

# CR Inférence Bayésienne 1 : Filtre de Kalman

Paul-Emmanuel Pujas, Gatien Roujanski

December 2023

## Git et overleaf

Liens

Git :  
<https://github.com/FauconK/Inference>

Overleaf :  
<https://www.overleaf.com/read/rmqhjcddvxxq87f079>

## Introduction

L'objectif de ce TP est de suivre une cible mobile (Fusée, Oiseaux, ...) à partir uniquement de mesures bruitées.

Ainsi nous avons deux équations qui modélise la position de notre cible (équation d'état) et nos observations (équation d'observation).

Equation d'état :

$$X_k = \underbrace{\begin{pmatrix} 1 & T_e & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T_e \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\mathbf{F}_k} X_{k-1} + U_k$$

Avec  $X_0 \sim \mathcal{N}(\mathbf{m}_{0|0}, \mathbf{P}_{0|0})$ ,

Equation d'observation :

$$Y_k = \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{\mathbf{H}_k} X_k + V_k$$

Avec  $U_k$  et  $V_k$  du bruit gaussien :  $U_k \sim \mathcal{N}(\mathbf{0}_{4 \times 1}, \mathbf{Q})$  et  $V_k \sim \mathcal{N}(\mathbf{0}_{2 \times 1}, \mathbf{R})$ . De plus on suppose que la suite  $[U_k, V_k]^T$  est indépendante, indépendante de  $X_0$  et que  $U_k$  et  $V_i$  sont indépendants quelques soient  $i$  et  $k$ .

On considère ce problème comme un suivi de cible classique donc on choisit Q et R de la forme suivante :

$$\mathbf{Q} = \sigma_Q^2 \begin{pmatrix} \frac{T_e^3}{3} & \frac{T_e^2}{2} & 0 & 0 \\ \frac{T_e^2}{2} & T_e & 0 & 0 \\ 0 & 0 & \frac{T_e^3}{3} & \frac{T_e^2}{2} \\ 0 & 0 & \frac{T_e^2}{2} & T_e \end{pmatrix} \text{ et } \mathbf{R} = \begin{pmatrix} \sigma_{p_x}^2 & 0 \\ 0 & \sigma_{p_y}^2 \end{pmatrix}$$

On se propose d'appliquer un filtre de Kalman afin d'obtenir une trajectoire non bruitée à partir des observations  $Y_k$

Dans un premier temps nous nous proposons de démontrer une partie du filtre de Kalman : le passage de  $\hat{x}_{k|k}$  et de  $P_{k|k}$  à  $\hat{x}_{k+1|k+1}$  et de  $P_{k+1|k+1}$

On sait que  $\hat{x}_{k|k} = m_{k|k} = E[X_k|y_{0:k}]$  et on a  $P_{k|k} = \text{Covar}[X_k|y_{0:k}]$

## 1 Le filtre de Kalman

### 1.1 Solution

Le filtre de Kalman donne une solution récursive de  $\hat{x}_{n|n}$  grâce aux hypothèses suivantes :

u et v sont des processus centrés, décorrélés, mutuellement décorrélés et décorrélés de  $x_0$

On obtient la solution suivante :

- Initialisation :  $\hat{\mathbf{x}}_{0|-1} = \bar{\mathbf{x}}_0, \mathbf{P}_{0|-1} = \mathbf{P}_0$

- Innovation de  $\tilde{\mathbf{y}}_n$  de  $\mathbf{y}_n$  :

$$\begin{aligned} \tilde{\mathbf{y}}_n &= \mathbf{y}_n - \mathbf{H}_n \hat{\mathbf{x}}_{n|n-1} \\ \mathbf{L}_n &= \mathbf{H}_n \mathbf{P}_{n|n-1} \mathbf{H}_n^T + \mathbf{R}_n \end{aligned}$$

