Tutorial: State Estimation by Kalman Filtering

ARS4 A22

Philippe Bonnifait, Antoine Lima
{philippe.bonnifait,antoine.lima}@hds.utc.fr

In this tutorial, we estimate the position and velocity of a vehicle moving along a one-dimensional line. It will be split in two two-hour sessions, first dedicated to formalizing the system and implementing a basic Kalman Filter then deriving a more complex observation model and implement an extended Kalman Filter.

1 System modeling

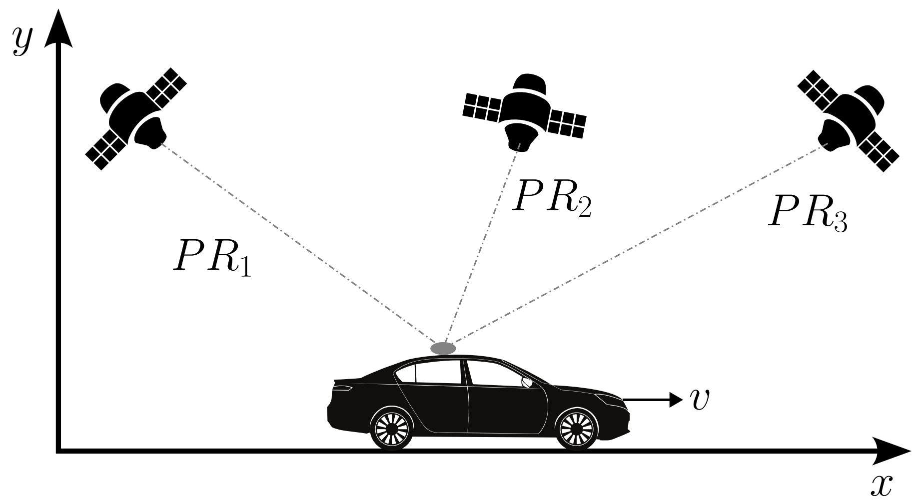

Figure 1: Car driving on a straight road with three satellites above.

We want to model this system with a state vector $\boldsymbol{x}$ and estimate it assuming the following relations:

$$\begin{cases} \dot{\boldsymbol{x}}(t) &= \boldsymbol{f}(\boldsymbol{x}(t)) + \boldsymbol{\alpha}(t) \\ \boldsymbol{y}(t) &= \boldsymbol{g}(\boldsymbol{x}(t)) + \boldsymbol{\beta}(t) \end{cases}$$

with $\boldsymbol{y}$ a snapshot observation of the system, and $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ noises affecting it.

1. Ignoring satellites and focusing on the vehicle for the moment, what are the quantities that $\boldsymbol{x}$ should contain to faithfully describe the system?
2. How does the vehicle evolve? In other words, what is the function $\boldsymbol{f}$ that links $\dot{\boldsymbol{x}}(t)$ to $\boldsymbol{x}(t)$?
3. Discretize the evolution model in matricial form $\boldsymbol{A}$ with a time step Δt, assuming that velocity is constant between two time-steps.
4. Modify the previous discretization to use the average velocity between two time-steps.
5. The limits of our model introduces errors in the estimation. This is usually modeled with a white and zero-centered noise $\boldsymbol{\alpha}(t)$. Its power can be described with a matrix $\boldsymbol{Q}$. What is the form of $\boldsymbol{Q}$?

2 Loosely-coupled GNSS Observation

We now consider the vehicle equipped with a GNSS receiver that always sees enough satellites to compute a position. In this first part, we assume that the receiver processes the satellite signals as a black box and provides a position observation and an associated uncertainty.

1. What is the function $\boldsymbol{g}$ linking the state $\boldsymbol{x}$ and observation $\boldsymbol{y}$ for this sensor?
2. What are the contents of $\boldsymbol{C}$ and $\boldsymbol{R}$ used in the discrete stochastic observation model?
3. Check that this model is observable.
4. Download the provided Matlab / Octave kit and familiarize yourself with the provided data that is organized as:

`Te`	Sampling period
`t`	Time
`gps.x`	Observation done by the GNSS receiver
`gps.sx`	Standard deviation associated to `gps.x`
`strada.x`	True position (to calculate errors)
`strada.v`	True velocity (to calculate errors)

5. Implement a Kalman Filter using the previously derived models and

$$\begin{aligned}
\boldsymbol{x}_{k+1|k} &= \boldsymbol{A}\boldsymbol{x}_{k|k} \\
\boldsymbol{P}_{k+1|k} &= \boldsymbol{A}\boldsymbol{P}_{k|k}\boldsymbol{A}^T + \boldsymbol{Q} \\
\boldsymbol{K} &= \boldsymbol{P}_{k+1|k}\boldsymbol{C}^T\left(\boldsymbol{C}\boldsymbol{P}_{k+1|k}\boldsymbol{C}^T + \boldsymbol{R}\right)^{-1} \\
\boldsymbol{\epsilon} &= \boldsymbol{y} - \boldsymbol{C}\boldsymbol{x}_{k+1|k} \\
\boldsymbol{x}_{k+1|k+1} &= \boldsymbol{x}_{k+1|k} + \boldsymbol{K}\boldsymbol{\epsilon} \\
\boldsymbol{P}_{k+1|k+1} &= \left(\boldsymbol{I} - \boldsymbol{K}\boldsymbol{C}\right)\boldsymbol{P}_{k+1|k}\left(\boldsymbol{I} - \boldsymbol{K}\boldsymbol{C}\right)^T + \boldsymbol{K}\boldsymbol{R}\boldsymbol{K}^T
\end{aligned}$$

6. Find suitable values for $\boldsymbol{Q}$ and $\boldsymbol{R}$[1]. A good position or velocity estimation should be the smallest possible without getting out of the its 3σ bound.

