

# ROB312 - TP3 : Filtrage Particulaire

Dajing GU

Décembre 2020

## Introduction

Le **Filtrage Particulaire** est une méthode générale de **Filtrage de Kalman**. L'idée du filtrage particulaire est basée sur la méthode de Monte Carlo, qui utilise des ensembles de particules pour représenter la probabilité. En remplaçant l'opération intégrale par **la moyenne de l'échantillon**, on peut acquérir l'approximation de la fonction de densité de probabilité.

Contraire au Filtrage Kalman dont l'espace d'état doit être linéaire et le bruit doit correspondre la distribution gaussienne, le modèle de l'espace d'état et le bruit ne sont pas limités dans le Filtrage Particulaire. Il peut exprimer une distribution plus large que le modèle gaussien, et il a une capacité de modélisation plus forte pour les caractéristiques **non linéaires**.

Malgré les avantages de filtrage particulaire, on ne peut pas ignorer **le grand nombre d'échantillons** demandé: plus le nombre d'échantillons est grand, plus la complexité de l'algorithme est élevée. En plus, la phase de ré-échantillonnage entraîne une perte de validité et de diversité de l'échantillon. .

## Q1 - La structure du code et la présentation des paramètres

Dans cette section, la structure du code et les paramètres sont présentés. Afin d'avoir une connaissance plus claire de la structure, on va présenter tout d'abord les différents paramètres.

- **X<sub>reel</sub>** : L'état vrai du système, qui est inconnu donc on donne une valeur d'estimation. Il est initialisé par des valeurs estimées et des erreurs d'initialisation.
- **Y** : La simulation physique des mesures.
- **R** : La matrice de covariance du bruit de mesure réelle.
- **T** : La durée de la simulation.
- **N** : Le nombre de particules.
- **X<sub>hat</sub>** : L'estimation initiale  $(x, y, z, \theta)$
- **P<sub>hat</sub>** : La matrice de covariance.
- **Q<sub>f</sub>** : La matrice de covariance de bruit de dynamique

- $R_f$  : La matrice de covariance du bruit de mesure du filtre
- `threshold_resampling` : Le seuil de ré-échantillonnage ( $\theta_{eff}$ ). Le ré-échantillonnage va être fait si  $\frac{1}{\sum w_p^2} < N \cdot \theta_{eff}$
- $R_f$  : La matrice de covariance du bruit de mesure du filtre
- $X_p$  : Le tirage des particules autour de  $X_{hat}$  initial
- $w_p$  : Les poids initiaux associés aux particules

En utilisant les paramètres présentés ci-dessus, la structure du code est facilement présenté dans la figure 0.1.

