ROB312 - Navigation pour les systèmes autonomes
TP3 Filtrage Particulaire pour la navigation

SANTOS SANO Matheus

Palaiseau, France

Octobre 2023

Exercice 1

Le code est l'implémentation du filtre particulaire pour la localisation d'un robot. Contrairement au filtre de Kalman, le filtre particulaire est très efficace pour traiter les systèmes non gaussiens. Ce filtre est une méthode probabiliste qui vise à estimer l'état d'un système dynamique. Dans notre cas, le filtre est implémenté pour localiser un robot mobile dans un environnement simulé.

Au début, un ensemble de N particules représentant les possibles états actuels du robot est initialisé. Les particules sont des échantillons aléatoires qui suivent une distribution gaussienne, comme indiqué ci-dessous :

$$x_i^0 \sim \mathcal{N}(\hat{x_0}, \hat{P_0}), \quad i \in [1, N] \tag{1}$$

Une fois les particules initialisées, l'étape de prédiction est réalisée. Chaque particule est mise à jour par une prédiction d'état en fonction de la dynamique du robot. Dans ce cas, on implémente un modèle mathématique (Équation (2)) pour estimer le prochain état de chaque particule. Pour cette étape de prédiction, il faut prendre en compte l'incertitude associée au mouvement du robot qui est représentée par le bruit v_k dans l'équation (2).

$$x_k^i = f(x_{k-1}^i, u_k, v_k^i) \tag{2}$$

Après la prédiction, l'étape de correction est réalisée, dans laquelle les mesures réelles obtenues par l'odomètre sont comparées aux états prévus dans l'étape de prédiction. Ainsi, chaque particule é attribuée à un poids en fonction de l'adéquation entre son état prédit et les mesures réelles. De cette manière, les particules dont l'état correspond mieux à la mesure réelle se voient attribuer un poids plus élevé. Le calcul du poids de chaque particule (Équation (3)) présente le terme d'innovation qui met à jour sa valeur en fonction de la comparaison entre la mesure réelle obtenue et son état estimé.

$$\tilde{w_k^i} = w_{k-1}^i exp(-\frac{1}{2}(y_k - h(x_k^i))^T R_k^{-1}(y_k - h(x_k^i))) \tag{3}$$

Il convient de noter qu'il est important que les poids obtenus soient normalisés, puisque leur somme doit être égale à 1. Ainsi, les particules ayant les poids les plus élevés représentent les états ayant la plus grande probabilité d'être l'état réel du robot. Une fois les particules associées à leurs poids respectifs, l'estimation de l'état actuel du robot est calculée à partir de la moyenne pondérée des particules.

Le re-échantillonnage est ensuite effectué. À ce stade, les particules sont re-échantillonnées de manière probabiliste avec remplacement sur la base de leurs poids. De cette manière, les particules qui correspondent bien aux mesures se voient attribuer un poids plus élevé, tandis que les particules qui correspondent mal aux mesures se voient attribuer un poids plus faible. L'algorithme de ré-échantillonnage implémenté a pour but de déterminer les indices des premières particules dont la somme cumulée des poids est supérieure à chaque poids. Cela signifie que les particules dont la somme cumulée des poids est supérieure aux poids des particules ont tendance à être dupliquées, tandis que celles dont la somme cumulée des poids est inférieure aux poids des particules ont une probabilité plus faible d'être sélectionnées et peuvent donc être supprimées.

Il convient de noter que l'échantillonnage est effectué si l'efficacité du filtre est inférieure à un certain seuil :

$$N_{eff} = \frac{1}{\sum_i (w_k^i)^2} < \theta_{eff} N \tag{4}$$

La première partie du code est l'implémentation de la classe appelée **Simulation** qui simule le mouvement du robot. Dans cette classe, les paramètres de simulation sont initialisés, tels que le temps de simulation (**Tf**), l'intervalle de temps entre deux prédictions (**dt_pred**), l'intervalle de temps entre deux mesures (**dt_meas**), la position réelle du robot (**xTrue**), la matrice de covariance de l'erreur de processus et de l'erreur de mesure (**QTrue** et **RTrue**), la position odométrique (**xOdom**) et la carte des amers (**Map**). En outre, cette classe possède des méthodes pour obtenir le contrôle réel du robot à l'instant k (*get_robot_control*), simuler le monde avec la position réelle du robot à l'instant k (*simulate_world*), obtenir les mesures d'odométrie bruitées u_k (*get_odometry*) et mesurer l'état bruité du système z_k (*get_observation*).

L'autre partie du code consiste à implémenter les fonctions du filtre. La fonction *motion_modèle* calcule la prédiction de l'état suivant du robot en utilisant la fonction $f(\cdot)$ (2). La fonction *observe_model* calcule l'état du système z_k à partir de l'état du robot **xVeh** et d'un point de référence. Le point de référence est choisi au hasard à chaque étape k de la matrice **Map**. L'indice du point de référence actuel est **iFeature**. L'ensemble des particules est également initialisé et chaque particule est une variable aléatoire qui suit une distribution gaussienne.

La fonction *re_sampling*, à son tour, est responsable du ré-échantillonnage, en donnant plus de poids aux particules qui correspondent le mieux aux mesures du capteur.

