

TP2: Filtrage de Kalman

Navigation pour les systèmes autonomes

ROB312

Yu WANG

October 7, 2021

1 Introduction

Le filtre de Kalman est un filtre à réponse impulsionnelle infinie qui estime les états d'un système dynamique à partir d'une série de mesures incomplètes ou bruitées. Le filtre a été nommé d'après le mathématicien et informaticien américain d'origine hongroise Rudolf Kalman.

Il fait appel à la dynamique de la cible qui définit son évolution dans le temps pour obtenir de meilleures données, éliminant ainsi l'effet du bruit. Ces données peuvent être calculées pour l'instant présent (filtrage), dans le passé (lissage), ou sur un horizon futur (prédiction).

2 Partie 1 : Cas Linéaire

2.1 Q1 - La structure du code et les paramètres

La logique de cette partie du code est la suivante :

- X représente le vecteur d'état du système et on l'initialise d'abord.
 - L'état $X = [x, y, z, v_x, v_y, v_z]$
 - **X_reel** : l'état vrai
 - **X_hat** : la valeur estimée du filtre de Kalman
- On simule la trajectoire physique et le capteur séparément au temps T .
 - **Xv_sim** représente tous les états (x, y, z, v_x, v_y, v_z)
 - **Y_sim** représente seulement les 3 premiers état (x, y, z)

- Après on fait le boucle de simulation du filtre de Kalman:
 - Initialisation des paramètres
 - * **P_hat** : La matrice de covariance
 - * **F** : La matrice de dynamique
 - * **Qf** : La matrice de covariance du bruit de dynamique
 - * **H** : La matrice d'observation
 - * **R** : La matrice de covariance du bruit de mesure
 - On fais des prédictions dans la boucle selon la formule de prédiction.
 - On fais la correction dans la boucle en fonction de la formule d'observation.
- Enfin, on dessine le résultat dans la figure

2.2 Q2 - Compléter le code et commenter les résultats

Tout d'abord, selon l'équation de prédiction et l'équation de correction dans le cas linéaire (comme Eq. (2.1) et Eq. (2.2)), on complète le code.

L'équation de prédiction:

$$\begin{aligned}\hat{\mathbf{x}}_{k|k-1} &= \mathbf{F}_k \hat{\mathbf{x}}_{k-1} + \mathbf{G}_k \mathbf{u}_k \\ \hat{\mathbf{P}}_{k|k-1} &= \mathbf{F}_k \hat{\mathbf{P}}_{k-1} \mathbf{F}_k^T + \mathbf{Q}_k\end{aligned}\quad (2.1)$$

L'équation de correction:

$$\begin{aligned}\hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}) \\ \hat{\mathbf{P}}_k &= (\mathbf{I}_d - \mathbf{K}_k \mathbf{H}_k) \hat{\mathbf{P}}_{k|k-1} \\ \mathbf{K}_k &= \hat{\mathbf{P}}_{k|k-1} \mathbf{H}_k^T (\mathbf{R}_k + \mathbf{H}_k \hat{\mathbf{P}}_{k|k-1} \mathbf{H}_k^T)^{-1}\end{aligned}\quad (2.2)$$

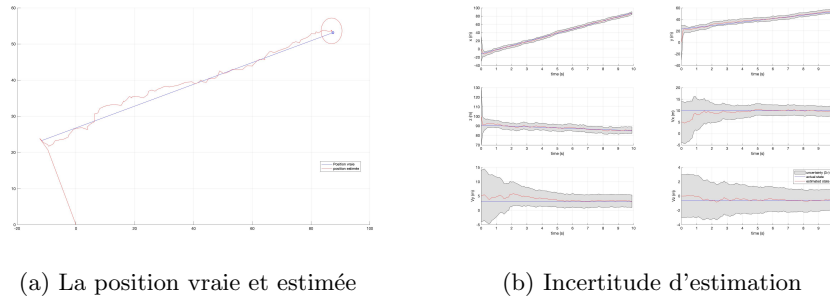


Figure 1: Résultat

Exécutez le code après avoir terminé le code, et le résultat est illustré à la Figure 1. On constate que même s'il y a du bruit (erreur de mesure), la valeur estimée dans chaque direction est toujours cohérente avec la valeur réelle. Le fonctionnement du filtre de Kalman est bien vérifié.