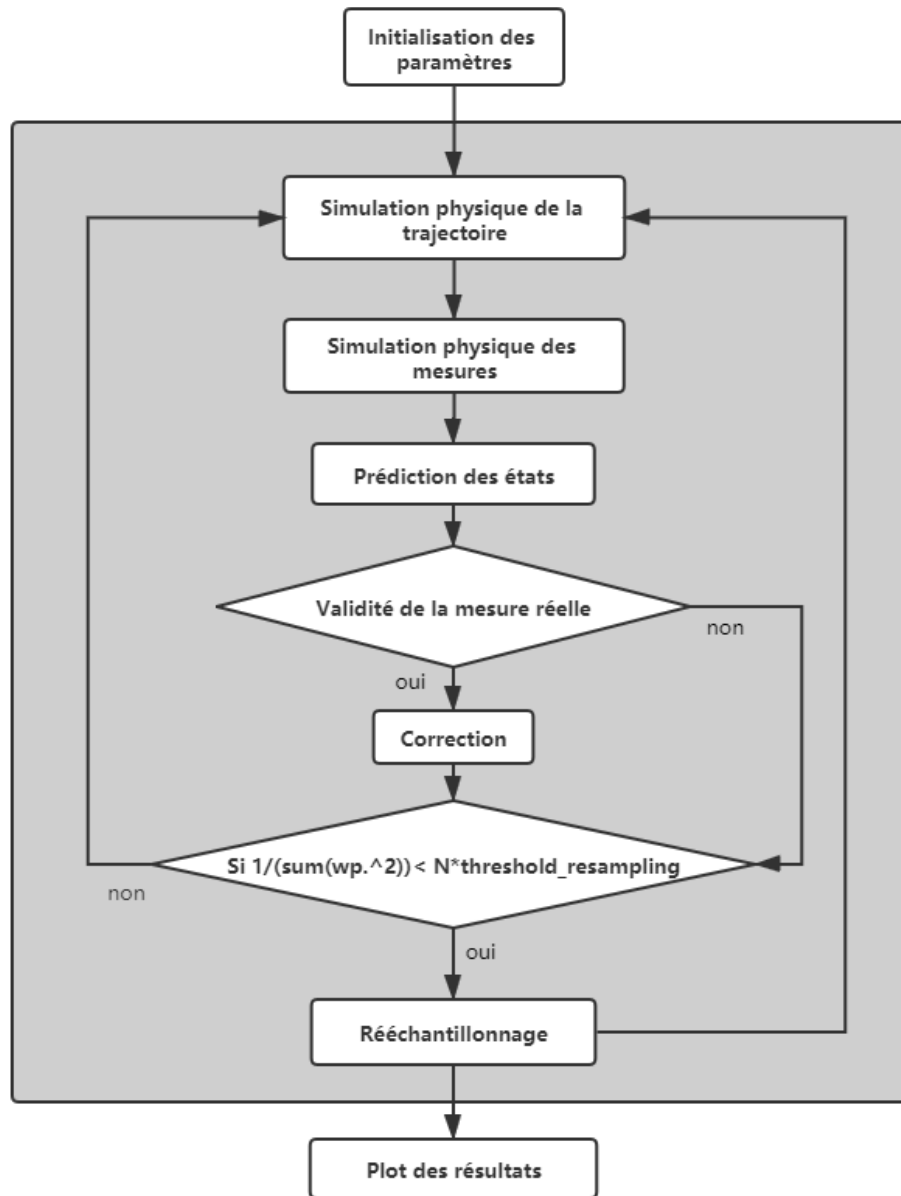
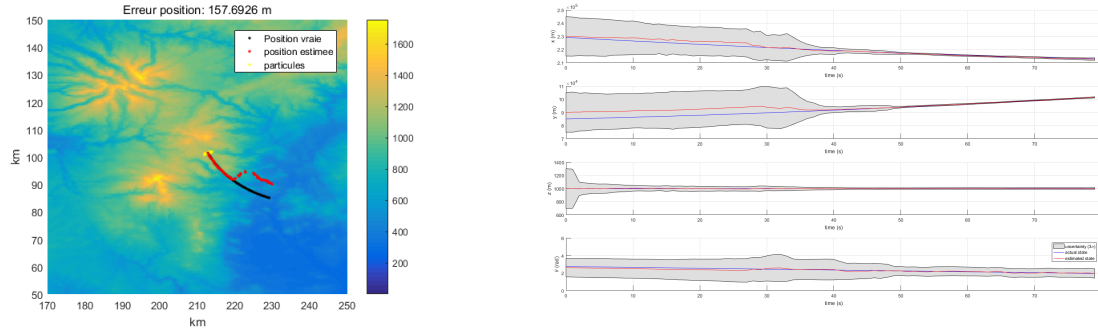


Figure 0.1: Schéma de la structure du code

## Q2 - Remplissage du code et commentaire des résultats

En fonction des équations du filtre PF (slide 24), du modèle dynamique et du modèle de mesure (slide 28), on a réussi à remplir le code. Les résultats sont présentés dans la figure 0.2.



