ISTS-2017-d-079／ISSFD-2017-079

Regularized Particle Filter with Roughening for Gyros Bias and Attitude Estimation using Simulated Measurements

By William SILVA,[1)] Roberta GARCIA,[2)] Hélio KUGA[1)] and Maria ZANARDI[3)]

[1)]*Technological Institute of Aeronautics, ITA, São José dos Campos, Brazil*
[2)]*São Paulo University, USP, Lorena, Brazil*
[3)]*Federal University of ABC, UFABC, Santo André, Brazil*

(Received June 21st, 2017)

This work describes the attitude determination and the gyros drift estimation using the Regularized Particle Filter (RPF) with Roughening and Unscented Kalman Filter (UKF) for nonlinear systems. The Particle Filter has some similarities with the Unscented Kalman Filter which transforms a set of points (cloud) through known nonlinear equations and combines the results to estimate the moments (mean and covariance) of the state. However, in the Particle Filter the points (particles cloud) are chosen randomly, whereas in the Unscented Kalman Filter the points are carefully picked up based on a very specific criterion. In this way, the number of points (particles) used in a Particle Filter generally needs to be much greater than the number of points (sigma-points) in an Unscented Kalman Filter. The application of the Regularized Particle Filter in this work uses simulated measurements of a real satellite CBERS-2 (China Brazil Earth Resources Satellite 2) which was at a polar sun-synchronous orbit with an altitude of 778km, crossing Equator at 10:30am in descending direction, frozen eccentricity and perigee at 90 degrees, and provided global coverage of the world every 26 days. These simulated measurements were yielded by the package PROPAT, a Satellite Attitude and Orbit Toolbox for Matlab. The attitude dynamical model for CBERS-2 is described by nonlinear equations involving the quaternions. The attitude sensors available are two DSS (Digital Sun Sensors), two IRES (Infra-Red Earth Sensor), and one triad of mechanical gyros. In this work, attitude dynamics as well as sensors measurements are simulated to reproduce realistic scenarios with low or high sampling rates, or different levels of accuracy. .

Key Words: Regularized particle filter, Unscented Kalman Filter, Nonlinear System, Attitude Estimation, Gyro Calibration

Nomenclature

$f(.)$	:	Process model
$h(.)$	:	Measurements model
h	:	Kernel bandwidth
k	:	Discret time
m	:	Measurements number
n	:	State number
$p(\boldsymbol{x}_0)$	:	Probability densit function
q	:	Quaternions
t	:	Time
$\boldsymbol{v}_k$	:	Measurement noise
$\boldsymbol{x}_k$	:	State vector
$\hat{\boldsymbol{x}}_k$	:	Estimated state vector
$\boldsymbol{w}_k$	:	Process noise
$\boldsymbol{y}_k$	:	Measurements vector
$\hat{\boldsymbol{y}}_k$	:	Estimated measurements vector
$\boldsymbol{F}_k$	:	Jacobian matrix of $f(.)$
$\boldsymbol{H}_k$	:	Jacobian matrix of $fh(.)$
$\boldsymbol{P}_k$	:	Covariance matrix of $\boldsymbol{x}_k$
$\boldsymbol{Q}_k$	:	Covariance matrix of $\boldsymbol{w}_k$
$\boldsymbol{R}_k$	:	Covariance matrix of $\boldsymbol{v}_k$
Ψ_i	:	Likelihood
$\boldsymbol{\Omega}_\omega$	:	Anti-symmetric matrix

1. Introduction

Many spacecraft missions require precise pointing of their sensors and demand a real time precise attitude determination, which involves estimating both sattelite attitude and gyros bias, from output information of the attitude measurements system. The importance of determining the attitude is related not only to the performance of the attitude control system but also to the precise usage of information obtained by payload experiments performed by the satellite.

Attitude estimation is a process of determining the orientation of a satellite with respect to an inertial reference system by processing data from attitude sensors. Given a reference vector, the attitude sensor measures the orientation of this vector with respect to the satellite system. Then, it is possible to estimate the orientation of the satellite processing computationally these vectors using attitude estimation methods. The bias can be defined as a output component not related to input to which the sensor is subjected and its components have features deterministic and stochastic. Therefore, you need to know and to characterize it, consequently set the method for estimating.[1)]

The attitude stabilization here is done in three axes namely geo-targeted, and can be described in relation to the orbital system. In this frame, the movement around the direction of the orbital speed is named *roll* (ϕ), the movement around the normal direction to the orbit is called *pitch* (θ), and finally the movement around the Zenith/Nadir direction is called *yaw* (ψ). See Fig. 1.

In this work, the attitude model is described by quaternions, with state estimation performed by the Regularized Particle Filter (RPF) compared to the Unscented Kalman Filter (UKF),[6)] used with reference. This methods is capable of estimating nonlinear systems states from data obtained from different sensors

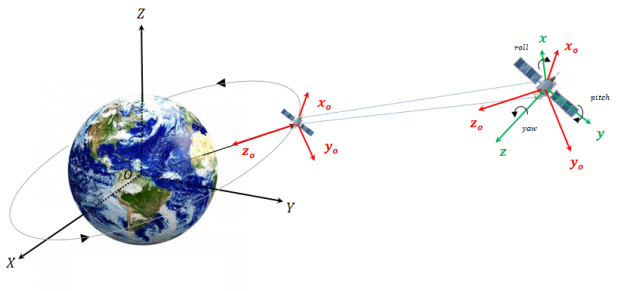

Fig. 1. The orbital local system $(\mathbf{x}_o, \mathbf{y}_o, \mathbf{z}_o)$ and the attitude system $(\mathbf{x}, \mathbf{y}, \mathbf{z})$

of attitude. It was considered real data supplied by gyroscopes, infrared Earth sensors and digital solar sensors. These sensors are on board the CBERS-2 satellite (China-Brazil Earth Resources Satellite), and the measurements were recorded by the Satellite Control Center of INPE (Brazilian National Institute for Space Research).

2. ATTITUDE REPRESENTATION BY EULER ANGLES

The attitude of a satellite is defined by a set of parameters that allow, uniquely correlating in an instant of time, a fixed coordinate system of the satellite (which accompanies his movement of rotation and translation) to another inertial system, which is usually related to the Earth.[2)]

In general it is considered inertial or near-inertial, which means that its movement in relation to the system truly inertial is despicable, when compared with the movement of the body itself.

