Ben-Gurion University of the Negev
Mechanical Engineering Department
362-2-6752 - Kalman Filtering for Attitude Estimation

Project

The main purpose of this project is to let you design an Extended Kalman Filter for spacecraft Attitude Determination, and demonstrate its performance by means of simulations.

Prelude

Determining the attitude of a spacecraft means to estimate its spatial orientation with respect to some reference frame. This operation is central when the spacecraft attitude needs to be controlled; telecommunication satellite, for instance, require to have their antennas panel constantly oriented toward a specific relay station on the Earth. As some of you will learn in subsequent course, the control performance of a system under uncertainty are highly dependent on its estimation performance. This project is a practical introduction to the attitude estimation problem.

A Physical Model for the Attitude

Consider the spacecraft a rigid body, and, attached to it, a Cartesian coordinate frame, $\mathcal{B}$, which orientation with respect to some Cartesian inertial reference frame, $\mathcal{R}$, defines the inertial attitude of the spacecraft. There are several ways of mathematically representing the attitude; here, you will deal with the *quaternion of rotation*, $\mathbf{q}$, which has won constant popularity thanks to its nice computational properties. For our purpose, $\mathbf{q}$ is unit vector in $\mathbb{R}^4$, with a vector part $\mathbf{e}$ and a scalar part q; that is

$$\mathbf{q} = \begin{bmatrix} \mathbf{e} \\ q \end{bmatrix} \qquad\qquad \mathbf{q}^T\mathbf{q} = 1 \tag{1}$$

Determining the attitude, thus, consists in estimating the vector $\mathbf{q}$. It is known from kinematics theory[1] that the most general motion of a rigid body around its center of mass is a rotation which can be instantaneously characterized by an axis of rotation and an angular rate of rotation. A 3×1 physical vector, called the *angular velocity vector*, denoted by $\boldsymbol{\omega}_t$, is defined by a direction that coincides with that of the instantaneous rotation axis[2] and by a length equal to the angular rate of rotation. The rigid-body kinematics law in terms of $\mathbf{q}_t$ provides us with a vector differential equation for $\mathbf{q}_t$ as follows

$$\dot{\mathbf{q}}_t = \frac{1}{2}\Omega_t \mathbf{q}_t \tag{2}$$

where the 4×4 skew-symmetric matrix Ω_t is defined as

$$\Omega_t \triangleq \begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix} \tag{3}$$

and ω_1, ω_2, and ω_3 are the components of $\boldsymbol{\omega}_t$ along the axis of the body frame $\mathcal{B}$.

[1]Kinematics is concerned with the relationships governing the motion of bodies independently from the forces which create that motion

[2]and according to the right-hand rule; that is, when the right-hand thumb points in the direction of $\boldsymbol{\omega}_t$, the fingers indicate the direction of rotation

Sensors

It is assumed that a triad of *rate gyroscopes* measure $\boldsymbol{\omega}_t$ with some additive errors:

$$\boldsymbol{\omega}_t^m = \boldsymbol{\omega}_t + \boldsymbol{\mu}_t + \boldsymbol{\epsilon}_t \tag{4}$$

where $\boldsymbol{\epsilon}_t$ is a continuous zero-mean Gaussian white noise with intensity matrix $\sigma_\epsilon^2 I_3$, and $\boldsymbol{\mu}_t$ is a slowly varying drift. The following model equation is used for $\boldsymbol{\mu}_t$:

$$\dot{\boldsymbol{\mu}}_t = \mathbf{n}_t \tag{5}$$

where $\mathbf{n}_t$ is a continuous zero-mean Gaussian white noise with intensity matrix $\sigma_n^2 I_3$, and the initial condition for $\boldsymbol{\mu}_t$ is $\boldsymbol{\mu}_0$.