(a) La position vraie et la position estimé

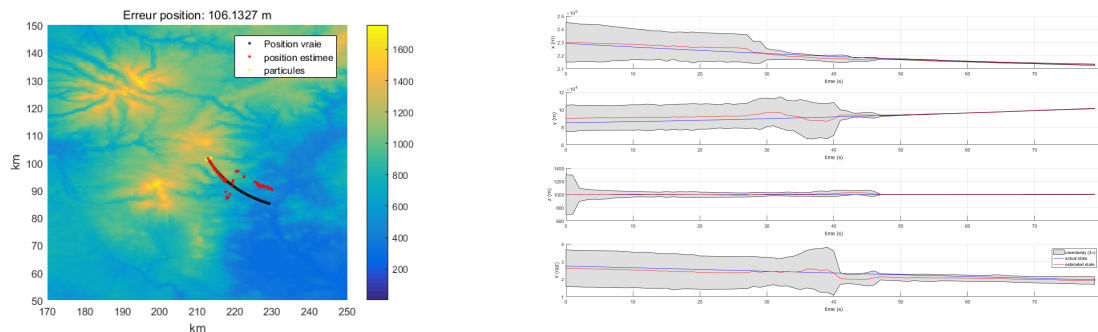
(b) Incertitude d'estimation

Figure 0.2: Résultat

On peut trouver que la différence entre les états réels et les états estimés est vraiment grande au début, avec une erreur d'environ 30,000 m. L'incertitude initiale est aussi grande. Mais le filtrage particulière a réussi à nous amener la convergence de l'erreur et de l'incertitude. Finalement, l'erreur entre l'état réel et l'état estimé a été réduit à 158m et l'incertitude devient aussi beaucoup plus petite qu'au début.

## Q3 - Influence de différents $Q_f$

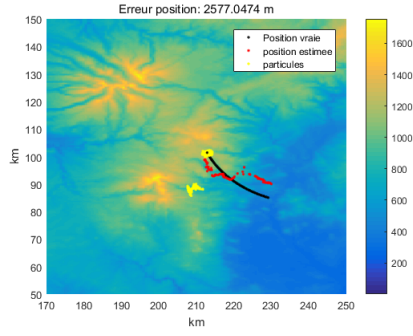
Dans cette section, on a varié le bruit de dynamique du filtre (matrice  $Q_f$ ) pour étudier l'influence du bruit de dynamique du filtre sur son comportement. On a utilisé 3 valeurs:  $0.1 \cdot Q_f$ ,  $10 \cdot Q_f$  et  $50 \cdot Q_f$ . On définit tous les autres paramètres par défaut.



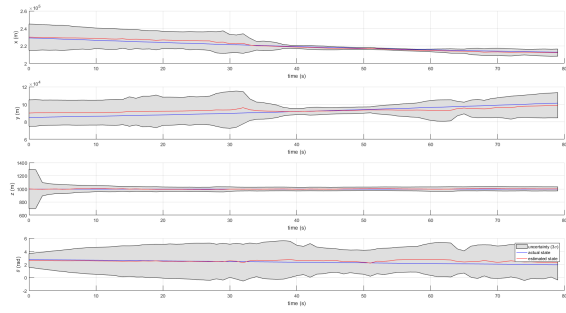
(a) La position vraie et la position estimé

(b) Incertitude d'estimation

Figure 0.3: Résultat avec  $0.1 \cdot Q_f$

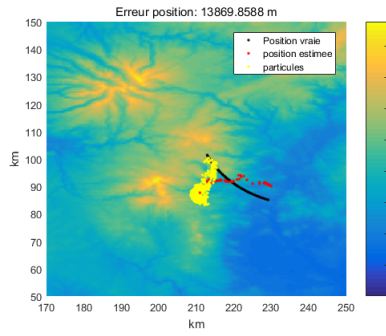


(a) La position vraie et la position estimé

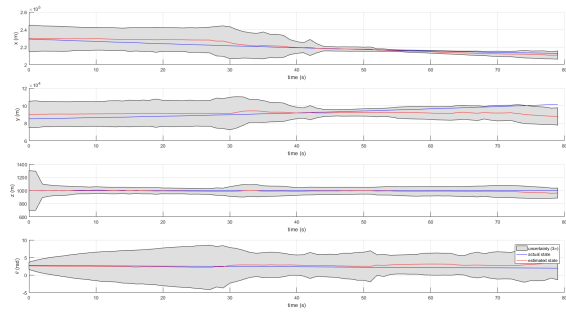


(b) Incertitude d'estimation

Figure 0.4: Résultat avec  $10 \cdot Q_f$



(a) La position vraie et la position estimé



(b) Incertitude d'estimation

Figure 0.5: Résultat avec  $50 \cdot Q_f$

Le bruit de dynamique ( $Q_f$ ) interfère dans la procédure de prédiction. Un trop grand bruit de dynamique va bloquer la prédiction correcte de l'état réel, qui nous amène ensuite une trop grande valeur d'erreur et d'incertitude.

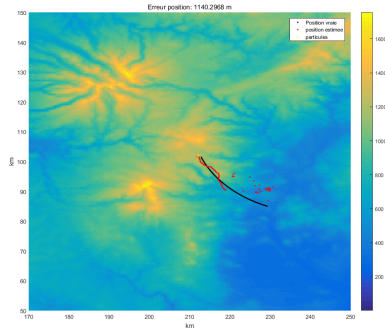
Selon les trois figure 0.3, 0.4 et 0.5, on trouve que plus le bruit de dynamique est petit, plus l'incertitude d'estimation est petite. En plus, plus le bruit de dynamique est petit, plus l'erreur finale d'estimation est petite.