En outre, certaines fonctions sont implémentées, telles que *angle_wrap*, qui ajuste l'angle entre $-\pi$ et π, et *tcomp*, qui effectue deux transformations représentant le modèle mathématique $f(\cdot)$. La fonction *plotParticles* trace la carte, les particules, la trajectoire réelle du robot, la trajectoire odométrique et l'estimation du filtre à particules. Elle trace également les courbes d'erreur (en bleu) et de covariance (en rouge).

La dernière partie du code est constituée du programme principal (main). Dans cette partie, divers paramètres sont initialisés, tels que les temps de simulation, l'estimation initiale de l'état du système $\hat{x}_0$, l'incertitude associée à l'estimation initiale de l'état $\hat{P}_0$, la carte des points de référence **Map** et les matrices de covariance **QEst** et **REst**. L'état est représenté par la position sur l'axe x, l'axe y et l'orientation θ.

Le programme principal lance le cycle de simulation, dans lequel, à chaque itération de la boucle, les étapes du filtre sont exécutées. Dans la partie prédiction, l'algorithme simule le mouvement du robot (*simulate_world*) et obtient les mesures d'odométrie (*get_odometry*).

Ensuite, les particules sont mises à jour par les prédictions faites par le modèle de l'équation (2) (motion_model), dans lequel un bruit (v_k) est ajouté aux mesures obtenues par l'odomètre. Ainsi, pour chaque prédiction, son poids respectif (wp[p]) est calculé avec le terme d'innovation qui corrige l'erreur de la prédiction par rapport à la mesure réelle. Après normalisation des poids, l'état prédit est calculé en utilisant la moyenne pondérée des particules et l'écart type des particules est calculé.

Après la prédiction et la correction de l'état, le ré-échantillonnage est effectué si l'efficacité du filtre à particules (N_{eff}) est inférieure à un certain seuil (N_{th}). En outre, le code conserve un historique des poses estimées et réelles du robot, des erreurs et des covariances afin de

fournir une visualisation de la trajectoire du robot, de l'odométrie, de la pose estimée et des graphiques d'erreur.

En résumé, ce code est une simulation complète de localisation basée sur le filtre particulaire (PF) pour un robot mobile opérant dans un environnement 2D. Il fournit un retour visuel sur la précision de l'estimation de la pose du robot par le filtre PF au fil du temps. Lorsque on exécute le code, il génère des graphiques montrant la trajectoire du robot, la trajectoire estimée et l'erreur.

Exercice 2

Comme le montre la Figure 1, le code trace l'environnement de simulation 2D avec les particules (points noirs), la trajectoire réelle du robot (True), la trajectoire estimée du robot basée sur les prédictions du modèle odométrique (odom) et la trajectoire estimée du robot basée sur le filtre particulaire (Part. Filt.).

En outre, trois graphiques sont tracés pour montrer les erreurs dans l'estimation de l'état en fonction du temps pour les coordonnées x, y et l'angle θ. La courbe bleue montre l'erreur d'estimation de l'état du robot par rapport à la position réelle du robot. Ainsi, lorsque l'erreur est faible, l'estimation est proche de la position réelle du robot et, par conséquent, le filtre particulaire estime correctement la position réelle du robot. Les courbes rouges, à son tour, représentent les covariance des particules de l'estimation de l'état et montrent à quel point l'estimation peut être dispersée par rapport à la position réelle du robot.

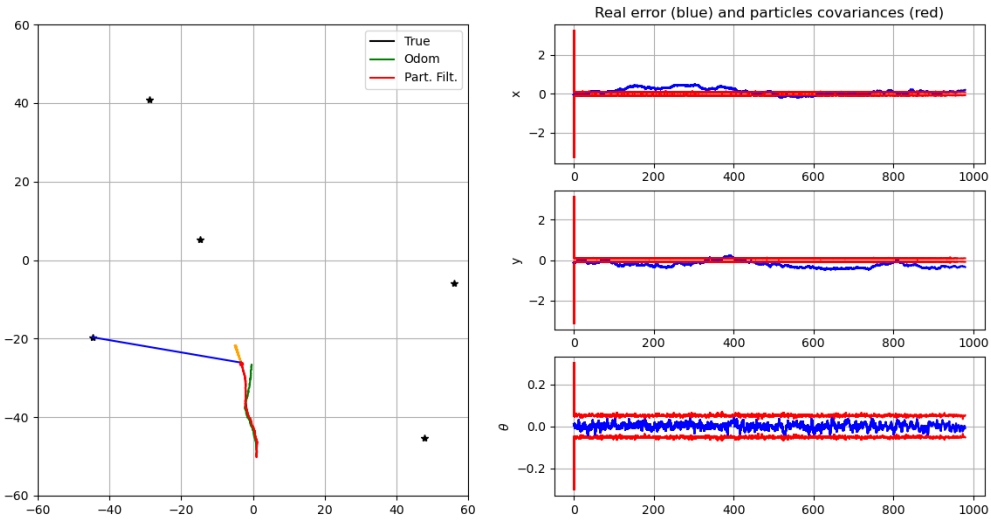


FIGURE 1 – Trajectoire estimée par l'odométrie (odom) et par le filtre particulaire (Part. Filt.). Erreurs dans l'estimation de l'état en fonction du temps pour les coordonnées x, y et l'angle θ.

