more accurate control can be performed. In the following chapter the performance of the algorithm in the tasks of attitude determination and control is analysed.

6 Results

A series of simulations was performed to assess the algorithm's functionalities.

The performance in the determination of attitude of the satellite was tested first. Different attitudes and initial conditions were simulated and their result analysed. The simulations were divided into tests with a static attitude quaternion, *i.e.*, with null angular rates and dynamic attitude quaternions (with constant non-null angular rates).

In order to initially test the attitude determination filter's stability a simulation of with a real fixed attitude quaternion $q_{real} = [1, 0, 0, 0]^T$ was performed.

The filter converged quickly to the correct values even though it was set initially to the wrong values. The convergence time was approximately $1800s(30min)$ which represents a very small value when compared to the expected mission time.

In the following simulation a non-null angular rate was considered. The initial real conditions of the simulation were $q_{real} = [1, 0, 0, 0]^T$ and $\omega_{real} = [0.01, 0, 0]^T$ ¹. The filter presented an initial deviation in both the attitude quaternion and angular rates but quickly converged again to the correct values. It was also visible that the filter was able to follow the variations of the attitude quaternion perfectly after converging.

Afterwards an error in the initial conditions of the filter was induced. The initial attitude quaternion was set to $q_k = [0, 0, 1, 0]^T$. The filter took longer to converge to the correct values, as it was expected. But it did converge after approximately $1500s$, even faster than in the static attitude simulation.

The simulation that followed induced an error also in the initial angular rate, in addition to the previous error in the attitude quaternion. An initial angular rate, $\omega_k = [0, 0, 0.5]^T$ was defined. The results showed an

initial instability of the filter, quickly changing it's estimations. This is explained by the relatively high and inaccurate angular rates (approximately 500 times higher than the expected mission angular rates). But even given this initial instability and inaccuracy, the filter converged quickly and followed the real values correctly afterwards. Once demonstrated the attitude determination capacities of the filter, the attitude control part of the algorithm was fully tested.

A simple simulation of the control's algorithm performance for a fixed reference (*i.e.*, desired) attitude quaternion was performed. The filter was set to an initial angular rate of $\omega_k = [0.01, 0, 0]^T$. The reference conditions were $q_{ref} = [1, 0, 0, 0]^T$ and $\omega_{ref} = [0.01, 0, 0]^T$.

Figures 2, 3 and 4 show the results including an uncertainty in the initial conditions ($q_k = [0, 1, 0, 0]^T$ and $q_{real} = [1, 0, 0, 0]^T$). Figure 4 represents the torques being applied to the system by the control system. These are always limited according to the hardware's physical limitations and are a function of the magnetorquers' maximum power and the external electro magnetic field, which depends on the position in orbit.

Note the slow response of the control system for this case. This was expected since the filter first needs to converge to the correct value. Note also, that filter follows a fixed reference of $q_{ref} = [-1, 0, 0, 0]^T$ rather than $q_{ref} = [1, 0, 0, 0]^T$. This might seem like an error at first glance, but if we recall Equation (4), we realize that $q = [-1, 0, 0, 0]^T$ and $q = [1, 0, 0, 0]^T$ correspond to the same exact attitude.

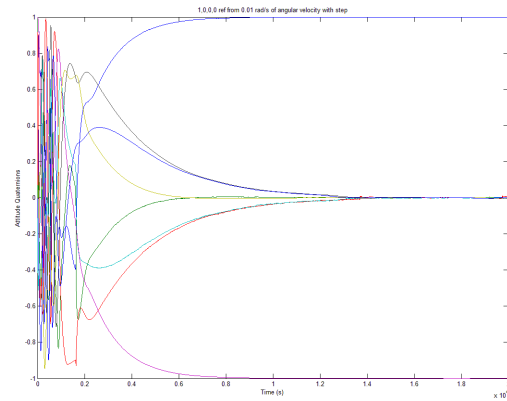


Figure 2: Behavior of the estimated attitude quaternion for initial conditions $q_k = [0, 1, 0, 0]^T$ and $\omega_k = [0.01, 0, 0]^T$, real initial conditions $q_{real} = [1, 0, 0, 0]^T$ and $\omega_{real} = [0.01, 0, 0]^T$ and reference conditions $q_{ref} = [1, 0, 0, 0]^T$ and $\omega_{ref} = [0, 0, 0]^T$. Colors are defined as $q_k = [dark-blue, red, green, teal]^T$ and $q_{real} = [purple, yellow, black, dark-blue]^T$. The reference is not represented.

¹ All angular rates are given in SI units (rad/s).

