Introduction à la robotique mobile
Travail pratique 2, Version 1.0
Date de remise : 1er décembre 2024, 23h59.

Instructeur : Philippe Giguère

17 novembre 2024

Instructions générales

Ce travail pratique se fait en équipe de 1 à 2. Dans l'évaluation du rapport, nous pourrons retirer jusqu'à 10 points sur la qualité de la présentation, notamment :

- la qualité du français ;
- la propreté de la présentation ;
- le degré d'élaboration et la présentation structurée des réponses : évitez donc de simplement donner la réponse sans un minimum de discussion.

N'oubliez pas d'inclure TOUS les scripts/fonctions/notebook Jupyter dans le fichier `.zip`. Vous pouvez faire le TP en matlab ou en python.

Pour les étudiants en GLO-7021, le rapport doit être rédigé obligatoirement en utilisant Latex.

1 Filtre de Kalman linéaire : estimation de la température d'un échantillon (22 pts)

Vous cherchez à estimer la température d'un échantillon avec une sonde électrique S, qui retourne un voltage de $0.03V/K$ (volts par Kelvin), et qui a un bruit avec écart type σ_S =0.05 V. Votre estimé initial de la température est X_0 =340 K, auquel vous associez une incertitude initiale de variance P_0 =6 K^2. L'échantillon baisse de température au rythme de 1.5 K par intervalle de temps (ce qui constituera la commande). De plus, cette chute de température est légèrement bruitée, selon un bruit gaussien d'écart type de C_ν =0.1 K entre chaque intervalle. Les mesures de la sonde S à chaque intervalle de temps (z_1 =9.928 V, z_2 =9.820 V et z_3 =9.817 V) sont reproduites dans le tableau 1. Important ! La chute de température s'effectue avant la mesure, à chaque itération.

1.1 a) (6 pts)

Quelles sont les valeurs des matrices Φ, Γ et Λ pour ce problème ?

1.2 b) Exécution du filtre (16 pts)

Calculez, à la main [1] pour les deux premières itérations les différentes étapes du filtre de Kalman, en complétant le tableau 1. Pas besoin d'inclure les démarches dans le rapport. Pour la troisième ligne, programmez le filtre de Kalman pour calculer les entrées. Utilisez-le d'ailleurs pour vérifiez vos calculs manuels !

Calcul	Iteration	X_{t-1}	P_{t-1}	X_{pred}	P_{pred}	Mesure z	Gain K	X_t	P_t
Manuel	1	340	6			9.928			
Manuel	2					9.820			
Programme	3					9.817			

Table 1 – Tableau à compléter pour l'exercice de Kalman.

Note : comme c'est un filtre récursif, les entrées X_{t-1} et P_{t-1} à l'itération 2 sont les mêmes que les entrées X_t et P_t à l'itération 1. Et ainsi de suite pour l'itération 3, etc.

Note : vous pourrez réutiliser le code de ce filtre pour la question à la section 2.

1. Pour vous pratiquer, d'un coup je pose une question de ce genre à l'examen...

2 Aladdin et son tapis volant magique ! (38 pts) pour GLO-4001, (50 pts) pour GLO-7021

Aladdin décide d'appliquer les principes du filtre de Kalman [2] ! Son tapis magique possède la propriété qu'il se déplace à une vitesse constante, sans aucune friction ni perturbation, après une impulsion au démarrage. Cependant, Aladdin n'a pas encore compris comment le commander, au delà de piaffer au départ pour se donner un élan. Aladdin se situe, au repos, à côté d'une borne kilométrique marquant le kilomètre 0. L'écart-type de position associé à cette borne est de $\sigma_x = 2\ m$. L'écart-type sur la vitesse de son tapis est de $\sigma_{V0} = 0.2\ m/s$. L'état décrivant le tapis sera :

$$X = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}, \tag{1}$$

où x sera en m et $\dot{x}$ en m/s. **Vous devez tenir compte de ces unités dans vos réponses aux questions**, car les mesures, comme vous le verrez plus loin, seront dans des unités différentes (km et km/h).

2.1 Matrice Φ (2 pts)

Quelle sera la matrice Φ pour décrire le comportement dynamique de ce système ? Pour votre information, la matrice de commande Γ sera nulle : l'impulsion de départ sera tenue en compte dans la section 2.3.

2.2 Matrice $P(0)$ (2 pts)

Quelle sera la matrice de covariance $P(0)$ initialisant le système ?

