

# Evaluation : Observateurs et filtre de Kalman

- **Tout document autorisé**, mais **communication** avec d'autres personnes **interdite**. De plus, l'encadrant ne vous aidera pas.
- **Rendre code MATLAB** et autres documents informatiques sur **Moodle** dans un **dossier zip** et éventuellement des documents manuscrits complémentaires si nécessaire. S'assurer que les documents rendus sont bien vos versions finales. Vérifier avec l'encadrant que les fichiers sont bien rendus.
- **Mettre en commentaires** directement dans le code toutes les informations et **explications** nécessaires pour que le correcteur comprenne ce qui est fait.
- **Ne pas rester bloqué si le code ne s'exécute pas** correctement, s'efforcer d'apporter une réponse (éventuellement partielle) à toutes les questions. Inversement, le code peut être incorrect (e.g. ne respecte pas les consignes) même si le résultat qui s'affiche semble bon !
- Eventuellement s'inspirer des codes des TD précédents pour aller plus vite.
- Fournir une **capture d'écran** (appuyer sur la touche Print Screen/Impression Ecran et faire coller dans Microsoft Paint) des divers résultats intermédiaires importants.
- **Faites des sauvegardes régulières**. Surveillez régulièrement si votre PC doit redémarrer. **Attention si vous renommez des dossiers**, un **bug** les fait parfois **disparaître** !
- Barème provisoire : 22 points ramenés sur 20.

## Part 1 : Questions théoriques (durée estimée : 20 min)

1. A quoi servent les étapes de « prédiction » et « correction » dans le filtre de Kalman ? [2 points]
2. Que représentent les différentes matrices de covariance en entrée et en sortie de la fonction kalman utilisée en MATLAB ? [2.5 points]
3. Quel est le principal apport d'un filtre de Kalman par rapport à une simple moyenne entre des mesures de capteurs et des estimations selon un modèle ? [0.5 point]

## Part 2 : Filtre de Kalman et régulation d'un robot bateau à moteur autonome (durée estimée : 130 min)

A l'occasion de Brest 2016 (grand rassemblement de bateaux qui a lieu tous les 4 ans), l'ENSTA Bretagne a présenté un robot bateau à moteur autonome (voir <https://youtu.be/sRBjk0NI2vA>). Celui-ci est muni d'un hydrojet pour sa propulsion (u1) qui

peut être orienté pour changer sa direction ( $u_2$ ), et ses capteurs sont une boussole pour connaître son cap ( $\theta$ ) et un GPS pour connaître sa position  $x, y$ .

La boussole sera considérée comme suffisamment précise donc  $\theta$  pourra être considéré comme connu, au même titre que les entrées  $u_1$  et  $u_2$ . En revanche, le GPS bon marché utilisé est souvent un peu bruité (« sauts » de position, etc.) ou ne fonctionne pas dans une piscine couverte. Pour pouvoir se localiser quand même précisément et avoir une estimation de sa précision, nous pouvons utiliser un modèle d'état simple et le fusionner avec les données GPS à l'aide d'un filtre de Kalman.

Un simulateur simple contrôlable au clavier est disponible ici : [http://www.ensta-bretagne.fr/lebars/Share/motorboat\\_simu.zip](http://www.ensta-bretagne.fr/lebars/Share/motorboat_simu.zip).

1. D'après la description et le simulateur proposé, quel est le vecteur des entrées du robot et à quoi correspondent-elles physiquement ? Quel est le vecteur d'état et à quoi correspondent physiquement les différentes variables d'état qui le constituent ? Quelle est l'équation d'évolution ? (**Durée estimée : 15 min**) [3 points]
2. Linéariser l'équation d'évolution du robot pour qu'elle devienne compatible avec les hypothèses du filtre de Kalman : donner  $A_k$ ,  $u_k$  et indiquez si vous passez par un vecteur d'état intermédiaire (souvent noté  $z_k$ ). (**Durée estimée : 20 min**) [3 points]
3. La position initiale sera considérée comme assez bien connue car en général, l'opérateur sait assez précisément où il a démarré son robot. Rajouter le code nécessaire pour indiquer que l'on connaît l'état initial du robot avec une variance de 0.1 (environ 0.5 m d'erreur avec une probabilité de 99%). (**Durée estimée : 5 min**) [1 point]
4. Supposons que l'on n'ait pas de GPS ou qu'il ne fonctionne pas (ce qui est le cas dans une piscine couverte). Ainsi, nous n'avons pas de mesures directes de  $x, y$ , donc nous n'avons pas de bruit de mesure non plus. Cependant, notre modèle étant forcément imprécis, nous avons du bruit d'état : on peut considérer par exemple qu'il est de variance 0.001 sur  $x, y, v$  (à ajuster éventuellement en fonction des tailles des ellipses qui seront dessinées par la suite). En fonction de toutes ces informations, utiliser la fonction kalman pour estimer à chaque instant la position du robot avec sa matrice de covariance associée. (**Durée estimée : 30 min**) [4 points]
5. Dessiner les ellipsoïdes de confiance contenant la position du robot avec une probabilité de 95%. Comment évolue la taille des ellipsoïdes et pourquoi ? (**Durée estimée : 10 min**) [2 points]
6. Modifier ensuite le code pour rajouter la prise en compte du GPS. Pour un GPS à bas coût, on peut considérer que le bruit de mesure est avec une variance de 2 (environ 5 m d'erreur avec une probabilité de 99%). Comment évolue la taille des ellipsoïdes et pourquoi ? (**Durée estimée : 15 min**) [2 points]
7. Rajouter le suivi autonome de waypoints (si votre code avec le Kalman ne fonctionne pas, vous pouvez juste reprendre pour base le simulateur de robot bateau à moteur [http://www.ensta-bretagne.fr/lebars/Share/motorboat\\_simu.zip](http://www.ensta-bretagne.fr/lebars/Share/motorboat_simu.zip) et supposer pour

simplifier qu'on a le droit d'accéder directement à toutes les variables d'état). (**Durée estimée : 20 min**) [**2 points**]