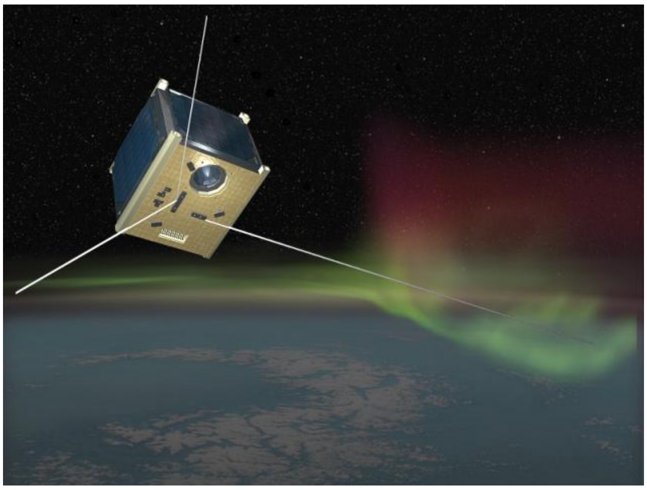
SwissCube Attitude Determination Algorithm Design and Validation

Master Project



Student: Jordi Martin Benet

MER: Dr. Denis Gillet
Supervisor: Dr. Philippe Müllhaupt
ULB Supervisor: Prof. André Preumont

Ecole Polytechnique Fédérale de Lausanne
Laboratoire d'Automatique

Lausanne, July 27, 2007

Contents

1 Introduction

The SwissCube is a small satellite in the CubeSat format designed by students from universities and high schools in Switzerland. Its main characteristics are a mass of 1 kg and a volume of $10cm \times 10cm \times 10cm$. This document is a report for a Master project and adresses the problem of attitude determination. The work is a continuation of the Controller Design [5] and its aim is to design an estimator for the satellite.

A brief overview of the ADCS subsystem, its components and the satellite's dynamics is presented in the first section. A review of the different methods commonly used in satellite attitude estimation and an introduction to the estimation theory is also given. The performance analysis for the different algorithms is the subject of the last section.

2 General Overview

2.1 Overview

In the context of spacecraft, **attitude control** is the control of the angular position and rotation of the spacecraft, either relative to the object that it is orbiting, or relative to the celestial body. On the surface of Earth there are straightforward references to determine a vehicle's position, whereas in space the Attitude Determination and Control System (ADCS) has the task to monitor the attitude of the satellite by combining data from different systems. Through the actuators, the ADCS can correct the angular position and speed.

Figure 1 is a global view of the ADCS system architecture. The subsystems can be arranged in two layers : hardware and software. The hardware layer includes the sensors, the microcontroller and the actuators. The software layer is constituted by the Earth Magnetic Field Model, the orbit propagator, the control algorithms and the determination algorithms.

The sensors used for the ADCS system are :

- A three-axis magnetometer to measure the Earth's magnetic field intensity and direction.
- A three-axis gyroscope setup to measure the spinning rate for each axis.
- Six Sun Sensors to indicate the direction of the sun.
- Temperature sensors to compensate the drift of the other sensors.

The actuators are :

- Three magnetotorquers (coils) to produce a torque thanks to their interaction with the Earth's magnetic field.
- (Passive actuators such as permanent magnets)
- (An Inertia wheel)

The system can be separated in four main parts :

- A propagator to compute the orbital position of the satellite.
- An Earth's magnetic field model to compute the magnetic field local intensity and direction.
- The control algorithm.
- The determination algorithm to filter the sensors measurements.

This report will present the design and evaluation of the determination algorithm. The analysis of the other subsystems performed by other students are presented in [5], [7], [8], [12], [13] and [14].

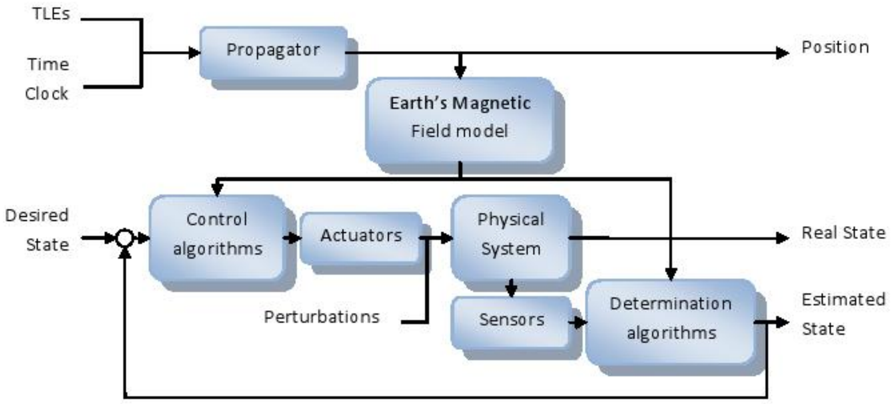


Figure 1: Overall view of the ADCS system

2.2 Model Validation

A model of the satellite dynamics was established in [5]. From there, the dynamics will be considered in a non-inertial frame without the inertia wheel. The orbital referential frame (ORF) speed will be considered constant ($\dot{\vec{\omega}}_o = 0$).

$$\boxed{\begin{array}{c} \dot{\vec{\omega}} = J^{-1}\vec{T} - J^{-1}(\vec{\omega} \times \vec{u}) + J^{-1}G\Delta[\vec{\omega}_o, \vec{u}]\mathbf{q} - \vec{s} \\ \\ \dot{\mathbf{q}} = \frac{1}{2}G^T\vec{\omega} \\ \\ \vec{u} := J(\vec{\omega} + R^T\vec{\omega}_o) \\ \vec{s} := (R^T\vec{\omega}_o) \times \vec{\omega} \end{array}} \tag{1}$$

The following table shows the symbols used and their meaning

$\vec{\omega}_o$	Orbital referential frame (ORF) speed
$\vec{\omega}$	Body rotational speed, relative to inertial referential
$\vec{T}$	Torque applied in the body referential
J	Body inertia
$\mathbf{q}$	Unit quaternion vector[1]
R	Quaternion rotation matrix

where

$$\Delta[\vec{v}, \vec{w}] = \begin{pmatrix} \vec{w} \cdot \vec{v} & (\vec{w} \times \vec{v})^T \\ \vec{w} \times \vec{v} & \vec{w}\vec{v}^T + \vec{v}\vec{w}^T - \vec{w} \cdot \vec{v}\ I_3 \end{pmatrix} \tag{2}$$

$$G = \begin{pmatrix} -q_1 & q_0 & q_3 & -q_2 \\ -q_2 & -q_3 & q_0 & q_1 \\ -q_3 & q_2 & -q_1 & q_0 \end{pmatrix} \tag{3}$$

[1] Note that the quaternions have 4 components and by imposing a norm constraint, $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$ they represent the 3 degrees of freedom in attitude.

The model can also be described as :

$$\begin{bmatrix} \dot{\omega}_1 \\ \dot{\omega}_2 \\ \dot{\omega}_3 \\ \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \begin{bmatrix} f_1\left(\vec{\omega}, \mathbf{q}, \vec{T}\right) \\ f_2\left(\vec{\omega}, \mathbf{q}, \vec{T}\right) \\ f_3\left(\vec{\omega}, \mathbf{q}, \vec{T}\right) \\ f_4\left(\vec{\omega}, \mathbf{q}, \vec{T}\right) \\ f_5\left(\vec{\omega}, \mathbf{q}, \vec{T}\right) \\ f_6\left(\vec{\omega}, \mathbf{q}, \vec{T}\right) \\ f_7\left(\vec{\omega}, \mathbf{q}, \vec{T}\right) \end{bmatrix} \tag{4}$$

Before the design of the estimator is started, certain aspects of the model require additional tests. The linearized version of SwissCube's dynamics is critical in order to achieve performance in the estimator. At first, the jacobian is obtained by symbolic computation in MATLAB from the nonlinear equations (1). The linearized model provided good performances compared to the nonlinear model, as illustrated in figure 2.2.

$$\begin{bmatrix} \dot{\vec{\omega}} \\ \dot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_{1..3}\left(\vec{\omega},\mathbf{q},\vec{T}\right)}{\partial \vec{\omega}} & \frac{\partial f_{1..3}\left(\vec{\omega},\mathbf{q},\vec{T}\right)}{\partial \mathbf{q}} \\ \frac{\partial f_{4..7}\left(\vec{\omega},\mathbf{q},\vec{T}\right)}{\partial \vec{\omega}} & \frac{\partial f_{4..7}\left(\vec{\omega},\mathbf{q},\vec{T}\right)}{\partial \mathbf{q}} \end{bmatrix} \Bigg|_{\substack{\vec{\omega} = \vec{\omega}_0 \\ \mathbf{q} = \mathbf{q}_0 \\ \vec{T} = \vec{T}_0}} \begin{bmatrix} \vec{\omega} \\ \mathbf{q} \end{bmatrix} + \begin{bmatrix} \frac{\partial f_{1..7}\left(\vec{\omega},\mathbf{q},\vec{T}\right)}{\partial \vec{T}} \end{bmatrix} \Bigg|_{\substack{\vec{\omega} = \vec{\omega}_0 \\ \mathbf{q} = \mathbf{q}_0 \\ \vec{T} = \vec{T}_0}} \vec{T} \tag{5}$$

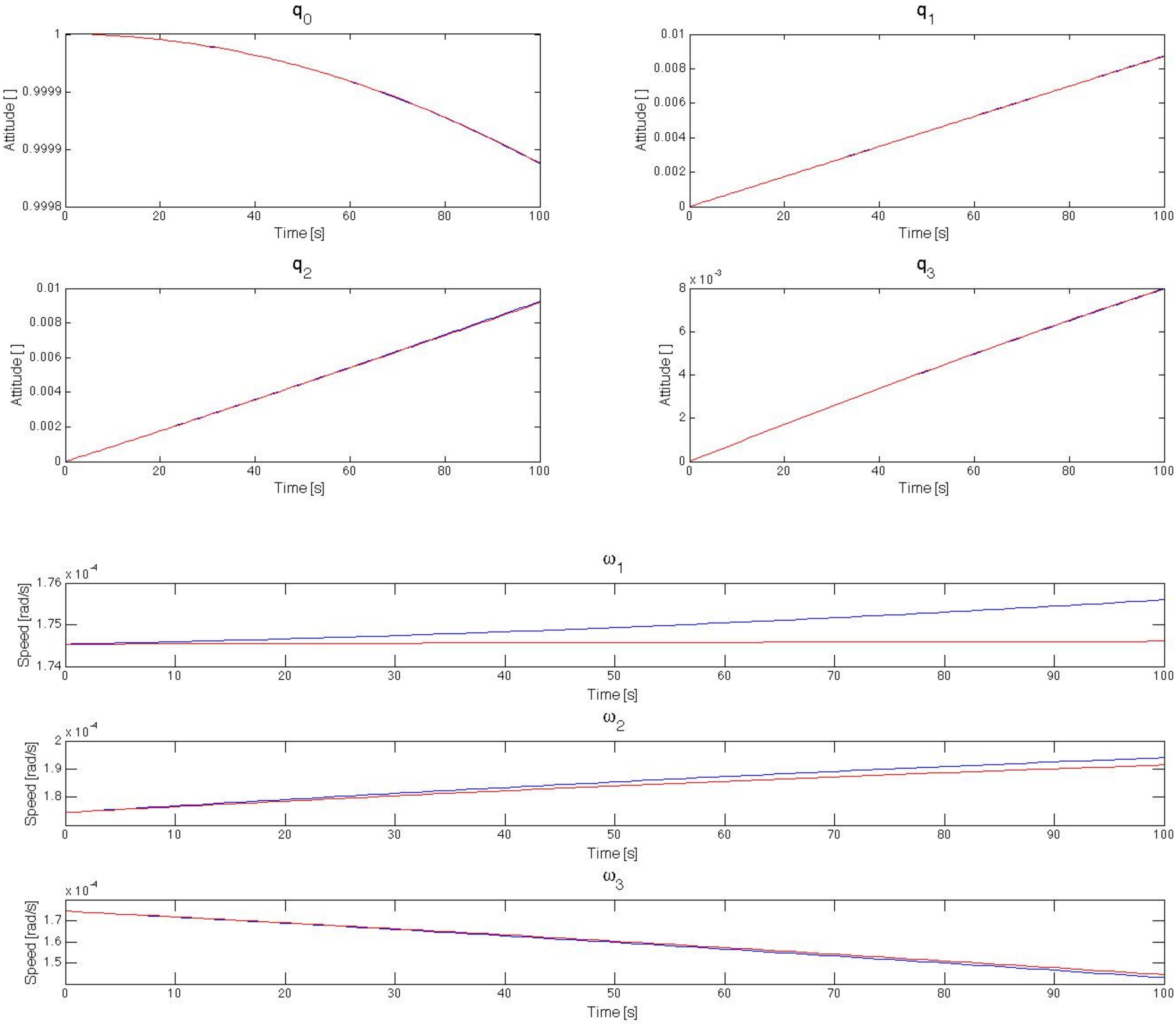


Figure 2: Linearised model obtained by straight symbolic derivation

3 Deterministic Attitude Estimation

Two classes of attitude determination algorithms will be considered : deterministic and recursive ones. The first only accounts for the present information handled by the sensors. On this section there will be a brief overview of the TRIAD algorithm and of Wahba's problem.