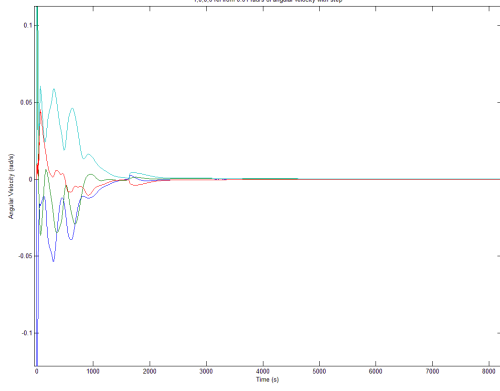


Figure 3: Behavior of the estimated angular rates for initial conditions $q_k = [0, 1, 0, 0]^T$ and $\omega_k = [0.01, 0, 0]^T$, real initial conditions $q_{real} = [1, 0, 0, 0]^T$ and $\omega_{real} = [0.01, 0, 0]^T$ and reference conditions $q_{ref} = [1, 0, 0, 0]^T$ and $\omega_{ref} = [0, 0, 0]^T$. Colors are defined as $\omega_k = [dark - blue, red, green]^T$ and $|\omega|$ in teal.

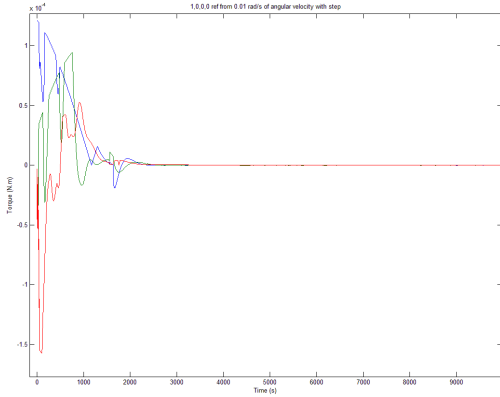


Figure 4: Torques applied by the control system for initial conditions $q_k = [0, 1, 0, 0]^T$ and $\omega_k = [0.01, 0, 0]^T$, real initial conditions $q_{real} = [1, 0, 0, 0]^T$ and $\omega_{real} = [0.01, 0, 0]^T$ and reference conditions $q_{ref} = [1, 0, 0, 0]^T$ and $\omega_{ref} = [0, 0, 0]^T$. Colors are defined as $T_{(x,y,z)} = [dark - blue, red, green]^T$.

Once demonstrated the algorithm's control capability for a given constant reference attitude value, its performance while following the mission's attitude requirements was tested. This is no longer a fixed attitude but rather an attitude constantly changing and defined by constant angular rates.

The mission design defined a first attitude requirement of constantly pointing of the payload towards the Earth. Other attitude requirements referred the need to also rotate about the longitudinal axis (one full rotation per orbit). This requirement is due to optimized pointing needs of the solar panels estimated by the Power Subsystem team.

This translates into a desired set angular rates $\omega_{ref} = [0.0007467, 0.0007467, 0]^T$, with $|\omega_{ref}| = 0.001056$, corresponding to a full rotation per orbit about an intermediate axis. This simulation was performed with initial angular rates of $\omega_{real} = [0.05, 0.05, 0.05]^T$ and achieved the reference attitude within 14000 s. The following test (Figures 5 to 7) simulated the same situation but accounted also for the initial uncertainty of the filter, i.e., considered an initial attitude angular rates completely different than the real ones. This lead to an initial period of high uncertainty and high angular rates (approximately 3 hours). Afterwards, the attitude estimation filter converged to real values and the control algorithm was able to controllably bring the satellite's attitude to the desired one, within approximately 3 more hours.

This last simulation corresponds to the closest approach to the real mission's conditions: high tumbling angular rates in all directions and high uncertainty in the initial phase of the mission. The algorithm developed demonstrated to be able to handle this situation and bring the satellite to its nominal position within a total of 6 hours (21600 s) since the beginning of mission.

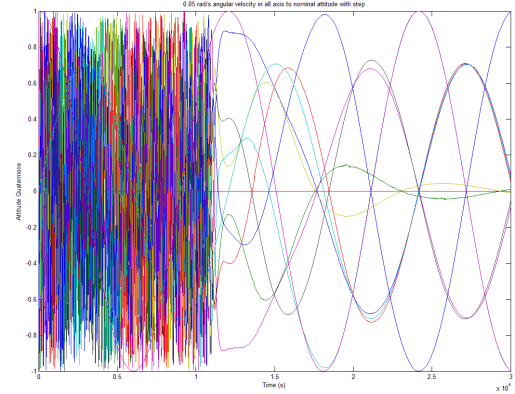


Figure 5: Behavior of the estimated attitude quaternion for initial conditions $q_k = [0, 1, 0, 0]^T$ and $\omega_k = [0, 0, 0]^T$, real initial conditions $q_{real} = [1, 0, 0, 0]^T$ and $\omega_{real} = [0.05, 0.05, 0.05]^T$ and reference conditions $q_{ref} = [1, 0, 0, 0]^T$ and $\omega_{ref} = [0.0007467, 0.0007467, 0]^T$. Colors are defined as $q_k = [dark - blue, red, green, teal]^T$, $q_{real} = [purple, yellow, black, dark - blue]^T$ and $q_{ref} = [purple, teal, red, green]^T$.

