

# Filtre de Kalman et utilisation

## Part 1 : Régulation du buggy simulé

1. Grâce à la position estimée (que l'on capte le GPS ou non), on va pouvoir faire un suivi de waypoints automatique. S'inspirer de [http://www.ensta-bretagne.fr/lebars/Share/Ecrazor\\_game.zip](http://www.ensta-bretagne.fr/lebars/Share/Ecrazor_game.zip) pour rajouter la génération de waypoints automatique dans le code du buggy simulé.
2. D'abord faire la régulation en cap (en utilisant juste la boussole) et tester son bon fonctionnement à plusieurs caps différents.
3. Pour aller dans la direction du waypoint, on va ensuite calculer le cap qu'il faudrait suivre et le passer en entrée de la régulation en cap, et on va rajouter une condition de validation qui fait passer au waypoint suivant quand on est dans un rayon de 2.5 m autour...

## Part 2 : Filtre de Kalman pour l'estimation de la position du buggy virtuel/réel et régulation

Dans un premier temps, on suppose que le buggy est en intérieur et donc que le GPS n'est pas disponible. La boussole du buggy est considérée comme très précise donc  $\theta$  est considéré comme connu, au même titre que les entrées  $u_1$  et  $u_2$ . Par contre, la position  $x, y$  est estimée par un filtre de Kalman (en mode prédicteur seulement si on n'utilise pas le GPS). La position initiale du buggy est considérée comme connue avec une variance de 0.1 (environ 0.5 m d'erreur avec une probabilité de 99%) car en général, l'opérateur sait assez précisément où il a démarré son robot. Dans le scénario actuel, 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 considérons du bruit d'état de variance 0.001 sur  $x$  et  $y$  (à ajuster en fonction des tailles des ellipses qui seront dessinées par la suite). Tout ceci a été fait lors de la dernière séance avec le buggy simulé et est dans [http://www.ensta-bretagne.fr/lebars/Share/buggy\\_simu\\_kalman.zip](http://www.ensta-bretagne.fr/lebars/Share/buggy_simu_kalman.zip).

1. Adapter le code pour le buggy virtuel/réel en partant de [http://www.ensta-bretagne.fr/lebars/Share/buggy\\_real\\_gps\\_for\\_kalman.zip](http://www.ensta-bretagne.fr/lebars/Share/buggy_real_gps_for_kalman.zip) et le tester pour contrôler le buggy virtuel/réel via MATLAB (lire ReadMe.txt pour faire les réglages nécessaires selon le buggy utilisé, bien vérifier et modifier si nécessaire les paramètres pour que le robot aille bien droit et dans la bonne direction).
2. Modifier ensuite le code pour rajouter la prise en compte du GPS (regarder les commentaires dans le code fourni). 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 90%).
3. Rajouter le suivi de waypoints automatique pour que le buggy virtuel/réel fasse en boucle un triangle (e.g. sur le terrain : 48.418318 ; -4.473576, 48.418841 ; -4.473656, 48.418467 ; -4.474332, à côté de bat. E : 48.418514, -4.472238 ; 48.418418, -

4.472498 ; 48.418612, -4.472510, ne pas oublier de convertir en coordonnées dans le repère local).