3.1 Introduction

The different satellite's sensor's collect informations on the physical environment. However, they only measure a physical magnitude that depends on the attitude. For the SwissCube, the magnetometers and the sun sensors collect data on the magnetic field components in the satellite's reference frame and the sun direction respectively. On-board, models compute both magnitudes in the ORF. These vector pairs are used by deterministic attitude estimation algorithms to produce an expression of the attitude in the form of a Direction Cosine Matrix or normed quaternion.

One of the simplest algorithms for attitude estimation, when given two pairs of measurements vector, is the TRIAD algorithm. More algorithm's are derived from the solutions to a least square problem such as the one proposed by Wahba [17].

3.2 TRIAD Algorithm

The TRIAD algorithm is by far the oldest and simplest algorithm used for attitude determination. It consists in constructing two orthonormal bases using two pairs of vector measurements : two in the orbital reference frame, noted r_1 and r_2, and two in the body reference frame, noted b_1 and b_2. b_i and r_i represent the same magnitude expressed in a different referential. It is assumed that the first vector measurements (r_1 and b_1) are more accurate.

The following equations are used to build $[t_{1b} \quad t_{2b} \quad t_{3b}]$, the basis attached to the body referential and $[t_{1r} \quad t_{2r} \quad t_{3r}]$ the basis attached to the orbital referential.

$$\begin{aligned} t_{1b} &= \tfrac{b_1}{|b_1|} & t_{1r} &= \tfrac{r_1}{|r_1|} \\ t_{2b} &= \tfrac{b_1 \times b_2}{|b_1 \times b_2|} & t_{2r} &= \tfrac{r_1 \times r_2}{|r_1 \times r_2|} \\ t_{3b} &= t_{1b} \times t_{2b} & t_{3r} &= t_{1r} \times t_{2r} \end{aligned} \tag{6}$$

It is well known that the relation between the vectors in the canonical base and an arbitrary base is the following

$$\begin{aligned} t_{ib} &= R_b e_i \\ r_{ir} &= R_r e_i \end{aligned} \tag{7}$$

where R_b and R_r are

$$\begin{aligned} R_b &= [t_{1b} \quad t_{2b} \quad t_{3b}] \\ R_r &= [t_{1r} \quad t_{2r} \quad t_{3r}] \end{aligned} \tag{8}$$

By taking the inverse basis change and by using the fact that R_b and R_r are orthonormal

$$\begin{aligned} e_i &= R_b^T t_{ib} \\ e_i &= R_r^T t_{ir} \end{aligned} \tag{9}$$

which yields

$$R_b^T b_i = R_r^T r_i \Longleftrightarrow b_i = R_b R_r^T r_i. \tag{10}$$

Therefore the attitude matrix is obtained through the multiplication of both base change matrices

$$D = [t_{1b} \quad t_{2b} \quad t_{3b}] \, [t_{1r} \quad t_{2r} \quad t_{3r}]^T . \tag{11}$$

3.3 Wahba's Problem

Wahba's problem consists in finding a least squares estimate of the rotation matrix R, which carries the known orbital reference france into the body reference frame [17]. Given two sets of n vectors $\{b_1, b_2, \cdots, b_n\}$ and $\{r_1, r_2, \cdots, r_n\}$, where $n > 2$, by minimizing the loss function L(R)

$$L(R) = \frac{1}{2} \sum_{i=1}^{n} a_i \left| b_i - R r_i \right| , \tag{12}$$

subject to the constraints that R be orthogonal and that $\det(R) = 1$. The nonnegative weights a_i are used for weighting the vector pairs. Wahba's loss function can also be written as

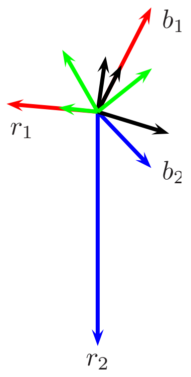


Figure 3: The TRIAD algorithm receives two vector pairs and build two orthonormal bases. $[t_{1b} \quad t_{2b} \quad t_{3b}]$ are represented in black and $[t_{1r} \quad t_{2r} \quad t_{3r}]$ in green

$$\begin{array}{l} L(R) = \sum_{i=1}^{n} a_i - tr(RB^T) \\ \\ B = \sum_{i=1}^{n} a_i b_i r_i^T \end{array} . \tag{13}$$

(13) can be minimized by maximizing $tr(RB^T)$ according to [4]. Note that in the present formulation, Wahba's problem is a single-frame attitude determination problem; it assumes that the vector measurements have been obtained for a constant attitude. The estimate takes place on a single time point and excludes the dynamics.

3.3.1 Davenport's q-method

In 1968, Davenport devised a method for computing the optimal quaternion **q** corresponding to the best least squares estimate of the rotation matrix R. The method may be summarized as follows. Given two sets of simultaneous vector observations $\{b_1, b_2, \cdots, b_n\}$, $\{r_1, r_2, \cdots, r_n\}$ and the corresponding weights a_i construct the 4×4 matrix K

$$K \equiv \begin{bmatrix} B + B^T - I_{3\times3} tr(B) & z \\ \\ z & tr(B) \end{bmatrix} \qquad z = \begin{bmatrix} B_{23} - B_{32} \\ B_{31} - B_{13} \\ B_{12} - B_{21} \end{bmatrix} \tag{14}$$

B is defined in (13). According to [10] the eigenvector of K with the largest eigenvalue will in this way give the optimal quaternion representing the rotation

$$K\mathbf{q}_{opt} = \lambda_{max}\mathbf{q}_{opt} \tag{15}$$

The key part of the algorithm consists in determining the eigenvector corresponding to the eigenvalue λ_{max}.

3.3.2 Optimal Two Observation Quaternion Estimation

According to [9], the Optimal Two Observation Quaternion Estimation Method has lower computational requirements and equivalent accuracy compared to the **q**-method. It uses three vector pairs for the determination of the attitude as a quaternion; the measurement vectors in the satellite referential frame, the model vectors in the orbital referential frame and two orthonormal vectors generated from the previous pairs as follows

$$r_n = \frac{r_1 \times r_2}{|r_1 \times r_2|} \qquad b_n = \frac{b_1 \times b_2}{|b_1 \times b_2|} \tag{16}$$

The transformation between the ORF to the SRF should include a rotation around r_n of an angle ϕ_r, followed by a rotation around b_n of an angle ϕ_b. ϕ is defined as the composition of ϕ_r and ϕ_b, which minimizes the following loss function

$$L(R) = a_1 + a_2 - \frac{\alpha\cos(\phi) + \beta\sin(\phi)}{1 + b_n \cdot r_n} \tag{17}$$

where α and β are given by

$$\begin{aligned} \alpha &= (1 + b_n \cdot r_n)(a_1 b_1 \cdot r_1 + a_2 b_2 \cdot r_2) + (b_n \times r_n)(a_1 b_1 \times r_1 + a_2 b_2 \times r_2) \\ \beta &= (b_n + r_n)(a_1 b_1 \times r_1 + a_2 b_2 \times r_2) \end{aligned} \tag{18}$$

The optimal rotation quaternion $\mathbf{q}_{opt}$ is expressed as follows

$$\mathbf{q}_{opt} = \begin{cases} \frac{1}{2\sqrt{\gamma(\gamma+\alpha)(1+b_n\cdot r_n)}} \begin{bmatrix} (\gamma+\alpha)(b_n \times r_n) + \beta(b_n + r_n) \\ (\gamma+\alpha)(1 + b_n \cdot r_n) \end{bmatrix} & \text{for } \alpha \geq 0 \\ \\ \frac{1}{2\sqrt{\gamma(\gamma-\alpha)(1+b_n\cdot r_n)}} \begin{bmatrix} \beta(b_n \times r_n) + (\gamma-\alpha)(b_n + r_n) \\ \beta(1 + b_n \cdot r_n) \end{bmatrix} & \text{for } \alpha < 0 \end{cases} \tag{19}$$

with $\gamma = \sqrt{\alpha^2 + \beta^2}$. The two weighting factors a_1 and a_2 are determined a priori based on the inverse variances of the random measurement errors.

4 Kalman Filter

In the previous section, Wahba's problem for the attitude estimation and various solutions were introduced. However, deterministic algorithms are often only used as a backup solution in the ADCS. Recursive estimators are often more convenient since they do not require storage of past data and allow real-time processing of new incoming observations. Recursive algorithms can be classified in two families: Kalman filter based attitude and estimators recursive quaternion estimators derived from the **q**-method. The first family will be introduced in this section, the second will be treated in a forthcoming section.

The Kalman filter is a set of equations that addresses the general problem of estimation. It is a very powerful tool that supports estimation of the past, present and future states of a system corrupted by a stochastic noise. This introduction includes a description of the basic discrete Kalman filter, used in linear systems, and its extension to the nonlinear field.

4.1 Basic discrete Kalman filter

4.1.1 Introduction

The following section about the Kalman Filter is based on [18].

Kalman filters are based on linear dynamical systems discretised in the time domain. The state of the system is represented as a vector of real numbers. At each time update, a linear operator is applied to the state, generating the new state with some noise added. Some information on the perturbations and the controls applied to the system can be added. The application of a second linear operator with noise generates the visible states of the system.

4.1.2 Hypotheses

Assuming the process to estimate can be modeled by the linear stochastic difference equation :

$$x_k = Ax_{k-1} + Bu_k + w_k \qquad x_k, x_{k-1} \in \Re^n \quad u_k \in \Re^l \tag{20}$$

x_k is the state of the system, generally hidden, related to the previous state x_{k-1} by A and the input u_k by B. With the measurements z_k

$$z_k = Hx_k + v_k \qquad z_k \in \Re^m \tag{21}$$

w_k and v_k represent respectively the model noise and the measurements noise. It may be assumed that they are uncorrelated white gaussian noises with the probability distributions

$$p(w) \sim N(0, Q)$$
$$p(v) \sim N(0, R)$$

Q and R are the *model noise covariance* and the *process noise covariance.* In practice A, B, H, Q and R can change at each time step; however, the case where they are constant will be considered for simplicity of development.

We define $\hat{x}_k^-$ as the *a priori* state estimation, computed from the previous state estimation $\hat{x}_{k-1}$. The *a posteriori* state estimation $\hat{x}_k$ corrected based on the measurement z_k. We define the *a priori* and *a posteriori* estimate errors

$$\begin{aligned} e_k^- &\equiv x_k - \hat{x}_k^- \\ e_k &\equiv x_k - \hat{x}_k \end{aligned} \tag{22}$$

and their covariance

$$\begin{aligned} P_k^- &\equiv E\left[e_k^- e_k^{-T}\right] \\ P_k &\equiv E\left[e_k e_k^T\right] \end{aligned} \tag{23}$$

4.1.3 Time Update and Measurement Update Equations

The Kalman filter estimates a process by using a form of feedback control: the filter estimates the process state at some time and collects feedback in the form of measurements. As such, the equations for the Kalman filter fall into two groups : time update equations and measurements update equations. The figure 4 represents the cycle a Kalman filter follows.

The *a priori* state estimate and covariance is obtained by the time update equations

$$\begin{aligned} \hat{x}_k^- &= A\hat{x}_{k-1} + Bu_k \\ P_k^- &= AP_{k-1}A^T + Q \end{aligned} \tag{24}$$

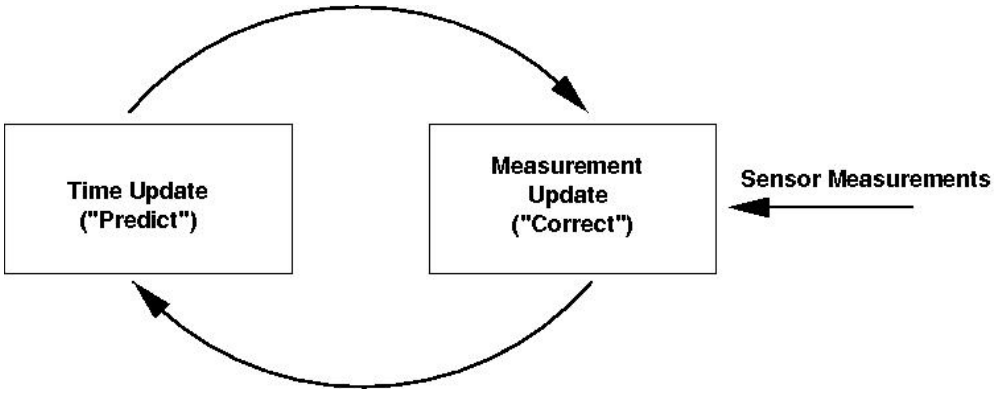


Figure 4: Kalman filter cycle

Equations (24) project forward in the future the state of the system and the covariance. The *a posteriori* state estimate and covariance is provided by the measurements update equations

$$\begin{aligned} K_k &= P_k^- H^T \left(H P_k^- H^T + R \right)^{-1} \\ \hat{x}_k &= \hat{x}_k^- + K_k \left(z_k - H \hat{x}_k^- \right) \\ P_k &= \left(I - KH \right) P_k^- \end{aligned} \tag{25}$$

The first task during the measurement update is to compute the Kalman gain, K_k . The next step is to actually measure the process to obtain z_k, and then to generate an *a posteriori* state estimate by incorporating the measurement. The final step is to obtain an *a posteriori* error covariance estimate P_k.

