Samy Labsir

Estimation and identification

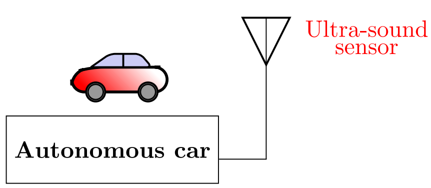
Au-512, 5th year SET, IPSA

Practical works 3: Autonomous car tracking with (Iterated)-Extended Kalman filter ($\sim$ 4H)

Tracking a autonomous car is a fundamental task and challenge in several applications as well as in urban environments as in aeronautic for plane inspection. Indeed, classically, it is equipped with several quantities of sensor (radar, lidar, ultra sound) which we need to track its position in its environment at the same time that the car moves. To resolve the tracking problem, one of the solutions is to place ourselves in a statistical context (to take into account the uncertainties on the measurements) by using the Bayesian filter and more particularly its analytical form given by Kalman filter and its variants.
The aim of this tutorial is to familiarize with conventional tracking algorithms to resolve this problem by implementing two extensions of the Kalman filter (KF): the extended KF and the iterated extended KF.
The implementation will be done in MATLAB. Questions relatives to theoretical aspects are indicated with ***Th.****, those relatives to numerical aspects are indicated with* ***Num.***

Part I: Starting up

We assume the existence of a robot moving in a indoor 2D environment and embedded with an ultra-sound sensor. At each discrete instant k, its dynamic is characterized by its position $\mathbf{p}_k$ and its orientation θ_k.



It performs a task in such a way that it is forced to make a turn regularly. Consequently, the discrete evolution model on its position $\mathbf{p}_k$ is driven by the following discrete equation:

$$\mathbf{p}_k = \mathbf{p}_{k-1} + \mathbf{R}(\theta_{k-1})\,\mathbf{v}\,\delta_t + \mathbf{n}_p \quad \mathbf{n}_p \sim \mathcal{N}(0, \sigma_p^2\,\mathbf{I}_{2\times 2}) \tag{1}$$

where $\mathbf{R}(\theta_k)$ is the rotation matrix associated to the angle θ_k, $\mathbf{v}$ is the robot velocity assumed constant. As far as concerns θ_k , it is controlled by the following linear model:

$$\theta_k = \theta_{k-1} + w\,\delta_t + n_\theta \quad n_\theta \sim \mathcal{N}(0, \sigma_\theta^2) \tag{2}$$

where w is the robot angular velocity assumed constant.

1) (**Th.**) Give the expression of $\mathbf{R}(\theta_k)$ and explain why the model (1) allows to take into account a turn in the trajectory.

2) (**Num.**) Implement this equation for different instant $k \in \{1, \dots, K\}$. We can set:

• $K = 100$	• $w = 0.01$ rad s^{-1}
• $\delta_t = 1$	• $\mathbf{v} = [1, 1]$ m s^{-1}
• $\mathbf{p}_0 = [0, 0]$ m	• $\sigma_p = 0.01$ m
• $\theta_0 = 0.1$ rad	• $\sigma_\theta = 10^{-4}$ rad

Comment an interpret.

Furthermore, we assume that the embedded ultra sound sensor allows to measure, at each instant k, $d_{i,k}$ the distance between itself and different points of interest. Let's assume the existence of N points of interest in the environment, we have:

$$d_{i,k} = \|\mathbf{z}_{i,k}\| \tag{3}$$

where $\mathbf{z}_{i,k}$ is the coordinate of i^{th} point of interest expressed in the robot local frame. It is linked to ***the robot position in the global frame $\mathbf{p}_k$ by****:*

$$\mathbf{z}_{i,k} = \mathbf{R}(\theta_k)\mathbf{p}_i + \mathbf{p}_k \quad \forall i \in \{1, \dots, N\} \tag{4}$$

and we measure:

$$d_{i,k} = \|\mathbf{R}(\theta_k)\mathbf{p}_i + \mathbf{p}_k\| + n_{i,k} \quad n_{i,k} \sim \mathcal{N}(0, \sigma_d^2) \tag{5}$$

We can refer to the figure 1 to illustrate the problem.

3) (**Th.**) To your opinion, how many ultra sound sensor measurements we need to invert the model (5) ?

4) (**Num.**) Implement the equation (5), at each instant k, by taking advantage of the evolution model coded in the question (2). We can set $N = 3$, $\mathbf{p}_1 = [40, 10]$, $\mathbf{p}_2 = [80, 25]$, $\mathbf{p}_3 = [120, 30]$, and $\sigma_d = 0.1$ m

4) (**Num.**) Draw the trajectory previously obtained superimposed to the set $\{\mathbf{p}_i\}_{i=1}^{3}$.