2.3 Et c'est un départ !

Aladdin piaffe au sol et son tapis volant se met à se déplacer sous l'impulsion. Il estime sa vitesse à environ 1.5 m/s, et son point de départ à 0 m, puisqu'il est à côté de la borne kilométrique 0. Pour son filtre, l'intervalle de temps entre les itérations sera $\Delta T = 0.5\ s$. Comme il ne prend pas de mesure au début, son <u>filtre n'exécute que les deux premières équations à chaque itération</u>, c'est-à-dire celles qui représentent le modèle de déplacement, incluant le bruit.

2.3.1 Impact sur la position de l'erreur en vitesse (4 pts)

Comme vu en classe [3], vous devriez voir qu'à mesure que le filtre itère sur les deux premières équations [4], l'erreur sur l'estimé de vitesse $\dot{x}$ aura un impact grandissant sur l'incertitude de l'estimé en position x. Quel élément de la matrice P, entre σ_x^2, $\sigma_{x\dot{x}}$ et $\sigma_{\dot{x}}^2$, viendra capturer cette dépendance entre l'incertitude en vitesse et en position ? De plus, tracez l'évolution de ces trois valeurs σ_x^2, $\sigma_{x\dot{x}}$ et $\sigma_{\dot{x}}^2$, en fonction des itérations, jusqu'à l'itération 100.

---

2. Bon, on ne s'offusquera pas du fait que Kalman et Aladdin n'était pas des contemporains, mais la mode est aux mélanges d'univers (crossovers) dans les films de superhéros. Nous ne ferons pas exception.

3. Diapositive 71 intitulée *Prédiction sur P : véhicule accéléré* dans `07-EstimationEtat.A2024-I.pdf`,

4. Car nous n'avons pas de mesures, donc pas de mise-à-jour.

2.3.2 Mesure (6 pts) ou (10 pts)

Aladdin passe à côté d'une pancarte indiquant la distance de $z_d = 0.2\ km$, à l'itération 243. La précision de sa localisation par rapport à cette pancarte est de $\sigma_d = 5\ m$.

a) **(2 pts)** Quelle seront les matrices Λ et C_w à cet instant ? Attention ! N'oubliez-pas que la mesure est en km, et que par conséquent votre matrice Λ ainsi que la matrice de bruit C_w devront être exprimées en tenant compte de ces unités. Sinon, vos résultats seront incorrect. Autrement dit, $z_d = 0.2$ et non pas $z_d = 200$.

b) **(4 pts)** Estimez la vitesse V manuellement (en m/s), en tenant compte du temps écoulé entre les deux pancartes. Faites l'hypothèse que le temps est exact, c'est-à-dire n'est pas une variable aléatoire.

c) **(4 pts)** (GLO-7021 seulement) Estimez la variance σ_V^2 de votre estimé de vitesse, en tenant compte des incertitudes reliées aux deux pancartes. Indice ! Vous devez faire intervenir la règle de variance $\mathbb{V}\{aX + bY\}$.

2.3.3 Mise-à-jour avec mesure de position (9 pts) ou (17 pts)

Faites la mise-à-jour, c'est-à-dire exécutez les équations 3 à 7 du filtre de Kalman.

a) **(4 pts)** Comparez l'incertitude de la position estimée par Kalman (entrée σ_x^2 de la matrice P) et la variance σ_d^2 de la position de la deuxième pancarte, après cette mise-à-jour. Que constatez-vous ? (Ceci correspond à la situation où la mesure z est beaucoup plus précise que l'estimé $x(k+1|k)$ provenant de la prédiction).

b) **(2 pts)** Quelle est la valeur de la vitesse $\dot{x}$ estimée par Kalman (c'est-à-dire la deuxième entrée du vecteur X ? Cette valeur devrait être très proche de votre valeur V estimée à la question 2.3.2 b).

c) **(4 pts)** (GLO-7021 seulement) Quelle est la valeur de l'incertitude sur la vitesse $\dot{x}$, telle qu'estimée par le filtre de Kalman (entrée $\sigma_{\dot{x}}^2$ de la matrice P) ? Cette valeur devrait être légèrement inférieure à σ_V^2, que vous avez calculé en 2.3.2 c). Pourquoi ? (Ici l'explication doit être de haut niveau : je ne cherche pas une preuve mathématique. Indice : quelles sont toutes les évidences utilisées par votre approche manuelle vs. celles utilisées par le filtre de Kalman).

d) **(4 pts)** (GLO-7021 seulement) Que devriez-vous faire pour que votre estimé V concorde avec celui de Kalman ? Montrez le détails du calcul. (Indice : fusionner toutes les évidences disponibles sur la vitesse). Notez que bien que je ne le demande pas, vous devriez aussi être capable de faire correspondre les deux variances σ_V^2 et $\sigma_{\dot{x}}^2$ si vous appliquiez ce même concept.

e) **(3 pts)** Repartez votre script afin de tracer les trois valeurs de la matrice P, jusqu'à l'itération 300. N'oubliez pas de faire la mise-à-jour à l'itération 243. Notez que c'est la seule mise-à-jour que vous faites dans ce script. Commentez et expliquer ce qui se passe pour chacune des trois courbes.