The quaternion is a four dimensional vector that defines a unit vector in space and the angle to rotate about that unit vector to transform from one frame to another.[3,4)] The quaternion can be written as follows:

$$\boldsymbol{q} = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix}^T = \begin{bmatrix} \boldsymbol{q}^* & q_4 \end{bmatrix}^T \tag{1}$$

where, $\boldsymbol{q}^* = \boldsymbol{e} \sin\dfrac{\zeta}{2}$ e $q_4 = \cos\dfrac{\zeta}{2}$

Here, $\boldsymbol{e} = \begin{bmatrix} e_1 & e_2 & e_3 \end{bmatrix}^T$ is the unit vector and ζ is the angle of rotation about unit vector $\boldsymbol{e}$. The quaternion satisfies the following constraint:

$$\boldsymbol{q}^T\boldsymbol{q} = q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1 \tag{2}$$

The state vector formed by the quaternion and the gyro bias vector is given by:

$$\boldsymbol{x} = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 & \varepsilon_x & \varepsilon_y & \varepsilon_z \end{bmatrix}^T \tag{3}$$

If the angular velocity vector $\boldsymbol{\omega} = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T$ of body frame is known with respect to another reference frame, the differential equation of the quaternion system becomes[3,5)]

$$\begin{aligned} \dot{\boldsymbol{q}} &= \frac{1}{2}\boldsymbol{\Omega}_\omega \boldsymbol{q} \\ \dot{\boldsymbol{\varepsilon}} &= 0 \end{aligned} \tag{4}$$

where, $\boldsymbol{\Omega}_\omega$ is an anti-symmetric matrix 4×4 given by:

$$\boldsymbol{\Omega}_\omega = \begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix} \tag{5}$$

Assuming that the working data is sampled at a fixed rate and the angular velocity vector in the satellite system is constant over the sampling interval, then a solution of the problem is:[4)]

$$\boldsymbol{q}(t_{k+1}) = \boldsymbol{\Phi}_{\boldsymbol{q}}(\Delta t, |\boldsymbol{\omega}|)\, \boldsymbol{q}(t_k) \tag{6}$$

where, Δt the sampling interval; $\boldsymbol{q}(t_k)$ is the attitude quaternion in time t_k; $\boldsymbol{q}(t_{k+1})$ is the quaternion of propagated attitude to time t_{k+1}; and $\boldsymbol{\Phi}_{\boldsymbol{q}}$ is the transition matrix carrying the system time t_{k+1} for t_k , given by:

$$\boldsymbol{\Phi}_{\boldsymbol{q}}(\Delta t, |\boldsymbol{\omega}|) = \cos\left(\frac{|\boldsymbol{\omega}|\,\Delta t}{2}\right)\boldsymbol{I} + \frac{1}{|\boldsymbol{\omega}|}\sin\left(\frac{|\boldsymbol{\omega}|\,\Delta t}{2}\right)\boldsymbol{\Omega}_\omega \tag{7}$$

3. THE PARTICLE FILTER

The particle filter, which was invented for implementing a numerical Bayesian estimator is a statistical approximation by brute force,[7,8)] for estimation problems that are difficult to solve with the conventional Kalman filter, or highly nonlinear systems.[9)]

Suppose a non-linear system described by the equation

$$\begin{aligned} \boldsymbol{x}_{k+1} &= f_k(\boldsymbol{x}_k, \boldsymbol{w}_k) \\ \boldsymbol{y}_k &= h_k(\boldsymbol{x}_k, \boldsymbol{v}_k) \end{aligned} \tag{8}$$

where k is the time index, $\boldsymbol{x}_k$ is the state and $\boldsymbol{w}_k$ is the process noise, $\boldsymbol{y}_k$ are the measures and $\boldsymbol{v}_k$ is the noise measure. The functions $f_k(.)$ and $h_k(.)$ is a nonlinear time-varying system and the equation measures respectively. The noises $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ are considered independent and whites with known probability density function.

To start the estimation of the problem, we generate randomly a given number N of vector states for the initial condition of the probability density function $p(\boldsymbol{x}_0)$, which is assumed known. These vectors are called states of particles and are represented as $\boldsymbol{x}_{0,i}^+$ $(i = 1, 2, ..., N)$.

$$\boldsymbol{x}_{0,i}^+ \sim p(\boldsymbol{x}_{0,i}) \qquad (i = 1, 2, ..., N) \tag{9}$$

At each step $k = 1, 2, ...$, the particles propagate to the next time step using the process dynamics of the equation $f(.)$. This step is known as sampling.

$$\boldsymbol{x}_{k,i}^- = f_{k-1}(\boldsymbol{x}_{k-1,i}^+, \boldsymbol{w}_{k-1}^i) \qquad (i = 1, 2, ..., N) \tag{10}$$

where, each noise vector $\boldsymbol{w}_{k-1}^i$ is generated randomly based on the known probability density function of $\boldsymbol{w}_{k-1}$.

After getting all the measures in time k, we compute the conditional relative likelihood of each particle $\boldsymbol{x}_{k,i}^-$ assessed by probability density function $p(\boldsymbol{y}_k \mid \boldsymbol{x}_{k,i}^-)$ obtained from equation measures $h(.)$ and noise probability density function of the measurements $\boldsymbol{v}_k$.[1,9)]

$$\begin{aligned} \Psi_i &= p(\boldsymbol{y}_k \mid \boldsymbol{x}_k^-) \\ \Psi_i &\sim \frac{1}{(2\pi)^{m/2}\,|\boldsymbol{R}|^{1/2}} exp\left(\frac{-(\boldsymbol{y}^* - h(\boldsymbol{x}_{k,i}^-))^T\boldsymbol{R}^{-1}(\boldsymbol{y}^* - h(\boldsymbol{x}_{k,i}^-))}{2}\right) \end{aligned} \tag{11}$$

Normalizing the relative likelihoods obtained by Eq. (11) as:

$$\tilde{\Psi}_i = \frac{\Psi_i}{\sum_{j=1}^{N} \Psi_j} \tag{12}$$

this ensures that the sum of all the probability is equal to 1.

The next step is to find a new dataset of $\boldsymbol{x}_{k,i}^{+}$ that is randomly generated based on the probability relative Ψ_i,[10] where it is shown that the probability density function of new particle $\boldsymbol{x}_{k,i}^{+}$ tends to probability density function of $p(\boldsymbol{x}_k \mid \boldsymbol{y}_k)$ with a number of samples N next to infinity.

