



ÉCOLE NATIONALE SUPÉRIEURE DE TECHNIQUES AVANCÉES

TP2 : Filtrage de kalman pour la navigation

Auteur :
Vanessa LOPEZ

Encadrants :
M. Nicolas MERLINGE

ROB312 - Navigation pour les systèmes autonomes

22 octobre 2023

Table des matières

1	Introduction	2
2	Developpement du code	3
3	Analyses de resultat	5
3.1	Fréquence des mesures	6
3.2	Bruit de dynamique du filtre	7
3.3	Bruit de mesure du filtre	8
3.4	Trou de mesures	9
3.5	Nombre d'amers	9
3.6	Mesures de distance uniquement	10
3.7	Mesures de direction uniquement	11

1 Introduction

On va appliquer **le filtre de Kalman étendu pour des modèles non linéaires** sur un programme représentant la trajectoire d'un véhicule, lequel suivre les étapes suivantes :

Étape de prédiction :

- À l'aide du modèle de l'espace d'états du système, on va prédire l'estimation de l'état à l'instant k en fonction de l'estimation de l'état à l'instant $k-1$ et de l'entrée de commande appliquée à l'instant $k-1$.
- Prédire l'estimation de la covariance de l'état sur la base de la covariance précédente et d'un peu de bruit.

Étape de mise à jour (correction) :

- Calculer la différence entre les mesures réelles du capteur à l'instant k et les mesures prédites par le modèle de mesure pour le temps k actuel.
- Calculer la covariance résiduelle des mesures.
- Calculer le gain de Kalman optimal.
- Calculer une estimation actualisée et l'estimation de la covariance de l'état pour le temps k .

Globalement, le programme crée un environnement de simulation virtuel avec les mesures d'odométrie que le véhicule calcule où une référence et les valeurs de position et d'angle sont entrées avec un peu de bruit pour recréer les mesures qu'un capteur réel prendrait. Ensuite, des points aléatoires (amers) sont générés sur la carte pour simuler les capteurs qui mesurent la position réelle du robot et enfin, l'algorithme du filtre de Kalman sera appliqué pour corriger la trajectoire orienté par l'odométrie et donner la priorité aux capteurs de position (amers).

2 Developpement du code

Evolution théorique de l'état (modèle dynamique) : On commence à définir le modèle dynamique sur la fonction `motion_model` pour le mouvement du véhicule suivant la configuration décrite ci-dessous. On s'appuiera sur la fonction `tcomp` déjà présente dans le programme, qui fournit la dynamique du système en prenant les vecteurs de la position, de la vitesse et de la différence de temps.

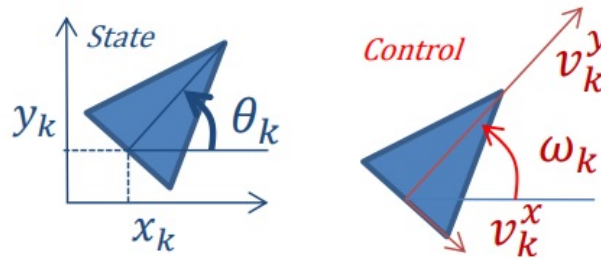


FIGURE 1 – Modèle dynamique

$$x_k = f(x_{k-1}, \tilde{u}_k) = \begin{bmatrix} x_{k-1} + (v_k^x \cdot \cos(\theta_{k-1}) - v_k^y \cdot \sin(\theta_{k-1})) \cdot \Delta t \\ y_{k-1} + (v_k^x \cdot \sin(\theta_{k-1}) + v_k^y \cdot \cos(\theta_{k-1})) \cdot \Delta t \\ \theta_{k-1} + w_k \cdot \Delta t \end{bmatrix} \quad (1)$$

Équation d'observation théorique (modèle de mesure) : Ensuite, avec le modèle d'observation, on utilise l'état pour estimer les mesures des capteurs qui représentent un vecteur de mesures prédites des capteurs y_k en ajoutant un peu de bruit. La carte des points de amers (représentés par les accords x^P, y^P) est connue a priori. À chaque temps k , un amers est choisi au hasard pour simuler une mesure.