2.3.4 Mise-à-jour avec mesure de position, partie II (7 pts)

Vous venez de voir un exemple de mesure incomplète, où une mesure en position x vient modifier la variable d'état de vitesse $\dot{x}$. Dans les équations de mise-à-jour, la seule matrice qui contient l'information pour faire cela se situe dans la matrice P. Quel élément de la matrice P, (σ_x^2, $\sigma_{x\dot{x}}$ ou $\sigma_{\dot{x}}^2$) sera responsable ? Pour valider votre réponse, mettez directement ce terme à zéro lors de l'itération correspondant à la mise-à-jour. N'oubliez pas qu'il y a deux entrées $\sigma_{x\dot{x}}$ dans P, si vous choisissez de mettre ce terme à zéro). Faites tourner le code de mise-à-jour avec cette modification à P. Rapporter et discutez :

— la position x et la vitesse $\dot{x}$, par rapport à la réponse à 2.3.3.
— la covariance $\sigma_{\dot{x}}^2$ après la mise-à-jour.

2.4 Nouveau problème ! Mesure de vitesse au lieu de position (8 pts)

Reprenez ce problème, mais à l'itération 243, au lieu d'une pancarte indiquant la distance 0.2 km, ce sera plutôt une borne radar intelligente. La vitesse affichée sur cette borne est $z_R = 5\ km/h$ lors du passage d'Aladdin et son tapis.

a) **(2 pts)** Quelle sera la matrice de mesure Λ associée avec cette situation ?

b) **(2 pts)** Faites la mise-à-jour avec cette mesure. Quel est l'effet sur la position x et la vitesse $\dot{x}$, en utilisant un bruit de mesure de $\sigma_R = 0.01\ km/h$?

c) **(2 pts)** Comparez la covariance σ_x^2 entre la nouvelle matrice P et la matrice $P(0)$ utilisées pour initialiser le filtre. Que constatez-vous ? Qu'est-ce que cela signifie ? Comme à la question 2.3.3, tracez les trois valeurs de la matrice P, jusqu'à l'itération 300, en incluant l'unique mise-à-jour. Commentez et expliquez ce qui se passe pour chacune des courbes.

d) **(2 pts)** Refaite la question c) mais cette fois avec un bruit beaucoup plus grand sur la mesure : $\sigma_R = 0.5\ km/h$. Commentez la différence entre les courbes de éléments de P obtenues en d) par rapport à c). Commentez aussi sur les valeurs de position x et vitesse $\dot{x}$ entre d) et c).

3 Localisation globale par filtre à particules (20 pts) pour GLO-4001, (28 pts) GLO-7021

Vous avez un robot initialement perdu, dans un environnement dont la carte est connue. Pour pouvoir le localiser, vous devrez donc utiliser un filtre à particules, puisque la distribution de croyance sur l'état du robot sera mulitmodale. Un aspect important de la question sera de trouver des bons paramètres, pour permettre au système de se localiser de manière robuste. Notez ici que nous ne chercherons pas à faire une solution qui puisse tourner en temps-réel ; ne vous préoccupez donc pas d'optimiser le code pour sa vitesse d'exécution.

Pour tous les cas, nous allons utiliser un monde décrit par un polygone, contenu dans le fichier `Carte.mat`. Le robot ponctuel (rayon=0) est équipé de 4 LiDARs, pointant dans les directions 0, $\pi/2$, π et 1.5π, tel qu'illustré à la Fig. 1. Le robot est soumis aux bruits gaussiens suivants (avec le nom de la variable entre parenthèses) :

— LiDAR (`Lidar`) : $\sigma_L = 0.01\ m$;
— Vitesse linéaire (`V`) : $\sigma_V = 0.01\ m/s$;
— Vitesse angulaire (`omega`) : $\sigma_\omega = 0.05\ rad/s$;
— Compas magnétique (`Compas`) : $\sigma_C = 0.01\ rad$.