Thus, the resampling can be summarized as:

$$\boldsymbol{x}_{k,i}^{+} = \sum_{i=1}^{N} \tilde{\Psi}_i \, \boldsymbol{x}_{k,i}^{-} \tag{13}$$

Thus, a set of particles $\boldsymbol{x}_{k,i}^{+}$ can be distributed according to the probability density function $p(\boldsymbol{x}_k \mid \boldsymbol{y}_k)$ and you can compute any desired statistical measure of this probability density function.

Finally, if we want to compute the expected value of $E(\boldsymbol{x}_k \mid \boldsymbol{y}_k)$ one can approximate it by averaging the algebraic sum of the particles.

$$E(\boldsymbol{x}_k \mid \boldsymbol{y}_k) \approx \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{x}_{k,i}^{+} \tag{14}$$

4. Sample improverishment

Sample improverishment occurs when the region of state space in probability density function $p(\boldsymbol{y}_k \mid \boldsymbol{x}_k)$ has significant values that do not overlap with the $p(\boldsymbol{x}_k \mid \boldsymbol{y}_{k-1})$. This means that if all of our *a-priori* particles are distributed according to $p(\boldsymbol{x}_k \mid \boldsymbol{y}_{k-1})$, and we then use the computed probability density function $p(\boldsymbol{y}_k \mid \boldsymbol{x}_k)$ to resample the particles, only a few particles will be resampled to become *a-posteriori* particles. This is because only a few of the *a-priori* particles will be in a region of state space where the computed pdf $p(\boldsymbol{y}_k \mid \boldsymbol{x}_k)$ has significant value. This means that the resampling process will select only a few distinct *a-priori* particles to become a *a-posteriori* particles. Eventually, all of the particle will collapse to the same value[1].

This problem will be exarcerbated if the measurements are not consistent with the process model (modelling errors). This can be overcome by a brute-force method of simply increasing the number of particles N, but this can quickly lead to unreasonable computational demands and often simply delays the inevitable sample impoverishment. Other more intelligent ways of dealing with this problem can be used that are presented in this research .[11,12]

4.1. Roughening

Roughening can be used to prevent sample improverishment, as shown in.[12] In this method, random noise is added to each particle after the resampling process. This is similar to adding artificial process noise to the Kalman Filter. In the roughening approach, the *a-posteriori* particles are modified as follows:[9]

$$\begin{array}{l} \boldsymbol{x}_{k,i}^{+}(m) = \boldsymbol{x}_{k,i}^{+}(m) + \Delta\boldsymbol{x}(m) \quad (m = 1, ..., n) \\ \Delta\boldsymbol{x}(m) \sim (0, K\boldsymbol{M}(m)N^{-1/n}) \end{array} \tag{15}$$

where $\Delta\boldsymbol{x}(m)$ is a zero-mean random variable (usually Gaussian). K is a scalar tuning parameter that specifies the amount of jitter that is added to each particle, N is the number of particle, n is the dimension of the state space and $\boldsymbol{M}$ is a vector containing the maximum difference between the particle elements before roughening. The mth element of the $\boldsymbol{M}$ vector is given as

$$\boldsymbol{M}(m) = \max_{i,j} \left| \boldsymbol{x}_{k,i}^{+}(m) - \boldsymbol{x}_{k,j}^{+}(m) \right| \quad (m = 1, ..., n) \tag{16}$$

where k is the time step, and i and j are particle numbers.

4.2. Regularized Particle Filter

Another way of preventing sample impoverishment is through the use of the Regularized Particle Filter (RPF).[9,13,14] This performs resampling from a continuous approximation of the pdf $p(\boldsymbol{y}_k \mid \boldsymbol{x}_{k,i}^{-})$ rather than from he discrete probability density function samples used thus far. Recall in our resampling step in Eq. (11) that we used the probability

$$\Psi_i = p((\boldsymbol{y}_k = \boldsymbol{y}^*) \mid (\boldsymbol{x}_k = \boldsymbol{x}_{k,i}^{-})) \tag{17}$$

to determine the likelihood of selecting an *a-priori* particle to be an *a-posteriori* particle. Instead, we can use the pdf $p(\boldsymbol{x}_k \mid \boldsymbol{y}_k)$ to perform resampling. That is, the probability of selecting the particle $\boldsymbol{x}_{k,i}^{-}$ to be an *a-posteriori* particle is proportional to the pdf $p(\boldsymbol{x}_k \mid \boldsymbol{y}_k)$ evaluted at $\boldsymbol{x}_k = \boldsymbol{x}_{k,i}^{-}$. In the RPF, this probability density function is approximated as

$$\hat{p}(\boldsymbol{x}_k \mid \boldsymbol{y}_k) = \sum_{i=1}^{N} \Psi_{k,i} K_h(\boldsymbol{x}_k - \boldsymbol{x}_{k,i}^{-}) \tag{18}$$

where $q_{k,i}$ are the weights that are used in the approximation. Later on, we will see that these weights should be set equal to the q_i probabilities that were computed in Eq. (11). Thus, $K_h(.)$ is given as

$$K_h(\boldsymbol{x}) = h^n K(\boldsymbol{x}/h) \tag{19}$$

where h is the positive scalar Kernel bandwidth, and n is the dimension of the state vector. $K(.)$ is a kernel density that is a symmetric probability density function that satisfies

$$\begin{array}{l} \int \boldsymbol{x} K(\boldsymbol{x}) d\boldsymbol{x} = 0 \\ \int \|\boldsymbol{x}\|_2^2 \, K(\boldsymbol{x}) d\boldsymbol{x} < \infty \end{array} \tag{20}$$

The Kernel $K(.)$ and the bandwidth h are chosen to minimize a measure of the error between the assumed true density $p(\boldsymbol{x}_k \mid \boldsymbol{y}_k)$ and the approximate density $\hat{p}(\boldsymbol{x}_k \mid \boldsymbol{y}_k)$:

$$\{K(\boldsymbol{x}), h\} = argmin \int \left[\hat{p}(\boldsymbol{x}_k \mid \boldsymbol{y}_k) - p(\boldsymbol{x}_k \mid \boldsymbol{y}_k)\right]^2 d\boldsymbol{x} \tag{21}$$

In the classic case of equal weights ($q_{k,i} = 1/N$ for $i = 1, ..., N$) the optimal kernel is given as