Comme le montre la Figure 1, la trajectoire estimée du robot basée sur le filtre particulaire (Part. Filt.) parvient à suivre correctement la trajectoire réelle du robot (True), ce qui témoigne de la bonne fonctionnalité du filtre particulaire implémenté. Ceci est également visible dans les graphiques d'erreur, où l'erreur dans l'estimation de l'état du robot par rapport à la position réelle du robot (courbe bleue) est faible.

En outre, au fil du temps, la trajectoire estimée par l'odométrie (odom) commence à s'écarter de la trajectoire réelle du robot (True), ce qui indique que le modèle odométrique ne suit plus avec précision le mouvement réel du robot et que le filtre particulaire doit corriger pour que sa prédiction suive correctement la trajectoire réelle.

Exercice 3

Comme la matrice QEst représente la covariance de l'erreur de processus, c'est-à-dire le bruit dynamique du filtre, sa variation affectera l'estimation du filtre particulaire. En diminuant les valeurs présentes dans QEst, le modèle d'odométrie est considéré comme plus précis et, par conséquent, le processus a moins de bruit. De cette façon, l'estimation a tendance à suivre de plus près les prédictions du modèle et, par conséquent, l'incertitude de l'estimation est plus faible. Cependant, si l'on réduit trop les valeurs de QEst, le filtre attribue une faible incertitude au modèle d'odométrie et s'appuie fortement sur les prédictions générées par le modèle d'odométrie pour estimer l'état du robot. De cette façon, le filtre particulaire sera plus sensible aux erreurs du modèle d'odométrie, comme le montre la Figure 2.

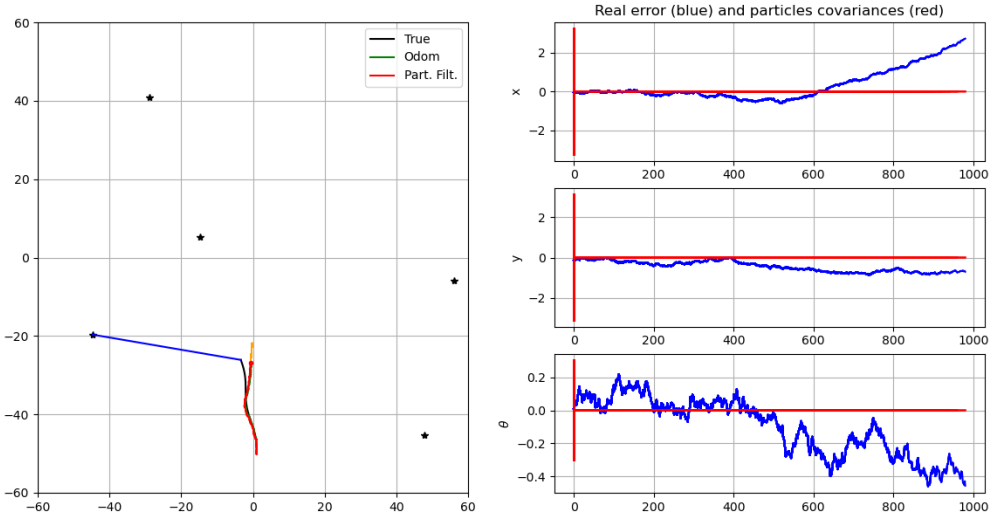


FIGURE 2 – Plot du code avec QEst 10^7 fois plus petit que celui indiqué dans la Figure 1

Comme le confirme la Figure 2, en réduisant considérablement le bruit dynamique, le filtre commence à faire beaucoup confiance aux prédictions générées par le odométrie et, ainsi, la trajectoire prédite par le filtre particulaire commence à se rapprocher de celle prédite par le odométrie (odom). De cette façon, le filtre particulaire devient plus sensible aux erreurs, ce qui augmente considérablement l'erreur d'estimation de l'état du robot par rapport à la position réelle.

En revanche, en augmentant le bruit dynamique du filtre QEst, le modèle d'odométrie est considéré comme moins précis par le filtre et, ainsi, les mesures ont une influence plus significative sur l'estimation. De cette façon, toute mesure inexacte ou bruyante aura un impact direct sur l'estimation, conduisant à une plus grande incertitude, comme le montre la Figure 3.