We will also assume that a *star-tracker* is mounted on-board the spacecraft and measures alternatively the directions to two different stars.[3] The measurement model equation corresponding to this type of sensor is developed as follows. Consider the physical unit vector, $\mathbf{u}$, that lies along the line of sight from the spacecraft to the observed star. The 3×1 vector representations of $\mathbf{u}$ in the two Cartesian frames $\mathcal{B}$ and $\mathcal{R}$ will be denoted by $\mathbf{b}^o$ and $\mathbf{r}$, respectively; they are related by the matrix of rotation from $\mathcal{R}$ to $\mathcal{B}$, A, also called the attitude matrix:

$$\mathbf{b}^o = A\mathbf{r} \tag{6}$$

It is known that the attitude matrix A can be expressed as a quadratic function of the elements of $\mathbf{q}$ as follows:

$$A(\mathbf{q}) = \left(q^2 - \mathbf{e}^T\mathbf{e}\right) I_3 + 2\mathbf{e}\mathbf{e}^T - 2\,q\,[\mathbf{e}\times] \tag{7}$$

where $[\mathbf{e}\times]$ is the cross-product matrix associated with $\mathbf{e}$ and is defined as

$$[\mathbf{e}\times] \triangleq \begin{bmatrix} 0 & -e_3 & e_2 \\ e_3 & 0 & -e_1 \\ -e_2 & e_1 & 0 \end{bmatrix} \tag{8}$$

Usually, $\mathbf{b}^o$ is measured - here by a star-tracker - while $\mathbf{r}$ is relatively accurately known from tables or almanacs. The measurement noise, $\boldsymbol{\delta b}$, is assumed to be an additive zero-mean white Gaussian noise with covariance matrix $\sigma_b^2 I_3$. The discrete-time measurement equation at time t_k is thus expressed as

$$\mathbf{b}_k = \mathbf{h}_k(\mathbf{q}_k) + \boldsymbol{\delta b}_k \tag{9}$$

where

$$\mathbf{h}_k(\mathbf{q}_k) = A(\mathbf{q}_k)\mathbf{r}_k \tag{10}$$

where the non-linear function $\mathbf{h}(\mathbf{q})$ is defined by substituting Eqs. (7) and (8) in Eq. (6). The sampling time of the star-tracker is 2 seconds, and two stars are alternatively observed. Assuming that the stars are far enough from the spacecraft, the associated reference vectors $\mathbf{r}^1$ and $\mathbf{r}^2$ are considered time-invariant. The initial vectors $\mathbf{q}_0$ and $\boldsymbol{\mu}_0$, and the noises $\boldsymbol{\delta b}_k$, $\boldsymbol{\epsilon}_t$, and $\mathbf{n}_t$ are mutually uncorrelated.

[3] Some star-trackers can simultaneously track tens of celestial objects but the minimal number of two is enough for the scope of this project.

Part 1: Mathematical Model

1. Define an augmented state vector $\mathbf{x}^T \stackrel{\triangle}{=} [\mathbf{q}^T, \boldsymbol{\mu}^T]$ and develop a non-linear state-variable model for $\mathbf{x}_t$ with a continuous-time process equation and a discrete-time measurement equation. The model equations will be written in the following form:

$$\dot{\mathbf{x}}_t = \mathbf{f}_t(\mathbf{x}_t) + G_t(\mathbf{x}_t)\,\mathbf{w}_t \tag{11}$$
$$\mathbf{z}_{k+1} = \mathbf{h}_{k+1}(\mathbf{x}_{k+1}) + \mathbf{v}_{k+1} \tag{12}$$

2. Develop the discrete-time linear perturbation model that is associated with the non-linear model. The model equations will be written in the following form:

$$\boldsymbol{\delta}\mathbf{x}_{k+1} = F_x(\mathbf{x}^*_k)\,\boldsymbol{\delta}\mathbf{x}_k + G(\mathbf{x}^*_k)\;\mathbf{w}_k \tag{13}$$
$$\boldsymbol{\delta}\mathbf{z}_{k+1} = H_x(\mathbf{x}^*_{k+1})\,\boldsymbol{\delta}\mathbf{x}_k + \mathbf{v}_{k+1} \tag{14}$$

where $\mathbf{x}^*_k$ is the nominal trajectory about which the non-linear model is linearized.