4.2 Extended Kalman Filter

4.2.1 Hypotheses

The basic Kalman filter is restricted to a linear assumption and it is a limiting constraint in reality. Most non trivial processes are nonlinear, either because of the process itself or because of the measurements. The Extended Kalman Filter is introduced to linearize about the current mean and covariance. Assuming that the internal state of the process can be represented by the state vector $x_k \in \Re^n$, but it is governed by the nonlinear stochastic equations

$$x_k = f(x_{k-1}, u_k, w_k) \tag{26}$$

With the measurements z_k

$$z_k = h(x_k, v_k) \tag{27}$$

where the random variables w_k and v_k again represent the process and measurement noises as in (20) and (21). In this case, the nonlinear function f relates the actual state x_k with its past state x_{k-1} and the control applied u_k. The nonlinear function h models the relation between the measurements z_k and the state of the process x_k.

In practice, w_k and v_k cannot be determined, but it is possible to approximate the state and measurement vectors with

$$\begin{gathered} \tilde{x}_k = f(\hat{x}_{k-1}, u_k, 0) \\ \tilde{z}_k = h(\tilde{x}_k, 0) \end{gathered} \tag{28}$$

$\hat{x}_{k-1}$ is some *a posteriori* estimate of the state (from a previous time step k). To estimate a process with nonlinear difference and measurement relationships, it necessary to write new governing equations that linearize an estimate about (28)

$$\begin{gathered} x_k \approx \tilde{x}_k + A(x_{k-1} - \hat{x}_{k-1}) + W w_k \\ z_k \approx \tilde{z}_k + H(x_k - \tilde{x}_k) + V v_k \end{gathered} \tag{29}$$

where

x_k, z_k : actual state and measurements vector

$\tilde{x}_k$, $\tilde{z}_k$: approximate state and measurements given by equations (28)

$\hat{x}_k$: estimate of the state at step k

w_k, v_k : process and measurement noise at step k

A : Jacobian matrix of partial derivatives of f with respect to x

$$A_{[i,j]} = \left.\frac{\partial f_i}{\partial x_j}\right|_{\hat{x}_k, u_k, w=0}$$

W : Jacobian matrix of partial derivatives of f with respect to w,

$$W_{[i,j]} = \left.\frac{\partial f_i}{\partial w_j}\right|_{\hat{x}_k, u_k, w=0}$$

H : Jacobian matrix of partial derivatives of h with respect to x,

$$H_{[i,j]} = \left.\frac{\partial h_i}{\partial x_j}\right|_{\tilde{x}_k, v=0}$$

V : Jacobian matrix of partial derivatives of h with respect to v,

$$V_{[i,j]} = \left.\frac{\partial h_i}{\partial v_j}\right|_{\tilde{x}_k, v=0}$$

By defining a new notation for the prediction error and the measurement residual

$$\begin{aligned} \tilde{e}_{x_k} &\equiv x_k - \tilde{x}_k \\ \tilde{e}_{z_k} &\equiv z_k - \tilde{z}_k \end{aligned} \tag{30}$$

4.2.2 Time Update and Measurement Update Equations

In practice, $\tilde{e}_{x_k}$ is not measurable because it relies on the knowledge of x_k which is normally hidden within the process. However, through the measures z_k, the quantity $\tilde{e}_{z_k}$ can be quantified. By combining (29) and (30) the equations governing the error process are obtained

$$\begin{aligned} \tilde{e}_{x_k} &\approx A(x_{k-1} - \hat{x}_{k-1}) + \epsilon_k \\ \tilde{e}_{z_k} &\approx H\tilde{e}_{x_k} + \eta_k \end{aligned} \tag{31}$$

where ϵ_k and η_k represent new independent random variables having zero mean and covariance matrices WQW^T and VRV^T, with Q and R as in 4.1.2.

$$p(\epsilon_k) \sim N(0, WQW^T)$$

$$p(\eta_k) \sim N(0, VRV^T)$$

Equations (31) are linear and similar to the initial formulation for the basic discrete Kalman filter (20) and (21). Therefore, to estimate the non-linear process, a second Kalman filter is set to estimate the prediction error $\tilde{e}_{x_k}$ as a hidden state. This estimate is called $\hat{e}_k$, which coupled with the first equation of (30), provides the *a posteriori* state estimate of the original process

$$\hat{x}_k = \tilde{x}_k + \hat{e}_k \tag{32}$$

and the distribution of the prediction error is normal with variance equal to $E[\tilde{e}_{x_k} \tilde{e}_{x_k}^T]$

$$p(\tilde{e}_{x_k}) \sim N(0, E[\tilde{e}_{x_k} \tilde{e}_{x_k}^T])$$

By supposing the predicted value of $\hat{e}_k$ equal, the Kalman filter provides the value of $\hat{e}_k$ based on the error of estimation of the measures $\tilde{e}_{z_k}$

$$\hat{e}_k = K_k \tilde{e}_{z_k} \tag{33}$$

by replacing (33) in (32) and using (30), it is to be seen that the second (hypothetical) Kalman filter is not necessary

$$\begin{aligned} \hat{x}_k &= \tilde{x}_k + K_k \tilde{e}_{z_k} \\ &= \tilde{x}_k + K_k (z_k - \tilde{z}_k) \end{aligned} \tag{34}$$

We are now able to rewrite the basic discrete Kalman filter time update (28) and measurement update equations (29) to account for the modified estimation process. The time update equations of the EKF are

$$\begin{aligned} \hat{x}_k^- &= f(\hat{x}_{k-1}, u_k, 0) \\ P_k^- &= A_k P_{k-1} A_k^T + W_k Q W_k^T \end{aligned} \tag{35}$$

and the measurement update equations

$$\begin{aligned} K_k &= P_k^- H_k^T \left(H_k P_k^- H_k^T + V_k R V_k^T \right)^{-1} \\ \hat{x}_k &= \hat{x}_k^- + K_k \left(z_k - h(\hat{x}_k^-, 0) \right) \\ P_k &= \left(I - K_k H_k \right) P_k^- \end{aligned} \tag{36}$$

An important feature of the EKF is that the Jacobian H_k in the equation for the Kalman gain K_k serves to correctly propagate or "magnify" only the relevant component of the measurement information. It may be noted that both versions of the filter have a similar implementation. They follow the prediction-correction cycle illustrated in figure 4.

4.3 Attitude determination using the Kalman Filter

There are several approaches to determine the attitude of an autonomous vehicle. The simplest way is to create a Kalman filter based on the system dynamics. This approach was adopted in the nCube design [11] and [15]. However, the solution requires to calculate online the nonlinear functions that give the state estimates. A second approach exposed in [2] proposes to estimate the direction cosine matrix , which is the transformation matrix between some reference coordinate system and the system whose attitude is to be determined.

The measurements are given in two cartesian coordinate systems. System v is attached to the vehicle and system u is bound to the reference coordinate system. The measurements in the system u and v result in sequences of vectors[1] $\{u_k\}_{k=1\ldots n}$ and $\{v_k\}_{k=1\ldots n}$ respectively. The aim is to compute $\hat{\mathbf{q}}$, the minimum variance estimate of $\mathbf{q}$, where $\mathbf{q}$ is the quaternion representing the rotation between u and v.

The relation between u and v can be described by the direction cosine matrix (DCM). The DCM can be expressed in terms of the quaternion of rotation $\mathbf{q}$ through a nonlinear relation. A filter similar to the extended Kalman filter (EKF) is used to estimate the difference between the actual quaternion and its estimate. Each newly updated estimate of this difference will be added to the quaternion estimate to form the newly updated (or current) whole quaternion estimate. As a first step in the algorithm development, the linear relations between $\delta\mathbf{q}$, u_k, and v_k are derived, where $\delta\mathbf{q}$ is the difference between $\mathbf{q}$ and its estimate $\hat{\mathbf{q}}$, .

The DCM[2] expression in terms of the quaternion $\mathbf{q} = [q_0 \quad q_1 \quad q_2 \quad q_3]^T$ is

$$D(\mathbf{q}) = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \tag{37}$$

[1] u_i and $v_i \in \Re^3$

[2] $D(\mathbf{q})$ is noted in section 2 as $R(\mathbf{q})$. However, in the current section R stands for the measurements noise covariance.

Supposing that $\mathbf{q}$ and $D(\mathbf{q})$ are known, a first-order Taylor series expansion may be used to compute $D(\mathbf{q}+\delta\mathbf{q})$

$$D(\mathbf{q}+\delta\mathbf{q}) \sim D(\mathbf{q}) + \sum_{j=0}^{3} \left.\frac{\partial D}{\partial q_j}\right|_{\mathbf{q}} \delta\mathbf{q}_j \tag{38}$$

Note that

$$A_j(\mathbf{q}) = \left.\frac{\partial D}{\partial q_j}\right|_{\mathbf{q}} \quad , \qquad j = 0,1,2,3$$

with

$$\begin{aligned} A_0(\mathbf{q}) &= 2\begin{bmatrix} q_0 & q_3 & -q_2 \\ -q_3 & q_0 & q_1 \\ q_2 & -q_1 & q_0 \end{bmatrix} & A_1(\mathbf{q}) &= 2\begin{bmatrix} q_1 & q_2 & q_3 \\ q_2 & -q_1 & q_0 \\ q_3 & -q_0 & -q_1 \end{bmatrix} \\ A_2(\mathbf{q}) &= 2\begin{bmatrix} -q_2 & q_1 & -q_0 \\ q_1 & q_2 & q_3 \\ q_0 & q_3 & -q_2 \end{bmatrix} & A_3(\mathbf{q}) &= 2\begin{bmatrix} -q_3 & q_0 & q_1 \\ -q_0 & -q_3 & q_2 \\ q_1 & q_2 & q_3 \end{bmatrix} \end{aligned} \tag{39}$$

Supposing $\hat{\mathbf{q}}_k^-$ the estimate of $\mathbf{q}$ at step k is known and a new pair of measurements u_k and v_k is obtained. The new error free measurements satisfy the relation

$$v_{0,k} = D(\mathbf{q}_k)u_{0,k} \tag{40}$$

The measurements of the vectors $u_{0,k}$ and $v_{0,k}$ are corrupted by the noises $n_{u,k}$ and $n_{v,k}$. It may be assumed that both noises are white and Gaussian. Their covariance matrices are denoted R_u and R_v. The relation between noise free measurements and real measurements is

$$\begin{aligned} u_k &= u_{0,k} + n_{u,k} \\ v_k &= v_{0,k} + n_{v,k} \end{aligned} \tag{41}$$

and $\mathbf{q}_k$ may be expressed as

$$\mathbf{q}_k = \hat{\mathbf{q}}_k^- + \delta\mathbf{q}_k \tag{42}$$

Substitution of (41) and (42) into (40) yields

$$v_k = D(\hat{\mathbf{q}}_k^- + \delta\mathbf{q}_k)(u_k - n_{u,k}) + n_{v,k} \tag{43}$$

By using (38) to rewrite (43) as

$$\begin{aligned} v_k - D(\hat{\mathbf{q}}_k^-)u_k = & \left[\textstyle\sum_{j=0}^{3} A_j(\hat{\mathbf{q}}_k^-)\delta\mathbf{q}_{k,j}\right] u_k \\ & - \left[\textstyle\sum_{j=0}^{3} A_j(\hat{\mathbf{q}}_k^-)\delta\mathbf{q}_{k,j}\right] n_{u,k} \\ & -D(\hat{\mathbf{q}}_k^-)n_{u,k} + n_{v,k} \end{aligned} \tag{44}$$

The second term on the right-hand side of (44) is a second-order term which can be omitted, while first term on the right side may be written as

$$\left[\sum_{j=0}^{3} A_j(\hat{\mathbf{q}}_k^-)\delta\mathbf{q}_{k,j}\right] u_k = H_k^-(\hat{\mathbf{q}}_k^-, u_k)\delta\mathbf{q}_k \tag{45}$$

with

$$\begin{aligned} H_k^- &= [h_1 \quad h_2 \quad h_3 \quad h_4] \\ h_j &= A_j(\hat{\mathbf{q}}_k^-)u_k \end{aligned}$$

A new noise variable n_k is introduced

$$n_k \equiv n_{v,k} - D(\hat{\mathbf{q}}_k^-)n_{u,k} \tag{46}$$

and the estimate of the DCM

$$\hat{D}_k \equiv D(\hat{\mathbf{q}}_k)$$

n_k is a zero mean white noise whose covariance matrix R is given by

$$\begin{aligned} R_k &\equiv \text{Cov}\{n_k\} \\ &= R_{v,k} + D_k^- R_{u,k} D_k^{-T} \end{aligned} \tag{47}$$

By defining $\hat{e}_k$ as follows

$$\hat{e}_k \equiv v_k - \hat{D}_k v_k \tag{48}$$

by using (45), (46) and (48), (44) may be rewritten as

$$\hat{e}_k = H_k^- \delta\mathbf{q}_k + n_k \tag{49}$$

Equation (49) is a linear relationship between the data vector $\hat{e}_k$ and the unknown vector $\delta\mathbf{q}_k$ which is the difference between the quaternion of rotation and its estimate. It may be noted that e_k, H_k^- and n_k depend on the current estimate of the quaternion, a feature of the EKF.