- Partie filtrage / mise à jour :

$$\begin{aligned} \hat{\mathbf{x}}_{n|n} &= \hat{\mathbf{x}}_{n|n-1} + \mathbf{K}_{n|n} \tilde{\mathbf{y}}_n \\ \mathbf{K}_{n|n} &= \mathbf{P}_{n|n-1} \mathbf{H}_n^T \mathbf{L}_n^{-1} \\ \mathbf{P}_{n|n} &= \mathbf{P}_{n|n-1} - \mathbf{K}_{n|n} \mathbf{L}_n \mathbf{K}_{n|n}^T. \end{aligned}$$

- Partie prédiction / propagation :

$$\begin{aligned} \hat{\mathbf{x}}_{n+1|n} &= \underbrace{\mathbf{F}_n \hat{\mathbf{x}}_{n|n-1}}_{\hat{\mathbf{x}}_{n+1|n-1}} + \mathbf{K}_{n+1|n} \tilde{\mathbf{y}}_n \\ \mathbf{K}_{n+1|n} &= (\mathbf{F}_n \mathbf{P}_{n|n-1} \mathbf{H}_n^T + \mathbf{G}_n \mathbf{S}_n) \mathbf{L}_n^{-1} \\ \mathbf{P}_{n+1|n} &= \underbrace{\mathbf{F}_n \mathbf{P}_{n|n-1} \mathbf{F}_n^T + \mathbf{G}_n \mathbf{Q}_n \mathbf{G}_n^T}_{\mathbf{P}_{n+1|n-1}} - \mathbf{K}_{n+1|n} \mathbf{L}_n \mathbf{K}_{n+1|n}^T \end{aligned}$$

## 1.2 Démonstration

La démonstration est tirée du polycopié de cours et du TD.

Quelques notations pour les calculs :

$$\begin{aligned}\tilde{x}_{n|n} &= x_n - \hat{x}_{n|n} = x_n - P_{H_n(y)}(x_n) \\ P_{n|n} &= (\tilde{x}_{n|n}, \tilde{x}_{n|n}) \\ \tilde{x}_{n|n-} &= x_n - \hat{x}_{n|n-} = x_n - P_{H_{n-1}(y)}(x_n) \\ P_{n|n-1} &= (\tilde{x}_{n|n-1}, \tilde{x}_{n|n-1}) \\ \tilde{y} &= y_n - \hat{y}_{n|n} = y_n - P_{H_n(y)}(y_n) \\ L_n &= (\tilde{y}_n, \tilde{y}_n)\end{aligned}$$

Tout d'abord on peut déduire quatre relations d'orthogonalités grace aux notations et au modèle :

$$\begin{aligned}u_n &\perp H_n(x) \\ u_n &\perp H_{n-1}(y) \\ v_n &\perp H_n(x) \\ v_n &\perp H_{n-1}(y)\end{aligned}$$

Pour les obtenir on développe les équations d'évolutions et d'observations et on remarque que les  $x_k$  et  $y_k$  sont des CL de  $x_0$  et les  $u_0...u_{k-1}$  /  $v_0...v_{k-1}$ . Donc ne dépend pas de  $u_i$  /  $v_i$  où  $i \geq k$ .

Ensuite on va procéder en deux temps : filtrer puis prédire.

### 1.2.1 Filtrage

Le but est d'obtenir :

$$\begin{aligned}P_{n|n} &= P_{n|n-1} - K_{n|n}L_nK_{n|n}^T \\ \hat{x}_{n|n} &= \hat{x}_{n|n-1} + K_{n|n}\tilde{y}_n \\ K_{n|n} &= P_{n|n-1}H_n^T L_n^{-1}\end{aligned}$$

Pour cela on commence par projeter  $x_n$  sur  $H_n(y)$  :

$$\hat{x}_{n|n} = \hat{x}_{n|n-1} + (\hat{x}_n, \tilde{y}_n)(\tilde{y}_n, \tilde{y}_n)^{-1}\tilde{y}_n$$

Puis on exprime  $(\tilde{y}_n, \tilde{y}_n)$ ,  $(\hat{x}_n, \tilde{y}_n)$  et  $\tilde{y}_n$  grâce aux paramètres du modèle.