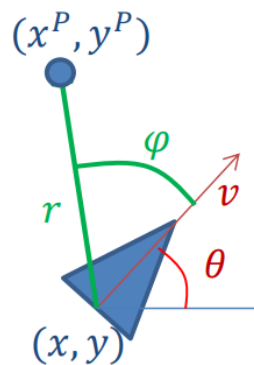


FIGURE 2 – Modèle de capteur

$$y_k = h(x_k) = \begin{bmatrix} r_k \\ \phi_k \end{bmatrix} = \begin{bmatrix} \sqrt{(x_k^P - x_k)^2 + (y_k^P - y_k)^2} \\ \text{atan} \frac{y_k^P - y_k}{x_k^P - x_k} - \theta_k \end{bmatrix} \quad (2)$$

Le système est maintenant exprimé comme un système non linéaire car l'état est une fonction des cosinus et des sinus (qui sont des opérations trigonométriques non linéaires). Alors, on doit linéariser ces équations en calculant le jacobien (matrice des dérivées partielles) déjà exprimée dans la matrice suivante :

$$H_k = \frac{\partial h}{\partial x} = \begin{bmatrix} -\frac{a}{\sqrt{(a^2+b^2)}} & -\frac{b}{\sqrt{(a^2+b^2)}} & 0 \\ \frac{b}{a^2+b^2} & \frac{a}{a^2+b^2} & -1 \end{bmatrix} \quad (3)$$

où $a = x_k^P - x_k$ et $b = y_k^P - y_k$.

Appliquer ensuite les équations du filtre de Kalman de prédiction calculées avec l'estimation du modèle illustrée à la figure 3 (a) et la correction de la prédiction en ajoutant le gain k_k (en rouge) pour l'innovation (en verte) illustrée à la figure 3 (b).

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

(a) Prediction step

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

(b) Correction step

FIGURE 3 – Extended Kalman Filter for nonlinear models

3 Analyses de resultat

Enfin, on peut constater que le EKF (Extended Kalman Filter) calcule une moyenne pondérée des mesures réelles des capteurs et des mesures prédites des capteurs afin de générer une meilleure estimation de l'état actuel.

Il est évident que le filtre de Kalman accorde une plus grande priorité à la localisation des amers qu'à l'odométrie. Sur la droite de la figure 8 se trouve l'erreur entre l'état estimé et l'état réel (courbe bleue) et sa covariance (courbe rouge) qui varie de $\pm 3\sigma$ et on peut observer que cette courbe se referme plus lorsqu'elle passe près des points du capteur (amers) car elle a une correction abrupte des valeurs de position par rapport à amers. La covariance d'angle reste généralement stable sur des valeurs de $\pm 3\sigma$ et présente quelques petits pics dus à la linéarisation du modèle.

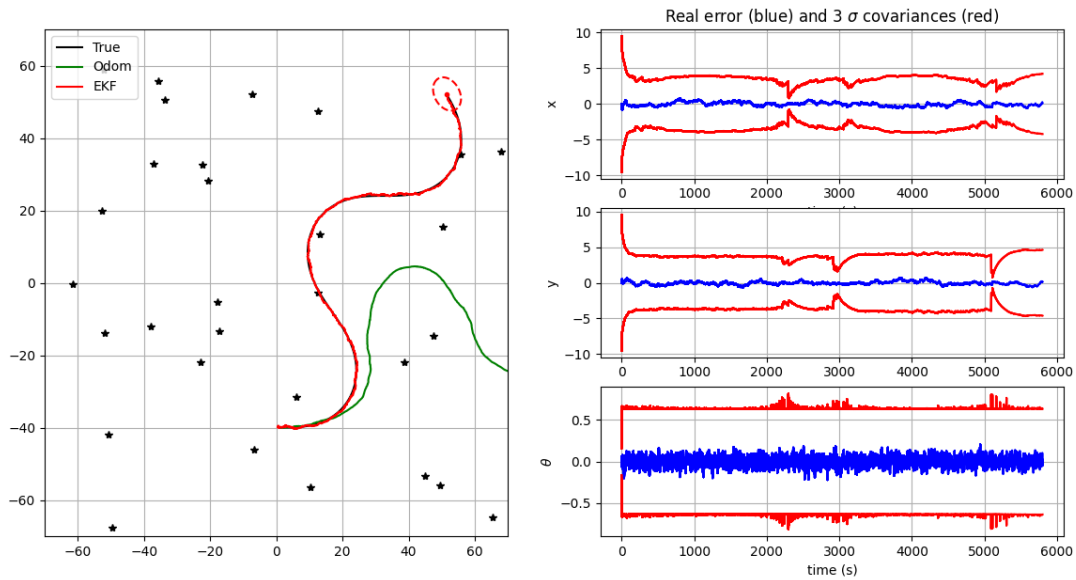


FIGURE 4 – Résultat EKF

3.1 Fréquence des mesures

On va modifier la fréquence des mesures en utilisant la variable `dt_mesure`.