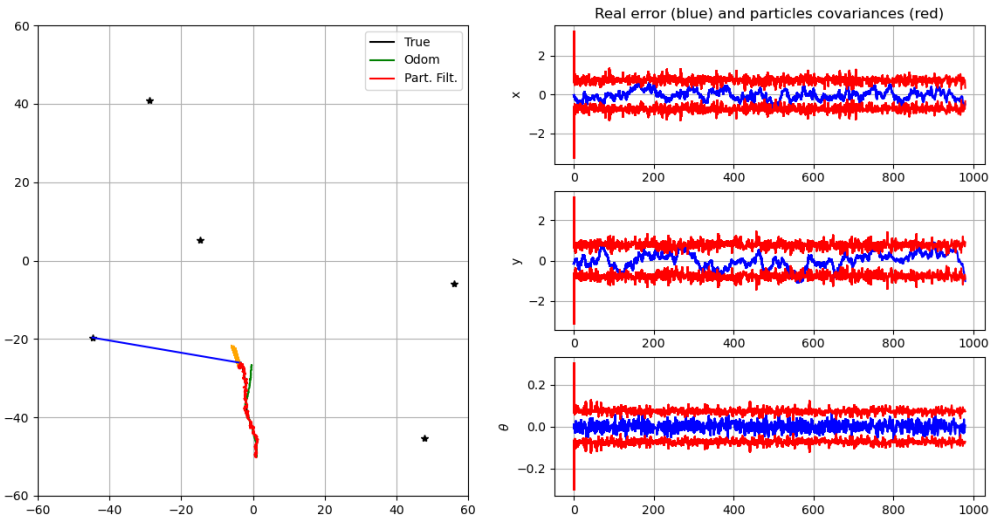


FIGURE 3 – Plot du code avec QEst 100 fois plus grand que celui indiqué dans la Figure 1

En bref, en augmentant fortement les valeurs de QEst, les mesures bruitées ont une grande influence sur le filtre, augmentant l'erreur moyenne et sa variance. D'autre part, si les valeurs QEst diminuent trop, le modèle de l'odomètre aura une grande influence sur le filtre et, par conséquent, son estimation sera basée sur l'odomètre, s'éloignant plus de la trajectoire réelle, ce qui augmente considérablement l'erreur moyenne et sa variance. Le Tableau 1 montre cette augmentation de l'erreur lorsque le QEst diminue et surtout lorsqu'il augmente.

QEst	Erreur moyenne	Variance
2 x QTrue	0.316	0.014
2e-7 x QTrue	0.399	0.043
200 x QTrue	0.822	0.052

TABLE 1 – Moyenne et variance des erreurs pour différentes valeurs de QEst.

Exercice 4

Alors que QEst représente l'incertitude associée au modèle d'odométrie, REst représente l'erreur des mesures du filtre. Par conséquent, si REst est très grand, le filtre attribue de grandes incertitudes aux mesures, ce qui donne plus de confiance dans la prédiction du modèle d'odométrie. De cette façon, l'estimation du filtre aura tendance à suivre de plus près les prédictions du modèle d'odométrie. Ce phénomène est visible sur la Figure 4.

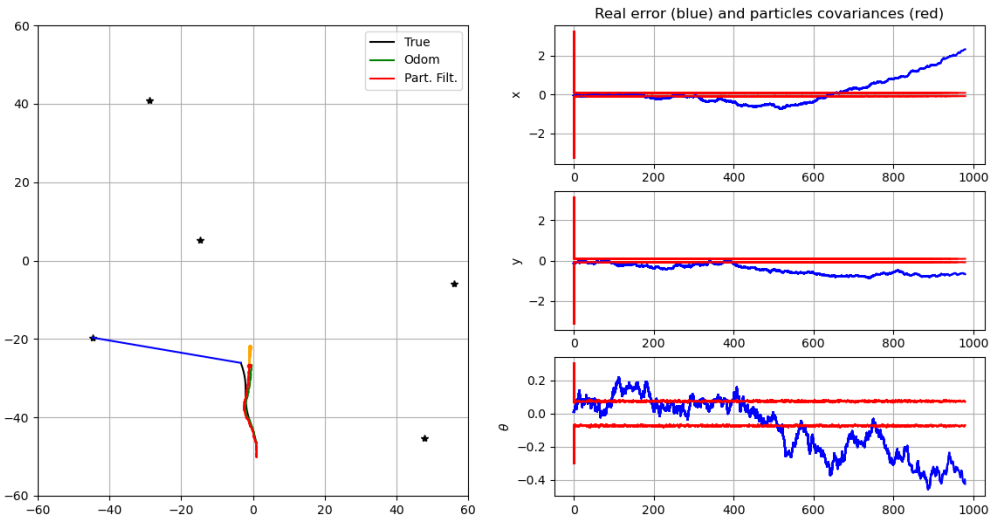


FIGURE 4 – Plot du code avec REst 10^7 fois plus grand que celui indiqué dans la Figure 1.

Comme le montre la Figure 4, avec une erreur de mesure très importante, la prédiction du filtre particulaire est plus proche de celle mesurée par le modèle d'odométrie. Par conséquent, le filtre devient plus sensible au bruit présent dans le modèle d'odométrie, ce qui présente une plus grande incertitude dans l'estimation.

En revanche, si l'erreur dans les mesures du filtre est très faible, le filtre accorde une grande confiance aux mesures et tient moins compte des prédictions du modèle odométrique. De cette manière, l'estimation du filtre particulaire tend à suivre plus mesures que les prédictions de l'odométrie, ce qui rend le filtre plus sensible aux mesures bruitées. Comme la trajectoire de l'odométrie est déjà très proche de la trajectoire réelle, l'erreur et l'incertitude de l'estimation du filtre diminuent. Cet effet est visible dans la Figure 5.