[1] This is called the black hole of particle filtering

$$K(\boldsymbol{x}) = \begin{cases} \frac{n+2}{2V_n}\left(1 - \|\boldsymbol{x}\|_2^2\right) & if\ \|\boldsymbol{x}\|_2^2 < 1 \\ 0 & otherwise \end{cases} \tag{22}$$

where V_n is the volume of the n-dimensional unit hypersphere. $K(\boldsymbol{x})$ is called the Epanechnikov Kernel.[13)]

An n-dimensional unit hypersphere is a volume in n dimensions in which all points are one unit from the origin.[15)] In one dimensions, the unit hypersphere is a line with a length of two and volume $V_1 = 2$. In two dimensions, the unit hypersphere is a circle with a radius of one and volume $V_2 = \pi$. In three dimensions, the unit hypersphere is a sphere with a radius of one and de volume $V_3 = \frac{4\pi}{3}$. In the n dimensions the unit hypersphere has a volume $V_n = \frac{2\pi V_{n-2}}{n}$.

If $p(\boldsymbol{x} \mid \boldsymbol{y}_k)$ is Gaussian with an identity covariance matrix then the optimal bandwidth is given as

$$h^* = \left[8V_n^{-1}(n+4)(2\sqrt{\pi})^n\right]^{\frac{1}{n+4}} N^{-\frac{1}{n+4}} \tag{23}$$

In order to handle the case of multimodal probability density functions[1)], we should use $h = \frac{h^*}{2}$.[13,16)] These choice for the Kernel and the bandwidth are optimal only for the case of equal weights and a Gausian pdf, but they still are often used in other situations for obtain good particle filtering results. Instead of selecting *a-priori* particles to become *a-posteriori* particles using the probabilities of Eq. (17), we instead select *a-posteriori* particles based in a pdf approximation given in Eq. (18). This allows more diversity as we perform the update from the *a-priori* particles to a *a-posteriori* particles. In general, we should set the $\Psi_{k,i}$ weights in Eq. (18) equal to the Ψ_i probabilities shown in Eq. (17).

Since this procedure assumes that the true density $p(\boldsymbol{x} \mid \boldsymbol{y}_k)$ has a unity covariance matrix, we numerically compute the covariance of the $\boldsymbol{x}_{k,i}^-$ at each time step. Suppose that this covariance is computed as $\boldsymbol{S}$ (an $n \times n$ matrix). Then we compute the matrix square root of $\boldsymbol{S}$, denoted as A, such that $\boldsymbol{A}\boldsymbol{A}^T = \boldsymbol{S}$ (*e.g.*, we can use Cholesky decomposition for this computation). Then we compute the kernel as

$$K_h(\boldsymbol{x}) = (\det \boldsymbol{A})^{-1} h_{-n} K\left(\frac{\boldsymbol{A}^{-1}\boldsymbol{x}}{h}\right) \tag{24}$$

5. THE UNSCENTED KALMAN FILTER

Consider again a non-linear system described by the Eq. (8).

An unscented transformation is based on two fundamental principles. First, it is easy to perform a nonlinear transformation on a single point(rather than an entire probability density function). Second, it is not too hard to find a set of individual points in space whose sample probability density function approximate the true probability density function of a state vector.[9,17)]

Taking these two ideas together, suppose that we know the mean $\bar{\boldsymbol{x}}$ and covariance $\boldsymbol{P}$ of a vector $\bar{\boldsymbol{x}}$. We then find a set of deterministic vectors called sigma points whose ensemble mean and covariance are equal to $\bar{\boldsymbol{x}}$ and $\boldsymbol{P}$. We next apply our known nonlinear function $\boldsymbol{y} = h(\boldsymbol{x})$ to each deterministic vector to obtain transformed vectors. The ensemble mean and covariance of the transformed vectors will give a good estimative of the true mean and covariance of $\boldsymbol{y}$. This is the key to the unscented transformation.

Suppose that $\boldsymbol{x}$ is a $n \times 1$ vector that is transformed by a nonlinear function $\boldsymbol{y} = h(\boldsymbol{x})$. Choose $2n + 1$ sigma points $\boldsymbol{x}^{(i)}$ as follows:

$$\begin{aligned} \boldsymbol{x}^{(0)} &= \boldsymbol{x}_{k-1}^+ \\ \boldsymbol{x}_{k-1}^{(i)} &= \boldsymbol{x}_{k-1}^+ + \tilde{\boldsymbol{x}}^{(i)} && i = 1, \ldots, 2n \\ \tilde{\boldsymbol{x}}^{(i)} &= \left(\sqrt{(n+\kappa)\,\boldsymbol{P}_{k-1}^+}\right)_i^T && i = 1, \ldots, n \\ \tilde{\boldsymbol{x}}^{(n+i)} &= -\left(\sqrt{(n+\kappa)\,\boldsymbol{P}_{k-1}^+}\right)_i^T && i = 1, \ldots, n \end{aligned} \tag{25}$$

where $\kappa \in \Re$ and $\sqrt{(n+\kappa)\,\boldsymbol{P}}$ is the matrix square root of the $(n+\kappa)\,\boldsymbol{P}$ such that, the relationship is true $\left(\sqrt{(n+\kappa)\,\boldsymbol{P}}\right)^T \sqrt{(n+\kappa)\,\boldsymbol{P}} = (n+\kappa)\,\boldsymbol{P}$, and $\left(\sqrt{(n+\kappa)\,\boldsymbol{P}}\right)_i$ is the ith row of the $\sqrt{(n+\kappa)\,\boldsymbol{P}}$.