Plus précisément, dans le cas de  $0.1 \cdot Q_f$ , l'erreur finale est 106 m, qui est plus petite que le cas dans la figure 0.2, mais dans le cas de  $10 \cdot Q_f$ , l'erreur est 2577 m. Dans le cas de  $50 \cdot Q_f$ , l'erreur est 13889 m, qui est complètement incorrect.

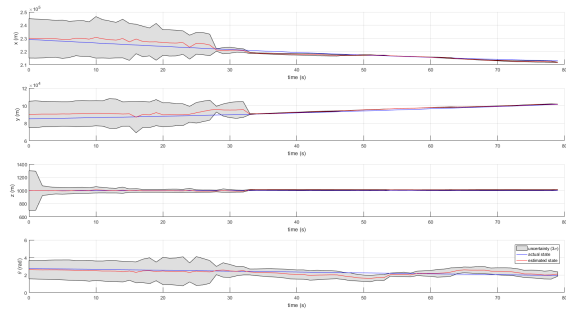
## Q4 - Influence de différents $R_f$

Dans cette section, on a varié le bruit de mesure (matrice  $R_f$ ) pour étudier l'influence du bruit de mesure du filtre sur son comportement. On a essayé différentes valeurs de  $R_f$  entre  $10^2$  et  $100^2$ :  $10^2$ ,  $20^2$ ,  $50^2$ ,  $100^2$ . On définit tous les autres paramètres par défaut.

$R_f$  interfère dans la correction des poids des particules, qui influence directement l'incertitude.