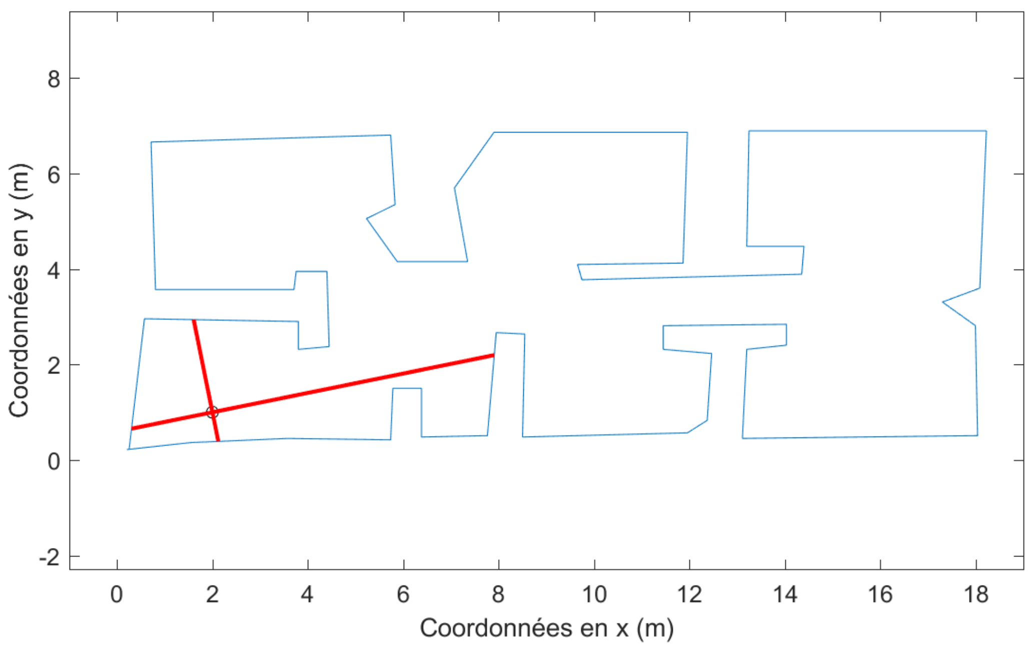

FIGURE 1 – Robot ponctuel dans un environnement décrit par un polygone. Les 4 lignes rouges indiquent les faisceaux laser.

Pour prédire les mesures LiDAR, vous allez utiliser des fonctions géométriques incluses dans la remise de l'énoncé. Le faisceau LiDAR sera simplement un segment de droite partant du centre du robot et allant à une distance maximale (20 m), selon l'angle du LiDAR (et bien sûr l'angle du robot, puisque le LiDAR est fixé sur ce dernier). Trouvez tous les points de cette droite qui interceptent le polygone de la carte, en utilisant la fonction `fast_four_way_raycast` fournie dans le fichier `raycast.py`. Cette fonction simule un LiDAR pour chacune des quatre directions, et retourne les intersections entre les rayons et la carte dans l'ordre suivant : avant, gauche, arrière et droite. La distance mesurée par un rayon LiDAR sera celle correspondante à l'intersection rayon-monde le plus proche du centre du robot. Pour ceux qui ont déjà fait des travaux en graphisme 3D et jeux vidéos, vous reconnaîtrez une procédure de lancée de rayons.

Pour l'initialisation des particules, vous pouvez les distribuer de manière uniforme dans un rectangle couvrant la carte au complet. Ne vous préoccupez pas des particules en dehors du polygone (essentiellement en dehors de l'environnement). Ils finiront par avoir des poids quasi-nuls et seront donc éliminé lors de la phase du resampling.

Nous vous fournissons le code de base `Q3.py` qui permet de charger les données. Pour faire le ré-échantillonage des particules lors de l'appauvrissement, utilisez la fonction `random.choices`. ainsi que de tracer certaines figures.

Pour installer les dépendances, exécutez la commande `pip3 install -r requirements.txt`.

3.1 Cas 1 : l'angle est connu (GLO-4001 : 20 pts, GLO-7021 : 10 pts)

Afin d'y aller en douceur, nous allons commencer par un problème d'estimation à deux dimensions, où l'état est simplement $X = [x\ y]^T$. En effet, la quantité de particules nécessaire est exponentielle en nombre de dimensions. Pour l'angle du robot, vous allez simplement prendre l'angle `Compas` mesuré par le compas magnétique (en rad, de $-\pi$ à π), et ajouter un bruit σ_{angle} aléatoire sur l'angle. Ce bruit permettra de "brasser" les particules un peu, pour permettre une convergence plus rapide, en plus de tenir compte de l'incertitude du compas. Faites un filtre à particules qui exploite les informations suivantes, contenues dans le fichier `Q1Trajectoire.mat` :

- les 4 mesures LiDAR (`Lidar`), en m ;
- la vitesse linéaire (`V`), en m/s ;
- l'angle du compas magnétique (`Compas`), en rad.