Propagate from time step $(k-1)$ to k, the known nonlinear system equation $f(.)$ to transform the sigma points into $\boldsymbol{x}_k^{(i)}$ vectors as follows:

$$\hat{\boldsymbol{x}}_k^{(i)} = f(\hat{\boldsymbol{x}}_{k-1}^{(i)}, t_k) \qquad i = 0, \ldots, 2n \tag{26}$$

Combine the $\hat{\boldsymbol{x}}_k^{(i)}$ vector to obtain the *a-priori* state estimate at time k.

$$\hat{\boldsymbol{x}}_k^- = \sum_{i=0}^{2n} W^{(i)} \hat{\boldsymbol{x}}_k^{(i)} \tag{27}$$

The $(2n + 1)$ weighting coefficients is denoted as follows:

$$\begin{aligned} W^{(0)} &= \frac{\kappa}{n+\kappa} \\ W^{(i)} &= \frac{1}{2\,(n+\kappa)} \qquad i = 1, \ldots, 2n \end{aligned} \tag{28}$$

Estimate the *a-priori* error covariance. We should add $\boldsymbol{Q}_{k-1}$ to the end of the equation to take the process noise into account:

$$\boldsymbol{P}_k^- = \sum_{i=0}^{2n} W^{(i)} \left(\hat{\boldsymbol{x}}_k^{(i)} - \hat{\boldsymbol{x}}_k^-\right)\left(\hat{\boldsymbol{x}}_k^{(i)} - \hat{\boldsymbol{x}}_k^-\right)^T + \boldsymbol{Q}_{k-1} \tag{29}$$

Use the known nonlinear measurements equations $h(.)$ to transform the sigma points into $\hat{\boldsymbol{y}}_k^{(i)}$ vectors (predicted measureaments):

$$\hat{\boldsymbol{y}}_k^{(i)} = h(\hat{\boldsymbol{x}}_k^{(i)}, t_k) \qquad i = 0, \ldots, 2n \tag{30}$$

Combine the $\hat{\boldsymbol{y}}_k^{(i)}$ vectors to obtain the predicted measurement at time k.

$$\hat{\boldsymbol{y}}_k = \sum_{i=0}^{2n} W^{(i)} \hat{\boldsymbol{y}}_k^{(i)} \tag{31}$$

---

1) A multimodal probability density functions have more than one local maxima

Estimate the covariance of the predicted measurement. We should add $\boldsymbol{R}_k$ to the end of the equation to take the measurement noise into account:

$$\boldsymbol{P}_y = \sum_{i=0}^{2n} W^{(i)} \left(\hat{\boldsymbol{y}}_k^{(i)} - \hat{\boldsymbol{y}}_k\right)\left(\hat{\boldsymbol{y}}_k^{(i)} - \hat{\boldsymbol{y}}_k\right)^T + \boldsymbol{R}_k \tag{32}$$

Estimate the cross covariance between $\hat{\boldsymbol{x}}_k^-$ and $\hat{\boldsymbol{y}}_k$ as follows:

$$\boldsymbol{P}_{xy} = \sum_{i=0}^{2n} W^{(i)} \left(\hat{\boldsymbol{x}}_k^{(i)} - \hat{\boldsymbol{x}}_k^-\right)\left(\hat{\boldsymbol{y}}_k^{(i)} - \hat{\boldsymbol{y}}_k\right)^T \tag{33}$$

The measurement update of the state estimation can be performed using the normal Kalman Filter equations [1)] as follows:[1,9)]

$$\begin{aligned} \boldsymbol{K}_k &= \boldsymbol{P}_{xy}\boldsymbol{P}_y^{-1} \\ \hat{\boldsymbol{x}}_k^+ &= \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k\left(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k\right) \\ \boldsymbol{P}_k^+ &= \boldsymbol{P}_k^- - \boldsymbol{K}_k\boldsymbol{P}_y\boldsymbol{K}_k^T \end{aligned} \tag{34}$$

6. Computer Simulation and Results

The nonlinear system that represents the process and measurements equations of the problem is given by:[19)]

$$\begin{bmatrix} \dot{\boldsymbol{q}} \\ \dot{\boldsymbol{\varepsilon}} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\boldsymbol{\Omega}_\omega & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} \end{bmatrix} \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{\varepsilon} \end{bmatrix} + \boldsymbol{w} \tag{35}$$

$$\boldsymbol{y}_k = \begin{bmatrix} \arctan\left(\dfrac{-S_y}{S_x\cos 60^\circ + S_z\cos 150^\circ}\right) \\ 24^\circ + \arctan\left(\dfrac{S_x}{S_z}\right) \\ \phi \\ \theta \end{bmatrix} + \boldsymbol{\upsilon}_k \tag{36}$$

where S_x, S_y and S_x are the components of the unit vector associated to the sun vector in the satellite system.

Equation (35) represent the kinematic equations of the problem and the Eq. (36) represent the model measurements given by two Digital Sun Sensor (DSS) and two Infrared Earth Sensor (IRES), respectively.[6,18–20)]

The initial conditions used in Euler angles were: $\boldsymbol{x}_0 = \begin{bmatrix} 0.0 & 0.0 & 0.0 & 5.76 & 4.83 & 2.68 \end{bmatrix}^T$; the covariance matrix whichs weigh the initial conditions $\boldsymbol{P}_0 = diag\,(0.25; 0.25; 4.0; 1.0; 1.0; 1.0)$; the process error matrix $\boldsymbol{Q}_0 = diag\left(6.08; 5.47; 6.08; 4\times10^{-3}; 4\times10^{-3}; 4\times10^{-3}\right)\times 10^{-3}$, which weighs the process noise, and the measurements error matrix $\boldsymbol{R}_0 = diag\,(0.36; 0.36; 0.0036; 0.0036)$, which weighs the measurements noise. For the vector $\boldsymbol{x}_0$, the first three elements are in deg and the others elements are in deg/h, for the matrices $\boldsymbol{P}_0$, $\boldsymbol{Q}_0$ and $\bar{\boldsymbol{P}}_0$ first three elements are in deg^2 and the others elements are in deg^2/h^2, and finally, for the matrix $\boldsymbol{R}_0$ all the elements are in deg^2.

6.1. Simulated Data

The orbit and attitude simulations were made by propagator PROPAT[21)] coded in MatLab software and presented here with a sampling rate of the $0.5s$ for $10min$ of observation.

In the estimation states process using the UKF were used $2n+1$ sigma points, where n is the number of states. Already in the estimation states process using the RPF were used $N = 500$ particles, respectively.

Figures 2 and 3 as follow, shows to the attitude and gyros bias estimated states, respectively, the resampling of the a-posteriori particle distribution for the RPF in the time step of $t = 99s$ to $t = 124s$ using the measurements of DSS_1.