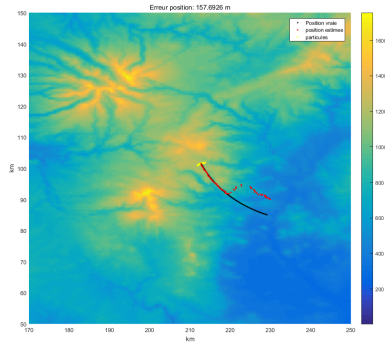


(a) La position vraie et la position estimé

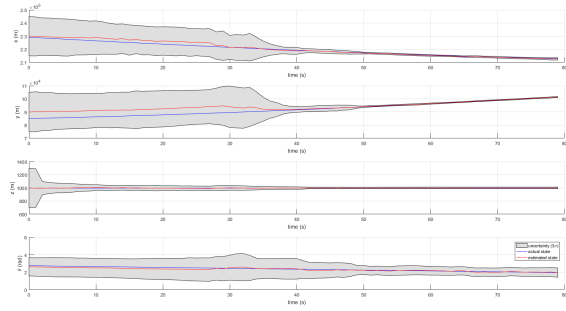


(b) Incertitude d'estimation

Figure 0.6: Résultat avec  $R_f = 10^2$

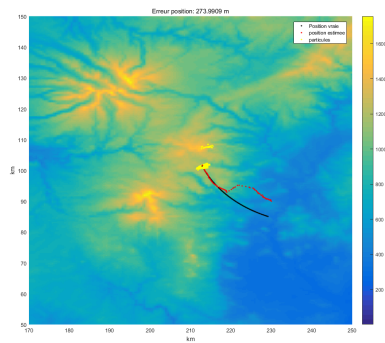


(a) La position vraie et la position estimé

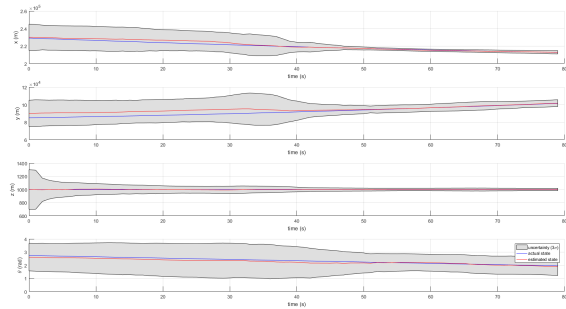


(b) Incertitude d'estimation

Figure 0.7: Résultat avec  $R_f = 20^2$

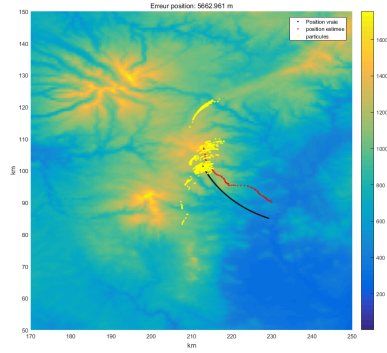


(a) La position vraie et la position estimé

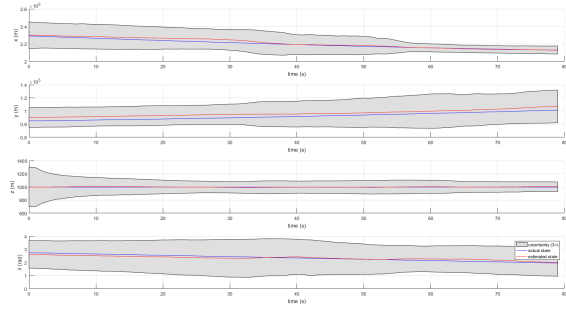


(b) Incertitude d'estimation

Figure 0.8: Résultat avec  $R_f = 50^2$



(a) La position vraie et la position estimé



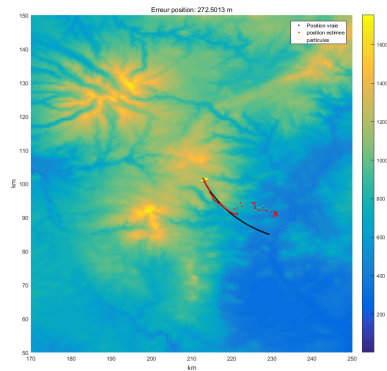
(b) Incertitude d'estimation

Figure 0.9: Résultat avec  $R_f = 100^2$

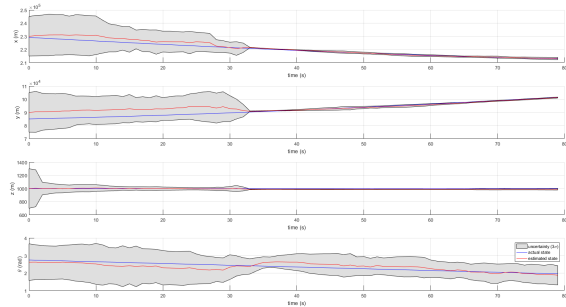
Selon les trois figure 0.6, 0.7, 0.8 et 0.9, on trouve que plus le bruit de mesure est petit, plus la convergence d'incertitude se passe tôt, c'est à dire que ça prend moins de temps pour que l'incertitude devienne petite. De plus, quand  $R_f$  est trop grand, l'incertitude va devenir grande pour toute la durée de simulation.

## Q5 - Influence de nombre de particule N

Dans cette section on va étudier l'influence de nombre de particule N sur le comportement de filtre en essayant différentes valeurs de N: 300, 30000 (par défaut N = 300, qui est présenté dans la figure 0.2). On définit tous les autres paramètres par défaut.