Part 2: Extended Kalman Filter

1. Using the results of Part 1, write down the algorithm of a discrete-time Additive Extended Kalman Filter (AEKF) for estimation of the quaternion, $\mathbf{q}$, and of the gyro drift, $\boldsymbol{\mu}$. State carefully, whenever it is necessary, how the nominal state trajectory, $\mathbf{x}^*_k$, is chosen. The AEKF thus obtained will be denoted by AEKF1.

2. As you can notice from the measurement update stage in AEKF1, the a posteriori estimate of the quaternion is not necessarily of unit length, although the true quaternion certainly is. This issue is addressed in two different ways.

   (a) After the measurement update stage of AEKF1, add the following *brute-force* normalization step of the quaternion estimate:

$$\widehat{\mathbf{q}}^*_{k/k} = \frac{\widehat{\mathbf{q}}_{k/k}}{\|\widehat{\mathbf{q}}_{k/k}\|} \tag{15}$$

   The modified filter, which consists of AEKF1 and of the quaternion normalization step, Eq. (15), will be called AEKF2.

   (b) The technique of *pseudo-measurement* can constrain the estimation process to comply with the unit-length property of $\mathbf{q}$. This is done as follows. Consider the unit-length constraint equation:

$$1 = \mathbf{q}_k^T\mathbf{q}_k \tag{16}$$

   The idea is to imagine a fictitious device which output would be the norm of the quaternion.[4] Equation (16) represents a perfect measurement which, in the case of linear systems, is handled by reduced estimation; in our case, however, the non-linear nature of Eq. (16) would destroy some nice properties of the current model, which leads us to use the following trick: we add a fictitious noise term, $\mathbf{v}^{nor}_k$ to the right-hand-side of Eq. (16), and we obtain, thus, a legitimate non-linear measurement equation:

$$\mathbf{z}_k = \mathbf{g}_k(\mathbf{x}_k) + \mathbf{v}^{nor}_k \tag{17}$$

   where $\mathbf{z}_k = 1$ and $\mathbf{g}_k(\mathbf{x}_k) = \mathbf{q}_k^T\mathbf{q}_k$, and $\mathbf{v}^{nor}_k$ is assumed to be a zero-mean white Gaussian noise with variance σ^2_{nor}. The parameter σ_{nor} is a measure of our trust in the

[4] That device doesn't exist so its output is called pseudo-measurement.

pseudo-measurement (small variance meaning high trust and high variance meaning low trust), and provides us with a mean of enforcing the unit-length property along the estimation process.

Develop the measurement update stage that is associated with the pseudo-measurement of Eq. (17), and add it to the measurement update stage of AEKF1: this filter will be called AEKF3. Note that the value of σ_{nor} is not given a priori. It is a design parameter that you will determine throughout the simulations when tuning the filter.

3. Using the same approach as developed in the course, write down the algorithm of a discrete-time Multiplicative Extended Kalman Filter (MEKF) for estimation of the quaternion $\mathbf{q}$ and of the gyro drift $\boldsymbol{\mu}$. The filter thus developed will be called MEKF.

Part 3: Simulation

The time step for all simulations will be $\Delta t = 0.1$ second. It is, by assumption, the same time step as that of the discretization procedure in Part 1. The simulation runs will last 6000 seconds.

1. Run the four filters AEKF1, AEKF2, AEKF3 and MEKF, starting with the same initial conditions for the filter estimate, $\widehat{\mathbf{x}}_{0/0} = [0, 0, 0, 1, 0, 0, 0]$, and for the estimation error covariance matrix, $P_{0/0} = 5I_7$. Use the filter's noise variances to tune the filters. Compare the results. Conclusion.

2. Same thing but with the following initial conditions for the estimate: $\widehat{\mathbf{q}}_{0/0} = [0.3780, 0.7560, 0.3780, -0.3780^T]$, and $\widehat{\boldsymbol{\mu}}_{0/0} = 200\,[1, 1, 1]\,\frac{deg}{hour}$.
Compare the results. Conclusion. Choose the best of the four filters according to the simulation performances. That filter will be called BEKF.

3. Using BEKF perform a Monte-Carlo (MC) simulation over 50 runs. Compare the MC-standard deviations and the filter standard deviations of the quaternion estimation error.