Tout d'abord  $\tilde{y}_n$  :

$$\tilde{y} = y_n - \hat{y}_{n|n} = y_n - P_{H_n(y)}(y_n)$$

Donc on peut projeter l'eq d'observation sur  $H_{n-1}(y)$ . On obtient :

$$P_{H_{n-1}(y)}(y_n) = H_n P_{H_{n-1}(y)}(x_n) + P_{H_{n-1}(y)}(v_n)$$

Donc

$$\tilde{y}_n = y_n - H_n \hat{x}_{n|n-1}$$

Ensuite intéressons nous à  $(\tilde{y}_n, \tilde{y}_n)$  :

On injecte l'équation d'observation dans l'équation obtenue juste avant

$$\tilde{y}_n = v_n + H_n \hat{x}_{n|n-1}$$

On calcule ensuite  $(\tilde{y}_n, \tilde{y}_n)$  :

$$(\tilde{y}_n, \tilde{y}_n) = (H_n \hat{x}_{n|n-1}, H_n \hat{x}_{n|n-1}) + (H_n \hat{x}_{n|n-1}, v_n) + (H_n \hat{x}_{n|n-1}, v_n)^T + (v_n, v_n)$$

On demontre que le 2e et le 3e terme sont nuls :  $\hat{x}_{n|n-1} = x_n - \hat{x}_{n|n-1}$  qui appartiennent respectivement à  $H_n(x)$  et  $H_{n-1}(y)$  et on a vu que  $v_n \perp H_n(x)$  et  $v_n \perp H_{n-1}(y)$ . Donc le produit scalaire est nul.

On obtient ainsi

$$L_n = H_n P_{n|n-1} H_n^T + R_n$$

$$\text{avec } L_n = (\tilde{y}_n, \tilde{y}_n), R_n = (v_n, v_n) \text{ et } H_n P_{n|n-1} H_n^T = (H_n \hat{x}_{n|n-1}, H_n \hat{x}_{n|n-1})$$

Dernièrement on calcule  $(\hat{x}_n, \tilde{y}_n)$  grâce aux calculs fait avant :

$$(\hat{x}_n, \tilde{y}_n) = (x_n, v_n + H_n \hat{x}_{n|n-1})$$

$$= (x_n, H_n \hat{x}_{n|n-1})$$

$$= (\hat{x}_{n|n-1} + \tilde{x}_{n|n-1}, H_n \hat{x}_{n|n-1})$$

Or  $\hat{x}_{n|n-1}$  appartient à  $H_{n-1}(y)$  et  $\hat{x}_{n|n-1} \perp H_{n-1}(y)$

Donc

$$(\hat{x}_n, \tilde{y}_n) = (\tilde{x}_{n|n-1}, H_n \hat{x}_{n|n-1})$$

$$= P_{n|n-1} H_n^T$$

Maintenant on calcule  $(\tilde{x}_{n|n}, \tilde{x}_{n|n}) = P_{n|n}$  :

$$P_{n|n} = (\tilde{x}_{n|n-1} - K_{n|n} \tilde{y}_n, \tilde{x}_{n|n-1} - K_{n|n} \tilde{y}_n)$$

$$= (\tilde{x}_{n|n-1}, \tilde{x}_{n|n-1}) - K_{n|n} (\tilde{y}_n, \tilde{x}_{n|n-1}) - (\tilde{x}_{n|n-1}, \tilde{y}_n) K_{n|n}^T + K_{n|n} (\tilde{y}_n, \tilde{y}_n) K_{n|n}^T$$

Mais  $\tilde{x}_{n|n-1} = -\hat{x}_{n|n-1} + x_n$  et  $\hat{x}_{n|n-1}$  appartient à  $H_{n-1}(y)$ . Donc  $(\tilde{y}_n, \tilde{x}_{n|n-1}) = (x_n, \tilde{y}_n) = K_{n|n} L_n$