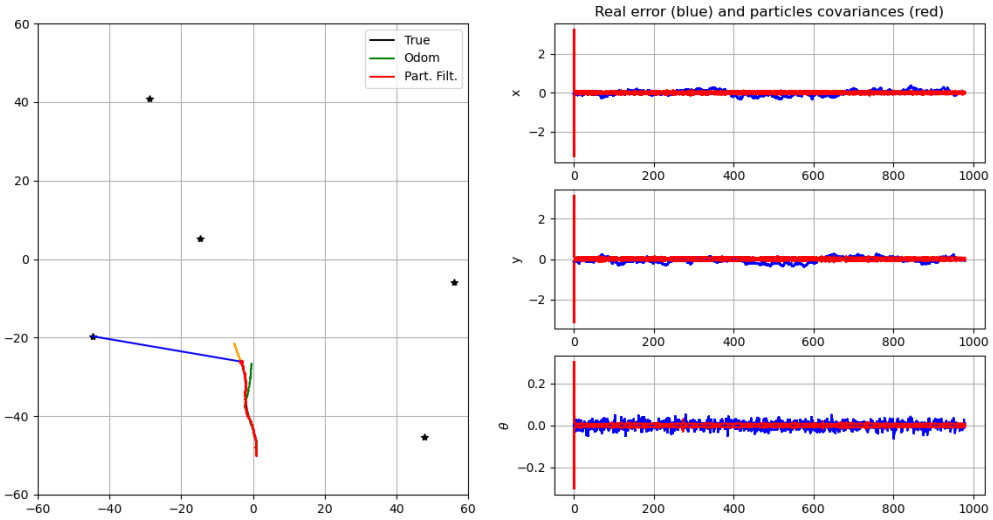


FIGURE 5 – Plot du code avec REst 100 fois plus petit que celui indiqué dans la Figure 1.

En bref, en augmentant fortement les valeurs de REst, les mesures bruitées ont une grande influence sur le filtre, augmentant l'erreur moyenne et sa variance. D'autre part, si les valeurs REst diminuent trop, ce qui diminue l'erreur moyenne et sa variance. Le Tableau 2 montre les valeurs de l'erreur moyenne et la variance pour chaque REst.

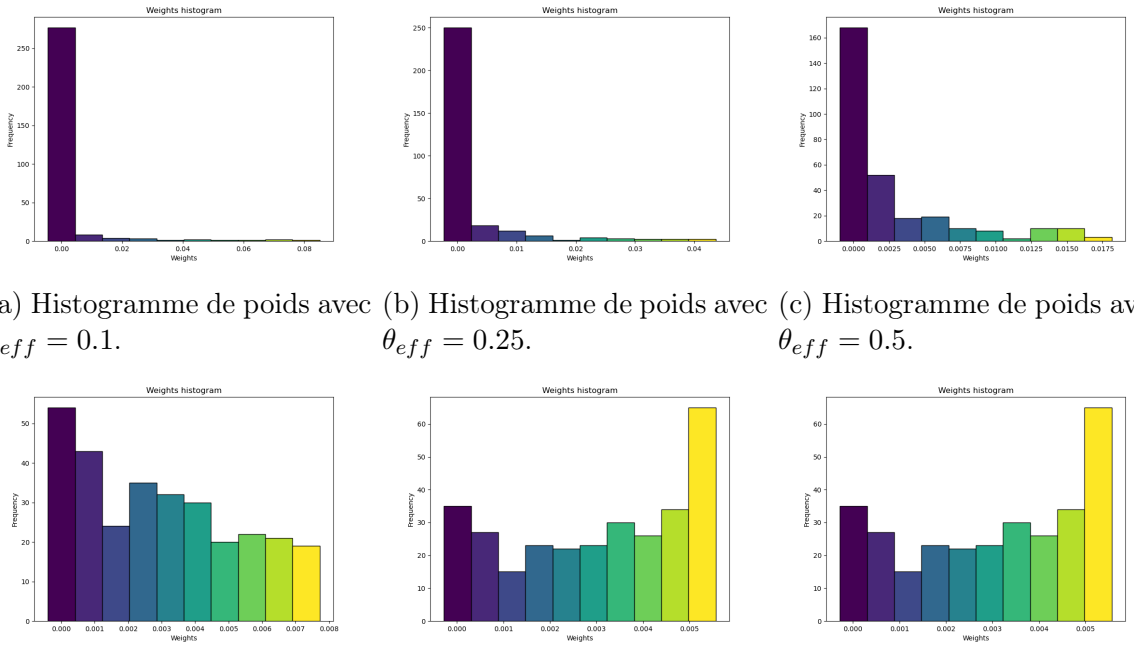
REst	Erreur moyenne	Variance
2 x RTrue	0.316	0.014
0.02 x RTrue	0.185	0.007
2e7 x RTrue	0.762	0.346

TABLE 2 – Moyenne et variance des erreurs pour différentes valeurs de REst.

Exercice 5

Le phénomène de dégénérescence se produit lorsque la plupart des poids des particules tendent à s'approcher de zéro, tandis qu'une petite partie des poids reste non nulle. Ce phénomène est observé après l'étape de ré-échantillonnage des particules. Par conséquent, le ré-échantillonnage est particulièrement souhaitable lorsque les poids sont uniformément répartis, ce qui indique que les particules sont bien distribuées.

Par conséquent, en faisant varier le seuil de ré-échantillonnage (*theta_ eff*), il est possible de déterminer dans quelle distribution de points le ré-échantillonnage est effectué. La Figure 6 montre les histogrammes de poids avant le premier ré-échantillonnage pour différentes valeurs de *theta_ eff*.