2.3 Q3 - Un trou de mesures entre $t = 3s$ et $t = 7s$

```
1 is_measurementValid = true;
2 if tk*dt ≥ 3 && tk*dt ≤ 7 % t ≥ 3s et t ≤ 7s
3     is_measurementValid = false;
4 end
```

En utilisant le code ci-dessus, on peut simuler un trou de mesures entre $t = 3s$ et $t = 7s$ et le résultat est illustré à la Figure 2.

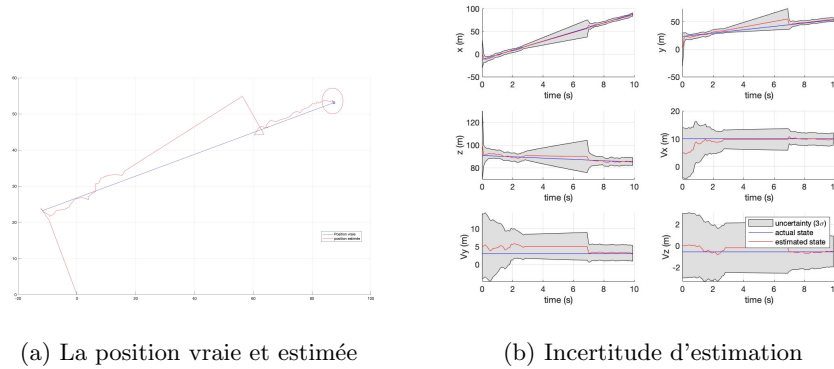
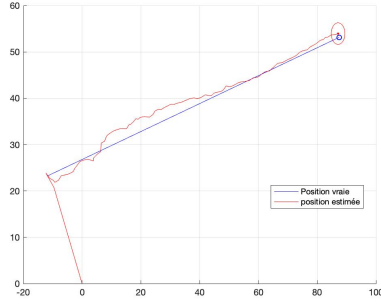


Figure 2: Résultat avec un trou de mesures

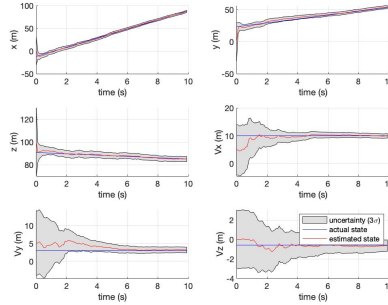
Selon la figure 2, on peut constater que la position réelle et la position estimée ne sont plus cohérentes pendant le trou de mesures et l'incertitude augmente également. Ce phénomène est dû au fait que il n'y a pas de correction entre $t = 3s$ et $t = 7s$. Même s'il y a une grande différence entre la position réelle et la position estimée à $t = 7s$, la correction parvient à rétablir la correspondance. Le fonctionnement de la correction est bien vérifié.

2.4 Q4 - Faire varier le bruit de dynamique du filtre

Dans cette section, on change Qf à $0.1 * Qf$ et $10 * Qf$ et illustre respectivement les résultats dans la Figure 3 et la Figure 4. En le comparant avec la Figure 1, on constate qu'à mesure que le bruit dynamique augmente, l'incertitude augmente également. Mais au final, le filtre de Kalman a tout de même réussi à obtenir la correspondance entre la position réelle et la position estimée.