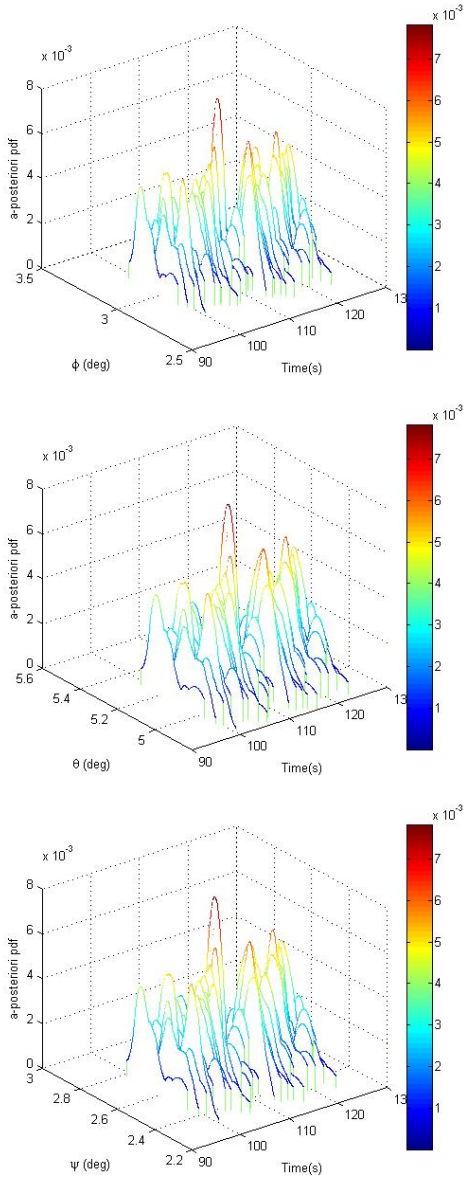

Fig. 2. Resampling of the *a-posteriori* particle distribution for attitude estimated using the RPF for the measurements of DSS_1, in the time step of $t = 99s$ to $t = 124s$.

1) The Normal Kalman Filter is statistical derivation of the Kalman Filter

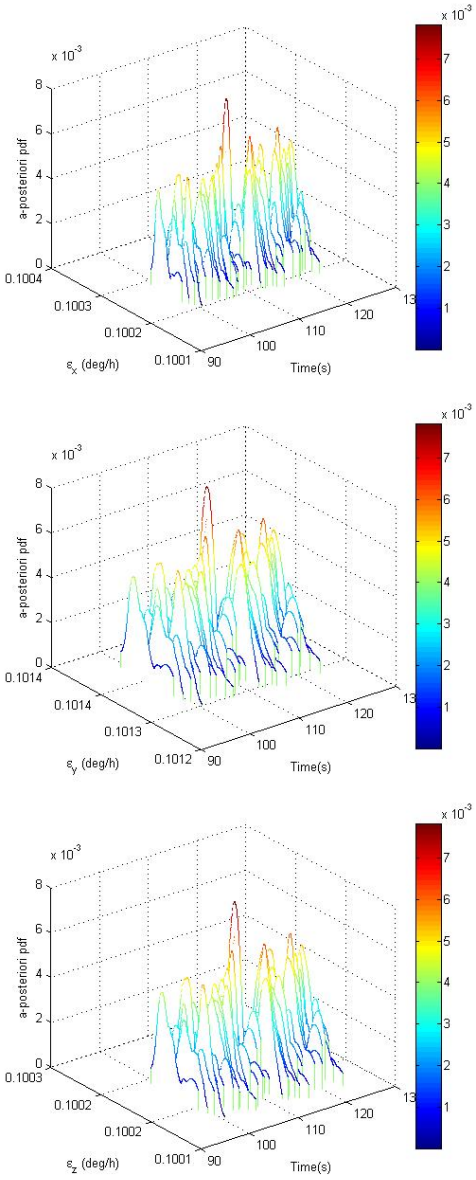

Fig. 3. Resampling of the *a-posteriori* particle distribution for gyros bias estimated using the RPF for the measurements of DSS_1, in the time step of $t = 99s$ to $t = 124s$.

Figures 4 and 5, present the attitude angles and gyros bias estimation using the RPF for $N = 500$ particles comparated with UKF.

Before analyzing the accuracy of the filters in question, it is important to analyze their convergencedone through configuration of residual for the two DSS and for the two IRES, in the Fig. 6 shows the residual frequency for DSS and IRES for each of the filters in the analysis (RPF and UKF), presenting characteristics similar to a Gaussian distribution

It is said that a Filter is converging when your residual is close to zero average and the Tab. 1 shows the average value and the standard deviation of the DSS and IRES residuals for each of the filters presented in Fig. 6.

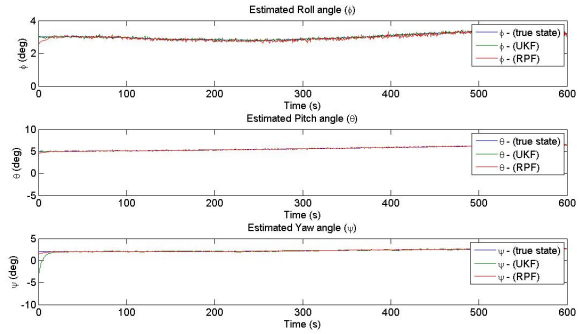

Fig. 4. Attitude angles estimated (ϕ, θ, ψ)