4. Using BEKF, run three simulations with $\widehat{\mathbf{x}}_{0/0} = [0, 0, 0, 1, 0, 0, 0]$, and $P_{0/0}$ taking the three different values, $P_{0/0} = 0.001I_7$, $P_{0/0} = I_7$, and $P_{0/0} = 100I_7$. Compare the results. Conclusion.

5. Using BEKF, run three simulations with three different sampling time for the star-tracker: 1 second, 5 seconds, and 25 seconds. Compare the results. Conclusion.

Numerical data

$$\boldsymbol{\omega}(t) = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \sin(2\pi t/150) \quad \frac{deg}{sec}$$

$$\sigma_\epsilon^2 = 10^{-13} \quad \frac{rad^2}{sec}$$

$$\sigma_n^2 = 10^{-11} \quad \frac{rad^2}{sec^3}$$

$$\sigma_b = 100 \quad \text{arcsec}$$

$$\mathbf{q}_0 = \begin{bmatrix} 0.3780 & -0.3780 & 0.7560 & 0.3780 \end{bmatrix}^T$$

$$\boldsymbol{\mu}_0 = 3.10^{-5} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad \frac{rad}{sec}$$

$$\begin{bmatrix} \mathbf{r}^1 & \mathbf{r}^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

How to Present the Results ?

The results will be presented in three distinct parts as indicated above.

With regards to the simulation part, the results for each question will at least consist of two graphs. One graph will depict the time variations of the angular estimation error, $\delta\phi$, which is the angle of the small rotation that brings the estimated body-frame onto the true body-frame (see how to compute $\delta\phi$ in Appendix A.) In question no. 3, the MC-mean of $\delta\phi$ will be plotted. (See Appendix B for the computation of the MC-mean.)
A second graph will show the filter standard deviations of the quaternion estimation error; that is, the square-root of the elements on the principal diagonal of the matrix $P_{k/k}$. In question no.3, the filter standard deviations will be plotted on the same graph as the MC standard deviation for the sake of comparison. (See Appendix B for the computation of the MC-standard deviations.)

Appendix A

At each time step t_k, the angular estimation error, $\delta\phi_{k/k}$, is computed as follows:

1. Compute $\mathbf{q}_k$ and $\widehat{\mathbf{q}}_{k/k}$,
2. Compute $A_k = A(\mathbf{q}_k)$ and $\widehat{A}_{k/k} = A(\widehat{\mathbf{q}}_{k/k})$,
3. Compute $\delta\widehat{A}_{k/k} = A_k \widehat{A}_{k/k}^{-1}$,
4. Compute $\delta\phi_{k/k} = \arccos\{\frac{1}{2}[\,\mathrm{tr}(\delta\widehat{A}_{k/k}) - 1]\}$.

Appendix B

Given N realizations of a vector random process $\mathbf{e}(k,l)$, $k = 1, 2, \ldots$, and $n = 1, 2, \ldots, N$, the MC-mean and the MC-standard deviations over N realizations of $\mathbf{e}(k)$ are denoted by $\mathbf{m}_e^N(k)$ and $\mathbf{v}_e^N(k)$, respectively. They are defined as follows:

$$\mathbf{m}_e^N(k) = \frac{1}{N}\sum_{n=1}^{N}\mathbf{e}(k,n)$$
$$\mathbf{v}_e^N(k) = \frac{1}{N}\sum_{n=1}^{N}[\mathbf{e}(k,n) - \mathbf{m}_e^N(k)]^2$$

In practice, you can use the following recursion rules to compute them:

$$\mathbf{m}_e^n(k) = \frac{n-1}{n}\,\mathbf{m}_e^{n-1}(k) + \frac{1}{n}\,\mathbf{e}(k,n)$$
$$\mathbf{v}_e^n(k) \simeq \frac{n-1}{n}\,\mathbf{v}_e^{n-1}(k) + \frac{1}{n}\,[\mathbf{e}(k,n) - \mathbf{m}_e^n(k)]^2$$

The MC-standard deviation is simply the square-root of the MC-variance.