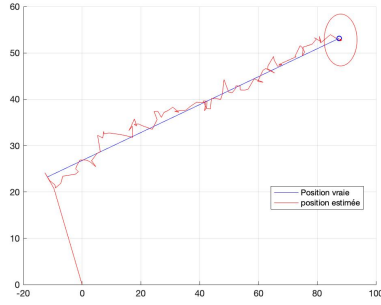


(a) La position vraie et estimée

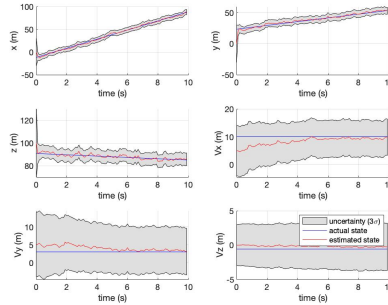


(b) Incertitude d'estimation

Figure 3: Résultat avec $0.1 * Qf$



(a) La position vraie et estimée

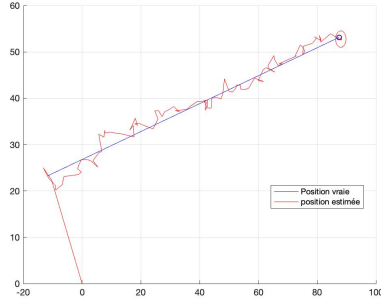


(b) Incertitude d'estimation

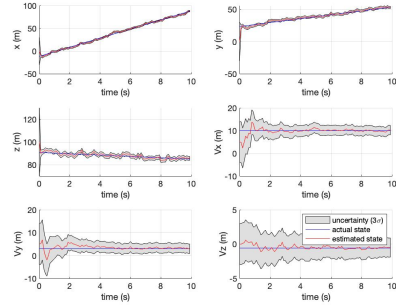
Figure 4: Résultat avec $10 * Qf$

2.5 Q5 - Faire varier le bruit de mesure du filtre

Dans cette section, on change R à $0.1 * R$ et $10 * R$ et illustre respectivement les résultat dans la Figure 5 et la Figure 6. On a simulé l'estimation d'états du système en variant le bruit pour avoir une connaissance claire de l'influence de la qualité de capteur sur la qualité d'estimation. En le comparant avec la Figure 1, on peut facilement déduire que un petit bruit de mesure nous amène des incertitudes petites. Donc un bon capteur est capable d'améliorer la qualité d'estimation.

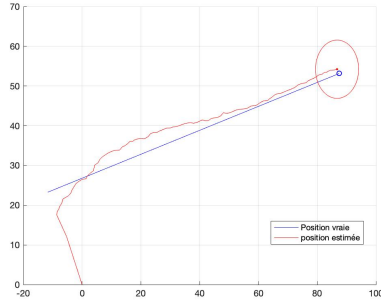


(a) La position vraie et estimée

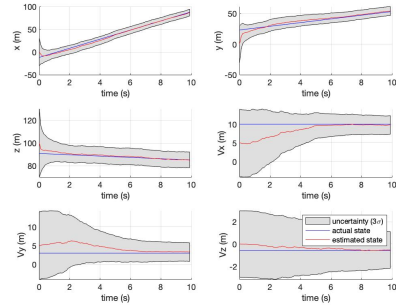


(b) Incertitude d'estimation

Figure 5: Résultat avec $0.1 * R$



(a) La position vraie et estimée



(b) Incertitude d'estimation

Figure 6: Résultat avec $10 * R$

3 Partie 2 : Cas Non-Linéaire

Dans cette partie, on étudiera l'application du filtre de Kalman dans le cas de non-linéaire : **Kalman Étendu**. Ce filtre permet de linéariser localement le problèmes et l'application des équations classiques du filtre de Kalman.

3.1 Q1 - Compléter le code et commenter les résultats

Tout d'abord, selon l'équation de prédiction et l'équation de correction dans le cas non-linéaire (comme Eq. (3.1) et Eq. (3.2)), on complète le code.

L'équation de prédiction:

$$\begin{aligned}\hat{\mathbf{x}}_{k|k-1} &= f(\hat{\mathbf{x}}_{k-1} + \mathbf{u}_k) \\ \hat{\mathbf{P}}_{k|k-1} &= \mathbf{F}_k \hat{\mathbf{P}}_{k-1} \mathbf{F}_k^T + \mathbf{Q}_k \\ \mathbf{F}_k &= \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_k} \in \mathbb{R}^{d \times d}\end{aligned}\tag{3.1}$$