La variable `dt_mesure` est utilisée pour obtenir le module par rapport aux itérations du programme (6000) dans la méthode `get_observation` pour obtenir le modèle d'observation avec un certain bruit. Rappelons que ce modèle utilise l'état (ou l'état prédit au temps suivant) pour estimer (ou prédire) les mesures des capteurs. Par conséquent, si on augmente le temps pendant lequel ces mesures sont calculées et prédites, la trajectoire sera moins précise.

Les figures ci-dessous montrent qu'à mesure que la fréquence d'échantillonnage diminue, la covariance du modèle augmente de manière significative, car les conditions préalables du modèle sont étroitement espacées et tendent à produire des estimations médiocres. La figure 5 (d) présente des pics de variance importants, car l'incertitude est grande pendant les périodes sans échantillonnage, mais lorsque l'échantillonnage est effectué, l'incertitude diminue considérablement.

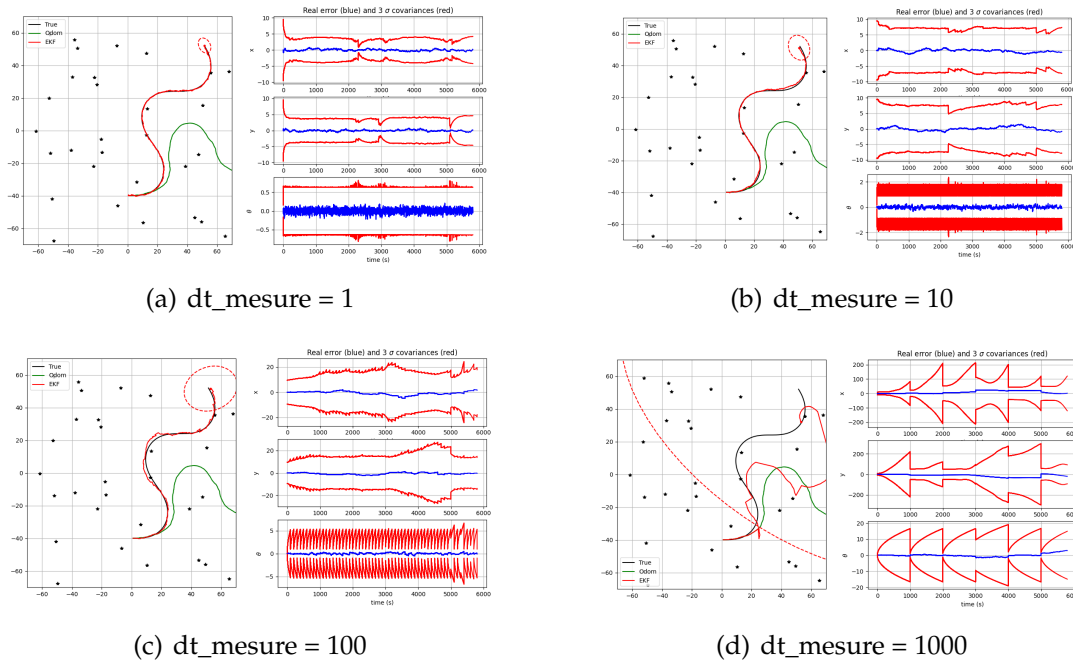


FIGURE 5 – Modification de fréquence des mesures

L'augmentation de la fréquence des mesures ne serait pas très optimale, car on prendrait plus d'échantillons que le nombre de fois où une prédiction est faite (`dt_pred=1`) et pour certaines valeurs, le temps de prédiction ne coïncide pas avec le temps d'échantillonnage, donc le module ne sera jamais égal à 0 et ne générerait pas le modèle d'observation.

3.2 Bruit de dynamique du filtre

On va faire varier le bruit de dynamique du filtre (matrice Q_{Est}).