On obtient bien :

$$P_{n|n} = P_{n|n-1} - K_{n|n} L_n K_{n|n}^T$$

$$\hat{x}_{n|n} = \hat{x}_{n|n-1} + K_{n|n} \tilde{y}_n$$

$$K_{n|n} = P_{n|n-1} H_n^T L_n^{-1}$$

## 1.2.2 Prediction

Grâce à la partie Filtrage on peut exprimer  $\hat{x}_{n|n}$  et  $P_{n|n}$  en fonction de leur antécédent  $\hat{x}_{n|n-1}$ ,  $P_{n|n-1}$ ,  $y_n$  et les données

Maintenant on veut calculer  $\hat{x}_{n+1|n}$  et  $P_{n+1|n}$

On commence par projeter  $x_{n+1}$  sur  $H_n(y)$  :

$$\hat{x}_{n+1|n} = \hat{x}_{n+1|n-1} + (x_{n+1}, \tilde{y}_n)(\tilde{y}_n, \tilde{y}_n)^{-1} \tilde{y}_n$$

$$\text{Et } (x_{n+1}, \tilde{y}_n)(\tilde{y}_n, \tilde{y}_n) = K_{n+1|n}$$

Grâce à  $u_n \perp H_{n-1}(y)$  on décompose le premier terme :

$$\hat{x}_{n+1|n-1} = P_{H_{n-1}(y)}(x_{n+1}) = F_n P_{H_{n-1}(y)}(x_n) + G_n P_{H_{n-1}(y)}(u_n)$$

$$\text{Et } P_{H_{n-1}(y)}(u_n) = 0$$

Grâce au résultat  $\tilde{y}_n = v_n + H_n \hat{x}_{n|n-1}$  et à l'équation d'évolution on peut obtenir :

$$x_{n+1}, \tilde{y}_n) = (F_n X_n + G_n u_n, H_n \tilde{x}_{n|n-1} + v_n)$$

$$= (F_n x_n, H_n \tilde{x}_{n|n-1}) + (F_n x_n, v_n) + (G_n u_n, H_n \tilde{x}_{n|n-1}) + (G_n u_n, v_n)$$

$$\begin{aligned}\hat{x}_{n+1|n} &= F_n \hat{x}_{n|n-1} + K_{n+1|n} \hat{y}_n \\ K_{n+1|n} &= (F_n P_{n|n-1} H^T + G_n S_n) L_n^{-1}\end{aligned}$$
$$P_{n+1|n} = F_n P_{n|n-1} F_n^T + G_n Q_n G_n^T - K_{n+1|n} L_n K_{n+1|n}^T$$
$$P_{n+1|n} = P_{n+1|n-1} - K_{n+1|n} L_n K_{n+1|n}^T$$

Regardons maintenant faire plusieurs simulations avec plusieurs valeurs de  $\sigma_Q$ ,  $\sigma_{px}$  et  $\sigma_{py}$

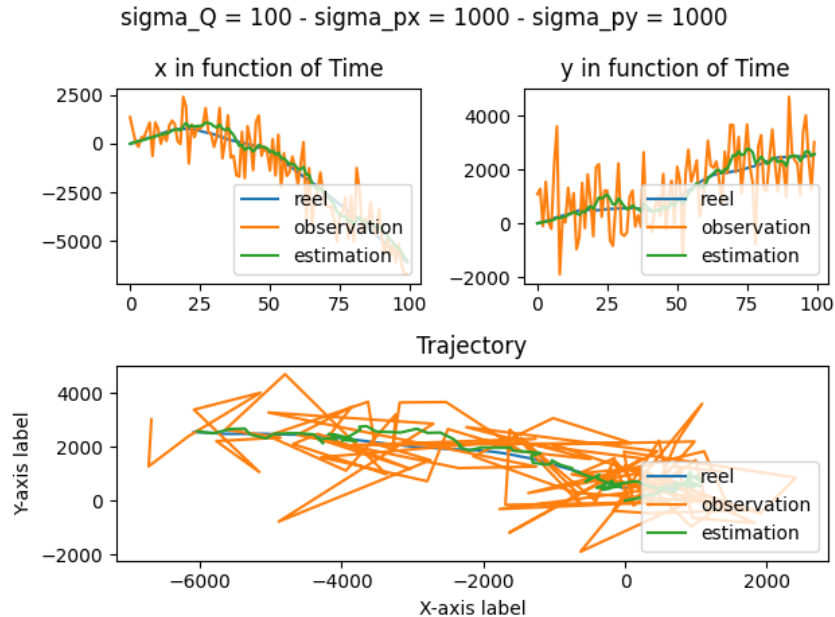
$\sigma_Q / \sigma_{px} \text{ et } \sigma_{py}$	10	30	100	1000
1	6.74	15.55	43	98
10	10.88	23.95	65	237
100	15.64	35.26	79	306
1000	40.16	48.92	109	618