L'équation de correction:

$$\begin{aligned}\hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{y}_k - h(\hat{\mathbf{x}}_{k|k-1})) \\ \hat{\mathbf{P}}_k &= (\mathbf{I}_d - \mathbf{K}_k \mathbf{H}_k) \hat{\mathbf{P}}_{k|k-1} \\ \mathbf{K}_k &= \hat{\mathbf{P}}_{k|k-1} \mathbf{H}_k^T (\mathbf{R}_k + \mathbf{H}_k \hat{\mathbf{P}}_{k|k-1} \mathbf{H}_k^T)^{-1} \\ \mathbf{H}_k &= \left. \frac{\partial h}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_k} \in \mathbb{R}^{d_m \times d}\end{aligned}\tag{3.2}$$

Exécutez le code après avoir terminé le code, et le résultat est illustré à la Figure 7. On constate qu'au début, la position estimée est tellement éloignée de la position réelle que les deux positions mettent plus de temps à s'accorder. Mais cette fois, l'incertitude est beaucoup plus faible que dans le cas linéaire.

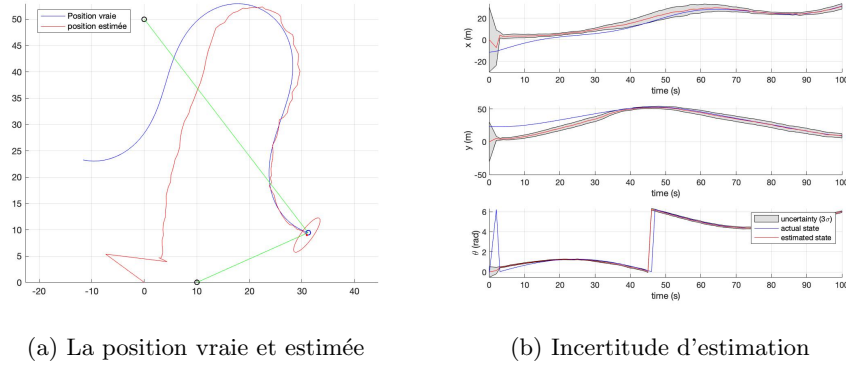


Figure 7: Résultat

3.2 Q2 - Modifier la fréquence de Mesures

On modifie la fréquence de la mesure (à 1 Hz) en utilisant la variable `dt_mesure=1` et les résultats sont présentés à la figure 8. On peut constater qu'au début la courbe de position estimée ne correspond pas à la position réelle dans le temps et est sujette à des changements périodiques, car la correction de la mesure est mise à jour après que 10 prédictions ont déjà été faites. Mais au final, on a une bonne estimation de la position réelle.

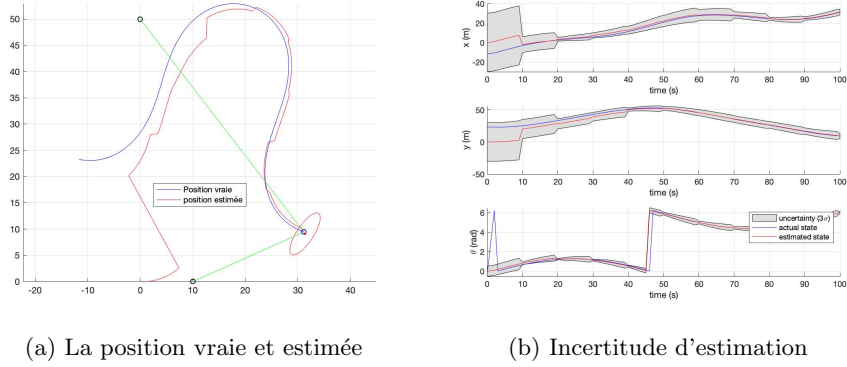


Figure 8: Résultat avec $dt_mesure=1$

3.3 Q3 - Faire varier le bruit de dynamique du filtre

Dans cette section, on change Qf à $0.1 * Qf$ et $10 * Qf$ dans le cas où la fréquence de mesure est 10Hz (par défaut) et illustre respectivement les résultats dans la Figure 9 et la Figure 10. En le comparant avec la Figure 7, on constate que: si le bruit dynamique est plus faible, la courbe de la position estimée devient plus lisse, avec moins d'oscillations et moins d'incertitude, mais l'homogénéité des deux courbes commence plus tard qu'avant. Si le bruit dynamique est plus important, l'incertitude devient plus grande, mais l'homogénéité des deux courbes commence plus tôt qu'avant. En d'autres termes, plus le bruit dynamique est important, plus il est probable que la position estimée soit proche de la position réelle dans un court laps de temps.