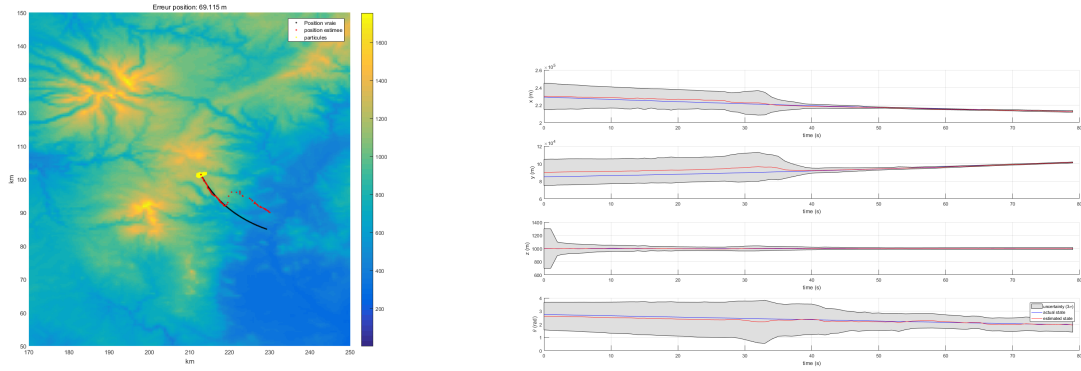


(a) La position vraie et la position estimé



(b) Incertitude d'estimation

Figure 0.10: Résultat avec  $N = 300$



(a) La position vraie et la position estimé

(b) Incertitude d'estimation

Figure 0.11: Résultat avec  $N = 30,000$

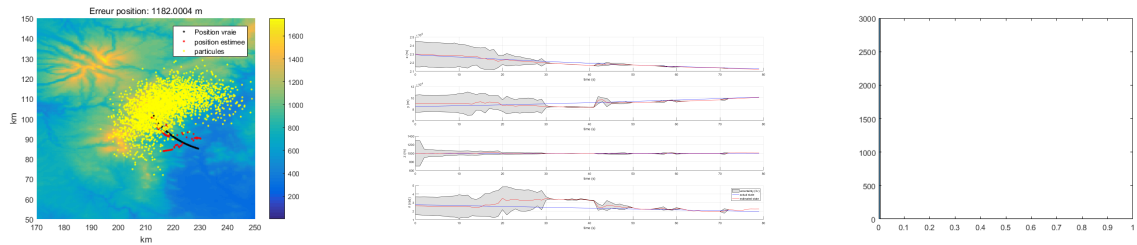
Selon la figure 0.10 et 0.11, on trouve que l'incertitude devient petite plus tôt quand il y a moins de particules, mais l'erreur associée est plus grande.

On peut en déduire que plus le nombre de particules est élevé, plus la vitesse de calcul est lente et meilleur est l'effet de convergence (plus l'erreur est faible), mais plus le temps demandé pour réduire l'incertitude est grand.

## Q6 - Influence de seuil de ré-échantillonnage

Dans cette section on va étudier l'influence de valeur de seuil de ré-échantillonnage sur le comportement de filtre en essayant différentes valeurs de seuil: 0, 0.5 et 1. Les histogrammes des poids vont aussi être tracés, le phénomène de dégénérescence et l'impact du ré-échantillonnage vont être identifiés. On définit tous les autres paramètres par défaut.

### Seuil = 0



(a) La position vraie et la position estimé

(b) Incertitude d'estimation

(c) Histogramme des poids

Figure 0.12: Résultat avec  $\theta_{eff} = 0$

Quand  $\theta_{eff} = 0$ , on ne fait jamais le ré-échantillonnage et le problème de dégénérescence apparaît car tous les poids des particules sont nuls en fonction de l'histogramme des poids.

Seuil = 0.5

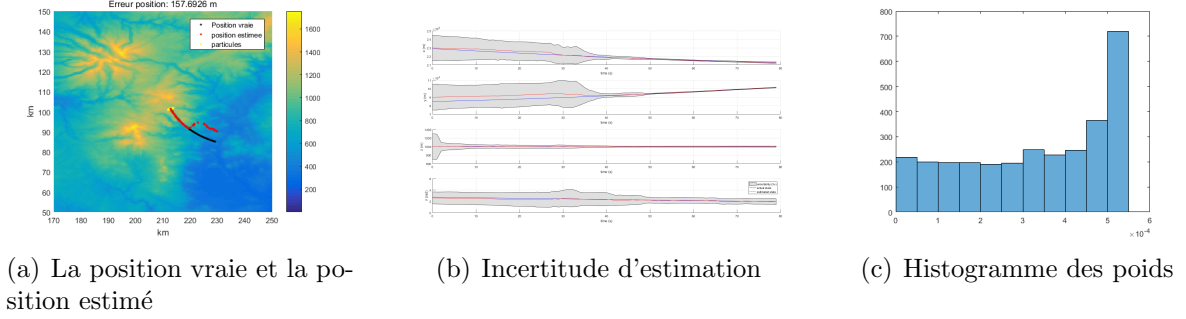


Figure 0.13: Résultat avec  $\theta_{eff} = 0.5$

Quand  $\theta_{eff} = 0.5$  (par défaut), le ré-échantillonnage est fait et les poids des particules ne sont plus des valeurs nulles.