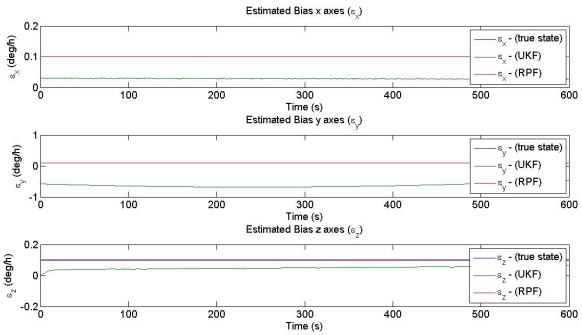

Fig. 5. Gyros bias estimated $(\epsilon_x, \epsilon_y, \epsilon_z)$

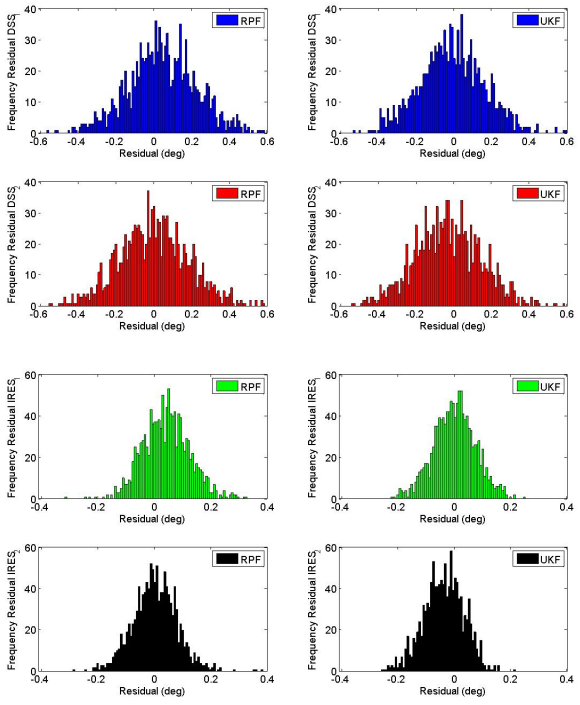

Fig. 6. Frequency Residuals for the DSS and IRES on board the CBERS-2 satellite

Table 1. Mean and standard deviation statistics of the DSS and IRES Residuals

	RPF	UKF
DSS_1 Res. (deg)	0.0481 ± 0.1918	0.0273 ± 0.9766
DSS_2 Res. (deg)	0.0038 ± 0.1985	-0.0310 ± 1.0472
$IRES_1$ Res. (deg)	0.0438 ± 0.0890	0.0026 ± 0.1065
$IRES_2$ Res. (deg)	0.0060 ± 0.0918	-0.0334 ± 0.1693

The standard deviation of the Residuals is calculated by Eq (37):

$$\sigma = \sqrt{\frac{1}{\tilde{K}} \sum_{k=1}^{\tilde{K}} (\tilde{\boldsymbol{y}}_k - \bar{\boldsymbol{y}})^2} \tag{37}$$

where $\bar{\boldsymbol{y}} = \frac{1}{\tilde{K}} \sum_{k=1}^{\tilde{K}\tilde{K}} \tilde{\boldsymbol{y}}_k$ and $\tilde{K}$ is the total number of estimation residuals.

It is said that a Filter is converging when your residual is close to zero average and it happens with the results presented in Table 1. To analyze the accuracy of the filters studied, it is presented, in Fig. 7, the error attitude estimation by the RPF and the UKF.

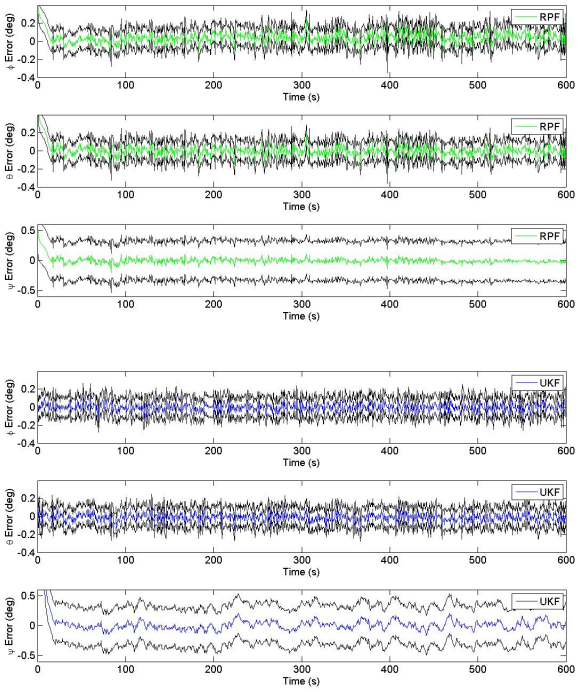

Fig. 7. Error attitude estimation

Table 2 shows the average value and the standard deviation of the error attitude estimation presented in Figure (7)

Table 2. Mean and standard deviation statistics of the error attitude estimation

	RPF	UKF
ϕ Error (deg)	0.0441 ± 0.0648	0.0015 ± 0.0493
θ Error (deg)	0.0041 ± 0.0686	-0.0085 ± 0.0551
ψ Error (deg)	$-5.470 \times 10^{-4} \pm 0.0572$	0.0507 ± 0.3679

With a small change, the standard deviation of the error state estimation is calculated by Eq. (38):

$$\sigma = \sqrt{\frac{1}{\tilde{K}} \sum_{k=1}^{\tilde{K}} (\tilde{\boldsymbol{x}}_k - \bar{\boldsymbol{x}})^2} \tag{38}$$

where $\tilde{\boldsymbol{x}}_k = \boldsymbol{x}_k - \hat{\boldsymbol{x}}_k$, $\bar{\boldsymbol{x}} = \frac{1}{\tilde{K}} \sum_{k=1}^{\tilde{K}} \tilde{\boldsymbol{x}}_k$ and $\tilde{K}$ is the total number of estimation.

Analyzing the Table 2 it can be seen that, the average results for the error attitude estimation in the RPF are better when compared with the UKF, especially for the ψ Error angles.

The following, in Fig. (8) presented the error bias estimation by the RPF and the UKF.

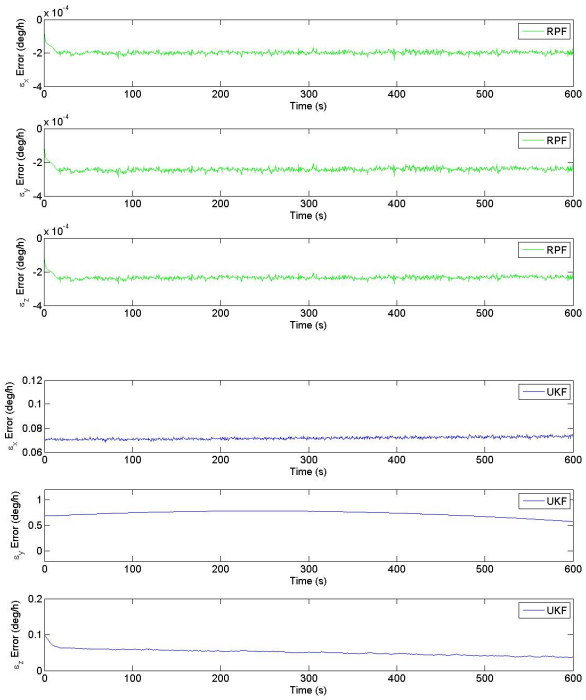

Fig. 8. Error gyros bias estimation

Table 3 shows the average value and the standard deviation of the error bias estimation presented in Fig. (8).