En augmentant le bruit de la dynamique du filtre, on peut observer que la trajectoire contient plus de bruit mais donne la priorité aux mesures des capteurs amers et la covariance augmente parce que le bruit dynamique du filtre continue à s'accumuler.

Si nous regardons les équations de la figure 3 (b), nous pouvons voir qu'en ayant un Q très grand, la covariance prédite de l'estimation de l'état ($\hat{P}_{K|k-1}$) et le gain K_k (qui dépend de cette covariance) augmentent et donnent la priorité au terme d'innovation (en vert), qui est la différence entre les observations réelles du capteur et les observations prédites du capteur (erreur de prédiction) en se concentrant sur le fait de baser la trajectoire sur les mesures des capteurs.

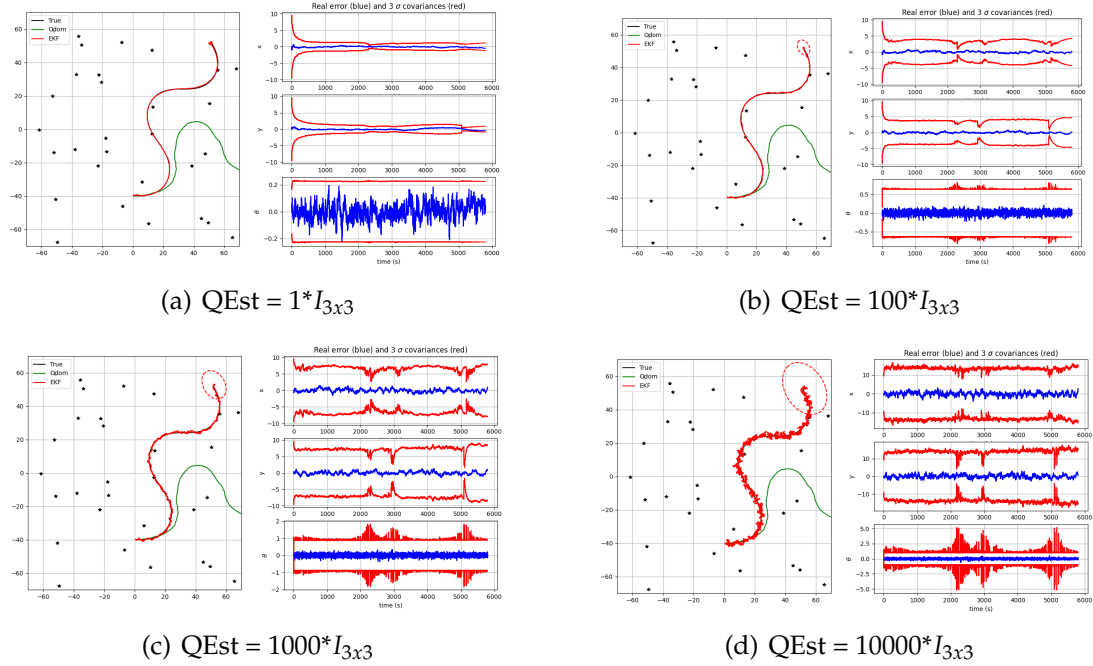


FIGURE 6 – Modification de bruit de dynamique du filtre

3.3 Bruit de mesure du filtre

On va faire varier le bruit de mesure du filtre (matrice R_{Est})

Il est évident avec ces modifications que le modèle tend à se détacher de la trajectoire des mesures des capteurs amers et commence à suivre en priorité l'odométrie, car avec un R très grand, le gain K_k devient très petit à cause de l'inverse de S (qui contient le R dans son équation) donnant très peu d'importance à l'innovation et ignorant presque les mesures des capteurs.