(a) Histogramme de poids avec $\theta_{eff} = 0.1$.
(b) Histogramme de poids avec $\theta_{eff} = 0.25$.
(c) Histogramme de poids avec $\theta_{eff} = 0.5$.
(d) Histogramme de poids avec $\theta_{eff} = 0.75$.
(e) Histogramme de poids avec $\theta_{eff} = 0.8$.
(f) Histogramme de poids avec $\theta_{eff} = 0.99$.

FIGURE 6 – Histogrammes de poids pour différentes valeurs de *theta_ eff*.

Comme le montre la Figure 6, lorsque on diminue la limite de ré-échantillonnage, la distribution des poids présente déjà un phénomène de dégénérescence, puisqu'il n'y a pas eu assez d'itérations pour que les poids soient uniformément distribués et que les particules soient plus distribuées, avec un plus grand nombre de poids tendant vers 0. Si l'on procède à un ré-échantillonnage dans ce cas, la régénération augmentera plus.

En augmentant la limite de ré-échantillonnage, la distribution des poids à ré-échantillonner est plus uniforme, ce qui montre qu'il y a déjà eu suffisamment d'itérations pour que les particules se répartissent. Cependant, si le seuil est trop élevé, le ré-échantillonnage n'est effectué que lorsqu'un grand nombre de particules ont un poids élevé. Dans ce cas, le ré-échantillonnage est tardif et la limite de ré-échantillonnage doit être abaissée pour qu'il ait lieu dans le cas idéal, où les valeurs de poids sont plus uniformément réparties.

Exercice 6

Dans l'intervalle entre $t = 250s$ et $t = 350s$, aucun amers n'est observé. Le filtre ne dispose donc d'aucune mesure et ses prédictions sont basées uniquement sur les prédictions de l'odomètre. De cette façon, dans l'intervalle de temps où le filtre se base uniquement sur les prédictions du modèle odométrique, la trajectoire prédite par le filtre de Kalman se rapproche de la trajectoire prédite par l'odomètre, puisqu'il n'y a pas d'étape de correction de la prédiction faite par le modèle odométrique.

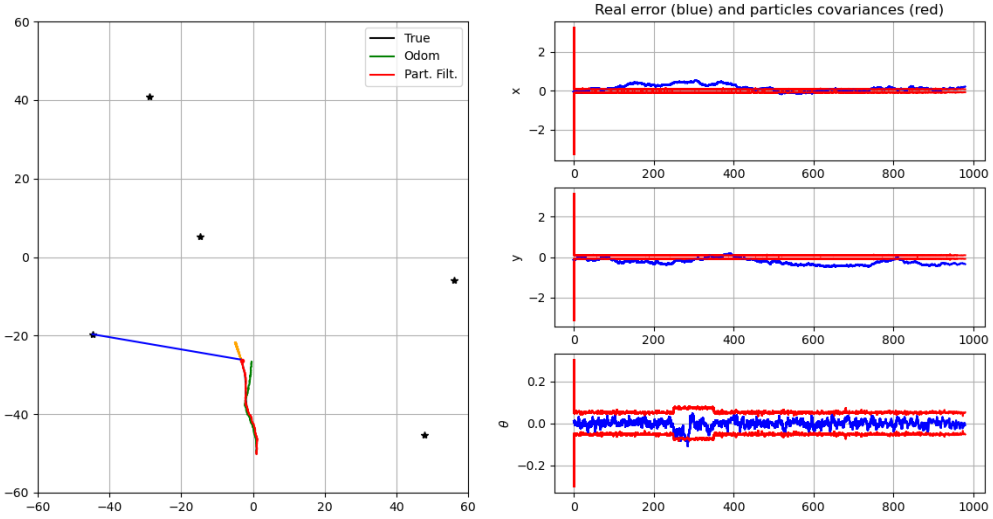


FIGURE 7 – Plot du code avec un trou de mesures entre $t = 250s$ et $t = 350s$.

Comme le montre la Figure 7, la trajectoire prédite par l'odomètre est très proche de la trajectoire réelle entre $t = 250s$ et $t = 350s$. Par conséquent, comme le filtre est basé uniquement sur l'estimation du odomètre et que celle-ci est déjà très proche de la trajectoire réelle, il n'y a pas d'augmentation significative de l'erreur d'estimation du filtre dans cet intervalle.

Cependant, l'absence de mesures entre $t = 250s$ et $t = 350s$ se traduit par une légère augmentation de la covariance de l'angle θ dans cet intervalle. En outre, l'erreur moyenne de l'estimation du filtre montre également une légère augmentation, en raison de ce trou de mesures, comme le montre le Tableau 3.

	Erreur moyenne
Sans trou	0.316
Avec trou	0.333

TABLE 3 – Erreur moyenne de l'estimation du filtre sans et avec un trou de mesures entre $t = 250s$ et $t = 350s$.

Exercice 7

La Figure 1 montre le fonctionnement du filtre avec $dt\_meas = 1$. Lorsque on diminue la fréquence des mesures à 0.1 Hz, le temps des mesures augmente à $dt\_meas = 10$. De cette façon, le filtre particulaire reçoit moins d'informations pour corriger l'estimation de l'état entre deux mesures consécutives. Par conséquent, l'erreur dans l'estimation augmentent, car le filtre dispose de moins d'informations pour réduire cette erreur.