En augmentant le bruit de  $\sigma_Q$  on remarque une augmentation de l'erreur moyenne. Ce résultat est logique : une augmentation de la variance de la position augmente une plus grande variation des positions - c'est à dire plus éloignée de la trajectoire rectiligne classique.

L'augmentation de  $\sigma_{px}$  et de  $\sigma_{py}$  entraîne également une augmentation de l'erreur moyenne. Cette fois c'est dû à l'augmentation du bruit sur les observations.

Dans les deux cas on remarque que l'augmentation de l'erreur moyenne est globalement logarithmique : chaque augmentation d'un facteur 10 d'un sigma augmente l'erreur moyenne de la même quantité.

Le filtre donne des résultats impressionnants : malgré un fort sigma il arrive à reconstituer la trajectoire assez fidèlement : exemple avec  $\sigma_Q = 100$  et  $\sigma_{px} = \sigma_{py} = 1000$



### 3 Poursuite angle distance

Dans cette partie nous n'avons plus les positions en coordonnées cartésiennes mais l'angle et la distance à la cible.

Pour obtenir les coordonnées cartésienne il suffit d'appliquer quelques formules de trigonométrie basique :

$$r = \sqrt{px^2 + py^2}$$

$$\theta = \arctan\left(\frac{py}{px}\right)$$

#### 3.1 Adaptation du nouveau système au filtrage de Kalman

La nouvelle relation entre les observations et la réalité de la forme  $y_n = h(x_n) + v_n$  où  $h$  n'est pas linéaire. Ce nouveau système ne répondant plus aux hypothèses de départ on ne peut plus appliquer le filtre de Kálmán directement. L'idée va être de linéariser cette relation. Pour cela nous allons tout d'abord faire un développement limité à l'ordre 1. Cela pose alors la question du point utilisé pour faire ce développement. En effet il doit être au plus proche de la valeur réelle pour minimiser l'erreur due au développement. On va alors choisir l'estimateur juste avant de prendre en compte la nouvelle observation i.e. :  $\hat{x}_{n+1|n}$

On fait donc l'approximation :

$$y_n = h(\hat{x}_{n|n-1}) + \Delta h(\hat{x}_{n|n-1})(x_n - \hat{x}_{n|n-1}) + v_n$$

On ne peut toujours pas utiliser Kalman car l'expression doit être de la forme :

$$y_n = F_n x_n + v_n$$

On pose alors la nouvelle observation factice :

$$z_n = y_n - h(\hat{x}_{n|n-1}) + \Delta h(\hat{x}_{n|n-1})\hat{x}_{n|n-1} \text{ on obtient:}$$

$$z_n = \Delta h(\hat{x}_{n|n-1})x_n + v_t$$

Finalement on applique Kalman sur l'observation factice comme vu précédemment.

$$\text{NB : On a } \Delta h(\hat{x}_{n|n-1}) = \begin{pmatrix} \frac{px}{\sqrt{px^2+py^2}} & 0 & \frac{py}{\sqrt{px^2+py^2}} & 0 \\ \frac{-1}{px^2+py^2} & 0 & \frac{px}{px^2+py^2} & 0 \end{pmatrix}$$

où  $\hat{x}_{n|n-1}$  est de la forme  $\begin{pmatrix} px & vx & py & vy \end{pmatrix}$

NB2 : Cette technique s'appelle filtre de Kalman étendu car grâce à la linéarisation et au changement de variable on peut l'appliquer à beaucoup plus de cas.