If the coordinate system v rotates with respect to u, then the change of the quaternion $\mathbf{q}$ must be taken in account. It is well known that the rate of change $\dot{\mathbf{q}}$ is related to $\mathbf{q}$ by the following relation

$$\dot{\mathbf{q}} = \Omega \mathbf{q} \tag{50}$$

where

$$\Omega = \frac{1}{2} \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} \tag{51}$$

Ω elements are the components of the vector $\vec{\omega}' = [\omega_x \quad \omega_y \quad \omega_z]^T$. The true quaternion propagates according to (50), however the estimated quaternion propagates according to

$$\dot{\hat{\mathbf{q}}} = \tilde{\Omega} \hat{\mathbf{q}} \tag{52}$$

where the matrix $\tilde{\Omega}$ has the same expression as Ω except that its entries are $\tilde{\vec{\omega}} = [\tilde{\omega}_x \quad \tilde{\omega}_y \quad \tilde{\omega}_z]^T$, the measured angular rates in the body referential. Since a noise contaminates our measurements

$$\tilde{\vec{\omega}} = \vec{\omega} + n_\omega \tag{53}$$

n_ω is the noise vector for the angular rate. (50) may be expressed to account for the noise (53), as follows

$$\dot{\mathbf{q}} = (\tilde{\Omega} + \delta\Omega)\mathbf{q} \tag{54}$$

where $\delta\Omega$ is similar to (51), with n_ω as its elements. Using the second relation of (1), (54) may also be noted as

$$\dot{\mathbf{q}} = \tilde{\Omega}\mathbf{q} - \frac{1}{2} G^T n_\omega \tag{55}$$

Where

$$G^T = \begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix}$$

By substracting (52) to (55), the following dynamics equation is obtained for $\delta\dot{\mathbf{q}}$

$$\delta\dot{\mathbf{q}} = \tilde{\Omega}\delta\mathbf{q} + Ln_{\omega} \tag{56}$$

With $L = -\frac{1}{2}G^T$. When (56) is discretized, the following difference equation is obtained for the propagation of $\delta\mathbf{q}$

$$\delta\mathbf{q}_k = \phi_{k-1}\delta\mathbf{q}_{k-1} + L_{k-1}n_{\omega,k-1} \tag{57}$$

Note that ϕ_{k-1} is a function of the measured angular rate vector $\tilde{\vec{\omega}}$ and L_{k-1} is a function of $\mathbf{q}_{k-1}$. Since $\mathbf{q}_{k-1}$ itself is not known its estimate $\hat{\mathbf{q}}_{k-1}$ is used to compute L_{k-1}. The latter is a well known characteristic of the EKF.

Following (57), $\delta\hat{\mathbf{q}}$ is propagated between measurements according to

$$\delta\hat{\mathbf{q}}_k = \phi_{k-1}\delta\hat{\mathbf{q}}_{k-1} \tag{58}$$

Equations (49) and (58) may be used to create an extended Kalman filter. $\hat{e}_k$ is the innovation term and $\delta\hat{\mathbf{q}}_k$ is the *a priori* estimation error. Therefore, the estimation error covariance matrix is propagated according to

$$P_k^- = \phi_{k-1}P_{k-1}\phi_{k-1}^T + L_{k-1}Q_{k-1}L_{k-1}^T \tag{59}$$

where $Q_{k-1} = \text{Cov}\{n_{\omega,k-1}\}$. From this point, it will be assumed that $n_{\omega,k-1}$ is a static noise. Moreover, (52) yields the following relation

$$\hat{\mathbf{q}}_k^- = \phi_{k-1}\hat{\mathbf{q}}_{k-1} \tag{60}$$

Equations (58)-(60) will be used to establish the time update equations and measurements update equations of the EKF. This leads to the following algorithm summarized in table 1

Time update	Measurements update
$\hat{\mathbf{q}}_k^- = \phi_{k-1}\hat{\mathbf{q}}_{k-1}$ $P_k^- = \phi_{k-1}P_{k-1}\phi_{k-1}^T + L_{k-1}Q_{k-1}L_{k-1}^T$	$K_k = P_k^- H_k^T \left(H_k P_k^- H_k^T + R_k\right)^{-1}$ $\delta\hat{\mathbf{q}}_k = K_k \hat{e}_k$ $\hat{\mathbf{q}}_k = \hat{\mathbf{q}}_k^- + \delta\hat{\mathbf{q}}_k$ $P_k = (I - K_k H_k) P_k^- \times$ $(I - K_k H_k) + K_k R_k K_k^T$

Table 1: Modified extended Kalman filter for attitude estimation

5 Recursive Quaternion Estimators

5.1 REQUEST algorithm

Davenport's $\mathbf{q}$-method is an algorithm determination and relies on the spectral decomposition of K for a single-frame attitude as defined in (14). As seen in a previous section, several solutions have been devised to compute effectively the eigenvalues and the corresponding optimal quaternion . However, the solutions to Wahba's problem require at least two sets of vector measurements, to be implementable. It is a limiting constraint for real systems as some of the sensors may become unavailable for short periods of time (e.g. the sun sensors). The REQUEST algorithm [3] constructs K recursively from a single pair of vector measurements.

It is well known that the body angular motion can be described in terms of the attitude quaternion by the differential equation

$$\dot{\mathbf{q}} = \frac{1}{2}\Omega\mathbf{q} \tag{61}$$

where Ω is a 4×4 skew symmetric matrix and is a function of the angular velocity of the body with respect to the reference frame, given in the body frame and denoted by $\vec{\omega}'$. Ω is defined as follows

$$\Omega = \begin{bmatrix} -[\vec{\omega}' \times] & \vec{\omega}' \\ \vec{\omega}'^T & 0 \end{bmatrix} \tag{62}$$

The solution of (61) in discrete time[3] is

$$\mathbf{q}_{k+1} = \phi_k \mathbf{q}_k \tag{63}$$

Based on equation (63) an optimal attitude prediction step is devised in terms of the matrix K. $K_{i/j}$ denotes the matrix representing the attitude at time t_i and constructed from the measurements up to t_j. Its propagation from t_k to t_{k+1} is formulated as

$$K_{k+1/k} = \phi_k K_{k/k} \phi_k^T \tag{64}$$

Assuming that a single pair of vector measurements $\{b_{k+1} \quad r_{k+1}\}$ is acquired at t_{k+1}, the corresponding matrix $K_{k+1/k+1}$ may be computed. First, define

[3] The matrix ϕ_k can only be estimated through a first order approximation for a given $\vec{\omega}'$

$$\begin{aligned} B_{k+1} &\equiv b_{k+1} r_{k+1}^T \qquad S_{k+1} \equiv B_{k+1} + B_{k+1}^T \\ z_{k+1} &\equiv b_{k+1} \times r_{k+1} \qquad \sigma_{k+1} \equiv tr(B_{k+1}) \end{aligned} \tag{65}$$

then, calculate δK_{k+1} as

$$\delta K_{k+1} = \begin{bmatrix} S_{k+1} - \sigma_{k+1} I & z_{k+1} \\ z_{k+1}^T & \sigma_{k+1} \end{bmatrix} \tag{66}$$

Denoting by a_i the scalar weighting coefficient of the i^{th} observation, the following scalars are recursively computed

$$m_0 = 0 \qquad m_{k+1} = m_k + a_{k+1} \tag{67}$$

and finally update $K_{k+1/k}$

$$K_{k+1/k+1} = \rho_{k+1} \frac{m_k}{m_{k+1}} K_{k+1/k} + \frac{a_{k+1}}{m_{k+1}} \delta K_{k+1} \tag{68}$$

The coefficients m_k are used to normalize the weights a_i, to maintain the largest eigenvalue of $K_{k+1/k+1}$ close to 1. If the matrix ϕ_k is error-free (perfect speed measurements), the coefficient ρ is set equal to 1. If noisy speed measurements are used for the propagation calculations, ρ is set between 0 and 1 for filtering purposes. It is plain to see that the choice of ρ is heuristic, making the filter suboptimal.

5.2 Optimal REQUEST

In the REQUEST update stage, the choice of the fading memory factor ρ_{k+1} is heuristic. Moreover, its determination is only based on the noise present in the propagation stage (speed measurement noise) and has no direct relation with the measurement noise. The REQUEST algorithm can be updated by computing an optimal ρ_{k+1} in the update stage. This value accounts for the various noises present in the measurements to improve accuracy in the estimation of the K matrix.

5.2.1 Prediction stage

It is required for $K_{k+1/k}$ to be linear in $K_{k/k}$ and to produce an unbiased estimate. These requirements yield the prediction formula of the REQUEST algorithm

$$K_{k+1/k} = \phi_k K_{k/k} \phi_k^T \tag{69}$$

Using the process equations for the K matrix 96, the prediction equation (69) and the definitions of the prediction errors 95, the error propagation equation is obtained

$$\Delta K_{k+1/k} = \phi_k \Delta K_{k/k} \phi_k^T + W_k \tag{70}$$

Knowing the propagation of the error, the error covariance propagation can be computed using

$$P_{k+1/k} = \phi_k P_{k/k} \phi_k^T + Q_k \tag{71}$$

Where Q_k is given in equation (107).

5.2.2 Measurement update stage

The update stage for the Optimal REQUEST is a slightly modified version of the update stage for the original REQUEST algorithm. The updated estimate $K_{k+1/k+1}$ in (68) is reshaped as a convex combination of the prediction estimate $K_{k+1/k}$ and the new observation δK_{k+1}

$$K_{k+1/k+1} = (1-\rho_{k+1})\frac{m_k}{m_{k+1}} K_{k+1/k} + \rho_{k+1}\frac{\delta m_{k+1}}{m_{k+1}} \delta K_{k+1} \tag{72}$$

where δm_{k+1} is a scalar weight and m_{k+1} is computed recursively using

$$m_{k+1} = (1-\rho_{k+1}) m_k + \rho_{k+1} \delta m_{k+1} \tag{73}$$

The error covariance propagation is derived in Annex B

$$P_{k+1/k+1} = \left((1-\rho_{k+1})\frac{m_k}{m_{k+1}}\right)^2 P_{k+1/k} + \left(\rho_{k+1}\frac{\delta m_{k+1}}{m_{k+1}}\right)^2 R_{k+1} \tag{74}$$

5.2.3 Optimal Gain

For each new processed observation, the estimation uncertainty should decrease considerably. According to [3], a cost function may be defined on the basis of the expectation of the error propagation equation

$$L_{k+1}(\rho_{k+1}) \equiv tr\left(E\left[\Delta K_{k+1/k+1} \Delta K_{k+1/k+1}^T\right]\right) = tr\left(P_{k+1/k+1}\right) \tag{75}$$

The design problem for the filter gain ρ_{k+1} reduces to solving the following minimization problem with respect to ρ_{k+1}

$$min\left\{L_{k+1} = tr\left(P_{k+1/k+1}\right)\right\} \tag{76}$$

Inserting (74) into the expression for L_{k+1} yields

$$L_{k+1}(\rho_{k+1}) = \left((1-\rho_{k+1})\frac{m_k}{m_{k+1}}\right)^2 tr\left(P_{k+1/k}\right) + \left(\rho_{k+1}\frac{\delta m_{k+1}}{m_{k+1}}\right)^2 tr\left(R_{k+1}\right) \tag{77}$$

A necessary condition for an extremum is

$$\begin{aligned} \tfrac{dL_{k+1}}{d\rho_{k+1}} &= 2\left[\left(\tfrac{m_k}{m_{k+1}}\right)^2 tr\left(P_{k+1/k}\right) + \left(\tfrac{\delta m_{k+1}}{m_{k+1}}\right)^2 tr\left(R_{k+1}\right)\right]\rho_{k+1} \\ &- 2\left(\tfrac{m_k}{m_{k+1}}\right)^2 tr\left(P_{k+1/k}\right) = 0 \end{aligned} \tag{78}$$

yielding the condition for ρ^*_{k+1} to be a minimum

$$\rho^*_{k+1} = \frac{m_k^2 tr\left(P_{k+1/k}\right)}{m_k^2 tr\left(P_{k+1/k}\right) + \delta m_{k+1}^2 tr\left(R_{k+1}\right)} \tag{79}$$

Note that ρ^*_{k+1} as computed from (79) lies in the interval $[0,1]$ for any $m_k \neq 0$ and $\delta m_k \neq 0$. It may be seen that the gain depends directly on the actual state estimation error covariance and indirectly on the measurement noise covariance through the modified noise matrix R_{k+1}. For a condition of low uncertainty in the measurements with respect to a high uncertainty in the a priory estimate, the gain stays close to 1 and the algorithm remains active. In the opposite condition, the gain tends towards 0 and small corrections only are introduced to the a priori estimate.

6 Simulation Study

6.1 Introduction

The aim of this section is to present the results obtained for the different estimation algorithms exposed previously. The deterministic algorithms are not treated. The tests could not be carried out as the models of the sun sensor's are not yet available. The Kalman Filter in its continuous version is used as a reference for the different recursive algorithms as it provides the best performances for the system. However, it is not usable on the ADCS controller due to limiting constraints in computational power.

6.2 Continuous EKF

6.2.1 Computational model

A continuous version of the EKF was built based on the equations (1) and (5). In practice, its implementation on the SwissCube satellite is impossible due to the large requirements in terms of computational power. For correct operation, the filter requires a numerical ordinary differential equation (ODE) solver to be implemented, working with sufficiently small integration step. However, it was designed to serve as a reference in terms of convergence time and tracking capabilities. The continuous Kalman Filter equations are found in [6]. A SIMULINK model was programmed in S-Function language to simulate the behaviour of the filter with different noise levels. Figure 5 is an overall picture of the model.

Three parameters influence the behaviour of the continuous Kalman Filter: the integration step used in the ODE solver, the modelling of the noise through R and Q, the operating conditions (noise levels and angular speed) and the initial conditions. To examine the performances of the filter developped, simulations were carried out with different parameter sets and noise conditions.