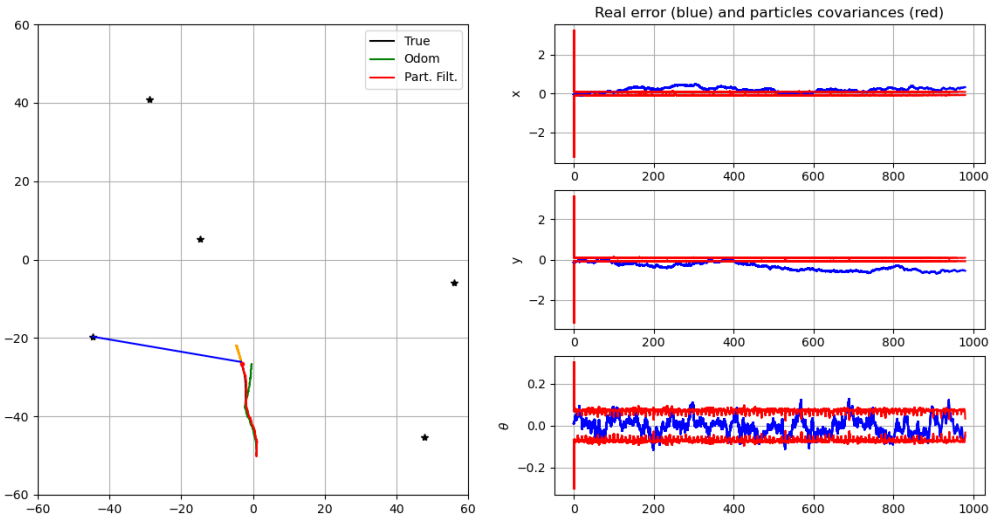


FIGURE 8 – Plot du code avec $dt\_meas = 10$.

En comparant la Figure 1 avec la Figure 8, on peut noter qu'en augmentant le intervalle de temps entre deux mesures dt_meas, la covariance et l'erreur dans l'estimation augmentent. On peut vérifier cette augmentation de l'erreur et de l'incertitude de l'estimation en comparant la moyenne et la variance de l'erreur de l'estimation du filtre particulier pour $t = 1s$ et $t = 10s$ (Tableau 4).

dt_meas	Erreur moyenne	Variance
1 s	0.316	0.014
10 s	0.438	0.026

TABLE 4 – Erreur moyenne et incertitude de l'estimation du filtre pour $dt_meas = 1$s et pour $dt_meas = 10$s.

Exercice 8

En utilisant la Figure 1 comme point de référence, laquelle présente l'utilisation de 5 amers, il est visible que lorsque le nombre de points est réduit à seulement 2 (Figure 9), la trajectoire prédite par le filtre reste similaire à celle avec 5 points. Par contre, la précision de la mesure de l'état est compromise. Cette réduction du nombre d'amers entraîne une incertitude, ce qui, à son tour, a un impact négatif sur les performances du filtre.

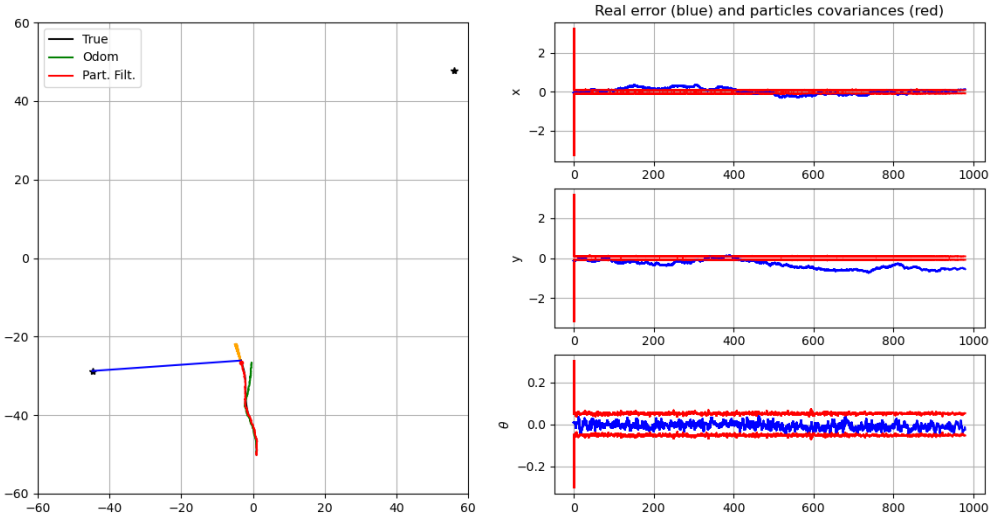


FIGURE 9 – Plot du code avec un nombre d'amers égal à 2.

En revanche, en augmentant considérablement le nombre de amers à 50 (Figure 10), la trajectoire prédite par le filtre reste similaire à celle avec 5 amers, puisque dans les deux cas il y a suffisamment d'amers pour que la mesure de l'état soit fiable. Ainsi, dans les deux cas, la performance du filtre est similaire.