NB3 : On remarque que le choix du point de linéarisation est très important et que si la fonction n'est pas "douce" cette linéarisation peut mener à une instabilité du filtre.

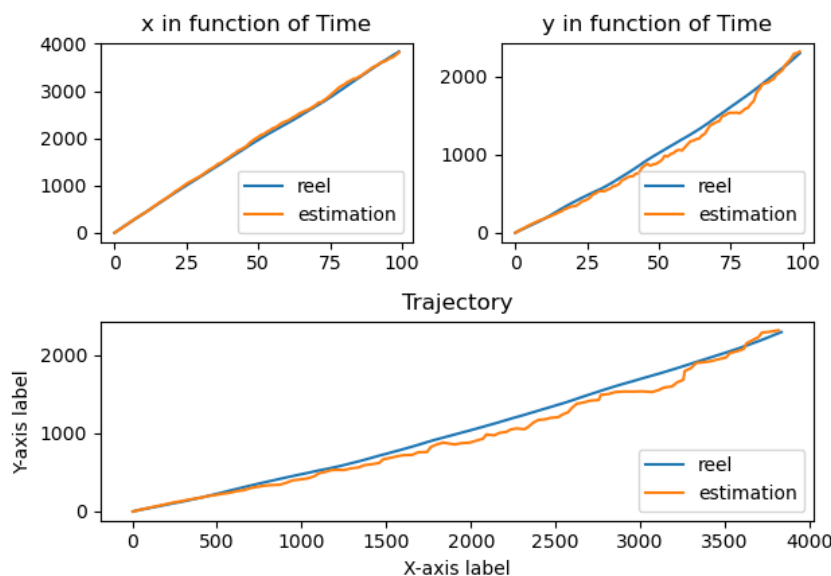
Les erreurs s'accumulant le filtre peuvent "dériver". Une solution à ce problème peut être d'estimer ces erreurs et d'en prendre en compte lors des prédictions futures.

### 3.2 Analyse des résultats

On pose  $\sigma_{angle} = \frac{\pi}{180}$  et  $\sigma_{dist} = 10$  avec  $R = \begin{pmatrix} \sigma_{dist}^2 & 0 \\ 0 & \sigma_{angle}^2 \end{pmatrix}$

Comme précédemment le programme va commencer par générer la trajectoire en coordonnées cartésiennes. Ensuite il génère des observations bruitées qui sont en coordonnées polaire cette fois-ci. Enfin il applique le filtre de Kalman étendu décrit précédemment.

Voici les résultats obtenus :

$$\sigma_Q = 1 - \sigma_{\text{dist}} = 10 - \sigma_{\text{angle}} = \pi/180$$


L'erreur quadratique moyenne est : 51.78.

Regardons maintenant faire plusieurs simulations avec plusieurs valeurs de  $\sigma_Q$ ,  $\sigma_{dist}$  et  $\sigma_{angle}$ . Pour simplifier les choses nous allons multiplier  $\sigma_{dist}$  et  $\sigma_{angle}$  par un même facteur par rapport à leur valeur par défaut.