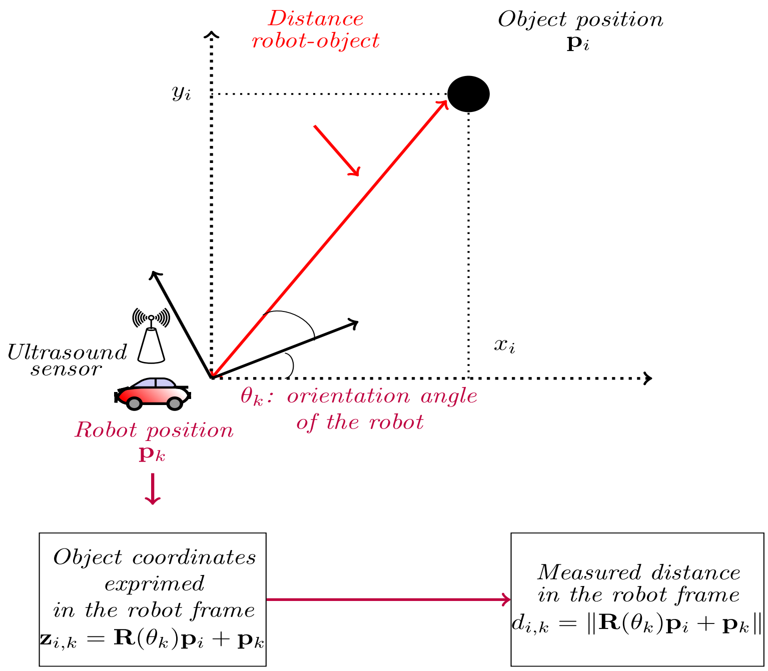


Figure 1: *Illustration of the ultrasound measurements process*

Part II: Implementation of tracking algorithms

In this part, we deal with the implementation of the tracking algorithm, particularly Extended Kalman Filter and Iterated Extended Kalman filter. We propose to handle the problem by considering two cases, the first, assumed the orientation angle is known, and the second one, unknown. Then, the considered models in the part I, and specifically the state vector are different according to the two cases. ***In the following, we keep the same simulation parameters as previously.***

Case I: θ_k known

In this case, the considered state vector is $\mathbf{x}_k = \mathbf{p}_k$.

1) (**Th.**) Write the models under the form:

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}) + \mathbf{v}_k \quad \mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k) \tag{6}$$
$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{n}_k \quad \mathbf{n}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k) \tag{7}$$

where $\mathbf{f}$ and $\mathbf{h}$ are to identify as well as $\mathbf{Q}_k$ and $\mathbf{R}_k$.

2) (**Th.**) Compute the Jacobian matrix of $\mathbf{f}$ and $\mathbf{h}$ according to $\mathbf{p}_k$. What do you observe ?

EKF implementation

Now, we propose to study and implement the first tracking algorithm: the Extended Kalman filter (EKF). The underlying idea of this algorithm is to find an estimator of the state vector by performing a development at order 1 on the function $\mathbf{f}$ *and* $\mathbf{g}$ *considered for the evolution and observation models.*

- *Prediction step:*

3) (**Th.**) Show that under the following assumptions:

$$\mathbf{x}_k \simeq \mathbf{f}(\widehat{\mathbf{x}}_{k-1|k-1}) + \mathbf{J}_{\mathbf{f}} \left(\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|k-1}\right) \tag{8}$$

$$\widehat{\mathbf{x}}_{k-1|k-1} = \mathbb{E}(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) \tag{9}$$

then, the estimator of $\mathbf{x}_k$, denoted $\widehat{\mathbf{x}}_{k|k-1} \triangleq \mathbb{E}(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ is given by:

$$\widehat{\mathbf{x}}_{k|k-1} = \mathbf{f}(\widehat{\mathbf{x}}_{k-1|k-1}) \tag{10}$$

and the covariance of the estimator $\mathbf{P}_{k|k-1} \triangleq \mathbb{E}(\left(\mathbf{x}_k - \widehat{\mathbf{x}}_{k-1|k-1}\right)\left(\mathbf{x}_k - \widehat{\mathbf{x}}_{k-1|k-1}\right)^\top)$ is given by:

$$\mathbf{P}_{k|k-1} = \mathbf{J}_{\mathbf{f}} \mathbf{P}_{k-1|k-1} \mathbf{J}_{\mathbf{f}}^\top + \mathbf{Q}_k \tag{11}$$

4) (**Th.**) We observe that the prediction step in this scenario is the same as the **Kalman filter**, why ?

5) (**Num.**) Implement the prediction step of the EKF filter in a function *EKFprediction-step.m*

- *Correction step:*

6) (**Th.-Num.**) Write the equations of the EKF correction step and implement them in a function *EKFcorrectionstep.m*.

