Torus test report

Hugo Fidelin

September 2024

Introduction

This report document my work on the Kalman filter exercise. The source code can be found in my GitHub repository `https://github.com/hfidelin/Torus_test`, Numpy and Matplotlib are required.

Contents

1 Preliminary work

After reading carefully the problem, the first objective was to recreate the different figure of the trajectory of the satellites and the receiver.

The aim of this exercise is to estimate the receiver position and ambiguity number. A state vector is defined as follows :

$$\mathbf{x} = \begin{pmatrix} \mathbf{p} \\ \mathbf{v} \\ \mathbf{n} \end{pmatrix}$$

With :

- $\mathbf{p} = \begin{pmatrix} p_x \\ p_y \end{pmatrix} \in \mathbb{R}^2$ the position of the receiver,
- $\mathbf{v} = \begin{pmatrix} v_x \\ v_y \end{pmatrix} \in \mathbb{R}^2$ the **constant** velocity of the receiver,
- $\mathbf{n} = \begin{pmatrix} n_0 \\ \vdots \\ n_M \end{pmatrix} \in \mathbb{Z}^M$ the ambiguity vector.

To implement properly the Kalman filter algorithm, the explicit operator presented in the problem are needed. Some of them are given in the document. At step $k\mathbb{N}$ the operator can be expressed :

- the **state transition** matrix, $\mathbf{F}_k \in \mathbb{R}^{(4+M)\times(4+M)}$ is explicit :

$$\mathbf{F}_k = \begin{pmatrix} 1 & 0 & \Delta t & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \Delta t & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

- the **process noise covariance** matrix, $\mathbf{Q}_{x,k} \in \mathbb{R}^{(4+M)\times(4+M)}$ is also explicit :

$$\mathbf{Q}_{x,k} = \begin{pmatrix} \sigma_p^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_p^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Another useful matrix is the **Jacobian matrix** of the observation function $\mathbf{H}_k$ defined by

$$\mathbf{H}_k = \frac{\partial \mathbf{h}_k(\mathbf{x})}{\partial \mathbf{x}}$$

Where $\mathbf{h}_k = \begin{pmatrix} \rho_{r,k}^1 \\ \vdots \\ \rho_{r,k}^M \\ \rho_{r,k}^1 \\ \vdots \\ \rho_{r,k}^M \end{pmatrix} \in \mathbb{R}^{2M}$ contains the distance between the satellite and the receiver. For the j-th satellite, the distance is expressed as :

$$\rho_{r,k}^j = \left\| \begin{pmatrix} p_X^j \\ p_Y^j \end{pmatrix} - \begin{pmatrix} p_{X,r} \\ p_{Y,r} \end{pmatrix} \right\|_2$$

With $\|.\|_2$ the Euclidean norm. To obtain explicitly the coefficient of the matrix $\mathbf{H}_k$, we need to differentiate this expression with respect to the component of the state vector $\mathbf{x}$:

$$\frac{\partial h_k}{\partial p_x} = -\frac{p_X^j - p_{X,r}}{\left\| \begin{pmatrix} p_X^j \\ p_Y^j \end{pmatrix} - \begin{pmatrix} p_{X,r} \\ p_{Y,r} \end{pmatrix} \right\|_2}$$

$$\frac{\partial h_k}{\partial p_y} = -\frac{p_Y^j - p_{Y,r}}{\left\| \begin{pmatrix} p_X^j \\ p_Y^j \end{pmatrix} - \begin{pmatrix} p_{X,r} \\ p_{Y,r} \end{pmatrix} \right\|_2}$$

The rest of the Jacobian matrix coefficient is null (the distance $\rho_{r,k}^j$ does not depend on the velocities $\mathbf{v}$ or ambiguity number $\mathbf{n}$).

The last matrix that is given is the covariance matrix associated to measurement noise $\mathbf{R}_k \in \mathbb{R}^{2M \times 2M}$

$$\mathbf{R}_k = \begin{pmatrix} \sigma_\epsilon^2 & & & & & \\ & \ddots & & & & \\ & & \sigma_\epsilon^2 & & & \\ & & & \sigma_\eta^2 & & \\ & & & & \ddots & \\ & & & & & \sigma_\eta^2 \end{pmatrix}$$

With those matrices, the computation of the different required operator is now possible :

- the **residual vector** is obtained by the following

$$\mathbf{v}_k = \mathbf{y}_k - \mathbf{H}_k \mathbf{x}_k$$

  With $\mathbf{y}_k \in \mathbb{R}^{2M}$ the measurement model.

- the $\mathbf{S}_k$ matrix :

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

- the **Kalman gain** matrix $\mathbf{K}_k$:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1}$$

With those operator the two step of Kalman filter can be performed :

- the **prediction step**:

$$\mathbf{x}_{k+1|k} = \mathbf{F}_k \mathbf{x}_{k|k}$$
$$\mathbf{P}_{k+1|k} = \mathbf{F}_k \mathbf{P}_{k|k} \mathbf{F}_k^T + \mathbf{Q}_{x,k}$$

- the **updating step**:

$$\mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + \mathbf{K}_k \mathbf{v}_k$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_{k|k-1}$$

2 Implementation

To implement such method, a python class has been made (see kf.py). This class will have as attribute the global variable (predicted position/velocity of the receiver at iteration k, ambiguity vector, number of satellite...)

The main two methods of this class is the predicted step of Kalman filter, and updating step.

2.1 Prediction step

In this step, we compute first the state matrix $\mathbf{F}_k$ and the process noise covariance matrix $\mathbf{Q}_{x,k}$ and then compute the new state vector and new covariance matrix. This is done in the method called *predict* in kf.py.

2.2 Updating step

The second important method is the updating step, the require the computation of two preliminary operator : the Jacobian matrix $\mathbf{H}_k$ and noise measurement covariance matrix $\mathbf{R}_k$.

2.2.1 $\mathbf{H}_k$ computation

To compute $\mathbf{H}_k$, two methods are use, first, are function that compute the $(i,j) - th$ coefficient : *_coef_H* in kf.py Then, a method that initiate, then build $\mathbf{H}_k$ using the previous function : *_init_H*.

2.2.2 $\mathbf{R}_k$ computation

The measurement noise covariance matrix is a diagonally matrix build directly in the method *_init_R* in kf.py.

2.2.3 Updating step

With those operator, an updated state vector and covariance matrix can be compute using the EKF equation.

3 Result

Sadly, the error in EKF algorithm couldn't be fixed in time. In main.py, some plotting part for the different requirement are still prepared.