3 Loosly-coupled GNSS with Odometer

We now consider that the vehicle also measures its own velocity with an odometer. These measurements are provided by

`tachy.v`	Observation done by the odometer
`tachy.sv`	Standard deviation associated with `tachy.s`

1. Derive a new observation model $\boldsymbol{C}$ that incorporates both GNSS and odometer observations at once.
2. Implement this new observation model and find suitable values of $\boldsymbol{Q}$ and $\boldsymbol{R}$[2].
3. Show the improvement of this estimation over the previous.

[1] `gps.sx` can be used
[2] `tachy.sv` can be used

4 Sequential data fusion

When measurements are not correlated with one another, the state can be updated sequentially, which is lighter to compute and easier to manage with asynchronous sensors.

1. Adapt your Kalman Filter implementation to update with the GNSS measurement first then with the odometer.

2. Compare this estimation with the previous.

5 Tightly-coupled GNSS with Odometer

We now want to integrate the GNSS measurements more tightly. For this, we now consider that the GNSS receiver provides the position (x_i, y_i), pseudo-ranges[3] PR_i and associated uncertainty for every satellite i. However, when working with such signals, we know that several relativist effects impact the measured pseudo-ranges. For simplification we will consider that all satellite are distorted by a constant distance offset due to this.

1. What is the expected distance between the vehicle and a satellite i?

2. Propose a new state, evolution and observation model to incorporate raw signals while accounting for the offset.

3. $\boldsymbol{g}$ is non-linear, meaning that an EKF will have to be implemented. First, calculate the Jacobian $\boldsymbol{C}$ of $\boldsymbol{g}$, defined as

$$\boldsymbol{C}_i = \frac{\partial \boldsymbol{g}_i}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial \boldsymbol{g}_i}{\partial x} & \frac{\partial \boldsymbol{g}_i}{\partial v} & \dots \end{bmatrix}$$

4. Implement an EKF using this new modeling

$$\begin{aligned}
\boldsymbol{x}_{k+1|k} &= \boldsymbol{f}(\boldsymbol{x}_{k|k}) \\
\boldsymbol{P}_{k+1|k} &= \boldsymbol{A}\boldsymbol{P}_{k|k}\boldsymbol{A}^T + \boldsymbol{Q} \\
\boldsymbol{K} &= \boldsymbol{P}_{k+1|k}\boldsymbol{C}^T \left(\boldsymbol{C}\boldsymbol{P}_{k+1|k}\boldsymbol{C} + \boldsymbol{R}\right)^{-1} \\
\boldsymbol{\epsilon} &= \boldsymbol{y} - \boldsymbol{g}(\boldsymbol{x}_{k+1|k}) \\
\boldsymbol{x}_{k|k} &= \boldsymbol{x}_{k+1|k} + \boldsymbol{K}\boldsymbol{\epsilon} \\
\boldsymbol{P}_{k|k} &= (\boldsymbol{I} - \boldsymbol{K}\boldsymbol{C})\,\boldsymbol{P}_{k+1|k}\,(\boldsymbol{I} - \boldsymbol{K}\boldsymbol{C})^T + \boldsymbol{K}\boldsymbol{R}\boldsymbol{K}^T
\end{aligned}$$

   and provided data in the form:

`c*dt`	True offset (speed of light times clock offset, to calculate errors)
`gps.s1.x`	Abscissa of satellite 1
`gps.s1.y`	Ordinate of satellite 1
`gps.s1.PR`	Pseudo-Range measurement for satellite 1
`gps.s1.sPR`	Standard deviation associated to `gps.s1.PR`

5. Find suitable noises and compare with previous estimations. How do you explain the differences?

6. Does-it works with only two satellites? With only one? Study the state observability to understand why.

[3] Distance measurement derived from the time between emission and reception of a GNSS signal.

6 Handling outliers measurements

Pseudo-ranges are sensitive to multi-paths which can generates "outliers". It is common to exclude measurements that are too far from the estimated by considering that the state is sufficiently good and it must be the measurement that is wrong.

1. Use the Mahalanobis test to reject bad GNSS pseudo-ranges which are not coherent with the predicted measurements.

2. Use the file `simulated_data_with_outliers.mat` in which only satellite 1 is affected by multi-paths and compare the estimations with and without error detection.

7 Other filters

1. Implement an UKF to filter this system and compare with previous estimations.

2. Implement an CIF to filter this system and compare with previous estimations.

What are your conclusions on the different filters? Which one yields the best results or is the simplest to tune?