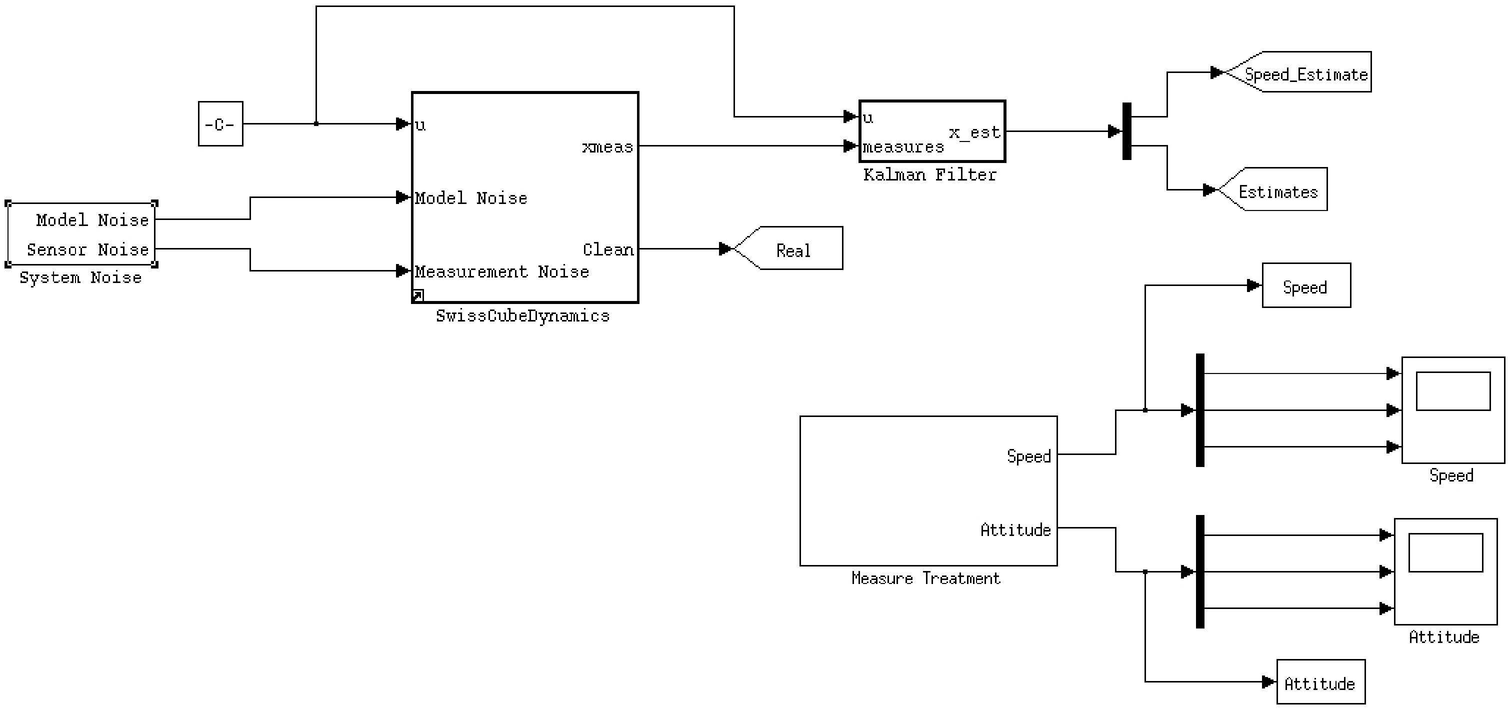


Figure 5: SIMULINK model for the Continuous EKF

6.2.2 Analysis of some results

For the purpose of comparing $\hat{\mathbf{q}}$ with the true quaternion, two types of estimation errors commonly used in quaternion estimators [3] were considered. The first is the multiplicative estimation error, denoted here by $\delta\mathbf{q}$ and is defined as $\delta\mathbf{q} = \mathbf{q}^* \otimes \hat{\mathbf{q}}$. The unit vector $\delta\mathbf{q}$ is itself a quaternion of rotation that represents the small rotation which brings the axis of the estimated body frame onto those of the true body frame. From the scalar component of $\delta\mathbf{q}$, denoted by δq, the value of the rotation angle $\delta\phi$ is extracted through the relation $\delta\phi = 2arccos(\delta q)$. The second type of estimation error is the norm of the additive estimation error, $\parallel \Delta\mathbf{q} \parallel$, which is defined as $\parallel \Delta\mathbf{q} \parallel = \parallel \mathbf{q} - \hat{\mathbf{q}} \parallel$.

In the simulations conducted, the body referential rotation speed, was chosen as a variable. The speed and attitude measurement noises were Gaussian zero-mean white noises with a standard deviation of 3.2×10^{-3} deg/s and 1×10^{-2} respectively. The speed and attitude model noises were Gaussian zero-mean white noises with a standard deviation of 1×10^{-3} deg/s and 1×10^{-4} respectively. Each simulated time span was 10 sec. Figure 6 depicts the multiplicative and additive estimation errors . The convergence time for the Kalman Filter is in average inferior to 1×10^{-3} s for the Continuous Extended Kalman Filter.

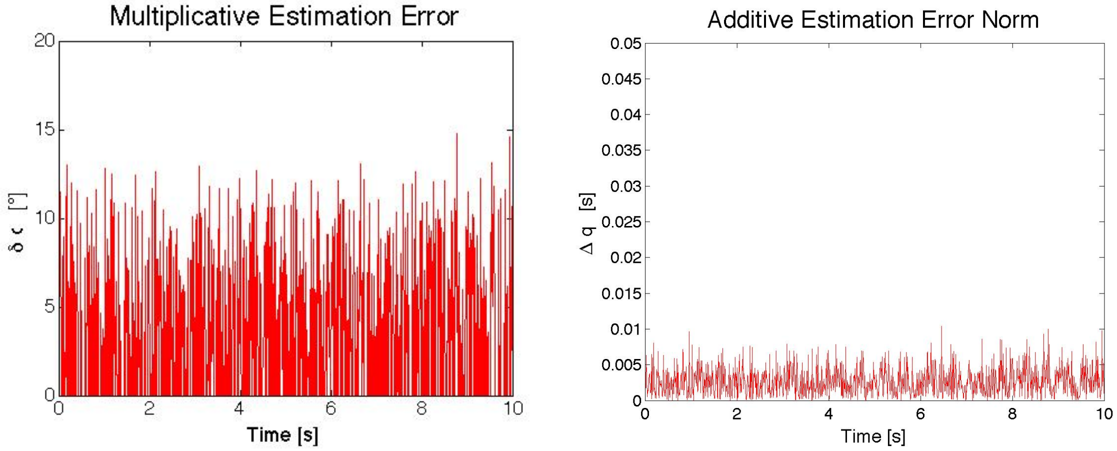


Figure 6: Multiplicative and Additive Estimation Errors for 5 deg/s

The results in table 2 illustrate the filter's errors in the operational range of speeds of the system. The additive and the multiplicative errors remain low despite the noisy inputs and no significant increase is observed at high speeds. The values obtained will be used as a reference for the other estimators proposed in this work.

	μ		σ	
Speed [deg/s]	Multiplicative [°]	Additive [ ]	Multiplicative [°]	Additive [ ]
5	1.89	2.45×10^{-3}	4.44	3.41×10^{-2}
2	1.53	1.99×10^{-3}	3.54	3.08×10^{-2}
1	1.44	1.34×10^{-3}	2.63	2.98×10^{-2}
0.5	1.76	1.24×10^{-3}	2.98	2.08×10^{-2}
0.1	1.44	1.46×10^{-3}	4.96	3.41×10^{-2}

Table 2: Statistical characteristics of the two criteria of estimation error for different speeds

6.2.3 Separation principle

For linear time invariant systems, the controller and the estimator may be designed separately under some assumptions. But the system is non-linear, which implies that compatibility tests are to be conducted with the controller designed in a previous project [5]. The controller is a PD which uses Earth's magnetic field and the magnetotorquers to generate a torque. The control law used is

$$\vec{C} = -k_v \vec{\omega}' - k_p \vec{q} \tag{80}$$

Where $\vec{\omega}'$ is the angular speed expressed in the body reference frame. $\vec{q}$ is the imaginary part of the attitude quaternion. k_v and k_p are respectively the derivative and the proportionnal terms. In order to reduce unnecessary energy consumption, the magnetic field generated by the magnetotorquers is chosen perpendicular to the local Earth magnetic field, which yields the following expression

$$\vec{M}' = \frac{\vec{B}'}{\vec{B}'^T \vec{B}'} \times (-k_v \vec{\omega}' - k_p \vec{q}) \tag{81}$$

The simulation conditions to test the compatibility of the controller were the same as those used in the point 6.2.2. The derivative and the proportional terms chosen were 10^{-5} $\frac{N.s}{rad}$ and 10^{-8} $N.m$, respectively. The figure 7 illustrates the results obtained in simulation with a starting speed of 0.1 deg/s .

Some “Burst-out” problems occured with the Kalman Filter when the system reached a steady state. In the final version of the filter, a reinitialisation was introduced every hour of simulated time to avoid those problems. The effects can be seen in the error norm graphic, where spikes appear every hour due to the filter modification. It should be noted that the local Earth magnetic field supplied to the controller is constant and therefore does not reflect reality.

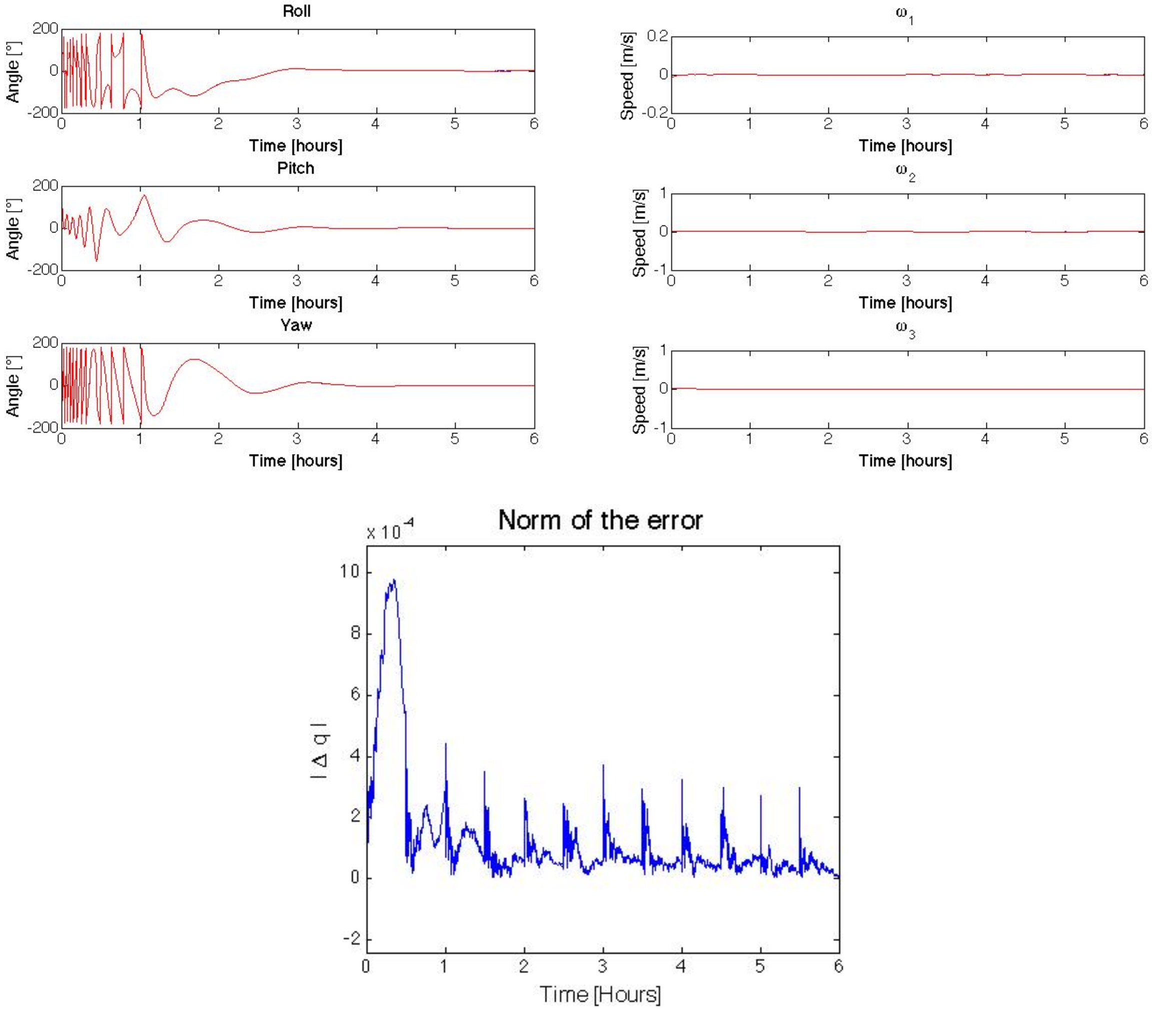


Figure 7: Simulation of the Continuous EKF combined with the controller. The top figures show the attitude and the speed. The bottom figure illustrates the additive estimation error norm.

