Evaluation : Observateurs et filtre de Kalman

- **Tout document autorisé**, mais **communication** avec d'autres personnes **interdite**. En particulier, tout accès à un **objet connecté** autre que l'ordinateur de la salle de TP (ou ordinateur personnel s'il manque des ordinateurs dans la salle) ou un **logiciel de communication** directe entre personnes ou **partage de fichiers** sera **automatiquement sanctionné sans préavis** (-5 sur la note finale, cumulable). De plus, le surveillant ne vous aidera pas.

- **Rendre code MATLAB** et autres documents informatiques sur **Moodle** dans un **dossier zip** et les **documents manuscrits** complémentaires **au surveillant**. S'assurer que les documents rendus sont bien vos versions finales. Vérifier avec le surveillant que les fichiers sont bien rendus.

- **Nouveauté 2019 :** pour lutter contre les méthodes de triche du type envoi d'une photo du sujet à une personne extérieure et réception de sa réponse dans un format numérique, **des documents manuscrits avec des explications sur ce qui est fait dans le code sont demandés**. Tous les points ne seront pas attribués si ces explications sont trop limitées ou incohérentes par rapport au code.

- **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. Ne pas utiliser un appareil photo ou smartphone !

- **Faites des sauvegardes régulières.** Surveillez régulièrement si votre PC doit redémarrer. **Attention si vous renommez des dossiers**, un **bug** les fait parfois **disparaître** !

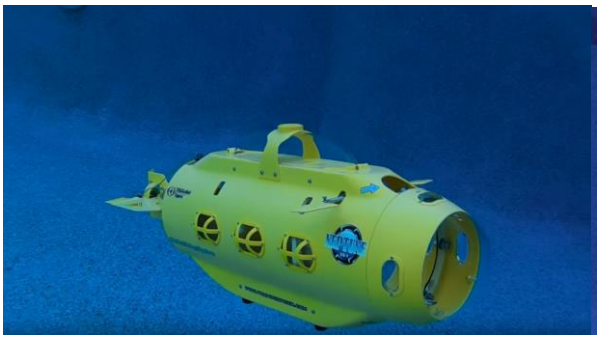
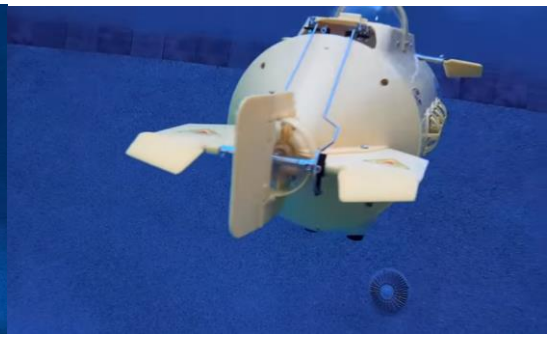
- Barème provisoire : 22 points ramenés sur 20.

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

1. Quelle utilité voyez-vous au filtre de Kalman ? [**2 points**]

2. Expliquer le rôle des différents paramètres et valeurs de retour de la fonction kalman utilisée en MATLAB. [**3 points**]

Part 2 : Filtre de Kalman et régulation du robot sous-marin Thunder Tiger Neptune SB-1 (durée estimée : 95 min)

Lors du baptême de la promo 2012, CGG, l'entreprise marraine de promo a offert au club robotique un mini robot sous-marin téléguidé de type Thunder Tiger Neptune SB-1 (voir e.g. https://youtu.be/gnb-MNwhAoY). En plus de gouvernes/ailerons latéraux à l'avant et à l'arrière permettant de changer son roulis et son tangage, et d'un ballast permettant de changer sa profondeur, ce robot sous-marin peut se déplacer dans le plan à l'aide d'une hélice propulsive à l'arrière (u1) et d'un gouvernail (u2) en face de celle-ci. En vue de son automatisation (on supposera qu'il est déjà stable en roulis et tangage et que ses mouvements verticaux sont gérés indépendamment des déplacements dans le plan), il est prévu d'y installer une boussole pour connaitre son cap (theta), un GPS pour connaitre sa position x,y lorsqu'il est en surface et un loch à hélice (hélice libre qui va tourner plus ou moins vite selon le courant généré par le déplacement du sous-marin et dont on mesure la vitesse de rotation pour en déduire la vitesse de déplacement) pour mesurer sa vitesse v.

La boussole sera considérée comme suffisamment précise donc theta pourra être considéré comme connu, au même titre que les entrées u1 et u2. En revanche, le GPS ne peut fonctionner que lorsque le sous-marin est à la surface. Pour pouvoir se localiser quand même quand on est sous l'eau et avoir une estimation de sa précision, nous pouvons utiliser un modèle d'état simple et le fusionner avec les données des capteurs (si applicable) à l'aide d'un filtre de Kalman.

Un simulateur simple contrôlable au clavier est disponible ici : http://www.ensta-bretagne.fr/lebars/Share/neptune_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. Réécrire l'équation d'évolution du robot pour qu'elle devienne compatible avec les hypothèses du filtre de Kalman : donner Ak, uk et indiquez si vous passez par un vecteur d'état intermédiaire (souvent noté zk). **(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 (notamment grâce au GPS qui fonctionne en surface). Rajouter le code nécessaire pour indiquer que l'on connait 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 soit sous l'eau et que donc le GPS ne fonctionne pas, et que de plus on n'ait pas encore installé le loch à hélice. Ainsi, nous n'avons pas de mesures directes de x,y,v 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 0.0001 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 : 20 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 : 5 min)** [**2 points**]

6. Modifier ensuite le code pour rajouter la prise en compte du loch à hélice. On pourra considérer que le bruit de mesure est avec une variance de 0.00001 (à ajuster éventuellement en fonction des tailles des ellipses qui seront dessinées par la suite). Comment devrait évoluer la taille des ellipsoïdes par rapport à précédemment et pourquoi ? **(Durée estimée : 10 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 fourni (http://www.ensta-bretagne.fr/lebars/Share/neptune_simu.zip) et supposer pour simplifier qu'on a le droit d'accéder directement à toutes les variables d'état). **(Durée estimée : 15 min)** [**2 points**]