Seuil = 1

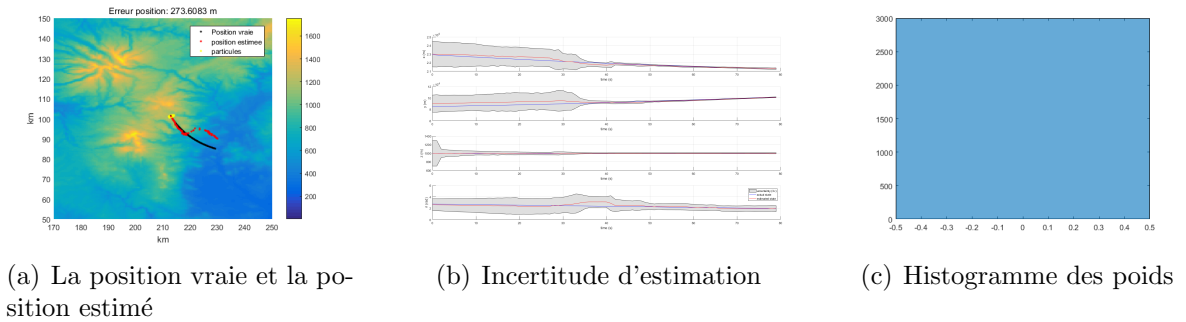


Figure 0.14: Résultat avec  $\theta_{eff} = 1$

Quand  $\theta_{eff} = 1$ , on fait plus de ré-échantillonnage et le problème de dégénérescence ré-apparaît car tous les poids des particules ont la même valeur, qui n'est pas zéro cette fois mais  $3.3 \times 10^{-4}$ .

## Q7 - Simulation avec un trou de mesure

Dans cette section, on a simulé un trou entre  $t = 50s$  et  $t = 75s$  en utilisant le code ci-dessous. On définit tous les autres paramètres par défaut.

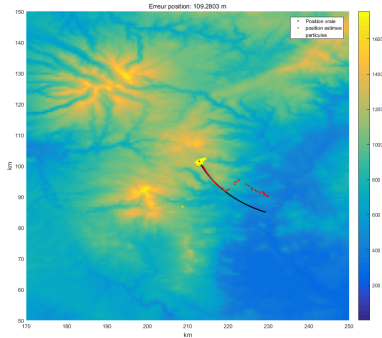
```
1 % validite de la mesure reelle (a completer pour la gestion des ...
   frequences et des trous de mesures)
```



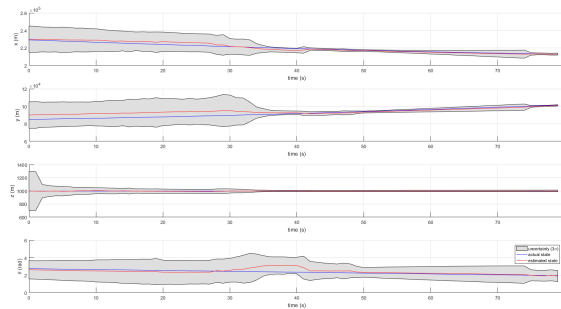
```

2 is.measurementValid = true;
3 if tk > 50/dt && tk < 75/dt
4     is.measurementValid = false;
5 end

```



(a) La position vraie et la position estimé



(b) Incertitude d'estimation

Figure 0.15: Résultat avec une simulation de trou de mesure

Selon la figure 0.15, on peut trouver que l'incertitude augmente dans le trou à cause de manque de mesure et la correction des poids. Malgré manque de mesure, finalement le filtre a bien estimé l'état, qui est présenté par le petit erreur 109 m.

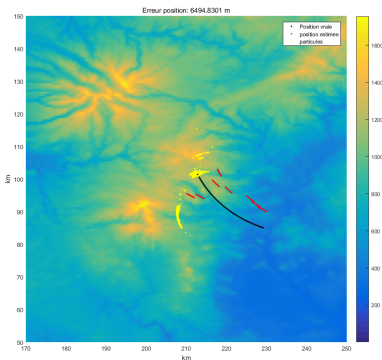
## Q8 - Modification de fréquence

Dans cette section, on a modifié la fréquence de mesure ( $0.1 \text{ Hz}$ ) en utilisant la variable `is.measurementValid`.