Table 3. Mean and standard deviation statistics of the error bias estimation

	RPF	UKF
ε_x Error (deg/h)	$-1.9 \times 10^{-4} \pm 1.1 \times 10^{-5}$	0.071 ± 0.001
ε_y Error (deg/h)	$-2.4 \times 10^{-4} \pm 1.3 \times 10^{-5}$	0.725 ± 0.054
ε_z Error (deg/h)	$-2.3 \times 10^{-4} \pm 1.1 \times 10^{-5}$	0.050 ± 0.008

Analyzing the Table 3 it can be seen that, the average results for the error bias estimation in the RPF again are beter when compared with the UKF.

7. Conclusions

The main objective of this study was to estimate the attitude of a CBERS-2 like satellite, using real data provided by sensors that are on board the satellite. To verify the consistency of the estimator, the attitude was estimated by least squares method. The usage of real data from on-board attitude sensors, poses difficulties like mismodelling, mismatch of sizes, misalignments, unforeseen systematic errors and post-launch calibration errors.

In general terms, for nonlinear system, the Kalman filtering can be used for state estimation, but the Particle filtering may give better results at the price of additional computational effort. In system that has non-Gaussian noise, the Kalman filtering is optimal linear filter, but again the Particle filtering may perform better. The Unscented Kalman Filter provides a balance between the low computational effort of te Kalmam filtering and the high performance of the Particle filtering.[9)]

It should be added that there is some similarity between the filters, the UKF transform a set of points via known nonlinear equations and combines the results to estimate the mean and covariance of the state. However, in the Particle Filtering (and the RPF used in this work) the points (the cloud of particles) are chosen randomly, whereas in the UKF the points (the cloud of sigma points) are chosen on the basis of a specific algorithm.

Because of this, the number of points used in a Particle Filter generally needs to be much greater than the number of points in a Unscented Kalman Filter and, consequently, with this large number of particles for the Particle Filtering the computational effort will increase

Finally, this work provide a kinematic attitude solution besides estimating biases (gyro drifts) and it can be concluded that the algorithm of the RPF converges and the results are in close agreement with the results in previou work,[6)] which used the UKF for estimation of attitude and gyros bias.

Acknowledgments

The authors would like to thank the financial support received by CAPES, FAPESP 2012/ 21023-6, CNPQ 303119/ 2010-1, and the partial support from project SIA-DCTA-INPE under contract FINEP 0.1.06.1177.03

References

1) Crassidis, J. L. and Junkins, J. L.: *Optimal Estimation of Dynamic Systems*. New York: Chapman and Hall/CRC Applied Mathematics and Nonlinear Science, 2011.
2) Silva, W. R., Zanardi, M. C., Cabette, R. E. S. and Formiga, J. K. S.: *Study of stability of rotational motion of spacecraft with canonical variables*,Mathematical Problems in Engeneering, v. 2012, p. 1 - 19, 2012.
3) Shuster, M. D.: *A Survey of Attitude Representations*, Journal of Astronautical Sciences, v. 41, p. 439 - 517, 1993.
4) Wertz, J. R.: *Spacecraft attitude determination and control*. Dordrecht, Holanda: D. Reidel, 1978.
5) Lefferts, E. J., Markley, F. L. and Shuster, M. D.: *Kalman filtering for spacecraft attitude estimation*, Journal of Guidance, v.5, p. 417 - 429, 1982.
6) Garcia, R. V., Kuga, H. K. and Zanardi, M. C.: *Unscented Kalman filter applied to the spacecraft attitude estimation with euler angles*, Mathematical Problems in Engineering, Vol. 2012, 2012, pp. 112.
7) Metropolis, N. and Ulam, S.: *The Monte Carlo method*, Journal of the American Statistical Association, Vol. 44, No. 247, 1949, pp. 335 341.
8) Wiener, N.: *I am a Mathematician*. Cambridge, Massachesetts: MIT Pres, 1956.
9) Simon, D.: *Optimal State Estimation: Kalman, H1, and Nonlinear Approaches*. New York: Wiley, 2006.
10) Smith, A. and Gelfand, A.: *Bayesian statistics without tears: A sampling resampling perpective*, The American Statistician, Vol. 46, No. 2, 1992, pp. 84 88.
11) Arulampalam, M., Maskell, S., Gordon, N. and Clapp, T.: *A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking*, IEEE Transactions on Signal Processing, Vol. 50, No. 2, 2002, pp. 174 188.
12) Gordon, N., Salmond, D. and Smith, A.: *Novel approach to nonlinear/non-Gaussian Bayesian state estimation*, IEEE Proceedings-F, Vol. 140, No. 2, 1993, pp. 107 1813.
13) Doucet, A., Freitas, N. D. and Gordon, N.: *Sequential Monte Carlo Methods in Practice*. New York: Springer-Verlag, 2001.
14) Ristic, B., Arulampalam, S. and Gordon, N.: *Beyound the Kalman Filter: Particle Filters for Tracking Applications*. Massachusetts: Artech House, Norwell, 2004.
15) Coxeter, H.: *Regular Polytopes*. New York: Dover, 1973.
16) Silverman, B.: *Density Estimation for Statistics and Data Analysis*. London: Chapman and Hall, 1986.
17) Julier, S. J. and Uhlmann, J. K.: *A new extension of the Kalman Filter for nonlinear systems*, Signal Processing, Sensor Fusion, and Target Recognition, Vol. 3068, 1997, pp. 182 193.
18) Fuming, H. and Kuga, H. K.: *CBERS simulator mathematical models*, CBTT Project, CBTT/ 2000/ MM/ 001, 1999.
19) Silva, W. R., Kuga, H. K. and Zanardi, M. C.: *Application fo the Extended H1 Filter for Attitude Determination and Gyro Calibration*, Advances in the Astronautical Sciences, 2014, pp. 1501 1515.
20) Garcia, R. V., Kuga, H. K. and Zanardi, M. C.: *Unscented Kalman filter for spacecraft attitude estimation using quaternions and euler angles*, 22nd International Symposium on Space Flight Dynamics, 2011.
21) Carrara, V.: *PROPAT Satellite Attitude and Orbit Toolbox for Matlab*, Avaliable in: http://www2.dem.inpe.br/val/projetos/propat/default.htm, 1994.
22) Lopes, R. V. F. and Kuga, H. K.: *CBERS-2: On Ground Attitude Determination From Telemetry Data*, INPE, Internal Report C-ITRP, 2005.