7 Conclusions

A fully functional Attitude Determination and Control System (ADCS) software was developed and tested. The software was developed in two different steps. Firstly the

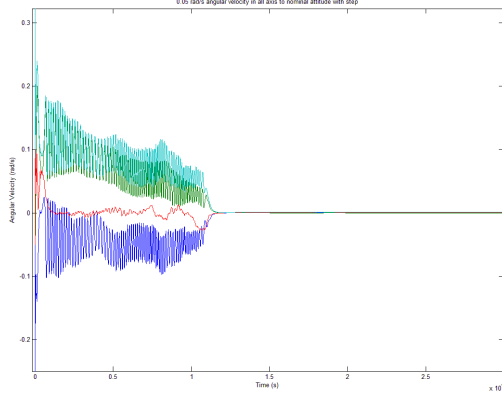


Figure 6: Behavior of the estimated angular rates for initial conditions $q_k = [0, 1, 0, 0]^T$ and $\omega_k = [0, 0, 0]^T$, real initial conditions $q_{real} = [1, 0, 0, 0]^T$ and $\omega_{real} = [0.05, 0.05, 0.05]^T$ and reference conditions $q_{ref} = [1, 0, 0, 0]^T$ and $\omega_{ref} = [0.0007467, 0.0007467, 0]^T$. Colors are defined as $\omega_k = [dark - blue, red, green]^T$ and $|\omega|$ in teal.

algorithm for the correct attitude determination of the attitude was designed. This algorithm makes use of a 6-state Kalman filter. The algorithm integrates measurements from GPS, gyroscopes, magnetometers and solar arrays to obtain the best possible attitude estimation.

Given the correct determination of attitude, the attitude control algorithm was developed. This algorithm has two different functionalities: de-tumbling control and normal mission control. In it, a PD control system is used as it proved to have a better performance than the initial PID control system.

Once completely developed, the complete ADCS algorithm was put through a series of tests and simulations to access its performance. The algorithm was firstly tested on its attitude determination capacity, where it had no problems converging to the true attitude even for non constant true attitude and a totally inaccurate initial attitude. Afterwards test were made on the control part of the system. Given static or changing reference attitudes, the algorithm had no difficulties following them. Finally, both attitude and control were tested, in a simulation that mimicked the conditions expected in the real mission. The simulation was set with high initial angular rates in all axis ($\omega = [0.05, 0.05, 0.05]$ corresponding to the highest expected tumbling rates) and purposely inaccurate initial attitude estimation. This would be a "worst case scenario" simulation. The results showed that the ADCS system is capable of recovering from the tumbling situation within the first $1.1 \times 10^4 s$, (approximately 3 hours) and converges to the proper attitude within $2.3 \times 10^4 s$ (a little over 6 hours) from the mission's beginning. Afterwards, the filter follows the reference attitude without problems.

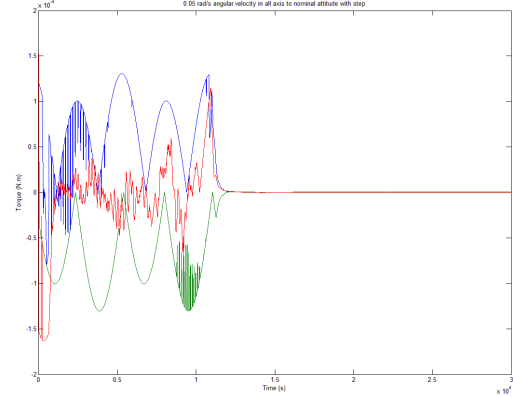


Figure 7: Torques applied by the control system for initial conditions $q_k = [0, 1, 0, 0]^T$ and $\omega_k = [0, 0, 0]^T$, real initial conditions $q_{real} = [1, 0, 0, 0]^T$ and $\omega_{real} = [0.05, 0.05, 0.05]^T$ and reference conditions $q_{ref} = [1, 0, 0, 0]^T$ and $\omega_{ref} = [0.0007467, 0.0007467, 0]^T$. Colors are defined as $T_{(x,y,z)} = [dark - blue, red, green]^T$.

8 Future Work

This thesis was performed within UVic's ECOSat project. A fully functional ADCS software was developed and fully tested. Results showed its proper performance and its readiness to be integrated into the hardware as soon as possible.