Aussi, rappelez-vous qu'un des problèmes pour les filtres à particules est qu'ils ne convergent pas bien si la fonction de capteur est trop précise. Gardez cela en tête, quand viendra le moment de choisir la valeur de σ_{lidar} dans la fonction de vraisemblance $p(z_{lidar}|x_t)$.

Dans votre rapport, discutez des éléments suivants :

- une justification de votre choix σ_{angle} ;
- les divers paramètres utilisés dans le filtre (pour GLO-7021, la question 3.2 vous demande d'élaborer sur certains de ces paramètres, alors ne pas dupliquer) ;
- description qualitative de la distribution des particules au fil du temps.

Incluez deux tracés de trajectoires estimées par votre filtre, l'une pour une exécution réussie du filtre et l'autre pour une exécution qui n'a pas convergée. La trajectoire est basée sur le calcul, à chaque itération, de la moyenne des particules (pondérée par leur poids respectif). Dans le cas où le filtre a échoué, donnez une explication qualitative. Incluez aussi dans vos graphique la trajectoire vérité-terrain (`xPose`,`yPose`,`anglePose`) contenue dans le fichier `Q1Trajectoire.mat`. Bien entendu, vous ne pouvez pas utiliser cette information vérité-terrain dans le filtre à particules, seulement à titre de débuggage et du calcul des erreurs dans le rapport. N'oubliez pas d'appliquer `wrap_to_pi` lorsque vous calculez des erreurs sur les angles, afin de tous rammener entre $-\pi$ et π.

Pour les étudiants de GLO-7021, votre note dépendra de la qualité des explications et justifications. Par exemple, il serait intéressant de voir des graphiques d'erreur en fonction du temps vs. choix de σ_{angle}.

Pour les étudiants en GLO-4001, le TP se termine ici !

3.2 Impact du bruit utilisé dans la vraisemblance LiDAR (GLO-7021 seulement, 6 pts)

Lors de la mise-à-jour, vous devez choisir l'écart-type σ_{LiDAR} pour évaluer la vraisemblance des mesures du LiDAR dans la fonction $p(z|x)$. Quel est l'impact de cette valeur, en terme *a)* du nombre nécessaire de particules pour converger et *b)* de la vitesse de convergence du filtre. Pour*a)*, faites dix essais par nombre de particules, et rapportez le pourcentage de succès. Vous devrez trouver, par essai et erreur, des valeurs "intéressantes" de nombres de particules pour ces tests, i.e. pour lesquels le taux de succès change. Pour aller plus vite, ne laissez pas tourner le filtre sur toutes les itérations : dès que vous voyez que les particules ont condensé en une région, vous pouvez établir si le filtre a divergé ou non par rapport à la vérité-terrain. L'utilisation de la variance `numpy.var()` vous permettra de détecter cette condensation.

Prenez des mesures sur la vitesse d'appauvrissement, afin de voir l'impact de σ_{LiDAR} sur cette vitesse.

Selon vos résultats, en quoi serait-il intéressant d'augmenter ou de diminuer graduellement σ_{LiDAR} au fil du temps ? Si vous comprenez bien cette question, cela vous sera utile pour la question 3.3 pour accélérer le filtrage.

3.3 Cas 2 : l'angle est inconnu (GLO-7021 Seulement, 12 pts)

Pour ce cas-ci, la dimensionalité du problème va augmenter, car il vous faudra maintenant estimer l'état en y incluant l'angle θ du robot : $X = [x\ y\ \theta]^T$. Tel que vu en classe, le nombre de particules est exponentiel en dimension de l'état... Alors vous devrez probablement être patient lors de l'exécution de votre filtre ! Le fichier de donnée pour ce cas d'estimation, `Q2Trajectoire.mat`, contient les commandes de vitesses linéaires et les commandes angulaires, les mesures LiDAR et la vérité-terrain. Notez ici que vous n'avez plus de données du compas. Vous pouvez partir de votre implémentation du filtre à particule de la section 3.1.

Refaites les mêmes expérience qu'à la section 3.2 et commentez sur ces résultats.

À partir de votre expérience de l'impact de σ_{LiDAR} et du nombre de particules sur les chances de convergences, adaptez ces valeurs dynamiquement afin d'accélérer l'exécution du filtre à particules. Expliquez votre stratégies, notamment à l'aide de graphiques montrant le nombre de particules, la valeur de σ_{LiDAR} et les erreur en distance/orientation en fonction du temps. Appuyez ausis votre réponse avec des histogrammes d'erreur correspondant au régime pour lequel le filtre a convergé.

Comment pouvez-vous justifier votre approche, d'un point de vue théorique ?