6.3 Modified Discrete EKF

6.3.1 Computational model

For the modified EKF, the sensors as well as an orbit propagator and the earth magnetic field model were incorporated in the SIMULINK model. The sensors were modelled as a moving average coupled with a quantization and a saturation as developped in [8] and [13] (see Figure 8). While representing a rather good approximation of reality, it introduces additional non-linearities in the overall system. The Earth magnetic field is given by the look-up tables calculated during semester project [12]. It is a simplified model of the WMM-2005 (World Magnetic Model, a model used by international organisations) where the three components of the magnetic field are calculated in various points of a grid with a step of 5° in latitude and longitude. The overall precision of this approach is 200 nT, which is less than the theoretical precision of 1 μT achievable by the magnetometers. The orbit propagator chosen in [7] was included through the S-Function language of SIMULINK. An overall view is available in Figure 9

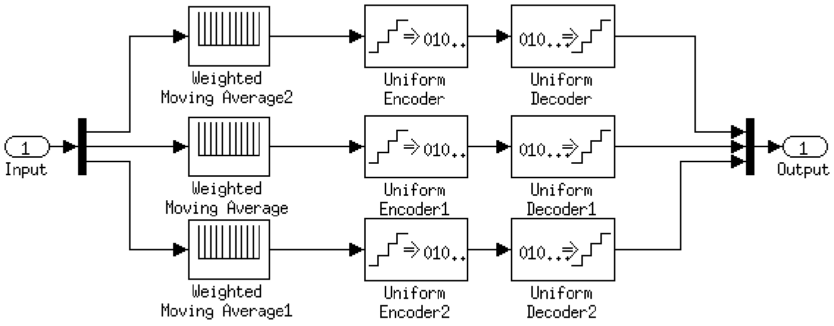


Figure 8: SIMULINK model for the sensors

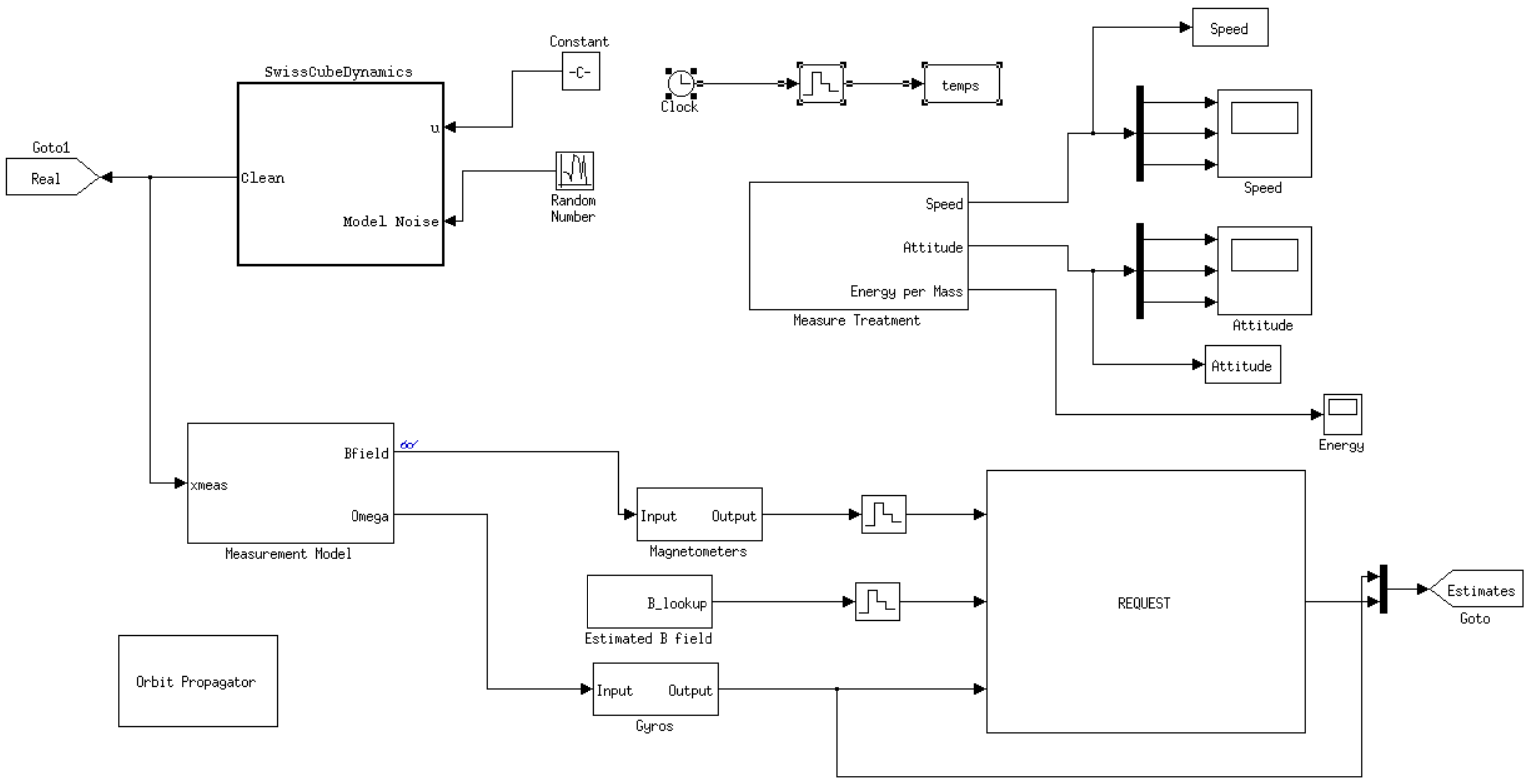


Figure 9: SIMULINK model for the Discrete EKF

6.3.2 Analysis of some results

If the sensor models are not included for the simulation of the system, the Modified Discrete EKF displays good characteristics. However, the addition of sensors makes the system unstable and makes it consequently unusable. This behaviour can be explained through the great sensitivity that the filter displays to the measurement noise. The quantization noise added by the introduction of the sensors is enough to destroy the filter stability as shown in figure 10. Even though stability for short periods of time can be achieved through tuning of the different filter parameters, unpredictable behaviours appear on the long term.

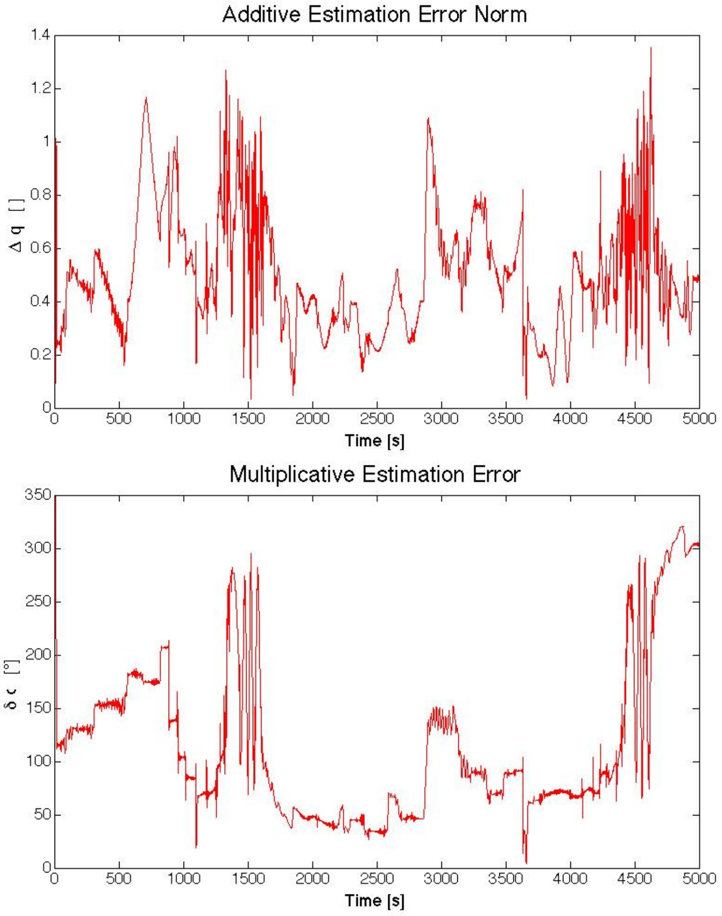


Figure 10: Simulation of the Modified Discrete EKF

6.4 Optimal REQUEST

6.4.1 Computational model

The setup used in the tests of the Optimal REQUEST algorithm is identical to the one used for the Modified Kalman Filter.

6.4.2 Analysis of some results

In the simulations conducted, the body coordinate system rotation speed, was chosen as a variable. The speed and magnetic measurement noises were Gaussian zero-mean white noises with a standard deviation of 3.2×10^{-4} m/s and 1000 nT, respectively. The speed and attitude model noises were Gaussian zero-mean white noises with a standard deviation of 1×10^{-4} m/s and 1×10^{-4} 1000 nT, respectively. Each simulated time span was 5000 seconds. Figure 11 depicts the multiplicative and additive estimation errors. The convergence time for the Optimal REQUEST algorithm is on average less than 100 s.

The tables 3 and 4 show the behaviour of the estimation error with different simulated speeds and sampling times for the Optimal REQUEST. In order to achieve acceptable levels of estimation error, several simulations were repeated with various values of the noise parameters of the filter η_{noise} and μ_{noise}. It can be seen that the sampling time t_s has a great influence on the estimation precision. No good results were obtained with sampling times exceding $1\,s$. The smallest sampling the time system can perform by is $0.5\,s$ the achievable precision is limited. The behaviour of the filter worsens in high speeds but the errors mean value remains within the tolerances stipulated by the application [7].

	μ		σ	
Speed $[deg/s]$	Multiplicative [°]	Additive [ ]	Multiplicative [°]	Additive [ ]
5	331.98	0.22	8.21	3.91×10^{-2}
2	11.49	9.03×10^{-2}	16.07	3.67×10^{-2}
1	6.73	4.95×10^{-2}	17.73	4.33×10^{-2}
0.5	4.63	3.83×10^{-2}	7.75	2.87×10^{-2}
0.1	4.37	6.24×10^{-2}	8.53	4.44×10^{-2}

Table 3: Statistical characteristics of the two criteria of estimation error for different speeds. The sampling time is $t_s = 0.5\ s$. Noise parameters are $\mu_{noise} = 3.81 \times 10^4$ and $\eta_{noise} = 1.82 \times 10^{-9}$

	μ		σ	
Speed $[deg/s]$	Multiplicative [°]	Additive [ ]	Multiplicative [°]	Additive [ ]
5	236.2	0.72	13.42	0.23
2	54.29	0.4	10.85	0.11
1	23.28	0.18	13.07	9.78×10^{-2}
0.5	13.83	0.11	9.58	7.76×10^{-2}
0.1	14.95	0.12	11.17	6.7×10^{-2}

Table 4: Statistical characteristics of the two criteria of estimation error for different speeds. The sampling time is $t_s = 1\ s$. Noise parameters are $\mu_{noise} = 3.81 \times 10^4$ and $\eta_{noise} = 1.82 \times 10^{-9}$

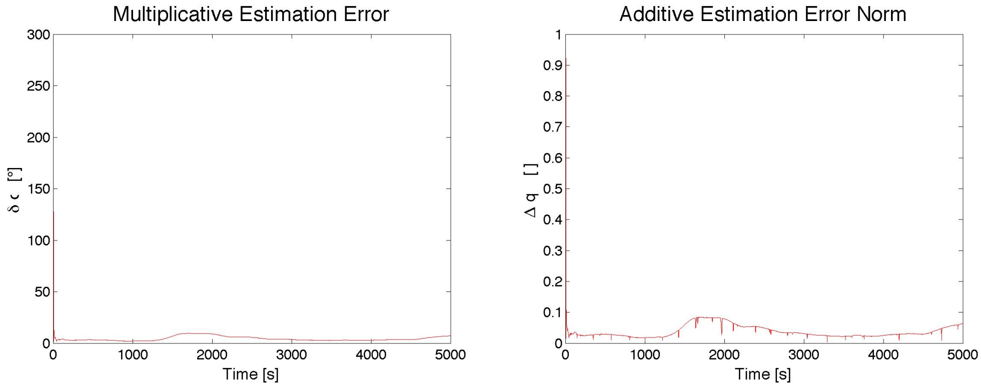


Figure 11: Multiplicative and additive estimation errors for 0.5 deg/s and $t_s = 0.5\ s$ with $\mu_{noise} = 3.81 \times 10^4$ and $\eta_{noise} = 1.82 \times 10^{-9}$

6.4.3 Separation Principle

The compatibility of the Optimal REQUEST Algorithm with the controller exposed in 6.2.3 was tested. The simulation conditions were identical to those used in the section 6.4.2. The derivative and the proportional terms chosen were $10^{-5}\ \frac{N.s}{rad}$ and $10^{-8}\ N.m$, respectively, and the initial angular speed of the satellite was 0.1 deg/s. It may be seen in figure 12, that the additive and multiplicative estimation errors remain within acceptable values. Figure 13 shows that the pair controller-estimator manages to dissipate the energy present in the system. The spikes in the energy graphic may be due to the magnetic field; this behaviour may be explained by an unfavourable position of the satellite with respect to the Earth field.