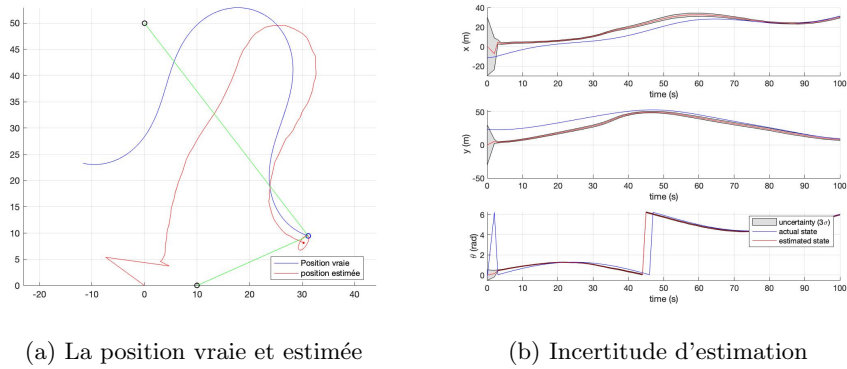


Figure 9: Résultat avec $0.1 * Qf$

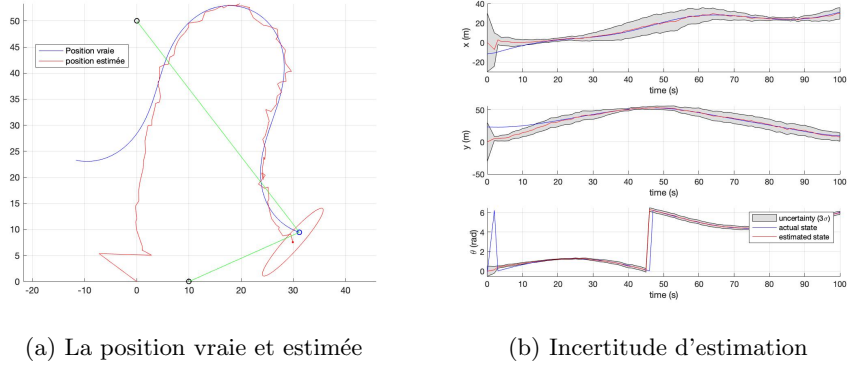


Figure 10: Résultat avec $10 * Qf$

3.4 Q4 - Faire varier le bruit de mesure du filtre

En réglant tous les autres paramètres sur la valeur par défaut, on modifie le bruit de la mesure à une nouvelle valeur ($Rf_i = \text{diag}([3, 1 * \pi / 180].^2)$) et illustre respectivement les résultats dans la Figure 11. En les comparant avec la Figure 7, on constate que l'incertitude est beaucoup plus faible. Un capteur de bonne qualité peut donc améliorer la qualité du contrôle et réduire son incertitude.