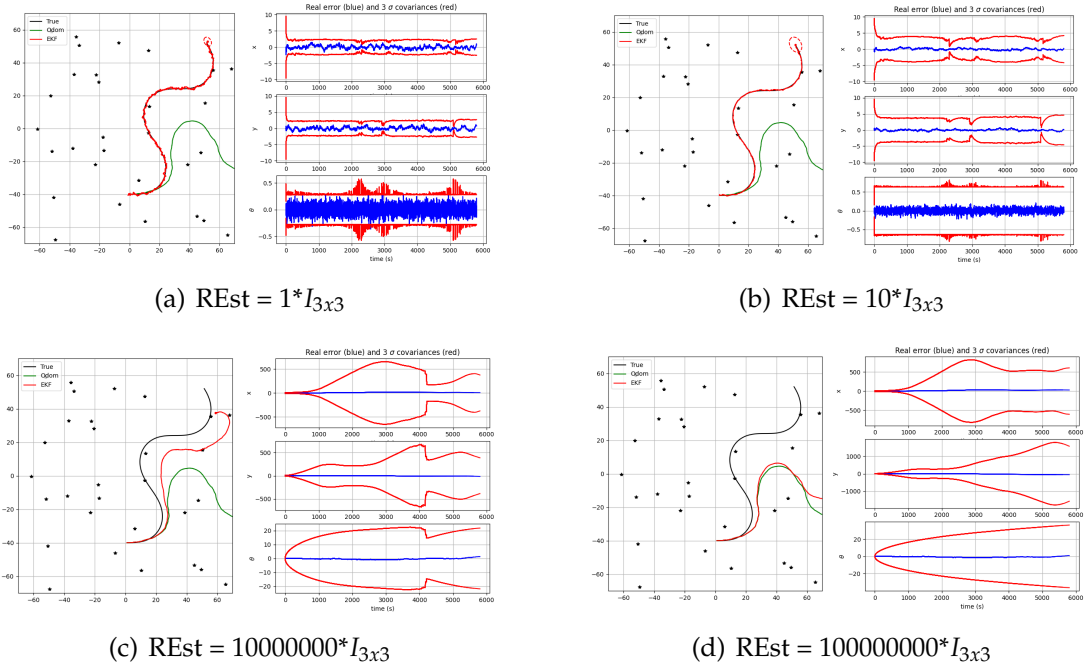


FIGURE 7 – Modification de bruit de dynamique du filtre

3.4 Trou de mesures

On va simuler un trou de mesures entre $t = 2500$ s et $t = 3500$ s en utilisant la ligne de code suivante :

```
notValidCondition = False if k < 2500 or k > 3500 else True
```

En faisant un trou dans les mesures, pendant la période de non-échantillonnage, il n'effectue que la partie prédiction et n'effectue pas la correction 3 (b) du filtre de kalman. Par conséquent, un écart par rapport à la trajectoire peut être observé, car il avait une priorité plus élevée dans les valeurs du capteur (Q_{Est} est plus élevé que R_{Est}) et a cessé de fournir des informations, générant ainsi une perte de trajectoire dans ce laps de temps.

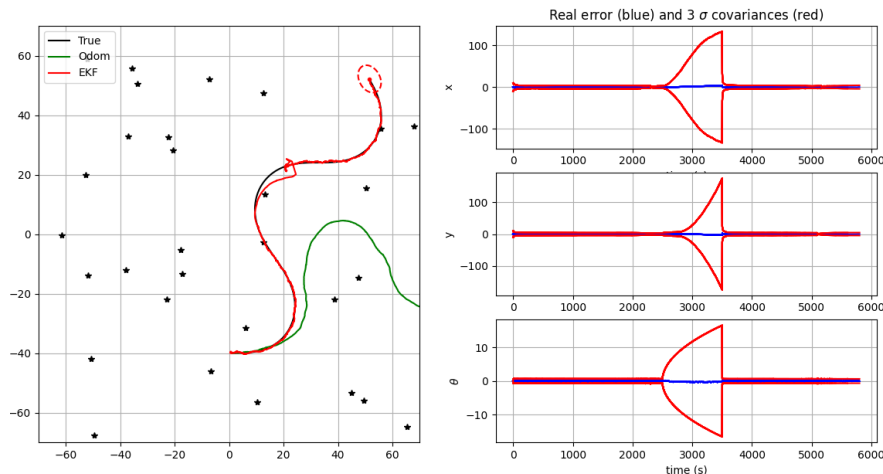


FIGURE 8 – Trou de mesures entre $t = 2500$ s et $t = 3500$ s

3.5 Nombre d'amers

On va faire varier le nombre d'amers et étudier les performances du filtre en fonction du nombre d'amers. Ensuite, on réglerait le filtre pour obtenir les meilleures performances possibles.