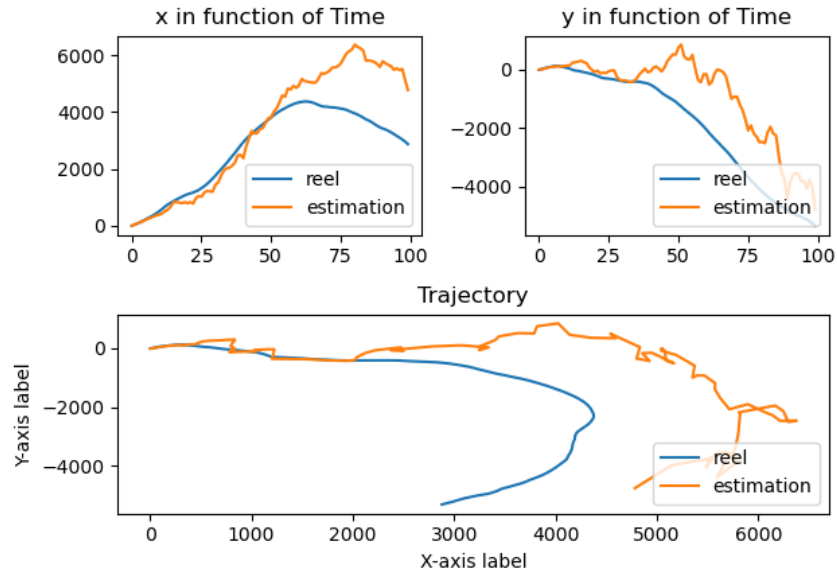
$\sigma_Q$ / facteur <sub>mul</sub> de $\sigma_{dist}$ et $\sigma_{angle}$	1	10	100	1000
1	51.78	94.01	178.53	398.56
10	100.39	348.35	646.39	1116.21
100	188.05	471.59	5598.60	1970.17
1000	575.01	709.89	5213.73	16745.62

On observe les mêmes tendances que dans la partie 2. Cependant le filtre semble beaucoup plus sensible au bruit que précédemment. Cela peut être dû au fait que les erreurs de prédiction dues à la linéarisation du système ne sont pas



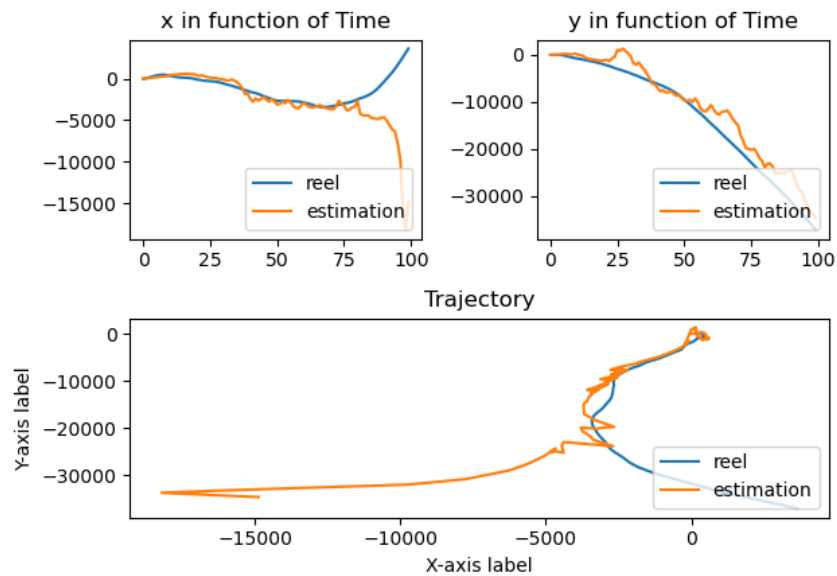
prises en compte et tendent à s'accumuler si les observations sont trop bruitées. Cela permet au filtre de dériver comme on peut le voir dans ces exemples.

$\sigma_Q = 100$  -  $\sigma_{\text{dist}} = 1000$  -  $\sigma_{\text{angle}} = 100 \cdot \pi / 180$



Ici on peut observer une légère dérive mais la trajectoire prédite suit encore "globalement" la trajectoire réelle.

$\sigma_Q = 1000$  -  $\sigma_{\text{dist}} = 10000$  -  $\sigma_{\text{angle}} = 1000 \cdot \pi / 180$



Cela montre l'importance de réduire au maximum le bruit.

L'adaptation du programme réalisé est assez similaire pour ces deux nouvelles fonctionnalités. En effet dans le cas de l'absence de mesure il suffit de renvoyer la prédiction comme la valeur filtrée ie  $\hat{x}_{n+1|n}$  au lieu de  $\hat{x}_{n+1|n+1}$

10