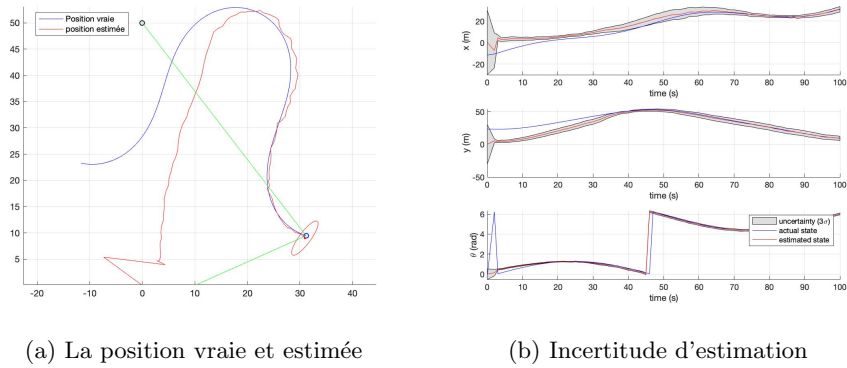


Figure 11: Résultat avec une nouvelle Rf_i

3.5 Q5 - Le performance du filtre en fonction du nombre d'amers

Dans cette section on définit plus d'amers (de 3 à 5) et on va étudier l'influence du nombre d'amers, la position d'amer est respectivement $[10;0]$, $[0;50]$, $[0;0]$, $[-10;25]$ et $[-20;40]$. Les résultats sont illustrés respectivement dans la Figure 12,

Figure 13 et Figure 14. En les comparant avec la Figure 7, on constate que plus le nombre d'amers était élevé, plus la concordance entre les emplacements réels et estimés apparaissait rapidement.

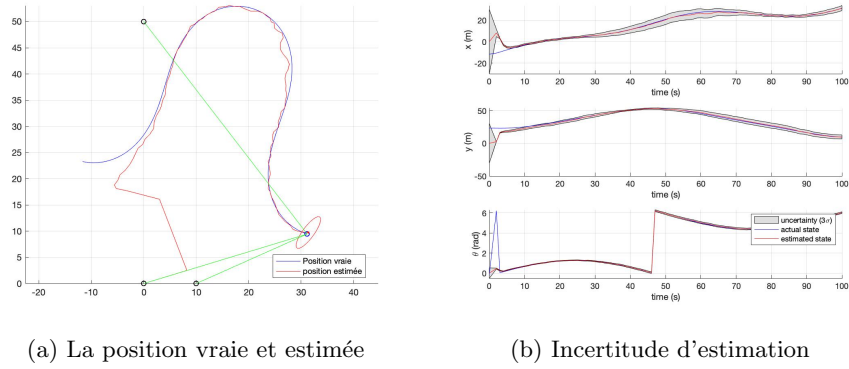


Figure 12: Résultat avec 3 amers

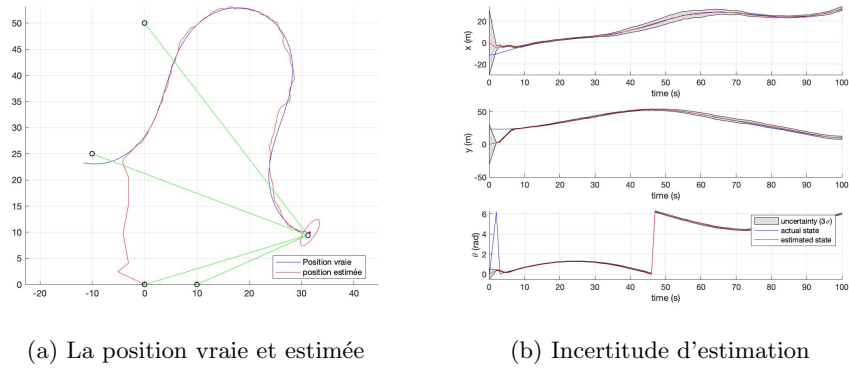
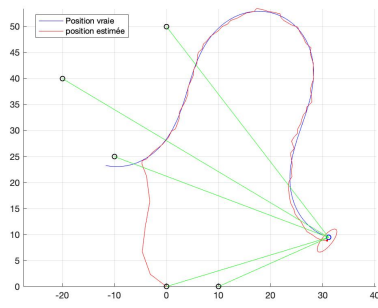
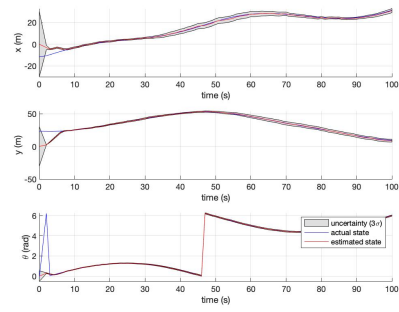


Figure 13: Résultat avec 4 amers



(a) La position vraie et estimée



(b) Incertitude d'estimation

Figure 14: Résultat avec 5 amers