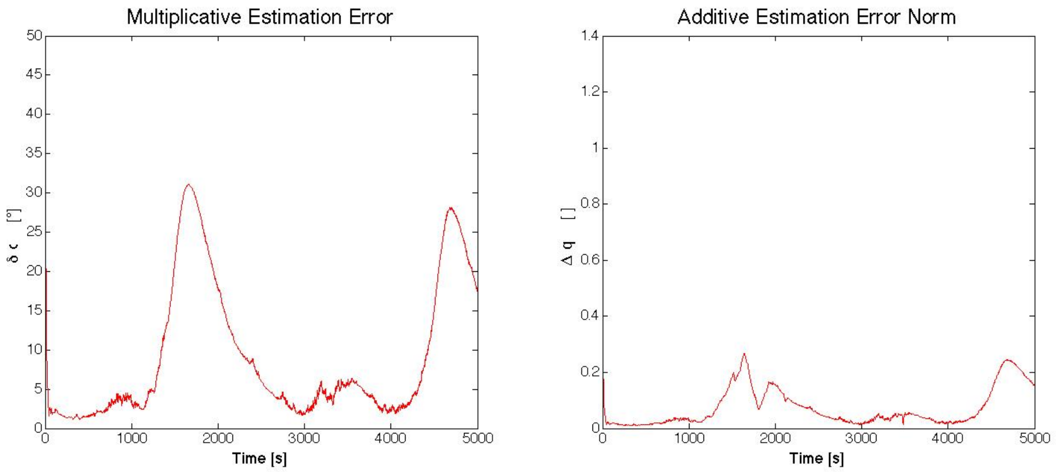


Figure 12: Multiplicative and additive estimation errors for 0.5 deg/s and $t_s = 0.5\,s$ with $\mu_{noise} = 3.81 \times 10^4$, $\eta_{noise} = 1.82 \times 10^{-9}$, $k_v = 10^{-5}$ and $k_p = 10^{-8}$

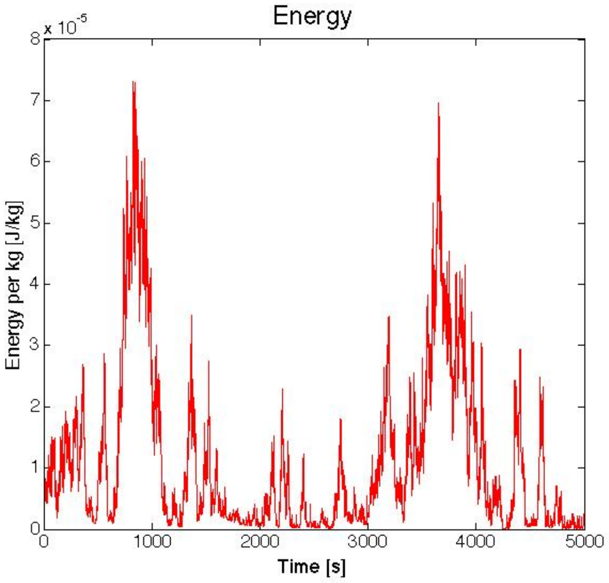


Figure 13: Energy per mass unit for 0.5 deg/s and $t_s = 0.5\,s$ with $\mu_{noise} = 3.81 \times 10^4$, $\eta_{noise} = 1.82 \times 10^{-9}$, $k_v = 10^{-5}$ and $k_p = 10^{-8}$

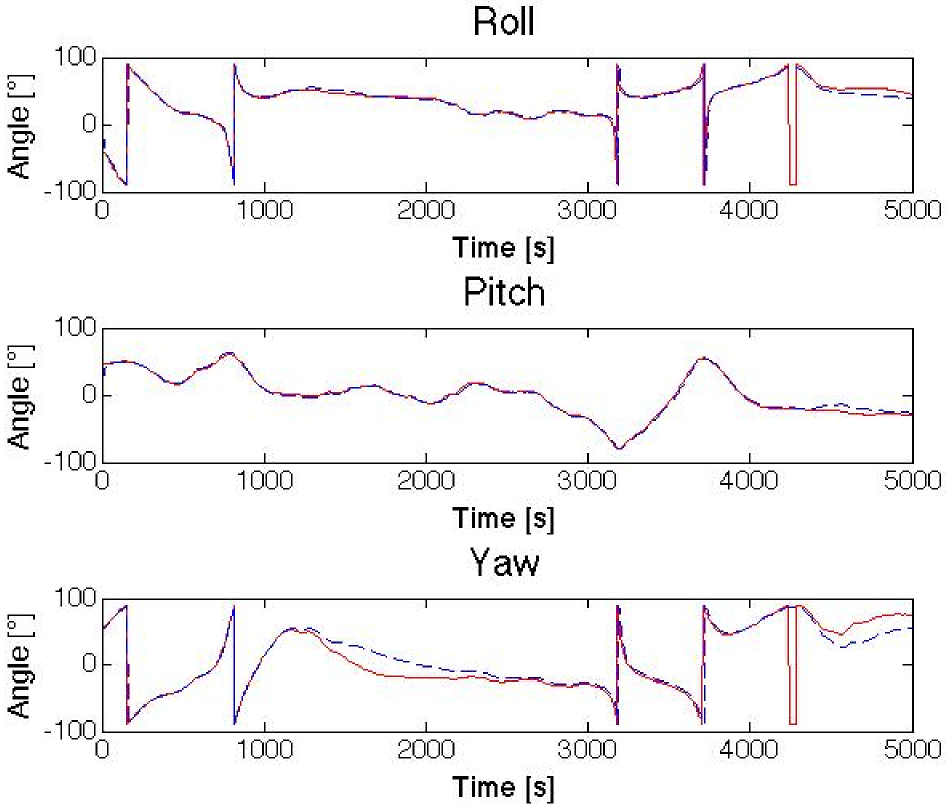


Figure 14: Attitude angles for 0.5 deg/s and $t_s = 0.5s$ with $\mu_{noise} = 3.81\times10^4$, $\eta_{noise} = 1.82 \times 10^{-9}$, $k_v = 10^{-5}$ and $k_p = 10^{-8}$

7 Conclusion

During this project, different methods to estimate the attitude have been considered.
The Optimal REQUEST algorithm shows satisfying performances. Moreover, when combined with the previously designed controller it provides sufficient precise information to dissipate the system's kinetic energy. However, the results were obtained through simulations, which implies that they should be interpreted cautiously. No tests have been conducted with the algorithm implemented in the ADCS controller. Several factors could affect the behaviour of the real system :

- No Floating Point Unit is available in the microcontroller.
- Computational power restrictions may impose a slower sampling rate than the optimal value found.
- Unmodelled factors may appear, such as the perturbation torques.

The first and second potential problems may be avoided with a correct transcription of the Optimal REQUEST. The third may be addressed by adding, if possible, an adaptative procedure as suggested in [3].

Finally, I would like to thank everyone the “Laboratoire d'Automatique”, in particular my supervisors Philippe Müllhaupt for his help and valuable advice, Levente Bodizs for his explanations on the Kalman filter and Basile Graf for his insight on teamwork and help with quaternions.

Annexes

A The referentials

The standard referential considered will be the geocentric referential.

A.1 Geocentric referential

Geocentric coordinates are the polar coordinates centered at the center of the Earth. Points are defined by the geocentric longitude, geocentric latitude, and distance from the planet center. It is to be noted that, given the non-spherical nature of the Earth, the geocentric latitude does not correpond exactly with the latitude used on maps.

A.2 Geodetic referential

Geodetic coordinates are represented by longitude, latitude, and elevation above sea level. These are the coordinates may read on a map or seen on a GPS receiver. However, as already mentioned, there is equivalence between the latitudes given in the geocentric and geodetic coordinate systems. Additional information is available in [16]

A.3 Orbital referential

The orbital referential is used by the attitude determination algorithm of the satellite. It is fixed to the orbit with the positive x-direction pointing in the displacement direction and a positive z-direction pointing towards the center of the Earth. The referential is completed by a y vector generated in order to have an orthonormal basis. See figure 15. Further details are provided in section A.4.2.

A.4 Transformation between referentials

A coordinate transformation is a conversion from one referential to another. For each reference frame, the transformation matrix transforming cartesian reference frame towards itself will be presented. The conversion between R and R' can be obtained through the cartesian referential. Given x_C a vector in the cartesian frame C and an arbitrary referential frame R

$$\begin{aligned} x_R &= A x_C \\ x_C &= A^{-1} x_R \end{aligned} \tag{82}$$

Where A and A^{-1} are the matrices for the direct and inverse transformation, respectively. Given a third referential R' so that

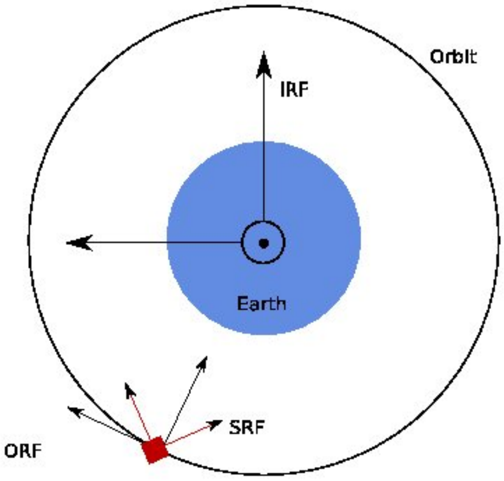


Figure 15: Orbital coordinates

$$\begin{aligned} x_{R'} &= Bx_C \\ x_C &= B^{-1}x_{R'} \end{aligned} \tag{83}$$

B and B^{-1} are the transformation matrices between C and R'. The transformation between two arbitrary referentials is computed as follows

$$\begin{aligned} x_{R'} &= BA^{-1}x_R \\ x_R &= AB^{-1}x_{R'} \end{aligned} \tag{84}$$

A.4.1 Cartesian to Geocentric

The relation between the geocentric and the cartesian coordinates can be expressed as

$$\begin{aligned} x &= \rho\cos\phi\cos\theta \\ y &= \rho\cos\phi\sin\theta \\ z &= \rho\sin\phi \end{aligned} \tag{85}$$

and the direct transformation matrix is

$$A = \begin{bmatrix} \cos\phi\cos\theta & \cos\phi\sin\theta & \sin\phi \\ -\sin\theta & \cos\theta & 0 \\ \sin\phi\cos\theta & \sin\phi\sin\theta & \cos\phi \end{bmatrix} \tag{86}$$

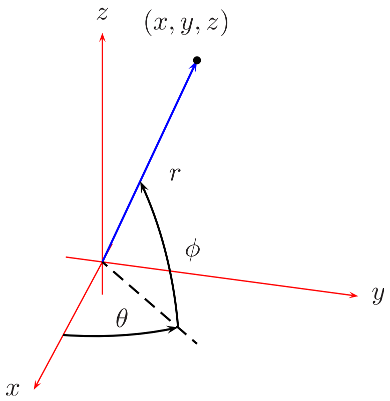


Figure 16: Angles in the geocentric reference frame

A.4.2 Cartesian to Orbital Reference Frame

The orbital referential is attached to the orbit of the satellite and therefore is a curvilinear coordinate system. As illustrated in (17), two normed and orthogonal vectors are defined using the radius and the speed of the satellite. The third component is built with a cross product in order to form an orthonormal base. The construction is summarized as

$$\begin{aligned} \vec{1_r} &= -\frac{\vec{r}}{|\vec{r}|} \\ \vec{1_v} &= \frac{\vec{v}}{|\vec{v}|} \\ \vec{1_u} &= \vec{1_r} \times \vec{1_v} \end{aligned} \tag{87}$$

Where $\vec{r}$ and $\vec{v}$ are respectively the radius and the speed vectors expressed in the cartesian referential. The transformation matrix may be defined as

$$A = \begin{bmatrix} 1_{r,x} & 1_{r,y} & 1_{r,z} \\ 1_{v,x} & 1_{v,y} & 1_{v,z} \\ 1_{u,x} & 1_{u,y} & 1_{u,z} \end{bmatrix} \tag{88}$$

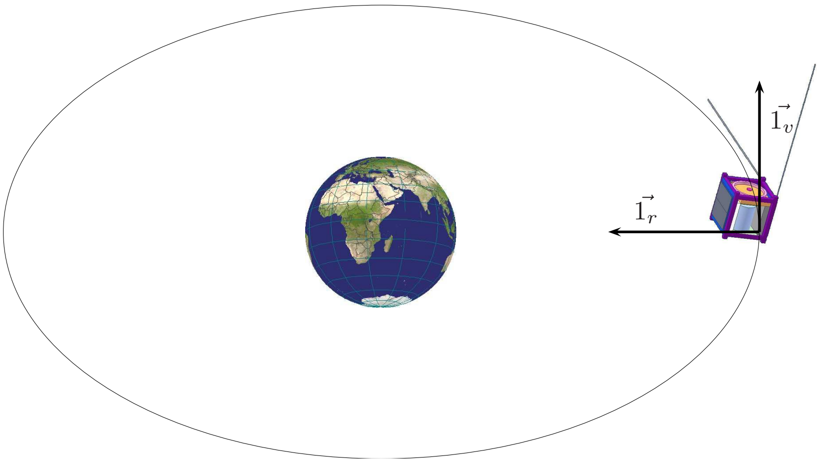


Figure 17: Definition of the orbital referential main vectors

B Mathematical annex

B.1 Optimal REQUEST Measurement Equation

The measurement equation of the Optimal REQUEST K matrix is derived as follows. D_{k+1} denotes the Direction Cosine Matrix at the time t_{k+1}. Assuming that the reference unit vector r_{k+1} is known exactly and the vector observation b_{k+1} is corrupted by a noise δb_{k+1} yields

$$b_{k+1} = D_{k+1} r_{k+1} + \delta b_{k+1} \tag{89}$$

Given V_{k+1}, a 4×4 symmetric matrix defined as

$$V_{k+1} = \frac{1}{a_{k+1}} \begin{bmatrix} S_b + \sigma_b I & \mathbf{z}_b \\ \mathbf{z}_b^T & \sigma_b \end{bmatrix} \tag{90}$$

Where S_b, σ_b and $\mathbf{z}_b$ are

$$\begin{gathered} B_b \equiv a_{k+1} \delta b_{k+1} r_{k+1}^T \qquad S_{k+1} \equiv B_b + B_b^T \\ \mathbf{z}_b \equiv a_{k+1} \delta b_{k+1} \times r_{k+1} \qquad \sigma_b \equiv tr(B_b) \end{gathered} \tag{91}$$

V_{k+1} is the error in the measurement equation for the REQUEST algorithm

$$\delta K_{k+1} = \delta K_{k+1}^0 + V_{k+1} \tag{92}$$

δK_{k+1} is the innovation for the estimation process computed on the basis of the noisy measurements b_{k+1} and r_{k+1} while δK_{k+1}^0 is built with an imaginary noise free vector. Note that V_{k+1} is linear in δb_{k+1} and r_{k+1}, which implies that V_{k+1} will be random if δb_{k+1} is random.