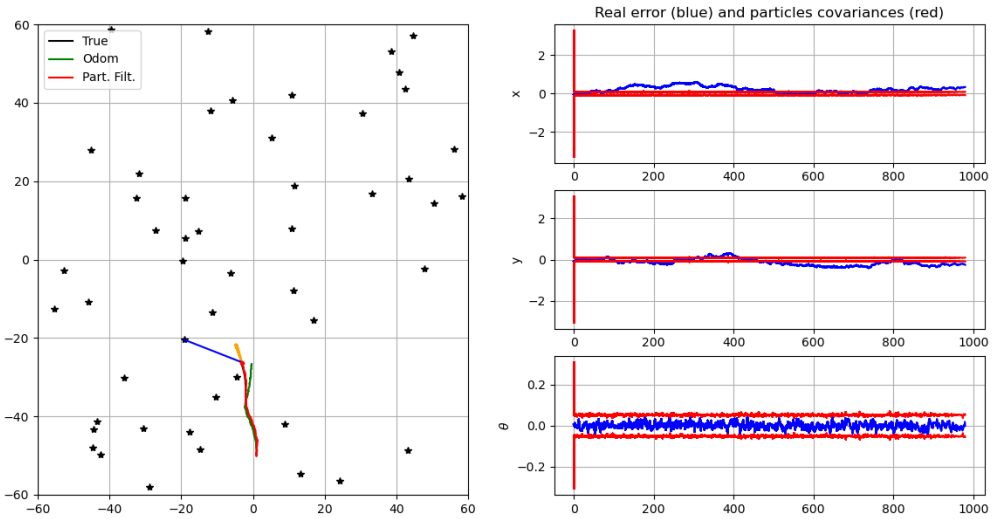


FIGURE 10 – Plot du code avec un nombre d'amers égal à 50.

On peut donc conclure que la variation du nombre d'amers n'affecte pas significativement les performances du filtre. Un facteur qui a un impact réel sur les performances du filtre est

la distribution des amers, et pas nécessairement leur nombre. Cependant, il est intéressant de noter qu'avec plus d'amers, l'erreur et l'incertitude des prédictions du filtre sont plus faibles. Le Tableau 5 montre que dans le cas de 50 points, la moyenne et la variance des erreurs sont plus faibles que dans le cas de 5 points.

Nombre d'amers	Erreur moyenne	Variance
2	0.379	0.032
5	0.316	0.014
50	0.318	0.015

TABLE 5 – Erreur moyenne et incertitude de l'estimation du filtre pour 2, 5 e 50 amers.

Exercice 9

La méthode implémentée est le ré-échantillonnage résiduel. Ce code consiste à calculer les résidus en divisant chaque poids par le poids moyen et en arrondissant le résultat à l'entier le plus proche. Ces résidus représentent le nombre de fois qu'une particule doit être répliquée. La fraction des particules qui doivent encore être répliquées est calculée en soustrayant les résidus arrondis des résidus originaux. La somme cumulée de ces fractions est également calculée.

Avec la méthode de ré-échantillonnage multinomial, on détermine quelles particules doivent être répliquées. Elle recherche la première particule dont la somme cumulée des résidus est supérieure à la valeur échantillonnée.

Comme la méthode implémentée précédemment (ré-échantillonnage multinominal), le ré-échantillonnage résiduel donne également de bons résultats dans la prédiction de la trajectoire du robot, comme le montre la Figure 11.

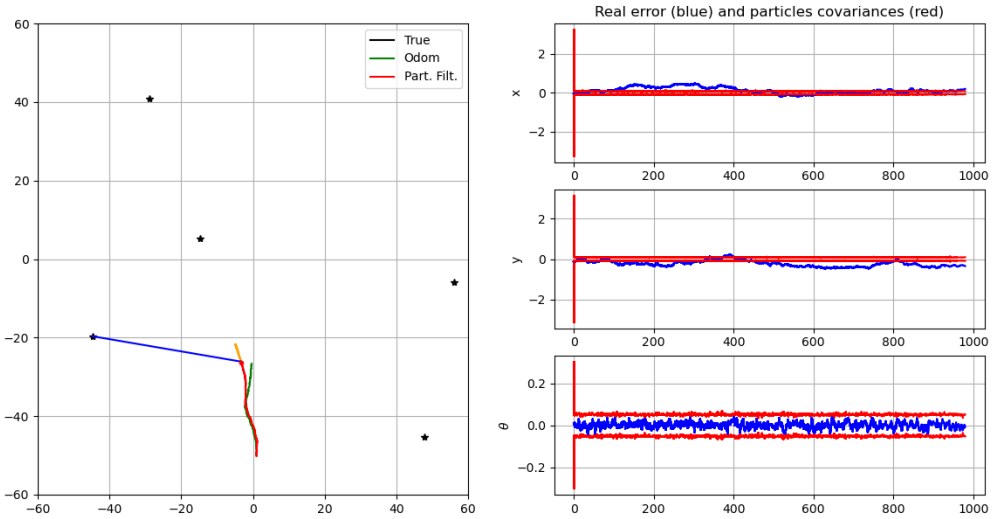


FIGURE 11 – Filtre particulaire avec ré-échantillonnage résiduel.