- *Algorithm test:*

7) (**Num.**) Test the algorithm by plotting the instantaneous error estimation $(\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k})_l$ on each component $l \in \{1, 2\}$, and then by drawing the Root Mean-Square Error (RMSE) approximated by Monte-Carlo by:

$$\frac{1}{N_r} \sum_{nr=1}^{N_r} \|\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k}^{(nr)}\|^2 \tag{12}$$

where N_r is a number of realizations set to $N_r = 100$.

IEKF implementation

In this subsection, we propose to study and implement the second tracking algorithm: the Iterated Extended Kalman filter (IEKF). Contrary to the EKF, the non-linearities of the observation model is preserved for the resolution of the estimator. As the prediction step is similar to the EKF, we directly deal with the correction step.

The correction step of the IEKF consists to find the the estimated state vector $\widehat{\mathbf{x}}_{k|k}$ verifying:

$$\widehat{\mathbf{x}}_{k|k} = \underset{\mathbf{x}_k}{\operatorname{argmax}}\ p(\mathbf{x}_k|\mathbf{z}_{1:k}) \tag{13}$$

8) (**Th.**) Show that this problem is equivalent to resolve the following optimization problem:

$$\widehat{\mathbf{x}}_{k|k} = \underset{\mathbf{x}_k}{\operatorname{argmin}}\ \|\boldsymbol{\phi}(\mathbf{x}_k)\|^2_{\boldsymbol{\Sigma}_\phi} \tag{14}$$

where:

- $\boldsymbol{\phi}$ is a smooth function depending on $\mathbf{x}_k$, $\widehat{\mathbf{x}}_{k|k-1}$ and $\mathbf{z}_{1:k}$
- $\boldsymbol{\Sigma}_{\boldsymbol{\phi}}$ to identify.

The covariance matrix of $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ *is determined by fitting a Gaussian distribution on* $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ *with mean* $\widehat{\mathbf{x}}_{k|k}$ *such as:*

$$\boldsymbol{\phi}(\mathbf{x}_k) \simeq \boldsymbol{\phi}(\widehat{\mathbf{x}}_{k|k}) + \mathbf{J}_{\boldsymbol{\phi}}\left(\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k}\right) \tag{15}$$

with $\mathbf{J}_{\boldsymbol{\phi}}$: Jacobian matrix of $\boldsymbol{\phi}$ computed at $\widehat{\mathbf{x}}_{k|k}$.

9) (**Th.**) By identifying the relation between $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ and $\|\boldsymbol{\phi}(\mathbf{x}_k)\|^2_{\boldsymbol{\Sigma}_\phi}$, show that this approximation allows to find:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) \simeq\propto \exp\left(-0.5\,\|\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k}\|_{\boldsymbol{\Sigma}_{k|k}}\right) \tag{16}$$

where $\boldsymbol{\Sigma}_{k|k}$ is to identify in function of $\mathbf{J}_{\boldsymbol{\phi}}$

10) (**Th.**) Write the equations of the correction step of the IEKF.

11) (**Num.**) Implement the IEKF correction in a function *IEKFcorrectionstep.m*, then test the algorithm.

12) (**Num.**) In the same way as the EKF filter, draw the instantaneous error estimation and the MSE at each instant k. Interpret and comment by comparison with the results obtained with the EKF.

Case II: θ_k unknown (Optional)

In this section, we assume that θ_k is integrated in the unknown parameters so that $\mathbf{x}_k = \left[\mathbf{p}_k^\top, \theta_k\right]^\top$. Then, the challenge here is to deal with the rotation matrix $\mathbf{R}(\theta_k)$ and its Jacobian matrix.

13) (**Th.**) Write the new function $\mathbf{f}^{(a)}$ and $\mathbf{h}^{(a)}$ such that:

$$\mathbf{x}_k = \mathbf{f}^{(a)}(\mathbf{x}_{k-1}) + \mathbf{v}^{(a)} \qquad \mathbf{v}^{(a)} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k^{(a)}) \tag{17}$$

$$\mathbf{z}_k = \mathbf{h}^{(a)}(\mathbf{x}_k) + \mathbf{n}^{(a)} \qquad \mathbf{n}^{(a)} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k^{(a)}) \tag{18}$$

14) (**Th.**) Determine the new Jacobian matrices of $\mathbf{f}^{(a)}$ and $\mathbf{h}^{(a)}$ and the corresponding covariance matrices $\mathbf{Q}_k^{(a)}$ and $\mathbf{R}_k^{(a)}$.

15) (**Num.**) Use the results of question (13)-(14) to:

- $\hookrightarrow$ implement the new EKF filter,
- $\hookrightarrow$ implement the new IEKF filter,
- $\hookrightarrow$ compare the two filters in terms of MSE and comment the results compared to those previously obtained with θ_k known.