The software was developed to make this integration the easiest possible. An extensive "readme" file was included in the software's folder. This file covers all the changes that need to be made in order to integrate the software properly within the hardware. It is expected that anyone with a basic knowledge of the C programming language and satellite attitude will have no problem understanding and working with this software.

One aspect that could still be improved would be the de-tumbling algorithm. Due to the tight deadlines usual for this type of projects, the B-dot de-tumbling control system was not explored in full detail. The simpler angular rate proportional control system presented a better performance in the initial testing and was, therefore, selected for the de-tumbling task of the filter. However, the author believes that, given more time, the B-dot controller could have been fine tuned into a better performance.

References

- [1] Wang, P.; Shtessel, Y. B.; Wang, Y. (1998). *Satellite Attitude Control Using only Magnetorquers*. IEEE American Control Conference.
- [2] Wiśniewski, R.; Blanke, M. (1997). *Three-axis Satel-*

- lite Attitude Control Base on Magnetic Torquing*. 13th IFAC World Congress.
- [3] R.E. Kalman (1960) *A New Approach to Linear Filter Prediction Problems*. Journal of Basic Engineering, Vol. 82, pp. 35-45.
 - [4] R.E. Kalman; R.S. Bucy (1961) *New Results in Linear Filtering and Prediction Theory*. Journal of Basic Engineering, Vol. 83, pp. 95-108.
 - [5] Zarchan, P.; Musoff, H. (2005). *Fundamentals of Kalman Filtering: A Practical Approach*. AIAA.
 - [6] G.L. Smith; S.F. Schmidt (1961) *The Application of Statistical Filter Theory to Optimal Trajectory Determination Onboard a Circumlunar Vehicle*. presented at AAS Meeting, Aug. 1-3, 1961.
 - [7] S.F. Schmidt (1961) *The Kalman Filter: Its Recognition and Development for Aerospace Applications*. Journal of Guidance and Control, Vol. 4, pp. 4-7.
 - [8] J.L. Crassidis; J.L. Junkins (2004) *Optimal Estimation of Dynamic Systems*. Chapman Hall/CRC, Boca Raton, FL, USA.
 - [9] F.L. Markley (2003) *Attitude Error Representations for Kalman Filtering*. Journal of Guidance, Control and Dynamics, Vol. 63, pp. 311-317.
 - [10] J.L. Crassidis; F.L. Markley; Y. Cheng (2007) *A Survey of Nonlinear Attitude Estimation Methods*. University of Buffalo, NY, USA.
 - [11] Ovchinnikov, M. Yu.; Penkov, V. I.; Ilyin, A. A.; Selivanov, S. A. (2005). *Magnetic Attitude Control Systems of the Nanosatellite TNS-Series*. 5th IAA Symposium on Small Satellites for Earth Observation, Berlin.
 - [12] Hamilton, W. R. (1866). *Elements of Quaternions*. Longmans, Green & Co., London.
 - [13] Hoag, D. (1963). *Apollo Guidance and Navigation - Considerations of Apollo IMU Gimbal Lock*. MIT, Instrumental Laboratory Document E-1344.
 - [14] Trawny, N.; Roumenliotis S. I. (2005). *Indirect Kalman Filter for 3D Attitude Estimation*. Multiple Autonomous Robotic Systems Laboratory, Technical Report no. 2005-002, Rev. 57.
 - [15] Breckenridge, W. G. (1999). *Quaternions - Proposed Standard Conventions*. JPL, Tech. Rep. INTEROFFICE MEMORANDUM IOM 343-79-1199.
 - [16] Lousada, J. (2013). *Design and development of the ECOSat's Attitude Determination and Control System (ADCS) onboard software*. MSc. Thesis, Instituto Superior Técnico.
 - [17] M.J. Hale; P. Vergez; M.J. Meerman; Y. Hashida (2004) *Kalman Filtering and the Attitude Determination and Control Task*. Department of Astronautics, US Air Force Academy, CO, USA
 - [18] R.M. Plessis (1997) *Poor Man's Explanation of Kalman Filtering or How I Stopped Worrying and Learned to Love Matrix Inversion*. Rockwell International, CA, USA.
 - [19] Y. Hashida (2004) *ADCS for Future UoSat Standard Platform: REVISION*. SSTL Internal Technical Note, SSTL, Guilford, Surrey, United Kingdom.
 - [20] E.J. Lefferts; F.L. Markley; M.D. Shuster (1982) *Kalman Filter for Spacecraft Attitude Estimation*. Journal of Guidance, Control and Dynamics, Vol. 5, No. 5, pp. 417.
 - [21] F.L. Markley (2004) *Attitude Estimation or Quaternion Estimation?*. Journal of Astronautical Sciences, Vol. 51, No. 1, pp. 75-101.