On peut observer qu'il n'y a pas d'amélioration significative du filtre de Kalman avec l'augmentation du nombre d'amers. Il n'y a donc pas de quantité optimale de points amers, mais ce qui nous intéresse le plus est leur distribution, puisque les prédictions seront beaucoup plus précises si certains points sont proches de la trajectoire effectuée par le robot, par opposition à des mesures de capteurs éloignées de cette trajectoire.

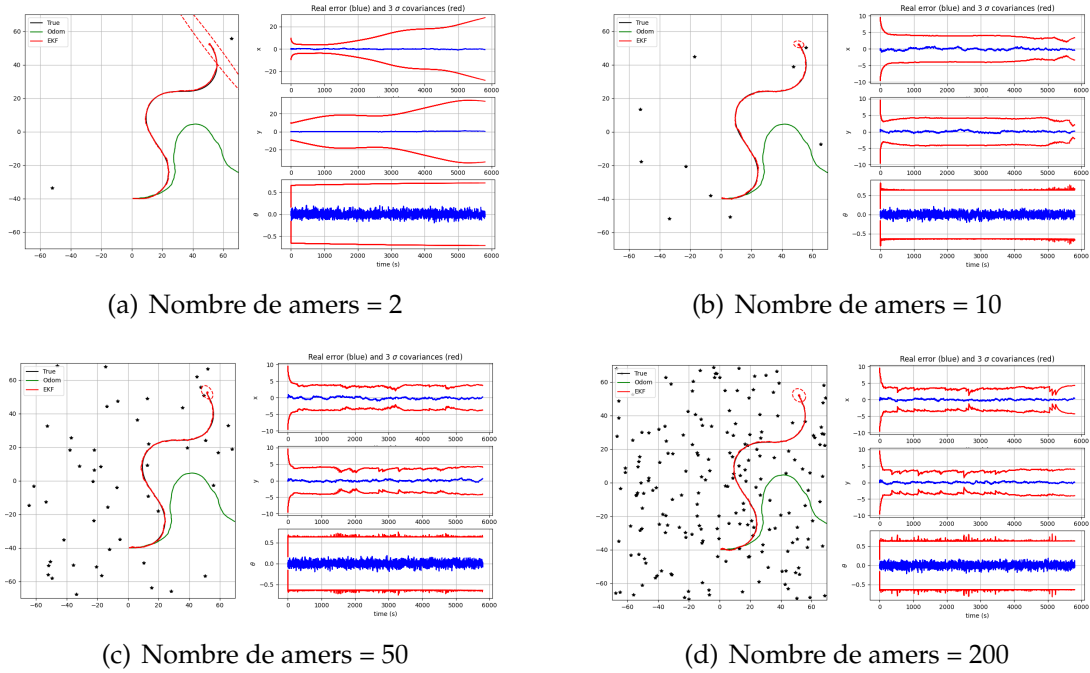


FIGURE 9 – Modification de nombre d'amers

3.6 Mesures de distance uniquement

On va simuler le cas où seulement les mesures de distance sont disponibles (range only) : en modifiant le code, en réglant le filtre (matrices de covariances QEst et REst). On pourra au besoin jouer sur le nombre d'amers.