B.2 Dynamics of the K Matrix Error

The equation governing matrix K^0, computed on the basis of error free measurements are

$$\begin{aligned} K_{k+1/k}^0 = {} & \phi_k K_{k/k}^0 \phi_k \\ K_{k+1/k+1}^0 = {} & (1 - \rho_{k+1}) \tfrac{m_k}{m_{k+1}} K_{k+1/k}^0 + \rho_{k+1} \tfrac{\delta m_k}{m_{k+1}} \delta K_{k+1}^0 \end{aligned} \tag{93}$$

The estimated K matrices are

$$\begin{aligned} K_{k+1/k} = {} & \phi_k K_{k/k} \phi_k \\ K_{k+1/k+1} = {} & (1 - \rho_{k+1}) \tfrac{m_k}{m_{k+1}} K_{k+1/k} + \rho_{k+1} \tfrac{\delta m_k}{m_{k+1}} \delta K_{k+1} \end{aligned} \tag{94}$$

The estimation errors may be defined as

$$\begin{aligned} \Delta K_{k+1/k} \equiv &\ K^0_{k+1/k} - K_{k+1/k} \\ \Delta K_{k+1/k+1} \equiv &\ K^0_{k+1/k+1} - K_{k+1/k+1} \end{aligned} \tag{95}$$

Where $\Delta K_{k+1/k}$ and $\Delta K_{k+1/k+1}$ denote respectively the *a priori* and the *a posteriori* estimation errors of the algorithm. It is assumed that the *a priori* estimation error is unbiased, that is $E\left[\Delta K_{k+1/k}\right] = 0$. Substracting (94) from (93) yields

$$\begin{aligned} \Delta K_{k+1/k} = &\ \phi_k \Delta K_{k/k} \phi_k \\ \Delta K_{k+1/k+1} = &\ \left(1-\rho_{k+1}\right) \tfrac{m_k}{m_{k+1}} \Delta K_{k+1/k} + \rho_{k+1} \tfrac{\delta m_k}{m_{k+1}} \left(\delta K^0_{k+1} - \delta K_{k+1}\right) \end{aligned} \tag{96}$$

The term $\delta K^0_{k+1} - \delta K_{k+1}$ may be identified as the measurement error V_{k+1} by using (92). Therefore the second equation of (96) is rewritten as

$$\Delta K_{k+1/k+1} = \left(1-\rho_{k+1}\right) \frac{m_k}{m_{k+1}} \Delta K_{k+1/k} + \rho_{k+1} \frac{\delta m_k}{m_{k+1}} V_{k+1} \tag{97}$$

Taking the expectation of both sides, yields

$$E\left[\Delta K_{k+1/k+1}\right] = \left(1-\rho_{k+1}\right) \frac{m_k}{m_{k+1}} \left[\Delta K_{k+1/k}\right] + \rho_{k+1} \frac{\delta m_k}{m_{k+1}} \left[V_{k+1}\right] \tag{98}$$

Under the assumptions that the measurement error V_{k+1} and the *a priori* estimation error $\Delta K_{k+1/k}$ are zero-mean random variables, the *a posteriori* estimation error $\Delta K_{k+1/k+1}$ has zero mean. The propagation of the covariance is defined as

$$\begin{aligned} P_{k+1/k} \equiv &\ E\left[\Delta K_{k+1/k} \Delta K^T_{k+1/k}\right] \\ P_{k+1/k+1} \equiv &\ E\left[\Delta K_{k+1/k+1} \Delta K^T_{k+1/k+1}\right] \end{aligned} \tag{99}$$

Using (97), $P_{k+1/k+1}$ is computed

$$\begin{aligned} \Delta K_{k+1/k+1} \Delta K^T_{k+1/k+1} = &\ \left[\left(1-\rho_{k+1}\right) \tfrac{m_k}{m_{k+1}}\right]^2 \Delta K_{k+1/k} \Delta K^T_{k+1/k} \\ &\ \left(1-\rho_{k+1}\right) \rho_{k+1} \tfrac{\delta m_{k+1} m_k}{m^2_{k+1}} \left(\Delta K_{k+1/k} V^T_{k+1} V_{k+1} \Delta K^T_{k+1/k}\right) \\ &\ \left(\rho_{k+1} \tfrac{\delta m_k}{m_{k+1}}\right)^2 V_{k+1} V^T_{k+1} \end{aligned} \tag{100}$$

It may be shown that the measurement noise V_{k+1} and the *a priori* estimation error $\Delta K_{k+1/k}$ are uncorrelated, which implies

$$\begin{aligned} E\left[K_{k+1/k}V_{k+1}^T\right] &= 0 \\ E\left[V_{k+1}K_{k+1/k}^T\right] &= 0 \end{aligned} \tag{101}$$

Taking the expectation of both sides of (100) and using (101) yields the expression for $P_{k+1/k+1}$

$$\begin{aligned} P_{k+1/k+1} = \; & \left((1-\rho_{k+1})\,\frac{m_k}{m_{k+1}}\right)^2 E\left[\Delta K_{k+1/k}\Delta K_{k+1/k}^T\right] \\ & + \left(\rho_{k+1}\frac{\delta m_k}{m_{k+1}}\right)^2 E\left[V_{k+1}V_{k+1}^T\right] \end{aligned} \tag{102}$$

By identifying $E\left[\Delta K_{k+1/k}\Delta K_{k+1/k}^T\right]$ as the *a priori* estimation error $P_{k+1/k}$ and by noting $E\left[V_{k+1}V_{k+1}^T\right]$ as the measurement noise covariance, (102) becomes

$$P_{k+1/k+1} = \left((1-\rho_{k+1})\frac{m_k}{m_{k+1}}\right)^2 P_{k+1/k} + \left(\rho_{k+1}\frac{\delta m_{k+1}}{m_{k+1}}\right)^2 R_{k+1} \tag{103}$$

B.3 Computation of the Matrices Q_k and R_k

In the Kalman filter, the noise matrices Q_k and R_k are directly related to the process noise covariance and the measurement noise covariance, respectively. However, in the Optimal REQUEST the estimate is computed through the matrix K, which requires the introduction of modifications into the definition of the noise matrices. The demonstrations related to the present section are available in [3]

Stochastic models for the random variables W_k and V_k

To estimate the state of the system accurately, the sensors collect information from two sets of physical values: attitude information and angular speeds. Only basic models will be considered to account for the measurement errors. The measured values may be described by the following equations

$$\begin{aligned} \omega_k = \; & \omega_k^0 + \epsilon_k \\ b_k = \; & b_k^0 + \delta b_k \end{aligned} \tag{104}$$

In order to derive the stochastic models for W_k and V_k, the stochastic models for ϵ_k and δb_k are required. ϵ_k is modelled as a zero-mean white gaussian noise vector process whose components are identically distributed with variance η_k

$$E[\epsilon_k] = 0 \qquad E[\epsilon_k \epsilon_{k+i}^T] = \eta_k I_{(3\times3)} \delta_{k,k+i} \qquad k = 1, 2... \tag{105}$$

$\delta_{k,k+i}$ is the Kronecker's delta function. For b_k, the noise model given in [3] has its first and second moments equal to

$$E[\delta b_k] = 0 \qquad E[\delta b_k \delta b_{k+i}^T] = \mu_k (I_{3\times3} - b_k b_{k+i}^T) \delta_{k,k+i} \qquad k = 1, 2... \tag{106}$$

Where μ_k is the variance of the component of b_k along a direction normal to $E[\delta b_k]$. Furthermore, it is assumed that both processes are mutually uncorrelated.

Q_k matrix

Recall that

$$Q_k \equiv E\left[W_k W_k^T\right] \tag{107}$$

The 4×4 matrix Q_k can be partitioned in

$$Q_k = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \Delta t^2 \tag{108}$$

with

$$\begin{aligned} Q_{11} &= \eta_k \left\{ \left[\hat{z}^T \hat{z} - tr(\hat{B}\hat{B}^T) \right] I_{3\times3} + 2 \left[\hat{B}^T \hat{B} - \hat{B}^2 (\hat{B}^T)^2 \right] \right\} \\ Q_{12} &= -\eta_k \left(\mathbf{y} + \hat{B}^T \hat{z} \right) \\ Q_{21} &= Q_{12}^T \\ Q_{22} &= \eta_k \left[tr(\hat{B}\hat{B}^T) + \hat{\sigma}^2 + \hat{z}^T \hat{z} \right] \end{aligned} \tag{109}$$

$\hat{B}$, $\hat{z}$ and $\hat{\sigma}$ are computed using the estimated values and η_k is defined in (105). $\mathbf{y}$ is a 3×1 vector defined as follows

$$\begin{aligned} M &\equiv \hat{B}(\hat{B} - \hat{\sigma} I_{3\times3}) \\ (\mathbf{y}\times) &\equiv M - M^T \end{aligned} \tag{110}$$

$(\mathbf{y}\times)$ stands for the cross product.

R_k **matrix**

Recall that

$$R_k \equiv E\left[V_k V_k^T\right] \tag{111}$$

The 4×4 matrix R_k can be partitioned in

$$R_k = \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix} \tag{112}$$

with

$$\begin{aligned} R_{11} = &\ \mu_k \left[3 - (r_k^T)^2\right] I_{3\times3} + (r_k^T b_k)(r_k b_k^T + b_k r_k^T) + (r_k\times)(b_k b_k^T)(r_k\times)^T \\ R_{12} = &\ 0 \\ R_{21} = &\ 0 \\ R_{22} = &\ 2\mu_k \end{aligned} \tag{113}$$

where r_k and b_k are the vector measurements taken respectively in the body reference frame and in the body frame. μ_k is defined in (106).

References

[1] Itzhack Bar-Itzhack. Optimum normalization of a computed quaternion of rotation. *IEEE transactions on aerospace and electronic systems*, March 1971.

[2] Itzhack Bar-Itzhack and Yaakov Oshman. Attitude determination from vector observations: Quaternion estimation. *IEEE transactions on aerospace and electronic systems*, AES-21, Issue: 1, 1985.

[3] Daniel Choukroun. *Novel Methods For Attitude Determination Using Vector Observations.* PhD thesis, Israel Institute of Technology, May 2003.

[4] J. L. Farrell, J. C. Stuelpnagel, R. H. Wessner, J. R. Velman, and J. E. Brook. A least squares estimate of satellite attitude (Grace Wahba). *SIAM Review*, 8(3):384–386, 1966.

[5] Basile Graf. Swisscube control algorithm design and validation. Master's thesis, Ecole Polytechnique Federale de Lausanne, February 2007.

[6] Mohinder S. Grewal and Angus P. Andrews. *Kalman Filtering - Theory and Practice.* Information and system science series. Prentice Hall, 1993.

[7] Daniel Håkansson. Mission design. Technical report, Ecole Polytechnique Federale de Lausanne - Swisscube Design Team, 2007.

[8] Kaspar Jenni. ADCS gyroscopes. Technical report, Ecole Polytechnique Federale de Lausanne - Swisscube Design Team, 2007.

[9] Kristian Krogh and Elmo Schreder. Attitude determination for AAU CubeSat. Technical report, Aalborg University - Department of Control Engineering, June 2002.

[10] F. L. Markley and D. Mortari. *Quaternion attitude estimation using vector observations.*, August 1999.

[11] Stian Sondersrod Ose. Attitude determination for the norwegian student satellite nCube. Master's thesis, Norwegian University of Science and Technology, June 2004.

[12] Hervé Péter-Contesse. Earth's magnetic field model. Technical report, Ecole Polytechnique Federale de Lausanne - Swisscube Design Team, 2007.

[13] Hervé Péter-Contesse. ADCS hardware and system. Technical report, Ecole Polytechnique Federale de Lausanne - Swisscube Design Team, 2007.

[14] Andreas Schmocker. ADCS sun-sensors. Technical report, Ecole Polytechnique Federale de Lausanne - Swisscube Design Team, 2007.

[15] Bernt Ove Sunde. Attitude determination for the student satellite nCube II: Kalman Filter. Technical report, Norwegian University of Science and Technology, 2003.

[16] http://en.wikipedia.org/wiki/Geographic_coordinate_system. *Geographic Coordinate System (visited 27/07/2007).*

[17] Grace Wahba. A least squares estimate of satellite attitude. *SIAM Review*, 7(3):409–409, 1965.

[18] Greg Welch and Gary Bishop. An introduction to the Kalman Filter. *Department of Computer Science University of North Carolina at Chapel Hill*, February 2001.