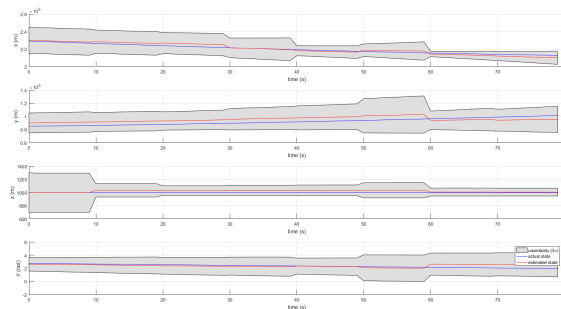
```

1 is.measurementValid = false;
2 if mod(tk, 10) == 0
3     is.measurementValid = true;
4 end

```



(a) La position vraie et la position estimé



(b) Incertitude d'estimation

Figure 0.16: Résultat avec la fréquence de mesure =  $0.1 \text{ Hz}$

En utilisant une fréquence de mesure = 0.1 Hz, on peut voir dans la figure 0.16 l'incertitude diminue chaque 10 secondes. En comparaison avec la figure 0.2 où la mesure est toujours disponible, cela nous permet d'avoir une connaissance plus claire comment la mesure nous aide de réduire l'incertitude et comment l'incertitude augmente quand il n'y a pas de mesure.

## Q9 - Une autre façon de ré-échantillonnage de poids

Dans cette section, on a utilisé une autre façon de ré-échantillonnage de poids. Comme l'algorithme de ré-échantillonnage multinomial a déjà été utilisé dans le code, on a choisi d'utiliser la méthode de ré-échantillonnage stratifié, dont le pseudo-code présenté dans la figure 0.17.

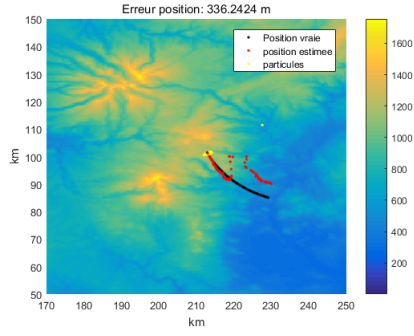
**Code 3:** Multinomial/stratified/systematic resampling.

```
[{ $\tilde{x}_t^{(n)}$ }]n=1N = Resample[{ $x_t^{(m)}$ ,  $w_t^{(m)}$ }]m=1M, N]

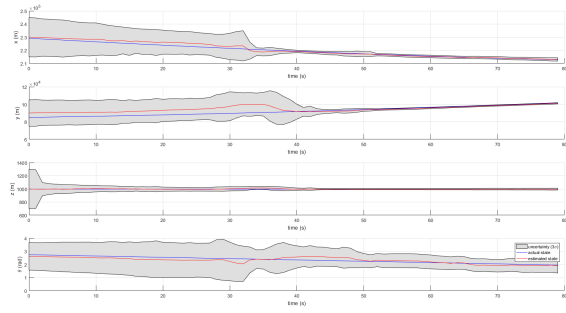
[ $Q_t^{(m)}$ ]m=1M = CumulativeSum[{ $w_t^{(m)}$ }]m=1M
n = 0
/Systematic/stratified choice runs:
m = 1
/Systematic choice runs
u0 ~ U(0, 1/N]
WHILE (n ≤ N)
/Systematified choice runs
u0 ~ U(0, 1/N]
/Systematic/stratified choice runs
u = u0 + n/N
/Multinomial choice runs
u ~ U(0, 1]; m = 1
WHILE ( $Q_t^{(m)} < u$ )
m = m + 1
END
n = n + 1
 $\tilde{x}_t^{(n)} = x_t^{(m)}$ 
END
```

Figure 0.17: Pseudo-code de ré-échantillonnage

```
1 function [index] = stratified(wp)
2     N = length(wp);
3     index = zeros(1,N);
4     Q = cumsum(wp);
5     n = 1;
6     m = 1;
7     while (n ≤ N && m ≤ N)
8         u0 = rand/N;
9         u = u0 + n/N;
10        while (Q(m) < u && m < N)
11            m = m+1;
12        end
13        index(n) = m;
14        n = n+1;
15    end
16 end
```



(a) La position vraie et la position estimé



(b) Incertitude d'estimation

Figure 0.18: Résultat avec la méthode de ré-échantillonnage stratifié

Le résultat de la méthode de ré-échantillonnage stratifié est présenté dans la figure 0.18, le filtrage particulière a réussi à nous amener la convergence de l'erreur et de l'incertitude. Mais l'erreur cette fois est 336 m, qui est un peu plus grande que le cas par défaut (la méthode de ré-échantillonnage multinomial).