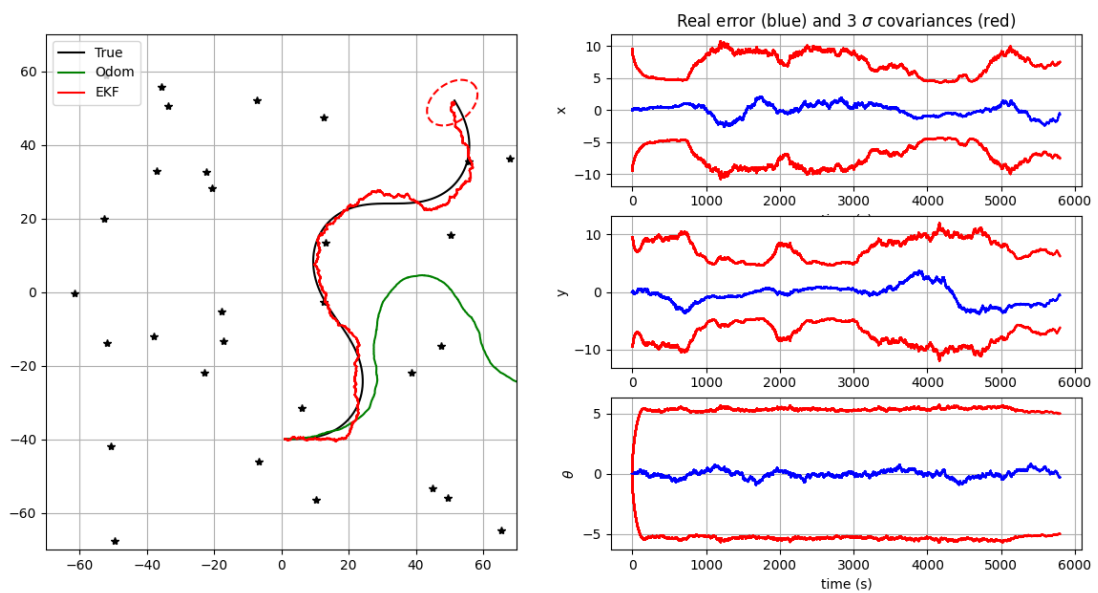


FIGURE 10 – Mesures de distance uniquement

L'EKF a peut-être rencontré des difficultés parce qu'il traite le système comme étant linéaire (sans des opérations trigonométriques non linéaires) en conduisant à des prédictions inexactes et lorsque le modèle linéarisé ne correspond pas au comportement non linéaire, la matrice de covariance n'est plus adaptée. En outre, il est plus probable qu'avec la distance, la direction ne soit pas totalement claire car elle peut être également éloignée de 2 points.

3.7 Mesures de direction uniquement

On va simuler le cas où seulement les mesures de direction sont disponibles (angles only) : en modifiant le code, en réglant le filtre (matrices de covariances QEst et REst). On pourra au besoin jouer sur le nombre d'amers

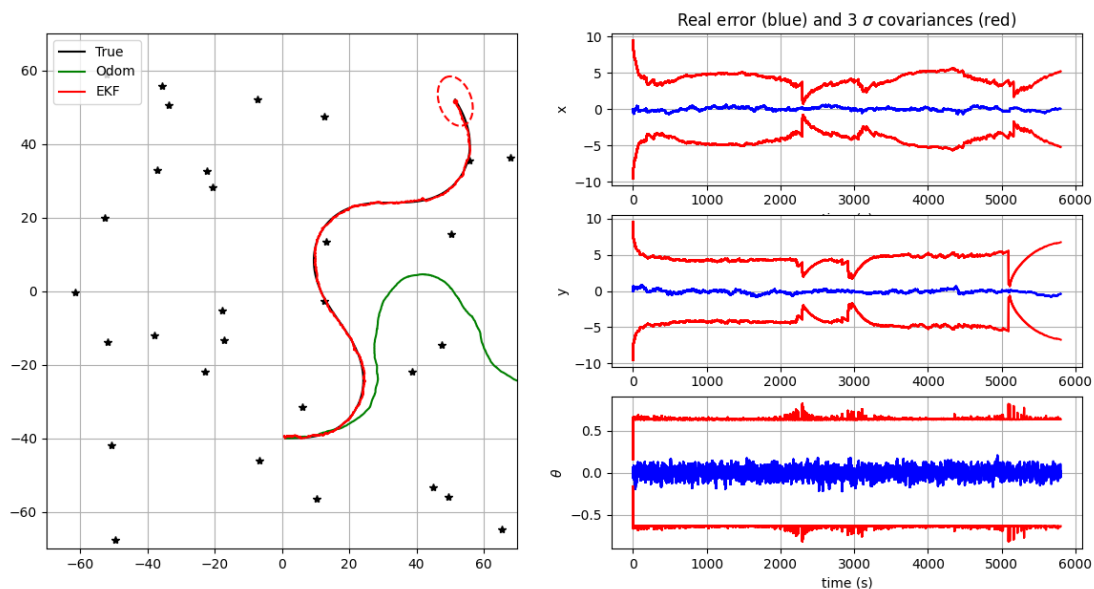


FIGURE 11 – Mesures d'angle uniquement

Lorsque le système de mesure ne contient que l'angle du robot, l'EKF peut gérer la non-linéarité du système de manière plus efficace et de plus en plus précise afin de faire une prédiction exacte. Aussi, on n'a pas le même problème de distance parce qu'on connaît mieux la direction d'angle et que le filtre obtient une localisation plus précise.